Zdeněk Dostál, Axel Klawonn, Tomáš Kozubek, Ulrich Langer, Luca F. Pavarino, Jakub Šístek, Olof B. Widlund

Editors

Domain Decomposition Methods in Science and Engineering XXVII (Preprint version)

October 2023

Preface

This volume presents a selection of 62 peer-reviewed papers that were submitted to the proceedings of the 27th International Conference on Domain Decomposition Methods held in Prague, Czech Republic, from July 25 to 29, 2022.

Background of the Conference Series

With its first meeting in Paris in 1987, the International Conferences on Domain Decomposition Methods have been held in 16 countries in Asia, Europe, and North America, and now for the first time in the Czech Republic. The conference is held at roughly 18-month intervals. A complete list of the 27 meetings appears below.

Domain decomposition is often seen as a form of the divide-and-conquer approach for mathematical problems posed over a physical domain, reducing a large problem into a collection of smaller problems, each of which is much easier to solve computationally than the undecomposed problem, and most or all of which can be solved independently and concurrently, and then solved iteratively in a consistent way. A lot of the theoretical interest in domain decomposition algorithms lies in ensuring that the number of iterations required to converge is very small. Domain decomposition algorithms can be tailored to the properties of the physical system, as reflected in the mathematical operators, to the number of processors available, and even to specific architectural parameters, such as cache size and the ratio of memory bandwidth to floating point processing rate. Consequently, domain decomposition methods prove to be an ideal paradigm for large-scale simulation on advanced parallel computers and supercomputers.

While the technical content of the conference revolves mainly around mathematics, its underlying motivation lies in enabling efficient utilization of distributed memory computers for complex scientific and engineering applications. Although research on domain decomposition methods is presented at various events, the International Conference on Domain Decomposition Methods stands as the singular recurring international forum dedicated to fostering interdisciplinary interactions

between theoreticians and practitioners. These interactions span the development, analysis, software implementation, and applications of domain decomposition methods.

As we are entering the era of exascale computing, with the most powerful supercomputers now capable of sustaining 10^{18} floating-point operations per second, the need for efficient and mathematically sound methods for solving large-scale systems becomes increasingly vital. Furthermore, these methods must align well with the modern high-performance computing (HPC) architectures. The massive parallelism inherent in exascale computing necessitates the development of new solution methods that effectively leverage the abundance of computing cores and hierarchical memory access patterns. Ongoing advancements, such as parallelization in time, asynchronous iterative methods and nonlinear domain decomposition methods show that this massive parallelism not only calls for novel solution and discretization approaches but also facilitates their further development.

Here is a list of the 27 conferences on Domain Decomposition Methods:

1. Paris, France, January 7–9, 1987
2. Los Angeles, USA, January 14–16, 1988
3. Houston, USA, March 20–22, 1989
4. Moscow, USSR, May 21–25, 1990
5. Norfolk, USA, May 6–8, 1991
6. Como, Italy, June 15–19, 1992
7. University Park, Pennsylvania, USA, October 27–30, 1993
8. Beijing, China, May 16–19, 1995
9. Ullensvang, Norway, June 3–8, 1996
10. Boulder, USA, August 10–14, 1997
11. Greenwich, UK, July 20–24, 1998
12. Chiba, Japan, October 25–20, 1999
13. Lyon, France, October 9–12, 2000
14. Cocoyoc, Mexico, January 6–11, 2002
15. Berlin, Germany, July 21–25, 2003
16. New York, USA, January 12–15, 2005
17. St. Wolfgang–Strobl, Austria, July 3–7, 2006
18. Jerusalem, Israel, January 12–17, 2008
19. Zhangjiajie, China, August 17–22, 2009
20. San Diego, California, USA, February 7–11, 2011
21. Rennes, France, June 25–29, 2012
22. Lugano, Switzerland, September 16–20, 2013
23. Jeju Island, Korea, July 6–10, 2015
24. Spitsbergen, Svalbard, Norway, February 6–10, 2017
25. St. John's, Newfoundland, Canada, July 23–27, 2018
26. Hong Kong SAR (virtual format), China, December 7–12, 2020
27. Prague, Czech Republic, July 25–29, 2022

International Scientific Committee on Domain Decomposition Methods

- Petter Bjørstad, University of Bergen, Norway
- Susanne Brenner, Louisiana State University, USA
- Xiao-Chuan Cai, CU Boulder, USA
- Martin Gander, University of Geneva, Switzerland
- Laurence Halpern, University Paris 13, France
- David Keyes, KAUST, Saudi Arabia
- Hyea Hyun Kim, Kyung Hee University, Korea
- Axel Klawonn, Universität zu Köln, Germany
- Ralf Kornhuber, Freie Universität Berlin, Germany
- Ulrich Langer, University of Linz, Austria
- Luca F. Pavarino, University of Pavia, Italy
- Olof B. Widlund, Courant Institute, USA
- Jinchao Xu, Penn State, USA
- Jun Zou, Chinese University of Hong Kong, Hong Kong

About the 27th Conference

The twenty-seventh International Conference on Domain Decomposition Methods had 200 participants (187 onsite and 13 online) from 25 different countries. The conference featured 11 invited presentations selected by the International Scientific Committee with both experienced and younger speakers, 17 minisymposia on specific topics and 6 contributed paper sessions. The present proceedings contain a selection of 62 papers, grouped into three separate groups: 4 papers by plenary speakers, 48 minisymposium papers, and 10 contributed papers.

Organizers

- VSB – Technical University of Ostrava
- Institute of Mathematics of the Czech Academy of Sciences
- Czech Technical University in Prague

Sponsoring organizations

- Hewlett Packard Enterprise (gold partner)
- Atos
- M Computers
- RSJ Foundation
- Research Center for Informatics

Local Organizing/Program Committee Members

- Zdeněk Dostál, VSB – Technical University of Ostrava, Czech Republic (Chair)
- Axel Klawonn, University of Cologne, Germany
- Tomáš Kozubek, IT4Innovations & VSB – Technical University of Ostrava, Czech Republic
- Jaroslav Kruis, Czech Technical University in Prague, Czech Republic
- Ulrich Langer, Johannes Kepler University Linz, Austria
- Daniel Langr, Czech Technical University in Prague, Czech Republic
- Jakub Šístek, Institute of Mathematics of the Czech Academy of Sciences, Prague, Czech Republic

Plenary Presentations

- Silvia Bertoluzza (CNR — Istituto di Matematica Applicata e Tecnologie Informatiche “Enrico Magenes")
  Domain decomposition for the Virtual Element Method
- Xiao-Chuan Cai (Department of Mathematics, University of Macau)
  Schwarz for complex fluid and solid problems in biomechanics
- Alexander Heinlein (Delft University of Technology)
  Robust, algebraic, and scalable Schwarz preconditioners with extension-based coarse spaces
- Florence Hubert (Institut de Mathematiques de Marseille, Aix-Marseille Université, France)
  On discrete optimized Schwarz algorithms for elliptic problems
- Hyea Hyun Kim (Kyung Hee University, Korea)
  Domain decomposition algorithms for neural network approximation of partial differential equations
- Maksymilijan Dryja (the winner of the Olof Widlund Prize), a talk presented by Marcus Sarkis (University of Warsaw / Worcester Polytechnic Institute)
  NOSAS and RAS/ASH
- Robert Scheichl (Heidelberg University, Germany)
  Multiscale Generalised Finite Element Methods
- Jonathan W. Siegel (Pennsylvania State University/Texas A&M University)
  Approximation Properties of Neural Networks and Applications to Numerical PDEs
- Jakub Šístek (Institute of Mathematics of the Czech Academy of Sciences)
  Applications of multilevel BDDC to problems of incompressible flows
- Barbara Wohlmuth (Technical University of Munich)
  Multi-physics models with mixed dimensions: Bio-medical and seismic applications
- Stefano Zampini (King Abdullah University of Science and Technology, Saudi Arabia)
  Device Accelerated solvers with PETSc: current status, future perspectives, and applications

Acknowledgments

The organizers would like to thank all the participants for their enthusiasm and carefully prepared contributions that made this meeting a very successful event. A warm thanks also to our sponsors that made the budget come together. We have experienced a unique meeting, resuming to a mostly in-person event after a break caused by the COVID 19 pandemic. The AMCA agency helped with the organization of the conference. Mrs Hana Bílková was responsible for the technical editing of the contributions.

Prague, October 2023

Zdeněk Dostál
VSB – Technical University of Ostrava, Czech Republic

Axel Klawonn
University of Cologne, Germany

Tomáš Kozubek
IT4Innovations & VSB – Technical University of Ostrava, Czech Republic

Ulrich Langer
Johannes Kepler University Linz, Austria

Luca F. Pavarino
University of Pavia, Italy

Jakub Šístek
Institute of Mathematics of the Czech Academy of Sciences, Prague, Czech Republic

Olof B. Widlund
Courant Institute, USA

Contents

List of Contributors

Stefano Berrone
Politecnico di Torino, Italy, e-mail: stefano.berrone@polito.it

Sven Beuchler
Leibniz University Hannover, Institute of Applied Mathematics, Welfengarten 1, 30167 Hannover, Germany, e-mail: beuchler@ifam.uni-hannover.de

Eric Blayo
Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LJK, Grenoble, France, e-mail: eric.blayo@univ-grenoble-alpes.fr

Marcella Bonazzoli
Inria, UMA, ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France, e-mail: marcella.bonazzoli@inria.fr

Niall Bootland
STFC Rutherford Appleton Laboratory, Harwell Campus, UK, e-mail: niall.bootland@stfc.ac.uk

Sahar Borzooei
University Côte d'Azur, CNRS, LJAD, France, e-mail: Sahar.Borzooei@univ-cotedazur.fr

Ignace Bossuyt
Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium, e-mail: ignace.bossuyt1@kuleuven.be

Miranda Boutilier
Université Côte d'Azur, LJAD, France, e-mail: miranda.boutilier@univ-cotedazur.fr

Marek Brandner
University of West Bohemia in Pilsen, Faculty of Applied Sciences, Univerzitní 22, Pilsen, Czech Republic, e-mail: brandner@kma.zcu.cz

Konstantin Brenner
Université Côte d'Azur, LJAD, CNRS, INRIA, France, e-mail: konstantin.brenner@univ-cotedazur.fr

Duc Quang Bui
Université Sorbonne Paris Nord, Villetaneuse, France, e-mail: bui@math.univ-paris13.fr

Juan G. Calvo
CIMPA – Escuela de Matemática, Universidad de Costa Rica, Costa Rica, e-mail: juan.calvo@ucr.ac.cr

Faycal Chaouqui
COMSOL, INC, Burlington, Mass., USA, e-mail: chaouqui@temple.edu

Bastien Chaudet-Dumas
University of Geneva, Switzerland e-mail: bastien.chaudet@unige.ch

Laurent Chédot
SuperGrid-Institute, 23 rue Cyprian, 69100 Villeurbanne, e-mail: laurent.chedot@supergrid-institute.com

Gabriele Ciaramella
MOX, Dipartimento di Matematica, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy, e-mail: gabriele.ciaramella@polimi.it

Xavier Claeys
Sorbonne Université, CNRS, Université Paris Cité, LJLL, Paris, France, e-mail: xavier.claeys@sorbonne-universite.fr

Simon Clement
Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LJK, Grenoble, France, e-mail: simon.clement2@univ-grenoble-alpes.fr

Pierre-Henri Cocquet
Université de Pau et des Pays de l'Adour, SIAME, rue de l'Université, Pau, France, e-mail: pierre-henri.cocquet@univ-pau.fr

Eric C. Cyr
Sandia National Laboratories, Albquerque, NM, 87123, e-mail: eccyr@sandia.gov

Ruiyang Dai
Laboratory J.L. Lions, Sorbonne Université, Paris, France, e-mail: ruiyang.dai@upmc.fr

Bérangère Delourme
Université Sorbonne Paris Nord, Villetaneuse, France, e-mail: delourme@math.univ-paris13.fr

Marco Discacciati
Department of Mathematical Sciences, Loughborough University, Epinal Way, Loughborough, LE11 3TU, United Kingdom, e-mail: m.discacciati@lboro.ac.uk

Victorita Dolean
University of Strathclyde, Dept. of Maths and Stats and Université Côte d'Azur, LJAD, CNRS, France, e-mail: work@victoritadolean.com

Maksymilian Dryja
University of Warsaw, Poland. e-mail: dryja@mimuw.edu.pl

Abdeladim El Akri
Lab. MSDA, Mohammed VI Polytechnic University, Green City, Morocco e-mail: abdeladim.elakri@um6p.ma

Stephanie Friedhoff
Bergische Universität Wuppertal, e-mail: friedhoff@math.uni-wuppertal.de

José Galaz
INRIA, Team LEMON, Centre Inria d'Université Côte d'Azur, Antenne Montpellier, Bat 5 CC05 017, 860 rue Saint-Priest, 34095 Montpellier Cedex 5 France, e-mail: jose.galaz@inria.fr

Marco Gambarini
MOX, Dipartimento di Matematica Politecnico di Milano, Italy, e-mail: marco.gambarini@polimi.it

Martin J. Gander
University of Geneva, rue du Conseil-Général 7-9, Geneva, Switzerland e-mail: martin.gander@unige.ch

Peter Gangl
Johann Radon Institute for Computational and Applied Mathematics, Altenberger Straße 69, 4040 Linz, Austria, e-mail: peter.gangl@oeaw.ac.at

Gobinda Garai
School of Basic Sciences (Mathematics), Indian Institute of Technology Bhubaneswar, India, e-mail: gg14@iitbbs.ac.in

Mario Gobrial
Institute of Applied Mathematics, TU Graz, Steyrergasse 30, 8010 Graz, Austria e-mail: gobrial@math.tugraz.at

Viktor Grimm
Department of Mathematics and Computer Science, University of Cologne, Weyertal 86-90, 50931 Köln, Germany, e-mail: viktor.grimm@uni-koeln.de

Laurence Halpern
Université Sorbonne Paris Nord, Villetaneuse, France, e-mail: halpern@math.univ-paris13.fr

Martin Hanek
Institute of Mathematics of the Czech Academy of Sciences, Žitná 25, Prague, Czech Republic and Czech Technical University in Prague, Technická 4, Prague, Czech Republic, e-mail: martin.hanek@fs.cvut.cz

Alexander Heinlein
Delft University of Technology, Delft Institute of Applied Mathematics, Mekelweg 4, 2628 CD Delft, Netherlands e-mail: a.heinlein@tudelft.nl

César Herrera
Purdue University, West Lafayette, IN 47907, e-mail: herre125@purdue.edu

Florence Hubert
Aix-Marseille Université, CNRS, e-mail: florence.hubert@univ-amu.fr

Khale Jbilou
Lab. MSDA, Mohammed VI Polytechnic University, Green City, Morocco and Lab. LMPA, University of Littoral Côte d'Opale, Calais cedex, France, e-mail: khalide.jbilou@univ-littoral.fr

Maria Kazolea
INRIA, Team CARDAMOM, INRIA Bordeaux-Sud-Ouest, 200 Avenue de la Vieille Tour, 33405 Talence cedex France e-mail: maria.kazolea@inria.fr

Amirreza Khodadadian
Leibniz University Hannover, Institute of Applied Mathematics and Cluster of Excellence PhoenixD, Welfengarten 1, 30167 Hannover, Germany, e-mail: khodadadian@ifam.uni-hannover.de

Hyea Hyun Kim
Department of Applied Mathematics and Institute of Natural Sciences, Kyung Hee University, Korea, e-mail: hhkim@khu.ac.kr

Sebastian Kinnewig
Leibniz University Hannover, Institute of Applied Mathematics, Welfengarten 1, 30167 Hannover, Germany, e-mail: kinnewig@ifam.uni-hannover.de

Axel Klawonn
Department of Mathematics and Computer Science, University of Cologne, Weyertal 86-90, 50931 Köln, Germany and Center for Data and Simulation Science, University of Cologne, Germany, e-mail: axel.klawonn@uni-koeln.de

Tobias Knoke
Leibniz University Hannover, Institute of Applied Mathematics, Welfengarten 1, 30167 Hannover, Germany, e-mail: tobias.knoke@stud.uni-hannover.de

Stephan Köhler
Institut für Numerische Mathematik und Optimierung, Fakultät für Mathematik und Informatik, Technische Universität Bergakademie Freiberg, 09596 Freiberg, e-mail: stephan.koehler@math.tu-freiberg.de

Alena Kopaničáková
Brown University, USA, e-mail: alena.kopanicakova@brown.edu

Hardik Kothari
Università della Svizzera italiana, Switzerland, e-mail: hardik.kothari@usi.ch

Rolf Krause
Università della Svizzera italiana, Switzerland, e-mail: rolf.krause@usi.ch

Stella Krell
Université Côte d'Azur, Inria, CNRS, LJAD, e-mail: stella.krell@univ-cotedazur.fr

Miroslav Kuchta
Simula Research Laboratory, Kristian Augusts gate 23, 0164 Oslo, e-mail: miroslav@simula.no

Niteen Kumar
Section de Mathématiques, Université de Genève, e-mail: niteen.kumar@unige.ch

Felix Kwok
Université Laval, Québec, Canada, e-mail: felix.kwok@mat.ulaval.ca

Lahcen Laayouni
School of Science and Engineering, Al Akhawayn University, Avenue Hassan II, 53000, P.O. Box 1630, Ifrane, Morocco, e-mail: L.Laayouni@aui.ma

Martin Lanser
Department of Mathematics and Computer Science, University of Cologne, Weyertal 86-90, 50931 Köln, Germany, e-mail: martin.lanser@uni-koeln.de

Pilhwa Lee
Department of Mathematics, Morgan State University, 1700 E. Cold Spring Lane, Baltimore, MD, USA, e-mail: Pilhwa.Lee@morgan.edu

Florian Lemarié
Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LJK, Grenoble, France, e-mail: florian.lemarie@inria.fr

Liu-Di Lu
University of Geneva, Switzerland, e-mail: liudi.lu@unige.ch

Bankim C. Mandal
School of Basic Sciences (Mathematics), Indian Institute of Technology Bhubaneswar, India, e-mail: bmandal@iitbbs.ac.in

Leszek Marcinkowski
Faculty of Mathematics, University of Warsaw, Banacha 2, 02-097 Warszawa, Poland, e-mail: Leszek.Marcinkowski@mimuw.edu.pl

Véronique Martin
UMR CNRS 7352, Université de Picardie Jules Verne, Amiens, France, e-mail: veronique.martin@u-picardie.fr

Per-Gunnar Martinsson
Oden Institute, e-mail: pgm@oden.utexas.edu

Ilario Mazzieri
MOX, Politecnico di Milano, Italy, e-mail: ilario.mazzieri@polimi.it

Conor McCoid
Université Laval, e-mail: conor.mccoid.1@ulaval.ca

Edie Miglio
MOX, Dipartimento di Matematica Politecnico di Milano, Italy, e-mail: edie.miglio@polimi.it

Siddhartha Mishra
ETH Zürich, Computational and Applied Mathematics Laboratory / ETH AI Center, Rämistrasse 101, 8092 Zürich, Switzerland, e-mail: siddhartha.mishra@sam.math.ethz.ch

Ben Moseley
ETH Zürich, Computational and Applied Mathematics Laboratory / ETH AI Center, Rämistrasse 101, 8092 Zürich, Switzerland, e-mail: benjamin.moseley@ai.ethz.ch

Frédéric Nataf
Sorbonne Université, CNRS, Université Paris Cité, LJLL, Paris, France, e-mail: frederic.nataf@sorbonne-universite.fr

Nima Noii
Leibniz University Hannover, Institute of Continuum Mechanics, An der Universität 1, 30823 Garbsen, Germany e-mail: noii@ikm.uni-hannover.de

Nouredine Ouhaddou
Lab. MSDA, Mohammed VI Polytechnic University, Green City, Morocco e-mail: nouredine.ouhaddou@um6p.ma

Michal Outrata
University of Geneva, e-mail: michal.outrata@unige.ch

Maryam Parvizi
Leibniz University Hannover, Institute of Applied Mathematics, Welfengarten 1, 30167 Hannover, Germany, e-mail: parvizi@ifam.uni-hannover.de

Luca F. Pavarino
Dipartimento di Matematica, Università degli Studi di Pavia, via Ferrata 5, 27100 Pavia, Italy, e-mail: luca.pavarino@unipv.it

Aušra Pogoželskytė
University of Geneva, rue du Conseil-Général 7-9, Geneva, e-mail: ausra.pogozelskyte@unige.ch

Talal Rahman
Faculty of Engineering and Science, Western Norway University of Applied Sciences, Inndalsveien 28, 5063 Bergen, Norway, e-mail: Talal.Rahman@hvl.no

Ahmed Ratnani
MSDA-Modeling Simulation and Data Analysis, Mohammed VI Polytechnic University, Green City, Morocco, e-mail: ahmed.ratnani@um6p.ma

Oliver Rheinbach
Institut für Numerische Mathematik und Optimierung, Fakultät für Mathematik und Informatik, Technische Universität Bergakademie Freiberg, 09596 Freiberg, e-mail: oliver.rheinbach@math.tu-freiberg.de

Abdessadek Rifqui
MSDA-Modeling Simulation and Data Analysis, Mohammed VI Polytechnic University, e-mail: abdessadek.rifqui@um6p.ma

Daniel J. Rixen
Chair of Applied Mechanics, TUM School of Engineering and Design, Technical University of Munich, Boltzmannstr. 15 D-85748 Garching, e-mail: rixen@tum.de

Jake Robinson
Department of Mathematical Sciences, Loughborough University, Epinal Way, Loughborough, LE11 3TU, United Kingdom

Antoine Rousseau
INRIA, Team LEMON, Centre Inria d'Université Côte d'Azur, Antenne Montpellier, Bat 5 CC05 017, 860 rue Saint-Priest, 34095 Montpellier Cedex 5 France, e-mail: antoine.rousseau@inria.fr

Giovanni Samaey
Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium, e-mail: giovanni.samaey@kuleuven.be

Marcus Sarkis
Worcester Polytechnic Institute, 100 Institute Rd, Worcester, USA, e-mail: msarkis@wpi.edu

Simone Scacchi
Dipartimento di Matematica, Università degli Studi di Milano, via Saldini 50, 20133 Milano, Italy, e-mail: simone.scacchi@unimi.it

Andreas S. Seibold
Chair of Applied Mechanics, TUM School of Engineering and Design, Technical University of Munich, Boltzmannstr. 15 D-85748 Garching, e-mail: andreas.seibold@tum.de

Filánder A. Sequeira
Escuela de Matemática, Universidad Nacional, Heredia, Costa Rica, e-mail: filander.sequeira@una.cr

Hélena Shourick
SuperGrid-Institute, 23 rue Cyprian, 69100 Villeurbanne, University of Lyon, UMR5208 U.Lyon1-CNRS, Institut Camille Jordan, e-mail: helena.shourick@supergrid-institute.com

Jakub Šístek
Institute of Mathematics of the Czech Academy of Sciences, Žitná 25, Prague, Czech Republic, e-mail: sistek@math.cas.cz

Olaf Steinbach
Institute of Applied Mathematics, TU Graz, Steyrergasse 30, 8010 Graz, Austria
e-mail: o.steinbach@tugraz.at

Daniel B. Szyld
Department of Mathematics, Temple University (038-16), 1805 N. Broad Street, Philadelphia, Pennsylvania 19122-6094, USA, e-mail: szyld@temple.edu

Pierre-Henri Tournier
Sorbonne Université, CNRS, Université Paris Cité, Inria, Laboratoire Jacques-Louis Lions (LJLL), F-75005 Paris, France, e-mail: pierre-henri.tournier@sorbonne-universite.fr

Damien Tromeur-Dervout
University of Lyon, UMR5208 U.Lyon1-CNRS, Institut Camille Jordan, e-mail: damien.tromeur-dervout@univ-lyon1.fr

Serge Van Criekingen
Institut du Développement et des Ressources en Informatique Scientifique (IDRIS), CNRS, Université Paris-Saclay, F-91403, Orsay, France and Université Paris-Saclay, UVSQ, CNRS, CEA, Maison de la Simulation, 91191, Gif-sur-Yvette, France, e-mail: serge.van.criekingen@idris.fr

Stefan Vandewalle
Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium, e-mail: stefan.vandewalle@kuleuven.be

Tommaso Vanzan
Ecole Polytecnique Fédérale de Lausanne, Switzerland, e-mail: tommaso.vanzan@epfl.ch

A. Vieira
Université de la Réunion, PIMENT, Sainte-Clotilde, France, e-mail: alexandre.vieira@univ-reunion.fr

Yiying Wang
School of Mathematics, Jilin University, P. R. China, e-mail: yiyingw20@mails.jlu.edu.cn

Adam Wasiak
Department of Mathematics and Computer Science, Division of Mathematics, University of Cologne, Weyertal 86-90, 50931 Cologne, Germany, e-mail: adam.wasiak@uni-koeln.de

Janine Weber
Department of Mathematics and Computer Science, University of Cologne, Weyertal 86-90, 50931 Köln, Germany, e-mail: janine.weber@uni-koeln.de

Thomas Wick
Leibniz University Hannover, Institute of Applied Mathematics and Cluster of Excellence PhoenixD, Welfengarten 1, 30167 Hannover, Germany, e-mail: thomas.wick@ifam.uni-hannover.de

Olof B. Widlund
Courant Institute, 251 Mercer Street, New York, NY 10012, USA, e-mail: widlund@cims.nyu.edu

Yingxiang Xu
School of Mathematics and Statistics, Northeast Normal University, P. R. China, e-mail: yxxu@nenu.edu.cn

Hee Jun Yang
Department of Mathematics, Kyung Hee University, Korea, e-mail: yhjj109@khu.ac.kr

Anna Yesypenko
Oden Institute, e-mail: annayesy@utexas.edu

Yi Yu
Guangxi University, Nanning, Guangxi, P. R. China, e-mail: yiyu@gxu.edu.cn

Stefano Zampini
Extreme Computing Research Center, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia, e-mail: stefano.zampini@kaust.edu.sa

Part I
Plenary Talks (PT)

A Short Note on Solving Partial Differential Equations Using Convolutional Neural Networks

Viktor Grimm, Alexander Heinlein, and Axel Klawonn

1 Introduction

Solving partial differential equations (PDEs) is a common task in numerical mathematics and scientific computing. Typical discretization schemes, for example, finite element (FE), finite volume (FV), or finite difference (FD) methods, have the disadvantage that the computations have to be repeated once the boundary conditions (BCs) or the geometry change slightly; typical examples requiring the solution of many similar problems are time-dependent and inverse problems or uncertainty quantification. Every single computation, however, can be very time consuming, motivating the development of surrogate models that can be evaluated quickly. There exist some possible surrogate models, including linear reduced order models [9, 21, 26, 29] and neural network-based models [6, 7, 8, 14, 19, 22, 24, 25].

In this work, we will discuss an approach for predicting the solution of boundary value problems using convolutional neural networks (CNNs). This approach is particularly interesting in the context of surrogate models which predict the solution based on a parametrization of the model problem, for instance, with respect to variations in the geometry or BCs; cf. Fig. 1 for a sketch of the CNN-based surrogate modeling approach. If the parametrization is high-dimensional, that is, if it consists of a large number of parameters, neural network-based approaches are particularly well-suited since they are know to be able to overcome the curse of dimensional-

Viktor Grimm
Department of Mathematics and Computer Science, University of Cologne, Weyertal 86-90, 50931 Köln, Germany, e-mail: viktor.grimm@uni-koeln.de

Alexander Heinlein
Delft University of Technology, Delft Institute of Applied Mathematics, Mekelweg 4, 2628 CD Delft, Netherlands e-mail: a.heinlein@tudelft.nl

Axel Klawonn
Department of Mathematics and Computer Science, University of Cologne, Weyertal 86-90, 50931 Köln, Germany e-mail: axel.klawonn@uni-koeln.de
Center for Data and Simulation Science, University of Cologne, Germany

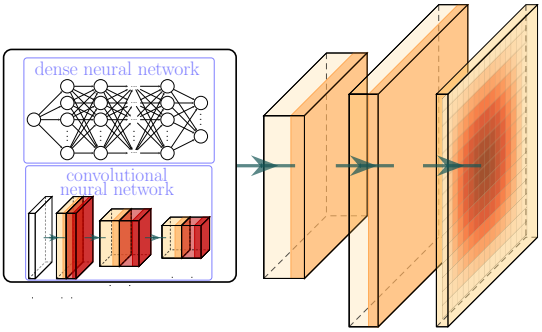


Fig. 1 Exemplary CNN-based surrogate model. The first block transforms the problem parametrization into a low-dimensional representation (latent representation) of the solution, and the right part of the model decodes the corresponding image of the solution field.

ity [4, 17]. In [5, 6, 15], a CNN model has been trained to predict stationary flow inside a channel with an obstacle of varying geometry; the model is trained in a purely data-based way using high-fidelity simulation data.

Here, we use a physics-based loss function in the CNN approach, that is, we optimize the network with respect to the residual of the partial differential equation (PDE) as well as the BCs of the BVP; this is also denoted as physics-informed or physics-aware machine learning (ML). Therefore, our approach is related to physics-informed neural networks (PINNs), which have been introduced in [28] and are an extension of the pioneering work [20]. However, different from [20, 28], we employ a finite difference-based discretization inside the loss function and predict the coefficients using a CNN. In the classical PINN approach, however a dense neural network (DNN) is employed as the discretization, and the derivatives are computed exactly via the backpropagation algorithm.

Physics-informed CNN approaches have already been considered. In particular, in [31], a model for predicting the solutions of the stationary diffusion equation for a single fixed geometry but varying BCs, encoded as an input image, is proposed. In [10], the authors employ a physics-based CNN model for predicting incompressible Navier–Stokes flow in parameterized geometries that is, the exact placement of the boundaries of the geometries depend on a parameter. More recently, the authors of this work have extended the previous approaches to a physics-aware CNN for predicting incompressible Navier–Stokes flow in more general geometries and also varying boundary conditions; cf. [13]. For further works on CNN-based surrogate models for the approximating the solutions of PDE, see, for instance, [7, 8, 11, 22, 25]. Furthermore, for scientific machine learning (SciML) overview papers with a broader scope and additional references on related approaches, we refer to [3, 35].

In this paper, we will compare the accuracy and convergence of a CNN model, optimized using a (stochastic) gradient descent-type method using a physics-based loss function, with a classical FD discretization, solving the resulting discrete linear

system of equations using an (unpreconditioned) conjugate gradient (CG) method, for a simple stationary diffusion problem. In order to focus on these aspects and remove any other complexities, we focus on a single problem configuration, that is, we neglect the encoder part in Fig. 1 and focus on training the decoder path. The paper is organized as follows: In Section 2, we introduce our stationary diffusion model problem and the simple difference discretization employed. Then, in Section 3, we briefly discuss how to solve the resulting discrete system of equations using the CG method as well as how to optimize a CNN model for predicting the same solution. Finally, we compare the performance of both solution frameworks with respect to accuracy and convergence in Section 5.

2 Model problem and discretization

Finite difference discretization

Let us consider a simple stationary diffusion problem on computational domain $\Omega := [0,1]^2$: find a function u, such that

$$\begin{aligned} -\Delta u &= f \qquad \text{in } \Omega, \\ u &= 0 \qquad \text{on } \partial\Omega, \end{aligned} \tag{1}$$

where f is some right hand side function. We discretize (1) using FDs. In particular, we consider a uniform grid $\Omega_h = \{(x_i, y_j)_{i,j}\}$ with $x_i := ih$ and $y_j := jh$, the step size $h = 1/n$, and $u_{i,j} := u(x_i, y_j)$. Using a central difference scheme, we obtain the following approximation of the Laplacian:

$$\Delta u(x_i) \approx \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2}. \tag{2}$$

Hence, the discrete form of (1) corresponds to the sparse system of linear equations

$$Au = f. \tag{3}$$

with a symmetric positive definite (SPD) matrix. Here, for simplicity, we use the same symbol for the solution and right hand side as in (1).

Reformulation of the finite difference problem via the cross-correlation

Before we explain our physics-based network model, let us discuss how (3) can be written equivalently using the cross-correlation operation

$$(I * K)_{ij} = \sum_m \sum_n I_{i-m,j-n} K_{m,n},$$

where I and K are two matrices. For simplicity, we omit the the range of the sums, and regard each matrix coefficient as zero which is outside the range of indices. Note that the discrete convolution and cross-correlation operations are related in the sense that one can be obtained from the other by transposition. Moreover, the cross corelation is actually implemented as the operation of convolutional layers in NN libraries; cf. [12, Section 9.1].

Now, let $U = \left(u_{i,j}\right)_{i,j}$ and $F = \left(f(x_i, y_j)\right)_{i,j}$ be $n \times n$ matrices resulting from re-arranging the solution and right hand side vectors in (3). Then, we obtain

$$Au = f \quad \Leftrightarrow \quad U * K = F, \tag{4}$$

where $*$ is the cross-correlation operation and K is given by

$$K = \frac{1}{h^2}\begin{pmatrix} K_{-1,-1} & K_{-1,0} & K_{-1,1} \\ K_{0,-1} & K_{0,0} & K_{0,1} \\ K_{1,-1} & K_{1,0} & K_{1,1} \end{pmatrix} = \frac{1}{h^2}\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \tag{5}$$

which is also denoted as the kernel matrix or filter. This can be easily seen by comparing the coefficients in (2), (5). Only for enforcing the boundary conditions for certain coefficients or pixels, respectively, the kernel K has to be modified, as is standard in the implementation of boundary conditions in finite difference discretizations.

3 Solving the finite difference problem using classical methods versus using convolutional neural networks

Efficient classical numerical solvers

Since our model problem, that is, stationary diffusion on the unit square, is arguably one of the most investigated problems for the development of solvers, there is a wide range of efficient solvers for (3). Hence, we keep this discussion rather short. A standard solver for systems with an SPD matrix is the conjugate gradient (CG) method. The convergence of the CG method is determined by the spectrum of the matrix, and in particular, it can be bounded in terms of the condition number of the system matrix A, which scales with $\frac{1}{h^2}$ for our model problem. The h dependence of the convergence of the CG method can be fixed by acceleration using preconditioners, such as domain decomposition [32] and multigrid [33] methods, to name just two popular classes of efficient and scalable preconditioners for (3).

For the purpose of comparing numerical solvers against a closely related ML approach for solving a stationary problem, we will use the CG method without preconditioning as the prototypical solver. It would be interesting to include state-of-the-art preconditioners in our study and discuss if and how preconditioning could be applied in the optimization of the CNNs. However, this is out of the scope of this short paper, and therefore, we will leave this to future research.

A finite difference solver based on convolutional neural networks

Solving (3) corresponds to finding the coefficients $u_{i,j}$, which are structured based on the uniform grid $\Omega_h = \{(x_i, y_j)_{i,j}\}$. We can simply interpret the discrete solution as a pixel image, with each pixel corresponding to one coefficient in the solution vector u. Hence, in several works, CNNs, which are very effective in image processing, have been trained to learn the discrete solution of a partial differential equation; cf. Fig.1 for a sketch of this approach and the discussion below. In practice, as we will also see in Section 5, this approach is not competitive for solving a single BVP. However, when used as a reduced order model for a parametrized model problem (e.g., with respect to the geometry), the higher computing costs for the training can be justified if the solutions of multiple BVPs can be predicted using a single model.

Here, we focus on training a neural network using a physics-informed, sometimes also referred to as physics-aware or physics-constraint, approach. Then, a neural network $\mathcal{NN}$ is trained to minimize the norm of the residual of the differential equation, i.e.,

$$\|\Delta\mathcal{NN} + f\|_\Omega^2 + \|\mathcal{NN}\|_{\partial\Omega}^2 \to \min,$$

where $\|\cdot\|_\Omega$ and $\|\cdot\|_{\partial\Omega}$ are some norms defined based on collocation points inside the domain Ω and on the boundary $\partial\Omega$; as mentioned in Section 1, this corresponds to the classical PINN approach if a dense NN is used as the discretization. If the output of the neural network corresponds to an image, that is, if the output data is a discrete vector on the uniform grid $\Omega_h = \{(x_i, y_j)_{i,j}\}$, we can employ an FD scheme to formulate the residual of the PDE, resulting in

$$\|b - A \cdot \mathcal{NN}\|_2^2 \to \min, \tag{6}$$

where the term corresponding to the boundary conditions vanishes since they are hard-coded within the matrix A. Note that this can be efficiently implemented in state-of-the-art ML libraries, such as Tensorflow: the matrix A does not have to be assembled, but it can be applied in a matrix-free fashion by using the FD stencil (2) as a fixed kernel in a convolutional layer and applying it to the output of the network; cf. the discussion in Section 2.

We note that solving (6) directly for u is equivalent to solving the least-squares problem corresponding to (3), which amounts to solving the normal equations

$$A^\top A u = A^\top b. \tag{7}$$

The system matrix $A^\top A$ is still SPD, so (7) can also be solved using the CG method. However, the convergence will be much slower, as the condition number

$$\kappa\left(A^\top A\right) = \kappa\left(A\right)^2.$$

The situation is changed further once u is replaced by a neural network $\mathcal{NN}$. Hence, minimizing the loss function with respect to the network parameters θ does not correspond to solving a linear system anymore. Moreover, the loss function is, in

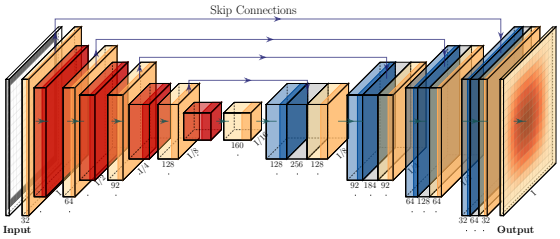


Fig. 2 Exemplary model architecture with a depth of four levels, resulting in 8×8 feature maps on the deepest level. Each level is composed of convolutions (orange), strided convolution (red), upsampling (blue) and/or concatenation (grey) layers. In total, this model has 834 627 parameters.

general, not even a convex function with respect to the network parameters anymore. Thus, in addition to solving a problem (6) that has a significantly worse conditioning than the original problem (3), we cannot use the CG method let alone another Krylov subspace method anymore.

Minimizing (6) with respect to the network parameters, which is also denoted as training the neural network, is usually performed using either a variant of stochastic gradient descent (SGD), such as the Adam (adaptive moments) optimizer [18], or a second order quasi-Newton method, such as L-BFGS [23]. Those optimizers and their parameters are typically chosen based on heuristics, which clearly shows that, at this point, we have lost most of the properties of the original problem (3) beneficial for a numerical solver.

Extension to more complex problems

Even though, in this paper, we focus on a linear problem on a simple square domain, our approach can be extended to nonlinear problems on more general geometries in a straight-forward way. In particular, the linear operator A in (6) can be easily replaced by a nonlinear operator F, which yields the minimization problem

$$\|b - F(\mathcal{NN})\|_2^2 \to \min. \tag{8}$$

In particular, in the CNN approach for a nonlinear PDE, the operator F corresponds to the finite difference discretization of the nonlinear differential operator of the PDE; cf. [13] for the application to the Navier–Stokes equations. Even though it cannot be directly implemented using a simple cross-correlation anymore, it can typically be written as a composition of cross-correlations and element-wise tensor-operations. Hence, it can still be easily and efficiently implemented using optimized functions from state-of-the-art deep learning libraries. To extend the approach to more complex geometries, boundary conditions have to be implemented for the corresponding output coefficients or pixels, respectively. A parametrization of the problem, for instance, with the respect to the geometry, can be incorporated via the input of the CNN; cf. Figs. 1 and 2. For more details, we refer to [13].

4 Network architecture and hyper parameters

As is usual in the context of NNs, the training performance and prediction accuracy of model strongly depend on the choice of the hyperparameters, which include the specific network architecture and parameters of the optimizer. In advance of our numerical study, we have carried out a detailed hyperparameter optimization to obtain a good performance of the CNN models. In particular, we used the optimized model for more complex computational fluid dynamics problems with varying geometries; cf. [13]. Similar to [5, 6, 15], in [13], the CNN model is employed as a reduced order surrogate model for varying geometries. As a result of the hyper parameter optimization, we ended up using an architecture which is inspired by the U-Net [30]; cf. Fig. 2. The model is composed of an encoder and a decoder part, each consisting of several levels. The corresponding levels of the encoder and decoder are connected with skip connections. Here, each level of the encoder part consists of a convolutional layer with an increasing number of 3×3 filters and a downsizing convolutional layer with 2×2 filters and stride of 2. In the decoder part, each level consists of a normal 3×3 convolutional layer, a concatenation layer for the skip connections and an up-sampling through nearest-neighbor interpolation layer.

In the hyper parameter optimization, we varied the activation function, the number of filters in the convolutional layers as well as the number of levels of the U-Net type architecture. Moreover, we performed numerical experiments for different learning rates, indicating the best performance for GD with a learning rate of 10^{-5} and for Adam with a learning rate of $5.0 \cdot 10^{-5}$. For more details on the hyper parameter optimization, we refer to [13].

In this paper, we focus on the effect of different solvers rather than the effect of different choices of the neural network architecture. In this sense, our major concern was to obtain a model architecture which is sufficient for approximating the solution of our the considered model problem. As we can observe based on the results in Section 5, this is the case for our model. In fact, the number of parameters and the model capacity could probably be reduced significantly for this model problem, at the cost of an additional hyper parameter optimization. Of course, a variation of the hyperparameters could have some impact on the convergence results in Section 5 but it is not obvious how to take the hyperparameter optimization into account in the comparison in a fair way. Moreover, we do not expect a major difference in the performance of the different approaches when varying the hyper parameters.

5 Numerical results

In this section, we compare different solution methods for an FD discretization of (1). In particular, we employ the gradient descent (GD) and conjugate gradient (CG) methods for the original equations (3) as well as the normal equations (7) arising from a least-squares formulation of the problem. We compare those results against training a CNN to predict the coefficient vector using the GD and Adam [18] meth-

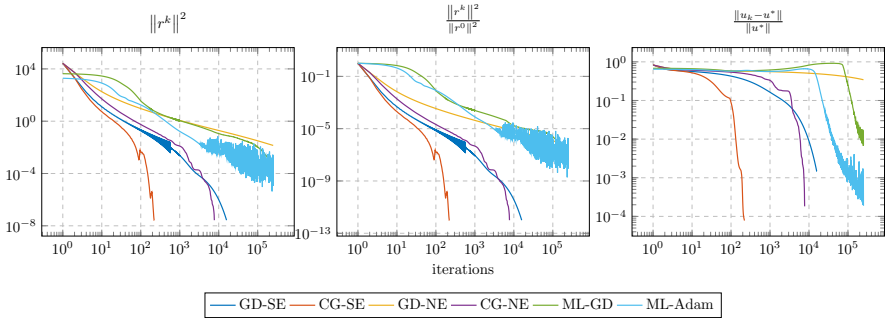


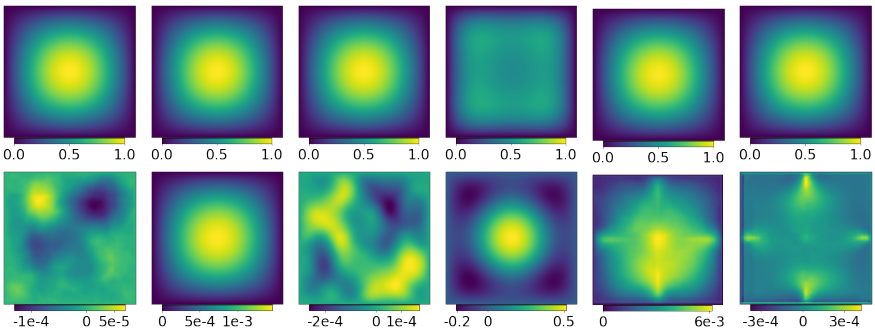
Fig. 3 Convergence of the GD, CG and Adam methods for the original linear equation system (3) and the least-squares problem eq. (6) for the FD discretization u and the CNN u_{NN}. Comparison of the absolute and relative residuum $\|r_k\|^2 \,/\, \|r_0\|^2$ where $r_k = b - Au_k$, and the relative error $\|u_k - u^*\| \,/\, \|u^*\|$.

ods for the physics-informed loss function, which corresponds to the least-squares formulation (6). All CNN computations were performed on NVIDIA V100-GPUs with CUDA 10.1 using python 3.6 and tensorflow-gpu 2.4 [1].

For our experiments, we choose $f = 2\pi^2 \sin(\pi x)\sin(\pi y)$ as the right hand side. The resulting BVP has the analytical solution $u^* = \sin(\pi x)\sin(\pi y)$, which we use as the reference. In this work, we exlusively consider an FD discrezation of the computational domain Ω with $N = 128$ grid nodes in each direction; this results in a total problem size of 16 384 nodes or degrees of freedom, respectively. For the classical methods, we use a fixed but random initial guess, the parameters of the CNNs are randomly initialized using the He normal initialization [16]. We compare the convergence of the methods via the squared relative residual $\|r_k\|^2 / \|r_0\|^2$, which corresponds to a relative mean squared error (MSE). For the classical numerical methods, we stop the iteration once a tolerance of 10^{-12} for the relative residual or an iteration count of 250 k iterations is reached. The CNNs are always trained for 250 k iterations or epochs.

We compare the relative residuals for the various methods applied to the standard and normal equations in Fig. 3. As expected, the CG method applied to the standard equation (CG-SE) converges the fastest after 221 iterations; note again that the convergence could be significantly improved using preconditioning techniques. The CG method applied to the normals equation (CG-NE) converges within 7 737 iterations, the GD method on the original equation (GD-SE) in 15 811 iterations. The GD method on the normal equations (GD-NE) does not converge within 250 k iterations and reaches a relative residual of $5.2 \cdot 10^{-7}$ at termination of the iteration.

As can be seen in Fig. 4d, the GD-NE solution, which has not converged within 250 k iterations, has a large relative L_2-error of 34 % compared with the analytical solution. For CG-SE, CG-NE, and GD-SE, we obtain errors of 0.008 %, 0.02 %, and 0.15 %, respectively, at convergence. In terms of convergence with respect to the relative squared residual norm, the ML approaches perform worse. Both ML-GD



(a) CG-SE (b) GD-SE (c) CG-NE (d) GD-NE (e) ML-GD (f) ML-Adam

Fig. 4 The solutions (top row) achieved with the various methods and the corresponding erorrs $(u^* - u)$ (bottom row) w.r.t the analytical solution at the grid nodes.

and ML-Adam do not achieve a relative tolerance of 10^{-12} and the training is stopped after 250 k iterations/epochs with a final relative residual of $1.5 \cdot 10^{-7}$ for ML-GD and $3.1 \cdot 10^{-8}$ for ML-Adam. Nonetheless, we achieve relative L_2-errors of 0.7 % for ML-GD and 0.02 % for ML-Adam. These are significantly lower than for GD-NE, even though the methods terminate at a similar relative residual. In fact, the accuracy is within one order of magnitude of the CG solutions and even better than the GD-SE solution; cf. also Fig, 4.

Spectral bias in the CNN training

Let us discuss why, in comparison, the error may be much lower for the CNN compared to the classical numerical solvers for a residual in the same order of magnitude. In particular, for the error e and the residual r, we have

$$Ae = A(u^* - u) = b - Au = r$$

Hence, of course, the relation of $\|e\|$ and $\|r\|$ depends on how the error decomposes into eigenfunctions of high/low eigenvalues. Since the CNNs were able to achieve comparatively low error while exhibiting higher absolute and relative residual, especially compared to the CG solutions, this suggests that the corresponding error is mainly composed of eigenfunctions corresponding to high eigenvalues. In particular, this implies that the CNNs exhibit some form of spectral bias, i.e., that they tend to learn eigenfunctions corresponding to low eigenvalues. Note that the spectral bias has been previously studied for DNNs [2, 27] and for PINNs [34]. However, to the best of the authors' knowledge, it has not been studied for the physics-informed CNN approach considered here. A more detailed study is out of the scope of this paper but will be discussed in future research.

6 Conclusion

In this work, we have compared physics-informed CNNs with classical methods for solvinge PDEs on the example of the stationary diffusion problem. We have shown that solution methods that take advantage of properties of the problem, such as the CG method, outperform the ML approach both in the accuracy achieved and in the speed of convergence. Yet, the ML solutions learned were within an order of magnitude of the CG solutions, i.e., they were not infeasible. But the much slower convergence coupled with the need for hyperparameter optimization as well as the heuristic nature of the choice of method parameters argue for the use of classical methods. Nonetheless, with an ML approach it is possible to include parameters, such as boundary conditions, geometry, etc., as input. In such cases, ML approaches are superior to classical methods and thus there is a sound reason again to use them. The extension of this study to more complex problems, the incorporation of preconditioning, as well as a more detailed discussed of the spectral bias will be the subject of future research.

Acknowledgements This work was performed as part of the Helmholtz School for Data Science in Life, Earth and Energy (HDS-LEE) and received funding from the Helmholtz Association of German Research Centers. We gratefully acknowledge the use of the computational facilities of the Center for Data and Simulation Science (CDS) at the University of Cologne.

References

1. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems (2016).
2. Cao, Y., Fang, Z., Wu, Y., Zhou, D.-X., and Gu, Q. Towards Understanding the Spectral Bias of Deep Learning (2020).
3. Cuomo, S., di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., and Piccialli, F. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *arXiv* (2022).
4. De Ryck, T. and Mishra, S. Error analysis for physics-informed neural networks (PINNs) approximating Kolmogorov PDEs. *Advances in Computational Mathematics* **48**(6), 79 (2022).
5. Eichinger, M., Heinlein, A., and Klawonn, A. Stationary flow predictions using convolutional neural networks. In: *Numerical Mathematics and Advanced Applications ENUMATH 2019*, 541–549. Springer (2021).
6. Eichinger, M., Heinlein, A., and Klawonn, A. Surrogate convolutional neural network models for steady computational fluid dynamics simulations. *Electronic Transactions on Numerical Analysis* **56**, 235–255 (2022).
7. Franco, N. R., Fresca, S., Manzoni, A., and Zunino, P. Approximation bounds for convolutional neural networks in operator learning (2023). ArXiv:2207.01546 [cs, math].
8. Fresca, S., Dede', L., and Manzoni, A. A Comprehensive Deep Learning-Based Approach to Reduced Order Modeling of Nonlinear Time-Dependent Parametrized PDEs. *Journal of Scientific Computing* **87**(2), 61 (2021).

9. F.R.S, K. P. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **2**(11) (1901). Publisher: Taylor & Francis pages = 559–572,.
10. Gao, H., Sun, L., and Wang, J. Phygeonet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state pdes on irregular domain. *Journal of Computational Physics* **428**, 110079 (2021).
11. Gonzalez, F. J. and Balajewicz, M. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems (2018). ArXiv:1808.01346 [physics].
12. Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press (2016).
13. Grimm, V., Heinlein, A., and Klawonn, A. Physics-aware convolutional neural networks for two-dimensional flow predictions. In preparation.
14. Guo, X., Li, W., and Iorio, F. Convolutional Neural Networks for Steady Flow Approximation. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 481–490. Association for Computing Machinery, New York, NY, USA (2016).
15. Guo, X., Li, W., and Iorio, F. Convolutional neural networks for steady flow approximation. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, 481–490. Association for Computing Machinery, New York, NY, USA (2016).
16. He, K., Zhang, X., Ren, S., and Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification (2015). ArXiv:1502.01852 [cs].
17. Jentzen, A., Salimova, D., and Welti, T. A proof that deep artificial neural networks overcome the curse of dimensionality in the numerical approximation of Kolmogorov partial differential equations with constant diffusion and nonlinear drift coefficients. *Communications in Mathematical Sciences* **19**(5), 1167–1205 (2021). Publisher: International Press of Boston.
18. Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
19. Kovachki, N., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A., and Anandkumar, A. Neural Operator: Learning Maps Between Function Spaces (2022). ArXiv:2108.08481 [cs, math].
20. Lagaris, I., Likas, A., and Fotiadis, D. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks* **9**(5), 987–1000 (1998). Conference Name: IEEE Transactions on Neural Networks.
21. Lassila, T., Manzoni, A., Quarteroni, A., and Rozza, G. Model order reduction in fluid dynamics: challenges and perspectives. *Reduced Order Methods for modeling and computational reduction* 235–273 (2014).
22. Lee, K. and Carlberg, K. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders (2019). ArXiv:1812.08373 [cs].
23. Liu, D. and Nocedal, J. On the limited memory bfgs method for large scale optimization. *Mathematical Programming* **45**, 503–528 (1989).
24. Lu, L., Jin, P., and Karniadakis, G. E. DeepONet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *Nature Machine Intelligence* **3**(3), 218–229 (2021). ArXiv:1910.03193 [cs, stat].
25. Maulik, R., Lusch, B., and Balaprakash, P. Reduced-order modeling of advection-dominated systems with recurrent neural networks and convolutional autoencoders. *Physics of Fluids* **33**(3), 037106 (2021). Publisher: American Institute of Physics.
26. Quarteroni, A., Manzoni, A., and Negri, F. *Reduced basis methods for partial differential equations*, *Unitext*, vol. 92. Springer, Cham (2016).
27. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F. A., Bengio, Y., and Courville, A. On the Spectral Bias of Neural Networks (2019). ArXiv:1806.08734 [cs, stat].
28. Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics* **378**, 686–707 (2019).
29. Rathinam, M. and Petzold, L. R. A New Look at Proper Orthogonal Decomposition. *SIAM Journal on Numerical Analysis* **41**(5), 1893–1925 (2003).

30. Ronneberger, O., Fischer, P., and Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab, N., Hornegger, J., Wells, W. M., and Frangi, A. F. (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, 234–241. Springer International Publishing, Cham (2015).
31. Sharma, R., Farimani, A. B., Gomes, J., Eastman, P., and Pande, V. Weakly-supervised learning of heat transport via physics informed loss. *arXiv* (2018). URL arXiv:1807.11374.
32. Toselli, A. and Widlund, O. *Domain decomposition methods-algorithms and theory*, vol. 34. Springer Science & Business Media (2004).
33. Trottenberg, U., Oosterlee, C. W., and Schuller, A. *Multigrid*. Elsevier (2000).
34. Wang, S., Yu, X., and Perdikaris, P. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics* **449**, 110768 (2022).
35. Willard, J., Jia, X., Xu, S., Steinbach, M., and Kumar, V. Integrating scientific knowledge with machine learning for engineering and environmental systems. *arXiv* (2021).

Optimized Robin Transmission Conditions for Anisotropic Diffusion on Arbitrary Meshes

Martin J. Gander, Laurence Halpern, Florence Hubert, and Stella Krell

1 Introduction

We are interested in solving in parallel anisotropic diffusion problems of the form

$$\mathcal{L}u := -\mathrm{div}(A\nabla u) + \eta u = g \quad \text{in } \Omega \subset \mathbb{R}^2, \qquad u = 0 \quad \text{on } \partial\Omega, \tag{1}$$

where A is a symmetric positive definite matrix with $W^{1,\infty}$ coefficients,

$$(x,y) \in \Omega \mapsto A(x,y) = \begin{pmatrix} A_{xx} & A_{xy} \\ A_{xy} & A_{yy} \end{pmatrix},$$

and $(x,y) \in \Omega \mapsto \eta(x,y) \geq 0$ is in $L^\infty(\Omega)$. Schwarz algorithms for such problems are naturally formulated and studied at the continuous level. For a decomposition of the domain Ω into possibly non-overlapping subdomains Ω_j, $j = 1, 2, \ldots, J$, the parallel optimized Schwarz algorithm with Robin transmission conditions for the anisotropic diffusion problem (1) computes for iteration index $\ell = 1, 2, \ldots$

$$\begin{aligned} \mathcal{L}u_j^\ell &= g && \text{in } \Omega_j, \\ u_j^\ell &= 0 && \text{on } \partial\Omega_j \cap \partial\Omega, \\ A\nabla u_j^\ell \cdot \mathbf{n}_j + p u_j^\ell &= -A\nabla u_i^{\ell-1} \cdot \mathbf{n}_i + p u_i^{\ell-1} && \text{on } \Gamma_{ji}, \end{aligned} \tag{2}$$

Martin J. Gander
Section de Mathématiques, Université de Genève, e-mail: martin.gander@unige.ch

Laurence Halpern
LAGA, Université Sorbonne Paris-Nord, e-mail: halpern@math.univ-paris13.fr

Florence Hubert
Aix-Marseille Université, CNRS, e-mail: florence.hubert@univ-amu.fr

Stella Krell
Université Côte d'Azur, Inria, CNRS, LJAD, e-mail: stella.krell@univ-cotedazur.fr

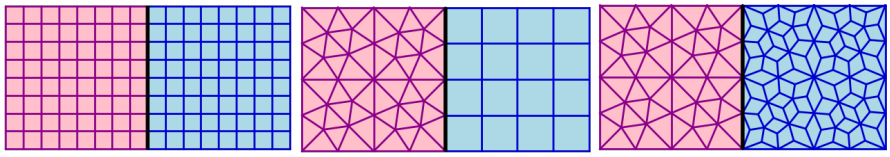


Fig. 1 Three typical discretizations for two subdomain decompositions: square-square (ss), triangle-square (ts) and triangle-quadrangle (tq).

where $\mathbf{n}_j$ denotes the unit outer normal on the boundary of Ω_j, and Γ_{ji} denotes the portion of the interface where Ω_j takes data from Ω_i. The efficiency of the algorithm is known to depend on the choice of the parameter p, which is usually optimized for a simple two subdomain decomposition, see [3] for the Laplace case. In [5], we showed at the continuous level for a general constant diffusion matrix A that for $\Omega := (-a, a) \times (0, b)$ decomposed into two non-overlapping subdomains $\Omega_1 := (-a, 0) \times (0, b)$ and $\Omega_2 := (0, a) \times (0, b)$ with the interface $\Gamma_{12} = \Gamma_{21} := \partial\Omega_1 \cap \partial\Omega_2$, the optimized parameters and associated convergence factors are of the form

$$p^* = \sqrt{\tilde{f}(k_{\min})\tilde{f}(k_{\max})}, \quad \rho^* = \frac{\sqrt{\tilde{f}(k_{\max})} - \sqrt{\tilde{f}(k_{\min})}}{\sqrt{\tilde{f}(k_{\max})} + \sqrt{\tilde{f}(k_{\min})}}, \tag{3}$$

where for a general constant diffusion matrix A

$$\tilde{f}(k) := f(r(k)) \quad \text{with} \quad r(k) := \frac{1}{A_{xx}}\sqrt{\eta A_{xx} + \left(\frac{\pi k}{b}\right)^2 \det A}, \tag{4}$$

with the function f defined for unbounded and bounded domains by

$$f(r) := \begin{cases} f_\infty(r) := A_{xx} r & a = \infty, \\ f_a(r) := f_\infty(r) \coth(ar) & a < \infty. \end{cases} \tag{5}$$

For both cases, the smallest frequency is $k_{\min} = 1$ and the largest frequency can be estimated by $k_{\max} = \frac{b}{h_y}$ for cell centered (cc) discretization, and $k_{\max} = \frac{b}{h_y} - 1$ for vertex centered (vc) discretizations, which are almost the same for small mesh size h_y in the y direction, see below for more information.

We show for the three example meshes in Figure 1 the numerically computed convergence factors $\check{\rho}$ in Table 1 when running the optimized Schwarz algorithm discretized by Discrete Duality Finite Volumes (DDFV, see [5] for the DDFV Schwarz algorithm, and [7, 2, 1] for DDFV discretizations in general) for the Laplace problem, $A(x, y) = I$, and four anisotropic diffusion matrices, and characteristic mesh size $h_x = h_y =: h = \frac{1}{16}$, i.e the meshes in Figure 1 twice refined. We used the theoretically optimized value $p^* = p^*_{\infty,\text{cvc}}$ from (3) with $k_{\max} = \frac{b}{h_y} - 1$ corresponding to the vc scheme (index cvc for continuous vertex centered), see the comment at the end of section 3, and then also determined the numerically best working parameter $\check{p}^*$ and associated convergence factor $\check{\rho}^*$, which we computed (throughout the paper) per-

Table 1 Numerically measured convergence factors $\check{\rho}$ of the optimized Schwarz algorithm for the three example meshes square-square (ss), triangle-square (ts) and triangle-quadrangle (tq) for the Laplace problem and four anisotropic diffusion problems with the theoretical parameter $p^*_{\infty,\mathrm{cvc}}$ and the numerically best working one $\check{p}^*$.

Problem			ss	ts	tq	ss		ts		tq	
A_{xx}	A_{yy}	$p^*_{\infty,\mathrm{cvc}}$	$\check{\rho}$	$\check{\rho}$	$\check{\rho}$	$\check{p}^*$	$\check{\rho}^*$	$\check{p}^*$	$\check{\rho}^*$	$\check{p}^*$	$\check{\rho}^*$
1	1	12.87	0.592	0.592	0.593	11.89	0.567	10.87	0.566	11.63	0.559
16	1	51.50	0.452	0.521	0.602	49.84	0.439	46.29	0.475	44.79	0.556
16	$\frac{1}{16}$	16.01	0.351	0.343	0.586	23.50	0.174	19.88	0.254	11.07	0.487
1	16	50.35	0.821	0.744	0.687	75.14	0.732	57.22	0.712	57.61	0.647
$\frac{1}{16}$	16	12.59	0.949	0.919	0.891	26.84	0.884	22.46	0.841	21.52	0.842

forming each time 100 iterations and using the last 40 to fit the linear convergence, to avoid initial fluctuations due to starting with a random initial guess.

We see from this experiment that for the Laplace problem the theoretically determined best parameter at the continuous level $p^*_{\infty,\mathrm{cvc}}$ performs very well on all meshes, and is close to the numerically best working one $\check{p}^*$, with $\check{\rho} \approx \check{\rho}^*$. For anisotropic diffusion however this is not the case: the performance now depends on the mesh structure, and the numerically optimized parameter $\check{p}^*$ can be rather different from the theoretical parameter $p^*_{\infty,\mathrm{cvc}}$. It is this difference we want to better understand, in particular for DDFV discretizations, which are highly accurate for anisotropic diffusion.

To start with our investigation, we plot in Figure 2 an example subdomain solution on the right subdomain Ω_2 with interface value equal to 1 and vanishing source term for the Laplace case and two anisotropic diffusion cases. We see that the anisotropy deforms the solution quite a bit, and for A_{xx} large, the subdomain clearly sees the boundary conditions at the outer boundary $\partial\Omega$ (Figure 2 middle), whereas for A_{yy} large a boundary layer is forming close to the interface Γ_{21} (Figure 2 right). This indicates that both the subdomain size, as well as the discretization, i.e. the mesh size, should influence the behavior of the optimized Schwarz method for anisotropic diffusion, and thus the best value of the parameter p.

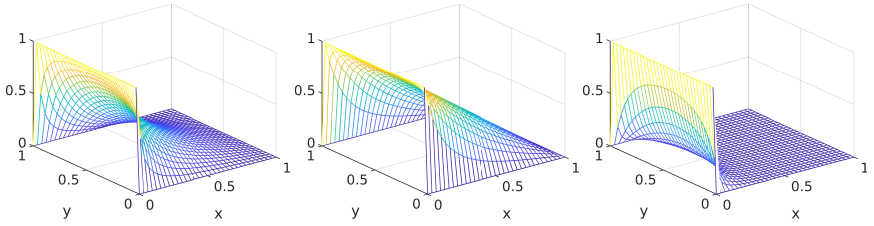


Fig. 2 Solutions for $A_{xx} = 1$, $A_{yy} = 1$ (left), for $A_{xx} = 16$, $A_{yy} = 1$ (middle) for $A_{xx} = 1$, $A_{yy} = 16$ (right), on an isotropic mesh.

2 Optimized parameters at the discrete level

For rectangular meshes and for a diagonal anisotropy ($A_{xy} = 0$), it is easy to see (see e.g. [4]) that the DDFV scheme leads to two decoupled classical finite difference schemes, a cell centered (cc) scheme with unknowns at the cell centers, and a vertex centered (vc) scheme with unknowns at the vertices. In [4], we performed the optimization analysis in the same rectangular domain configuration as above, for a discretization associated to the step sizes h_x and h_y for both the cc and vc schemes for unbounded ($a = \infty$) and bounded ($a < \infty$) domains. The optimized parameters and associated convergence factors are again of form (3), with

$$\begin{gathered}\tilde{f}(k) := f(\nu(k)),\ \nu(k) := -\ln(\lambda(k)),\ \lambda(k) := 1 + \frac{\mu(k)}{2} - \sqrt{\mu(k) + \frac{\mu(k)^2}{4}}, \\ \mu(k) := \frac{h_x^2}{A_{xx}}\left(4\frac{A_{yy}}{h_y^2}\sin^2\left(\frac{k\pi h_y}{2b}\right) + \eta\right),\end{gathered} \tag{6}$$

and the function f is defined for the cc and vc schemes on unbounded and bounded domains by

$$f(\nu) := \begin{cases} f_{\infty,\mathrm{cc}}(\nu) := 2\frac{A_{xx}}{h_x}\tanh\left(\frac{\nu}{2}\right), & a = \infty, \\ f_{a,\mathrm{cc}}(\nu) := f_{\infty,cc}(\nu)\coth\left(\frac{a\nu}{h_x}\right), & a < \infty, \\ f_{\infty,\mathrm{vc}}(\nu) := \frac{A_{xx}}{h_x}\sinh\left(\nu\right), & a = \infty, \\ f_{a,\mathrm{vc}}(\nu) = f_{\infty,vc}(\nu)\coth\left(\frac{a\nu}{h_x}\right), & a < \infty. \end{cases} \tag{7}$$

Again the smallest frequency $k_{\min} = 1$, and the maximum frequencies can be estimated by $k_{\max} = \frac{b}{h_y}$ for the cc scheme and $k_{\max} = \frac{b}{h_y} - 1$ for the vc scheme.

3 Asymptotic analysis

In order to understand the difference in the performance of the optimized Schwarz method in the anisotropic case, we now present a new asymptotic analysis of the optimized parameters and associated convergence factors. We look at the asymptotic behavior as h_x and h_y tend to zero, their ratio being constant.

We start with the asymptotic analysis of the optimization results (4)–(5) at the continuous level. When inserting the smallest frequency $k = k_{\min}$ into (4)–(5), we get in the unbounded domain case

$$\tilde{f}_\infty(k_{\min}) = \sqrt{\eta A_{xx} + \left(\frac{\pi}{b}\right)^2 \det A},$$

and in the bounded domain case

$$\tilde{f}_a(k_{\min}) = \sqrt{\eta A_{xx} + \left(\frac{\pi}{b}\right)^2 \det A}\ \coth\left(\frac{a}{A_{xx}}\sqrt{\eta A_{xx} + \left(\frac{\pi}{b}\right)^2 \det A}\right).$$

At the largest frequency $k = k_{\max}$, we obtain the same asymptotics, namely

$$\tilde{f}_\infty(k_{\max}) = \tilde{f}_a(k_{\max}) = \frac{\pi\sqrt{\det A}}{h_y} + O(1). \tag{8}$$

Now when h_y tends to zero, we see from (4) that $ar(k_{\max})$ tends to infinity, and therefore $\coth(ar(k_{\max})) = 1 + o(h_y)$. We thus obtain for the unbounded domain case $a = \infty$ for the optimized parameter and associated convergence factor

$$p^*_\infty \sim \left(\eta A_{xx} + \left(\frac{\pi}{b}\right)^2 \det A\right)^{\frac{1}{4}} \left(\pi\sqrt{\det A}\right)^{\frac{1}{2}} h_y^{-\frac{1}{2}},$$
$$\rho^*_\infty \sim 1 - 2\left(\eta A_{xx} + \left(\frac{\pi}{b}\right)^2 \det A\right)^{\frac{1}{4}} \left(\pi\sqrt{\det A}\right)^{-\frac{1}{2}} h_y^{\frac{1}{2}},$$

where $f(h_y) \sim g(h_y)$ means $\lim_{h_y\to 0} \frac{f(h_y)}{g(h_y)} = 1$, and when $a < \infty$, we get

$$p^*_a \sim \left(\eta A_{xx} + \left(\frac{\pi}{b}\right)^2 \det A\right)^{\frac{1}{4}} \left(\pi\sqrt{\det A}\right)^{\frac{1}{2}} \left(\coth\left(\frac{a}{A_{xx}}\sqrt{\eta A_{xx} + \left(\frac{\pi}{b}\right)^2 \det A}\right)\right)^{\frac{1}{2}} h_y^{-\frac{1}{2}},$$
$$\rho^*_a \sim 1 - 2\left(\eta A_{xx} + \left(\frac{\pi}{b}\right)^2 \det A\right)^{\frac{1}{4}} \left(\pi\sqrt{\det A}\right)^{-\frac{1}{2}} \left(\coth\left(\frac{a}{A_{xx}}\sqrt{\eta A_{xx} + \left(\frac{\pi}{b}\right)^2 \det A}\right)\right)^{\frac{1}{2}} h_y^{\frac{1}{2}}.$$

We see that the asymptotic behavior in the mesh size is the same, but the constants differ between the bounded and unbounded domain case, clearly indicating that the continuous analysis on the bounded domain can take into account the anisotropy observed in Figure 2.

We next perform an asymptotic analysis of the optimization results (6) and (7) at the discrete level. For a diagonal diffusion matrix A, at the minimum frequency, $k = k_{\min}$, we obtain from (6)

$$\mu(k_{\min}) = \frac{h_x^2}{A_{xx}}\left(4\frac{A_{yy}}{h_y^2}\sin^2\left(\frac{\pi h_y}{2b}\right) + \eta\right) = \frac{h_x^2}{A_{xx}^2}\left(\eta A_{xx} + \left(\frac{\pi}{b}\right)^2 A_{xx}A_{yy} + O(h_y^2)\right).$$

Hence $\mu(k_{\min}) \to 0$ when the mesh is refined, and because $\lambda(k_{\min}) \sim 1 - \sqrt{\mu(k_{\min})}$ and $\tilde{f}(k_{\min}) \sim \frac{A_{xx}}{h_x}\sqrt{\mu(k_{\min})}$, we obtain

$$\tilde{f}_{\infty,\mathrm{cc}}(k_{\min}) \sim \tilde{f}_{\infty,\mathrm{vc}}(k_{\min}) \sim \sqrt{\eta A_{xx} + \left(\frac{\pi}{b}\right)^2 A_{xx}A_{yy}}. \tag{9}$$

At the highest frequency, $k = k_{\max}$, we obtain for the cc scheme

$$\mu_{\rm cc}(k_{\max}) = \frac{h_x^2}{A_{xx}}\left(4\frac{A_{yy}}{h_y^2}\sin^2\left(\frac{\pi}{2}\right)+\eta\right) = \frac{h_x^2}{A_{xx}^2}\left(\eta A_{xx}+4\frac{A_{xx}A_{yy}}{h_y^2}\right) \sim 4\beta,$$

where $\beta := \frac{A_{yy}}{h_y^2}\frac{h_x^2}{A_{xx}}$, and similarly for the vc scheme,

$$\mu_{\rm vc}(k_{\max}) = \frac{h_x^2}{A_{xx}}\left(4\frac{A_{yy}}{h_y^2}\sin^2\left(\frac{\pi}{2}(1-h_y)\right)+\eta\right) = \frac{h_x^2}{A_{xx}^2}\left(4\frac{A_{xx}A_{yy}}{h_y^2}+O(1)\right) \sim 4\beta.$$

Note that the case of a Laplacian with an isotropic square mesh corresponds to the parameter value $\beta = 1$. By hyperbolic trigonometric calculus, and $\frac{A_{xx}}{h_x} = \frac{\sqrt{A_{xx}A_{yy}}}{h_y\sqrt{\beta}}$, we obtain the alternative formula $f_{\infty,\rm cc}(\nu(k)) = 2\frac{A_{xx}}{h_x}\frac{1-\lambda(k)}{1+\lambda(k)}$, which yields

$$\begin{aligned}\tilde{f}_{\infty,\rm cc}(k_{\max}) &= 2\frac{A_{xx}}{h_x}\frac{-\beta+\sqrt{\beta+\beta^2}}{1+\beta-\sqrt{\beta+\beta^2}} = \frac{\sqrt{A_{xx}A_{yy}}}{h_y\sqrt{\beta}}2\frac{-\beta+\sqrt{\beta+\beta^2}}{1+\beta-\sqrt{\beta+\beta^2}}\\ &= \frac{\sqrt{A_{xx}A_{yy}}}{h_y\sqrt{\beta}}2\frac{\sqrt{\beta+\beta^2}}{1+\beta} := \frac{\sqrt{A_{xx}A_{yy}}}{h_y}\psi_{\rm cc}(\beta),\end{aligned}$$

with $\psi_{\rm cc}(\beta) = \frac{2}{\sqrt{1+\beta}}$. Similarly, since $f_{\infty,\rm vc}(\nu(k)) = \frac{A_{xx}}{h_x}\frac{1-\lambda(k)^2}{2\lambda(k)}$ by hyperbolic trigonometric calculus, we obtain

$$\begin{aligned}f_{\infty,\rm vc}(k_{\max}) &= \frac{A_{xx}}{h_x}\frac{2\left(-\beta+\sqrt{\beta+\beta^2}\right)\left(1+\beta-\sqrt{\beta+\beta^2}\right)}{1+2\beta-2\sqrt{\beta+\beta^2}}\\ &= \frac{\sqrt{A_{xx}A_{yy}}}{h_y\sqrt{\beta}}2\left(-\beta+\sqrt{\beta+\beta^2}\right)\left(1+\beta-\sqrt{\beta+\beta^2}\right)\left(1+2\beta+2\sqrt{\beta+\beta^2}\right)\\ &= \frac{\sqrt{A_{xx}A_{yy}}}{h_y\sqrt{\beta}}2\sqrt{\beta+\beta^2} := \frac{\sqrt{A_{xx}A_{yy}}}{h_y}\psi_{\rm vc}(\beta),\end{aligned}$$

with $\psi_{\rm vc}(\beta) = 2\sqrt{1+\beta}$. Note that in the special case $\beta = 1$, we get $\psi_{\rm cc}(\beta) = \sqrt{2}$ and $\psi_{\rm vc}(\beta) = 2\sqrt{2}$, a factor 2 difference. For the unbounded domain case, $a = \infty$, we then obtain for the optimized parameters and associated convergence factors of the cc and vc schemes

$$\begin{aligned}p^*_{\infty,\rm cc} &\sim \psi_{\rm cc}(\beta)^{\frac{1}{2}}\sqrt{A_{xx}A_{yy}}\left(\frac{\eta}{A_{yy}}+\left(\frac{\pi}{b}\right)^2\right)^{\frac{1}{4}}h_y^{-\frac{1}{2}},\\ p^*_{\infty,\rm vc} &\sim \psi_{\rm vc}(\beta)^{\frac{1}{2}}\sqrt{A_{xx}A_{yy}}\left(\frac{\eta}{A_{yy}}+\left(\frac{\pi}{b}\right)^2\right)^{\frac{1}{4}}h_y^{-\frac{1}{2}},\\ \rho^*_{\infty,\rm cc} &\sim 1-2\psi_{\rm cc}(\beta)^{-\frac{1}{2}}\left(\frac{\eta}{A_{yy}}+\left(\frac{\pi}{b}\right)^2\right)^{\frac{1}{4}}h_y^{\frac{1}{2}},\\ \rho^*_{\infty,\rm vc} &\sim 1-2\psi_{\rm vc}(\beta)^{-\frac{1}{2}}\left(\frac{\eta}{A_{yy}}+\left(\frac{\pi}{b}\right)^2\right)^{\frac{1}{4}}h_y^{\frac{1}{2}}.\end{aligned}$$

In the bounded domain case, $a < \infty$, we see that $\coth\left(\frac{a\nu(k_{\max})}{h_x}\right) \sim 1$ and when $\mu(k_{\min}) \to 0$, we have $\nu(k_{\min}) \sim -\sqrt{\mu(k_{\min})}$, which implies

$$\frac{a\nu(k_{\min})}{h_x} \sim \frac{a}{\sqrt{A_{xx}}}\sqrt{\eta + \left(\frac{\pi}{b}\right)^2 A_{yy}} \Rightarrow \coth\left(\frac{a\nu(k_{\min})}{h_x}\right) \sim \coth\left(\frac{a}{\sqrt{A_{xx}}}\sqrt{\eta + \left(\frac{\pi}{b}\right)^2 A_{yy}}\right). \tag{10}$$

We therefore get for the optimized parameters and associated convergence factors for the cc and vc schemes in the bounded domain case

$$p^*_{\mathrm{a,cc}} \sim \psi_{\mathrm{cc}}(\beta)^{\frac{1}{2}}\sqrt{A_{xx}A_{yy}}\left(\frac{\eta}{A_{yy}} + \left(\frac{\pi}{b}\right)^2\right)^{\frac{1}{4}} \coth\left(\frac{a}{\sqrt{A_{xx}}}\sqrt{\eta + \left(\frac{\pi}{b}\right)^2 A_{yy}}\right)^{\frac{1}{2}} h_y^{-\frac{1}{2}},$$

$$p^*_{\mathrm{a,vc}} \sim \psi_{\mathrm{vc}}(\beta)^{\frac{1}{2}}\sqrt{A_{xx}A_{yy}}\left(\frac{\eta}{A_{yy}} + \left(\frac{\pi}{b}\right)^2\right)^{\frac{1}{4}} \coth\left(\frac{a}{\sqrt{A_{xx}}}\sqrt{\eta + \left(\frac{\pi}{b}\right)^2 A_{yy}}\right)^{\frac{1}{2}} h_y^{-\frac{1}{2}},$$

$$\rho^*_{\mathrm{a,cc}} \sim 1 - 2\psi_{\mathrm{cc}}(\beta)^{-\frac{1}{2}}\left(\frac{\eta}{A_{yy}} + \left(\frac{\pi}{b}\right)^2\right)^{\frac{1}{4}} \coth\left(\frac{a}{\sqrt{A_{xx}}}\sqrt{\eta + \left(\frac{\pi}{b}\right)^2 A_{yy}}\right)^{\frac{1}{2}} h_y^{\frac{1}{2}},$$

$$\rho^*_{\mathrm{a,vc}} \sim 1 - 2\psi_{\mathrm{vc}}(\beta)^{-\frac{1}{2}}\left(\frac{\eta}{A_{yy}} + \left(\frac{\pi}{b}\right)^2\right)^{\frac{1}{4}} \coth\left(\frac{a}{\sqrt{A_{xx}}}\sqrt{\eta + \left(\frac{\pi}{b}\right)^2 A_{yy}}\right)^{\frac{1}{2}} h_y^{\frac{1}{2}}.$$

These formulas take both the domain size and the mesh resolution into account, also when the mesh is not chosen appropriately for the anisotropy under consideration.

If one can not use separate parameters for the cc and vc components in a DDFV implementation, it was shown in [4] that the optimized choice for one parameter is of the form

$$p^*_{\mathrm{a,ddfv}} = \sqrt{f_{\mathrm{a,cc}}(\nu(k_{\min}))f_{\mathrm{a,vc}}(\nu(k_{\max}))},$$

and since asymptotically we have $f_{\mathrm{a,cc}}(\nu(k_{\min})) \sim f_{\mathrm{a,vc}}(\nu(k_{\min}))$ from (9) and (10), one should use the optimized parameter $p^*_{\mathrm{a,vc}} \sim p^*_{\mathrm{a,ddfv}}$ in that case.

The continuous and discrete asymptotic results lead to the following general theorem.

Theorem 1 (Optimized Robin parameter for diagonal anisotropic diffusion)

The optimized Schwarz method (2) *for the anisotropic diffusion problem* (1) *with diagonal diffusion matrix A and a subdomain decomposition of the rectangle* $\Omega = (-a, a) \times (0, b)$ *into two non-overlapping subdomains* $\Omega_1 := (-a, 0) \times (0, b)$ *and* $\Omega_2 := (0, a) \times (0, b)$ *has for small mesh size* h_y *the asymptotically optimized parameter and associated convergence factor*

$$p^* \sim \psi^{\frac{1}{2}}\sqrt{A_{xx}A_{yy}}\left(\frac{\eta}{A_{yy}} + \left(\frac{\pi}{b}\right)^2\right)^{\frac{1}{4}} c^{\frac{1}{2}} h_y^{-\frac{1}{2}}, \tag{11}$$

$$\rho^* \sim 1 - 2\psi^{-\frac{1}{2}}\left(\frac{\eta}{A_{yy}} + \left(\frac{\pi}{b}\right)^2\right)^{\frac{1}{4}} c^{\frac{1}{2}} h_y^{\frac{1}{2}}, \tag{12}$$

where in the unbounded domain case, $a = \infty$*, we have* $c = 1$*, whereas in the bounded domain case,* $a < \infty$*, we have*

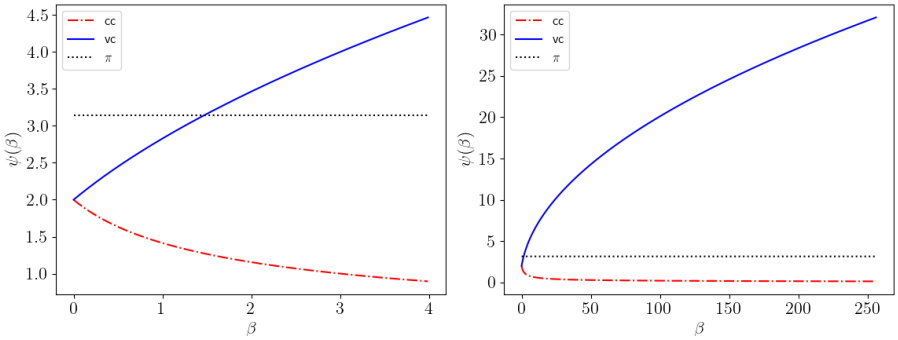


Fig. 3 Graph of the functions $\psi_{\rm cc}(\beta)$ and $\psi_{\rm vc}(\beta)$ for the discrete analysis, compared to $\psi(\beta) = \pi$ (dotted) from the continuous analysis for small and large β range.

$$c := c(a, b, A_{xx}, A_{yy}, \eta) = \coth\left(\frac{a}{\sqrt{A_{xx}}}\sqrt{\eta + \left(\frac{\pi}{b}\right)^2 A_{yy}}\right). \tag{13}$$

Furthermore, in the continuous case $\psi = \pi$, and in the discrete case we have

$$\psi := \psi_{\rm cc}(\beta) = \frac{2}{\sqrt{1+\beta}} \quad or \quad \psi = \psi_{\rm vc}(\beta) := 2\sqrt{1+\beta} \tag{14}$$

for the cell centered or vertex centered discretizations, with

$$\beta := \frac{A_{yy}}{h_y^2}\frac{h_x^2}{A_{xx}}. \tag{15}$$

Plotting the $\psi(\beta)$ functions in Figure 3, we see that if $\beta = 1$ then the continuous and discrete analyses give about the same optimized parameter p^* and associated convergence factor, especially for the vc scheme. Since $\beta = \frac{A_{yy}}{h_y^2}\frac{h_x^2}{A_{xx}}$, this can be achieved by having equal mesh sizes $h_x = h_y$ and isotropic diffusion $A_{xx} = A_{yy}$, or by adapting the mesh sizes to the anisotropy, $h_y^2 = \frac{A_{xx}}{A_{yy}}h_x^2$. Such an adaptation is also recommended for accuracy, since a Taylor expansion gives

$$\begin{aligned} &A_{xx}\frac{u(x+h_x,y)-2u(x,y)-u(x-h_x,y)}{h_x^2} + A_{yy}\frac{u(x,y+h_y)-2u(x,y)-u(x,y-h_y)}{h_y^2} \\ &\quad = (A_{xx}\partial_{xx} + A_{yy}\partial_{yy})u(x,y) + \tfrac{1}{12}(A_{xx}h_x^2\partial_x^4 + A_{yy}h_y^2\partial_y^4)u(x,y) + \ldots, \end{aligned} \tag{16}$$

and from the separation of variables solution $u(x,y) = e^{-\frac{k\pi}{b}\sqrt{\frac{A_{yy}}{A_{xx}}}x}\sin(\frac{k\pi}{b}y)$ we see that the fourth derivative in x scales like $\frac{A_{yy}^2}{A_{xx}^2}$, while the fourth derivative in y does not scale in these entries, and hence to balance the error term, we should choose

$$A_{xx}h_x^2\frac{A_{yy}^2}{A_{xx}^2} \approx A_{yy}h_y^2 \quad \Longrightarrow \frac{h_x^2}{A_{xx}}\frac{A_{yy}}{h_y^2} = \beta \approx 1. \tag{17}$$

Hence for $\beta \approx 1$, we can use the continuous analysis results and expect good performance, also in highly anisotropic cases, provided the mesh is adapted accordingly. If β is very different from one, we should use the parameters from the discrete analysis to get good performance. We also see from Figure 3 (right) that for large β the optimized parameters for the cc and vc schemes are becoming more and more different, and (12) together with (14) indicates that the cc scheme is converging much faster than the vc scheme in these not well resolved mesh situations. In the DDFV case with general meshes, where both cc and vc discretizations are involved, the importance will then lie on a good optimization of the vc parameter, the cc parameter playing only a secondary role in these not well resolved cases.

Next, we see from Theorem 1 that if $c \approx 1$, then we can use the unbounded domain analysis, since the only term depending on the domain bound a on the left and right is c. Now $c \approx 1$ if the argument of the coth is large, i.e. either the domains and thus $\frac{a}{b}$ is large, or η is large, or $\frac{A_{yy}}{A_{xx}}$ is large, which is illustrated in Figure 1 on the right, where we see that the outer boundary on the right does not play a major role any more[1]. If none of these hold, then the bounded domain analysis needs to be used to obtain good performance.

Finally, from ρ^* in Theorem 1, we see the algorithm will converge very fast with the well chosen p^*, provided A_{yy} is small or η large, or $\psi(\beta)$ is small. Having $\psi(\beta)$ small is however not advisable, because the discretization accuracy is only good for $\beta \approx 1$, see (17).

4 Numerical experiments

We can now explain the discrepancies we observed in Table 1 as soon as we solve anisotropic diffusion problems. There are two reasons: the first one is that when using the optimized parameter p_∞^* from the continuous, unbounded domain analysis, the fact that the subdomains are actually bounded in a concrete computation becomes important as soon as the diffusion in the orthogonal direction to the interface is large, and the cross diffusion tangential to the interface is small. This is visible also in Figure 1 showing a corresponding solution in the middle, where we can clearly see that the boundary on the right makes the solution decay linearly in the direction orthogonal to the interface, in stark contrast to the Laplace case on the left in Figure 1, where the decay is exponential. The second reason for discrepancies is the uniform discretization, which can not resolve well the boundary layers close to the top and bottom boundaries in Figure 1 (middle), and close to the interface in Figure 1 (right) which also influence the convergence of the Schwarz method.

[1] For example, in the case $A_{xx} = 1$ and $A_{yy} = 16$, the difference is of order 10^{-11}.

Table 2 Results corresponding to Table 1 but now using the theoretical parameter $p^*_{\text{a,cvc}}$ from the bounded domain analysis.

Problem			ss	ts	tq	ss		ts		tq	
A_{xx}	A_{yy}	$p^*_{\text{a,cvc}}$	$\check{\rho}$	$\check{\rho}$	$\check{\rho}$	$\check{p}^*$	$\check{\rho}^*$	$\check{p}^*$	$\check{\rho}^*$	$\check{p}^*$	$\check{\rho}^*$
1	1	12.48	0.582	0.581	0.583	11.89	0.567	10.87	0.566	11.63	0.559
16	1	60.59	0.514	0.578	0.651	49.84	0.439	46.29	0.475	44.79	0.556
16	$\frac{1}{16}$	28.04	0.258	0.436	0.741	23.50	0.174	19.88	0.254	11.07	0.487
1	16	48.75	0.826	0.751	0.695	75.14	0.732	57.22	0.712	57.61	0.647
$\frac{1}{16}$	16	12.19	0.950	0.921	0.894	26.84	0.884	22.46	0.841	21.52	0.842

Table 3 Results corresponding to Table 2 but now using the discrete theoretical parameters $p^*_{\text{a,cc}}$ and $p^*_{\text{a,vc}}$, and the numerically best working ones $\check{p}^*_{\text{cc}}$ and $\check{p}^*_{\text{vc}}$.

Problem				ss	ts	tq	ss			ts			tq		
A_{xx}	A_{yy}	$p^*_{\text{a,cc}}$	$p^*_{\text{a,vc}}$	$\check{\rho}$	$\check{\rho}$	$\check{\rho}$	$\check{p}^*_{\text{cc}}$	$\check{p}^*_{\text{vc}}$	$\check{\rho}^*$	$\check{p}^*_{\text{cc}}$	$\check{p}^*_{\text{vc}}$	$\check{\rho}^*$	$\check{p}^*_{\text{cc}}$	$\check{p}^*_{\text{vc}}$	$\check{\rho}^*$
1	1	8.62	12.22	0.573	0.572	0.574	8.62	11.93	0.566	7.73	11.38	0.533	10.49	10.49	0.527
16	1	49.16	50.56	0.444	0.509	0.592	49.59	49.87	0.439	45.87	45.89	0.468	39.61	40.13	0.514
16	$\frac{1}{16}$	23.48	23.48	0.174	0.347	0.698	23.50	23.44	0.173	19.75	20.24	0.242	11.42	11.65	0.466
1	16	19.07	84.09	0.723	0.728	0.733	20.01	80.71	0.714	44.46	66.21	0.653	13.78	58.50	0.621
$\frac{1}{16}$	16	1.84	54.59	0.806	0.834	0.861	1.13	51.09	0.796	1.90	36.72	0.756	0.69	30.80	0.733

As a first remedy, we use the optimized parameter p^*_a from the continuous, bounded domain analysis to take into account the boundedness of the domains. From Table 2 we see that this already improves the performance of the method when the diffusion is large in the orthogonal direction to the interface and small tangentially. However for the other cases using the bounded domain analysis is not sufficient due to the bad mesh resolution in the anisotropic case.

We therefore now use the discrete optimized formulas $p^*_{\text{a,vc}}$ and $p^*_{\text{a,cc}}$ in our DDFV Schwarz code, which are perfectly adapted to the anisotropy of the problem we are solving on bounded subdomains, and truly optimize both the vc and cc scheme component convergence also for the not well chosen mesh resolution. We show the corresponding results in Table 3. We see that now our parameters predicted by the discrete analysis for the cc and vc schemes give performance close to the truly best possible ones for rectangular meshes, and still work well on general meshes for which our analysis is not valid any more. Furthermore, the performance still follows our asymptotic analysis, as the plots of the convergence factors under mesh refinement in Figure 4 indicate.

We finally show numerical results using an anisotropic mesh which gives better approximate discrete solutions, see the truncation error analysis in (16). We show the corresponding results for such meshes in Table 4, and in Figure 5. We see that the continuous analysis gives now very good predictions for the optimized parameters for the vc scheme, while for the cc scheme their value is still a bit overestimated. This does however not influence the performance very much.

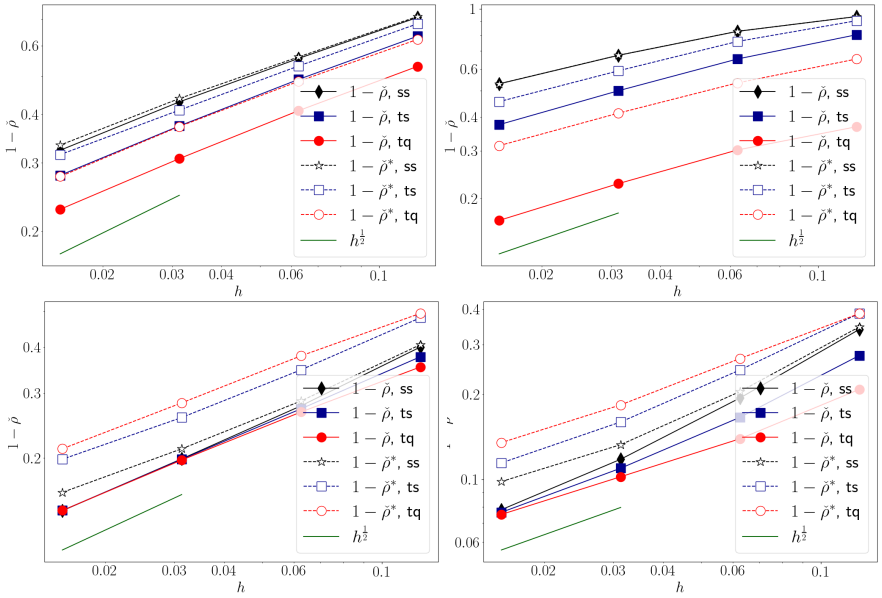


Fig. 4 Asymptotic dependence of $1-\check{\rho}$ on the mesh size for isotropic meshes and the anisotropic diffusion problems in Table 3, with $h = h_y = h_x$. From top left to bottom right: $(A_{xx}, A_{yy}) = (16, 1), (16, \frac{1}{16}), (1, 16), (\frac{1}{16}, 16)$.

5 Conclusions

Using asymptotic analysis, we explained rigorously numerical observations on the performance of DDFV optimized Schwarz methods applied to anisotropic diffusion. We showed that for strong anisotropic diffusion solved on uniform, non-adapted meshes, one needs optimized parameters from a more subtle discrete analysis, continuous optimization does not suffice. When using suitably adapted, anisotropic meshes such that the discrete solution is a good approximation of the continuous one, optimized parameters from a continuous analysis perform however well. We also showed numerically that this remains true if one uses meshes for which a detailed asymptotic analysis as ours on Cartesian meshes can not be performed. For extensions of the DDFV Schwarz algorithm to Navier-Stokes problems, see [6].

Table 4 Results obtained using the discrete optimized parameters for adapted anisotropic meshes.

Problem		ss aniso								
A_{xx}	A_{yy}	$p^*_{\mathrm{a,ccc}}$	$p^*_{\mathrm{a,cvc}}$	$\check{\rho}_{\mathrm{c}}$	$p^*_{\mathrm{a,cc}}$	$p^*_{\mathrm{a,vc}}$	$\check{\rho}$	$\check{p}^*_{\mathrm{cc}}$	$\check{p}^*_{\mathrm{vc}}$	$\check{\rho}^*$
16	1	125.13	124.15	0.730	83.94	118.73	0.718	82.30	111.96	0.705
16	$\frac{1}{16}$	115.32	115.09	0.749	77.37	109.43	0.737	77.37	102.45	0.724
1	16	50.35	48.75	0.601	33.67	47.67	0.581	33.37	46.43	0.573
$\frac{1}{16}$	16	12.59	12.19	0.601	8.42	11.92	0.580	8.42	11.63	0.574

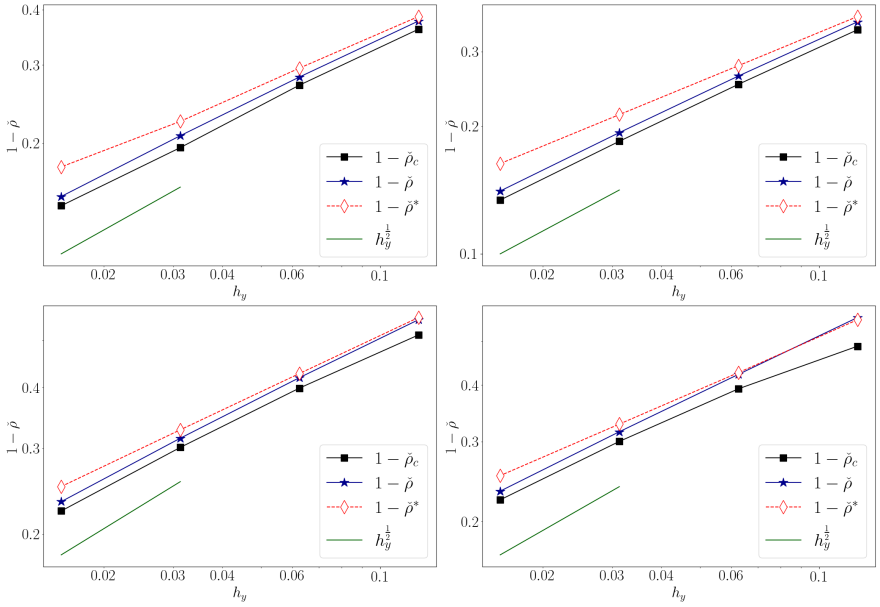


Fig. 5 Asymptotic dependence of $1-\check{\rho}$ on the mesh size for anisotropic meshes and the anisotropic diffusion problems in Table 4. From top left to bottom right: $(A_{xx}, A_{yy}) = (16,1), (16,\frac{1}{16}), (1,16), (\frac{1}{16},16)$.

References

1. Andreianov, B., Boyer, F., and Hubert, F. Discrete duality finite volume schemes for Leray-Lions type problems on general 2D meshes. *Numerical Methods for PDE* **23**(1), 145–195 (2007).
2. Domelevo, K. and Omnes, P. A finite volume method for the Laplace equation on almost arbitrary two-dimensional grids. *M2AN Math. Model. Numer. Anal.* **39**(6), 1203–1249 (2005).
3. Gander, M. J. Optimized Schwarz methods. *SIAM Journal on Numerical Analysis* **44**(2), 699–731 (2006).
4. Gander, M. J., Halpern, L., Hubert, F., and Krell, S. Discrete optimization of Robin transmission conditions for anisotropic diffusion with discrete duality finite volume methods. *Vietnam Journal of Mathematics* **49**, 1349–1378 (2021).
5. Gander, M. J., Halpern, L., Hubert, F., and Krell, S. Optimized Schwarz Methods for Anisotropic Diffusion with Discrete Duality Finite Volume Discretizations. *Moroccan Journal of Pure and Applied Analysis* **7**(2), 182–213 (2021).
6. Goudon, T., Krell, S., and Lissoni, G. Non-overlapping Schwarz algorithms for the incompressible Navier–Stokes equations with DDFV discretizations. *ESAIM: Mathematical Modelling and Numerical Analysis* **55**(4), 1271–1321 (2021).
7. Hermeline, F. Approximation of diffusion operators with discontinuous tensor coefficients on distorted meshes. *Comput. Methods Appl. Mech. Engrg.* **192**(16-18), 1939–1959 (2003).

Domain Decomposition Algorithms for Neural Network Approximation of Partial Differential Equations

Hyea Hyun Kim and Hee Jun Yang

1 Introduction

With the success of deep learning technology in many application areas, there have been pioneering approaches to approximate solutions of partial differential equations by neural network functions [2, 10, 12, 13]. Such approaches have advantages over the classical approximation methods in that they can be used without generating meshes adaptive to problem domains or developing equation dependent numerical schemes. However, its accuracy, stability, and efficiency questions have not yet been fully answered. In addition, long training time makes the neural network solution very expensive.

To enhance the neural network solution accuracy, large or deep neural network functions are usually employed. When training parameters in such large or deep neural networks, the optimization error becomes problematic to pollute the resulting computed solution accuracy. To address this issue in the neural network approximation, we approximate the solution by using partitioned local neural network functions. For that we first form an iterative scheme based on domain decomposition methods and we then find local neural network functions that approximate the local problem solutions at each iteration. Contrary to the single large or deep neural network case, the local neural network parameter training can be done more efficiently with less optimization errors.

There have been previous studies that utilize domain decomposition algorithms [14] to enhance the neural network efficiency and accuracy. In [8, 9], alternating Schwarz algorithms were developed to second order elliptic problems and in the author's previous study [6], additive Schwarz algorithms were proposed to

Hyea Hyun Kim
Department of Applied Mathematics and Institute of Natural Sciences, Kyung Hee University, Korea, e-mail: hhkim@khu.ac.kr

Hee Jun Yang
Department of Mathematics, Kyung Hee University, Korea, e-mail: yhjj109@khu.ac.kr

the same model problems, where the neural network functions are formed based on overlapping subdomain partitions. In both approaches, the proposed methods showed promising results but concrete convergence study has not been fully considered. In [4, 5], partitioned neural network functions are formed based on a non-overlapping subdomain partition and the global cost function is formed to train the parameters in the partitioned neural network functions. In their approach, the communication cost between local neural networks becomes enormous, since the number of epochs in the parameter training easily becomes more than several tens of thousands in practice.

In the author's recent work [7], a concrete convergence analysis on one-level and two-level additive Schwarz algorithms was provided with an assumption on the approximation error in the local and coarse neural network solutions. The numerical results on the one-level method are consistent with the convergence analysis. However, those on the two-level methods show that the coarse problem does not help to accelerate the convergence and it even pollutes the solution accuracy. By the NTK (Neural Tangent Kernel) theory [3, 15], when training the parameters in the neural network approximation, the smooth part of solutions is well approximated and the residual loss for the differential equation is well trained than that for the boundary condition. The local neural network solution errors in our proposed method thus showed high contrast errors near the subdomain boundary that resulted a less smooth global error during the iteration. The proposed coarse problem in [7] was not suitable to correct such a non-smooth global error.

In this work, we propose a partitioned neural network by utilizing a partition of unity functions and we then apply the additive Schwarz algorithm to propose an iterative solution procedure on the partitioned neural network functions, where local neural network parameters are trained to approximate local problem solutions at each iteration. When training the local parameters, only the residual loss for the differential equation in each subdomain problem comes in the cost function, and the boundary condition is enforced directly by multiplying the partition of unity function as an ansatz to the local neural network function. With this idea, the optimization error can be reduced when training the local parameters at each iteration compared to the approaches in [7]. As reported in our previous work [7], the coarse problem in the two-level method did not work due to the high contrast optimization errors observed near the boundary of subdomain overlapping region. By utilizing the partition of unity functions in forming the partitioned neural network approximation, we can remove such error problems and the coarse problem in the two-level method is thus expected to work more effectively. Such a partitioned neural network function using ansatz was first proposed in [11] with the aim of obtaining a more accurate neural network approximation to highly oscillatory solutions.

This paper is organized as follows. In Section 2, we introduce neural network approximation methods for solving partial differential equations and in Section 3 we propose one-level and two-level additive Schwarz algorithms for the partitioned neural network functions, where we present the two methods in our previous work [7] and extend those methods to the partitioned neural network functions. In Section 4, numerical results are presented for model elliptic problems and conclusions are given.

2 Neural network approximation for partial differential equations

Among several neural network approaches to solutions of partial differential equations, we will consider the PINN (Physics Informed Neural Network) method by [12]. Our domain decomposition approach can be applied to other neural network approximation methods by [2, 10, 13] as well. In the PINN methods, the solution is approximated with a neural network function $U(x;\theta)$ and the parameters θ in the neural network function are trained to solve supervised learning tasks in order to satisfy any given laws of physics described by partial differential equations,

$$\mathcal{L}(u) = f, \quad \text{in } \Omega, \quad \mathcal{B}(u) = g, \quad \text{on } \partial\Omega, \tag{1}$$

where $\mathcal{L}$ denotes a differential operator defined for a function u and $\mathcal{B}$ describes a given boundary condition on u, and f, g are given functions.

We assume that the model problem in (1) is well-posed and the solution u exists. We then approximate the solution u in (1) by a neural network, $U(x;\theta)$, where the parameters θ are trained to minimize the cost function

$$\mathcal{J}(\theta) = \mathcal{J}_{X_\Omega}(\theta) + \mathcal{J}_{X_{\partial\Omega}}(\theta),$$

where

$$\mathcal{J}_{X_\Omega}(\theta) := \frac{1}{|X_\Omega|} \sum_{x \in X_\Omega} |\mathcal{L}(U(x;\theta)) - f(x)|^2,$$
$$\mathcal{J}_{X_{\partial\Omega}}(\theta) := \frac{1}{|X_{\partial\Omega}|} \sum_{x \in X_{\partial\Omega}} |\mathcal{B}(U(x;\theta)) - g(x)|^2.$$

In the above, X_D denotes the collection of points chosen from the region D and $|X_D|$ denotes the number of points in the set X_D. The cost function $\mathcal{J}_{X_\Omega}(\theta)$ and $\mathcal{J}_{X_{\partial\Omega}}(\theta)$ are designed so that the optimized neural network $U(x;\theta)$ satisfies the equations in (1) derived from physics laws. When training the parameters θ, the following gradient based method is used,

$$\theta^{(n+1)} = \theta^{(n)} - \epsilon \nabla_\theta \mathcal{J}(\theta^{(n)})$$

for a given initial $\theta^{(0)}$ and with a suitable learning rate ϵ. Each gradient update step is called an epoch and usually more than several hundreds of thousand epochs are needed in such neural network approximation methods. Overall computation cost in the PINN is thus very expensive compared to the classical approximation methods.

The error between the exact solution $u(x)$ and the computed neural network solution $U(x;\widetilde{\theta})$ can be analyzed as follows. Letting $U(x;\theta^*)$ be the optimal approximate solution, we obtain

$$u(x) - U(x;\widetilde{\theta}) = (u(x) - U(x;\theta^*)) + (U(x;\theta^*) - U(x;\widetilde{\theta})),$$

where the first term in the right hand side is called the approximation error and the second is the optimization error. The approximation error can be controlled by enlarging the network size, while the optimization error is difficult to deal with. The optimization error depends on how to choose the training data set, how to form the loss functions, and how to perform the gradient based method.

In [11], it was numerically verified that for highly oscillatory model solutions PINN requires larger neural network functions and larger training epochs to increase the approximation solution accuracy. Such approximation property in the PINN was also analyzed by the NTK (Neural Tangent Kernel) theory, see [15]. To enhance the training efficiency and accuracy, in [11], the approximate solution is formed by using partitioned neural network functions with a much lesser number of parameters in each local neural network function than those in the single large neural network function. When a highly oscillatory solution is localized to a small subdomain, it becomes less oscillatory and thus it can be well approximated with a smaller neural network function. The parameter training cost in a smaller neural network function also becomes much smaller than that in the single larger neural network function. By utilizing the partitioned neural network functions, we thus expect that for difficult model problems we can reduce both the approximation error and the optimization error more effectively than just using a single large neural network function. In addition, utilizing the parallel computing resources, we can even make our one-level and two-level methods much more efficient than the single neural network case.

3 Additive Schwarz algorithms for neural network approximation

In this section, we first review the one-level and two-level additive Schwarz algorithms that were proposed in our previous work [7] and their convergence results under the approximation error assumption on each local and coarse neural network solutions. We then introduce a partitioned neural network function to approximate the solution and propose iterative methods on the partitioned neural network function to find the convergent iterates to the solution. The iteration methods can be analyzed as the same way in our previous study [7] to give the same convergence result. As we will see in the numerical results later, the partitioned neural network function gives less optimization errors and thus it gives faster convergence than in the previous work [7].

Our method is developed for the following model elliptic problem in a bounded domain Ω, i.e., to find u in the Hilbert space $H^1(\Omega)$ satisfying

$$-\triangle u = f \text{ in } \Omega, \quad u = g \text{ on } \Omega, \tag{2}$$

where $H^1(\Omega)$ denotes the space of square integrable functions up to the first derivatives. In the one-level additive Schwarz method, for a given overlapping subdomain partition,$\{\Omega_i\}_i$ of the domain Ω, with an overlapping width δ, the following iterative

scheme is proposed to find its solution u. For a given $u^{(n)}$, the following problem in each subdomain Ω_i is solved to find $u_i^{(n+1)}$,

$$-\triangle u_i^{(n+1)} = f \text{ in } \Omega_i, \quad u_i^{(n+1)} = u^{(n)} \text{ in } \overline{\Omega} \setminus \Omega_i. \tag{3}$$

Using $u_i^{(n+1)}$, the next iterate is then formed to give

$$u^{(n+1)} = (1 - N\tau)u^{(n)} + \tau \sum_{i=1}^{N} u_i^{(n+1)},$$

where N is the number of subdomains in the partition and τ is a relaxation parameter. Let N_c be the maximum number of subdomains sharing the same geometric position in Ω. With $\tau \leq 1/N_c$, $u^{(n)}$ converges to the solution u of (2) under a suitably chosen space of functions, see [14, 16]. The algorithm can be further extended into a two-level method by introducing the coarse problem,

$$-\triangle w_0^{(n+1)} = f + \triangle u^{(n)} \text{ in } \Omega, \quad w_0^{(n+1)} = 0 \text{ on } \partial\Omega,$$

and by including the coarse problem solution to the iteration formula,

$$u^{(n+1)} = (1 - N\tau)u^{(n)} + \tau \left(\left(\sum_{i=1}^{N} u_i^{(n+1)} \right) + w_0^{(n+1)} \right).$$

In [7], following similarly as in the analysis for the variational inequalities [1], under the assumptions on the stable decomposition property and the strengthened Cauchy-Schwarz inequality, see [14, Section 2.3], the iterates $u^{(n)}$ converge to the exact solution u with the convergence rate $R(\tau)$,

$$a(u - u^{(n+1)}, u - u^{(n+1)}) \leq R(\tau) a(u - u^{(n)}, u - u^{(n)}), \tag{4}$$

where $a(u, v) = \int_\Omega \nabla u \cdot \nabla v \, dx$, and $R(\tau)$ is

$$R(\tau) = 1 - \frac{2}{2 + C_0}\tau + N_c^2\tau^2 \text{ and } R(\tau) = 1 - \frac{2}{2 + C_0}\tau + 2(N_c^2 + 1)\tau^2$$

in the one-level case and two-level case, respectively. In the above, the constant C_0 is that appears in the stable decomposition property. In a more detail, in the one-level case, the constant C_0 follows the growth of $N_c NH/\delta$ and in the two-level case, the constant C_0 follows the growth of $N_c H/\delta$, under the approximation property assumption on the coarse Hilbert subspace, see [14, Sections 3.5 and 3.6]. Combining our convergence analysis in (4) with the bound for C_0, we can thus conclude that for a suitable choice of τ, the iterates $u^{(n)}$ converge to u in the Hilbert space $H_0^1(\Omega)$,

$$|u^{(n+1)} - u|_1 \leq C|u^{(n)} - u|_1,$$

with the constant $C < 1$ increasing to 1 as N increasing in the one-level case, while with the constant C being robust as N increasing in the two-level case. The convergent rate C in the one-level method deteriorates as the more subdomains in the partition while it is robust to the increase of the number of subdomains in the two-level method, that have been also observed in additive Schwarz preconditioners to algebraic systems in classical numerical methods.

To find a neural network approximate solution, at each iteration in the additive Schwarz methods, we approximate the local problem solution and the coarse problem solution with neural network functions $U_i(x;\theta_i^{(n+1)})$ and $W_0(x;\theta_0^{(n+1)})$ and train the parameters $\theta_i^{(n+1)}$ and $\theta_0^{(n+1)}$ to minimize the cost functions related to each local problem and the coarse problem, respectively. The neural network iterates $U^{(n+1)}$ are then defined as

$$U^{(n+1)} = (1 - N\tau)U^{(n)} + \tau\left(\sum_{i=1}^{N} U_i^{(n+1)} + W_0^{(n+1)}(x,\theta_0^{(n+1)})\right),$$

where $U_i^{(n+1)}(x)$ are $U_i(x;\theta_i^{(n+1)})$ in Ω_i and $U^{(n)}(x)$ in the rest part, i.e., $\overline{\Omega} \setminus \Omega_i$. In the iteration method, we should store all the previous step parameters to obtain the resulting final step solution as a function of x, which is not desirable in the practical calculation.

To obtain a more practical method, we rewrite the above iteration formula as follows: for any x in Ω

$$U^{(n+1)}(x) = (1-|s(x)|\tau)U^{(n)}(x)+\tau\left(\sum_{i\in s(x)} U_i(x,\theta_i^{(n+1)}) + W_0^{(n+1)}(x,\theta_0^{(n+1)})\right), \quad (5)$$

where $s(x)$ denotes the set of subdomain indices sharing x and $|s(x)|$ denotes the number of elements in the set $s(x)$. We introduce

$$\widehat{U}^{(n+1)}(x) := \frac{1}{|s(x)|}\left(\sum_{i\in s(x)} U_i(x,\theta_i^{(n+1)}) + W_0^{(n+1)}(x,\theta_0^{(n+1)})\right) \quad (6)$$

and rewrite the above iteration formula into

$$U^{(n+1)}(x) = (1 - |s(x)|\tau)U^{(n)}(x) + |s(x)|\tau\widehat{U}^{(n+1)}(x).$$

For the iterates $\widehat{U}^{(n+1)}(x)$, they also converge to $u(x)$ in the L^2-norm, see [7], and the following practical one-level (without the term $W_0^{(n+1)}$ in the iteration formula in (5) and (6)) and two-level additive Schwarz algorithms are finally obtained:

Algorithm 1: One-level method (input: $U^{(0)}$, output: $\widehat{U}^{(n+1)}$)
Step 0: Let $U^{(0)}(x)$ be given and $n = 0$.

Step 1: Find $\theta_i^{(n+1)}$ in $U_i(x;\theta_i^{(n+1)})$ for

$$-\triangle u = f \text{ in } \Omega_i, \quad u = U^{(n)} \text{ on } \partial\Omega_i.$$

Step 2: Update $U^{(n+1)}$ at each data set $X_{\partial\Omega_i}$ as, see (6),

$$U^{(n+1)}(x) = (1-\tau|s(x)|)U^{(n)}(x) + \tau|s(x)|\widehat{U}^{(n+1)}.$$

Step 3: Go to **Step 1** with $n = n+1$ or set the output as $\widehat{U}^{(n+1)}$ if the stopping condition is met.

Algorithm 2: Two-level method (input: $U^{(0)}$, output: $\widehat{U}^{(n+1)}$)
Step 0: Let $U^{(0)}(x)$ be given and $n = 0$.
Step 1-1: Find $\theta_i^{(n+1)}$ in $U_i(x;\theta_i^{(n+1)})$ for

$$-\triangle u = f \text{ in } \Omega_i, \quad u = U^{(n)} \text{ on } \partial\Omega_i.$$

Step 1-2: Find $\theta_0^{(n+1)}$ in $W_0(x;\theta_0^{(n+1)})$ for

$$-\triangle w = f + \triangle U^{(n)} \text{ in } \Omega, \quad w = 0 \text{ on } \partial\Omega.$$

Step 2: Update $U^{(n+1)}$ at each data set $X_{\partial\Omega_i}$ as, see (6),

$$U^{(n+1)}(x) = (1-\tau|s(x)|)U^{(n)}(x) + \tau|s(x)|\widehat{U}^{(n+1)}.$$

Step 3: Go to **Step 1-1** with $n = n+1$ or set the output as $\widehat{U}^{(n+1)}$ if the stopping condition is met.

For the neural network iterates $U^{(n)}$ and $\widehat{U}^{(n)}$, the following convergence results are shown

$$|U^{(n+1)} - u|_1 \le |u^{(n+1)} - u|_1 + \frac{1}{1-C}\epsilon,$$

$$\|\widehat{U}^{(n+1)} - u\|_0 \le \frac{C_p}{\tau}\left(|u^{(n+1)} - u|_1 + |u^{(n)} - u|_1 + \frac{2}{1-C}\epsilon\right),$$

where ϵ denotes the approximation error in the local and coarse neural network solutions, $u^{(n)}$ are the iterates in the Hilbert space, C denotes the convergence rate in the Hilbert space iterates $u^{(n)}$, $\|\cdot\|_0$ denotes the L^2-norm, and C_p is the constant in the Poincare inequality, see [7].

As reported in numerical results in [7], the optimization errors also appear in the computed neural network solutions and they resulted in less accurate approximate solutions at each iteration. The resulting errors are observed to have high contrast near the boundary of the overlapping region, that is harder to be approximated by the coarse neural network function. Such optimization error behaviors in the neural network approximation have been analyzed by NTK theory [3, 15]. Regarding the local problems in our iteration method, the parameters in the local neural network function are trained to minimize the cost function, consisting of the residual loss

to the differential equation, and the residual loss to the boundary condition. The residual loss to the boundary condition is harder to optimize and such optimization behavior remains as the high contrast error near the overlapping region boundary.

To address such a drawback in our previous method, we form a partitioned neural network function to approximate the solution $u(x)$,

$$U(x;\theta_1,\cdots,\theta_N)=\sum_{i=1}^{N}\phi_i(x)U_i(x;\theta_i),$$

where $\phi_i(x)$ are a partition of unity functions for the given overlapping subdomain partition,

$$\sum_{i=1}^{N}\phi_i(x)=1,\ 0\le\phi_i(x)\le 1,\quad \phi_i(x)=0,\forall x\in\Omega\setminus\Omega_i.$$

We note that in [11] the parameters θ_i are trained to minimize the following global cost function without utilizing the partitioned neural network structure for parallel computing algorithms,

$$\begin{aligned}L(\theta_1,\cdots,\theta_N)=&\frac{1}{|X_\Omega|}\sum_{x\in X_\Omega}|\triangle U(x;\theta_1,\cdots,\theta_N)+f(x)|^2\\&+\frac{1}{|X_{\partial\Omega}|}\sum_{x\in X_{\partial\Omega}}|U(x;\theta_1,\cdots,\theta_N)-g(x)|^2.\end{aligned}$$

In our work, we propose an iteration method where each local parameters θ_i can be trained in parallel for a localized problem at each iteration. Such an iterative solution procedure is more desirable for the partitioned neural networks.

Our new iteration method is as follows:

Algorithm 3: PNN One-level method (input: $U^{(0)}$, output: $\widehat{U}^{(n+1)}$)
Step 0: Set the initial iterate $U^{(0)}=U(x;\theta_1^{(0)},\cdots,\theta_N^{(0)})$.
Step 1: Find $\theta_i^{(n+1)}$ in $\phi_i(x)U_i(x;\theta_i)$ to approximate the local problem solution;

$$-\triangle u=f+\triangle((1-\phi_i(x))U^{(n)})\text{ in }\Omega_i,$$

$$u=0\text{ on }\partial\Omega_i\cap\Omega,\quad u=g\text{ on }\partial\Omega_i\cap\partial\Omega.$$

Step 2: Set the next iterate;

$$U^{(n+1)}=(1-\alpha)U^{(n)}+\alpha\sum_{i=1}^{N}\phi_i(x)U_i(x;\theta_i^{(n+1)})$$

Step 3: If the stoping condition is met then set the output $\widehat{U}(x)=\sum_{i=1}^{N}\phi_i(x)U_i(x;\theta_i^{(n+1)})$, otherwise continue the iteration to go back to Step 1.

In Algorithm 3, for the case of floating subdomains, the zero boundary condition is already enforced in $\phi_i(x)U_i(x;\theta_i^{(n+1)})$ by the partition of unity function $\phi_i(x)$ and only the differential equation comes in the loss function. We can thus expect that the parameter optimization for such a local problem has less optimization errors than those in Algorithms 1 and 2. Its two-level version can be derived by adding the coarse correction term $W_0^{(n+1)}$ to the iterates

$$U^{(n+1)} = (1-\alpha)U^{(n)} + \alpha\left(\sum_{i=1}^{N}\phi_i(x)U_i(x;\theta_i^{(n+1)}) + W_0(x;\theta_0^{(n+1)})\right),$$

where $W_0(x;\theta_0^{(n+1)})$ is the neural network approximation to the global coarse problem, i.e., to find w in a coarse subspace V_0 of $H_1(\Omega)$ such that

$$-\triangle w = f + \triangle U^{(n)}, \text{ in } \Omega, \quad w = 0, \text{ on } \partial\Omega.$$

For the Algorithm 3 and its two-level version, their convergence can be shown by following similarly as in our previous work [7]. More rigorous convergence analysis will be provided in a complete version of the proceeding paper. We note that at each iteration the local parameters are fully trained for the given differential equation. Even the case, local parameter training cost per each iteration is much smaller than the training cost in the single large neural network. The number of training epochs is much smaller and the gradient update per epoch is also much cheaper for the smaller local neural network functions. The trade-off is that the total training cost in our proposed method also depends on the number of outer iterations. As the more subdomains in the partition, the more outer iterations are needed. It is thus important to include the coarse component to speed up the outer iterations.

4 Numerical results

For the proposed iterative methods, we consider the following simple one-dimensional model problem to compare their convergence behavior,

$$-u'' = f(x) \text{ in } (-1\ 1), \quad u(x) = 0 \text{ at } x = -1,\ 1,$$

where $f(x)$ is chosen to give the exact solution $u(x) = \sin(\pi x)$.

For the domain $(-1\ \ 1)$ we introduce an overlapping subdomain partition with 10 subdomains. We then consider a partitioned neural network with 10 local neural network functions, that are defined on each subdomains in the overlapping subdomain partition. For all the local neural network functions, the number of parameters is set as 106. We also use the same size of the coarse neural network function with 106 parameters in the two-level method. When training the parameters in the local and coarse neural network functions, we use 10000 training epochs in the gradient method, where we use the Adam optimizer with the learning rate 0.001.

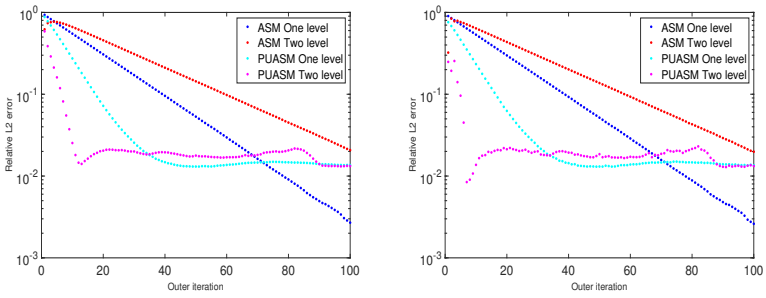


Fig. 1 Error decay history: Left figure (Error plots for $U^{(n)}$), Right figure (Error plots for $\widehat{U}^{(n)}$), ASM One level (Algorithm 1), ASM Two level (Algorithm 2), PUASM One level (Algorithm 3), PUASM Two level (Algorithm 3 with the coarse correction term).

We use randomly selected 100 training data points in each local and coarse problem parameter training.

In Fig. 1, the convergence history in the proposed methods is presented up to 100 outer iterations. The relative L^2-errors in the neural network approximate solutions at each outer iteration are plotted and compared. In the left figure, the errors for the neural network iterates $U^{(n)}$ to the exact solution are plotted for the four proposed methods. In the right figure, the errors for the practical neural network iterates $\widehat{U}^{(n)}$ are plotted, see (6) for the ASM Two level (Algorithm 2) and $\widehat{U}^{(n)} = (\sum_{i=1}^{N} \phi_i(x) U_i(x; \theta_i^{(n)})) + W_0(x; \theta_0^{(n)})$ for PUASM Two level (Algorithm 3 with the coarse correction term). The error plots in both figures show that the coarse correction term in the ASM Two level method does not help to speed up the convergence of the ASM One level. The convergence rate is even larger than the ASM One level method. As discussed earlier, this is related to the optimization error behaviors in the local neural network parameter training, that produce high contrast errors near the boundary of the overlapping region.

In the case of the PUASM Two level method, the coarse problem accelerates the convergence greatly at the early outer iterations. The local problem in the PUASM case has only the residual loss for the differential equation and the high contrast optimization error problems are alleviated in this case with the help of the partition of unity functions. However, at the later outer iterations, the errors can not be further reduced due to the practical implementation issue in the partition of unity functions. The practical implementation issue with the partition of unity functions needs a further investigation and our future research will be focused on proposing some new idea in forming and implementing the partition of unity functions that are suitable for neural network approximation.

Acknowledgements The first author is supported by NRF-2022R1A2C100388511.

References

1. Badea, L. and Wang, J. An additive Schwarz method for variational inequalities. *Mathematics of Computation* **69**(232), 1341–1354 (2000).
2. E, W., Han, J., and Jentzen, A. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Commun. Math. Stat.* **5**(4), 349–380 (2017).
3. Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems* **31** (2018).
4. Jagtap, A. D. and Karniadakis, G. E. Extended physics-informed neural networks (xpinns): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. *Communications in Computational Physics* **28**(5), 2002–2041 (2020).
5. Jagtap, A. D., Kharazmi, E., and Karniadakis, G. E. Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering* **365**, 113028 (2020).
6. Kim, H. H. and Yang, H. J. Domain decomposition algorithms for physics-informed neural networks. In: *Proceedings of the 26th International Conference on Domain Decomposition Methods* (2021).
7. Kim, H. H. and Yang, H. J. Additive Schwarz algorithms for neural network approximate solutions. *arXiv preprint arXiv:2211.00225* (2022).
8. Li, K., Tang, K., Wu, T., and Liao, Q. D3M: A deep domain decomposition method for partial differential equations. *IEEE Access* **8**, 5283–5294 (2019).
9. Li, W., Xiang, X., and Xu, Y. Deep domain decomposition method: Elliptic problems. In: *Mathematical and Scientific Machine Learning*, 269–286. PMLR (2020).
10. Long, Z., Lu, Y., and Dong, B. PDE-Net 2.0: learning PDEs from data with a numeric-symbolic hybrid deep network. *J. Comput. Phys.* **399**, 108925, 17 (2019).
11. Moseley, B., Markham, A., and Nissen-Meyer, T. Finite basis physics-informed neural networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. *arXiv preprint arXiv:2107.07871* (2021).
12. Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707 (2019).
13. Sirignano, J. and Spiliopoulos, K. DGM: a deep learning algorithm for solving partial differential equations. *J. Comput. Phys.* **375**, 1339–1364 (2018).
14. Toselli, A. and Widlund, O. *Domain decomposition methods—algorithms and theory*, *Springer Series in Computational Mathematics*, vol. 34. Springer-Verlag, Berlin (2005).
15. Wang, S., Yu, X., and Perdikaris, P. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics* **449**, 110768 (2022).
16. Xu, J. and Zikatanov, L. The method of alternating projections and the method of subspace corrections in Hilbert space. *J. Amer. Math. Soc.* **15**(3), 573–597 (2002).

Convergence Bounds for One-Dimensional ASH and RAS

Marcus Sarkis and Maksymilian Dryja

1 Introduction

The ASH and RAS methods were introduced in [2] and rate of convergence theory is still missing; apparently it does not fall into the abstract theory of Schwarz methods since the nonsymmetric terms are no compact perturbations of H^1-norms. As far as we know, the algebraic convergence theory using weighted max norms introduced in [3] is the only theoretical work which establishes convergence however no rate of convergence. Here, we introduce new techniques to analyze RAS and ASH for the one-dimensional case. Some of these techniques can be used to establish rate of convergence in higher dimensions and they will be discussed elsewhere.

Let

$$Au = f \tag{1}$$

be a system of linear algebraic equations corresponding to the finite difference approximations of the Poisson problem $-u^*_{xx} = f$ on the interval $\Omega = (0,1)$ with homogeneous Dirichlet boundary conditions on a uniform mesh in $\overline{\Omega}_h = \Omega_h \cup x_0 \cup x_{n+1}$, where $\Omega_h = \{x_j\}_{j=1}^n$ is the set of interior nodes of the mesh, and $x_0 = 0$ and $x_{n+1} = 1$ are the boundary nodes. Denote $h = 1/(n+1)$ as the mesh size. The discretization is obtained by setting $u(x_0) = u(x_{n+1}) = 0$ and

$$(-\Delta_h u)(x_j) = h^{-2}\left(-u(x_{j-1}) + 2u(x_j) - u(x_{j+1})\right) \quad j = 1, \cdots, n.$$

Denote the inner product in $L^2_h(0,1)$ (which we denote by V_h) by

$$(u,v) \equiv (u,v)_h = h\sum_{j=1}^{n} u(x_j)\, v(x_j) \quad \text{and denote} \quad \|v\|^2 = (v,v).$$

Marcus Sarkis
Worcester Polytechnic Institute, Worcester, USA, e-mail: msarkis@wpi.edu
Maksymilian Dryja
University of Warsaw, Poland. e-mail: dryja@mimuw.edu.pl

We introduce the matrix A

$$(v, Au) = (v, -\Delta_h u).$$

also as an operator defined on $L_h^2(0,1)$ with inner product $(\cdot,\cdot)$ and zero Dirichlet data at $x_0 = 0$ and $x_{n+1} = 1$. Here the matrix and the operator A will be denoted by the same letter. It is known that $(Av, v) = (\nabla I_h v, \nabla I_h v)_{L^2(0,1)}$ for $v \in V_h$, where $I_h v$ is the piecewise linear and continuous function with given $v(x_j)$ for $0 \le j \le n+1$.

In order to avoid proliferation of constants, we will often use the notation $A \preceq B$ ($A \succeq B$) to represent $A \le cB$ ($A \ge cB$) where the positive constant c is independent of h, H, δ, ℓ and r.

2 ASM, RAS, ASH and RASH methods

Let us decompose the nodes of Ω_h into N subdomains and without loss of generality assume that $m = n/N$ is an integer; see Fig. 1 with $n = 28$, $N = 4$ and $\ell = 2$. Define the nonoverlapping subdomains nodes of Ω_{ih}

$$\Omega_{ih} = \{x_{j+1}, x_{j+2}, \cdots, x_{j+m}\}, \quad \text{where} \quad j = (i-1)\,m,\ 1 \le i \le N.$$

Let $\ell \ge 0$ be an integer and let $\delta = (1+\ell)h$. We note that $\ell = 0$ is related a block diagonal preconditioner. Let the extended subdomain nodes of $\Omega_{i\delta}$ be obtained by extending by ℓ nodes to each side of Ω_{ih} inside Ω_h, that is,

$$\Omega_{i\delta} = \{x_{j+1-\ell}, x_{j+2-\ell}, \cdots, x_{j+m+\ell}\} \cap \Omega_h, \quad \text{where} \quad j = (i-1)m,\ 1 \le i \le N.$$

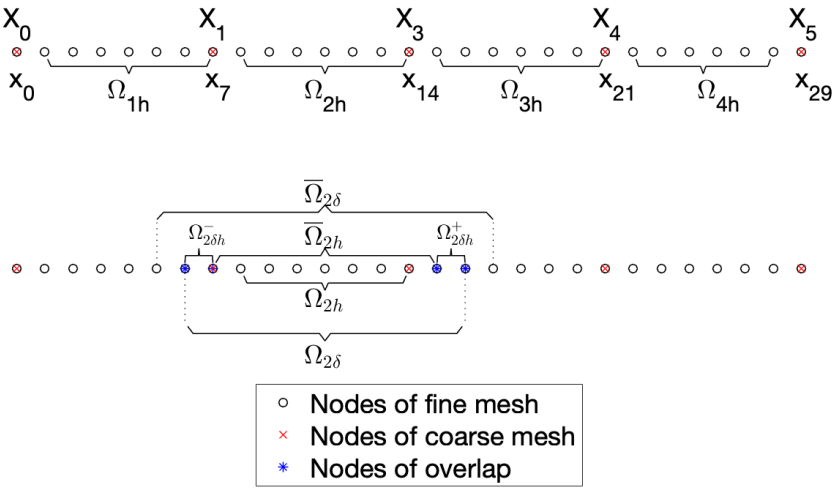


Fig. 1 (top) Ω_h with $n = 28$ nodes decomposed into four subdomains Ω_{ih} with V_0^1 coarse nodes. (below) The visualization of Ω_{ih}, $\overline{\Omega}_{ih}$, $\Omega_{i\delta}$, $\overline{\Omega}_{i\delta}$, and $\Omega_{i\delta h} = \Omega^-_{i\delta h} \cup \Omega^+_{i\delta h}$ when $i = 2$ and $\ell = 2$.

The mathematical analysis introduced below can also be extended easily for the case the domain decomposition is obtained by nonoverlapping subdomains elements. We also use the notation $\overline{\Omega}_{i\delta} = \{x_{j-\ell}, x_{j+1-\ell}, \cdots, x_{j+m+\ell+1}\} \cap \overline{\Omega}_h$ and $\overline{\Omega}_{ih} = \{x_j, x_{j+1}, \cdots, x_{j+m+1}\} \cap \overline{\Omega}_h$ to include their boundary nodes $\partial\Omega_{i\delta}$ and $\partial\Omega_{ih}$, respectively. Note that here and below j is a function of i given by $j = (i-1)m$ for $1 \le j \le N$.

Associated to each $\Omega_{i\delta}$, we introduce the restriction operator $R_{i\delta}$. In matrix terms, $R_{i\delta}$ is an $m_i \times n$ matrix such that $(R_{i\delta}v)(x_j) = v(x_j)$ for $x_j \in \Omega_{i\delta}$, $\forall v \in V_h$. Here, $m_1 = m + \ell$, $m_i = m + 2\ell$ for $2 \le i \le N-1$ and $m_N = m + \ell$. Define $A_{i\delta} = R_{i\delta} A R_{i\delta}^T$.

Associated to each $\Omega_{i\delta}$ and Ω_{ih}, we introduce the restriction operator $\tilde{R}_{ih}$. In matrix terms, $\tilde{R}_{ih}$ is an $m_i \times n$ matrix such that $(\tilde{R}_{ih}v)(x_j) = v(x_j)$ for $x_j \in \Omega_{ih}$ and $(\tilde{R}_{ih}v)(x_j) = 0$ for $x_j \in \Omega_{i\delta}\backslash\Omega_{ih}$, $\forall v \in V_h$. The superscript tilde notation is used to recall $\tilde{R}_{ih}$ maps to $\Omega_{i\delta}$ rather than Ω_{ih}. For analysis, we will also consider $R_{i\delta h} = R_{i\delta} - \tilde{R}_{ih}$ and denote $\Omega_{i\delta h} = \Omega_{i\delta}\backslash\Omega_{ih}$.

We will also consider preconditioners with a coarse problem. In order to mimic the 2D and 3D difficulties, we consider two cases of coarse spaces, the V_0^1 and the V_0^2 coarse spaces.

V_0^1 *case*: The coarse nodes are given by $\Omega_H = \{X_i\}_{i=1}^{N-1}$ and $\overline{\Omega}_H = \{X_i\}_{i=0}^{N}$ where $X_i = imh$ for $0 \le i \le N$ and with a zero Dirichlet data at $X_0 = x_0$ and $X_N = x_{n+1}$. In other words, the coarse node X_i is the rightmost node of Ω_{ih} for $1 \le i \le N-1$. In this case, the coarse nodes belong to the overlapping region (if $\ell \ge 1$).

V_0^2 *case*: The coarse nodes are given by $\Omega_H = \{X_i\}_{i=1}^{N}$ and $\overline{\Omega}_H = \{X_i\}_{i=0}^{N+1}$ where the coarse nodes are $X_i = (i-1)mh + \lfloor m/2 \rfloor h$ for $1 \le i \le N$, and $X_0 = x_0$ and $X_{N+1} = x_{n+1}$. Here, $\lfloor m/2 \rfloor$ is the integer part of $m/2$. In other words, the coarse node X_i is about the mid node of Ω_{ih}. This is the case the coarse nodes belong to just one extended subdomain when ℓ is not too large.

In both cases, zero Dirichlet data is imposed at the end nodes. The extrapolation operator R_0^T from Ω_H to Ω_h is the embedding piecewise linear and continuous coarse functions on the coarse triangulation $\overline{\Omega}_H$ to the fine mesh Ω_h. Define the coarse matrix by $A_0 = R_0 A R_0^T$.

The Additive Schwarz Method–ASM preconditioner is defined by

$$T_{\text{asm}} = B_{\text{asm}}^{-1} A = \left(\sum_{i=1}^{N} R_{i\delta}^T A_{i\delta}^{-1} R_{i\delta} + R_0^T A_0^{-1} R_0 \right) A.$$

The Restricted Additive Schwarz Method–RAS preconditioner is defined by

$$T_{\text{ras}} = B_{\text{ras}}^{-1} A = \left(\sum_{i=1}^{N} \tilde{R}_{ih}^T A_{i\delta}^{-1} R_{i\delta} + R_0^T A_0^{-1} R_0 \right) A.$$

The Additive Schwarz with Harmonic Overlap Method–ASH preconditioner is given by

$$T_{\text{ash}} = B_{\text{ash}}^{-1}A = \left(\sum_{i=1}^{N} R_{i\delta}^T A_{i\delta}^{-1}\tilde{R}_{ih} + R_0^T A_0^{-1} R_0\right) A.$$

The symmetrized RAS method, denoted by RASH, is defined by

$$T_{\text{rash}} = B_{\text{rash}}^{-1}A = \left(\sum_{i=1}^{N} \tilde{R}_{ih}^T A_{i\delta}^{-1}\tilde{R}_{ih} + R_0^T A_0^{-1} R_0\right) A.$$

By construction, the matrices B_{asm}^{-1}, B_{ras}^{-1}, B_{ash}^{-1} and B_{rash}^{-1} are well defined. It is well known that B_{asm}^{-1} is symmetric positive definite. The contributions of this paper proceedings are: 1) to show that B_{ras}^{-1} and B_{ash}^{-1} are nonsymmetric and positive definite on subspaces of V_h and, 2) to establish their lower and upper bounds for exact local solvers. Lower and upper bounds for B_{rash}^{-1} are also established.

The original system (1) is solved by Richardson iterative methods with an optimal relaxation parameter (or GMRES) with a B^{-1} left preconditioner, where B^{-1} will be B_{asm}^{-1}, B_{ras}^{-1}, B_{ash}^{-1} or B_{rash}^{-1}. We discuss two interpretations (residual and solution vectors) of the methods. Then the analysis of convergence of the discussed method is given. The Richardson iterative method for the solution vector is given by

$$u^{k+1} = u^k - \tau B^{-1}(Au^k - f), \tag{2}$$

where $\tau > 0$ is a relaxation parameter. By multiplying (2) by A and setting the residual vector $r^k = Au^k - f$ we get

$$r^{k+1} = r^k - \tau AB^{-1}r^k. \tag{3}$$

We recall that $(u, v) = h\sum_{i=1,n} u(x_i)v(x_i)$ and denote $\|u\|_C^2 = (u, Cu)$ for any symmetric positive definite matrix C. The convergence analysis of $\|u - u^k\|_A$-norm follows from the convergence analysis of (3) with the $\|r^k\|_{A^{-1}}$-norm, and vice-versa, since $r^k = A(u^k - u)$. A bound for the convergence rate for (3) with the optimal parameter τ_k, or for the GMRES on the A-norm, is given by the following well known lemma, for example, see Lemma C.11 of [4].

Lemma 1. *Assume that for any* $r \in \mathbb{R}^n$

$$\gamma_1(A^{-1}r, r) \le (B^{-1}r, r) \tag{4}$$

and

$$(AB^{-1}r, B^{-1}r) \le \gamma_2(A^{-1}r, r). \tag{5}$$

Then the iterative method (3) *converges with rate*

$$\|r^{k+1}\|_{A^{-1}} \le \rho_*^k \|r^k\|_{A^{-1}} \quad \textit{with optimal} \quad \tau_* = \gamma_1/\gamma_2 \quad \textit{and} \quad \rho_* = (1 - \gamma_1^2/\gamma_2)^{1/2}.$$

3 Reduction of the iterative scheme to a subspace

3.1 ASH inital correction

We first discuss B_{ash}^{-1} without the coarse problem. Let u^0 be determined by

$$u^0 = B_{\text{ash}}^{-1} Au = B_{\text{ash}}^{-1} f.$$

The problem (1) now reduces to solving $A\hat{u} = \hat{f}$ where $\hat{f} = f - Au^0$ and $\hat{u} = u - u^0$. Denote $\mathbb{R}^n$ as the Euclidean space, and denote $\mathbb{R}_{\text{ash}}^n \subset \mathbb{R}^n$ as the set of residual vectors which are zero at all nodes except at the nodes of $\cup_{i=1}^N \partial\Omega_{i\delta} \cap \Omega_h$. It is easy to see, by using that $\sum_{i=1}^N R_{i\delta}^T \tilde{R}_{ih} = I_n$ that $\hat{f} \in \mathbb{R}_{\text{ash}}^n$. Let $\mathbb{V}_{\text{ash}}^h = A^{-1}\mathbb{R}_{\text{ash}}^n$ be the space of discrete harmonic vectors on Ω_h except at the nodes of $\cup_{i=1}^N \partial\Omega_{i\delta} \cap \Omega_h$. Note that $\hat{u} \in V_{\text{ash}}^h$. We also note that the subspace $\mathbb{R}_{\text{ash}}^n$ is a natural choice since $A(u^k - u^{k-1}) \in \mathbb{R}_{\text{ash}}^n$ for the preconditioned Richardson with $\tau = 1$ without the initial correction. From now on, we assume this initial correction was performed and the superscript hat is dropped. Consider the Richardson method, with $u^0 = 0$,

$$u^{k+1} = u^k - \tau B_{\text{ash}}^{-1}(Au^k - f) \quad k = 0, 1, \cdots \tag{6}$$

It is not hard to see, by recursion, that $r^k \in \mathbb{R}_{\text{ash}}^n$ and $u^k \in V_{\text{ash}}^h$ for $k = 0, 1, 2, \cdots$.

Lemma 2. [1] *For* $u \in V_{\text{ash}}^h$

$$B_{\text{ash}}^{-1} Au = B_{\text{asm}}^{-1} Au.$$

Proof. It follows from $\tilde{R}_{ih} Au = R_{i\delta} Au$ for $u \in \mathbb{V}_{\text{ash}}^n$. □

As consequence, the upper and lower bounds for B_{asm}^{-1} on the space V_{ash} are also the upper and lower bounds for B_{ash}^{-1}. We note Lemma 2 also holds for the strip case in 2D and 3D since no more than two extended subdomains overlap the same node.

We now consider the ASH method with a coarse space. First note that the image of AR_0^T vanishes at all nodes except the coarse nodes. Therefore if there are no coarse nodes in any of the $\Omega_{i\delta h}$, then Lemma 2 holds and this is the V_0^2 case. Therefore, we consider coarse spaces where the coarse nodes are in the overlapping regions, which is the V_0^1 coarse space case. It is easy to see after the initial correction u^0, $\mathbb{R}_{\text{ash}}^n \subset \mathbb{R}^n$ is now the set of residual vectors which are zero at all nodes except for the nodes of $\cup_{i=1}^N \partial\Omega_{i\delta} \cap \Omega_h$ and at the coarse nodes. It easy to see that all the $u^k \in \mathbb{V}_{\text{ash}}^n := A^{-1}\mathbb{R}_{\text{ash}}^n$ and that Lemma 2 does not hold. New techniques are introduced below to treated this case.

3.2 RAS and RASH initial corrections

After an initial correction $\hat{u}^0 = B_{\text{ras}}^{-1} f$, $\mathbb{R}^n_{\text{ras}} \subset \mathbb{R}^n$ is now the set of RAS residual vectors which are zero at all nodes except for the nodes on $\cup_{i=1}^N \partial\Omega_{ih} \cap \Omega_h$ and at the coarse nodes. After a correction $\hat{u}^0 = B_{\text{ras}}^{-1} f$ or $\hat{u}^0 = B_{\text{rash}}^{-1} f$, $\mathbb{R}^n_{\text{rash}} = \mathbb{R}^n_{\text{ras}}$.

4 Lower and upper bounds for ASH, RAS and RASH methods

Note that $B_{\text{ras}}^{-1} \geq \gamma_1 A^{-1}$ is equivalent to $B_{\text{ash}}^{-1} \geq \gamma_1 A^{-1}$ on the space $\mathbb{R}^n$ since

$$(B_{\text{ras}}^{-1} r, r) = (r, B_{\text{ash}}^{-1} r) = (B_{\text{ash}}^{-1} r, r) \quad r \in \mathbb{R}^n. \tag{7}$$

We note however that the lower bound for B_{ash}^{-1} for $r \in \mathbb{R}^n_{\text{ash}}$ is not necessarily equivalent to the lower bound for B_{ras}^{-1} for $r \in \mathbb{R}^n_{\text{ras}}$, therefore, separate analyses are done for the ASH and RAS methods. In order to establish the lower bounds for the ASH and RAS, we introduce the following interesting result:

Lemma 3. *For any $r \in \mathbb{R}^n$,*

$$2(B_{\text{ash}}^{-1} r, r) = 2(B_{\text{ras}}^{-1} r, r) = (B_{\text{asm}}^{-1} r, r) + (B_{\text{rash}}^{-1} r, r) - \sum_{i=1}^N (A_{i\delta}^{-1} R_{i\delta h} r, R_{i\delta h} r). \tag{8}$$

Proof. First we add and subtract $\tilde{R}_{i\delta h}$ to obtain

$$(B_{\text{ash}}^{-1} r, r) = \sum_{i=1}^N (A_{i\delta}^{-1} R_{i\delta} r, \tilde{R}_{ih} r) + (A_0^{-1} R_0 r, R_0 r) = (B_{\text{asm}}^{-1} r, r) - \sum_{i=1}^N (A_{i\delta}^{-1} R_{i\delta} r, R_{i\delta h} r),$$

and using $R_{i\delta} = R_{i\delta h} + \tilde{R}_{ih}$ we have

$$\begin{aligned}
(B_{\text{ash}}^{-1} r, r) =& (B_{\text{asm}}^{-1} r, r) - \sum_{i=1}^N (A_{i\delta}^{-1} \tilde{R}_{ih} r, R_{i\delta h} r) - \sum_{i=1}^N (A_{i\delta}^{-1} R_{i\delta h} r, R_{i\delta h} r), \quad \text{hence,} \\
(B_{\text{ash}}^{-1} r, r) =& (B_{\text{asm}}^{-1} r, r) - \sum_{i=1}^N (A_{i\delta}^{-1} \tilde{R}_{ih} r, R_{i\delta} r) + \sum_{i=1}^N (A_{i\delta}^{-1} \tilde{R}_{ih} r, \tilde{R}_{ih} r) \\
& - \sum_{i=1}^N (A_{i\delta}^{-1} R_{i\delta h} r, R_{i\delta h} r)
\end{aligned}$$

and the lemma follows by adding and subtracting $(A_0^{-1} R_0 r, R_0 r)$. □

In order to use equation (8) to establish the lower bound of RAS and ASH, we need to understand the lower bound for RASH, which is treated at the end of this section.

We assume from now on that $\Omega_{(i+1)\,\delta} \cap \Omega_{(i-1)\,\delta} = \emptyset$, that is, the overlap $\delta = (1+\ell)h$ is not too large. We recall that $\ell = 0$ is the block Jacobi preconditioner and that ASH, RAS and RASH are all equal to the ASM.

We first consider the ASH lower bound with $B^{-1} = B^{-1}_{\text{ash}}$. Since the coarse space V_0^2 has already been treated in the previous section, in the next lemma we consider only the V_0^1 case.

Lemma 4. *For any $r \in \mathbb{R}^n_{\text{ash}}$, there exists $\gamma_1 = O(1 + \frac{H}{\delta})^{-1}$ for which* (4) *holds.*

Proof. The strategy of the proof is the following: Consider the equality (8) and use the following three steps:
Step 1: Consider the equality (8)
Step 2: Find a positive number c_1 such that

$$(A_{i\delta}^{-1} R_{i\delta h} r, R_{i\delta h} r) \le c_1 h^2 \|R_{i\delta h} r\|^2 \quad 1 \le i \le N.$$

Step 3: Find positive numbers c_2 and c_3 and let $0 \le \gamma \le 1$ such that

$$\sum_{i=1}^{N} \|R_{i\delta h} r\|^2 \le h^{-2} \sum_{i=1}^{N} \left(\gamma c_2 (A_{i\delta}^{-1} R_{i\delta} r, R_{i\delta} r) + (1-\gamma) c_3 (A_{i\delta}^{-1} \tilde{R}_{ih} r, \tilde{R}_{ih} r) \right).$$

Then using Steps 1 and 2 we obtain

$$\sum_{i=1}^{N} (A_{i\delta}^{-1} R_{i\delta h} r, R_{i\delta h} r)| \le \gamma c_1 c_2 (B^{-1}_{\text{asm}} r, r) + (1-\gamma) c_1 c_3 (B^{-1}_{\text{rash}} r, r).$$

Step 3: Choose a γ such that $\max\{\gamma c_1 c_2, (1-\gamma) c_1 c_3\} < 1$, independent of H, h and δ. Then use equality (8), and the RASH lower bound (see Lemma 8) and the ASM lower bound [4] to obtain the lower bound $O(1 + H/\delta)^{-1}$.

Step 1 Assume that $r \in \mathbb{R}^n_{\text{ash}}$ and let $u_{i\delta h} := A_{i\delta}^{-1} \tilde{R}_{i\delta h} r$. The $\Omega_{i\delta}$ is given by (see Fig. 1)

$$\Omega_{i\delta} = \{x_{j+1-\ell}, \cdots, x_{j+m+\ell}\} \cap \Omega_h, \quad j = j(i) = (i-1)\, m, \quad 1 \le i \le N$$

see Fig. 1, and let

$$\overline{\Omega}_{i\delta} = (x_{j-\ell} \cup \Omega_{i\delta} \cup x_{j+m+\ell+1}) \cap \overline{\Omega}_h.$$

Remember that $\Omega_{i\delta h} = \Omega_{i\delta} \backslash \Omega_{ih}$. Decompose $\Omega_{i\delta h} = \Omega^-_{i\delta h} \cup \Omega^+_{i\delta h}$, where

$$\Omega^-_{i\delta h} = \{x_{j+1-\ell}, \cdots x_j\} \cap \Omega_h \quad \text{and} \quad \Omega^+_{i\delta h} = \{x_{j+m+1}, \cdots x_{j+m+\ell}\} \cap \Omega_h.$$

Note that $\Omega^-_{1\delta h}$ and $\Omega^+_{N\delta h}$ are empty sets and $\Omega^-_{i\delta h} \subset \Omega_{(i-1)h}$ for $2 \le i \le N$, and $\Omega^+_{i\delta h} \subset \Omega_{(i+1)h}$ for $1 \le i \le N-1$.

The only node where $R_{i\delta h} r$ is not necessarily zero is at $x_j \in \Omega^-_{i\delta h}$ since for the coarse nodes of V_0^1, it has no coarse nodes in $\Omega^+_{i\delta h}$. We have

$$(A_{i\delta}^{-1} R_{i\delta h} r, R_{i\delta h} r) = (u_{i\delta h}, R_{i\delta h} r) = h u_{i\delta h}(x_j) r(x_j) = \|R_{i\delta h} r\| h^{1/2} |u_{i\delta h}(x_j)|.$$

Note that $u_{i\delta h} = A_{i\delta}^{-1} R_{i\delta h} r$ vanishes at $x_{j-\ell}$ (the node on the boundary of $\overline{\Omega}_{i\delta}$ inside $\Omega_{(i-1)h}$), and it is linear (harmonic) from $x_{j-\ell}$ to x_j. We can relate $|u_{i\delta h}(x_j)|$ with its energy on the interval $(x_{j-\ell}, x_j)$ since $u_{i\delta h}(x_{j-\ell}) = 0$ and

$$hu_{i\delta h}^2(x_j) = \ell h^2 \,(\frac{u_{i\delta h}(x_j) - u_{i\delta h}(x_{j-\ell})}{h\ell})^2 \ell h = \ell h^2 |u_{i\delta h}|^2_{H^1(x_{j-\ell}, x_j)},$$

and

$$|u_{i\delta h}|^2_{H^1(x_{j-\ell}, x_j)} \leq (A_{i\delta} u_{i\delta h}, u_{i\delta h}) = (A_{i\delta}^{-1} R_{i\delta h} r, R_{i\delta h} r).$$

Hence, we obtain $c_1 = \ell$.

Step 2 Denote $R_{i\delta h}^{(i-1)} = R_{(i-1)\delta} R_{i\delta}^T R_{i\delta h}$. Easy to see that

$$\begin{aligned}\|R_{i\delta h} r\|^2 =& r(x_j)^2 \\ =& \frac{\gamma}{2}(R_{(i-1)\delta} r, R_{i\delta h}^{(i-1)} r) + \frac{\gamma}{2}(R_{i\delta} r, R_{i\delta h} r) + (1-\gamma)(\tilde{R}_{(i-1)h} r, R_{i\delta h}^{(i-1)} r).\end{aligned}$$

Let us first bound $(R_{(i-1)\delta} r, R_{i\delta h}^{(i-1)} r)$. Denote $u_{(i-1)\delta} = A_{(i-1)\delta}^{-1} R_{(i-1)\delta} r$. First see that $u_{(i-1)\delta}$ vanishes at $x_{j+1+\ell}$ (the rightmost node of $\overline{\Omega}_{(i-1)\delta}$), is linear from x_j (a coarse node) to $x_{j+1+\ell}$, and is linear from $x_{j-\ell}$ (the leftmost node of $\overline{\Omega}_{i\delta}$) to x_j. Hence, we obtain $u_{(i-1)\delta} = A_{(i-1)\delta}^{-1} R_{(i-1)\delta} r$,

$$(R_{(i-1)\delta} r, R_{i\delta h}^{(i-1)} r) = (A_{(i-1)\delta} u_{(i-1)\delta}, R_{i\delta h}^{(i-1)} r) = (A_{(i-1)\delta} u_{(i-1)\delta}, E(R_{i\delta h}^{(i-1)} r)),$$

where $E(R_{i\delta h}^{(i-1)} r) \in V_h(\Omega_{(i-1)\delta})$ is an extension of $r(x_j)$, where $(E(R_{i\delta h}^{(i-1)} r)(x_j) = r(x_j)$, vanishes at $x_{j+1+\ell}$ and $x_{j-\ell}$ and is linear in the subintervals $(x_{j-\ell}, x_j)$ and $(x_j, x_{j+1+\ell})$. We have

$$(R_{(i-1)\delta} r, R_{i\delta h}^{(i-1)} r) \leq |u_{(i-1)\delta}|_{H^1(x_{j-\ell}, x_{j+1+\ell})} |E(R_{i\delta h}^{(i-1)} r)|_{H^1(x_{j-\ell}, x_{j+1+\ell})}.$$

And using the same arguments as above, we have

$$|E(R_{i\delta h}^{(i-1)} r)|^2_{H^1(x_{j-\ell}, x_{j+1+\ell})} = \frac{1}{h^2}\left(\frac{1}{\ell} + \frac{1}{\ell+1}\right) h r^2(x_j).$$

Hence,

$$(R_{(i-1)\delta} r, R_{i\delta h}^{(i-1)} r) \leq h^{-1}\left(\frac{1}{\ell} + \frac{1}{\ell+1}\right)^{1/2} |u_{(i-1)\delta}|_{H^1(x_{j-\ell}, x_{j+1+\ell})} \|R_{i\delta h} r\|.$$

Now let us bound $(\tilde{R}_{(i-1)h} r, R_{i\delta h}^{(i-1)} r)$. Define $u_{(i-1)h} = A_{(i-1)\delta}^{-1} \tilde{R}_{(i-1)h} r$ and see that $u_{(i-1)h}$ is also harmonic on the subintevals $(x_{j-\ell}, x_j)$ and $(x_j, x_{j+1+\ell})$. Using the same arguments as above we obtain

$$(R_{(i-1)h} r, R_{i\delta h}^{(i-1)} r) \leq h^{-1}\left(\frac{1}{\ell} + \frac{1}{\ell+1}\right)^{1/2} |u_{(i-1)h}|_{H^1(x_{j-\ell}, x_{j+1+\ell})} \|R_{i\delta h} r\|.$$

Now let us bound $(R_{i\delta}r, R_{i\delta h}r$ and let $u_{i\delta} = A_{i\delta}^{-1}R_{i\delta}r)$. Using similar arguments

$$(R_{i\delta}r, R_{i\delta h}r) \le h^{-1}\left(\frac{1}{\ell} + \frac{1}{\ell+1}\right)^{1/2} |u_{i\delta}|_{H^1(x_{j-\ell}, x_{j+1+\ell})} \|R_{i\delta h}r\|.$$

Hence, we obtain $2c_2 = c_3 = \left(\frac{1}{\ell} + \frac{1}{\ell+1}\right)$

Step 3 A proper choice is $\gamma = 2/3$ which gives $\gamma c_1 c_2 = (1-\lambda)c_1 c_3 < 2/3$. □

We now consider the RAS lower bound for $B^{-1} = B_{\text{ras}}^{-1}$ for both V_0^1 and V_0^2. Independently if we use V_0^1 or V_0^2, we have nonzero residuals at x_j, x_{j+1}, x_{j+m} and x_{j+m+1}. If V_0^2 is used, a nonzero residuals will show up also at $x_{j+\lfloor m/2 \rfloor}$.

Lemma 5. *For any $r \in \mathbb{R}^n_{\text{ras}}$, there exists $\gamma_1 = O(1 + \frac{H}{h})^{-1}$ for which* (4) *holds.*

Proof. We follow the same strategy as in the proof of the previous lemma.

Step 1 Assume $2 \le i \le N-1$. Decompose

$$R_{i\delta h} = R^-_{i\delta h} + R^+_{i\delta h},$$

where $R^-_{i\delta h}r$ and $R^+_{i\delta h}r$ vanish on $\Omega_{i\delta}$ except at the nodes x_j and x_{j+m+1}, respectively. We have

$$(A_{i\delta}^{-1}R_{i\delta h}r, R_{i\delta h}r) = hu_{i\delta h}(x_j)r(x_j) + hu_{i\delta h}(x_{j+m+1})r(x_{j+m+1})$$

and the $|u_{i\delta h}(x_j)|$ and $|u_{i\delta h}(x_{j+m+1})|$ are now controlled by the energy on the intervals $(x_{j-\ell}, x_j)$ and $(x_{j+m+1}, x_{j+m+1+\ell})$, respectively. Using the same arguments as above we obtain

$$(A_{i\delta}^{-1}R_{i\delta h}r, R_{i\delta h}r) \le h^2\,\ell\left(\|R^-_{i\delta h}r\|^2 + \|R^+_{i\delta h}r\|^2\right).$$

Step 3 Assume $2 \le i \le N-1$. Denote $R_{i\delta h}^{(i-1)-} = R_{(i-1)\delta}R_{i\delta}^T R_{i\delta h}$. We have

$$\|R^-_{i\delta h}r\|^2 = r(x_j)^2 = \gamma(R_{(i-1)\delta}r, R_{i\delta h}^{(i-1)-}r) + (1-\gamma)(\tilde{R}_{(i-1)h}r, R_{i\delta h}^{(i-1)-}r).$$

The $R^+_{i\delta h}$ case can be treated similarly. A difference now with respect to the ASH analysis is also that $u_{i\delta}$ now is not discrete harmonic at x_{j+1}, therefore, $E(R_{i\delta h}^{(i-1)-}r)$ can be extended from $r(x_j)$ linearly on the interval $(x_{j-\ell}, x_j)$ however with just a zero extension on (x_j, x_{j+1}). Another difference is that we cannot include the term $(R_{i\delta}r, R^-_{i\delta h}r)$ because the estimates would overlap with estimates for $(R_{i\delta}r, R_{(i-1)\delta h}^{(i)+}r)$ on the interval (x_j, x_{j+1}). Fortunately, the region where $u_{(i-1)h}$ and $u_{(i-1)\delta}$ now are harmonic in the larger region from $x_{j-m+\lfloor m/2 \rfloor}$ (the midpoint of Ω_{ih}) to x_j. Denote $L_i^- = (x_{j-m+\lfloor m/2 \rfloor}, x_{j+1+\ell})$. We obtain

$$h^2\|R_{i\delta h}^- r\|^2 \leq \gamma\left(\frac{1}{m-\lfloor m/2\rfloor}+1\right)|u_{(i-1)\delta}|^2_{H^1(L_i^-)} + (1-\gamma)\left(\frac{1}{m-\lfloor m/2\rfloor}+\frac{1}{1+\ell}\right)|u_{(i-1)h}|^2_{H^1(L_i^-)}.$$

Gathering Steps 1 and 2 together we obtain

$$\sum_{i=1}^{N}(A_{i\delta}^{-1}R_{i\delta h}r, R_{i\delta h}r) \leq \gamma\left(1+\frac{\ell}{\lfloor m/2\rfloor}\right)(B_{\text{asm}}^{-1}r,r) + (1-\gamma)\left(\frac{\ell}{1+\ell}+\frac{\ell}{\lfloor m/2\rfloor}\right)(B_{\text{rash}}^{-1}r,r).$$

Step 3 Let us choose $\gamma = 1/(2+\ell)$, that is, when $\gamma\ell = (1-\gamma)\frac{\ell}{1+\ell}$. We obtain

$$(1+\ell/2+o(1))(B_{\text{ras}}^{-1}r,r) \geq (B_{\text{asm}}^{-1}r,r) + (B_{\text{rash}}^{-1}r,r),$$

where $o(1)$ is a tiny positive number when m is large compared to ℓ. The result follows from the lower bounds for ASM and RASH since $O(1+H/\delta) * (1+\delta/h+o(1)) = O(1+H/h)$. □

We now consider the ASH upper bound.

Lemma 6. *For all $r \in \mathbb{R}^n_{\text{ash}}$, there exists $\gamma_1 = O(1)$ for which* (5) *holds.*

Proof. Since a node does not belong to more than two extended subdomains, we have

$$(AB_{\text{ash}}^{-1}r, B_{\text{ash}}^{-1}r) \leq 3\sum_{i=1}^{N}\left(AR_{i\delta}^T A_{i\delta}^{-1}\tilde{R}_{ih}r, R_{i\delta}^T A_{i\delta}^{-1}\tilde{R}_{ih}r\right) + 3\left(AR_0^T A_0^{-1}R_0 r, R_0^T A_0^{-1}R_0 r\right)$$

and see that

$$\left(AR_0^T A_0^{-1}R_0 r, R_0^T A_0^{-1}R_0 r\right) = (R_0 r, A_0^{-1}R_0 r),$$
$$\left(AR_{i\delta}^T A_{i\delta}^{-1}\tilde{R}_{ih}r, R_{i\delta}^T A_{i\delta}^{-1}\tilde{R}_{ih}r\right) = (A_{i\delta}^{-1}\tilde{R}_{ih}r, \tilde{R}_{ih}r)$$

and using the same analysis of Step 2 of Lemma 4 with $\gamma = 1$, and the classical ASM upper bounds

$$(A_{i\delta}^{-1}\tilde{R}_{ih}r, \tilde{R}_{ih}r) \leq 2(A_{i\delta}^{-1}R_{i\delta}r, R_{i\delta})r + 2(A_{i\delta}^{-1}R_{i\delta h}r, R_{i\delta h}r) \leq (2+\ell(\frac{1}{\ell}+\frac{1}{1+\ell}))(A_{i\delta}^{-1}R_{i\delta}r, R_{i\delta}r).$$

□

We now consider the RAS upper bound.

Lemma 7. *For all $r \in \mathbb{R}^n_{\text{ras}}$, there exists $\gamma_2 = O(1+\ell)$ for which* (5) *holds.*

Proof. Following the initial steps of the proof of Lemma 6, we now need to estimate

$$\left(\tilde{R}_{ih}^T A_{i\delta}^{-1} R_{i\delta} r, A\tilde{R}_{ih}^T A_{i\delta}^{-1} R_{i\delta} r\right) = (\tilde{R}_{ih}^T u_{i\delta}, A\tilde{R}_{ih}^T u_{i\delta}) \quad \text{where} \quad u_{i\delta} = A_{i\delta}^{-1} R_{i\delta} r.$$

We have

$$\begin{aligned}(\tilde{R}_{ih}^T u_{i\delta}, A\tilde{R}_{ih}^T u_{i\delta}) =& |u_{i\delta}|^2_{H^1(x_{j+1}, x_{j+m})} + \frac{1}{h} u_{i\delta}(x_{j+1})^2 + \frac{1}{h} u_{i\delta}(x_{j+m})^2 \\ \leq& (1+\ell)(u_{i\delta}, A_{i\delta} u_{i\delta}).\end{aligned}$$

The result follows from the classical ASM upper bound [4]. □

Due to space limitations and since the analysis for RASH follows the classical abstract Schwarz theory for positive symmetric definite operators, the proofs for the RASH lower and upper bounds are ommited.

Lemma 8. *For any $r \in \mathbb{R}^n$, there exists $\gamma_1 = O(1 + \frac{H}{\delta})^{-1}$ for which* (4) *holds.*

Lemma 9. *For all $r \in \mathbb{R}^n_{\text{ras}}$, there exists $\gamma_2 = O(1+\ell)^2$ for which* (5) *holds.*

Final Remark: The techniques used in the proofs for the two-level ASH and RAS hold also for their one-level versions, where in Step 3 we replace the lower bounds for the ASM and RASH from $O(1 + H/\delta)$ by $O(1 + 1/H\delta)$.

5 Numerical section and conclusions and future directions

We consider $\Omega = (0, 1)$ and fix $H/h = 64$ and $1/H = 8$ and vary ℓ. We now test numerically the optimal lower and upper bounds of Lemma 1 by finding the smallest eigenvalue of $\frac{1}{2}(B^{-1} + B^{-T})r = \lambda_1 A^{-1}$ and the largest eigenvalue of $B^{-T} A B^{-1} v = \lambda_2 A^{-1}$. Here B^{-T} stands for the transpose of B^{-1}. The convergence rate of GMRES or the Richardson with optimal parameter is related to $\sqrt{1 - (\gamma_1/\sqrt{\gamma_2})^2}$, hence, we provide numerically γ_1 and $\sqrt{\gamma_2}$.

In Table 1, γ_1 and $\sqrt{\gamma_2}$ (in parenthesis) are provided for ASH, RAS, RASH and ASM with no coarse space. The generalized eigenvalue problems described above are solved on reduced spaces, that is, on the subspace $\mathbb{R}^n_{\text{ash}}$ for ASH and ASM methods, and on the subspace $\mathbb{R}^n_{\text{ras}}$ for RAS and RASH. As predicted by Lemma 2, ASH and ASM methods are the same method and satisfy the $O(1+1/(H\delta))^{-1}$ (since we have no coarse space) for the lower bound and the $O(1)$ for the upper bound. The theory for the RASH method is also sharp by Lemmas 8 and 9. Clearly, RASH is not a good method due to mostly the upper bound. We were successful in showing that B_{ras}^{-1} is positive on the subspace $\mathbb{R}^n_{\text{ras}}$ however we can see from the Table 1 that the theoretical upper and lower bounds are not sharp by a $O(1 + \ell)$ factor. It is an open problem to improve both bounds.

In Table 2, we run the previous test except that we add the coarse space V_0^2. The conclusions are similar except that the lower bounds are related to $O(1 + H/\delta)^{-1}$.

The techniques introduced here allowed us to obtain the first results on convergence rate and positiveness of $B_{\rm ras}^{-1}$ and $B_{\rm ash}^{-1}$. We also understand why $B_{\rm rash}^{-1}$ is not a good method. Some open problems are:
1) Is it possible to improve the lower and upper bounds for $B_{\rm ras}^{-1}$?
2) Is it possible to extend the new theory to the space $\mathbb{R}^n$ rather than for the reduced spaces, and also for inexact local solvers?, and
3) The extension of the new theory to the two-dimensional case, with and without a coarse space, and with or without cross points.

Table 1 No coarse space. The reduced systems: $\min \lambda_1$ and in parenthesis $\max \sqrt{\lambda_2}$

prec	$\ell = 0$	$\ell = 1$	$\ell = 2$	$\ell = 3$
ASH	0.0012(1.9988)	0.0035(1.9965)	0.0059(1.9941)	0.0083(1.9917)
RAS	0.0012(1.9988)	0.0035(1.9965)	0.0059(1.9941)	0.0083(1.9919)
RASH	0.0012(1.9988)	0.0024(3.9931)	0.0035(5.9830)	0.0047(7.9690)
ASM	0.0012(1.9988)	0.1058(1.9965)	0.1594(1.9941)	0.0083(1.9917)

Table 2 Coarse space V_0^2. The reduced systems: $\min \lambda_1$ and in parenthesis $\max \sqrt{\lambda_2}$

prec	$\ell = 0$	$\ell = 1$	$\ell = 2$	$\ell = 3$
ASH	0.0491(2.1180)	0.1058(2.2045)	0.1594(2.2638)	0.2100(2.3119)
RAS	0.0491(2.1180)	0.1058(2.2412)	0.1592(2.3730)	0.2097(2.5122)
RASH	0.0491(2.1180)	0.0767(4.0147)	0.1028(6.0013)	0.1274(7.9861)
ASM	0.0491(2.1180)	0.1058(2.2045)	0.1594(2.2638)	0.2100(2.3119)

References

1. Cai, X.-C., Dryja, M., and Sarkis, M. Restricted additive Schwarz preconditioners with harmonic overlap for symmetric positive definite linear systems. *SIAM Journal on Numerical Analysis* **41**(4), 1209–1231 (2003).
2. Cai, X.-C. and Sarkis, M. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing* **21**(2), 792–797 (1999).
3. Frommer, A. and Szyld, D. B. An algebraic convergence theory for restricted additive Schwarz methods using weighted max norms. *SIAM Journal of Numerical Analysis* **39**(2), 463–479 (2001).
4. Toselli, A. and Widlund, O. *Domain Decomposition Methods-Algorithms and Theory*, vol. 34. Springer Science & Business Media (2006).

Part II
Talks in Minisymposia

Weak Scalability of Domain Decomposition Methods for Discrete Fracture Networks

Stefano Berrone and Tommaso Vanzan

1 Introduction

Discrete Fracture Networks (DFNs) are complex three-dimensional structures characterized by the intersections of planar polygonal fractures, and are used to model flows in fractured media. Despite being suitable for Domain Decomposition (DD) techniques, there are relatively few works on the application of DD methods to DFNs, see, e.g., [1, 7] and references therein.

In this manuscript, we present a theoretical study of Optimized Schwarz Methods (OSMs) applied to DFNs. Interestingly, we prove that the OSMs can be weakly scalable (that is, they converge to a given tolerance in a number of iterations independent of the number of fractures) under suitable assumptions on the domain decomposition. This contribution fits in the renewed interest on the weak scalability of DD methods after the works [2, 4, 3], which showed weak scalability of DD methods for specific geometric configurations, even without coarse spaces.

Despite simplifying assumptions which may be violated in practice, our analysis provides heuristics to minimize the computational efforts in realistic settings. Finally, we emphasize that the methodology proposed can be straightforwardly generalized to study other classical DD methods applied to DFNs (see, e.g., [3]).

2 Scalability analysis for one-dimensional DFNs

We start considering a simplified DFN made of one-dimensional fractures F_i, $i = 1, \dots, N$ arranged in a staircase fashion depicted in Fig 1. The DFN is $\Omega := \cup_{i=1}^{N} F_i$.

Stefano Berrone
Politecnico di Torino, Italy, e-mail: stefano.berrone@polito.it

Tommaso Vanzan
Ecole Polytecnique Fédérale de Lausanne, Switzerland, e-mail: tommaso.vanzan@epfl.ch

The boundary of the fractures is denoted with ∂F_i and it holds $\partial\Omega = \cup_{i=1}^{N} \partial F_i$. Further, $\partial\Omega$ can be decomposed into a Dirichlet boundary Γ_D and a Neumann boundary Γ_N, so that $\partial\Omega = \Gamma_D \cup \Gamma_N$. The intersections between fractures are called traces and are denoted by S_m, $m = 1, \ldots, N-1 =: M$. We assume that both the vertical and

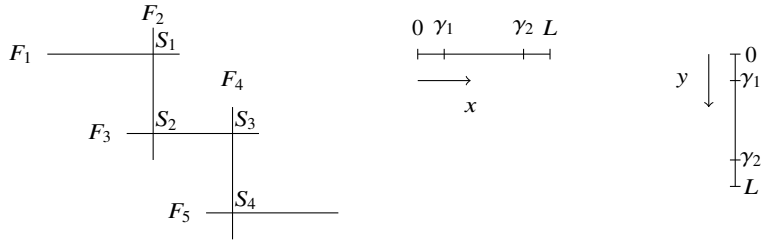


Fig. 1 Geometry of the simplified DFN and of its one-dimensional fractures.

horizontal fractures have two traces located at $\tau = \gamma_1$ and $\tau = \gamma_2$ with $\gamma_1 < \gamma_2$, (τ being the local coordinate), except the first and last fracture. The mathematical DFN model consists in the coupled system of partial differential equations for the hydraulic heads u_j,

$$-\nu_j \partial_{\tau_j \tau_j} u_j = f \quad \text{in } F_j, \quad \mathcal{B}_j(u) = 0 \quad \text{on } \partial F_j, \quad j = 1, \ldots, N, \tag{1}$$

$$u_{|F_i} = u_{|F_{i+1}} \quad \text{on } S_i, \ i = 1, \ldots, M, \tag{2}$$

$$\left[\left[\frac{\partial u_i}{\partial \tau_i}\right]\right] + \left[\left[\frac{\partial u_{i+1}}{\partial \tau_{i+1}}\right]\right] = 0 \quad \text{on } S_i, \ i = 1, \ldots, M, \tag{3}$$

where $\mathcal{B}_j$ represent boundary conditions (b.c.) (specified later), ν_j is the local diffusion coefficient, and $[[v]]$ is the jump of v across the intersection of fractures. The local solutions u_j are coupled through (2)–(3) which enforce continuity of the hydraulic heads, and balance between the jumps of the co-normal derivatives across the traces.

System (1)–(3) is clearly prone to a DD approach. We consider a nonoverlapping DD in which each subdomain corresponds to a single fracture, and the optimized Schwarz method (OSM) that, starting from an initial guess u_j^0 computes for $n = 1, 2, \ldots$ until convergence

$$\begin{aligned}
&-\nu_j \partial_{\tau_j \tau_j} u_j^n = f_j \quad \text{in } F_j, \quad \mathcal{B}_j(u_j^n) = 0 \quad \text{on } \partial F_i, \\
&\left[\left[\frac{\partial u_j^n}{\partial \tau_j}\right]\right] + s_{j-1}^+ u_j^n = -\left[\left[\frac{\partial u_{j-1}^{n-1}}{\partial \tau_{j-1}}\right]\right] + s_{j-1}^+ u_{j-1}^{n-1} \quad \text{on } S_{j-1}, \\
&\left[\left[\frac{\partial u_j^n}{\partial \tau_j}\right]\right] + s_j^- u_j^n = -\left[\left[\frac{\partial u_{j+1}^{n-1}}{\partial \tau_{j+1}}\right]\right] + s_j^- u_{j+1}^{n-1} \quad \text{on } S_j.
\end{aligned} \tag{4}$$

for $j = 2, \ldots, N-1$, while for $j = 1, N$,

$$
\begin{aligned}
&-\nu_1\partial_{\tau_1\tau_1}u_1^n = f_1 \text{ in } F_1,\ \mathcal{B}_1(u_1^n) = 0,\ -\nu_N\partial_{\tau_N\tau_N}u_N^n = f_N \text{ in } F_N,\ \mathcal{B}_N(u_N^n) = 0,\\
&\left[\left[\frac{\partial u_1^n}{\partial\tau_1}\right]\right] + s_1^- u_1^n = -\left[\left[\frac{\partial u_2^{n-1}}{\partial\tau_2}\right]\right] + s_1^- u_2^{n-1} \text{ on } S_1, \qquad (5)\\
&\left[\left[\frac{\partial u_N^n}{\partial\tau_N}\right]\right] + s_{N-1}^+ u_N^n = -\left[\left[\frac{\partial u_{N-1}^{n-1}}{\partial\tau_{N-1}}\right]\right] + s_{N-1}^+ u_{N-1}^{n-1} \text{ on } S_{N-1}.
\end{aligned}
$$

The functions f_j are the restrictions of the force term on the fracture F_j and $s_j^{+,-}$, $j = 1,\dots,M$ are positive parameters.

To carry out the scalability analysis, we assume for the sake of simplicity that $s_j^{+,-} = p \in \mathbb{R}^+$ and $\nu_j = 1$ for all j. We study later how to optimize the choice for $s_j^{+,-}$. We first discuss the case in which every $\mathcal{B}_j$ represents a Dirichlet boundary condition, and then we treat the case in which Neumann b.c. are imposed everywhere, except at the left boundary of F_1 (source fracture) and at the right boundary of F_N. More general configurations can be included straightforwardly in our analysis.

Due to the linearity of the problem, we define the errors $e_j^n := u - u_j^n$ and study their convergence to zero. The errors e_j satisfy an error system obtained setting $f_j = 0$ in (4)–(5). Inside each fracture, e_j is harmonic and has the analytical expression

$$
\begin{aligned}
e_1^n &= \frac{\hat{e}_1^n\tau_1}{\gamma_2}\chi_{[0,\gamma_2]} + \frac{\hat{e}_1^n(L-\tau_1)}{L-\gamma_2}\chi_{[\gamma_2,L]}, \qquad (6)\\
e_j^n &= \frac{\hat{e}_j^{1,n}\tau_j}{\gamma_1}\chi_{[0,\gamma_1]} + \frac{\hat{e}_j^{1,n}(\gamma_2-\tau_j) + \hat{e}_j^{2,n}(\tau_j-\gamma_1)}{\gamma_2-\gamma_1}\chi_{[\gamma_1,\gamma_2]} + \frac{\hat{e}_j^{2,n}(L-\tau_j)}{L-\gamma_2}\chi_{[\gamma_2,L]},\\
e_N^n &= \frac{\hat{e}_N^n\tau_N}{\gamma_1}\chi_{[0,\gamma_1]} + \frac{\hat{e}_N^n(L-\tau_N)}{L-\gamma_1}\chi_{[\gamma_1,L]},
\end{aligned}
$$

$j = 2,\dots,N$. Unknown coefficients are collected into $\mathbf{e}^n := (\hat{e}_1^n, \hat{e}_2^{1,n}, \hat{e}_2^{2,n}, \dots, \hat{e}_N^n)^\top \in \mathbb{R}^{\tilde{N}}$, $\tilde{N} := 2(N-2)+2$, and represent the values of the error functions at the traces, while $\chi_{[a,b]}$ are characteristic functions which satisfy $\chi(\tau) = 1$ if $\tau \in [a,b]$ and zero otherwise. Inserting these expressions into the transmission conditions (2)-(3), we aim to express $\hat{e}_j^{i,n}$ in terms of the coefficients of the errors in fractures $j-1$ and $j+1$ at iteration $n-1$. A direct calculation, which we omit due to space limitation (see [8] for more details) leads to the recurrence relation $\mathbf{e}^n = T_N^D\mathbf{e}^{n-1} = M_N^{-1}N_N\mathbf{e}^{n-1}$, where $M_N, N_N \in \mathbb{R}^{\tilde{N},\tilde{N}}$ have the block structure

$$
M_N := \begin{pmatrix} F_1 & & & & \\ & F_2 & & & \\ & & F_2 & & \\ & & & \ddots & \\ & & & & F_2 \\ & & & & & F_4 \end{pmatrix}, \quad
N_N := \begin{pmatrix} & a & b & & & & & \\ d_2 & & & & & & & \\ & & a & b & & & & \\ & b & c & & & & & \\ & & & & \ddots & \ddots & & \\ & & & b & c & & & \\ & & & & \ddots & \ddots & a & b \\ & & & & & & & d_1 \\ & & & & & & b & c \end{pmatrix} \qquad (7)
$$

with blocks

$$F_1 := p + \frac{L}{\gamma_2(L-\gamma_2)}, \quad F_2 := \begin{pmatrix} p + \frac{\gamma_2}{\gamma_1(\gamma_2-\gamma_1)} & -\frac{1}{\gamma_2-\gamma_1} \\ -\frac{1}{\gamma_2-\gamma_1} & p + \frac{L-\gamma_1}{(L-\gamma_2)(\gamma_2-\gamma_1)} \end{pmatrix}, \quad F_4 := p + \frac{L}{\gamma_1(L-\gamma_1)},$$
$$a := p - \frac{\gamma_2}{\gamma_1(\gamma_2-\gamma_1)}, \; b := \frac{1}{\gamma_2-\gamma_1}, \; c := p - \frac{L-\gamma_1}{(L-\gamma_2)(\gamma_2-\gamma_1)}, \; d_j := p - \frac{L}{\gamma_j(L-\gamma_j)}.$$

The next theorem shows that the spectral radius of T_N^D is bounded strictly below 1 for every N if the Dirichlet b.c. are imposed on each fracture. Thus, the number of iterations to reach a given tolerance is independent of N, and OSM is weakly scalable.

Theorem 1 *Let $\gamma_1 + \gamma_2 = L$ and $s_j^{+,-} = p$, $\forall j$. Then, OSM is weakly scalable for the solution of problem* (1) *with Dirichlet b.c. on each F_i, in the sense that $\rho(T_N^D) \leq C < 1$, independently of N for every $p > 0$.*

Proof Notice that $\rho(T_N^D) = \rho(M_N^{-1}N_N) = \rho(N_N M_N^{-1}) \leq \|N_N M_N^{-1}\|_\infty$. Direct calculations show that

$$\|N_N M_N^{-1}\|_\infty = \max\left\{ \left| \frac{p\gamma_2(L-\gamma_2) - L}{p\gamma_2(L-\gamma_2) + L} \right|, \frac{2p(L-\gamma_2)^2 + \left| L + (L-2\gamma_2)(L-\gamma_2)^2 p^2 \right|}{(p(L-\gamma_2)+1)(p(L-\gamma_2)(2\gamma_2 - L) + L)} \right\}.$$

The first term is clearly less than 1 for every $p > 0$. For the second term, we distinguish two cases: if $L + (L-2\gamma_2)(L-\gamma_2)^2 p^2 < 0$, then it simplifies to $\left| \frac{-1+(L-\gamma_2)p}{1+(L-\gamma_2)p} \right|$ which strictly less than 1. Similarly, if $L + (L-2\gamma_2)(L-\gamma_2)^2 p^2 \geq 0$, then $\frac{2p(L-\gamma_2)^2 + \left| L+(L-2\gamma_2)(L-\gamma_2)^2 p^2 \right|}{(p(L-\gamma_2)+1)(p(L-\gamma_2)(2\gamma_2-L)+L)} = \left| \frac{p(L-\gamma_2)(2\gamma_2-L)-L}{p(L-\gamma_2)(2\gamma_2-L)+L} \right| < 1$ being $2\gamma_2 > L$. Thus, $\exists C < 1$ independent on N such that $\|N_N M_N^{-1}\|_\infty < C$ for every $p > 0$. □

The hypothesis $\gamma_1 + \gamma_2 = L$ is used to simplify the otherwise cumbersome calculations, but it has not been observed in numerical experiments.

We emphasize that OSMs are not scalable for one-dimensional chains of fixed size-subdomains [3]. In our setting, the scalability is due to the geometrical configuration typical for DFNs, which permits to impose Dirichlet b.c. on each fracture, being the transmission conditions imposed in the interior. Thus, we observe error contraction before information is propagated through the iterations across the subdomains (see [3, Section 3]). With a similar argument, we expect OSM not to be scalable if Neumann b.c. are applied on each fracture, as the errors in the middle fractures would require about $N/2$ to start contracting. To verify this, we can perform the same analysis by replacing (6) with appropriate subdomains solutions. We then obtain the recurrence relation $\mathbf{e}^n = T_N^N \mathbf{e}^{n-1} = \widetilde{M}_N^{-1} \widetilde{N}_N \mathbf{e}^{n-1}$, where $\widetilde{M}_N$ $\widetilde{N}_N$ have the same structure of (7), but with blocks

$$\widetilde{F}_1 := p + \frac{1}{\gamma_2}, \quad \widetilde{F}_4 := p + \frac{1}{L-\gamma_1}, \quad \widetilde{F}_2 := \begin{pmatrix} p + \frac{1}{\gamma_2-\gamma_1} & -\frac{1}{\gamma_2-\gamma_1} \\ -\frac{1}{\gamma_2-\gamma_1} & p + \frac{1}{\gamma_2-\gamma_1} \end{pmatrix},$$
$$\widetilde{a} := p - \frac{1}{\gamma_2-\gamma_1}, \quad \widetilde{b} := \frac{1}{\gamma_2-\gamma_1}, \quad \widetilde{c} := \widetilde{a}, \quad \widetilde{d}_j := p - \frac{1}{(L-\gamma_j)}.$$

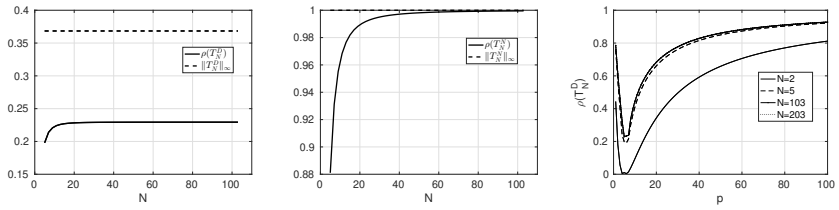


Fig. 2 Left and center panel: spectral radii of T_N^D and T_N^N as the number of fractures increases. Right panel: spectral radius of T_N^D as p varies. Parameters: $L = 1$, $\gamma_1 = 0.2$, $\gamma_2 = 0.6$, $\nu = 1$.

The first two panels of Fig. 2 show the dependence of the spectral radii of T_N^D and T_N^N as N increases. While $\rho(T_N^D)$ remains bounded below one, $\rho(T_N^N)$ tends rapidly to one as N grows, thus OSMs are not weakly scalable if the Neumann b.c. are used.

We remark that in applications it is quite common to impose homogeneous Neumann b.c. in internal fractures because at the tip of the fracture the flow exchange with the surrounding matrix is negligible. In such cases, the analysis suggests two possible heuristics to improve the convergence of DD solvers. The first one is to stress the importance of an efficient partition of the fractures into subdomains (each subdomain generally contains more than one fracture). Such partition should minimize the maximum, over floating subdomains, of the distance of each subdomain from the Dirichlet boundary Γ_d (see [8] for numerical experiments). Recall that a subdomain Ω_j is called "floating subdomain" if $\partial\Omega_j \cap \Gamma_D = \emptyset$. The second heuristic is to replace the Neumann b.c. with Robin ones (which would also model the realistic case of a flux across ∂F_j). Ref [5] suggest that Robin b.c. would permit to recover scalability of OSMs for DFN as in the Dirichlet case.

Notice that the rate of convergence of OSMs, which may be independent of N (see discussion above), still depends on the transmission conditions, hence it is important to have good estimates of the parameters $s_j^{+,-}$. To estimate them, we consider two fractures F_1 and F_2, which are coupled across a single trace. The general solutions are given by

$$e_1^n = \frac{\hat{e}_1^n \tau_1}{\gamma_2}\chi_{[0,\gamma_2]} + \frac{\hat{e}_1^n (L-\tau_1)}{L-\gamma_2}\chi_{[\gamma_2,L]}, \quad e_2^n = \frac{\hat{e}_2^n \tau_1}{\gamma_1}\chi_{[0,\gamma_1]} + \frac{\hat{e}_2^n (L-\tau_1)}{L-\gamma_1}\chi_{[\gamma_1,L]},$$

where the unknowns are two coefficients $\hat{e}_1^n$ and $\hat{e}_2^n$. Inserting these solutions in the transmission conditions we obtain the scalar recurrence relation for $j = 1, 2$,

$$\hat{e}_j^n = \rho_{1D}(s_1^-, s_1^+, \nu_1, \nu_2)\hat{e}_j^{n-2},\ \rho_{1D}(s_1^-, s_1^+, \nu_1, \nu_2) = \frac{\left(\frac{\nu_2 L}{\gamma_1(L-\gamma_1)} - s_1^-\right)\left(\frac{\nu_1 L}{\gamma_2(L-\gamma_2)} - s_1^+\right)}{\left(\frac{\nu_1 L}{\gamma_2(L-\gamma_2)} + s_1^-\right)\left(\frac{\nu_2 L}{\gamma_1(L-\gamma_1)} + s_1^+\right)}.$$

If we chose $s_1^- = s_1^{-,\mathrm{opt}} := \frac{\nu_2 L}{\gamma_1(L-\gamma_1)}$ and $s_1^+ = s_1^{+,\mathrm{opt}} := \frac{\nu_1 L}{\gamma_2(L-\gamma_2)}$, we would have $\rho(s_1^-, s_1^+, \nu_1, \nu_2) = 0$, that is, OSM is nilpotent. The right panel of Fig. 2 verifies that two fracture analysis provides very good estimates for the optimal Robin parameters in the many-fractures case.

3 Scalability analysis for two-dimensional DFNs

In this section we consider two dimensional extension of Fig. 2. Each fracture F_j is a two dimensional polygon, see Fig. 3, and the traces, denoted by S_j, are straight segments crossing the whole fracture. On each fracture, the local reference system has coordinates $\{\tau_1, \tau_2\}$. Due to the geometrical configuration, the error can be

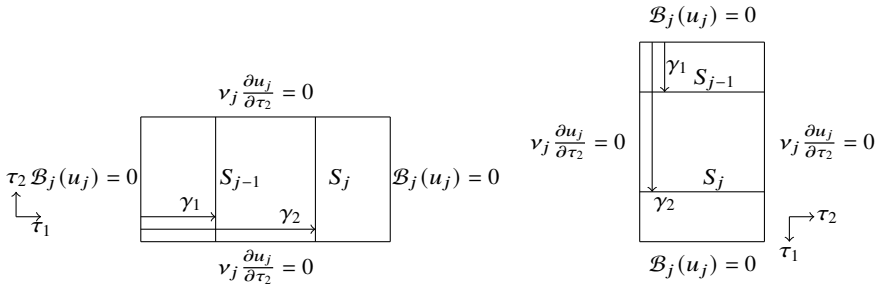


Fig. 3 Geometry of a two dimensional fracture.

expanded in Fourier series in each fracture, i.e. $e_j = \sum_{k=0}^{\infty} \tilde{e}_j(\tau_1, k) \cos(\frac{k\pi}{L}\tau_2)$. The Fourier coefficients $\tilde{e}_j(\tau_1, k)$ are obtained imposing the b.c. and the transmission conditions. The long expressions are omitted due to space limitation (see for complete expressions [8]). We only report the expressions for the first subdomain

$$\tilde{e}_1^n(\tau_1, k) = \hat{e}_1^n(k) \frac{\sinh(\frac{k\pi}{L}\tau_1)}{\sinh(\frac{k\pi}{L}\gamma_2)} \chi_{[0,\gamma_2]} + \hat{e}_1^n(k) \frac{\sinh(\frac{k\pi}{L}(L-\tau_1))}{\sinh(\frac{k\pi}{L}(L-\gamma_2))} \chi_{[\gamma_2,L]}, \ k > 0,$$
$$\tilde{e}_1^n(\tau_1, 0) = \frac{\hat{e}_1^n(0)\tau_1}{\gamma_2} \chi_{[0,\gamma_2]} + \frac{\hat{e}_1^n(0)(L-\tau_1)}{L-\gamma_2} \chi_{[\gamma_2,L]}, \ k = 0.$$

The unknowns $\hat{e}_j^{i,n}(k)$ are the values attained by the k-th mode of the Fourier expansions at each trace. In numerical computations, $k \in [k_{\min}, k_{\max}]$ for the Dirichlet b.c., while $k \in [0, k_{\max}]$ for the Neumann b.c., $k_{\max} = \frac{\pi}{h}$ being the maximum frequency supported by the numerical grid and $k_{\min} = \frac{\pi}{L}$. Similarly to the 1D case, one can obtain recurrence relations which link the Fourier coefficients of one fracture at iteration n as functions of the Fourier coefficients of the neighbouring fractures at iteration $n-1$. In particular for $k = 0$, $\mathbf{e}_0^n := \left(\hat{e}_1^n(0), \hat{e}_2^{1,n}(0), \hat{e}_2^{2,n}(0), \ldots, \hat{e}_N^n(0)\right)^\top$ satisfies $\mathbf{e}_0^n = T_N^D \mathbf{e}_0^{n-1}$, where T_N^D is the matrix of the 1D system with the Dirichlet b.c.. For $k > 0$, we obtain instead $\mathbf{e}_k^n = T_N^{2D}(k)\mathbf{e}_k^{n-1}$, where $T_N^{2D} = M_{2D}^{-1} N_{2D}$ has the same block structure of the 1D case but with blocks defined as

$$F_2 := \begin{pmatrix} p + \coth(\frac{k\pi}{L}\gamma_1) + \coth(\frac{k\pi}{L}(\gamma_2 - \gamma_1)) & -\frac{1}{\coth(\frac{k\pi}{L}(\gamma_2-\gamma_1))} \\ -\frac{1}{\sinh(\frac{k\pi}{L}(\gamma_2-\gamma_1))} & p + \coth(\frac{k\pi}{L}(L-\gamma_2)) + \coth(\frac{k\pi}{L}(\gamma_2-\gamma_1)) \end{pmatrix},$$

$F_1 := p + \coth(\frac{k\pi}{L}\gamma_2) + \coth(\frac{k\pi}{L}(L-\gamma_2))$ and $F_4 := p + \coth(\frac{k\pi}{L}(L-\gamma_1)) + \coth(\frac{k\pi}{L}(\gamma_1))$. On the other hand, the coefficients of N_{2D} are

$$a := p - \coth\left(\frac{k\pi}{L}\gamma_1\right) - \coth\left(\frac{k\pi}{L}(\gamma_2-\gamma_1)\right), \quad b := \frac{1}{\sinh(\frac{k\pi}{L}(\gamma_2-\gamma_1))},$$
$$c := p - \coth\left(\frac{k\pi}{L}(L-\gamma_2)\right) - \coth\left(\frac{k\pi}{L}(\gamma_2-\gamma_1)\right),$$
$$d_j := p - \coth\left(\frac{k\pi}{L}(L-\gamma_j)\right) - \coth\left(\frac{k\pi}{L}\gamma_j\right).$$

Fig 4 shows numerically that OSM is scalable also for a 2D DFN with the Dirichlet b.c. Observing that the frequency $k = 0$ behaves according to the 1D analysis, we expect OSM with the Neumann b.c. on each fracture except on the first and last ones not to be weakly scalable. Repeating the calculations one finds an iteration matrix $\widetilde{T}_N^{2D}$ and Fig. 4 confirms this conclusion.

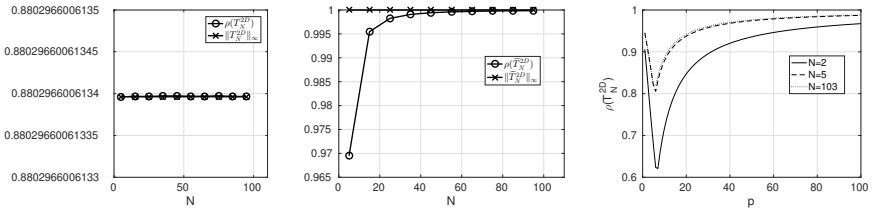


Fig. 4 Left and center panel: spectral radii of $\max_{k\in[k_{\min},k_{\max}]} T_N^{2D}(k)$ and $\max_{k\in[0,k_{\max}]} \widetilde{T}_N^{2D}(k)$ as N grows. Parameters: $L = 1$, $\gamma_1 = 0.2$, $\gamma_2 = 0.6$ and $p = 20$. Right panel: $\max_{k\in[k_{\min},k_{\max}]} T_N^{2D}(k)$ as p varies.

We now derive the optimized parameters by analyzing the coupling of two fractures. Inserting the Fourier expansions into the transmission conditions and defining

$$f_j(k) := \frac{\nu_{3-j}k\pi}{L}\left(\coth\left(\frac{k\pi}{L}\gamma_j\right) + \coth\left(\frac{k\pi}{L}(L-\gamma_j)\right)\right), \quad j=1,2,$$

we obtain $\hat{e}_j^n(k) = \rho(k, s_1^-, s_1^+)\hat{e}_j^{n-2}(k)$, for $k > 0, j = 1, 2$, where $\rho(k, s_1^-, s_1^+) := \frac{f_1(k)-s_1^-}{f_2(k)+s_1^-}\frac{f_2(k)-s_1^+}{f_1(k)+s_1^+}$. On the other hand, for the constant mode $k = 0$ we recover the 1D result: $\hat{e}_j^n(0) = \rho_{1D}(s_1^-, s_1^+)\hat{e}_j^{n-2}(0)$. To derive optimized parameters, we set $s_1^- = f_1(p)$, $s_1^+ = f_2(p)$ for some $p \in \mathbb{R}^+$, and we study

$$\min_{p\in\mathbb{R}^+} \max\left\{\rho_{1D}(p), \max_{k\in[\frac{\pi}{L},k_{\max}]} \rho(k,p)\right\}. \tag{8}$$

Despite $\rho(k,p)$ is not defined at $k = 0$ since $\coth(\cdot)$ has a singularity, we observe that $\lim_{k\to 0}\rho(k,p) = \rho_{1D}(p)$. Thus, we introduce the function $\widetilde{\rho}(k,p) = \rho(k,p)$

for $k > 0$ and $\widetilde{\rho}(0, p) = \rho_{1D}(p)$, and further simplify the min-max problem to

$$\min_{p \in \mathbb{R}^+} \max_{k \in [0, k_{\max}]} \widetilde{\rho}(k, p). \tag{9}$$

The next theorem can be proved using the same steps of [6, Theorem 2.3]. Fig. 4 confirms that effectiveness of the analysis even in the many-fractures case.

Theorem 2 *The solution of the min-max problem* (9) *is given by the unique* $p^\star$ *which satisfies* $\widetilde{\rho}(0, p^\star) = \widetilde{\rho}(k_{\max}, p^\star)$.

Future works will focus on testing the results of the analysis presented on more realistic DFN configurations.

References

1. Berrone, S., Pieraccini, S., and Scialò, S. A PDE-constrained optimization formulation for discrete fracture network flows. *SIAM Journal on Scientific Computing* **35**(2), B487–B510 (2013).
2. Cancès, E., Maday, Y., and Stamm, B. Domain decomposition for implicit solvation models. *The Journal of chemical physics* **139**(5), 054111 (2013).
3. Chaouqui, F., Ciaramella, G., Gander, M. J., and Vanzan, T. On the scalability of classical one-level domain-decomposition methods. *Vietnam Journal of Mathematics* **46**(4), 1053–1088 (2018).
4. Ciaramella, G., Hassan, M., and Stamm, B. On the scalability of the Schwarz method. *The SMAI journal of computational mathematics* **6**, 33–68 (2020).
5. Ciaramella, G. and Mechelli, L. On the effect of boundary conditions on the scalability of Schwarz methods. *arXiv preprint arXiv:2103.14848* (2021).
6. Gander, M. J. and Vanzan, T. Heterogeneous optimized Schwarz methods for second order elliptic pdes. *SIAM Journal on Scientific Computing* **41**(4), A2329–A2354 (2019).
7. Pichot, G., Poirriez, B., Erhel, J., and Dreuzy, J.-R. d. A mortar BDD method for solving flow in stochastic discrete fracture networks. In: *Domain decomposition methods in science and engineering XXI*, 99–112. Springer (2014).
8. Vanzan, T. *Domain decomposition methods for multiphysics problems*. Ph.D. thesis, Université de Genève (2020).

How Does the Partition of Unity Influence SORAS Preconditioner?

Marcella Bonazzoli, Xavier Claeys, Frédéric Nataf, and Pierre-Henri Tournier

1 Introduction

The Symmetrized Optimized Restricted Additive Schwarz (SORAS) preconditioner, first introduced in [8] for the Helmholtz equation and called OBDD-H, was later studied in [6] for generic symmetric positive definite problems and viewed as a symmetric variant of ORAS preconditioner. Its convergence was rigorously analyzed in [5] for the Helmholtz equation, and in [1] we generalized this theory to generic non self-adjoint or indefinite problems. Moreover, as an illustration of our theory, we proved new estimates for the specific case of the heterogeneous reaction-convection-diffusion equation. In the numerical experiments in [1], we noticed that the number of iterations for convergence of preconditioned GMRES appears not to vary significantly when increasing the overlap width. In the present paper, we show that actually this is due to the particular choice of the partition of unity for the preconditioner. The influence of five different kinds of partition of unity on SORAS solver and preconditioner for the Laplace equation has been briefly studied in the conclusion of [4], where the method is named ORASH. Here, for the reaction-convection-diffusion equation, we focus on two kinds of partitions of unity, and study the dependence on the overlap and on the number of subdomains.

Marcella Bonazzoli
Inria, UMA, ENSTA Paris, Institut Polytechnique de Paris, Palaiseau, France,
e-mail: marcella.bonazzoli@inria.fr

Xavier Claeys, Frédéric Nataf, Pierre-Henri Tournier
Sorbonne Université, CNRS, Université Paris Cité, LJLL, Paris, France,
e-mail: xavier.claeys@sorbonne-universite.fr, frederic.nataf@sorbonne-universite.fr,
pierre-henri.tournier@sorbonne-universite.fr

2 SORAS preconditioner and two kinds of partition of unity

Let A denote the $n \times n$ matrix, not necessarily positive definite nor self-adjoint, arising from the discretization of the problem to be solved, posed in an open domain $\Omega \subset \mathbb{R}^d$. Given a set of overlapping open subdomains Ω_j, $j = 1, \dots, N$, such that $\Omega = \bigcup_{j=1}^N \Omega_j$ and each $\overline{\Omega_j}$ is a union of elements of the mesh $\mathcal{T}^h$ of Ω, we consider the set $\mathcal{N}$ of the unknowns on the whole domain, so $\#\mathcal{N} = n$, and its decomposition $\mathcal{N} = \bigcup_{j=1}^N \mathcal{N}_j$ into the non-disjoint subsets corresponding to the different overlapping subdomains $\overline{\Omega}_j \cap \Omega$, with $\#\mathcal{N}_j = n_j$. Denote by δ the width of the overlap between subdomains. The following matrices are then the classical ingredients to define overlapping Schwarz domain decomposition preconditioners (see e.g. [2, §1.3]):

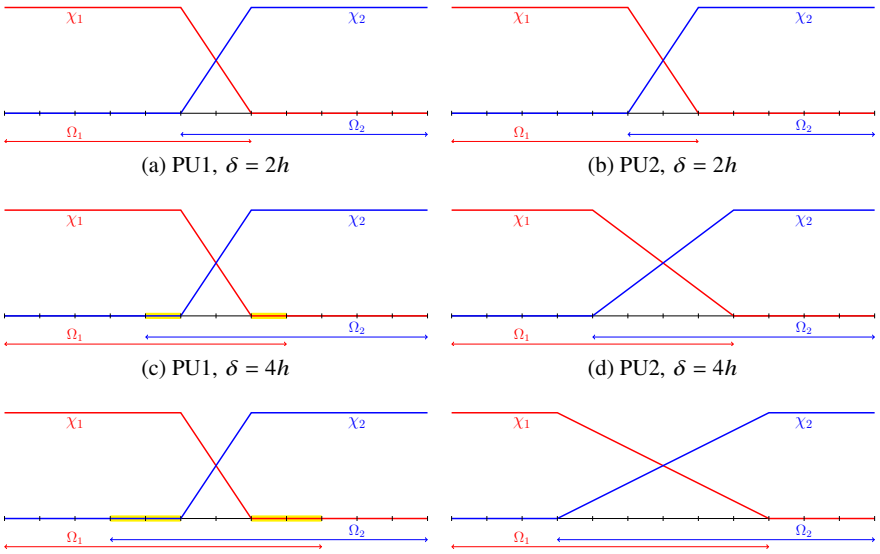
- restriction matrices R_j from Ω to $\overline{\Omega}_j \cap \Omega$, which are $n_j \times n$ Boolean matrices whose (i, i') entry equals 1 if the i-th unknown in $\mathcal{N}_j$ is the i'-th one in $\mathcal{N}$ and vanishes otherwise;
- extension by zero matrices R_j^T from $\overline{\Omega}_j \cap \Omega$ to Ω;
- partition of unity matrices D_j, which are $n_j \times n_j$ diagonal matrices with real non-negative entries such that $\sum_{j=1}^N R_j^T D_j R_j = I$ and which can be seen as matrices that properly weight the unknowns belonging to the overlap between subdomains;
- local matrices B_j, of size $n_j \times n_j$, which arise from the discretization of subproblems posed in $\overline{\Omega}_j \cap \Omega$, with for instance Robin-type or more general absorbing transmission conditions on the interfaces $\partial\Omega_j \setminus \partial\Omega$.

Then the one-level Symmetrized Optimized Restricted Additive Schwarz (SORAS) preconditioner is defined as

$$M^{-1} \coloneqq \sum_{j=1}^{N} R_j^T D_j B_j^{-1} D_j R_j. \tag{1}$$

Note that M^{-1} is not self-adjoint when B_j is not self-adjoint, even if we maintain the SORAS name, where S stands for 'Symmetrized'. In fact, this denomination was introduced in [6] for symmetric positive definite problems, since in that case SORAS preconditioner is a symmetric variant of ORAS preconditioner $\sum_{j=1}^N R_j^T D_j B_j^{-1} R_j$. Thus, the adjective 'Symmetrized' stands for the presence of the rightmost partition of unity D_j. We recall that 'Restricted' indicates the presence of the leftmost partition of unity D_j and that 'Optimized' refers to the choice of transmission conditions other than standard Dirichlet conditions in the local matrices B_j.

Here we focus on the influence exerted by the choice of partition of unity matrices D_j on the convergence of GMRES preconditioned by (1). Indeed, several definitions of the diagonal matrices D_j are possible to ensure property $\sum_{j=1}^N R_j^T D_j R_j = I$. In general, the diagonals of the D_j can be constructed by the interpolation of continuous partition of unity functions $\chi_j : \Omega \to [0, 1]$, $j = 1, \dots, N$: $\sum_{j=1}^N \chi_j = 1$ in $\overline{\Omega}$, and $\mathrm{supp}(\chi_j) \subset \Omega_j$, so in particular χ_j is zero on the subdomain interfaces $\partial\Omega_j \setminus \partial\Omega$.



(a) PU1, $\delta = 2h$ (b) PU2, $\delta = 2h$

(c) PU1, $\delta = 4h$ (d) PU2, $\delta = 4h$

(e) PU1, $\delta = 6h$ (f) PU2, $\delta = 6h$

Fig. 1 Illustration in a one-dimensional two-subdomain case of the two kinds of partition of unity functions $\chi_j : \Omega \to [0, 1]$ (PU1 on the left and PU2 on the right), with increasing width of the overlap δ from top to bottom.

In addition, in the case of ORAS fixed-point iterative solver, also the first derivatives of χ_j are required to be equal to zero on $\partial\Omega_j \setminus \partial\Omega$, because this property ensures that the continuous version of ORAS solver is equivalent to Lions' algorithm, see e.g. [2, §2.3.2] for a particular model problem. An instructive calculation for a simple one- (and two-) dimensional problem, which shows an analogous equivalence property for RAS solver, is given in [3]; a more general equivalence result for ORAS solver is proved in [10, Theorem 3.4]. This first choice of Partition of Unity (PU1), where the gradient of χ_j is zero on the subdomain interfaces $\partial\Omega_j \setminus \partial\Omega$, is illustrated in a one-dimensional two-subdomain case in Figure 1, left, and starting from an overlap $\delta = 4h$. Note that PU1 in Figure 1 is actually different from the original RAS/ORAS partition of unity, which is defined for any overlap size δ multiple of h, but essentially just at the discrete level, and takes only the values 0 or 1; in the original RAS/ORAS articles, the D_j are indeed hidden inside the definition of special extension matrices $\widetilde{R}_j^T$ related to an auxiliary non-overlapping partition of the domain (see e.g. [3, 10] and references therein). However, since the PU1 functions χ_j in Figure 1 are symmetrical to each other, defining the D_j by interpolation of the χ_j is more practical for a parallel implementation.

A second kind of Partition of Unity (PU2) is illustrated in Figure 1, right, where the χ_j functions are different from zero in the interior of the whole overlapping region. This choice is motivated by the fact that using PU1 for SORAS preconditioner can hinder the communication of information between subdomains since in (1) the matrix D_j is also applied before B_j^{-1}, that is before the local problem solve. Indeed,

the numerical experiments performed in [1], where PU1 was used, show that the number of iterations for convergence of preconditioned GMRES does not vary significantly when increasing the overlap size (see also Tables 1,2,3 in Section 4).

3 Definition of the model problem

As in the second part of [1], we consider the heterogeneous reaction-convection-diffusion problem in conservative form:

$$\begin{cases} c_0 u + \operatorname{div}(\mathbf{a}u) - \operatorname{div}(\nu \nabla u) = f & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma, \end{cases} \tag{2}$$

where $\Omega \subset \mathbb{R}^d$ is an open bounded polyhedral domain, $\Gamma = \partial\Omega$, $\mathbf{n}$ is the outward-pointing unit normal vector to Γ, $c_0 \in \mathrm{L}^\infty(\Omega)$, $\mathbf{a} \in \mathrm{L}^\infty(\Omega)^d$, $\operatorname{div}\mathbf{a} \in \mathrm{L}^\infty(\Omega)$, $\nu \in \mathrm{L}^\infty(\Omega)$, $f \in \mathrm{L}^2(\Omega)$ and all quantities are real-valued. We denote $\tilde{c} \coloneqq c_0 + \operatorname{div}\mathbf{a}/2$, and suppose that there exist $\tilde{c}_- > 0$, $\tilde{c}_+ > 0$ such that

$$\tilde{c}_- \le \tilde{c}(\mathbf{x}) \le \tilde{c}_+ \text{ a.e. in } \Omega, \tag{3}$$

and that there exist $\nu_- > 0$, $\nu_+ > 0$ such that $\nu_- \le \nu(\mathbf{x}) \le \nu_+$ a.e. in Ω. The variational formulation of problem (2) is (see e.g. [1, §4]): find $u \in \mathrm{H}^1_0(\Omega)$ such that

$$a(u, v) = F(v), \quad \text{for all } v \in \mathrm{H}^1_0(\Omega), \tag{4}$$

$$a(u, v) \coloneqq \int_\Omega \Big(\tilde{c} u v + \frac{1}{2}\mathbf{a} \cdot \nabla u \, v - \frac{1}{2} u \, \mathbf{a} \cdot \nabla v + \nu \nabla u \cdot \nabla v \Big), \quad F(v) \coloneqq \int_\Omega f v.$$

On each subdomain we consider the local problem with bilinear form

$$a_j(u, v) \coloneqq \int_{\Omega_j} \Big(\tilde{c} u v + \frac{1}{2}\mathbf{a} \cdot \nabla u \, v - \frac{1}{2} u \, \mathbf{a} \cdot \nabla v + \nu \nabla u \cdot \nabla v \Big) + \int_{\partial\Omega_j \setminus \Gamma} \alpha u v,$$

where we impose an absorbing transmission condition on the subdomain interface $\partial\Omega_j \setminus \partial\Omega$ given by $\alpha(\mathbf{x}) = \sqrt{(\mathbf{a} \cdot \mathbf{n})^2 + 4 c_0 \nu}/2$ (see e.g. [7]).

4 Numerical experiments

We simulate problem (4) with Ω a rectangle $[0, N \cdot 0.2] \times [0, 0.2]$, where N is the number of subdomains. In Tables 1,2,3 we take $N = 5$ and

$$f = 100 \exp\{-10((x - 0.5)^2 + (y - 0.1)^2)\}.$$

In Table 4, we test weak scaling by varying N, with

$$f = 100 \exp\{-10((x-0.1)^2 + (y-0.1)^2)\}.$$

The problem is discretized by piece-wise linear Lagrange finite elements on a uniform triangular mesh with 60 nodes on the vertical side of the rectangle and $N \cdot 60$ nodes on the horizontal one, resulting in 18361 degrees of freedom for $N = 5$, and 7381,

Table 1 Iteration numbers for SORAS preconditioner ($N = 5$).

	#PU1(PU2)			
$\mathbf{a} = 2\pi[-(y-0.1), (x-0.5)]^T$	$\delta = 2h$	$\delta = 4h$	$\delta = 6h$	$\delta = 8h$
$c_0 = 1,\ \nu = 1$	21(21)	20(17)	20(15)	19(14)
$c_0 = 1,\ \nu = 0.001$	14(14)	13(11)	12(11)	12(10)
$c_0 = 0.001,\ \nu = 1$	21(21)	20(18)	20(15)	19(14)
$c_0 = 0.001,\ \nu = 0.001$	15(15)	14(12)	13(11)	13(11)

Table 2 Repeat of Table 1 but with $\mathbf{a} = [-x, -y]^T$. In this case $\operatorname{div}\mathbf{a} = -2$ is negative and $\tilde{c} = c_0 - 1$ does not verify condition (3).

	#PU1(PU2)			
$\mathbf{a} = [-x, -y]^T$	$\delta = 2h$	$\delta = 4h$	$\delta = 6h$	$\delta = 8h$
$c_0 = 1,\ \nu = 1$	21(21)	21(19)	20(17)	20(15)
$c_0 = 1,\ \nu = 0.001$	16(16)	16(14)	16(13)	16(13)
$c_0 = 0.001,\ \nu = 1$	22(22)	22(19)	22(17)	21(16)
$c_0 = 0.001,\ \nu = 0.001$	17(17)	16(15)	16(14)	16(13)

Table 3 Repeat of Table 1 but with $\mathbf{a} = [1, 0]^T$ and with Streamline Upwind Petrov-Galerkin stabilization for the Galerkin approximation.

	#PU1(PU2)			
$\mathbf{a} = [1, 0]^T$	$\delta = 2h$	$\delta = 4h$	$\delta = 6h$	$\delta = 8h$
$c_0 = 1,\ \nu = 1$	20(20)	20(18)	20(16)	20(15)
$c_0 = 1,\ \nu = 0.001$	11(11)	11(12)	11(12)	11(12)
$c_0 = 0.001,\ \nu = 1$	20(20)	20(18)	20(16)	20(15)
$c_0 = 0.001,\ \nu = 0.001$	12(12)	12(12)	12(13)	12(12)

Table 4 Iteration numbers in a weak scaling test ($\delta = 4h$).

	#PU1(PU2)					
$\mathbf{a} = [1, 0]^T$	$N = 2$	$N = 4$	$N = 8$	$N = 16$	$N = 32$	$N = 64$
$c_0 = 1,\ \nu = 1$	18(15)	23(20)	28(24)	35(28)	36(29)	36(29)
$c_0 = 1,\ \nu = 0.001$	8(8)	10(12)	16(16)	23(24)	37(37)	63(61)
$c_0 = 0.001,\ \nu = 1$	18(15)	23(20)	29(25)	35(29)	36(29)	36(29)
$c_0 = 0.001,\ \nu = 0.001$	8(8)	10(12)	16(17)	24(25)	40(40)	71(71)

14701, 29341, 58621, 117181, 234301 degrees of freedom for $N = 2, 4, 8, 16, 32, 64$ respectively. The domain is partitioned into N vertical strips, then each subdomain is augmented with mesh elements layers of size $\delta/2$ to obtain the overlapping decomposition: the total width of the overlap between two subdomains is then δ. In particular, for $\delta = 2h, 4h, 6h, 8h$ the ratio between the subdomain width ($60h$) and δ is equal to 30, 15, 10, 7.5. We use GMRES with right preconditioning, with a zero initial guess in Tables 1,2,3 and a random initial guess in Table 4. The stopping criterion is based on the relative residual, with a tolerance of 10^{-6}. To apply the preconditioner, the local problems in each subdomain are solved with the direct solver MUMPS[1]. All the computations are done in the `ffddm` framework [11] of FreeFEM[2].

We compare the number of iterations for convergence (denoted by # in the tables) using the two kinds of partition of unity: the results for PU1 were also included in [1] and the results for PU2 are reported inside brackets in Tables 1–4. In `ffddm` framework, the first partition of unity is selected by the flag `-raspart`, while the second partition of unity is the one used by default.

As in [1], we examine several configurations for the coefficients in (2). First, in Table 1 we consider a rotating convection field $\mathbf{a} = 2\pi[-(y - 0.1), (x - 0.5)]^T$ and small/large values for the reaction coefficient c_0 and the viscosity ν. We can see that a larger overlap helps the convergence of the preconditioner, especially with PU2, while with PU1 the number of iterations does not vary significantly. Moreover, with both kinds of partition of unity, the number of iterations appears not very sensitive to the reaction coefficient c_0, while it increases when the viscosity ν is larger.

Then, in Table 2 we take $\mathbf{a} = [-x, -y]^T$, which has negative divergence $\operatorname{div} \mathbf{a} = -2$, to test the robustness of the method when condition (3) on the positiveness of $\tilde{c}$ is violated: in this case, $\tilde{c} = c_0 - 1$, so $\tilde{c} = 0$, $\tilde{c} = -0.999$ for $c_0 = 1$, $c_0 = 0.001$ respectively. We can still observe a convergence behavior similar to the one of Table 1.

Finally, in Table 3 we consider a horizontal convection field $\mathbf{a} = [1, 0]^T$, which is normal to the interfaces between subdomains. Since in this case non-physical numerical instabilities appear in the solution, we stabilize the discrete variational formulation using the Streamline Upwind Petrov-Galerkin (SUPG) method (see for instance [9, §11.8.6]). In this configuration for the convection field, for low viscosity $\nu = 0.001$ the dependence of the iteration number on the overlap size δ appears to be not significant, even with PU2.

Again in this third configuration with $\mathbf{a} = [1, 0]^T$ and SUPG stabilization, we perform a weak scaling test by taking $\Omega = [0, N \cdot 0.2] \times [0, 0.2]$ for increasing number of subdomains N, and $\delta = 4h$. We can see that especially in the cases with low viscosity $\nu = 0.001$, convergence deteriorates with N, as expected since we are testing a one-level preconditioner.

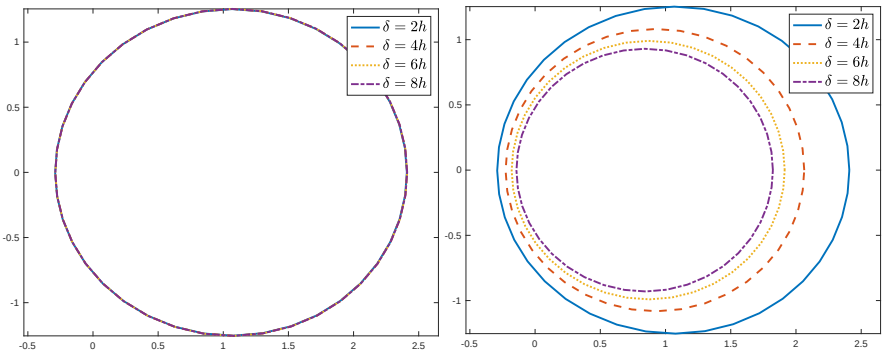
In summary, our numerical investigation shows that, for the considered SORAS preconditioner, PU2 generally improves the iteration counts obtained with PU1. Moreover, the first kind of partition of unity (PU1), which would be the natural choice for ORAS solver instead, yields for SORAS preconditioner iterations counts

[1] `http://mumps.enseeiht.fr/`

[2] `https://freefem.org/`

Table 5 Minimum and maximum eigenvalues of the preconditioned operator.

	PU1(PU2)			
$\mathbf{a} = [0,0]^T$	$\delta = 2h$	$\delta = 4h$	$\delta = 6h$	$\delta = 8h$
$\lambda_{\min}$	0.50 (0.50)	0.50 (0.50)	0.50 (0.50)	0.50 (0.50)
$\lambda_{\max}$	11.25 (11.25)	10.61 (5.98)	10.07 (4.01)	9.60 (3.02)



(a) PU1 (b) PU2

Fig. 2 Numerical range of the preconditioned operator ($\mathbf{a} = [-x, -y]^T$).

that do not vary significantly when increasing the overlap width, whereas using the second kind of partition of unity (PU2) a larger overlap gives faster convergence.

To conclude, we wish to provide a deeper explanation of the observed effects. First, we examine the symmetric positive definite case, with $\mathbf{a} = [0,0]^T$, $c_0 = 1$, $\nu = 1$ (so $\tilde{c} = c_0 > 0$), and report in Table 5 the largest and smallest eigenvalues of the preconditioned operator. We take $N = 2$ and 40 nodes on the vertical side of the rectangle, $2 \cdot 40$ nodes on the horizontal one. Note that SORAS preconditioner for generic symmetric positive definite problems was analyzed in [6], but no explicit discussion about the influence of the partition of unity was included there. On the one hand, the largest eigenvalue of the preconditioned operator is controlled by the modes of the local generalized eigenvalue problems defined in [6, Definition 3.1], where the partition of unity matrices appear in the local operator on the left-hand side: Table 5 shows that indeed $\lambda_{\max}$ is smaller for PU2, which is less steep than PU1, especially when increasing the overlap width δ (see Fig. 1). Moreover, with PU1, the dependence of $\lambda_{\max}$ on δ is much less significant than with PU2. On the other hand, the smallest eigenvalue of the preconditioned operator is controlled by the modes of the local generalized eigenvalue problems defined in [6, Definition 3.2], where the partition of unity is not involved: in Table 5 we can see that indeed $\lambda_{\min}$ is independent of the partition of unity.

For the non-symmetric case, with $\mathbf{a} = [-x, -y]^T$, $c_0 = 0.001$, $\nu = 0.001$ (so $\tilde{c} = c_0 - 1 < 0$), we plot in Fig. 2 the contour of the numerical range of the preconditioned operator for overlap widths that range from $\delta = 2h$ to $\delta = 8h$, for the two types of partition of unity. We can remark that for PU1 (Fig. 2, left) the numerical ranges practically coincide for the different overlap widths, whereas for PU2 (Fig. 2, right) the numerical range gets smaller for larger overlap width. This explains the more favorable convergence properties of preconditioned GMRES with PU2 when increasing the overlap width, and the much less significant influence of the overlap in the case of PU1.

References

1. Bonazzoli, M., Claeys, X., Nataf, F., and Tournier, P.-H. Analysis of the SORAS domain decomposition preconditioner for non-self-adjoint or indefinite problems. *Journal of Scientific Computing* **89**(1), 19 (2021).
2. Dolean, V., Jolivet, P., and Nataf, F. *An introduction to domain decomposition methods: algorithms, theory and parallel implementation*. SIAM, Philadelphia, PA (2015).
3. Efstathiou, E. and Gander, M. J. Why Restricted Additive Schwarz converges faster than Additive Schwarz. *BIT Numerical Mathematics* **43**(5), 945–959 (2003).
4. Gander, M. J. Does the partition of unity influence the convergence of Schwarz methods? In: *Domain Decomposition Methods in Science and Engineering XXV*, 3–15. Springer International Publishing, Cham (2020).
5. Graham, I. G., Spence, E. A., and Zou, J. Domain Decomposition with Local Impedance Conditions for the Helmholtz Equation with Absorption. *SIAM J. Numer. Anal.* **58**(5), 2515–2543 (2020).
6. Haferssas, R., Jolivet, P., and Nataf, F. A robust coarse space for optimized Schwarz methods: SORAS-GenEO-2. *C. R. Math. Acad. Sci. Paris* **353**(10), 959–963 (2015).
7. Japhet, C., Nataf, F., and Rogier, F. The optimized order 2 method: Application to convection–diffusion problems. *Future Generation Computer Systems* **18**(1), 17–30 (2001).
8. Kimn, J.-H. and Sarkis, M. Restricted overlapping balancing domain decomposition methods and restricted coarse problems for the Helmholtz problem. *Comput. Methods Appl. Mech. Engrg.* **196**(8), 1507–1514 (2007).
9. Quarteroni, A. M. *Numerical models for differential problems*, vol. 2. Springer (2009).
10. St-Cyr, A., Gander, M. J., and Thomas, S. J. Optimized Multiplicative, Additive, and Restricted Additive Schwarz preconditioning. *SIAM Journal on Scientific Computing* **29**(6), 2402–2425 (2007).
11. Tournier, P.-H. and Nataf, F. FFDDM: FreeFEM Domain Decomposition Methods. https://doc.freefem.org/documentation/ffddm/index.html (2019).

Convergence of the Micro-Macro Parareal Method for a Linear Scale-Separated Ornstein-Uhlenbeck SDE

Ignace Bossuyt, Stefan Vandewalle, and Giovanni Samaey

1 Model problem and motivation

In this work, we consider a two-dimensional slow-fast Ornstein-Uhlenbeck (OU) stochastic differential equation (SDE) [9], modelling the coupled evolution of a slowly evolving variable $x \in \mathbb{R}$ and a variable $y \in \mathbb{R}$ that quickly reaches its equilibrium distribution:

$$\begin{bmatrix} dx \\ dy \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \gamma/\epsilon & \zeta/\epsilon \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} dt + \sigma \begin{bmatrix} 1 & 0 \\ 0 & 1/\sqrt{\epsilon} \end{bmatrix} dW. \tag{1}$$

where $dW \in \mathbb{R}^2$ is a two-dimensional Brownian motion and $\epsilon \in \mathbb{R}$ is a (small) time scale separation parameter $\epsilon \ll 1$. The initial condition has a distribution with mean $\begin{bmatrix} m_{x,0} & m_{y,0} \end{bmatrix}$ and covariance matrix $\begin{bmatrix} \Sigma_{x,0} & \Sigma_{xy,0} \\ \Sigma_{xy,0} & \Sigma_{y,0} \end{bmatrix}$, and time $t \in [0, T]$.

Model problem (1) mimics the general situation where x is a low-dimensional quantity of interest whose evolution is influenced by a quickly evolving, high-dimensional variable y, all described by SDEs. The joint probability density of x and y obeys a Fokker-Planck equation (see, e.g. [3]). Instead of directly solving this partial differential equation using classical deterministic techniques, which suffer from the curse of dimensionality, the corresponding SDE can be solved using a Monte Carlo method. In this paper, our aim is to obtain insight in the convergence of a parallel-in-time (PinT) method applied to the low-dimensional linear OU model problem (1). In our method, the fine propagator of the SDE is based on a high-dimensional slow-fast microscopic model; the coarse propagator is based on a model-reduced version of the latter, that captures the low-dimensional, effective dynamics at the slow time scales. This problem allows for an analytic treatment, if

Ignace Bossuyt, Stefan Vandewalle, Giovanni Samaey
Department of Computer Science, KU Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium,
e-mail: ignace.bossuyt1@kuleuven.be, stefan.vandewalle@kuleuven.be,
giovanni.samaey@kuleuven.be

the quantities of interest are the mean and the (co)variance of x and y. We expect that this convergence analysis can be useful as a stepping stone for analysing PinT methods for higher-dimensional (nonlinear) SDEs.

1.1 Derivation of a reduced model

The averaging technique from [7, chapter 10, see, e.g., Remark 10.2] allows to define the reduced dynamics variable X, that approximates the slow variable x in (1). This technique exploits time-scale separation in order to integrate out the fast variable with respect to $\rho^\infty(y|x)$, the invariant distribution of the fast variable y conditioned on a fixed slow variable x.

The reduced model reads as follows (Λ_Σ and Σ_Σ are defined implicitly):

$$dX = A(X)dt + S(X)dW \tag{2}$$

with

$$A(X) = \int_{\mathcal{Y}} a(X,y)\rho^\infty(y|X)dy = \Lambda_\Sigma X \coloneqq \left(\alpha - \frac{\beta\gamma}{\zeta}\right) X$$
$$S(X)S(X)^T = \int_{\mathcal{Y}} s(X,y)s(X,y)^T\rho^\infty(y|X)dy = \Sigma_\Sigma \coloneqq \sigma,$$

where $\mathcal{Y}$ denotes the domain of y. It can be shown that for the OU system (1), the conditional distribution $\rho^\infty(y|x) = \mathcal{N}\left(\frac{\gamma x}{\zeta}, \frac{\sigma^2}{2\zeta}\right)$ (see [7, Example 6.19]).

The reduced model (2), while it is only an approximation to the slow dynamics, offers two computational advantages w.r.t. the full, scale-separated system (1): (i) it contains fewer degrees of freedom, and (ii) it is less stiff with a computational cost that is independent of ϵ. As ϵ approaches zero, the multiscale model (1) gets more stiff, while the (cheaper) reduced model becomes a more accurate approximation.

1.2 Moment system for the Ornstein-Uhlenbeck process

The evolution of mean and variance of a linear SDE can be described exactly using the moment models from [1]. Thus, for the linear Ornstein-Uhlenbeck SDE model problem, we can use these linear ODEs instead of using a Monte Carlo simulation.

Moments for reduced model. The evolution of the mean of X in (2) is given by

$$\frac{dm_X}{dt} = \left(\alpha - \frac{\beta\gamma}{\zeta}\right) m_X. \tag{3}$$

The evolution of the variance of the reduced system is given by the ODE

$$\frac{d\Sigma_X}{dt} = \Lambda_\Sigma\Sigma_X + \Sigma_\Sigma^2 = 2\left(\alpha - \frac{\beta\gamma}{\zeta}\right)\Sigma_X + \sigma^2. \tag{4}$$

Moments for multiscale model. The evolution of the mean of the multiscale SDE (1) is described by the following linear ODE:

$$\frac{d}{dt}\begin{bmatrix} m_x \\ m_y \end{bmatrix} = \begin{bmatrix} \alpha & \beta \\ \gamma/\epsilon & \zeta/\epsilon \end{bmatrix}\begin{bmatrix} m_x \\ m_y \end{bmatrix}. \tag{5}$$

The evolution of the covariance of (1) is given by the linear ODE $\dot{\Sigma} = B_\Sigma \Sigma + b_\Sigma$:

$$\frac{d}{dt}\begin{bmatrix} \Sigma_x \\ \Sigma_{xy} \\ \Sigma_y \end{bmatrix} = \left[\begin{array}{c|cc} 2\alpha & 2\beta & 0 \\ \hline \gamma/\epsilon & \alpha+\zeta/\epsilon & \beta \\ 0 & 2\gamma/\epsilon & 2\zeta/\epsilon \end{array}\right]\begin{bmatrix} \Sigma_x \\ \Sigma_{xy} \\ \Sigma_y \end{bmatrix} + \begin{bmatrix} \sigma^2 \\ 0 \\ \sigma^2/\epsilon \end{bmatrix} \tag{6}$$

where we define $\Sigma_q = \begin{bmatrix} \Sigma_{xy} & \Sigma_y \end{bmatrix}^T$, and where the blocks of B_Σ are named as $B_\Sigma = \left[\begin{array}{c|c} 2\alpha & p_\Sigma^T \\ \hline q_\Sigma/\epsilon & -A_\Sigma/\epsilon \end{array}\right]$, where $A_\Sigma = -\begin{bmatrix} \alpha+\zeta/\epsilon & \beta \\ 2\gamma/\epsilon & 2\zeta/\epsilon \end{bmatrix}$. To ensure stability of the fast dynamics, we assume that the parameters in (1) are chosen such that the real part of the eigenvalues of the matrix A_Σ are all positive $\mu_{\Sigma,i} \geq \mu_- > 0$. This condition is satisfied for instance for any $\alpha, \beta \in \mathbb{R}$ if ζ and γ are sufficiently small.

2 The Micro-Macro Parareal algorithm

The Micro-Macro Parareal (mM-Parareal) for scale-separated ODEs [5] and for SDEs [4], is a generalisation of the Parareal algorithm [6]. It combines two levels of description: (i) the micro variable u, with corresponding fine propagator $\mathcal{F}$, and (ii) the macro variable ρ, which is lower-dimensional, with coarse propagator C. These levels are related through coupling operators: the restriction operator $\mathcal{R}$ extracts macro information from a micro state, the lifting operator $\mathcal{L}$ produces a micro state that is consistent with a given macro state, and finally the matching operator $\mathcal{M}$ produces a micro state that is consistent with a given macro state, based on prior information of the micro state. Examples of these operators are given in the sequel. The mM-Parareal algorithm iterate at iteration k and time step n is given next. For $k = 0$ (initialization), we have

$$\rho_{n+1}^0 = C(\rho_n^0) \qquad u_{n+1}^0 = \mathcal{L}(\rho_{n+1}^0), \tag{7}$$

and for $k \geq 1$,

$$\begin{aligned} \rho_{n+1}^{k+1} &= C(\rho_n^{k+1}) + \mathcal{R}(\mathcal{F}(u_n^k)) - C(\rho_n^k) \\ u_{n+1}^{k+1} &= \mathcal{M}(\rho_{n+1}^{k+1}, \mathcal{F}(u_n^k)). \end{aligned} \tag{8}$$

If the coupling operators are chosen such that $\mathcal{M}(\mathcal{R}u, u) = u$, then at each iteration it holds that $\rho_n^k = \mathcal{R}u_n^k$. Classical Parareal [6] corresponds to the case $\mathcal{R} = \mathcal{L} = \mathcal{M} = \mathcal{I}$.

Convergence of Micro-Macro Parareal for linear scale-separated ODEs. In [5], the convergence of mM-Parareal for a linear scale-separated ODE is studied.

We briefly review the main ingredients of the theory, because we will use them further on to study the convergence for our model problem (1).

The test system in [5], modelling the coupled evolution of a slow variable $r \in \mathbb{R}$ and a fast variable $v \in \mathbb{R}^p$, $p \geq 1$, has the following structure:

$$\begin{bmatrix} \dot{r} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} a & p^T \\ q/\epsilon & -A/\epsilon \end{bmatrix} \begin{bmatrix} r \\ v \end{bmatrix} \tag{9}$$

where $A \in \mathrm{I} \times \mathrm{I}$ has positive eigenvalues: the fast component v is dissipative. The model for the approximate slow variable U, and the parameter Λ, are defined as follows:

$$\dot{U} = \Lambda U = \left(a + p^T A^{-1} q\right) U, \tag{10}$$

with $U(0) = U_0 = r_0$. In [5, equations (2.8), (2.13) and (2.14)] the following properties of the multiscale system (9) and its reduced model (10) are proven (the subscript $\cdot_0$ denotes the initial condition):

$$\sup_{t \in [0,T]} |r(t) - U_0 \exp(\Lambda t)| \leq C\epsilon(|r_0| + \|v_0 - A^{-1} q r_0\|), \tag{11}$$

$$\sup_{t \in [0,T]} |r(t)| \leq C(|r_0| + \epsilon \|v_0\|), \tag{12}$$

$$\sup_{t \in [t_{\mathrm{BL}},T]} \|v(t)\| \leq C(|r_0| + \epsilon \|v_0\|), \tag{13}$$

where the constant C only depends only on A, p, q, a and T (see (9)).

Using the properties (11)–(13), in [5], the convergence of mM-Parareal for the linear test problem (9) with coarse model (10) is analysed, using the restriction operator $\mathcal{R}(\begin{bmatrix} r, v \end{bmatrix}^T) = r$ (with $\mathcal{R}^\perp(\begin{bmatrix} r, v \end{bmatrix}^T) = v$), the lifting operator $\mathcal{L}(U) = \begin{bmatrix} U, A^{-1} q U \end{bmatrix}^T$ and the matching operator $\mathcal{M}(U, u) = \begin{bmatrix} U, \mathcal{R}^\perp u \end{bmatrix}^T$. We now present two minor extensions to existing Micro-Macro Parareal convergence lemmas for later use.

Lemma 1 (Convergence of mM-Parareal for nonhomogenous linear ODEs) *The mM-Parareal solution of the system $\dot{u} = Au + b$ equals the mM-Parareal solution of the system $\dot{v} = Av$, with $v = u = A^{-1} b$, if $v(0)$ is chosen $v(0) = u(0) - A^{-1} b$, with A and b constant. Assume that the (numerical) fine propagator satisfies the following property when it is applied on a linear system: $\mathcal{F}(u) = (I + A_{\mathcal{F}})u + B_{\mathcal{F}}$ with $B_{\mathcal{F}} = 0$ for the homogeneous system. (This assumption is not restrictive, e.g., it is satisfied by any Runge Kutta method.) Futher assume that $\mathcal{M}(\rho, u) - \mathcal{M}(\sigma, v) = \mathcal{M}(\rho - \sigma, u - v)$ and that the coarse propagator is linear. Then, the mM-Parareal iterates satisfy*

$$u_n^k = v_n^k + A^{-1} B \tag{14}$$

The proof of Lemma 1 can be constructed by induction on n.

Lemma 2 (Convergence of mM-Parareal without lifting in the zeroth iteration for linear scale-separated ODEs)

Using trivial lifting, that is $\mathcal{L}(X) = \left[X, v_0\right]$*, and using mM-Parareal, defined in* (7)–(8)*, with the specific choice of operators* (9)–(10)*, let* $E^k_{\Sigma_X,n} = u^k_n - \mathcal{R}u_n$ *be the macro error and* $e^k_n = u^k_n - u_n$ *be the micro error. Then, there exists* $\epsilon_0 \in (0,1)$*, that only depends on* α*,* p*,* q*,* A *and* T*, such that, for all* $\epsilon < \epsilon_0$ *and all* $\Delta t > t^{BL}_\epsilon$*, there exists a constant* C_k*, independent of* ϵ*, such that for all* $k \geq 0$*:*

$$\sup_{0\leq n\leq N} |E^k_n| \leq C_k \epsilon^{1+\lfloor (k+1)/2 \rfloor}, \tag{15}$$

$$\sup_{0\leq n\leq N} \|e^k_n\| \leq C_k \epsilon^{\lceil (k+1)/2 \rceil}. \tag{16}$$

The proof of Lemma 2 closely follows [5, proof of Theorem 13].

3 Convergence of Micro-Macro Parareal for model problem: theoretical analysis

The model problem is the multiscale Ornstein-Uhlenbeck process (1). We define the micro variable, describing the first two moments of its solution as $u^k_n = \left[m_x\ m_y\ \Sigma_x\ \Sigma_{xy}\ \Sigma_y\right]$. The macro variable is defined as $\rho^k_n = \left[m_x\ \Sigma_x\right]$.

For the fine propagator $\mathcal{F}$, we use the SDE (1), which we model via its moment models (5) and (6). The coarse propagator $\mathcal{C}$ simulates the reduced system (2), or equivalently the scalar ODEs (6) and (4). The restriction operator is defined as $\mathcal{R}\left(\left[m_x\ m_y\ \Sigma_x\ \Sigma_y\ \Sigma_{xy}\right]\right) = \left[m_x\ \Sigma_x\right]$, the lifting operator as: $\mathcal{L}\left(\left[M_X\ S_X\right]^T\right) = \left[M_X\ m_{y,0}\ S_X\ \Sigma_{y,0}\ \Sigma_{xy,0}\right]^T$, and the matching operator as $\mathcal{M}\left(\left[M_X\ S_X\right]^T, \left[m_x\ m_q\ \Sigma_x\ \Sigma_y\ \Sigma_{xy}\right]^T\right) = \left[M_X\ m_y\ S_X\ \Sigma_y\ \Sigma_{xy}\right]^T$. The lifting operator thus initializes the moments of the fast variable to its initial value.

Convergence of first moment. The moment equations (5) and (3), describing the evolution of the first moment obey the structure of the multiscale system (9), and therefore we can, after using Lemma 1, apply Lemma 2.

Converence of covariance. The evolution of the multiscale covariance (6) does not satisfy the same property as the model in equation (9) because (i) the submatrix A_Σ contains the parameter ϵ, and (ii) the reduced model is not defined using (10). Next we will prove that, although the models (6) and (9) are different, they both satisfy some key theoretical properties that were used in [5].

Lemma 3 (An equivalent of (11) **for model** (6) **instead of model** (9)) *For system* (6) *and its reduced model* (4)*, it holds true that*

$$\sup_{t\in[0,T]} |\Sigma_x(t) - \Sigma_{x,0}\exp(\Lambda_\Sigma t)| \leq C\epsilon(|\Sigma_{x,0}| + \|\Sigma_{y,0} - A^{-1}_\Sigma q_\Sigma \Sigma_{x,0}\|). \tag{17}$$

Proof From (11), (6) and (19), we have

$$\sup_{t\in[0,T]} |\Sigma_x(t) - \Sigma_{x,0}\exp(\lambda_\Sigma t)| \leq C\epsilon(|\Sigma_{x,0}| + \|\Sigma_{z,0}\|). \tag{18}$$

If we define

$$\lambda_\Sigma = 2\alpha + p_\Sigma^T A_\Sigma^{-1} q_\Sigma, \tag{19}$$

we can interpret the averaged model (4) as a limit of the reduced model (10) for the system (6): $\Lambda_\Sigma = \lim_{\epsilon\to 0} \lambda_\Sigma = 2\alpha - 2\frac{\beta\gamma}{\zeta}$. Now we define $\Delta\Lambda_\Sigma = \Lambda_\Sigma - \lambda_\Sigma$ (see (4) and (19)) and we observe that $\Delta\Lambda_\Sigma = O(\epsilon)$. It then holds that $\exp(\Lambda_\Sigma t) = \exp((\lambda_\Sigma + \Delta\Lambda_\Sigma)t) = \exp(\lambda_\Sigma t)\,[1 + O(\epsilon)]$. From the triangle inequality and the inequality (11) we have that

$$\begin{aligned}\sup_{t\in[0,T]} |\Sigma_x(t) - \Sigma_{x,0}\exp(\Lambda_\Sigma t)| &\leq \sup_{t\in[0,T]} |\Sigma_x(t) - \Sigma_{x,0}\exp(\lambda_\Sigma t)(1 + O(\epsilon))| \\ &\leq C\epsilon(|\Sigma_{x,0}| + \|\Sigma_{z,0}\|) + |\Sigma_{x,0}||O(\epsilon)| \\ &\leq K\epsilon(|\Sigma_{x,0}| + \|\Sigma_{z,0}\|),\end{aligned} \tag{20}$$

where $K > C$. This proves equation (17). □

Lemma 4 (An equivalent of (12) **and** (13) **for model** (6) **instead of model** (9)) *Assuming that that the eigenvalues $\mu_{\Sigma,i}$ of the matrix A_Σ (see* (6)*) are all positive, the properties in equation* (12) *and* (13) *hold true for the system* (6)*:*

$$\begin{aligned}\sup_{t\in[0,T]} |\Sigma_x(t)| &\leq C(|\Sigma_{x,0}| + \epsilon\|\Sigma_{q,0}\|), \\ \sup_{t\in[t_{BL},T]} \|\Sigma_q(t)\| &\leq C(|\Sigma_{x,0}| + \epsilon\|\Sigma_{q,0}\|).\end{aligned} \tag{21}$$

Proof The proof is similar to [5, Proof of Corollary 3]. In [5], the assumption that the eigenvalues of A_Σ are all positive is important. The structure of λ_Σ (or Λ_Σ) does not further influence the proof. □

The preceding lemmas allow us to formulate our main result.

Lemma 5 (Convergence of mM-Parareal for evolution of covariance) *Consider mM-Parareal, defined in* (7)–(8)*, with fine and coarse propagators the full system* (6) *and the reduced system* (4)*, respectively. Let $E^k_{\Sigma_X,n} = \rho^k_n - \mathcal{R}u_n$ be the macro error and $e^k_n = u^k_n - u_n$ be the micro error. Then there exists $\epsilon_0 \in (0,1)$, that only depends on α, p_Σ, q_Σ, A_Σ and T, such that, for all $\epsilon < \epsilon_0$ and all $\Delta t > t^{BL}_\epsilon$, there exists a constant C_k, independent of ϵ, such that for all $k \geq 0$:*

$$\sup_{0\leq n\leq N} |E^k_n| \leq C_k \epsilon^{1+\lfloor (k+1)/2\rfloor} \tag{22}$$

$$\sup_{0\leq n\leq N} \|e^k_n\| \leq C_k \epsilon^{\lceil (k+1)/2\rceil} \tag{23}$$

Proof Using Lemmas 1, 2, 3, and 4 the proof follows from [5, Proof of Theorem 2]. □

4 Numerical experiments

The test parameters for the numerical experiments are chosen to be:

$$\begin{bmatrix} \alpha & \beta \\ \gamma/\epsilon & \zeta/\epsilon \end{bmatrix} = \begin{bmatrix} -1. & -1. \\ 0.1/\epsilon & -1./\epsilon \end{bmatrix}, \qquad \sigma = 0.5 \tag{24}$$

The time interval is chosen as $[0, 10]$, the number of time intervals $N = 10$, and the initial value $\left[m_{x,0}\ m_{q,0}\ \Sigma_{x,0}\ \Sigma_{q,0}\ \Sigma_{xq,0}\right]^T = \left[100\ 100\ 100\ 0\ 0\right]^T$. In the experiments, which are shown in Figure 1, it is seen that the micro and macro errors in the mean follow the behaviour given by Lemma 2; those in the variance follow the behaviour as given by Lemma 5. Observe that mM-Parareal converges faster for computationally more expensive models (with small ϵ).

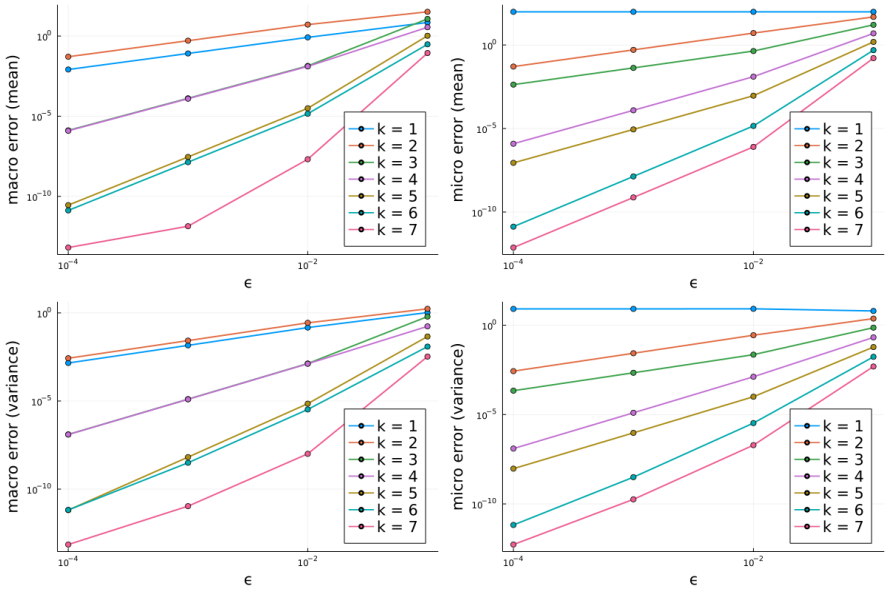


Fig. 1 Error as function of time-scale separation parameter ϵ. We used ∞-norm over time (only considering coarse discretisation points) and the 2-norm for the micro error. **Top left**: macro error on mean, **Top right**: micro error on mean, **Bottom left**: macro error on variance, **Bottom right**; micro error on variance. We used a numerical solver to discretise the moment equations (3)–(6) with a very stringent tolerance, so that the effect of numerical discretisation errors can be neglected.

5 Discussion and conclusion

Summary. We presented a convergence analysis of the Micro-Macro Parareal algorithm on scale-separated Ornstein-Uhlenbeck SDEs. We analysed its convergence behaviour w.r.t. the time scale separation parameter ϵ, using moment models. The convergence of the first moment is closely related to the analysis in [5]. For the covariance we presented some extensions to this theory.

Limitations. While the analysis using moment models quantifies the error on the mean and variance of the SDE solution, we cannot say anything about other quantities of interest, such as higher moments of the SDE solutions.

Also, by using the moment model (an ODE that we solved using very stringent tolerances), we exclusively looked at the model error, neglecting the discretisation errors and statistical errors (in e.g. Monte Carlo simulations) that arise in the discretisation of an SDE.

Open questions. It remains to be studied how the analysis generalises to higher dimensions, for instance when the slow variable is multi-dimensional. Also, an extension of the convergence analysis could cover nonlinear SDEs, or linear SDEs for which there is a coupling between mean and variance in the moment model ODEs. Another open problem is an analysis of convergence of the method w.r.t. the iteration number, in contrast to convergence w.r.t. the parameter ϵ. This would be more useful in practice.

Software. The code that is used for the numerical experiments, is available[1]. We used the Julia language [2] and the DifferentialEquations.jl package [8].

References

1. Arnold, L. *Stochastic differential equations: theory and applications*. Wiley, New York (1974).
2. Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B. Julia: A Fresh Approach to Numerical Computing. *SIAM Review* **59**(1), 65–98 (2017).
3. Gardiner, C. W. *Handbook of stochastic methods for physics, chemistry, and the natural sciences*. No. v. 13 in Springer series in synergetics. Springer-Verlag, Berlin ; New York (1983).
4. Legoll, F., Lelièvre, T., Myerscough, K., and Samaey, G. Parareal computation of stochastic differential equations with time-scale separation: a numerical convergence study. *Comput. Vis. Sci.* **23**(1-4), Paper No. 9, 18 (2020).
5. Legoll, F., Lelièvre, T., and Samaey, G. A micro-macro parareal algorithm: application to singularly perturbed ordinary differential equations. *SIAM J. Sci. Comput.* **35**(4), A1951–A1986 (2013).
6. Lions, J.-L., Maday, Y., and Turinici, G. Résolution d'EDP par un schéma en temps "pararéel". *C. R. Acad. Sci. Paris Sér. I Math.* **332**(7), 661–668 (2001).
7. Pavliotis, G. A. and Stuart, A. M. *Multiscale methods: Averaging and homogenization*, *Texts in Applied Mathematics*, vol. 53. Springer, New York (2008).
8. Rackauckas, C. and Nie, Q. DifferentialEquations.jl – A Performant and Feature-Rich Ecosystem for Solving Differential Equations in Julia. *Journal of Open Research Software* **5**(1), 15 (2017).
9. Uhlenbeck, G. E. and Ornstein, L. S. On the Theory of the Brownian Motion. *Physical Review* **36**(5), 823–841 (1930).

[1] https://gitlab.kuleuven.be/numa/public/mm-parareal-convergence-sde

A Trefftz-Like Coarse Space for the Two-Level Schwarz Method on Perforated Domains

Miranda Boutilier, Konstantin Brenner, and Victorita Dolean

1 Introduction and model problem

Numerical modeling of overland flows plays an increasingly important role in predicting, anticipating and controlling floods, helping to size and position protective systems including dams, dikes or rainwater drainage networks. One of the challenges of the numerical modeling of urban floods is that the small structural features (buildings, walls, etc.) may significantly affect the flow. Luckily, modern terrain survey techniques including photogrammetry and Laser Imaging, Detection, and Ranging (LIDAR) allow to acquire high-resolution topographic data for urban areas as well as for natural (highly vegetated) media. For example, the data set used in this article has been provided by Métropole Nice Côte d'Azur (MNCA) and allows for the infra-metric description of the urban geometries [3].

From the hydraulic perspective, these structural features can be assumed to be essentially impervious, and therefore represented as perforations (holes) in the model domain. Our long term modelling strategy is based on the Diffusive Wave equation [2]. However, understanding linear problems posed on perforated domains is a crucial preliminary step and the object of this contribution.

Let D be an open simply connected polygonal domain in $\mathbb{R}^2$, we denote by $\left(\Omega_{S,k}\right)_k$ a finite family of perforations in D such that each $\Omega_{S,k}$ is an open connected polygonal subdomain of D. The perforations are mutually disjoint, that is $\overline{\Omega_{S,k}} \cap \overline{\Omega_{S,l}} = \emptyset$ for any $k \neq l$. We denote $\Omega_S = \bigcup_k \Omega_{S,k}$ and $\Omega = D \setminus \overline{\Omega_S}$, assuming that the family $\left(\Omega_{S,k}\right)_k$ is such that Ω is connected. Note that the latter assumption implies that $\Omega_{S,k}$ are simply connected.

Miranda Boutilier, Konstantin Brenner
Université Côte d'Azur, LJAD, France
e-mail: miranda.boutilier@univ-cotedazur.fr, konstantin.brenner@univ-cotedazur.fr

Victorita Dolean
University of Strathclyde, Dept. of Maths and Stats and Université Côte d'Azur, LJAD, CNRS, France, e-mail: work@victoritadolean.com

Let $f \in L^2(\Omega)$, in this article we are interested in the boundary value problem

$$\begin{cases} -\Delta u = f & \text{in } \Omega, \\ -\dfrac{\partial u}{\partial \mathbf{n}} = 0 & \text{on } \partial\Omega \cap \partial\Omega_S, \\ u = 0 & \text{on } \partial\Omega \setminus \partial\Omega_S. \end{cases} \tag{1}$$

Depending on the geometrical complexity of the computational domain, the numerical resolution of (1) may become challenging. A typical data set that we are interested in, illustrated by Figure 3, may contain numerous perforations that are described on different scales. In this regard, our strategy relies on the use of a Krylov solver combined with domain decomposition (DD) methods. Generally, to achieve scalability with respect to the number of subdomains in overlapping Schwarz methods, coarse spaces/components are needed. Including a coarse space in a Schwarz preconditioner results in what is referred to as a two-level Schwarz preconditioner.

The model problem can be thought of as the extreme limit case of the elliptic model containing highly contrasting coefficients. Two-level domain decomposition methods have been extensively studied for such heterogeneous problems. There are many classical results for coarse spaces that are contructed so as to resolve the jumps of the coefficients; see [6, 7, 12] for further details. Approaches to obtain a robust coarse space without careful partitioning of the subdomains include spectral coarse spaces such as those given in [8, 13, 15]. Additionally, the family of GDSW (Generalized Dryja, Smith, Widlund) methods [5] employ energy-minimizing coarse spaces and can be used to solve heterogeneous problems on less regular domains. These spaces are discrete in nature and involve both edge and nodal basis functions.

Alternatively, robust coarse spaces can be constructed using the ideas from multiscale finite elements methods (MsFEM) [1, 10]. The combination of spectral and MsFEM methods can be found in [9]. Outside of the DD framework, specifically on domains with small and numerous perforations, the authors of [4, 11] also introduced an enriched MsFEM-like method.

Here, we present an efficient and novel coarse space in the overlapping Schwarz framework inspired by the Boundary Element based Finite Element (BEM-FEM) method [16]. In contrast with the classical BEM-FEM approach, the local multiscale basis functions are computed numerically such as in MsFEM methods. This approach is motivated by our interest in nonlinear time dependent models for which the analytical expression of the fundamental solutions may not be easily available.

2 Discretization and preliminary notations

We introduce a coarse discretization of Ω which involves a family of polygonal cells $(\Omega_j)_{j=1,\dots,N}$, the so-called coarse skeleton Γ, and the set of coarse grid nodes that will be referred to by $\mathcal{V}$.

The construction is as follows. Consider a finite nonoverlapping polygonal partitioning of D denoted by $(D_j)_{j=1,\dots,N}$ and an induced nonoverlapping partitioning of Ω denoted by $(\Omega_j)_{j=1,\dots,N}$ such that $\Omega_j = D_j \cap \Omega$. We will refer to $(\Omega_j)_{j=1,\dots,N}$ as the coarse mesh over Ω. Additionally, we denote by Γ the skeleton of the coarse mesh, that is $\Gamma = \bigcup_{j\in\{1,\dots,N\}} \partial\Omega_j \setminus \partial\Omega_S$.

Let $\mathrm{vert}(\Omega_j)$ denote the set of vertices of the polygonal domain Ω_j. The set of coarse grid nodes is given as $\mathcal{V} = \bigcup_{j\in\{1,\dots,N\}} \mathrm{vert}(\Omega_j) \cap \overline{\Gamma}$. The total number of coarse grid nodes is denoted by $N_{\mathcal{V}}$.We refer to Figure 1 for the illustration of the coarse mesh entities.

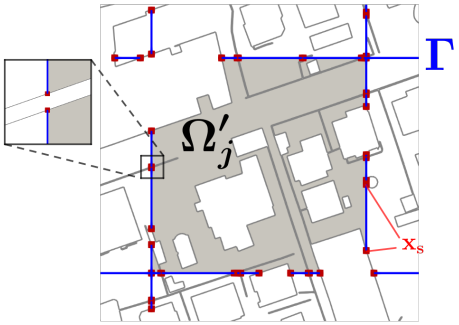
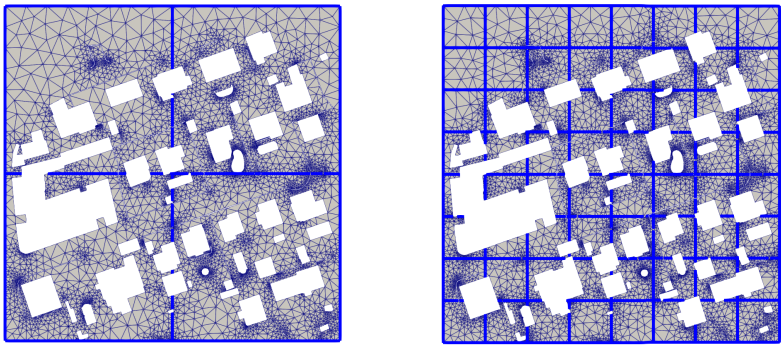


Fig. 1 Coarse grid cell Ω_j, nonoverlapping skeleton Γ (blue lines), and coarse grid nodes $\mathbf{x}_s \in \mathcal{V}$ (red dots). Coarse grid nodes are located at $\overline{\Gamma} \cap \partial\Omega_S$.

We discretize the model problem (1) with piecewise linear continuous finite elements on a triangular mesh of Ω. This mesh is conforming to the coarse polygonal $(\Omega_j)_{j=1,\dots,N}$; an example of the triangulation for various numbers of coarse cells N is given in Figure 2. The finite element discretization of (1) results in the linear system $\mathbf{Au} = \mathbf{f}$.



(a) 2×2 subdomains (b) 8×8 subdomains

Fig. 2 Conforming triangulation for the same domain Ω with different numbers of coarse cells N. The coarse skeleton Γ is shown by the blue lines.

Let $\left(\Omega_j'\right)_{j=1,\dots,N}$ denote the set of overlapping subdomains of Ω. In practice, each Ω_j' is constructed by propagating Ω_j by a few layers of triangles. Consider classical boolean restriction matrices $\mathbf{R}_j$ and corresponding extension matrices $\mathbf{R}_j^T$ associated to the family of overlapping subdomains $\left(\Omega_j'\right)_{j=1,\dots,N}$. With a coarse restriction matrix $\mathbf{R}_0$ that will be specified later, the two-level discrete Additive Schwarz (ASM) preconditioner is given by

$$M_{ASM,2}^{-1} = \mathbf{R}_0^T (\mathbf{R}_0 \mathbf{A} \mathbf{R}_0^T)^{-1} \mathbf{R}_0 + \sum_{j=1}^{N} \mathbf{R}_j^T (\mathbf{R}_j \mathbf{A} \mathbf{R}_j^T)^{-1} \mathbf{R}_j. \tag{2}$$

3 Description of the Trefftz-like coarse space

Here we introduce the Trefftz-like coarse space spanned by the functions that are piecewise linear on the skeleton Γ and discrete harmonic inside the nonoverlapping subdomains Ω_j. For any node $\mathbf{x}_s \in \mathcal{V}$, we introduce the function $g_s : \Gamma \to \mathbb{R}$, which is continuous on Γ and linear on each edge of Γ. It is clear that g_s is fully defined by its values at the nodes $\mathbf{x}_i \in \mathcal{V}$, for which we set

$$g_s(\mathbf{x}_i) = \begin{cases} 1, & s = i, \\ 0, & s \neq i. \end{cases}$$

To illustrate the construction of the nodal basis of the coarse space, we consider the following set of boundary value problems. For all Ω_j and for all $s = 1, \dots, N_{\mathcal{V}}$, find $\phi_s^j \in H^1(\Omega_j)$ such that ϕ_s^j is the weak solution to the following problem

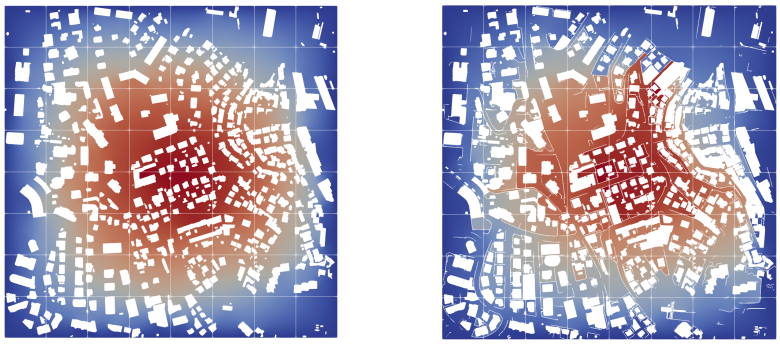
$$\begin{cases} -\Delta \phi_s^j = 0 & \text{in } \Omega_j, \\ -\dfrac{\partial \phi_s^j}{\partial \mathbf{n}} = 0 & \text{on } \partial\Omega_j \cap \partial\Omega_S, \\ \phi_s^j = g_s & \text{on } \partial\Omega_j \setminus \partial\Omega_S. \end{cases} \tag{3}$$

The finite element discretization of (3) results in the system of the form $\mathbf{A}_j' \boldsymbol{\phi}_s^j = \mathbf{b}_s^j$, where $\mathbf{A}_j'$ is the local stiffness matrix and $\mathbf{b}_s^j$ accounts for the Dirichlet boundary data in (3). Let $\overline{\mathbf{R}}_j$ denote the restriction matrices corresponding to $\overline{\Omega_j}$, and let $\boldsymbol{\phi}_s$ be a vector such that $\overline{\mathbf{R}}_j \boldsymbol{\phi}_s = \boldsymbol{\phi}_s^j$ for all $j = 1, \dots, N$. The coarse space is then defined as the span of the basis functions $\phi_s, s = 1, \dots, N_{\mathcal{V}}$, while the kth row of $\mathbf{R}_0$ is given by $\boldsymbol{\phi}_k^T$ for $k = 1, \dots, N_{\mathcal{V}}$.

4 Numerical results

We present below the numerical experiments concerning the performance of the conjugate gradient (CG) method using two-level preconditioner (2) and the Trefftz-like coarse space introduced in Section 3. For the sake of comparison we also report the numerical results obtained using a more standard Nicolaides coarse space which is going to be detailed later.

The data sets used in this experiment have been kindly provided by Métropole Nice Côte d'Azur and reflect the structural topography of the city of Nice. Although this type of data is available for the whole city [3], we focus here on a relatively small special frame (see Figure 3). In this numerical experiment we consider two kinds of structural elements - buildings (and assimilated small elevated structures) and walls. We note that the perforations resulting from the data sets we use (especially the wall data) can span across multiple coarse cells, which is a challenging situation for traditional coarse spaces.



(a) Without walls (b) With walls

Fig. 3 Approximate solution over a computational domain divided in $N = 8 \times 8$ nonoverlapping subdomains.

In this numerical experiment we consider the problem (1) with the-right-hand side given by $f = 1$. Figure 3 reports the finite element solution obtained for the data excluding and including walls. The figure also reflects the nonoverlapping partitioning into $N = 8 \times 8$ subdomains.

Figure 4 and Table 1 report the performance of the two-level preconditioner used in the PCG method, for varying number of subdomains N and two relative overlap sizes. As the computational domain Ω remains fixed independently of N, the results of this experiment could be interpreted in terms of a strong scalability. However, we wish to stress that the fine-scale triangulation is obtained based on the nonoverlapping partitioning $(\Omega_j)_{j=1,\dots,N}$. Consquentially, the linear system $\mathbf{Au} = \mathbf{f}$ changes from one coarse partitioning to another. Nevertheless we ensure that the dimension of the system is roughly constant throughout the experiment. Depending

on the chosen N, the linear system involves about $60k$ (buildings alone) and $180k$ (buildings and walls) nodal unknowns.

For the sake of comparison, we also provide numerical results for the well-known Nicolaides coarse space [14], made of flat-top partition of unity functions associated with the overlapping partitioning. As the scalability provided by the Nicolaides space relies on the Poincaré inequality over the subdomains, we further partition $(\Omega'_j)_j$ into a family of connected regions for this space. In other words, let m_j denote the number of disconnected components for each overlapping subdomain Ω'_j and let $\Omega'_{j,l}, l = 1, \ldots, m_j$ denote the corresponding disconnected component. Then our new overlapping partioning contains $m = \sum_{j=1}^{N} m_j$ total subdomains and is given by

$$(\widehat{\Omega}_k)_{k\in\{1,\ldots,m\}} = \left((\Omega'_{j,l})_{l\in\{1,\ldots,m_j\}}\right)_{j\in\{1,\ldots,N\}}.$$

Then, the Nicolaides coarse space is as follows. The kth row of $\mathbf{R}_0$ and therefore the kth column of $\mathbf{R}_0^T$ is given by $(\mathbf{R}_\mathbf{0}^\mathbf{T})_k = \widehat{\mathbf{R}}_k^T \widehat{\mathbf{D}}_k \widehat{\mathbf{R}}_k \mathbf{1}$ for $k = 1, \ldots, m$, where $\widehat{\mathbf{R}}_k$ and $\widehat{\mathbf{D}}_k$ are the restriction and partition of unity matrices corresponding to $\hat{\Omega}_k$ and $\mathbf{1}$ is a vector full of ones. The partition of unity matrices are constructed such that $\mathbf{I} = \sum_{k=1}^{m} \widehat{\mathbf{R}}_k^T \widehat{\mathbf{D}}_k \widehat{\mathbf{R}}_k$.

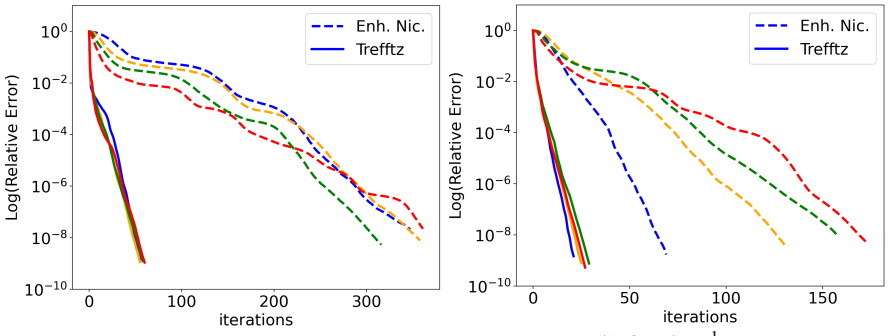
Figure 4 reports convergence histories of the preconditioned CG method using the Nicolaides and Trefftz-like coarse spaces for the data set including both walls and buildings. Table 1 summarizes the numerical performance for data sets including or excluding walls. In particular, for both preconditioners, it reports the dimensions of the coarse spaces, as well as the number of CG iterations required to achieve a relative l^2 error of 10^{-8}.

The performance of the Trefftz-like coarse space appears to be very robust with respect to both N and the complexity of the computational domain. The improvement with respect to the alternative Nicolaides approach is quite striking, especially in the case of the minimal geometric overlap. As expected, increased overlap in the first level of the Schwarz preconditioner provides additional acceleration in terms of iteration count. However, for the Trefftz-like space, the results with minimal geometric overlap appear to already be quite reasonable.

The dimensions and the relative dimensions of the two coarse spaces are reported in Table 1. Relative dimension refers to the would-be dimension of the coarse space in the case of a homogeneous domain with $\Omega_S = \emptyset$, that is, the relative dimensions are computed as $\frac{\dim(R_0)}{(\sqrt{N}+1)^2}$ for the Trefftz-like space and as $\frac{\dim(R_0)}{N}$ for the Nicolaides space. We observe that the Trefftz-like coarse space requires a much larger number of degrees of freedom, which naturally leads to a large coarse system to solve. We note that the contrast between the dimensions of two spaces reduces as N grows. In general, the dimension of the Trefftz-like coarse space seems reasonable given the geometrical complexity of the computational domain.

Table 1 PCG iterations, condition number, dimension, and relative dimension for the Trefftz-like and Nicolaides coarse spaces. Results are shown for minimal geometric overlap and $\frac{1}{20}H$, where $H = \max_j \operatorname{diam}(\Omega_j)$. As the dimension of the Nicolaides space will change with respect to the overlap, its dimension is given as the average dimension over the two overlap values.

		Nicolaides					Trefftz				
		it.		cond.		dim. (rel)	it.		cond.		dim. (rel)
N		min.	$\frac{H}{20}$	min.	$\frac{H}{20}$		min	$\frac{H}{20}$	min.	$\frac{H}{20}$	
16	no walls	149	51	581	82	21 (1.3)	52	28	59	11	170 (6.8)
	walls	348	70	6826	133	96 (6.0)	56	22	136	7	400 (16.0)
64	no walls	164	78	567	119	85 (1.3)	50	28	50	12	433 (5.3)
	walls	359	132	5902	297	256 (4.0)	56	26	57	9	880 (10.9)
256	no walls	136	81	273	89	312 (1.2)	56	27	54	10	1010 (3.5)
	walls	317	159	4575	12	719 (2.8)	59	30	60	13	1912 (6.6)
1024	no walls	120	83	341	149	1204 (1.2)	56	28	76	13	2500 (2.3)
	walls	362	174	3895	1310	2044 (2.0)	61	28	97	13	4253 (3.9)



(a) Minimal geometric overlap (b) Overlap $\frac{1}{20}L$

Fig. 4 Convergence curves for the Trefftz-like (solid lines) and Nicolaides (dashed lines) coarse spaces for the data set involving both buildings and walls and two overlap sizes. Colors correspond to the number of subdomains as follows: $N = 16$ (blue), $N = 64$ (orange), $N = 256$ (green), $N = 1024$ (red).

5 Conclusions

In this work we presented a novel Trefftz-like coarse space for the two-level ASM preconditioner, specifically designed for problems resulting from elliptic PDEs in perforated domains. This coarse space is robust with respect to data complexity and number of subdomains on a fixed total domain size, and provides significant acceleration in terms of Krylov iteration counts when compared to a more standard Nicolaides coarse space. This improvement comes at the price of a somewhat larger coarse problem. Current work in progress involves coarse approximation error and stable decomposition estimates and is left to a future article by the same authors. We are also planning to extend the presented two-level preconditioning strategy to nonlinear PDEs that model free-surface flows.

Acknowledgements This work has been supported by ANR Project Top-up (ANR-20-CE46-0005). The high-resolution structural data has been provided by Métropole Nice Côte d'Azur. We warmly thank Florient Largeron, chief of MNCA's SIG 3D project, for his help in preparation of the data and for the multiple fruitful discussions.

References

1. Aarnes, J. and Hou, T. Y. Multiscale domain decomposition methods for elliptic problems with high aspect ratios. *Acta Math. Appl. Sin. Engl. Ser.* **18**(1), 63–76 (2002).
2. Alonso, R., Santillana, M., and Dawson, C. On the diffusive wave approximation of the shallow water equations. *Eur. J. Appl. Math.* **19**(5), 575–606 (2008).
3. Andres, L. L'apport de la donnée topographique pour la modélisation 3D fine et classifiée d'un territoire. *Rev. XYZ* **133**(4), 24–30 (2012).
4. Brown, D. L. and Taralova, V. A multiscale finite element method for Neumann problems in porous microstructures. *Discrete & Continuous Dynamical Systems-S* **9**(5), 1299 (2016).
5. Dohrmann, C. R., Klawonn, A., and Widlund, O. B. Domain decomposition for less regular subdomains: Overlapping Schwarz in two dimensions. *SIAM J. Numer. Anal.* **46**(4), 2153–2168 (2008).
6. Dryja, M., Sarkis, M. V., and Widlund, O. B. Multilevel schwarz methods for elliptic problems with discontinuous coefficients in three dimensions. *Numer. Math.* **72**(3), 313–348 (1996).
7. Dryja, M., Smith, B. F., and Widlund, O. B. Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. *SIAM J. Numer. Anal.* **31**(6), 1662–1694 (1994).
8. Galvis, J. and Efendiev, Y. Domain decomposition preconditioners for multiscale flows in high-contrast media. *Multiscale Model. Simul.* **8**(4), 1461–1483 (2010).
9. Gander, M. J., Loneland, A., and Rahman, T. Analysis of a new harmonically enriched multiscale coarse space for domain decomposition methods. *arXiv preprint arXiv:1512.05285* (2015).
10. Graham, I. G., Lechner, P., and Scheichl, R. Domain decomposition for multiscale PDEs. *Numer. Math.* **106**(4), 589–626 (2007).
11. Le Bris, C., Legoll, F., and Lozinski, A. An MsFEM type approach for perforated domains. *Multiscale Model. Simul.* **12**(3), 1046–1077 (2014).
12. Mandel, J. and Brezina, M. Balancing domain decomposition for problems with large jumps in coefficients. *Math. Comput.* **65**(216), 1387–1401 (1996).
13. Nataf, F., Xiang, H., Dolean, V., and Spillane, N. A coarse space construction based on local Dirichlet-to-Neumann maps. *SIAM J. Sci. Comput.* **33**(4), 1623–1642 (2011).
14. Nicolaides, R. A. Deflation of conjugate gradients with applications to boundary value problems. *SIAM J. Numer. Anal.* **24**(2), 355–365 (1987).
15. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., and Scheichl, R. Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. *Numer. Math.* **126**(4), 741–770 (2014).
16. Weißer, S. *BEM-based Finite Element Approaches on Polytopal Meshes*, vol. 130. Springer (2019).

On Global and Monotone Convergence of the Preconditioned Newton's Method for Some Mildly Nonlinear Systems

Konstantin Brenner

1 Introduction

Let β be a diagonal mapping from $\mathbb{R}^N$ to itself and let A be an $N \times N$ real matrix. Given some $r \in \mathbb{R}^N$, we are interested in solving numerically the flowing system of nonlinear equations

$$\beta(u) + Au = r. \tag{1}$$

Such mildly nonlinear systems with only a diagonal nonlinearity are commonly found in the context of geophysical flow and transport modeling, where they result from the discretization of nonlinear evolutionary PDEs. They arise, for example, from Richards' and porous medium equations, or, alternatively, from the models of reactive transport involving equilibrium adsorption.

This contribution is concerned with the global convergence analysis of preconditioned Newton's methods applied to (1). Nonlinear preconditioning is an increasingly popular technique that may drastically improve the robustness and convergence rate of linearization schemes such as Newton's method. As in the case of linear problems, the nonlinear preconditioning consists in replacing the original system by an equivalent one that can be solved more efficiently. Since more than twenty years variety of nonlinear preconditioning methods has been proposed, including Schwarz-inspired methods ASPIN [4], MSPIN [9], [13] and RASPEN [6], as well as the nonlinear versions of FETI-DP [10] and BDDC [11].

Nonlinear preconditioning appears to be particularly efficient in the application to models of subsurface flow and reactive transport [9], [12], where the failures and lack of robustness of nonlinear solvers is one the major factors limiting the reliability of the simulation codes. In those applications, the major benefit seems to result in the form of an extended convergence region, which, in case of time-dependent problems, allows for larger time steps [12].

Konstantin Brenner
Université Côte d'Azur, LJAD, CNRS, INRIA, e-mail: konstantin.brenner@univ-cotedazur.fr

The present work aims to contribute to the theoretical analysis of nonlinear preconditioning methods, which remains relatively unexplored. We extend the previous work [2], concerned with the Jacobi-Newton method, and cover the nonlinear counterparts of some popular linear preconditioners based on multi-splitting of the system. Our analysis includes, in particular, nonlinear preconditioning by block Jacobi or RAS methods. We prove that, under appropriate assumptions discussed below, the one-level RAS-Newton method (or RASPEN [6]) applied to (1) exhibits global and essentially monotone convergence. The analysis of this method is carried out in the framework on nonlinear multi-splitting methods [7, 8], and extends to other methods such as, for example, block Gauss-Seidel.

As an alternative to nonlinear preconditioning, we study a simpler two-step scheme alternating the nonlinear multi-splitting and the standard Newton linearization steps. The two-step multi-splitting/Newton scheme enjoys the same global and monotone convergence properties as the full preconditioned method. We note that in the context of the RAS approach, such scheme has been proposed in [5] under the name of NKS-RAS method. It turns out that for simple splitting methods, like (block) Jacobi or Gauss-Seidel, the preconditioned Newton's method is equivalent to the former two-step approach.

Our convergence analysis relies on the Monotone Newton Theorem [1, 14]; and requires two major assumptions on the system (1), namely the concavity of the nonlinear map involved in (1) and the assumption that the Jacobian of the system has a nonnegative inverse. More specifically, we will assume the following

(A_1) For each $0 \leq i \leq N$, the functions $\beta_i \in C^1(\mathbb{R})$ are monotone and concave.
(A_2) For any $u \in \mathbb{R}^N$, the matrix $\beta'(u) + A$ is an M-matrix.

We wish to stress that the assumptions (A_1) and (A_2) are quite sub-optimal and aim to improve reader's experience at the expense of sharpness. For example, the generalizations of (A_1) can be performed along the following lines. First, one can relax the regularity assumption; clearly piecewise regular functions β_i would do. Secondly, the derivative of β_i need not to be bounded, or alternatively β_i need not be defined over $\mathbb{R}$, such case has been treated in [2]. As a matter of fact, we believe that the analysis presented here can be extended to β_i being merely maximal monotone and concave (in some appropriate sense). We also note that the analysis presented below applies to β convex instead on concave. The explicit concavity assumption is motivated the the applications to porous media flow models that we have in mind. Similarly, the assumption (A_2) can be relaxed by allowing positive off-diagonal elements in the Jacobian, assuming, for example, that A is nonsingular and $A^{-1} \geq 0$.

Before moving any further, let us recall some basic properties of the system (1):

Proposition 1 (Existence and uniqueness of solution)

Let $F(u) = \beta(u) + Au$. Under the assumptions (A_1) and (A_2), the mapping F^{-1} is well defined on $\mathbb{R}^N$ and is convex.

Next, we state the global version of the Monotone Newton Theorem, for which we refer to [14].

Theorem 1 (Global monotone Newton theorem)

Let $\mathcal{F}\colon \mathbb{R}^N \to \mathbb{R}^N$ be continuous, Gâteaux differentiable and concave. Suppose that $\mathcal{F}'(u)$ has a nonnegative inverse for all $u \in \mathbb{R}^N$, and assume that $\mathcal{F}(u) = 0$ has a solution. Then, for any $u_0 \in \mathbb{R}^N$, the sequence

$$u_{n+1} = u_n - \mathcal{F}'(u_n)^{-1}\mathcal{F}(u_n), \qquad n \geq 0$$

satisfies $\mathcal{F}(u_n) \leq 0$ and $u_n \leq u_{n+1} \leq \mathcal{F}^{-1}(0)$ for all $n \geq 1$. If, in addition, there exists an invertible $P \in \mathbb{M}(N)$ such that $\mathcal{F}'(u)^{-1} \geq P \geq 0$ for all $u \in \mathbb{R}^N$, then the sequence u_n converges to $\mathcal{F}^{-1}(0)$.

In view of Theorem 1 and Proposition 1 one can deduce that Newton's method applied to the system (1) converges regardless of the initial guess. Unfortunately, depending on the stiffness of the function β, this convergence may become arbitrarily slow as pointed out in [3] and [2]. While the lack of robustness with respect to the shape of β can be addressed by diagonal Jacobi preconditioning [2], the efficiency of the Jacobi-Newton method for systems resulting from the discretization of degenerate PDEs is still controlled by the mesh size, which motivates the use of nonlinear preconditioning of domain decomposition type.

Having in mind the application to the overlapping domain decomposition, we introduce in Section 2 the preconditioning technique based on the nonlinear multi-splitting of (1). We prove that the preconditioned system satisfies Theorem 1 and, therefore, Newton's method is unconditionally convergent. In Section 3 we present the numerical results based on a discretized porous media equation [16] and using some variants of nonlinear RAS method including RASPEN and RAS/Newton two-step methods (NKS-RAS method from [5]).

2 Nonlinear multi-splitting method

In this section we present the nonlinear preconditioning procedure inspired by the linear multi-splitting methods [7], [15].

Let $(P_i, Q_i)_{i=1,\dots,K}$ be a finite of family matrices such that $A = P_i - Q_i$. We denote $M_i(u) = \beta(u) + P_i u$ and $N_i(u) = Q_i u + r$. If $M_i(u)$ admits an inverse defined on $\mathbb{R}^N$, then one can reformulate the original problem (1) as

$$\mathcal{F}_i(u) := u - M_i^{-1}(N_i(u)) = 0. \tag{2}$$

Let $(E_i)_{i=1,\dots,K}$ be a family of nonnegative diagonal matrices such that $\sum_{i=1}^K E_i = I$. Multiplying (2) by E_i and summing over i we obtain the system

$$\mathcal{F}(u) := \sum_{i=1}^K E_i \mathcal{F}_i(u) = u - \sum_{i=1}^K E_i M_i^{-1}(N_i(u)) = 0. \tag{3}$$

Clearly, the solution of the original system satisfies (3). The proposition below states that $\mathcal{F}$ is concave and that $\mathcal{F}'(u)$ has nonnegative inverse for all u, which implies, in particular, that $\mathcal{F}$ is inverse isotone, and, therefore, the solution to (3) is unique.

Proposition 2 *Assume that for all $u \in \mathbb{R}^N$ and all i, the splitting*

$$F'(u) = M_i'(u) - Q_i$$

is weakly regular and that $M_i'(u)$ is an M-matrix. Then, the mapping $\mathcal{F}(u)$ from (3) *is a concave bijection from $\mathbb{R}^N$ to $\mathbb{R}^N$, and, for all $u \in \mathbb{R}^N$, the matrix $\mathcal{F}'(u)$ is an M-matrix satisfying $\mathcal{F}'(u)^{-1} \geq I$.*

Proof In view of Proposition 1 the mappings M_i^{-1} are well defined on $\mathbb{R}^N$ and are convex, which implies that $\mathcal{F}$ is concave since $E_i \geq 0$. Let us show that for all $u \in \mathbb{R}^N$, the matrix $\mathcal{F}'(u)$ is an M-matrix satisfying $\mathcal{F}'(u)^{-1} \geq I$. We begin with the following spectral bound, which is the founding stone for the analysis of the multi-splitting methods (see [15])

$$\rho\left(\sum E_i M_i'(u)^{-1} Q_i\right) < 1. \tag{4}$$

Let $\widetilde{u}_i = M_i^{-1}(N_i(u))$, we have $\mathcal{F}'(u) = I - \sum_i E_i M_i'(\widetilde{u}_i)^{-1} Q_i$. Let $w \in \mathbb{R}^N$ be a component-wise maximum of the vectors $\widetilde{u}_i$; that is $(w)_k = \max_i (\widetilde{u}_i)_k$. Since $u \mapsto M_i'(u)$ is antitone and $M_i'(u)$ has nonnegative inverse, we deduce that $M_i'(\widetilde{u}_i)^{-1} M_i'(w) \leq I$, and, since $M_i'(\widetilde{u}_i)^{-1} Q_i \geq 0$ and $M_i'(w)^{-1} Q_i \geq 0$, we obtain

$$M_i'(\widetilde{u}_i)^{-1} Q_i = M_i'(w)^{-1} Q_i + \left(M_i'(\widetilde{u}_i)^{-1} M_i'(w) - I\right) M_i'(w)^{-1} Q_i \leq M_i'(w)^{-1} Q_i$$

and $0 \leq \sum_i E_i M_i'(\widetilde{u}_i)^{-1} Q_i \leq \sum_i E_i M_i'(w)^{-1} Q_i$. It follows from (4) that

$$\rho\left(\sum_i E_i M_i'(\widetilde{u}_i)^{-1} Q_i\right) \leq \rho\left(\sum_i E_i M_i'(w)^{-1} Q_i\right) < 1,$$

which implies in turn that $\mathcal{F}'(u)^{-1} \geq 0$. Clearly the off-diagonal part of $\mathcal{F}'(u)$ is nonpositive, implying that it is M-matrix; moreover, since $\mathcal{F}'(u) \leq I$, we deduce that $\mathcal{F}'(u)^{-1} \geq I$. □

Based on Proposition 2 one shows that the mapping $\mathcal{F}$ satisfies the assumptions of Theorem 1. In addition, we consider the following multi-splitting/Newton two-step scheme: Given $u_0 \in \mathbb{R}^N$, compute for all $n \geq 0$

$$\widetilde{u}_n = \sum_i E_i M_i^{-1}(N_i(u_n)) \tag{5}$$

and

$$u_{n+1} = \widetilde{u}_n - F'(\widetilde{u}_n)^{-1} \left(F(\widetilde{u}_n) - r\right). \tag{6}$$

We note that (5) can be interpreted as a step of a quasi-Newton method applied to (3), where the matrix $\mathcal{F}(u)^{-1}$ has been replaced by its subinverse I. It can be shown that (5)–(6) leads again to a globally convergent scheme. Remarkably enough, in the case of a simple splitting, like (block) Jacobi or Gauss-Seidel, the two-step scheme is equivalent to the preconditioned Newton's method.

Proposition 3 *Let $A = P - Q$ be some splitting such that the inverse of $M = \beta(u) + Pu$ is well defined and $M'(u)$ is non-singular for all $u \in \mathbb{R}^N$. Then, the two-step scheme and the preconditioned Newton's generate the same iterates.*

Proof Let $\widetilde{u} = M^{-1}(N(u))$, we remark that $\mathcal{F}(u) = u - \widetilde{u}$ and $\mathcal{F}'(u) = I - M'(\widetilde{u})^{-1}Q = M'(\widetilde{u})^{-1}F'(\widetilde{u})$. Therefore, the update generated by the preconditioned Newton's method starting from u is given by

$$\delta_{\text{prec}}(u) = \left(M'(\widetilde{u})^{-1}F'(\widetilde{u})\right)^{-1}(\widetilde{u} - u)\,. \tag{7}$$

Now, let us consider the update generated by the two-step method (5)-(6). We have $\delta_{\text{two-step}}(u) = \widetilde{u} - u - F'(\widetilde{u})^{-1}F(\widetilde{u})$. We remark that $F(\widetilde{u}) = M(\widetilde{u}) - N(\widetilde{u}) = N(u) - N(\widetilde{u})$, and using linearity of N, we deduce that $F(\widetilde{u}) = Q(u - \widetilde{u})$. Therefore,

$$\delta u_{\text{two-step}} = \left(I + F'(\widetilde{u})^{-1}Q\right)(\widetilde{u} - u) = F'(\widetilde{u})^{-1}M'(\widetilde{u})(\widetilde{u} - u),$$

which, in view of (7), provides $\delta_{\text{two-step}}(u) = \delta_{\text{prec}}(u)$. □

3 Numerical experiment

We now proceed with the numerical experiment that illustrates the performance of block Jacobi-Newton, RASPEN and the two-step RAS/Newton methods applied to the system resulting from the discretization of a degenerate parabolic equation. The (block) Jacobi-Newton consists of applying Newton's method to the system of the form (2) obtained from a simple splitting $A = P - Q$, where P is a (block) diagonal part of A. On the other hand, RASPEN can be expressed as Newton's method applied to the system (3) resulting from a particular multi-splitting (wee refer to [8] for further details). Using same multi-splitting, the RAS/Newton method is given by (5)–(6).

The test case considered here is similar to the one presented in [2] to which we refer for more detailed discussion. In brief, we are interested in the algebraic system resulting from the implicit in time discretization of the porous media equation [16]. More specifically, focusing on a single step (of length τ) of the backward Euler time integration scheme, we consider the system of the form (1) resulting from the finite difference discretization of the following boundary value problem

$$\begin{cases} \beta(u) - \beta(u_{ini}) = \tau\partial^2_{xx}u & x \in (0,1), \\ \partial_x u(0) = -q, \quad \partial_x u(1) = 0, \end{cases} \tag{8}$$

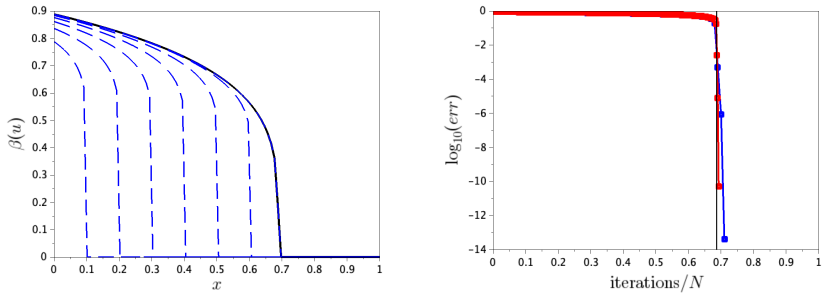


Fig. 1 Left: the solution for $N = 100$ (black) and the iterates $\beta(u_n)$ of the Jacobi-Newton method for $n = 10, 20, \ldots 70$. Right: convergence history of the Jacobi-Newton method for $N = 100$ (blue) and 400 (red), the iteration count is scaled by the size of the discrete system; the error is measured in l_∞ norm. The vertical black line positioned at N_f/N indicate the location of the solution front.

where $\beta(u) = u^{1/m}$ with $m > 1$. We consider the following set of parameters: $m = 10$, $q = 1$, $\tau = 0.5$, and $\beta(u_{ini}) = 10^{-6}$. The problem (8) is discretized using $N = 100$ or 400 degrees of freedom, and the vector u_{ini} is used as the initial guess by the iterative methods under consideration.

We note that, since the derivative of β is unbounded at the origin, one may consider the change of variable in (8). For example, using $\beta(u)$ as the new unknown will improve the performance of the straightforward Newton's method [3]. Unfortunately, the modified system is no longer concave and, therefore, the monotone convergence is lost; moreover, compared to the splitting-based preconditioning, the convergence of Newton's method applied to the modified system turns out to be slower [2].

The solutions of the porous media equation are characterized by the finite speed of propagation of the support. Qualitatively this behavior persists even for strictly positive but small initial data. For the discrete counterpart of the elliptic problem (8) the latter property is reflected in the performance of Newton's method. Typically, and unless some Schwarz-type preconditioning is performed, the solution fronts resulting from Newton's method can cross at most one degree of freedom at time. For the Jacobi-Newton method, this behavior is illustrated by Figure 1. The left sub-figure exhibits the final position of the solution front and some iterates of the method. The right sub-figure reports the convergence history of the method for two values of the mesh size. The numerical performance is characterized by two very distinct regimes: a very fast near-solution convergence is preceded by a long period of a slow error decrease. As a matter of fact, the length of the convergence plateau is proportional to the number of the degrees of freedom N_f that has to be crossed by the solution front, and can be expressed as σN_f, where σ is the cost of propagating the front trough one degree of freedom. As shown in [3] and [2], the parameter σ of the standard Newton's method can become arbitrarily large depending on the coefficient m and the initial data. In contrast, the Jacobi-Newton method [2] appears to be virtually independent of m and can handle general nonnegative initial data. Nevertheless, the efficiency of the latter method is still dependent on N_f and thus on

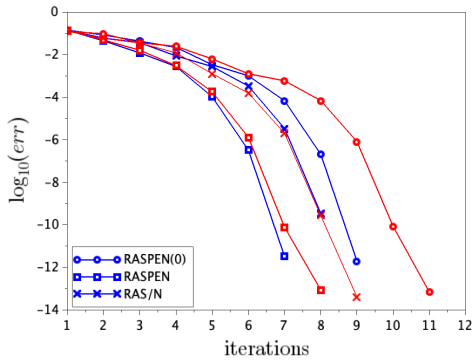


Fig. 2 Convergence history of preconditioned Newton's method for $N = 100$ (blue) and 400 (red); the error is measured in l_∞ norm.

the discretization. More precisely, the right side of Figure 1 reflects the convergence of the Jacobi-Newton for $N = 100$ and 400 degrees of freedom. By scaling the iteration count by N we observe that the performance of the method is essentially controlled by N_f. The vertical black line positioned at N_f/N reflects the final location of the front. The scaled convergence curves are almost identical, while the total iteration count measured for the Jacobi-Newton method is of 71 for $N = 100$ and 271 for $N = 400$.

The dependency of the mesh size can be removed by means of the nonlinear domain decomposition. We report on Figure 2 the convergence history of RASPEN and RAS/Newton (RAS/N) methods for 5 equally sized sub-domain with the relative overlap of 0.1. In addition, we consider the case of the minimal algebraic overlap, denoted RASPEN(0), corresponding to preconditioning based on the block Jacobi method. In this numerical experiment none of the considered methods appear to exhibit any substantial dependency on the mesh size. Unsurprisingly, the overlap seems to be beneficial for the convergence of both RASPEN and RAS/N. While being slightly less efficient than RASPEN, the two-step RAS/N method still appears as a competitive alternative. The convergence of the nonlinear RAS method, applied as a solver instead of being used as a preconditioner, is not reported here, but roughly speaking, the nonlinear RAS method is as inefficient as the linear one.

4 Conclusion

We have analyzed a family of preconditioned Newton methods based on the nonlinear multi-splitting approach in application to mildly nonlinear systems resulting from the discretization of some degenerate evolutionary PDEs such as porous media or Richards' equation. Based on the Monotone Newton Theorem we show that

the preconditioned method is globally convergent. The current result extends our previous analysis [2] to the one-level RASPEN method [6]. In addition, for the preconditioning based on a single nonlinear splitting, including the method presented in [2], the preconditioned Newton's method is equivalent to a simpler to implement predictor-corrector scheme. The numerical experiment based on discrete porous media equation shows that the performance of block Jacobi-Newton, RASPEN and RAS/Newton methods is essentially independent of the mesh size.

References

1. Baluev, A. On the abstract theory of Chaplygin's method. In: *Dokl. Akad. Nauk. SSSR*, vol. 83, 781–784 (1952).
2. Brenner, K. On the monotone convergence of Jacobi–Newton method for mildly nonlinear systems. *Journal of Computational and Applied Mathematics* **419**, 114719 (2023).
3. Brenner, K. and Cances, C. Improving Newton's method performance by parametrization: the case of the Richards equation. *SIAM Journal on Numerical Analysis* **55**(4), 1760–1785 (2017).
4. Cai, X.-C. and Keyes, D. E. Nonlinearly preconditioned inexact Newton algorithms. *SIAM Journal on Scientific Computing* **24**(1), 183–200 (2002).
5. Cai, X.-C. and Li, X. Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity. *Siam journal on scientific computing* **33**(2), 746–762 (2011).
6. Dolean, V., Gander, M. J., Kheriji, W., Kwok, F., and Masson, R. Nonlinear preconditioning: How to use a nonlinear Schwarz method to precondition Newton's method. *SIAM Journal on Scientific Computing* **38**(6), A3357–A3380 (2016).
7. Frommer, A. Parallel nonlinear multisplitting methods. *Numerische Mathematik* **56**, 269–282 (1989).
8. Frommer, A. and Szyld, D. B. An algebraic convergence theory for restricted additive Schwarz methods using weighted max norms. *SIAM journal on numerical analysis* **39**(2), 463–479 (2001).
9. Kern, M., Taakili, A., and Zarrouk, M. Preconditioned iterative method for reactive transport with sorption in porous media. *Mathematical Modelling and Analysis* **25**(4), 546–568 (2020).
10. Klawonn, A., Lanser, M., and Rheinbach, O. Nonlinear feti-dp and bddc methods. *SIAM Journal on Scientific Computing* **36**(2), A737–A765 (2014).
11. Klawonn, A., Lanser, M., and Rheinbach, O. Nonlinear BDDC methods with approximate solvers (2018).
12. Klemetsdal, Ø., Moncorgé, A., Møyner, O., and Lie, K.-A. A numerical study of the additive schwarz preconditioned exact newton method (aspen) as a nonlinear preconditioner for immiscible and compositional porous media flow. *Computational Geosciences* 1–19 (2021).
13. Liu, L. and Keyes, D. E. Field-split preconditioned inexact Newton algorithms. *SIAM Journal on Scientific Computing* **37**(3), A1388–A1409 (2015).
14. Ortega, J. M. and Rheinboldt, W. C. *Iterative solution of nonlinear equations in several variables*. SIAM (2000).
15. O'Leary, D. P. and White, R. E. Multi-splittings of matrices and parallel solution of linear systems. *SIAM Journal on algebraic discrete methods* **6**(4), 630–640 (1985).
16. Vázquez, J. L. *The porous medium equation: mathematical theory*. Oxford University Press on Demand (2007).

Optimized Schwarz Method in Time for Transport Control

Duc-Quang Bui, Bérangère Delourme, Laurence Halpern, and Felix Kwok

1 Introduction

Parallel-in-time methods for solving optimal control problems under time-dependent PDE constraints have gained much interest in the past decade (see, e.g., ParaOpt [5]). Among all the possible approaches, it is natural to consider Schwarz time domain decomposition techniques when one deals with transport equations, since the original control problem is equivalent to an elliptic problem in which the initial and target conditions play the role of boundary conditions (see e.g. [1]).

In this paper, we consider the following one-dimensional transport control problem. Let $T > 0$, and let y_{ini} and y_{tar} be two periodic functions in $\in L^2_{\text{loc}}(\mathbb{R})$ with period one. We want to find a control $v \in L^2_{\text{loc}}(\mathbb{R} \times (0,T))$, periodic in space of period one, such that the function y defined by

$$\begin{cases} \partial_t y + \partial_x y = v \ \text{ in } \mathbb{R} \times (0,T), \\ y(.,0) = y_{\text{ini}}, \end{cases} \tag{1}$$

verifies the exact constraint

$$y(.,T) = y_{\text{tar}}. \tag{2}$$

Over all the possible controls v, we shall seek the one with minimal L^2-norm, namely, we minimize the functional

$$J(v) = \frac{1}{2} \int_0^T \|v\|^2_{L^2(0,1)}. \tag{3}$$

Duc Quang Bui, Bérangère Delourme, Laurence Halpern
Université Sorbonne Paris Nord, Villetaneuse, France, e-mail: bui@math.univ-paris13.fr, delourme@math.univ-paris13.fr, halpern@math.univ-paris13.fr

Felix Kwok
Université Laval, Québec, Canada, e-mail: felix.kwok@mat.ulaval.ca

The optimization problem (1)-(2)-(3) admits a unique solution v_* that can be deduced from the following optimality system: find (y, λ), 1-periodic in space, such that

$$\begin{cases} \partial_t y + \partial_x y = \lambda & \text{in } \mathbb{R} \times (0,T), \\ \partial_t \lambda + \partial_x \lambda = 0 & \text{in } \mathbb{R} \times (0,T), \\ y(.,0) = y_{\text{ini}}, \\ y(.,T) = y_{\text{tar}}, \end{cases} \qquad v_* = \lambda. \tag{4}$$

2 Domain decomposition in time for the continuous problem

We apply Schwarz-in-time domain decomposition methods to (4). To do so, we decompose the time interval $(0,T)$ into two subdomains $(0,T_1)$ and (T_1,T) with $T_1 = \Delta T = \dfrac{T}{2}$. To start with, we solve the system (4) using the optimized Schwarz method with Robin transmission conditions on the interface $t = \Delta T$, with a single parameter $\mathfrak{p}$. More specifically, at iteration k, the functions y_1^k and λ_1^k (resp. y_2^k and λ_2^k) are solutions to (4) on $(0,T_1)$ (resp (T_1,T)) together with the following boundary condition:

$$\mathfrak{p} y_1^k + \lambda_1^k = \mathfrak{p} y_2^{k-1} + \lambda_2^{k-1}, \quad -\mathfrak{p} y_2^k + \lambda_2^k = -\mathfrak{p} y_1^{k-1} + \lambda_1^{k-1}. \tag{5}$$

Theorem 1 *Let* $\mathfrak{p} = \dfrac{1}{\Delta T}$. *Then the Schwarz iterative algorithm based on* (5) *and applied to the system* (4) *converges after* 1 *iteration.*

The theorem is proven by calculating explicitly the solutions of the sub-domain problems. We point out that in [6], a convergence proof using energy estimates has been given for all $\mathfrak{p} > 0$. On the other hand, to our knowledge, there has not been a detailed analysis of the convergence factor on the corresponding discrete systems (see [4, 7] for a convergence proof for semi-discrete schemes in the parabolic case). Understanding the behaviour of the discrete systems is the subject of the next sections.

3 Time-domain decomposition for a discrete problem

3.1 Discrete control problem

To discretize our problem, we consider a spatial discretization based on the upwind scheme with N uniform nodes and a mesh size of $\Delta x = 1/N$. We denote by $\mathcal{A}_{\Delta x} \in \mathcal{M}_N(\mathbb{R})$ the corresponding matrix: its diagonal terms are Δx^{-1}, its lower sub-diagonal ones are equal to $-\Delta x^{-1}$, and $[\mathcal{A}_{\Delta x}]_{1,N} = -\Delta x^{-1}$ (to take into account the periodicity), and zero coefficients elsewhere. The time discretization is made using the semi-implicit Euler scheme (explicit in y and implicit in v), using $M+1$

uniform nodes on $[0,T]$ and a mesh size of $\Delta t = \frac{T}{M}$. We denote by $\mathbf{y}_{\text{ini}}$, $\mathbf{y}_{\text{tar}}$ (vectors of $\mathbb{R}^N$), the discretization of y_{ini} and y_{tar}. We mimic the continuous minimization problem (1)-(2)-(3) by considering the following discrete one:

$$\min_{\mathbf{v}=(\mathbf{v}_i^n)\in\mathbb{R}^{N\times M}} J(\mathbf{v}) = \frac{1}{2}\Delta t\,\Delta x\,\|\mathbf{v}\|^2, \tag{6}$$

where the control $\mathbf{v} = (\mathbf{v}^1,\dots,\mathbf{v}^M)$ is such that $\mathbf{y} = (\mathbf{y}^0,\dots,\mathbf{y}^M) \in (\mathbb{R}^N)^{M+1}$ satisfies

$$\begin{cases} \dfrac{\mathbf{y}^m - \mathbf{y}^{m-1}}{\Delta t} + \mathcal{A}_{\Delta x}\mathbf{y}^{m-1} = \mathbf{v}^m \quad m = 1,\dots,M, \\ \mathbf{y}^0 = \mathbf{y}_{\text{ini}}, \end{cases} \tag{7}$$

as well as the target constraint

$$\mathbf{y}^M = \mathbf{y}_{\text{tar}}. \tag{8}$$

In the problem (6), $\|\cdot\|$ denotes the usual Euclidean norm on $\mathbb{R}^{N\times M}$. As in the continuous case, Problem (6)-(7)-(8) admits a unique solution $\mathbf{v}_*^m = \boldsymbol{\lambda}^m$, where $(\mathbf{y}^m,\boldsymbol{\lambda}^m)$ is the solution of the following optimality system (see [3]):

$$\begin{cases} \mathbf{y}^m - (\mathcal{I} - \Delta t\,\mathcal{A}_{\Delta x})\mathbf{y}^{m-1} = \Delta t\boldsymbol{\lambda}^m \ m = 1,\dots,M, \\ \boldsymbol{\lambda}^{m-1} - (\mathcal{I} - \Delta t\,\mathcal{A}_{\Delta x}^t)\boldsymbol{\lambda}^m = 0 \qquad m = 1,\dots,M, \\ \mathbf{y}^0 = \mathbf{y}_{\text{ini}}, \\ \mathbf{y}^M = \mathbf{y}_{\text{tar}}. \end{cases} \tag{9}$$

In the sequel, in order to guarantee the convergence of the scheme, we shall consider the standard relation between Δt and Δx given by

$$\frac{\Delta t}{\Delta x} = r, \tag{10}$$

where r is a given real parameter in $(0,1)$.

3.2 Schwarz domain decomposition

We apply the Schwarz method strategy (5) to the system (9). For the sake of simplicity, let us consider $M = 2L$, so that the interface $T/2$ corresponds exactly to the node L. The algorithm then reads: starting from an initial guess $(\boldsymbol{\xi}_1^0,\boldsymbol{\xi}_2^0) \in \mathbb{R}^{2N}$, at each iteration $k \geq 1$, we construct $(\mathbf{y}_1^{k,m},\boldsymbol{\lambda}_1^{k,m})$ (respectively $(\mathbf{y}_2^{k,m},\boldsymbol{\lambda}_2^{k,m})$) solution to (9) for $m = 1,\dots,L$ (resp. $m = L+1,\dots,M$) together with the transmission conditions

$$\mathfrak{p}\mathbf{y}_1^{k,L} + \boldsymbol{\lambda}_1^{k,L} = \boldsymbol{\xi}_1^{k-1}, \quad -\mathfrak{p}\mathbf{y}_2^{k,L} + \boldsymbol{\lambda}_2^{k,L} = \boldsymbol{\xi}_2^{k-1}. \tag{11}$$

Then, we update $\boldsymbol{\xi}_1^k$ by taking

$$\boldsymbol{\xi}_1^k = \mathfrak{p}\mathbf{y}_2^{k,L} + \boldsymbol{\lambda}_2^{k,L}, \quad \boldsymbol{\xi}_2^k = -\mathfrak{p}\mathbf{y}_1^{k,L} + \boldsymbol{\lambda}_1^{k,L}. \tag{12}$$

Remark 1 The local subdomain problems are indeed optimality systems associated with local control problems (see [6]).

The convergence analysis of the algorithm (9)-(11)-(12) relies on the Discrete Fourier transform in space $\mathbb{R}^N \to \mathbb{R}^N$, $(u_1, \ldots, u_{N-1}) \mapsto (\hat{u}_0, \ldots \hat{u}_{N-1})$ defined by $\hat{u}_\ell = \sum_{n=0}^{N-1} u_n \exp(-2\pi \mathrm{i} \ell n \Delta x)$. Indeed, (9)-(11)-(12) can be transformed as follows: at iteration k, in subdomain Ω_i, for any ℓ between 0 and $N-1$ (spatial frequency), $\hat{y}_{i,\ell}^{k,m}$ (with m denoting the time step) solves

$$\begin{cases} \hat{y}_{i,\ell}^{k,m} - (1 - \sigma(\ell)\,\Delta t)\,\hat{y}_{i,\ell}^{k,m-1} = \Delta t \hat{\lambda}_{i,\ell}^{k,m}, \\ (1 - \overline{\sigma(\ell)}\Delta t)\hat{\lambda}_{i,\ell}^{k,m} - \hat{\lambda}_{i,\ell}^{k,m-1} = 0, \end{cases} \quad \text{where } \sigma(\ell) = \frac{1 - \exp(-2\pi \mathrm{i} \ell \Delta x)}{\Delta x}, \tag{13}$$

together with boundary conditions

$$\begin{cases} \hat{y}_{1,\ell}^{k,0} = \hat{y}_{\mathrm{ini},\ell}, \\ \mathfrak{p}\hat{y}_{1,\ell}^{k,L} + \hat{\lambda}_{1,\ell}^{k,L} = \hat{\xi}_{1,\ell}^{k-1}, \end{cases} \qquad \begin{cases} -\mathfrak{p}\hat{y}_{2,\ell}^{k,L} + \hat{\lambda}_{2,\ell}^{k,L} = \hat{\xi}_{2,\ell}^{k-1}, \\ \hat{y}_{2,\ell}^{k,M} = \hat{y}_{\mathrm{tar},\ell}. \end{cases} \tag{14}$$

Then,

$$\hat{\xi}_{1,\ell}^{k} = \mathfrak{p}\hat{y}_{2,\ell}^{k,L} + \hat{\lambda}_{2,\ell}^{k,L}, \quad \hat{\xi}_{2}^{k} = -\mathfrak{p}\hat{y}_{1,\ell}^{k,L} + \hat{\lambda}_{1,\ell}^{k,L}. \tag{15}$$

As the problem is linear, the convergence analysis of the algorithm reduces to investigating the case $\hat{y}_{\mathrm{ini},\ell} = \hat{y}_{\mathrm{tar},\ell} = 0$, starting from given data $\hat{\xi}_{1,\ell}^0$ and $\hat{\xi}_{2,\ell}^0$. Eliminating $\hat{y}_{i,\ell}^{k,0}$ and $\hat{\lambda}_{i,\ell}^{k,0}$ by solving explicitly the recurrence equations (13)–(15), we see that $\hat{\xi}_{i,\ell}^{k+2}$ follows the geometric progression

$$\hat{\xi}_{i,\ell}^{k+2} = \rho_{\Delta t}(\mathfrak{p}, \ell)\, \hat{\xi}_{i,\ell}^{k} \quad \text{with} \quad \rho_{\Delta t}(\mathfrak{p}, \ell) = \left(\frac{1 - \mathfrak{p}\gamma_{\Delta t}(\ell)}{1 + \mathfrak{p}\gamma_{\Delta t}(\ell)}\right)\left(\frac{|\beta_{\Delta t}(\ell)|^2 - \mathfrak{p}\gamma_{\Delta t}(\ell)}{|\beta_{\Delta t}(\ell)|^2 + \mathfrak{p}\gamma_{\Delta t}(\ell)}\right),$$

where $\beta_{\Delta t}(\ell) = (1 - \sigma(\ell)\Delta t)^L$, and $\gamma_{\Delta t}(\ell) = \Delta t \sum_{m=0}^{L-1} |1 - \sigma(\ell)\Delta t|^{2m}$. As in [2], our objective is to minimize $|\rho_{\Delta t}|$ uniformly in ℓ, namely, to solve the problem

$$\min_{\mathfrak{p}>0} \left(\max_{\ell=0,\ldots,N-1} |\rho_{\Delta t}(\mathfrak{p}, \ell)| \right). \tag{16}$$

To analyse (16), and in view of Theorem 1, we first make the change of variables $p = \mathfrak{p}\Delta T$. Then, under the assumption (10), we see that

$$|1 - \sigma(\ell)\Delta t|^2 = 1 - 4 \cdot \frac{\Delta t}{\Delta x}\left(1 - \frac{\Delta t}{\Delta x}\right) \sin^2(\pi \ell \Delta x) = 1 - 4r(1-r)\sin^2\left(\pi \ell \frac{\Delta t}{r}\right).$$

It motivates us to introduce the new variable

$$z = 4r(1-r)\sin^2\left(\pi\ell\frac{\Delta t}{r}\right),$$

which varies between 0 and $z_{\max} = 4r(1-r)$ (take $\ell = N/2$) as ℓ varies from 0 to $N-1$. For the sake of simplicity, we choose to optimize $\rho_{\Delta t}$ over the whole interval $[0, z_{\max}]$ and to study

$$\min_{p>0}\left(\max_{0\le z\le z_{\max}}|\rho_{\Delta t}(p,z)|\right) \quad \rho_{\Delta t}(p,z) = \frac{\varphi_{\Delta t}(z)-p}{\varphi_{\Delta t}(z)+p}\cdot\frac{\psi_{\Delta t}(z)-p}{\psi_{\Delta t}(z)+p} \tag{17}$$

with

$$\varphi_{\Delta t}(z) = \frac{\Delta T}{\gamma_{\Delta t}(z)}, \quad \psi_{\Delta t}(z) = \frac{|\beta_{\Delta t}(z)|^2\Delta T}{\gamma_{\Delta t}(z)},$$

and $|\beta_{\Delta t}(z)|^2 = (1-z)^L$, $\gamma_{\Delta t}(z) = \Delta t\sum_{m=0}^{L-1}(1-z)^m$.

4 Existence, uniqueness and asymptotic study of the optimized parameter

The following theorem proves the well-posedness of the problem (17) and describes the asymptotic behaviour of the optimal convergence factor as Δt goes to 0.

Theorem 2 *For any $\Delta t > 0$, Problem* (17) *has a unique solution $p^*_{\Delta t}$, which is the unique solution larger than* 1 *of the following alternation equation*

$$\max_{0\le z\le z_{\max}}\rho_{\Delta t}(p,z) = -\min_{0\le z\le z_{\max}}\rho_{\Delta t}(p,z). \tag{18}$$

Moreover, as Δt goes to 0*,*

$$p^*_{\Delta t} = \sqrt{2\Delta T\, z_{\max}}\,\Delta t^{-1/2} + o\left(\Delta t^{-1/2}\right), \tag{19}$$

$$\max_{0\le z\le z_{\max}}|\rho_{\Delta t}(p^*_{\Delta t},z)| = 1 - \frac{2\sqrt{2}}{\sqrt{\Delta T\, z_{\max}}}\,\Delta t^{1/2} + o\left(\Delta t^{1/2}\right). \tag{20}$$

Remark 2 In (19)-(20), $o(\Delta t^s)$ (with $s = \pm 1/2$) means that the remainder is negligible relative to Δt^s. We also point out that, unless $r = 1$ (in which case the scheme is exact), we have $\lim_{\Delta t\to 0} p^*_{\Delta t} \neq 1$, meaning we do not recover the optimal parameter associated with the continuous DD algorithm.

The remainder of this section is dedicated to the sketch of the proof of Theorem 2.

Step 1: *We prove that the alternation Equation* (18) *has a unique solution* $p^*_{\Delta t}$ *larger than 1.* Let us introduce

$$\rho_{\Delta t,\max}(p) = \max_{0\le z\le z_{\max}} \rho_{\Delta t}(p,z), \quad \rho_{\Delta t,\min}(p) = \min_{0\le z\le z_{\max}} \rho_{\Delta t}(p,z),$$

and the function $s(p) = \rho_{\Delta t,\max}(p) + \rho_{\Delta t,\min}(p)$. We prove that s has a unique zero larger than one (and, consequently (18) has a unique root). Indeed,

- For $p > 1$, the function s is a continuous and strictly increasing function of p. In fact, for $p > 1$, a direct computation shows that $\partial_p \rho_{\Delta t}(p,z) > 0$. Therefore, $\rho_{\Delta t,\max}$, $\rho_{\Delta t,\min}$, and their sum s are strictly increasing functions of p.
- $s(1) < 0$ ($\rho_{\Delta t,\max}(1) \le 0$ and $\rho_{\Delta t,\min}(1) < 0$).
- $s(\varphi_{\Delta t}(z_{\max})) > 0$ ($\rho_{\Delta t,\max}(\varphi_{\Delta t}(z_{\max})) > 0$ and $\rho_{\Delta t,\min}(\varphi_{\Delta t}(z_{\max})) = 0$).

Thus, (18) has a unique solution $p^*_{\Delta t} > 1$.

Step 2: *We show that* $p^*_{\Delta t}$ *is the unique solution to Problem* (17). First, based on the properties of $\varphi_{\Delta t}$ and $\psi_{\Delta t}$, we can prove (by contradiction) that any solution p of (17) must be in the interval $(1, \varphi_{\Delta t}(z_{\max}))$. But,

- For $p \in (1, p^*_{\Delta t})$, a careful investigation leads to

$$\max_{0\le z\le z_{\max}} |\rho_{\Delta t}(p,z)| = -\rho_{\Delta t,\min}(p) > -\rho_{\Delta t,\min}(p^*_{\Delta t}) = \max_{0\le z\le z_{\max}} |\rho_{\Delta t}(p^*_{\Delta t},z)|.$$

- Similarly, for $p \in (p^*_{\Delta t}, \varphi_{\Delta t}(z_{\max}))$, we obtain

$$\max_{0\le z\le z_{\max}} |\rho_{\Delta t}(p,z)| = \rho_{\Delta t,\max}(p) > \rho_{\Delta t,\max}(p^*_{\Delta t}) = \max_{0\le z\le z_{\max}} |\rho_{\Delta t}(p^*_{\Delta t},z)|.$$

Therefore, $p^*_{\Delta t}$ is the unique global minimum of (17).

Step 3: *Asymptotics of the optimal parameter* $p^*_{\Delta t}$ *and its corresponding convergence factor with respect to* Δt. We first remark that Equation (18) is defined implicitly in p, so it is a priori difficult to tackle directly. However, we can approximate $\rho_{\Delta t,\max}(p)$ by $\rho_{\Delta t}(p,0)$: indeed, an attentive analysis shows that there exists $\Delta t_0 > 0$ and a constant C such that for $\Delta t < \Delta t_0$,

$$|\rho_{\Delta t,\max}(p) - \rho_{\Delta t}(p,0)| \le C p^{-1} \Delta t. \tag{21}$$

Consequently, for small Δt, it is sufficient to consider the 'approximate' equation

$$\rho_{\Delta t}(p,0) = -\rho_{\Delta t}(p, z_{\max}), \tag{22}$$

which turns out to be explicitly solvable. Its solution $p^*_{\mathrm{eq},\Delta t}$ is given by

$$p^*_{\mathrm{eq},\Delta t} = \left(S_{m,\Delta t} - \frac{P_{m,\Delta t}}{2} - \frac{1}{2} + \left(\left(S_{m,\Delta t} - \frac{P_{m,\Delta t}}{2} - \frac{1}{2} \right)^2 - P_{m,\Delta t} \right)^{1/2} \right)^{1/2},$$

where $S_{m,\Delta t} = \psi_{\Delta t}(z_{\max}) + \varphi_{\Delta t}(z_{\max})$ and $P_{m,\Delta t} = \psi_{\Delta t}(z_{\max})\varphi_{\Delta t}(z_{\max})$. From the asymptotic behaviour of $\psi_{\Delta t}(z_{\max})$ and $\varphi_{\Delta t}(z_{\max})$, we deduce that when $\Delta t \to 0$,

$$p^*_{\mathrm{eq},\Delta t} = \sqrt{2\Delta T\ z_{\max}}.\Delta t^{-1/2} + o\left(\Delta t^{-1/2}\right),$$

which implies

$$-\rho_{\Delta t}(p^*_{\mathrm{eq},\Delta t}, z_{\max}) = \rho_{\Delta t}(p^*_{\mathrm{eq},\Delta t}, 0) = 1 - \frac{2\sqrt{2}}{\sqrt{\Delta T\ z_{\max}}} \cdot \Delta t^{1/2} + o\left(\Delta t^{1/2}\right).$$

Finally, the asymptotic formulas (19)–(20) result from (21).

5 Numerical illustration

We illustrate the results of Theorem 2 in the case of $T = 1$. In the left panel of Figure 1, we plot $1 - |\rho_{\Delta t}|_{\max}(p^*_{\Delta t})$ with respect to Δt (in logarithmic scale) for three different values of r. In each case, the optimized parameter $p^*_{\Delta t}$ is computed using `fminsearch` in Matlab. As expected, whatever the choice of $r \in (0, 1)$, we obtain straight lines with slope equal to that of the curve $y = \sqrt{\Delta t}$.

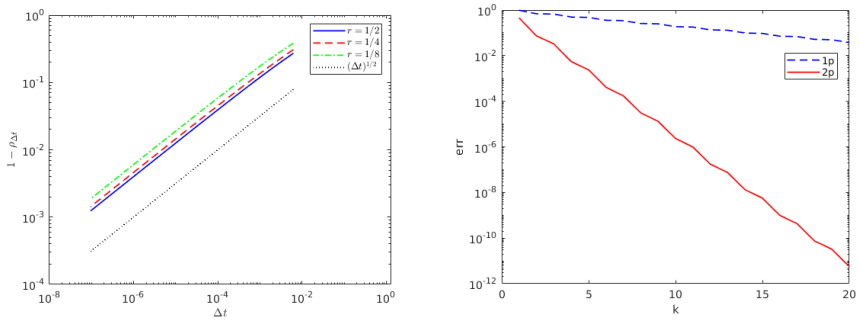


Fig. 1 Left: Asymptotic behaviour of $1 - |\rho_{\Delta t}|_{\max}(p^*_{\Delta t})$. Right: performance of $p^*_{\Delta t}$ for $\Delta t = 1/160$, $r = 1/2$.

Next, we test the performance of our domain decomposition-in-time algorithm. For the simulation, we take $\Delta t = 1/160$, $r = 1/2$, $y_{\mathrm{ini}} = y_{\mathrm{tar}} = 0$, and we start from a random initial guess $\boldsymbol{\xi}_i^0$ (i.e. we compute the zero solution). In the right panel of Figure 1, we display in blue the evolution of the error with respect to the number of iterations; in the present case, it just consists of computing the maximum of the L^2 norm of $\boldsymbol{\xi}_1^k$ and $\boldsymbol{\xi}_2^k$. The performance is as predicted by the theory. On the other hand, the convergence rate can be drastically improved by using a two-sided algorithm, where we allow for two different values p and q instead of $\mathfrak{p}$

in the formulas (5). The `fminsearch` function provides us with two optimized parameters $(p^*_{\Delta t}, q^*_{\Delta t}) = (1.1831, 8.5024 \times 10^{-2})$, leading to a convergence factor of 7.0728×10^{-2}. The performance of the two-sided algorithm for this value is displayed in red, and appears to be much better than the optimized one-sided one. The proof of that result will be given in a forthcoming publication.

Acknowledgements The work described in this paper is partially supported by a grant from the ANR/RGC joint research scheme sponsored by the Research Grants Council of the Hong Kong Special Administrative Region, China (Project no. A-CityU203/19) and the French National Research Agency (Project ALLOWAPP, grant ANR-19-CE46-0013-01).

References

1. Barker, A. and Stoll, M. Domain decomposition in time for pde-constrained optimization. *Comput. Phys. Commun.* **197**, 136–143 (2015).
2. Bennequin, D., Gander, M. J., and Halpern, L. A homographic best approximation problem with application to optimized Schwarz waveform relaxation. *Math. Comp.* **78**(265), 185–223 (2009).
3. Gander, M., Kwok, F., and Wanner, G. Constrained optimization: From lagrangian mechanics to optimal control and pde constraints. *Lecture Notes in Computational Science and Engineering* **101**, 151–202 (2014).
4. Gander, M. J. and Kwok, F. Schwarz methods for the time-parallel solution of parabolic control problems. In: *Domain Decomposition Methods in Science and Engineering XXII*, 207–216. Springer (2016).
5. Gander, M. J., Kwok, F., and Salomon, J. Paraopt: A parareal algorithm for optimality systems. *SIAM Journal on Scientific Computing* **42**(5), A2773–A2802 (2020).
6. Krug, R., Leugering, G., Martin, A., Schmidt, M., and Weninger, D. Time-domain decomposition for optimal control problems governed by semilinear hyperbolic systems. *SIAM Journal on Control and Optimization* **59**(6), 4339–4372 (2021).
7. Kwok, F. On the time-domain decomposition of parabolic optimal control problems. In: *Domain Decomposition Methods in Science and Engineering XXIII*, 55–67. Springer (2017).

An Overlapping Preconditioner for 2D Virtual Problems Posed in H(rot) with Irregular Subdomains

Juan G. Calvo, César Herrera, and Filánder A. Sequeira

1 Introduction

Given a bounded polygonal domain $\Omega \subset \mathbb{R}^2$, we seek $\boldsymbol{u} \in H_0(\mathrm{rot};\Omega)$ such that

$$a(\boldsymbol{u},\boldsymbol{v}) := \int_\Omega (\alpha \ \mathrm{rot}\ \boldsymbol{u}\ \mathrm{rot}\ \boldsymbol{v} + \beta \boldsymbol{u}\cdot\boldsymbol{v}) = \int_\Omega \boldsymbol{f}\cdot\boldsymbol{v} \quad \forall \boldsymbol{v} \in H_0(\mathrm{rot};\Omega), \tag{1}$$

where $\mathrm{rot}\ \boldsymbol{u} := \partial_{x_1}u_2 - \partial_{x_2}u_1$, $\boldsymbol{f} \in [L^2(\Omega)]^2$, and $\alpha,\beta \in L^\infty(\Omega)$ are positive functions that are uniformly bounded from below. The weak form (1) arises from implicit time integration of the eddy current model of Maxwell's equation [5] and is considered in several studies; see, e.g., [1, 13]. We recall that

$$H_0(\mathrm{rot};\Omega) := \left\{\boldsymbol{v} \in [L^2(\Omega)]^2 : \mathrm{rot}\,\boldsymbol{v} \in L^2(\Omega),\ \boldsymbol{v}\cdot\boldsymbol{t} = 0 \text{ on } \partial\Omega\right\},$$

where $\boldsymbol{t}$ denotes the unit tangential vector on $\partial\Omega$. The bilinear form $a(\cdot,\cdot)$ defined in (1) is obtained from the differential operator $\mathcal{L}\boldsymbol{u} := \mathbf{rot}\ (\alpha \mathrm{rot}\ \boldsymbol{u}) + \beta\boldsymbol{u}$, where $\mathbf{rot}\ q := \left(\partial_{x_2}q, -\partial_{x_1}q\right)^T$. The well-posedness of problem (1) can be established by a straightforward application of the Lax-Milgram lemma; for the sake of brevity we omit further details and refer to [15].

In this paper, we present a two-level overlapping Schwarz preconditioner for problem (1) discretized with finite or virtual element methods (FEM or VEM, respectively) in two dimensions. To the best of our knowledge, there are no theoretical

Juan G. Calvo
CIMPA – Escuela de Matemática, Universidad de Costa Rica, Costa Rica,
e-mail: juan.calvo@ucr.ac.cr

César Herrera
Purdue University, West Lafayette, IN 47907, e-mail: herre125@purdue.edu

Filánder A. Sequeira
Escuela de Matemática, Universidad Nacional, Heredia, Costa Rica,
e-mail: filander.sequeira@una.cr

results for preconditioning the linear system that arises from (1) when VEM are used. Our method allows us to handle irregular subdomains and general polygonal meshes, and applies to a broader range of material properties and subdomain geometries than previous studies.

First studies for problems posed in $H^1(\Omega)$ with FEM discretizations and irregular subdomains include [14, 17, 11], where discrete harmonic extensions are required for the construction of a coarse component of the preconditioner; for problems posed in $H(\mathrm{rot};\Omega)$ see [7, 8]. Such algorithms require us to solve a linear system on the fine mesh for each coarse function. The ideas introduced in [9, 10] allowed to extend standard Domain Decomposition Methods (DDM) from FEM to VEM for problems posed in $H^1(\Omega)$ in a natural way. Hence, we replace harmonic extensions by projectors onto polynomial spaces of degree at most k. In this variant, we need to solve a linear system with just $O(k^2)$ unknowns in order to construct a coarse function, reducing the complexity of the construction of coarse functions while preserving the dimension of the coarse space defined in [7] for FEM, which is equal to the number of interior subdomain edges. In this paper, we present such generalization for problems posed in $H(\mathrm{rot};\Omega)$.

In [7], a theoretical bound for the condition number κ of a two-level overlapping Schwarz preconditioner for FEM, based on discrete harmonic extensions, is given by

$$\kappa \leq C\left(1+\frac{H}{\delta}\right)\left(1+\log\frac{H}{h}\right),$$

where C only depends on α, β and some parameters related to the regularity of the subdomains. We observe similar results for our preconditioner when VEM and harmonic extensions are considered.

We remark that there are different DDM such as FETI-DP and BDDC methods; see [12, 8] for studies related to our problem. Nevertheless, the simplicity of implementing an overlapping additive Schwarz algorithm with competitive results gives relevance to our work.

The rest of this paper is organized as follows. We briefly describe the VEM for our model problem (1) in Section 2. We then describe the two-level overlapping additive Schwarz and the definition of our coarse space with detail in Section 3. Finally, some numerical results and conclusions are included in Section 4.

2 The virtual element method

We briefly describe a virtual element scheme for problem (1). Given an integer $\ell \geq 0$, let $\mathbb{P}_\ell(\mathcal{D})$ denote the space of polynomials defined in $\mathcal{D}$ of total degree at most ℓ. Let $\{\mathcal{T}_h\}_{h>0}$ be a family of decompositions of Ω into polygonal elements. We assume that there exists a constant $C_{\mathcal{T}} > 0$ such that for each decomposition $\mathcal{T}_h$ and for each $E \in \mathcal{T}_h$ it holds that (see, e.g., [6, Section 3.2] and [4, Section 2]):

1. the ratio between the shortest edge and the diameter h_E is bigger than $C_{\mathcal{T}}$, and
2. E is star-shaped with respect to a ball of radius $C_{\mathcal{T}} h_E$ and center $\boldsymbol{x}_E \in E$.

The lowest-order conforming Nédélec first-type local space

$$N_0^E := \{\boldsymbol{v} \in [\mathbb{P}_1(E)]^2 : \boldsymbol{v} = (-bx_2 + a_1, bx_1 + a_2)^T, a_1, a_2, b \in \mathbb{R}\}$$

is typically used for the discretization of (1) with triangular meshes; see [12, 7, 8]. For general polygonal meshes, we replace the Nédélec space N_0^E by the lowest-order local virtual element space W_0^E, defined as

$$W_0^E := \left\{\boldsymbol{v} \in [L^2(E)]^2 : \boldsymbol{v} \cdot \boldsymbol{t}|_e \in \mathbb{P}_0(e) \forall e \in \partial E, \operatorname{rot} \boldsymbol{v}, \operatorname{div} \boldsymbol{v} \in \mathbb{P}_0(E), \int_E \boldsymbol{v} \cdot \boldsymbol{x}_E = 0\right\}$$

where $e \in \partial E$ represents an edge of E, $\boldsymbol{x}_E = \boldsymbol{x} - \boldsymbol{b}_E$, and $\boldsymbol{b}_E$ is the barycenter of E; see [4, eq. (28)]. The degrees of freedom of a virtual function $\boldsymbol{v} \in W_0^E$ can be chosen as the moments $\lambda^e(\boldsymbol{v}) = \frac{1}{|e|}\int_e \boldsymbol{v} \cdot \boldsymbol{t}$ for each edge $e \in \partial E$, similar as what is done when Nédélec elements are used; see, e.g., [3, eq. (3.13)]. We remark that $\operatorname{rot} \boldsymbol{v} = \frac{1}{|E|}\int_E \operatorname{rot} \boldsymbol{v} = \frac{1}{|E|}\int_{\partial E} \boldsymbol{v} \cdot \boldsymbol{t} = \frac{1}{|E|}\sum_{e\in\partial E} |e|\lambda^e(\boldsymbol{v})$, and therefore we can compute the rotor of $\boldsymbol{v} \in W_0^E$ from its degrees of freedom. Thus, the term $\int_E \alpha \operatorname{rot} \boldsymbol{u} \operatorname{rot} \boldsymbol{v}$ of the bilinear form can be computed and we only require to modify the mass matrix. Given $\boldsymbol{v} \in [L^2(E)]^2$, let $\Pi_1^E : [L^2(E)]^2 \to [\mathbb{P}_1(E)]^2$ be the orhogonal projector given by

$$\int_E \Pi_1^E \boldsymbol{v} \cdot \boldsymbol{p} = \int_E \boldsymbol{v} \cdot \boldsymbol{p} \quad \forall \boldsymbol{p} \in [\mathbb{P}_1(E)]^2. \tag{2}$$

We remark that Π_1^E is computable for functions in W_0^E only knowing its degrees of freedom; we omit details and refer to [2, Remark 3]. For the mass-term, we then replace $\boldsymbol{v}$ by $\Pi_1^E \boldsymbol{v}$ in the local bilinear form. Therefore, as it is standard in VEM, a stabilizing term is required, which is defined as

$$s_E(\boldsymbol{w}, \boldsymbol{v}) := h_E \sum_{e\in\partial E} \int_e (\boldsymbol{w} \cdot \boldsymbol{t})(\boldsymbol{v} \cdot \boldsymbol{t}) \quad \forall \boldsymbol{w}, \boldsymbol{v} \in V_0(E);$$

see [4, Theorem A.2] and [3, eq. (4.8)] for further details. We then consider the local bilinear form

$$a_h^E(\boldsymbol{w}, \boldsymbol{v}) := \int_E \left(\alpha \operatorname{rot} \boldsymbol{w} \operatorname{rot} \boldsymbol{v} + \beta \Pi_1^E \boldsymbol{w} \cdot \Pi_1^E \boldsymbol{v}\right) + s_E\left(\boldsymbol{w} - \Pi_1^E \boldsymbol{w}, \boldsymbol{v} - \Pi_1^E \boldsymbol{v}\right)$$

for $\boldsymbol{w}, \boldsymbol{v} \in W_0^E$. The global virtual element space $V_h \subset H_0(\operatorname{rot}; \Omega)$ is then given by

$$V_h := \left\{\boldsymbol{v} \in H_0(\operatorname{rot}; \Omega) : \boldsymbol{v}|_E \in W_0^E \ \forall E \in \mathcal{T}_h\right\}, \tag{3}$$

and, as usual, the global bilinear form is obtained by assembling the local bilinear forms $a_h^E(\cdot, \cdot)$. We then define the virtual element scheme associated to (1): find $\boldsymbol{u}_h \in V_h$ such that

$$\sum_{E\in\mathcal{T}_h} a_h^E(\boldsymbol{u}_h,\boldsymbol{v}_h) = \sum_{E\in\mathcal{T}_h}\int_E \boldsymbol{f}\cdot\Pi_1^E\boldsymbol{v}_h \quad \forall\,\boldsymbol{v}_h\in V_h.$$

This problem is well-posed and standard estimates for the approximated solution can be obtained; for the sake of brevity we omit such details.

3 Overlapping Schwarz methods

In this section, we briefly describe two-level overlapping methods; see [16, Chapter 3] for further details. We partition the domain Ω into N non-overlapping subdomains $\{\Omega_i\}_{i=1}^N$ of diameter H_i which are the union of elements of $\mathcal{T}_h$. Subdomains are assumed to satisfy the same assumptions as the elements on the fine mesh; this implies that they are simply connected and the number of edges of each subdomain is uniformly bounded. The edges on this decomposition are denoted by e^H, which correspond to edges of the polygons Ω_i. We then construct overlapping subdomains $\Omega_i'\supset\Omega_i$ by adding layers of elements that are external to Ω_i, and we will denote by δ_i the minimum width of the region $\Omega_i'\setminus\Omega_i$.

We consider the usual local virtual spaces V_i, $1\le i\le N$, defined by

$$V_i := \left\{\boldsymbol{v}\in H_0(\mathrm{rot}\,;\Omega_i') : v|_E\in W_0^E\ \forall\, E\subset\Omega_i'\right\}.$$

Thus, the degrees of freedom of a function $\boldsymbol{v}_i\in V_i$ are $\lambda^e(\boldsymbol{v}_i)$ at the fine edges e that are in the interior of Ω_i'. We also consider the natural operators $R_i^T : V_i\to V_h$ given by the zero extension from the subdomain Ω_i' to Ω, $1\le i\le N$.

We can define the coarse space V_0 as the virtual element space (3) defined on the coarse mesh $\{\Omega_i\}_{i=1}^N$. Nevertheless, its dimension can be an inconvenience for parallel implementations in the presence of irregular subdomains with too many edges; see Figure 1. Instead, for each subdomain edge $\mathcal{E}^{ij}$ (defined as the interior of $\overline{\Omega}_i\cap\overline{\Omega}_j$), we define a coarse function $\boldsymbol{c}_{\mathcal{E}}\in V_0$ by defining its degrees of freedom of V_0. We set $\lambda^{e^H}(\boldsymbol{c}_{\mathcal{E}}) = \boldsymbol{d}_{\mathcal{E}}\cdot\boldsymbol{t}_{e^H}$ for every edge e^H in $\mathcal{E}$, and $\lambda^{e^H}(\boldsymbol{c}_{\mathcal{E}}) = 0$ otherwise. Here, $\boldsymbol{d}_{\mathcal{E}}$ denotes a unit vector in the direction between the endpoints of $\mathcal{E}$, and $\boldsymbol{t}_{e^H}$ is the unit tangent vector of e^H. The reduced coarse space V_0^R is then defined as the span of these coarse basis functions $\boldsymbol{c}_{\mathcal{E}}$. We remark that the dimension of V_0^R is equal to the number of subdomain edges, similar as in [7, 8, 12].

In order to define an operator $R_0^T : V_0^R\subseteq V_0\to V_h$ that approximates functions in the coarse space by elements in V_h, we can consider discrete harmonic extensions as in [7], for which a generalization for VEM can be established. Nevertheless, we can avoid discrete harmonic extensions by approximating virtual functions in V_0^R in the interior of subdomains by polynomials as follows. Consider the high-order virtual spaces of order $k\in\mathbb{N}$, defined on the coarse mesh, as the set

$$V_0^k = \{\boldsymbol{v}\in H_0(\mathrm{rot}\,;\Omega) : \boldsymbol{v}|_{\Omega_i}\in W_k^{\Omega_i}\ \forall i\in\{1,2,\ldots,N\}\},$$

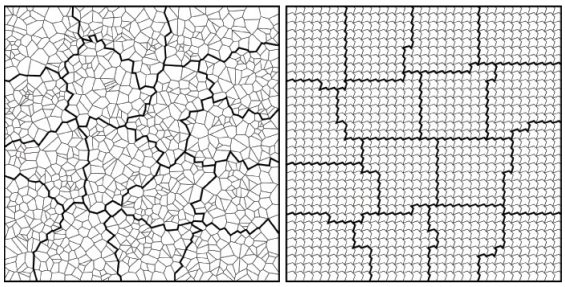


Fig. 1 (left) Voronoi mesh and (right) non-convex mesh with $N = 16$ irregular subdomains. Subdomains have, in average, 45 and 55 edges for the Voronoi and non-convex meshes, respectively.

where $W_k^{\Omega_i}$ is defined as the set

$$\left\{\boldsymbol{v} \in [L^2(\Omega_i)]^2 \colon \boldsymbol{v} \cdot \boldsymbol{t}|_{e^H} \in \mathbb{P}_k(e^H) \forall e^H \in \partial\Omega_i, \operatorname{rot} \boldsymbol{v} \in \mathbb{P}_k(\Omega_i), \operatorname{div} \boldsymbol{v} \in \mathbb{P}_{k-1}(\Omega_i)\right\}$$

Following [2, Section 3.2], the local degrees of freedom for $\boldsymbol{v} \in W_k^{\Omega_i}$ can be chosen as

$$\begin{aligned} m_q^{e^H}(\boldsymbol{v}) &:= \int_{e^H} (\boldsymbol{v} \cdot \boldsymbol{t})\, q \qquad \forall\, q \in \mathbb{P}_k(e^H)\,, \forall\, e^H \in \partial\Omega_i\,, \\ m_{p,\mathrm{rot}}^{\Omega_i}(\boldsymbol{v}) &:= \int_{\Omega_i} (\operatorname{rot} \boldsymbol{v})\, p \qquad \forall\, p \in \mathbb{P}_k(\Omega_i) \setminus \{1\}\,, \\ m_{p,\mathrm{div}}^{\Omega_i}(\boldsymbol{v}) &:= \int_{\Omega_i} (\boldsymbol{v} \cdot \boldsymbol{x}_{\Omega_i})\, p \qquad \forall\, p \in \mathbb{P}_{k-1}(\Omega_i)\,, \end{aligned}$$

which are unisolvent; see [2, Proposition 3.3]. It is clear that $V_0^R \subseteq V_0 \subseteq V_0^k$. For every coarse function $\boldsymbol{c}_{\mathcal{E}^{ij}} \in V_0^R$, we seek the degrees of freedom of the function $\widetilde{\boldsymbol{c}}_{\mathcal{E}^{ij}} \in V_0^k$, with the same degrees of freedom of $\boldsymbol{c}_{\mathcal{E}}$ on the interface, such that

$$\sum_{E \in \Omega_l} a_h^E(I^h(\Pi_k^{\Omega_l}\widetilde{\boldsymbol{c}}_{\mathcal{E}^{ij}}), I^h(\Pi_k^{\Omega_l}\widetilde{\boldsymbol{c}}_{\mathcal{E}^{ij}})) \tag{4}$$

is minimum for $l \in \{i, j\}$, where $\boldsymbol{w} = I^h\boldsymbol{v} \in V_h$ is the usual interpolant given by the condition $\lambda^e(\boldsymbol{v} - \boldsymbol{w}) = 0$ for all edge e, and $\Pi_k^{\Omega_l} : [L^2(\Omega_l)]^2 \to [\mathbb{P}_k(\Omega_l)]^2$ is the orthogonal projector onto Ω_l; see (2) for the case $k = 1$. The degrees of freedom of $\widetilde{\boldsymbol{c}}_{\mathcal{E}}$ given by $m_q^{e^H}(\widetilde{\boldsymbol{c}}_{\mathcal{E}})$ and $m_{p,\mathrm{rot}}^{\Omega_i}(\widetilde{\boldsymbol{c}}_{\mathcal{E}})$ are known and can be computed from $\boldsymbol{c}_{\mathcal{E}}$. Since $\int_{\Omega_i}(\boldsymbol{v} \cdot \boldsymbol{x}_{\Omega_i}) = 0$, the remaining degrees of freedom can be obtained just by solving a linear system with $k(k+1)/2 - 1$ equations for each subdomain with $\mathcal{E}$ on its boundary, obtained by directly computing the critical points of (4). For the sake of brevity we omit details and refer to [9] that includes how to obtain this linear system. Preserving degrees of freedom on the interface guarantees continuity across the interface when we interpolate coarse functions to the fine mesh. We then define $R_0^T\boldsymbol{c}_{\mathcal{E}} \in V_h$ by setting:

(a') $\lambda^e(R_0^T\boldsymbol{c}_{\mathcal{E}}) = \lambda^e(\boldsymbol{c}_{\mathcal{E}})$ if e is an edge on the interface;
(b') $\lambda^e(R_0^T\boldsymbol{c}_{\mathcal{E}}) = \lambda^e(\Pi_k^{\Omega_l}\widetilde{\boldsymbol{c}}_{\mathcal{E}})$ if e is an interior edge of Ω_l;
(c') $\lambda^e(R_0^T\boldsymbol{c}_{\mathcal{E}}) = 0$ otherwise;

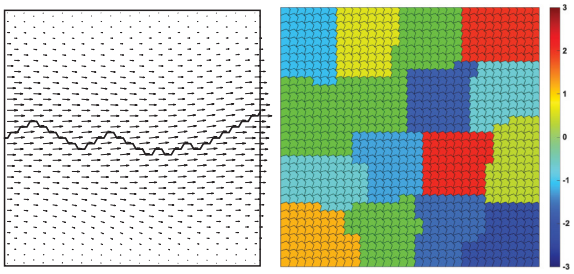


Fig. 2 (left) $R_0^T \boldsymbol{c}_{\mathcal{E}}$ for an irregular edge $\mathcal{E}$, evaluated in the interior of each subdomain by interpolating $\Pi_6^{\Omega_i}$ for $i \in \{1, 2\}$. (right) A discontinuous coefficient β varying from $\beta = 10^3$ (red) to $\beta = 10^{-3}$ (blue).

see Figure 2 where we show $R_0^T \boldsymbol{c}_{\mathcal{E}}$ for a given subdomain edge $\mathcal{E}$. We finally consider the two-level additive overlapping Schwarz preconditioner

$$P_{ad} := \sum_{i=0}^{N} P_i = A_{ad}^{-1} A, \text{ with } A_{ad}^{-1} = \sum_{i=0}^{N} R_i^T (R_i A R_i^T)^{-1} R_i, \tag{5}$$

where we consider exact solvers for each subspace for simplicity; see [16, Chap. 2].

4 Numerical results and conclusions

We present numerical results for the two-level additive overlapping Schwarz preconditioner (5). We solve the resulting linear systems using the preconditioned conjugate gradient method to a relative residual tolerance of 10^{-6}. We estimate the condition number $\kappa(P_{ad})$ and compute the number of iterations I_k (for spaces of degree k) and $I_{\mathcal{H}}$ (for the coarse space based on discrete harmonic extensions)

Table 1 Number of iterations I and condition number κ (in parenthesis) with Voronoi meshes and N METIS subdomains. I_3, I_6 and $I_{\mathcal{H}}$ correspond to $k = 3$, $k = 6$ and discrete harmonic extensions, respectively. $N_{\mathcal{E}}$ is the dimension of the coarse space.

		$\beta = 10^{-3}$			$\beta = 1$			$\beta = 10^{3}$		
	$N_{\mathcal{E}}$	$I_3\,(\kappa)$	$I_6\,(\kappa)$	$I_{\mathcal{H}}\,(\kappa)$	$I_3\,(\kappa)$	$I_6\,(\kappa)$	$I_{\mathcal{H}}\,(\kappa)$	$I_3\,(\kappa)$	$I_6\,(\kappa)$	$I_{\mathcal{H}}\,(\kappa)$
N		Test 1: $H/h = 8$, $H/\delta = 2$, $\alpha = 1$								
8^2	161	38 (47.5)	31 (23.9)	21 (7.7)	35 (38.5)	22 (10.5)	20 (7.2)	18 (7.3)	18 (7.3)	18 (7.3)
12^2	389	58 (112)	49 (79.0)	24 (9.4)	55 (95.7)	33 (20.3)	21 (8.0)	20 (8.3)	19 (8.2)	19 (8.2)
16^2	709	73 (201)	66 (181)	23 (9.3)	69 (163)	51 (51.6)	21 (8.0)	20 (7.9)	20 (7.9)	20 (7.9)
20^2	1128	90 (328)	84 (289)	25 (9.9)	84 (262)	68 (132)	22 (8.4)	20 (8.4)	20 (8.0)	20 (8.0)
H/δ		Test 2: $H/h = 32$, $N = 16$, $\alpha = 1$								
4	33	33 (25.4)	30 (24.1)	19 (6.1)	31 (22.8)	28 (19.8)	18 (6.0)	16 (5.1)	15 (5.1)	15 (5.1)
8	33	43 (58.3)	39 (52.3)	21 (7.5)	40 (49.5)	37 (35.6)	20 (7.0)	15 (4.8)	15 (4.9)	15 (5.0)
16	33	57 (136)	56 (124)	24 (12.7)	53 (100)	49 (65.7)	23 (13.0)	16 (6.1)	16 (6.1)	16 (6.0)
32	33	82 (300)	81 (300)	35 (28.6)	73 (166)	62 (101)	32 (23.0)	20 (8.6)	20 (8.4)	19 (7.8)
H/h		Test 3: $N = 16$, $H/\delta = 4$, $\alpha = 1$								
8	33	32 (32.6)	24 (12.4)	19 (7.2)	30 (22.4)	19 (6.9)	18 (7.0)	14 (5.4)	14 (5.3)	14 (5.3)
16	34	31 (26.8)	29 (26.1)	18 (6.1)	30 (22.9)	27 (14.1)	18 (5.8)	15 (5.1)	15 (5.0)	15 (5.1)
32	33	33 (25.4)	30 (24.1)	19 (6.1)	31 (22.8)	28 (19.8)	18 (6.0)	16 (5.1)	15 (5.1)	15 (5.1)
64	33	34 (22.6)	32 (22.6)	18 (5.5)	32 (21.6)	31 (19.3)	18 (5.5)	14 (5.3)	16 (5.1)	16 (5.1)

Table 2 Number of iterations I and condition number κ (in parenthesis) with non-convex meshes and N METIS subdomains. I_3, I_6 and $I_{\mathcal{H}}$ correspond to $k = 3, k = 6$ and discrete harmonic extensions, respectively. $N_{\mathcal{E}}$ is the dimension of the coarse space.

		$\beta = 10^{-3}$			$\beta = 1$			$\beta = 10^{3}$		
N	$N_{\mathcal{E}}$	$I_3\,(\kappa)$	$I_6\,(\kappa)$	$I_{\mathcal{H}}\,(\kappa)$	$I_3\,(\kappa)$	$I_6\,(\kappa)$	$I_{\mathcal{H}}\,(\kappa)$	$I_3\,(\kappa)$	$I_6\,(\kappa)$	$I_{\mathcal{H}}\,(\kappa)$
		Test 1: $H/h = 8$, $H/\delta = 2$, $\alpha = 1$								
8^2	158	38 (54.0)	28 (17.3)	20 (8.0)	36 (36.9)	21 (7.8)	19 (6.7)	20 (10.1)	20 (10.1)	20 (10.1)
12^2	379	57 (130)	45 (66.1)	21 (7.4)	53 (81.6)	29 (13.7)	20 (7.1)	23 (13.5)	23 (13.6)	22 (13.5)
16^2	699	76 (261)	61 (136)	21 (7.7)	69 (139)	34 (21.2)	20 (7.2)	24 (16.4)	24 (16.1)	24 (16.2)
20^2	1109	93 (537)	82 (265)	23 (8.3)	83 (245)	45 (38.3)	21 (8.5)	27 (20.9)	27 (20.7)	27 (20.9)
H/δ		Test 2: $H/h = 32$, $N = 16$, $\alpha = 1$								
4	33	33 (30.7)	30 (28.7)	18 (6.4)	31 (24.0)	29 (17.7)	17 (5.5)	15 (5.1)	15 (5.1)	15 (5.0)
8	33	43 (80.4)	42 (73.7)	22 (8.2)	42 (49.5)	35 (29.8)	20 (7.9)	17 (8.0)	17 (8.0)	16 (7.9)
16	33	56 (129)	57 (139)	26 (11.4)	52 (70.3)	45 (41.4)	24 (12.0)	21 (12.8)	21 (12.8)	20 (12.7)
32	33	78 (297)	78 (304)	34 (21.9)	67 (112)	55 (62.7)	32 (20.6)	30 (28.4)	30 (28.4)	29 (29.4)
H/h		Test 3: $N = 16$, $H/\delta = 4$, $\alpha = 1$								
8	33	31 (29.6)	23 (10.7)	18 (6.4)	27 (19.5)	18 (6.4)	17 (5.7)	19 (11.4)	19 (11.3)	19 (11.7)
16	31	31 (30.8)	29 (27.7)	18 (6.0)	29 (20.2)	23 (10.0)	17 (5.8)	16 (7.7)	16 (7.7)	16 (7.7)
32	33	33 (30.7)	30 (28.7)	18 (6.4)	31 (24.0)	29 (17.7)	17 (5.5)	15 (5.0)	15 (5.1)	15 (5.0)
64	33	38 (39.5)	33 (29.8)	19 (6.1)	35 (25.2)	31 (20.9)	18 (5.7)	16 (4.9)	16 (5.1)	15 (5.0)

Table 3 Number of iterations I and condition number κ (in parenthesis) with non-convex meshes and discontinuous values for β as in Figure 2. I_6 and $I_{\mathcal{H}}$ correspond to $k = 6$ and discrete harmonic extensions, respectively. $N_{\mathcal{E}}$ is the dimension of the coarse space.

N	$N_{\mathcal{E}}$	$I_6\ (\kappa)$	$I_{\mathcal{H}}\ (\kappa)$
8^2	158	20 (10.1)	20 (10.1)
12^2	379	23 (13.6)	22 (13.5)
16^2	699	24 (16.1)	24 (16.2)
20^2	1109	27 (20.7)	27 (20.9)

H/h	$N_{\mathcal{E}}$	$I_6\ (\kappa)$	$I_{\mathcal{H}}\ (\kappa)$
8	33	19 (11.3)	19 (11.7)
16	31	16 (7.7)	16 (7.7)
32	33	15 (5.1)	15 (5.0)
64	33	16 (5.1)	15 (5.0)

for each experiment; see results in Tables 1 and 2. We include different values for $\beta \in \{10^{-3}, 1, 10^3\}$ since previous bounds depend on the parameters α and β. We confirm the linear growth in the condition number as we increase H/δ and we observe no significant dependence on the parameter H/h. We observe that the coarse space based on discrete harmonic extensions is numerically scalable, and for small values of β the scalability is impaired when polynomial spaces are used. We remark that for the case of triangular meshes and square subdomains, our method recovers the same spaces as in [7]. We also include numerical results where β is piecewise constant on each subdomain; see Table 3 and Figure 2.

The theoretical bound for the condition number of the preconditioned system is in progress, where we have been able to obtain certain bounds for the coarse component of a decomposition for $\boldsymbol{u} \in V_h$, without considering Helmholtz decompositions as in [7]. There is also interest of implementing these ideas in 3D problems, in order to compare numerical results and running times with previous preconditioners. We also remark that similar results will hold for two-dimensional problems posed in $H(\mathrm{div}; \Omega)$, since two-dimensional Raviart-Thomas elements correspond to a 90° rotation of the elements considered in this paper.

Acknowledgements The authors gratefully acknowledges the institutional support for the project B8290 subscribed to *Vicerrectoría de Investigación, Universidad de Costa Rica*, and *Universidad Nacional, Costa Rica*, through the project 0140-20.

References

1. Beck, R., Hiptmair, R., Hoppe, R. H. W., and Wohlmuth, B. Residual based a posteriori error estimators for eddy current computation. *ESAIM: Math. Model. Numer. Anal.* **34**, 159–182 (2000).
2. Beirão da Veiga, L., Brezzi, F., Dassi, F., Marini, L., and Russo, A. Virtual element approximation of 2D magnetostatic problems. *Comput. Methods Appl. Mech. Eng.* **327**, 173–195 (2017).
3. Beirão da Veiga, L., Brezzi, F., Dassi, F., Marini, L., and Russo, A. Lowest order virtual element approximation of magnetostatic problems. *Comput. Methods Appl. Mech. Eng.* **332**, 343–362 (2018).
4. Beirão da Veiga, L. and Mascotto, L. Interpolation and stability properties of low-order face and edge virtual element spaces. *IMA J. Numer. Anal.* (2022).
5. Bossavit, A. Discretization of electromagnetic problems: The generalized finite differences approach. *Handb. Numer. Anal.* **13**, 105–197 (2005).
6. Cáceres, E. and Gatica, G. N. A mixed virtual element method for the pseudostress-velocity formulation of the Stokes problem. *IMA J. Numer. Anal.* **37**(1), 296–331 (2017).
7. Calvo, J. G. A two-level overlapping Schwarz method for $H(\mathrm{curl})$ in two dimensions with irregular subdomains. *Electron. Trans. Numer. Anal.* **44**, 497–521 (2015).
8. Calvo, J. G. A BDDC algorithm with deluxe scaling for $H(\mathrm{curl})$ in two dimensions with irregular subdomains. *Math. Comp.* **85**(299), 1085–1111 (2016).
9. Calvo, J. G. On the approximation of a virtual coarse space for domain decomposition methods in two dimensions. *Math. Models Methods Appl. Sci.* **28**(7), 1267–1289 (2018).
10. Calvo, J. G. An overlapping Schwarz method for virtual element discretizations in two dimensions. *Comput. Math. Appl.* **77**(4), 1163–1177 (2019).
11. Dohrmann, C. and Widlund, O. An alternative coarse space for irregular subdomains and an overlapping Schwarz algorithm for scalar elliptic problems in the plane. *SIAM J. Numer. Anal.* **50**, 2522–2537 (2012).
12. Dohrmann, C. R. and Widlund, O. B. An iterative substructuring algorithm for two-dimensional problems in $H(\mathrm{curl})$. *SIAM J. Numer. Anal.* **50**(3), 1004–1028 (2012).
13. Hiptmair, R. and Xu, J. Nodal auxiliary space preconditioning in $H(\mathrm{curl})$ and $H(\mathrm{div})$ spaces. *SIAM J. Numer. Anal.* **45**, 2483–2509 (2007).
14. Klawonn, A., Rheinbach, O., and Widlund, O. An analysis of a FETI-DP algorithm on irregular subdomains in the plane. *SIAM J. Numer. Anal.* **46**, 2484–2504 (2008).
15. Schöberl, J. Numerical methods for Maxwell equations. https://www.asc.tuwien.ac.at/ schoeberl/wiki/lva/notes/maxwell.pdf (2009). Accessed: 2022-11-22.
16. Toselli, A. and Widlund, O. *Domain decomposition methods-algorithms and theory*, *Springer Ser. Comput. Math.*, vol. 34. Springer (2005).
17. Widlund, O. Accommodating irregular subdomains in domain decomposition theory. In: Bercovier, M., Gander, M. J., Kornhuber, R., and Widlund, O. (eds.), *Domain Decomposition Methods in Science and Engineering XVIII*, *Lecture Notes in Computational Science and Engineering*, vol. 70, 87–98. Springer-Verlag (2009).

A Two-Level Restricted Additive Schwarz Method for Asynchronous Computations

Faycal Chaouqui and Daniel B. Szyld

1 Introduction

In this paper, we investigate the parallel performance of both synchronous and asynchronous domain decomposition methods (DDMs) for the solution of algebraic systems coming from the discretization of partial differential equations (PDEs). In particular, we extend the ideas introduced in [8] for different types of coarse space corrections. We consider a PDE of the form $L(u) = f$ on $\Omega \subset \mathbb{R}^2$ such that $u|_{\Omega} = 0$. The operator L after discretization yields a large sparse system of algebraic equations of the form

$$A\mathbf{u} = \mathbf{f}, \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$ and $\mathbf{f} \in \mathbb{R}^n$. Here, we focus our attention on the Restricted Additive Schwarz (RAS) domain decomposition solver [2, 4]. For the sake of simplicity, we assume that $L = -\Delta$. We assume that the domain Ω is decomposed into p overlapping subdomains $\Omega_1, \ldots, \Omega_p$. Let $R_i^\top$, $i = 1, \ldots, p$, denotes the boolean matrix that maps the local degrees of freedom defined in Ω_i to Ω. We define the local stiffness matrix $A_i = R_i A R_i^\top$. Let us also define the diagonal matrices D_i, $i = 1, \ldots p$, such that we satisfy the partition of unity, i.e., $\sum_{i=1}^{p} R_i^\top D_i R_i = I$, where I denotes the identity matrix in $\mathbb{R}^{n\times n}$. The RAS iteration is then defined as

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \sum_{i=1}^{p} R_i^\top D_i A_i^{-1} R_i(\mathbf{f} - A\mathbf{u}^k). \tag{2}$$

We note that in our case, the matrices D_i correspond to diagonal boolean matrices that are 1 in the non-overlapping partition, and 0 otherwise. We note also that there

Faycal Chaouqui
COMSOL, INC, Burlington, Mass., USA, e-mail: chaouqui@temple.edu

Daniel Szyld
Temple University, Philadelphia, Pa., USA, e-mail: szyld@temple.edu

are other ways for choosing those matrices, and we refer the reader to [4, 5]. In the next section, we will describe briefly the asynchronous RAS method.

2 Asynchronous restricted additive Schwarz

We briefly describe asynchronous iterations (see, e.g., [3]) for fixed point problems defined on a product space $U = U_1 \times \cdots \times U_p$, of the form $\mathbf{u} = \mathcal{T}\mathbf{u}$ with a unique solution. In other words, we have $\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_p)$ and $\mathcal{T} = (\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_p)$, with $\mathcal{T}_s \colon U \to U_s$. We have in mind that the operation in process s, of the form $\mathbf{u}_s = \mathcal{T}_s(\mathbf{u}_1, \ldots, \mathbf{u}_p)$ is performed without synchronization, i.e., without waiting for other processors to send new information.

For a mathematical model of these asynchronous iterations on p processors, we follow the model introduced by Bertsekas [1]. To that end, we define a time stamp k, $k \in \mathbb{N}$, and denote by $\{\sigma(k)\}_{k\in\mathbb{N}}$ the sequence of non-empty subsets of $\{1, \ldots, p\}$, defining which processes update their components at the time stamp k. Define also for $s, q \in \{1, \ldots, p\}$, $\left\{\tau_q^s(k)\right\}_{k\in\mathbb{N}}$ a sequence of integers, representing the update number (or time stamp) of the data coming from process q and available on process s at the time k. Thus, a *delay* would be $k - \tau_q^s(k)$. We begin with an initial approximation $\mathbf{u}^0 = \left(\mathbf{u}_1^0, \ldots, \mathbf{u}_p^0\right)$, and define, for each process s, the asynchronous iterations as follows.

$$\mathbf{u}_s^{k+1} = \begin{cases} \mathcal{T}_s\left(\mathbf{u}_1^{\tau_1^s(k)}, \ldots, \mathbf{u}_p^{\tau_p^s(k))}\right) & \text{if } s \in \sigma(k+1), \\ \mathbf{u}_s^k & \text{if } s \notin \sigma(k+1)\,. \end{cases} \tag{3}$$

In this model, one also assumes that the three following natural conditions are satisfied

$$\forall s, q \in \{1, \ldots, p\}, \forall k \in \mathbb{N}, \tau_q^s(k) \leq k, \tag{4}$$

$$\forall s \in \{1, \ldots, p\}, \text{card}\,\{k \in \mathbb{N} | s \in \sigma(k)\} = +\infty, \tag{5}$$

$$\forall s, q \in \{1, \ldots, p\}, \lim_{k\to+\infty} \tau_q^s(k) = +\infty. \tag{6}$$

Condition (4) represents the fact that data used at the time k must have been produced before time k, i.e., time does not flow backward. Condition (5) indicates that no process will ever stop updating its components. Condition (6) means that new data will always be provided to the process. In other words, no process will have a piece of data that is never updated.

One important theoretical result states that for a fixed point problem, say $T(\mathbf{u}) = \mathbf{u}$, on a product space, under conditions (4)–(6), if there is a norm such that the map T is contracting, i.e., if the (synchronous) fixed point iteration converges, then, the corresponding asynchronous iteration converges as well; see, e.g., [3] and references therein. For the RAS iteration, the map $\mathcal{T}_s$ defined in (3) is equivalent to

Algorithm 1 (Asynchronous RAS)

Input: $\mathbf{u}^0$.
Output: $\mathbf{u} \approx \mathbf{u}^*$.
Set $\mathbf{r}^0 = \mathbf{f} - A\mathbf{u}^0$, converged = false.
In parallel, each processor core s:
while converged = false **do**
    Set $\mathbf{u}_s = \mathcal{T}_s(\mathbf{u}_1, \ldots, \mathbf{u}_p)$ ▷ Update subdomain s
    Compute $\|D_s\mathbf{r}_s\|_2$ ▷ Compute local residual norm
    if s == 1 **then**
        Compute $\|\mathbf{r}\|_2 = \sqrt{\sum_{i=1}^{p} \|D_i\mathbf{r}_i\|_2^2}$
        if $\|\mathbf{r}\|_2/\|\mathbf{r}^0\|_2 \leq \epsilon$ **then** ▷ Check global convergence
            converged = true
        end if
    end if
end while (for processor s)
Set $\mathbf{u} = \sum_{s=1}^{p} R_s^\top D_s \mathbf{u}_s$ ▷ Assemble global solution

$$\mathcal{T}_s(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p) = \mathbf{u}_s + R_s \sum_{i=1}^{p} R_i^\top D_i A_i^{-1} \mathbf{r}_i, \tag{7}$$

where $\mathbf{r}_i = R_i(\mathbf{f} - A\mathbf{u})$ it the local residual for the subdomain Ω_i, $i = 1, \ldots p$. The implementation of iteration (3) is presented in Algorithm 1.

In Algorithm 1 each processor core computes and updates the components of the local vector as well as the corresponding local residual norms. A processor core is then in charge of accumulating all the local residuals and computing the global residual. The algorithm then stops when the global residual is smaller than the tolerance. We provide results of numerical examples illustrating the performance of both synchronous and asynchronous RAS. We consider $\Omega = [0, 1] \times [0, 1]$ decomposed into regular squares with a total of p subdomains and a minimal overlap. The source term $\mathbf{f}$ is chosen such that $\sin(\pi x)\sin(\pi y)$ corresponds to the exact monodomain solution. We partition the domain into $p = 4 \times 4$ subdomains with a total of 10k discretization points. We note that each processor core was assigned to one subdomain. All the tests were carried out on a shared memory machine which consists of 88 CPU cores / 176 threads and 1536GB of RAM. The implementation of Algorithm 1 was in C++ and the parallelization uses the OpenMP multithreading directives. We run two different types of experiments. In the first run, we assume all processors run at the same speed and compare both the timings required by both synchronous and asynchronous to reach a specified tolerance. This is illustrated in Figure 1 (left). We can see that in this case the synchronous is faster than the asynchronous. To show the advantage of the asynchronous approach, we repeat the experiment but with one processor core twice as slow. This can be realized by measuring the time needed for a single update and then forcing the processor to sleep (idle) for that amount of time. In this manner we mimic heterogeneous architectures, as well as cases where one subdomain is larger than the others. We can observe from Figure 1 (right) that the asynchronous is faster than the synchronous in this case.

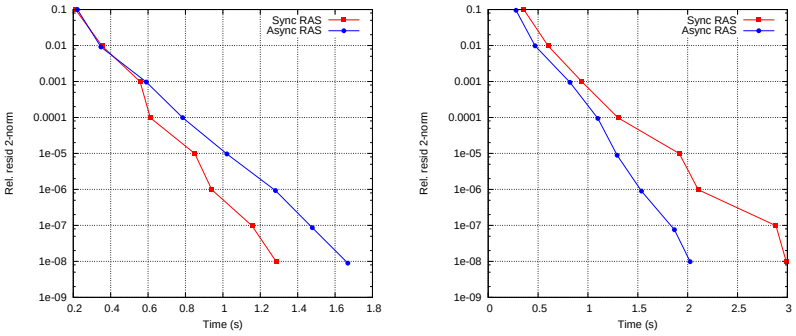


Fig. 1 Left: CPU time versus relative residual 2-norm for synchronous and asynchronous RAS with $p = 16$. Right: Same but one thread is twice as slow.

3 Two-level asynchronous restricted additive Schwarz

A second level is an essential component to obtain a robust domain decomposition method. It relies generally on solving a smaller problem on a coarser mesh so that there is global communication between the subdomains. The coarse space allows us then to construct the coarse restriction matrix R_0. The two-level RAS is then defined as

$$\begin{aligned} \mathbf{u}^{k+1/2} &= \mathbf{u}^k + \sum_{i=1}^{p} R_i^\top D_i A_i^{-1} R_i(\mathbf{f} - A\mathbf{u}^k) \\ \mathbf{u}^{k+1} &= \mathbf{u}^{k+1/2} + R_0^\top A_0^{-1} R_0(\mathbf{f} - A\mathbf{u}^{k+1/2}). \end{aligned} \tag{8}$$

In order to use iteration (8) asynchronously, we need to use the coarse grid in an additive way. This can be done by using a weighted additive version or multiplicative/additive variant of (8). The corresponding two-level mapping $\tilde{\mathcal{T}}$ can be expressed in the case of the additive variant as

$$\tilde{\mathcal{T}}_s(\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_p) = \mathbf{u}_s + R_s \left(\frac{1}{2} \sum_{i=1}^{p} R_i^\top D_i A_i^{-1} \mathbf{r}_i + \frac{1}{2} R_0^\top A_0^{-1} R_0 \sum_{i=1}^{p} R_i^\top D_i \mathbf{r}_i \right). \tag{9}$$

For work using a multiplicative additive variant, we mention [7, 9]. To avoid over-correction from the coarse grid, we have to make sure that no subdomain is corrected again until all the remaining subdomains have updated at least once [8]. We present in Algorithm 2, the implementation of the asynchronous two-level RAS.

We describe now the coarse correction we use. We would like the coarse grid to ensure the scalability of the method as a solver. In the same spirit of [6], we use harmonically extended coarse basis functions. Let n_i denotes the number of cross points for each subdomain $\Omega_i, i = 1, \ldots p$. Let φ_i^j, $j = 1, \ldots n_i$ define a piecewise

Algorithm 2 (Asynchronous two-level RAS)

Input: $\mathbf{u}^0$.
Output: $\mathbf{u} \approx \mathbf{u}^*$.
Set $\mathbf{r}^0 = \mathbf{f} - A\mathbf{u}^0$, converged = false.
Set update[s]=false, and correction[s]=false, $s = 1, \ldots, p$.
In parallel, each processor s:
while converged = false **do**
  if $s > 0$ **then**
    if correction[s] **then** ▷ Check if coarse correction is needed
      Set $\mathbf{u}_s = \widetilde{\mathcal{T}}_s(\mathbf{u}_1, \ldots, \mathbf{u}_p)$ ▷ Update subdomain s
      Set correction[s]=false
    else
      Set $\mathbf{u}_s = \mathcal{T}_s(\mathbf{u}_1, \ldots, \mathbf{u}_p)$ ▷ Update subdomain s
      Set update[s]=true
    end if
    Compute $\|D_s\mathbf{r}_s\|_2$ ▷ Compute local residual norm
    if s == 1 **then**
      Compute $\|\mathbf{r}\|_2 = \sqrt{\sum_{i=1}^{p} \|D_i\mathbf{r}_i\|_2^2}$
      if $\|\mathbf{r}\|_2/\|\mathbf{r}^0\|_2 \leq \epsilon$ **then** ▷ Check global convergence
        converged = true
      end if
    end if
  else
    if update[q], $\forall q = 1, \ldots, p$ **then** ▷ Check if all subdomains updated
      Compute the coarse correction.
      Set correction[i]=true, $i = 1, \ldots, p$
      Set update[i]=false, $i = 1, \ldots, p$
    end if
  end if
end while (**for processor** s)
Set $\mathbf{u} = \sum_{s=1}^{p} R_s^\top D_s \mathbf{u}_s$ ▷ Assemble global solution

linear function on $\partial\Omega_i$ that is 1 at one cross point and 0 on the others. We define the coarse basis functions ϕ_i^j, $j = 1, \ldots, n_i, i = 1, \ldots p$, as the solution of

$$\begin{cases} L|_{\Omega_i}(\phi_i^j) = 0, \text{on } \Omega_i \\ \qquad \phi_i^j = \varphi_i^j, \text{ on } \partial\Omega_i \ . \end{cases} \tag{10}$$

We define our coarse space $\mathcal{Z} \subset \mathbb{R}^d$ as the span of extended coarse functions ϕ_i^j, i.e.,

$$\mathcal{Z} = \text{span}\left\{ R_i^\top \phi_i^j, j = 1, \ldots, n_i \, i = 1, \ldots, p \right\}. \tag{11}$$

The columns of the matrix $R_0^\top$ forms a basis of $\mathcal{Z}$. We show in Table 1 the number of iterations needed to reach a tolerance $\epsilon = 10^{-8}$ with this specific coarse space for (synchronous) RAS as a solver. We can see that the two-level method outperforms the one-level method and is also scalable, i.e., the number of iterations does not increase when we grow the number of subdomains. We also report the iterations required

Table 1 Weak scalability of additive two-level RAS

p	n	$\dim \mathcal{Z}$	$\dim \mathcal{Z}_{MG}$	#iter (RAS)	#iter (RAS+$\mathcal{Z}$)	#iter (RAS+$\mathcal{Z}_{MG}$)
16	14400	36	49	630	93	144
25	22500	64	81	953	98	107
36	32400	100	121	1344	99	108
49	44100	144	169	1802	100	99
64	57600	196	225	2325	100	99

for two-level RAS constructed using a multigrid (MG) approach with four levels of coarsening. We can observe that the coarse grid considered in our simulation is asymptotically similar to MG for our model problem L. However, it has a smaller coarse grid size.

Next, we test the performance of the two-level asynchronous Algorithm 2 by comparing the time needed to reach a specified tolerance. In Figure 2 we plot the timing versus the residual norm for both synchronous and asynchronous two-level methods. We can observe that in this case the synchronous is faster than the asynchronous. We also see that the timing required to converge is faster than for the one-level method. The introduction of heterogeneity among processors yields a faster asynchronous two-level method. We note that as is the case for the local subdomains, the coarse problem was solved exactly since it is small for the coarse space defined (11). In Table 2, we report the timings required for both synchronous and asynchronous one and two-level RAS for processors with random time delays. We realize this by adding a random time delay to each processor core that follows a uniform density function of the form $\mathcal{U}(0, \varepsilon T_s)$, where T_s is the timings required for the processor s to finish its workload, and $\varepsilon = 0.01, 0.1, 1$. We can observe from Table 2 that the introduction of heterogeneities in the computation, even with a small magnitude reveals the advantages of asynchronous computations.

In Figure 3 we test the weak scalability of both the synchronous and asynchronous methods. To do so, we fix the tolerance to $\epsilon = 10^{-6}$ and the subdomain's size to 1600,

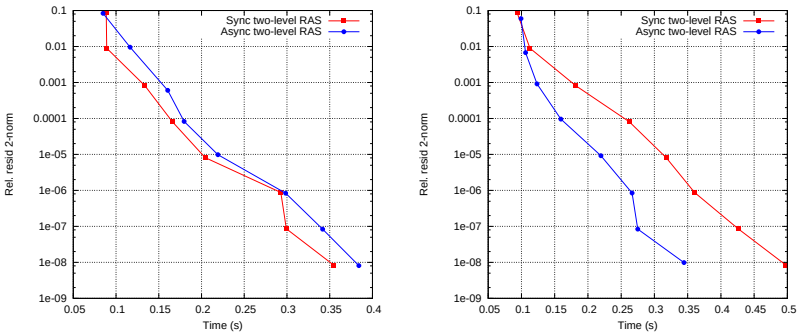


Fig. 2 Left: CPU time versus relative residual 2-norm for two-level synchronous and asynchronous RAS with $p = 16$. Right: Same but one thread is twice as slow.

Table 2 Timing required (in sec) of synchronous and asynchronous one- and two-level RAS to reach a tolerance of 10^{-8} for different levels of heterogeneities.

ε	Sync RAS	Async RAS	Sync two-level RAS	Async two-level RAS
0.01	2.3173	2.1973	0.4547	0.3539
0.1	2.4788	2.4051	0.4075	0.3601
1	5.9415	5.7202	1.0837	1.0295

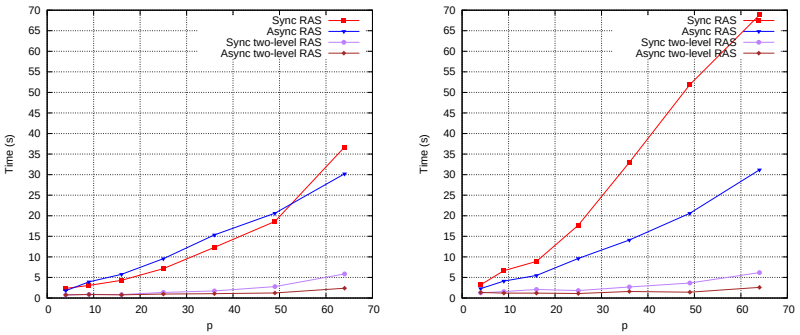


Fig. 3 Left: The number of subdomains versus the CPU time needed for convergence for one and two-level synchronous and asynchronous RAS. Right: Same but with a processor core twice as slow.

then run the two-level algorithms and measure the CPU time required to converge. We also plot the time required for the synchronous one as well. In Figure 3 (left), all the processors run at the same speed and there is no load imbalance. We can observe that in this case, the two-level asynchronous method is the fastest among all the four methods. The one-level synchronous method is still slightly faster than the one-level asynchronous (except for $p = 64$). In Figure 3 (right) we repeat the same experiment, but with one processor core twice as slow. We can see now that the asynchronous method outperforms the synchronous method. This is true for both the one- and two-level methods. Observe also that while the two-level synchronous method is slightly slower in the simulated heterogeneous architecture (for $p = 64$, 5.85 sec vs. 2.38 sec), the asynchronous method is faster (2.59 sec vs. 6.17 sec). The introduction of heterogeneity clearly shows how asynchronous can be effective in practice.

4 Conclusion

In this paper, we analyzed the performance of one and two-level synchronous and asynchronous RAS. In particular, we used a specific coarse grid correction for our asynchronous computations. Our numerical results suggest that the asynchronous methods exhibit good performance. In particular, we observed that for heterogeneous hardware, the asynchronous outperforms the synchronous method. This was valid for both the one and two-level methods.

References

1. Bertsekas, D. P. Distributed asynchronous computation of fixed points. *Mathematical Programming* **27**(1), 107–120 (1983).
2. Cai, X.-C. and Sarkis, M. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing* **21**, 792–797 (1999).
3. Frommer, A. and Szyld, D. B. On Asynchronous Iterations. *Journal of Computational and Applied Mathematics* **123**, 201–216 (2000).
4. Frommer, A. and Szyld, D. B. An algebraic convergence theory for restricted additive Schwarz methods using weighted max norms. *SIAM Journal on Numerical Analysis* **39**, 463–479 (2001).
5. Gander, M. J. Does the Partition of Unity Influence the Convergence of Schwarz Methods? In: Haynes, R., MacLachlan, S., Cai, X.-C., Halpern, L., Kim, H., Klawonn, A., and Widlund, O. (eds.), *Domain decomposition methods in science and engineering XXV*, *Lecture Notes in Computational Science and Engineering*, vol. 138, 3–15. Springer, Cham (2018).
6. Gander, M. J. and Loneland, A. SHEM: An optimal coarse space for RAS and its multiscale approximation. In: Lee, C.-O., Cai, X.-C., Keyes, D., Kim, H., Klawonn, A., Park, E.-J., and Widlund, O. (eds.), *Domain decomposition methods in science and engineering XXIII*, *Lecture Notes in Computational Science and Engineering*, vol. 116, 313–321. Springer (2017).
7. Gbikpi-Benissan, G. and Magoulès, F. Asynchronous Multiplicative Coarse-Space Correction. *SIAM Journal on Scientific Computing* **44**, C237–C259 (2022).
8. Glusa, C., Boman, E. G., Chow, E., Rajamanickam, S., and Szyld, D. B. Scalable Asynchronous Domain Decomposition Solvers. *SIAM Journal on Scientific Computing* **42**, C384–C409 (2020).
9. Wolfson-Pou, J. and Chow, E. Asynchronous multigrid methods. In: *2019 IEEE international parallel and distributed processing symposium (IPDPS)*, 101–110. IEEE (2019).

Cross-Points in the Neumann-Neumann Method

Bastien Chaudet-Dumas and Martin J. Gander

1 Introduction

The Neumann-Neumann method (NNM), first introduced in [1] in the case of two subdomains, is among the most popular non-overlapping domain decomposition methods. However, when used as a stationary solver at the continuous level, it has been observed that the method faced well-posedness issues in the presence of cross-points, see [2]. Here, our goal is to analyze in detail the behaviour of the NNM near cross-points on a simple, but rather instructive, bidimensional configuration.

Let $\Omega \subset \mathbb{R}^2$ be the square $(-1,1) \times (-1,1)$, divided into four non-overlapping square subdomains Ω_i, $i \in \mathcal{I} := \{1,2,3,4\}$, see Figure 1. This leads to one interior cross-point (red dot), and four boundary cross-points (black dots). We denote the interfaces between adjacent subdomains by $\Gamma_{ij} := \text{int}(\partial\Omega_i \cap \partial\Omega_j)$, the skeleton of the partition by $\Gamma := \bigcup_{i,j} \overline{\Gamma}_{ij}$, and $\partial\Omega_i^0 := \partial\Omega_i \cap \partial\Omega$. We consider the Laplace problem with Dirichlet boundary conditions on Ω, that is: find u solution to

$$-\Delta u = f \ \text{ in } \Omega, \quad u = g \ \text{ on } \partial\Omega, \tag{1}$$

where $f \in L^2(\Omega)$ and $g \in H^{\frac{3}{2}}(\partial\Omega)$, ensuring that $u \in H^2(\Omega)$.

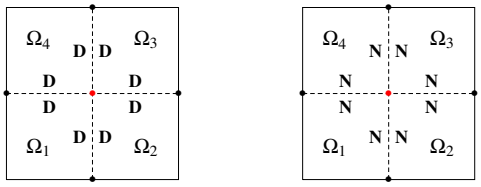


Fig. 1 Transmission conditions of the standard NNM for u (left) and ψ (right).

Bastien Chaudet-Dumas, Martin J. Gander
University of Geneva, Switzerland e-mail: bastien.chaudet@unige.ch, martin.gander@unige.ch

Given an initial couple (u^0, ψ^0), and a relaxation parameter $\theta \in \mathbb{R}$, each iteration $k \geq 1$ of the NNM applied to (1) can be split into two steps:

- *(Dirichlet step)* Solve for all $i \in \mathcal{I}$,

$$\begin{aligned} -\Delta u_i^k &= f \ \text{ in } \Omega_i\,, \quad u_i^k = g \ \text{ on } \partial\Omega_i^0\,, \\ u_i^k &= u_i^{k-1} - \theta\left(\psi_i^{k-1} + \psi_j^{k-1}\right) \ \text{ on } \Gamma_{ij},\ \forall j \in \mathcal{I} \ \text{ s.t. } \Gamma_{ij} \neq \emptyset\,. \end{aligned}$$

- *(Neumann step)* Compute the correction ψ^k, that is, solve for all $i \in \mathcal{I}$,

$$\begin{aligned} -\Delta \psi_i^k &= 0 \ \text{ in } \Omega_i\,, \quad \psi_i^k = 0 \ \text{ on } \partial\Omega_i^0\,, \\ \partial_{n_i}\psi_i^k &= \partial_{n_i} u_i^k + \partial_{n_j} u_j^k \ \text{ on } \Gamma_{ij},\ \forall j \in \mathcal{I} \ \text{ s.t. } \Gamma_{ij} \neq \emptyset\,. \end{aligned}$$

For the method to be well defined, it is assumed in the rest of this paper that the initial couple (u^0, ψ^0) is compatible with the Dirichlet boundary condition, i.e. it satisfies: $u^0 \in H^2(\Omega)$, $\psi^0 \in H^2(\Omega) \cap H_0^1(\Omega)$ and $u^0 \mid_{\partial\Omega\cap\Gamma} = g \mid_\Gamma$.

2 Convergence analysis of the Neumann-Neumann method

Definition 1 A measurable function $h : \Omega \to \mathbb{R}$ is said to be *even symmetric* (resp. *odd symmetric*) if for a.e. $(x, y) \in \Omega$, $h(-x, -y) = h(x, y)$ (resp. $-h(x, y)$). Moreover, any measurable function h can be uniquely decomposed into $h = h_e + h_o$ where h_e is even symmetric and h_o is odd symmetric.

Following this notion, as in [3], we introduce the so-called *even symmetric* and *odd symmetric parts* of problem (1): find u_e and u_o solutions to

$$-\Delta u_e = f_e \ \text{ in } \Omega, \quad u_e = g_e \ \text{ on } \partial\Omega, \tag{2a}$$

$$-\Delta u_o = f_o \ \text{ in } \Omega, \quad u_o = g_o \ \text{ on } \partial\Omega. \tag{2b}$$

If u denotes the solution to (1), it is known (see [3]) that the unique solutions u_e and u_o to these subproblems are precisely the even symmetric part and the odd symmetric part of u. In what follows, we will perform the convergence analysis of the NNM separately for the errors associated with the even and odd symmetric subproblems, as they lead to completely different behaviours of the method.

Case of the even symmetric part. The next Theorem states that the NNM is convergent when applied to the even symmetric part of (1).

Theorem 1 *Taking (u_e^0, ψ_e^0) as initial couple for the NNM applied to* (2a) *produces a sequence $\left\{u_e^k\right\}_k$ that converges geometrically to the solution u_e with respect to the L^2-norm and the broken H^1-norm for any $\theta \in (0, \frac{1}{2})$. Moreover, the convergence factor is given by $|1 - 4\theta|$, which also proves that the method becomes a direct solver for the specific choice $\theta = \frac{1}{4}$.*

Proof As in [3] for the Dirichlet-Neumann method, let us study the first iterations of the NNM in terms of the local errors $e^k_{e,i} := u_e|_{\Omega_i} - u^k_{e,i}$.
• *Iteration $k = 1$, Dirichlet step:* In each Ω_i, $i \in \mathcal{I}$, the errors satisfy

$$-\Delta e^1_{e,i} = 0 \ \text{ in } \Omega_i\,, \quad e^1_{e,i} = 0 \ \text{ on } \partial\Omega^0_i\,,$$
$$e^1_{e,i} = e^0_{e,i} + \theta\left(\psi^0_{e,i} + \psi^0_{e,j}\right) \ \text{ on } \Gamma_{ij},\ \forall j \in \mathcal{I} \ \text{ s.t. } \Gamma_{ij} \neq \emptyset\,.$$

Since (u^0_e, ψ^0_e) is compatible with the even symmetric part of the Dirichlet boundary condition, $e^1_{e,i}$ exists and is unique in $H^1(\Omega_i)$. Using the even symmetry properties of e^0_e and ψ^0_e, one can deduce that the $e^1_{e,i}$, for $i \in \{2,3,4\}$, can be expressed in terms of $e^1_{e,1}$ as follows:

$$e^1_{e,2}(x,y) = e^1_{e,1}(-x,y)\,, \qquad \text{for a.e. } (x,y) \in \Omega_2\,,$$
$$e^1_{e,3}(x,y) = e^1_{e,1}(-x,-y)\,, \qquad \text{for a.e. } (x,y) \in \Omega_3\,,$$
$$e^1_{e,4}(x,y) = e^1_{e,1}(x,-y)\,, \qquad \text{for a.e. } (x,y) \in \Omega_4\,.$$

• *Iteration $k = 1$, Neumann step:* We compute the correction $\psi^1_{e,i}$ in each subdomain Ω_i. For instance, taking $i = 1$, we get in Ω_1

$$-\Delta\psi^1_{e,1} = 0 \ \text{ in } \Omega_1\,, \quad \psi^1_{e,1} = 0 \ \text{ on } \Gamma_1\,,$$
$$\partial_{n_1}\psi^1_{e,1} = -\left(\partial_{n_1}e^1_{e,1} + \partial_{n_2}e^1_{e,2}\right) = -2\partial_{n_1}e^1_{e,1} \ \text{ on } \Gamma_{12}\,,$$
$$\partial_{n_1}\psi^1_{e,1} = -\left(\partial_{n_1}e^1_{e,1} + \partial_{n_4}e^1_{e,4}\right) = -2\partial_{n_1}e^1_{e,1} \ \text{ on } \Gamma_{41}\,.$$

Thus, uniqueness of $\psi^1_{e,1}$ in $H^1(\Omega_1)$ yields $\psi^1_{e,1} = -2e^1_{e,1}$ in Ω_1. A similar reasoning applies to each $\psi^1_{e,i}$, $i \in \{2,3,4\}$, therefore the recombined correction simply reads: $\psi^1_e = -2e^1_e$ in $\Omega \setminus \Gamma$.
• *Iteration $k \geq 2$:* At iteration $k = 2$, the transmission condition for the Dirichlet step in Ω_i on each Γ_{ij} is given by, $e^2_{e,i} = e^1_{e,i} + \theta\left(\psi^1_{e,i} + \psi^1_{e,j}\right) = (1-4\theta)e^1_{e,i}$. Uniqueness of $e^2_{e,i}$ in $H^1(\Omega_i)$ enables us to conclude that $e^2_{e,i} = (1-4\theta)e^1_{e,i}$ in Ω_i. Since this holds in each subdomain, the exact same reasoning as for iteration $k = 1$ applies, and we get after the Neumann step $e^2_e = (1-4\theta)e^1_e$ and $\psi^2_e = -2(1-4\theta)e^1_e$ in $\Omega \setminus \Gamma$. By induction, we obtain for any $k \geq 3$, $e^k_e = (1-4\theta)^{k-1}e^1_e$ in $\Omega \setminus \Gamma$. This leads to the following estimates for the error on the whole domain Ω in the L^2-norm and the broken H^1-norm:

$$\| u^k_e - u_e \|_{L^2(\Omega)} = \sum_{i\in\mathcal{I}} \| e^k_{e,i} \|_{L^2(\Omega_i)} \leq C|1-4\theta|^{k-1},$$
$$\sum_{i\in\mathcal{I}} \| u^k_{e,i} - u_{e,i} \|_{H^1(\Omega_i)} \leq C'|1-4\theta|^{k-1},$$

where C, C' are strictly positive constants depending on the data and the geometry of the domain decomposition. □

Case of the odd symmetric part. As for the Dirichlet-Neumann method, the NNM does not converge in general when applied to the odd symmetric part of (1).

Theorem 2 *The NNM applied to* (2b) *is not well-posed. More specifically, taking* (u_o^0, ψ_o^0) *as initial couple, there exists an integer* $k_0 > 0$ *such that the solution to the problem obtained at the* k_0*-th iteration is not unique. In addition, all possible solutions* $u_o^{k_0}$ *are singular at the cross-point, with a leading singularity of type* $(\ln r)^2$.

Theorem 3 *If we let the NNM go beyond the ill-posed iteration* k_0 *from Theorem 2, we end up with a sequence* $\{u_o^k\}_{k \geq k_0}$ *of non-unique iterates. Moreover, for each* $k \geq k_0$*, all possible* u_o^k *are singular at the cross-point, with a leading singularity of type* $(\ln r)^{2(k-k_0)+2}$.

Proof The proofs of these results rely on the exact same arguments as those in the proofs of [3, Theorem 7 and 8]. □

The previous results show that, at some point in the iterative process, the NNM method will lead to solving an ill-posed problem. This will generate a singular solution, and the generated singularity will then propagate through the following iterations.

3 Toward a modified Neumann-Neumann method

The conclusions from the previous section suggest that the transmission conditions of the standard NNM are naturally well adapted to the even symmetric part of the problem. Indeed, in this context, one may express at each iteration k all local errors $e_{e,i}^k$ in terms of only one, say $e_{e,1}^k$, by symmetry. This motivates the search for different transmission conditions such that a similar symmetry property holds for the odd symmetric part of the problem.

Fixing the odd symmetric case. In order to fix the well-posedness issue in the odd symmetric case, and obtain the symmetry property mentioned above, we propose a new distribution of Dirichlet and Neumann transmission conditions, as shown in Figure 2.

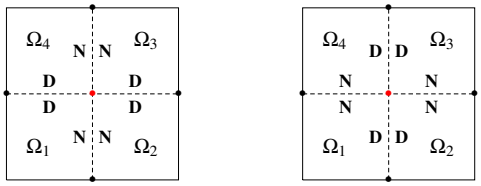


Fig. 2 Transmission conditions of the mixed NNM for u (left) and ψ (right).

Let us introduce $\Gamma_D^1, \Gamma_N^1, \Gamma_D^2, \Gamma_N^2$ the sets containing all parts of the interface Γ where transmission conditions of Dirichlet or Neumann type are imposed for u (superscript 1) and for ψ (superscript 2), that is :

$$\Gamma_D^1 := \{\Gamma_{23}, \Gamma_{41}\}, \quad \Gamma_N^1 := \{\Gamma_{12}, \Gamma_{34}\}, \quad \Gamma_D^2 := \{\Gamma_{12}, \Gamma_{34}\}, \quad \Gamma_N^2 := \{\Gamma_{23}, \Gamma_{41}\}.$$

Given an initial couple (u^0, ψ^0) and relaxation parameter θ, each iteration $k \geq 1$ of the proposed *mixed* Neumann-Neumann method can be split into two steps:

- *(First step)* Solve for all $i \in \mathcal{I}$

$$\begin{aligned}
-\Delta u_i^k &= f \ \text{ in } \Omega_i\,, \quad u_i^k = g \ \text{ on } \partial\Omega_i^0\,,\\
u_i^k &= u_i^{k-1} - \theta\left(\psi_i^{k-1} + \psi_j^{k-1}\right) \ \text{ on } \Gamma_{ij},\ \forall j \in \mathcal{I} \ \text{ s.t. } \Gamma_{ij} \in \Gamma_D^1\,,\\
\partial_{n_i} u_i^k &= \partial_{n_i} u_i^{k-1} + (-1)^i\theta\left(\partial_{n_i}\psi_i^{k-1} + \partial_{n_j}\psi_j^{k-1}\right) \ \text{ on } \Gamma_{ij},\ \forall j \in \mathcal{I} \ \text{ s.t. } \Gamma_{ij} \in \Gamma_N^1\,.
\end{aligned}$$

- *(Second step)* Compute the correction ψ^k, that is, solve for all $i \in \mathcal{I}$

$$\begin{aligned}
-\Delta\psi_i^k &= 0 \ \text{ in } \Omega_i\,, \quad \psi_i^k = 0 \ \text{ on } \partial\Omega_i^0\,,\\
\psi_i^k &= u_i^k - u_j^k \ \text{ on } \Gamma_{ij},\ \forall j \in \mathcal{I} \ \text{ s.t. } \Gamma_{ij} \in \Gamma_D^2\,,\\
\partial_{n_i}\psi_i^k &= \partial_{n_i} u_i^k + \partial_{n_j} u_j^k \ \text{ on } \Gamma_{ij},\ \forall j \in \mathcal{I} \ \text{ s.t. } \Gamma_{ij} \in \Gamma_N^2\,.
\end{aligned}$$

With this choice of transmission conditions, we are able to prove that the proposed mixed NNM is convergent when applied to the odd symmetric part of (1).

Theorem 4 *Taking (u_o^0, ψ_o^0) as initial couple for the mixed NNM applied to* (2b) *produces a sequence $\left\{u_o^k\right\}_k$ that converges geometrically to the solution u_o with respect to the L^2-norm and the broken H^1-norm for any $\theta \in (0, \frac{1}{2})$. Moreover, the convergence factor is given by $|1 - 4\theta|$, which also proves that the method becomes a direct solver for the specific choice $\theta = \frac{1}{4}$.*

Proof We follow the same steps as in the proof of Theorem 1.
• *Iteration $k = 1$, Dirichlet step:* In each Ω_i, $i \in \mathcal{I}$, the odd errors satisfy

$$\begin{aligned}
-\Delta e_{o,i}^1 &= 0 \ \text{ in } \Omega_i\,, \quad e_{o,i}^1 = 0 \ \text{ on } \partial\Omega_i^0\,,\\
e_{o,i}^1 &= e_{o,i}^0 + \theta\left(\psi_{o,i}^0 + \psi_{o,j}^0\right) \ \text{ on } \Gamma_{ij},\ \forall j \in \mathcal{I} \ \text{ s.t. } \Gamma_{ij} \in \Gamma_D^1\,,\\
\partial_{n_i} e_{o,i}^1 &= \partial_{n_i} e_{o,i}^0 - (-1)^i\theta\left(\partial_{n_i}\psi_{o,i}^0 + \partial_{n_j}\psi_{o,j}^0\right) \ \text{ on } \Gamma_{ij},\ \forall j \in \mathcal{I} \ \text{ s.t. } \Gamma_{ij} \in \Gamma_N^1\,.
\end{aligned}$$

These problems are well-posed since (u_o^0, ψ_o^0) is compatible with the odd symmetric part of the boundary condition. This time, using the mixed conditions enforced along Γ together with the odd symmetry properties of e_o^0 and ψ_o^0, we can deduce that

$$
\begin{aligned}
e^1_{o,2}(x,y) &= -e^1_{o,1}(-x,y)\,, \qquad \text{for a.e. } (x,y)\in\Omega_2\,,\\
e^1_{o,3}(x,y) &= -e^1_{o,1}(-x,-y)\,, \qquad \text{for a.e. } (x,y)\in\Omega_3\,,\\
e^1_{o,4}(x,y) &= e^1_{o,1}(x,-y)\,, \qquad \text{for a.e. } (x,y)\in\Omega_4\,.
\end{aligned}
$$

Indeed, for the first equality, taking $(x,y)\in\Omega_2$, we have on Γ_{23} and Γ_{12}

$$
\begin{aligned}
e^1_{o,2}(x,0) &= e^0_{o,2}(x,0) + \theta\left(\psi^0_{o,2}(x,0)+\psi^0_{o,3}(x,0)\right)\\
&= -e^0_{o,1}(-x,0) - \theta\left(\psi^0_{o,4}(-x,0)+\psi^0_{o,1}(-x,0)\right) = -e^1_{o,1}(-x,0)\,,\\
(\partial_{n_2}e^1_{o,2})(0,y) &= -(\partial_x e^0_{o,2})(0,y) - \theta\left((\partial_x\psi^0_{o,2})(0,y)+(\partial_x\psi^0_{o,1})(0,y)\right)\\
&= -(\partial_x e^0_{o,1})(0,y) - \theta\left((\partial_x\psi^0_{o,1})(0,y)+(\partial_x\psi^0_{o,2})(0,y)\right)\\
&= -(\partial_{n_1}e^1_{o,1})(0,y) = -(\partial_{n_2}e^1_{o,1}(-\cdot,\cdot))(0,y)\,.
\end{aligned}
$$

Then uniqueness of the solution to the subproblem in Ω_2 yields $e^1_{o,2} = -e^1_{o,1}(-\cdot,\cdot)$ a.e. in Ω_2. The two other equalities are obtained using similar arguments, see Figure 3 for an illustration of this symmetry property.

• *Iteration $k=1$, Neumann step:* For $i=1$, we get in Ω_1

$$
\begin{aligned}
-\Delta\psi^1_{o,1} &= 0 \ \text{ in } \Omega_1\,, \quad \psi^1_{o,1} = 0 \ \text{ on } \Gamma_1\,,\\
\psi^1_{o,1} &= -e^1_{o,1} + e^1_{o,2} = -2e^1_{o,1} \ \text{ on } \Gamma_{12}\,,\\
\partial_{n_1}\psi^1_{o,1} &= -\left(\partial_{n_1}e^1_{o,1} + \partial_{n_4}e^1_{o,4}\right) = -2\partial_{n_1}e^1_{o,1} \ \text{ on } \Gamma_{41}\,.
\end{aligned}
$$

Therefore, $\psi^1_{o,1} = -2e^1_{o,1}$ in Ω_1. Extending these arguments to the other subdomains yields a recombined correction $\psi^1_o = -2e^1_o$ in $\Omega\setminus\Gamma$.

• *Iteration $k\geq 2$:* At iteration $k=2$, the transmission conditions for the first step in Ω_1 are given by

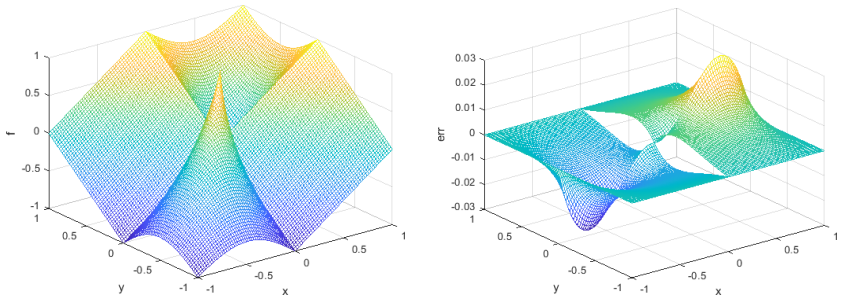


Fig. 3 Source term f (left), and absolute error at iteration 1 for $\theta = 0.25$ (right), in Example 2.

$$e^2_{o,1} = e^1_{o,1} + \theta\left(\psi^1_{o,1} + \psi^1_{o,4}\right) = (1-4\theta)e^1_{o,1} \text{ on } \Gamma_{41}\,,$$
$$\partial_{n_1} e^2_{o,1} = \partial_{n_1} e^1_{o,1} + \theta\left(\partial_{n_1}\psi^1_{o,1} + \partial_{n_2}\psi^1_{o,2}\right) = (1-4\theta)\partial_{n_1} e^1_{o,1} \text{ on } \Gamma_{12}\,.$$

This implies that $e^2_{o,1} = (1-4\theta)e^1_{o,1}$ in Ω_1. Using the same arguments in the other subdomains and performing the second step leads to $e^2_o = (1-4\theta)e^1_o$ and $\psi^2_o = -2(1-4\theta)e^1_o$ in $\Omega \setminus \Gamma$. As in the proof of Theorem 1, we obtain by induction that, for any $k \geq 3$, $e^k_o = (1-4\theta)^{k-1}e^1_o$ in $\Omega \setminus \Gamma$. The desired error estimates are then deduced from the last relation. □

The new NNM. Here are the different steps of our *new* NNM to solve (1) starting from an initial couple (u^0, ψ^0) compatible with the Dirichlet boundary condition, and a relaxation parameter $\theta \in (0, 1/2)$.

1. Decompose the data into their even/odd symmetric parts to get (2a) and (2b).
2. Solve in parallel:
   - (2a) using the standard NNM starting from (u^0_e, ψ^0_e),
   - (2b) using the mixed NNM starting from (u^0_o, ψ^0_o).
3. Recompose the solution $u = u_e + u_o$.

Remark 1 It is actually enough to solve for u_e and u_o in $\Omega_1 \cup \Omega_2$, and then extend them to the whole domain Ω by symmetry. One iteration of the new NNM thus costs the same as one iteration of the original NNM.

4 Numerical experiments

In order to test our new NNM, we apply it to two simple benchmarks: one with even symmetric data (Example 1: $g = 0$ and $f = 1$) and one with odd symmetric data (Example 2: $g = 0$ and $f = x + y + k$ where $k = \sin(2\phi)$ in Ω_1, $k = -\sin(2\phi)$ in Ω_3 and $k = 0$ in $\Omega_2 \cup \Omega_4$, with ϕ being the angle in polar coordinates, see Figure 3). The discretization of (1) is performed using a standard five point finite difference scheme on a cartesian grid of meshsize $h = 0.01$. When two Dirichlet conditions meet at a corner, the value of g at this node is set to the average of the two values. In addition, when Dirichlet and Neumann conditions meet at a corner, we choose the Dirichlet one to be enforced at this node. The results obtained show that the method behaves as predicted by Theorem 1 and Theorem 4. For $\theta = \frac{1}{4}$, the method converges after two iterations, see the left column in Figure 4. And for $\theta \in (0, \frac{1}{2})$, $\theta \neq \frac{1}{4}$, it converges geometrically to the solution with the expected convergence factor, see the right column in Figure 4 where $\theta_1 = 0.23$ and $\theta_2 = 0.247$. These two graphs also indicate that the convergence behaviour does not depend on h since, in each case, the error curves for $h = 0.01$ and $h = 0.005$ are almost overlaid on each other.

In this short paper, we gave a complete analysis of the standard NNM in a simple configuration involving one cross-point. The even/odd decomposition showed that the NNM was able to treat very efficiently the even symmetric part of the solution, while it faced well-posedness and convergence issues when applied to the

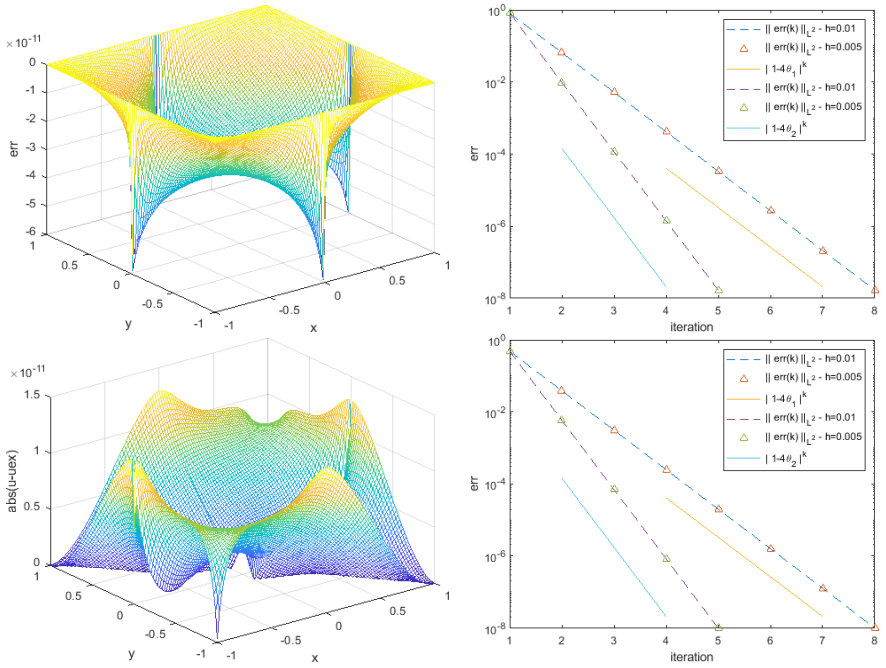


Fig. 4 Absolute error at iteration 2 for $\theta = 0.25$ (left column), and error curves for $\theta \in \{\theta_1, \theta_2\}$ and $h \in \{0.01, 0.005\}$ (right column), in Example 1 (top) and Example 2 (bottom).

odd symmetric part of the solution. Based on this observation, we proposed new mixed transmission conditions of Dirichlet/Neumann type to treat efficiently the odd symmetric part. We proved that the newly proposed NNM built upon a combination between the standard NNM and the new mixed method is convergent, and we validated this property by some numerical experiments. A natural extension of this work would be the 3D case of a cube divided into eight subcubes. It would also be interesting to generalize the notion of even/odd symmetry to the case of more general cross-points (not necessarily rectilinear, or with $N \neq 4$ subdomains).

References

1. Bourgat, J.-F., Glowinski, R., Le Tallec, P., and Vidrascu, M. Variational formulation and algorithm for trace operator in domain decomposition calculations. In: Chan, T., Glowinski, R., Périaux, J., and Widlund, O. (eds.), *Domain Decomposition Methods*. SIAM, Philadelphia, PA (1989).
2. Chaouqui, F., Gander, M. J., and Santugini-Repiquet, K. A local coarse space correction leading to a well-posed continuous Neumann-Neumann method in the presence of cross points. In: *International Conference on Domain Decomposition Methods*, 83–91. Springer (2018).
3. Chaudet-Dumas, B. and Gander, M. J. Cross-points in the Dirichlet-Neumann method I: well-posedness and convergence issues. *Numerical Algorithms* **92**(1), 301–334 (2023).

A Preconditioner for Free-Surface Hydrodynamics BEM

Gabriele Ciaramella, Marco Gambarini, and Edie Miglio

1 Introduction

The computation of hydrodynamic loads from sea surface waves on large arrays of objects is of physical and engineering interest. Typical applications are the simulation of arrays of wave energy converters [3] and the modeling of ice floes in the marginal ice zone [6]. The interest is in array sizes of the order of tens (for wave energy converter arrays) to hundreds (for ice floes) of objects. In these scenarios, the relatively small distances between the floating objects make the correct simulation of mutual hydrodynamic interactions essential. Under the assumptions of incompressible, irrotational, inviscid flow and small displacements, one can derive a linear potential model, which is widely used for the considered range of applications. This model is discretized using the boundary element method [2], resulting in a linear system characterized by a dense and complex matrix. The dimension of the discrete problem grows proportionally to the number of simulated objects. In general, iterative solvers are not scalable for the corresponding numerical solution: the number of iterations needed to achieve a given tolerance grows with the number of objects [5]. To tackle this problem, we propose a preconditioner for the efficient simulation of large arrays of objects and present its implementation using hierarchical matrices.

Consider an array of n floating objects. To compute all its hydrodynamic properties, a number of problems equal to the number of its degrees of freedom needs to be solved. Each problem corresponds to imposing a unit oscillation in one of the degrees of freedom, while keeping all others fixed. Exploiting linearity, the solution of the dynamic problem with loads from incident waves and possibly other external forces can then be written as a linear combination of such unit oscillations. Considering only vertical oscillations, system (1) needs to be solved for $i = 1, \dots, n$

Gabriele Ciaramella, Marco Gambarini, Edie Miglio
MOX, Dipartimento di Matematica Politecnico di Milano, Italy, e-mail:
gabriele.ciaramella@polimi.it, marco.gambarini@polimi.it, edie.miglio@polimi.it

$$\begin{cases} \Delta\phi = 0 & \text{in } \Omega \subset \mathbb{R}^3, \\ \dfrac{\partial\phi}{\partial n} = 0 & \text{on } \Gamma_b, \\ \dfrac{\partial\phi}{\partial z} - \dfrac{\omega^2}{g}\phi = 0 & \text{on } \Gamma_s, \\ \dfrac{\partial\phi}{\partial n} = n_z & \text{on } \Gamma_{o,i}, \\ \dfrac{\partial\phi}{\partial n} = 0 & \text{on } \Gamma_{o,j}, \quad j = 1, \dots, n \wedge j \neq i, \end{cases} \tag{1}$$

where ϕ is the velocity potential, Ω is the (3D) domain, bounded by the sea bottom Γ_b, the mean free surface Γ_s, and the immersed surfaces of the objects $\Gamma_{o,i}$, $i = 1, \dots, n$. Further, ω is the angular frequency of oscillations, g is the gravitational field, and n_z is the vertical component of the normal vector to the surface of objects. The numerical solution using a source-distribution boundary element method (BEM) is based on recasting (1) in integral form:

$$\frac{1}{2}\sigma(\boldsymbol{x}) + \int_{\cup_k \Gamma_{o,k}} \sigma(\boldsymbol{x}') \frac{\partial \mathcal{G}}{\partial n}(\boldsymbol{x}; \boldsymbol{x}') \, \mathrm{d}\boldsymbol{x}' = \begin{cases} n_z & \text{if } \boldsymbol{x} \in \Gamma_{o,i}, \\ 0 & \text{if } \boldsymbol{x} \in \Gamma_{o,j}, \quad j \neq i, \end{cases} \tag{2}$$

$$\phi(\boldsymbol{x}) = \int_{\cup_k \Gamma_{o,k}} \sigma(\boldsymbol{x}') \mathcal{G}(\boldsymbol{x}; \boldsymbol{x}') \, \mathrm{d}\boldsymbol{x}', \quad \forall \boldsymbol{x} \in \Omega. \tag{3}$$

Here, the unknown is the source distribution σ defined on body surfaces. The kernel is the Green function $\mathcal{G}$, a complex elementary solution of the Laplace equation satisfying the boundary conditions on the bottom and free surface [7, Sect. 16]. By discretizing the surfaces of objects into elements, Eq. (2) can be represented as the linear algebraic system $A\boldsymbol{\sigma} = \boldsymbol{b}$. Once this system has been solved, Eq. (3), in the discretized form $\boldsymbol{\phi} = B\boldsymbol{\sigma}$, can be used to compute the potential in any point of the domain.

2 The coarse-corrected block-Jacobi algorithm

The matrix A resulting from the discretization of Eq. (2) is full, because each element interacts with all others. Moreover, even though the Green function is symmetric with respect to an exchange of its arguments, matrix A is non-symmetric because interacting elements have in general different areas and orientations. The problem has a natural block structure

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & \ddots & \vdots \\ A_{n1} & \cdots & A_{nn} \end{bmatrix}, \quad \boldsymbol{\sigma} = \begin{bmatrix} \boldsymbol{\sigma}_1 \\ \vdots \\ \boldsymbol{\sigma}_n \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_n \end{bmatrix}, \tag{4}$$

where $\boldsymbol{\sigma}_j$ is a vector containing the unknowns corresponding to the j-th object. The diagonal block A_{ii} represents the interaction of body i with itself. The off-diagonal block A_{ij} represents the effect on body i of waves radiated from body j. The structure of (4) suggests the use of a block-Jacobi algorithm, equivalent to the parallel method of reflections [5]. This method, together with a coarse correction, has been presented in [5] for the real Laplace equation in perforated domains. Block-Jacobi is based on the splitting $A = D - N$, where D is the block-diagonal part of A. At each iteration, starting from $\boldsymbol{\sigma}^k$, it requires solving for $\boldsymbol{\sigma}^{k+1/2}$ in

$$D\boldsymbol{\sigma}^{k+1/2} = N\boldsymbol{\sigma}^k + \boldsymbol{b}. \tag{5}$$

The solution of (5) can be performed block by block in parallel. After the block-Jacobi step, a coarse correction is performed by solving the correction problem $A\boldsymbol{e} = \boldsymbol{r}^{k+1/2}$ in a low-dimensional (coarse) space C, where $\boldsymbol{r}^{k+1/2} = \boldsymbol{b} - A\boldsymbol{x}^{k+1/2}$ is the residual. Consider, for simplicity, a problem with n identical bodies, each one discretized with p elements, so that the full system has dimension np. Define $C = \text{span}\{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_m\}$, $m \ll np$. Then we can introduce a restriction operator $\mathcal{R}\colon \mathbb{R}^{np} \to C$ represented by matrix $R = \begin{bmatrix}\boldsymbol{c}_1 & \boldsymbol{c}_2 & \ldots & \boldsymbol{c}_m\end{bmatrix}^T$ and a prolongation operator $\mathcal{P} : \mathbb{R}^{np} \to C$ represented by matrix R^T. Let $\boldsymbol{e}_c \in \mathbb{R}^m$ be a vector such that $\hat{\boldsymbol{e}} = R^T \boldsymbol{e}_c$ is an approximation of the error $\boldsymbol{e}$. The coarse problem is

$$RAR^T \boldsymbol{e}_c = R\boldsymbol{r}^{k+1/2}, \tag{6}$$

where $A_c := RAR^T$. Once the coarse problem (6) has been solved, the update

$$\boldsymbol{\sigma}^{k+1} = \boldsymbol{\sigma}^{k+1/2} + R^T \boldsymbol{e}_c$$

is performed. The efficiency of the correction step is strongly related to the choice of the coarse space C. This has to be rich enough to well represent the main error components that block-Jacobi cannot deal with, but its dimension m must be relatively small, so that the cost of a single iteration is not increased significantly. A simple choice for the coarse space is taking a constant value of the source distribution σ on each body. This choice is suggested by the one presented in [5] and corresponds to $\boldsymbol{c}_i := \mathbf{1}_i$, $i = 1, \ldots, n$, $\mathbf{1}_i$ being the discrete indicator function of the i-th object. In this case, the dimension of C is equal to the number of objects n.

Our two-level block-Jacobi method is detailed in Algorithms 1 and 2. The former is a precomputation step, that does not depend on the right-hand side vector. Thus, if multiple systems with the same matrix and different right hand sides need to be solved, Alg. 1 needs to be performed only once. In this algorithm, matrix $\widetilde{R} = RA$ is efficiently (see Section 3) computed, so that the cost for computing the restricted residual at each iteration is reduced. Alg. 2 corresponds to the stationary method

$$\begin{aligned}\boldsymbol{\sigma}^{k+1} &= [I - (P_c + D^{-1} - P_c D^{-1})A]\boldsymbol{\sigma}^k + (P_c + D^{-1} - P_c D^{-1})\boldsymbol{b} \\ &= \boldsymbol{\sigma}^k + (P_c + D^{-1} - P_c D^{-1})\boldsymbol{r}^k,\end{aligned}$$

Algorithm 1 Two-level block-Jacobi algorithm: initialization

for $i = 1$ to n **do**
  Compute the LU decomposition of A_{ii}.
end for
Compute $\widetilde{R} = RA$, $A_c = \widetilde{R}R^T$.

Algorithm 2 Two-level block-Jacobi algorithm: solution

Require: Initial guess $\boldsymbol{\sigma}^0$, tolerance tol, maximum number of iterations $maxit$.
Set $k = 0$.
while $\|\boldsymbol{b} - A\boldsymbol{\sigma}^k\| > tol$ **and** $k < maxit$ **do**
  Compute $\boldsymbol{q} = \boldsymbol{b} - N\boldsymbol{\sigma}^k$.
  for $i = 1$ **to** n **do**
    Solve $A_{ii}\boldsymbol{\sigma}_i^{k+1/2} = \boldsymbol{q}_i$ using the LU decomposition of A_{ii}.
  end for
  Compute the restricted residual $\boldsymbol{r}_c = R\boldsymbol{b} - \widetilde{R}\boldsymbol{\sigma}^{k+1/2}$.
  Solve for $\boldsymbol{e}_c$ in $A_c\boldsymbol{e}_c = \boldsymbol{r}_c$.
  Update $\boldsymbol{\sigma}^{k+1} = \boldsymbol{\sigma}^{k+1/2} + R^T\boldsymbol{e}_c$.
  Update $k = k + 1$.
end while

with $P_c = R^T A_c^{-1} R$ and where we can recognize the inverse preconditioner $P^{-1} = P_c + D^{-1} - P_c D^{-1}$. Such preconditioner can then be used to accelerate a Krylov method. Using P^{-1}, the system is recast as $P^{-1}A\boldsymbol{\sigma} = P^{-1}\boldsymbol{b}$. Since the new system matrix $P^{-1}A$ is not symmetric, a classical choice is GMRES. In our implementation, the preconditioning matrix P^{-1} is not assembled explicitly; instead, GMRES is provided with a function (based on Alg. 2) computing the action of $P^{-1}A$ on an arbitrary vector.

3 Implementation details and $\mathcal{H}$-matrices

Hierarchical matrices, denoted here as $\mathcal{H}$-matrices, are an efficient tool for reducing the storage and computational cost of BEM problems. The method is based on defining a hierarchical cluster tree from the set of mesh elements. The system matrix is then built with a hierarchical block structure accordingly. Each block describes the interaction between two clusters of elements. If the centers of the two clusters are farther than a threshold, then a low-rank approximation on the block is built; otherwise, the block is built in dense form. If the tree is balanced and if we take as leaves of the tree the single objects, discretized with p elements, then both the costs of storage and of matrix-vector multiplication are $O(\max(r, p)np\log(np))$ [4, Th. 2.6, 2.8], where r is the maximum rank of matrix blocks.

Fig. 1 shows the tree and the hierarchical structure of matrix A for an example with 10 objects on a row, with spacing of 5 m. The ordinate of each node in the tree is the distance between the centers of its sons. In the matrix, blue blocks are dense, while white blocks are low-rank. Notice that dense blocks gather mostly close to

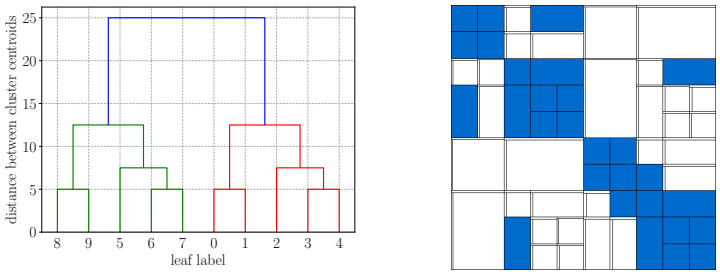


Fig. 1 Clustering of positions (left) and hierarchical structure of matrix A (right) for a test with 10 objects.

the diagonal. Information on the nodes of the tree is stored in the so-called linkage matrix. The leaves constitute the first n nodes of the tree. All other nodes are defined by the rows of the linkage matrix: its i-th row contains the labels of the sons of the $(n+i)$-th node.

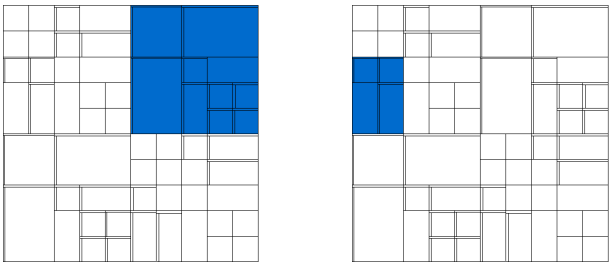
Our implementation of Alg. 1 and 2 is done starting from the BEM code Capytaine [2], which includes an $\mathcal{H}$-matrices engine. Matrix-vector multiplication in line 3 of Alg. 2 is performed with the built-in routine. Handling of diagonal blocks and building the coarse space, instead, require special care. Since the structure of matrix storage is hierarchical, extracting the diagonal blocks of A to build matrices D and N is not immediate. In order to do it, first a list `leaves`, whose i-th element is the list of leaves belonging to node i, is built by sweeping over the rows of the linkage matrix. Then, a list `paths` is constructed. This contains n sublists. The i-th sublist has size equal to the level of the i-th leaf. The ℓ-th element of this sublist is equal to 0 or 1, if at the ℓ-th level one has to turn left or right, respectively, to step down toward the i-th leaf. For example, in Fig. 1 (left) the path to leaf 5 is `paths`$[5] = [0, 1, 0]$. The lists `leaves` and `paths` are exploited to compute RA efficiently in Alg. 2. Because of the sparsity of the rows of R, that are vectors $\mathbf{c}_i^T$, for dense matrices this operation can be made very efficient by multiplying each of the $\mathbf{c}_i$ only by the rows of A corresponding to its non-zero elements. Slicing a hierarchical matrix, however, is not as trivial. For this reason, we propose the recursive procedure detailed in Alg. 3 and described graphically in Fig. 2. At the beginning, $\widetilde{A} = A$ and $\boldsymbol{v} = \boldsymbol{c}_i$ are set. The algorithm then descends from the root to the level above the i-th leaf following list `path` = `paths`$[i]$. In doing this, because of the structure of the tree, 2×2 blocks are encountered at each level. At level j, the nonzero contributions $\boldsymbol{c}_i A$ come only from the k-th block-row, with $k =$ `path`$[j]$. The off-diagonal part of the k-th block row is directly multiplied by the appropriate slice of $\boldsymbol{v}$; then, the algorithm is applied again to the diagonal block $\widetilde{A}_{kk}$. At the end, only the (dense) diagonal block corresponding to the interaction of the i-th object with itself is left, and this last multiplication is performed. The main advantage of this strategy is that, at each level of the hierarchy except the last, off-diagonal blocks, that are expected to be mostly low-rank, are multiplied.

Algorithm 3 Computation of $\widetilde{R} = RA$ for $\mathcal{H}$-matrices

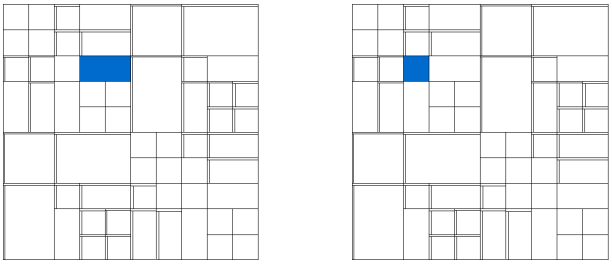
Require: A, $R = [\boldsymbol{c}_1, \cdots, \boldsymbol{c}_n]$ and `paths`.
for $i = 1$ **to** n **do**
  Select the path to the i-th leaf: `path` = `paths`$[i]$.
  Set $\widetilde{A} = A$, $\boldsymbol{v} = \boldsymbol{c}_i$, $a = 0$, $b = np$, and initialize a zero array $\boldsymbol{d}$ of size np.
  for $j = 1$ **to** length(`path`) **do**
    Set $k =$ `path`$[j]$ and N_{row} as the number of rows of $\widetilde{A}_{kk}$.
    if $k = 0$ **then**
      Select the first N_{row} rows of $\boldsymbol{v}$: $\boldsymbol{v} = \boldsymbol{v}[0 : N_{row}]$.
      Multiply $\boldsymbol{w} = \boldsymbol{v}\widetilde{A}_{01}$ and set $\boldsymbol{d}[b - \text{length}(\boldsymbol{w}) : b] = \boldsymbol{w}$.
      Update: $b = b - \text{length}(\boldsymbol{w})$.
    else
      Select the last N_{row} rows of $\boldsymbol{v}$: $\boldsymbol{v} = \boldsymbol{v}[\text{end} - N_{row} : \text{end}]$.
      Multiply $\boldsymbol{w} = \boldsymbol{v}\widetilde{A}_{10}$ and set $\boldsymbol{d}[a : a + \text{length}(\boldsymbol{w})] = \boldsymbol{w}$.
      Update: $a = a + \text{length}(\boldsymbol{w})$.
    end if
    Set $\widetilde{A} = \widetilde{A}_{kk}$.
  end for
  Diagonal block multiplication: $\boldsymbol{d}[a : b] = \boldsymbol{v}\tilde{A}$.
  Set $\widetilde{R}[i, :] = \boldsymbol{d}$.
end for

4 Numerical experiments

The method is implemented by integration with the BEM code Capytaine [2]. Hierarchical clustering on the positions of the objects is performed using SciPy. We simulate two geometries: line arrays and grid arrays. In both cases the objects are half-spheres of radius 2 m and the minimum distance between two bodies is 5 m. The results are reported in Table 1. Times for GMRES and preconditioned GMRES refer to the solution of the n systems required to build the radiation dataset; thus the number of systems needing to be solved increases with the number of objects. The loops described in Algorithms 2 and 3 are performed serially. We build the radiation dataset only for vertical motion; in the general case of a rigid body, $6n$ systems would need to be solved. In some cases, the number of iteration varies depending on the right hand side (i.e., depending on the radiating object).



(a) $j = 1$ (b) $j = 2$



(c) $j = 3$ (d) Diagonal block

Fig. 2 Blocks selected for multiplication in Algorithm 3 for leaf $i = 5$.

Table 1 Results of the numerical experiment. Top table: line geometry. Bottom table: grid geometry. In both tables, init is the time for initializing the coarse solver (coarse space definition).

		GMRES			Preconditioned GMRES			
n	storage (%)	niter	t (s)	t/n (s)	init (s)	niter	t (s)	t/n (s)
80	6.19	12	18	0.23	0.47	7	15	0.19
160	3.32	13	107	0.67	1.05	7	68	0.85
240	2.33	14	449	1.87	1.64	7	294	1.23
320	1.75	14-15	831	2.60	2.51	7	487	1.52
400	1.43	15	1182	2.95	3.46	7	698	1.74
480	1.23	15-16	1715	3.57	4.14	7	993	2.07

		GMRES			Preconditioned GMRES			
n	storage (%)	niter	t (s)	t/n (s)	init (s)	niter	t (s)	t/n (s)
16	48.42	10	0.51	0.03	0.07	7	1.03	0.06
64	20.00	13	35	0.55	0.87	8	25	0.40
144	10.82	17	281	1.95	3.40	8-9	157	1.09
256	6.72	24-26	2216	8.66	15.5	9	826	3.23
400	4.82	38-43	7215	18.03	23.5	9-10	2030	5.08

5 Discussion and conclusions

The presented results indicate that the preconditioned GMRES method has a lower cost than the standard GMRES method for large arrays of floating objects. The

advantage becomes larger as the number of bodies increases: speedups of up to a factor of 3.5 are obtained. For the line geometry, the number of iterations of GMRES tends to become constant with respect to the number of objects, while the iterations of preconditioned GMRES remain exactly constant and equal to 7. On the other hand, for the grid geometry the number of iterations of GMRES increases as n grows, while preconditioned GMRES scales well. The use of Alg. 3 for the construction of the coarse space, which needs to be performed only once, keeps the cost of such operation low. Thus, a substantial speedup can be obtained with respect to standard GMRES even when a small subset of the entire radiation dataset needs to be computed. In the grid test case the percentage of dense blocks is larger, resulting in a larger time for the initialization of Alg. 3.

Possible improvements include the parallelization of the loops in Alg. 2 and the use of a preconditioner also for the solution of the coarse problem, whose cost can become relevant for very large arrays. In the case of a single row of bodies, the coarse matrix A_c has a Toeplitz structure, and the natural choice in this case is to use a circulant preconditioner. This strategy has been explored at block level in [1], while some choices of circulant preconditioners are presented in [8].

References

1. Ancellin, M. and Dias, F. Using the floating body symmetries to speed up the numerical computation of hydrodynamics coefficients with Nemoh. Proceedings of the 37th International Conference on Ocean, Offshore and Artic Engineering (2018).
2. Ancellin, M. and Dias, F. Capytaine: a Python-based linear potential flow solver. *J. Open Source Softw.* **4**(36), 1341 (2019).
3. Babarit, A. On the park effect in arrays of oscillating wave energy converters. *Renewable Energy* **58**, 68–78 (2013).
4. Bebendorf, M. *Hierarchical Matrices*. Lecture Notes in Computational Science and Engineering. Springer Berlin, Heidelberg (2008).
5. Ciaramella, G., Gander, M. J., Halpern, L., and Salomon, J. Methods of Reflections: relations with Schwarz methods and classical stationary iterations, scalability and preconditioning. *SMAI J. Comput. Math* **5**, 161–193 (2019).
6. Squire, V. A. Ocean wave interactions with sea ice: A reappraisal. *Annual Review of Fluid Mechanics* **52**(1), 37–60 (2020).
7. Wehausen, J. V. and Laitone, E. V. Surface waves. In: Truesdell, C. (ed.), *Fluid Dynamics / Strömungsmechanik*, 446–778. Springer Berlin Heidelberg (1960).
8. Zhu, Z. and Wakin, M. B. On the asymptotic equivalence of circulant and Toeplitz matrices. *IEEE Transactions on Information Theory* **63**(5), 2975–2992 (2017).

A Performance Comparison of Classical Volume and New Substructured One- and Two-Level Schwarz Methods in PETSc

Gabriele Ciaramella, Martin J. Gander, Serge Van Criekingen, and Tommaso Vanzan

1 Introduction

Substructured Schwarz methods are interpretations of volume Schwarz methods as algorithms on interface variables. We compare here the Parallel Schwarz Method (PSM, equivalent to RAS) in volume to the new substructured version of PSM in [11, p.24] and recently extended to a two-level (i.e. coarse-corrected) framework in [6] and [5], using a geometric and spectral approach for the definition of the coarse space. The expected gain of substructured methods is due to the smaller size of the resulting problems, notably with Krylov-type acceleration techniques when the dimension of the subspace of approximants becomes large [12].

While the numerical results in [5, 6] were obtained sequentially, we present here a parallel performance comparison of volume and substructured Schwarz methods using PETSc [1, 2, 3], successively considering one- (Section 2) and two- (Section 3) level methods. The substructured results are compared to the ones obtained by the RAS method in volume [4] for which two-level results with various coarse spaces were presented in [9, 10] also using PETSc . For the two-level substructured method, four coarse spaces are introduced here, all based on a geometric approach. Note that, at this time, spectral approaches still require further investigations and are therefore

Gabriele Ciaramella
MOX, Dipartimento di Matematica, Politecnico di Milano, Piazza Leonardo da Vinci 32, 20133 Milano, Italy, e-mail: gabriele.ciaramella@polimi.it

Martin J. Gander
University of Geneva, e-mail: martin.gander@unige.ch

Serge Van Criekingen
Institut du Développement et des Ressources en Informatique Scientifique (IDRIS), CNRS, Université Paris-Saclay, F-91403, Orsay, France and Université Paris-Saclay, UVSQ, CNRS, CEA, Maison de la Simulation, 91191, Gif-sur-Yvette, France, e-mail: serge.van.criekingen@idris.fr

Tommaso Vanzan
CSQI Chair, EPFL Lausanne, e-mail: tommaso.vanzan@epfl.ch

not presented here (- the reason being that the eigenvectors on which spectral coarse spaces are based in [5] are in general complex and in turn necessitate a PETSc installation adapted to complex arithmetic, which has a negative influence on the resulting computational times).

2 The one-level substructured formulation

We consider the system $Au = f$ for the Laplace problem with Dirichlet boundary conditions discretized with finite differences. We first derive the substructured system

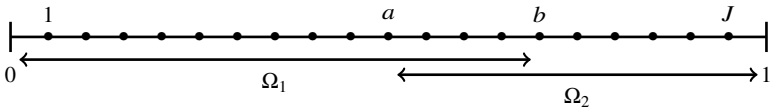


Fig. 1 Two subdomain decomposition in the 1-D case.

for the 1-D case, namely the [0,1] interval subdivided into $J+1$ mesh cells of size h as depicted in Fig. 1 in the two-subdomain case. Following [11], we decompose $A \subset \mathbb{R}^{(J-1)\times(J-1)}$ in two different ways as

$$A = \begin{pmatrix} A_1 & B_1 \\ C_1 & D_1 \end{pmatrix} = \begin{pmatrix} D_2 & C_2 \\ B_2 & A_2 \end{pmatrix}, \tag{1}$$

where $A_1 \subset \mathbb{R}^{(b-1)\times(b-1)}$ and $A_2 \subset \mathbb{R}^{(J-a)\times(J-a)}$. Our starting point is the discretized Parallel Schwarz Method (PSM) for $Au = f$ which reads

$$A_1 u_1^{n+1} = f_1 - \tilde{B}_1 u_2^n, \tag{2}$$
$$A_2 u_2^{n+1} = f_2 - \tilde{B}_2 u_1^n, \tag{3}$$

where $\tilde{B}_1 = [0_{b-1,d-1} B_1]$ and $\tilde{B}_2 = [B_2 0_{J-a,d-1}]$ (with $d = b - a$ the overlap) are extensions by zeros of the B_1 and B_2 matrices of (1) such that

$$\tilde{B}_1 u_2 = (0, ..., 0, -\frac{1}{h^2}(u_2)_b) \subset \mathbf{R}^{b-1},$$
$$\tilde{B}_2 u_1 = (-\frac{1}{h^2}(u_1)_a, 0, \ldots, 0) \subset \mathbf{R}^{J-a}.$$

Thus, $\tilde{B}_1$ maps a vector defined on Ω_2 into one defined on Ω_1, extended by zero out of Ω_2 (and similarly for $\tilde{B}_2$). We introduce the trace operators

$$G_1 \colon (v_1, \ldots, v_a, \ldots, v_{b-1}) \to v_a,$$
$$G_2 \colon (v_{a+1}, \ldots, v_b, \ldots, v_J) \to v_b,$$

such that $G_1 u_1 = (u_1)_a$ and $G_2 u_2 = (u_2)_b$, as well as the extension by zero operators

$$E_1 : v_b \to (0, \ldots, 0, v_b) \subset \mathbf{R}^{b-1},$$
$$E_2 : v_a \to (v_a, 0, \ldots, 0) \subset \mathbf{R}^{J-a},$$

such that $\tilde{B}_1 u_2 = -\frac{1}{h^2} E_1 (u_2)_b$ and $\tilde{B}_2 u_1 = -\frac{1}{h^2} E_2 (u_1)_a$. Applying the trace operators to the PSM system (2)–(3) then yields

$$(u_1^{n+1})_a = \frac{1}{h^2} G_1 A_1^{-1} E_1 (u_2^n)_b + G_1 A_1^{-1} f_1,$$
$$(u_2^{n+1})_b = \frac{1}{h^2} G_2 A_2^{-1} E_2 (u_1^n)_a + G_2 A_2^{-1} f_2.$$

Defining interface unknowns $g^T = (g_1, g_2) = ((u_1)_a, (u_2)_b)$, this is the block Jacobi method applied to the substructured system

$$Tg = f^g, \tag{4}$$

where

$$T = \begin{pmatrix} I & -\frac{1}{h^2} G_1 A_1^{-1} E_1 \\ -\frac{1}{h^2} G_2 A_2^{-1} E_2 & I \end{pmatrix} \text{ and } f^g = \begin{pmatrix} G_1 A_1^{-1} f_1 \\ G_2 A_2^{-1} f_2 \end{pmatrix}. \tag{5}$$

This system can also be solved using a Krylov method (`GMRES` here).

From a parallel data transfer point of view, in the two-subdomain case of Fig.1, we have that Ω_1 sends u_a to Ω_2, while Ω_2 sends u_b to Ω_1. In the three subdomain case (Fig.2), two trace operators are necessary for the central subdomain Ω_2, ex-

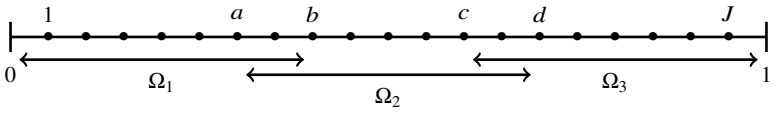
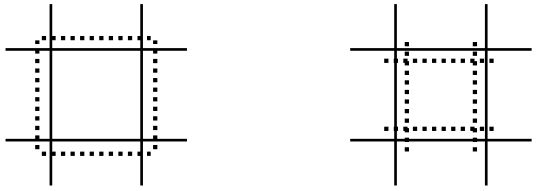


Fig. 2 Three subdomain decomposition in the 1-D case.

tracting respectively u_b and u_c and sending them to Ω_1 and Ω_3, again respectively. Meanwhile, subdomain Ω_2 receives u_a from Ω_1 and u_d from Ω_3.

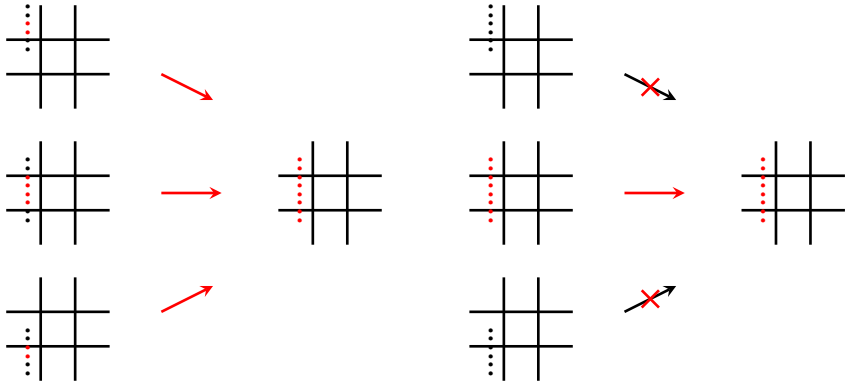
In 2-D, for a typical non-boundary subdomain, data exchange consists in receiving data on a square skeleton obtained by extending the domain by the size of the overlap (Fig. 3a) and sending local data from four "portions" within the domain, at overlap distance from the interface (Fig. 3b). Furthermore, in 2D a partition of unity is required and we investigated two data exchange options, with or without transfers from diagonal neighbours, as illustrated in Fig. 4 for the left-to-right data exchange.

The T substructured system matrix defined in (5) is implemented matrix-free in our `PETSc` implementation, using the `MatCreateShell` and `MatShellSetOperation` tools. Each multiplication by T implies data transfer (with or without



(a) (b)

Fig. 3 Dotted are the substructure values to be received (a) or sent (b) by the central subdomain.

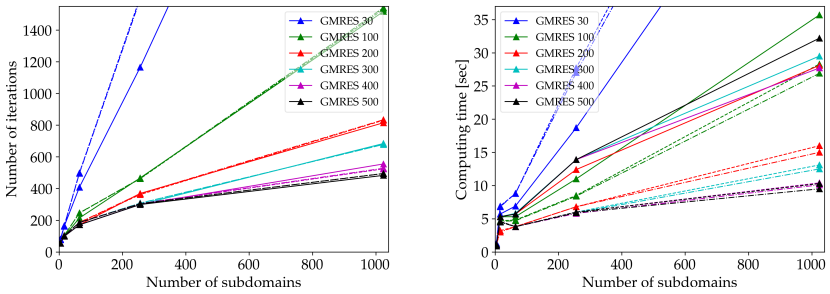


(a) (b)

Fig. 4 Schematic representation of left-to-right data exchange with (a) or without (b) transfers from diagonal neighbours. The transferred data are in red.

diagonal transfers), extension by zero (E_i), *exact* solve by the local matrices A_i (direct solver with LU decomposition computed only once) and taking the trace in the subdomain (G_i). To solve the substructured system (4), we apply `GMRES` without preconditioner, since this system is in fact already preconditioned by the Schwarz method.

We compare our substructured method to the (volume) RAS method [4] (implemented in `PETSc` as `PCASM`) on a weak scaling experiment for the 2-D Laplace problem on the unit square with 5-point finite difference scheme, using square decompositions into 2×2 to 32×32 subdomains (one processor per subdomain) and a 256×256 fine mesh within each subdomain (.004 fine-to-coarse mesh ratio). Several observations can be made from the results displayed in Fig.5. First, there is virtually no difference in the number of iterations with or without diagonal transfers, so that the extra cost of the diagonal transfers is not compensated by a decrease in iterations. Consequently, we stick to the no diagonal transfer option in the remainder of our study. Second, when looking at computational times, the optimal `GMRES` restart parameter for the substructured method (here 500, which in fact means no restart since a bit less than 500 iterations are then performed) appears to be larger than for the volume method (here 400 with 200 being very close), the smaller size of



(a) Iteration count (b) Timing results

Fig. 5 Weak scaling results for the 2×2 to 32×32 square decompositions, using various `GMRES` restart parameters. Volume methods (solid lines) and substructured methods with (dashed lines) or without (dashdot lines) diagonal transfer are used.

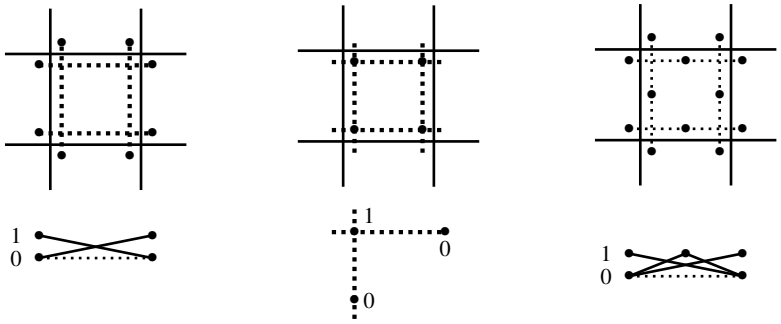
the substructured problem thus making a larger Krylov space profitable. Third, and most importantly, at high restart parameters and in particular at the optimal one, substructured methods yield better timing performances than volume methods. This appears to be due to the smaller size of the substructured systems since the number of iterations with both methods is similar.

3 Two-level substructured methods

We model our two-level substructured method on the (volume) two-level RAS methods ("RAS2") developped in [9], namely

$$u^{n+1/2} = u^n + \sum_{j=1}^{J} \tilde{R}_j^T \; A_j^{-1} \; R_j \; (f - Au^n),$$
$$u^{n+1} = u^{n+1/2} + R_c^T \; A_c^{-1} \; R_c \; (f - Au^{n+1/2}),$$

where R_j are restriction operators to the (possibly overlapping) Ω_j subdomains decomposing the global domain Ω, $\tilde{R}_j$ are the equivalents for a non-overlapping decomposition of Ω into $\tilde{\Omega}_j$, and R_c is the restriction operator to the coarse space. Moreover, we have defined the local matrices as $A_j = R_j \, A \, R_j^T$ and the coarse matrix as $A_c = R_c \, A \, R_c^T$. In our `PETSc` implementation, this is implemented as a multiplicative composition (`PCCOMPOSITE`) of RAS (`PCASM`) with a hand-made second-level correction (`PCSHELL` framework). The coarse solve A_c^{-1} is performed with the direct solver `MUMPS` with agglomeration of the coarse unknowns. A `GMRES` acceleration can be applied to the (full) iteration. The volume RAS2 coarse correction chosen here is `Q1`, a coarse space made out of linear functions with, in 2-D, four coarse nodes placed around each cross-point [7, 8, 9].



(a) `Linear` (b) `Linear` 4 (c) `Enriched`

Fig. 6 Schematic view of substrucutred coarse space options, with coarse point positions (above) and coarse function sketch (below).

We proceed similarly for our two-level substructured implementation: for the system $Tg = f^g$, our two-level method reads

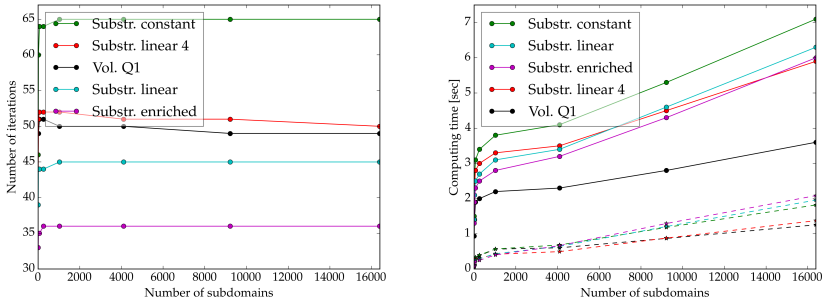
$$g^{n+1/2} = g^n + (f^g - Tg^n), \tag{6}$$

$$g^{n+1} = g^{n+1/2} + R_c^T \, T_c^{-1} \, R_c \, (f^g - Tg^{n+1/2}), \tag{7}$$

where R_c is again the restriction operator to the coarse space and $T_c = R_c T R_c^T$ is the coarse matrix. In `PETSc`, we proceed again with a multiplicative composition of, this time, `PCNONE` (no preconditioner) with a hand-made second-level correction. The T_c matrix is built once and for all at the begining of the calculation, as well as its LU decomposition using `MUMPS`. Here also `GMRES` can be applied to the full iteration.

Our substructured coarse space functions will be defined exclusively on the interfaces, more precisely, for each of them, on the four substructure portions of a typical non-boundary subdomain (Fig. 3b). We here consider four geometric substructured coarse spaces, namely `Constant` with one constant coarse function per portion (so 4 functions for a non-boundary subdomain), `Linear` (Fig. 6a) with two linear coarse functions per portion (so 8 coarse points and functions for a non-boundary subdomain), `Linear4` (Fig. 6b) with four linear functions (and as many coarse points) for a non-boundary subdomain (- this space can be seen as the volume `Q1` coarse space restricted to the substructure) and `Enriched` (Fig. 6c) with three linear coarse functions per portion (so 12 coarse points and functions for a non-boundary subdomain). Thus, for an $N \times N$ decomposition, the coarse space sizes asymptotically behave as $4N^2$ with `Constant` and `Linear4`, $8N^2$ with `Linear` and $12N^2$ with `Enriched`.

Figure 7 displays iteration count and computational (wall-clock) times for the weak scaling experiment described above using the two-level volume and substructured methods, with square decompositions up to 128×128 subdomains (- the



(a) Iteration count

(b) Wall-clock total (solid) and coarse solution (dashed) times

Fig. 7 Two-level numerical results up to 16,384 processors.

solution time results, not shown here, exhibit a very similar behavior). There is no `GMRES` restart performed here. We observe that all our two-level methods achieve scalability in terms of number of iterations. Scalability in terms of computational times is quite well achieved even though not perfectly, with performances slightly below the two-level volume `Q1` method. It is possible to improve the substructured computational times further by noting that the two-level iteration (6)–(7) requires the computation of two actions of the operator T, and one of them can be eliminated using the strategy proposed in [5, 6]. This is possible to do in PETSc as well, but requires a substantial modification in the implementation technique that goes beyond this short manuscript, and will appear elsewhere. Note also the particularly interesting behavior of the `Linear4` coarse space, yielding less iterations than the `Constant` one with asymptotically the same number of coarse functions. Its coarse solution time appears very close the `Q1` one in volume as shown in Fig. 7b (dashed lines).

4 Conclusions

A `PETSc` implementation of the substructured one-and two-level PSM has been presented. Our one-level results show that the smaller size of the substructured system compared to the volume one makes the use of larger Krylov spaces (i.e., using larger `GMRES` restart parameters, or no restart at all) profitable, resulting in better computational times. Furthermore, we introduced four new substructured geometric coarse spaces defined exclusively on the interfaces and our numerical results up to 16,384 cores show that the resulting two-level methods achieve a perfect scalability in terms of number of iterations and a very decent scalability in terms of computational solution and wall-clock times.

Acknowledgements This work was performed using HPC resources from GENCI-IDRIS.

References

1. Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E. M., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., Kong, F., Kruger, S., May, D. A., McInnes, L. C., Mills, R. T., Mitchell, L., Munson, T., Roman, J. E., Rupp, K., Sanan, P., Sarich, J., Smith, B. F., Zampini, S., Zhang, H., Zhang, H., and Zhang, J. PETSc Web page. https://petsc.org/ (2022).
2. Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E. M., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., Kong, F., Kruger, S., May, D. A., McInnes, L. C., Mills, R. T., Mitchell, L., Munson, T., Roman, J. E., Rupp, K., Sanan, P., Sarich, J., Smith, B. F., Zampini, S., Zhang, H., Zhang, H., and Zhang, J. PETSc/TAO users manual. Tech. Rep. ANL-21/39 - Revision 3.18, Argonne National Laboratory (2022).
3. Balay, S., Gropp, W., McInnes, L. C., and Smith, B. Efficient management of parallelism in object oriented numerical software libraries. In: Arge, E., Bruaset, A. M., and Langtangen, H. P. (eds.), *Modern Software Tools in Scientific Computing*, 163–202. Birkhäuser Press (1997).
4. Cai, X.-C. and Sarkis, M. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comp.* **21**(2), 239–247 (1999).
5. Ciaramella, G. and Vanzan, T. Spectral coarse spaces for the substructured parallel Schwarz method. *J. Sci. Comput.* **91**(69) (2022).
6. Ciaramella, G. and Vanzan, T. Substructured two-grid and multi-grid domain decomposition methods. *Numerical Algorithms* **91**, 413–448 (2022).
7. Dubois, O., Gander, M., Loisel, S., St-Cyr, A., and Szyld, D. The optimized Schwarz methods with a coarse grid correction. *SIAM J. Sci. Comp.* **34**(1), A421–A458 (2012).
8. Gander, M., Halpern, L., and Santugini, K. A new coarse grid correction for RAS/AS. In: *Domain Decomposition Methods in Science and Engineering XXI*, Lecture Notes in Computational Science and Engineering, 275–284. Springer-Verlag (2014).
9. Gander, M. and Van Criekingen, S. New coarse corrections for restricted additive Schwarz using PETSc. In: *Domain Decomposition Methods in Science and Engineering XXV*, Lecture Notes in Computational Science and Engineering, 483–490. Springer-Verlag (2019).
10. Gander, M. and Van Criekingen, S. Coarse corrections for Schwarz methods for symmetric and non-symmetric problems. In: *Domain Decomposition Methods in Science and Engineering XXVI*, Lecture Notes in Computational Science and Engineering, 589–596. Springer-Verlag (2021).
11. Gander, M. J. and Halpern, L. Méthodes de décomposition de domaine, encyclopédie électronique pour les ingénieurs. Tech. rep. (2012).
12. Saad, Y. *Iterative Methods for Sparse Linear Systems*. SIAM (2003).

Semi-Discrete Analysis of a Simplified Air-Sea Coupling Problem with Nonlinear Coupling Conditions

Simon Clement, Florian Lemarié, and Eric Blayo

1 Introduction

This paper addresses the mathematical properties of a simplified nonlinear coupling problem representing the air-sea exchanges: it is shown in [4] that there is room for improvement in the coupling methods of state-of-the-art Earth System Models. Key ingredients in the ocean-atmosphere coupling are the computation, exchange and diffusion of turbulent fluxes at the interface. Those ingredients are represented by a nonlinear coupling condition which depends on the space discretization. We focus here on showing the existence of strong unsteady solutions of the semi-discrete in space problem with a nonlinear interface condition. Moreover we study the convergence of Schwarz Waveform Relaxation (SWR) applied to this coupled problem. In [2], the existence of unsteady solutions of nonlinear turbulent models for oceanic surface mixing layers is proven with the help of the inverse function theorem. After introducing the coupled problem in §2, its well-posedness is discussed by applying the inverse function theorem in §3. A convergence analysis of SWR is then pursued in §4 and complemented by numerical experiments detailed in §5.

2 Simplified air-sea coupled problem

We examine the solutions U_a, U_o of coupled 1D linear reaction-diffusion equations which is a proxy for coupled ocean-atmosphere problems [3, 5]:

$$(\partial_t + \mathrm{i}f)U_j = \nu_j \partial_{zz} U_j + \mathrm{i}f u_G^j, \qquad (j = a, o), \tag{1}$$

Simon Clement, Florian Lemarié, Eric Blayo
Univ. Grenoble Alpes, CNRS, Inria, Grenoble INP, LJK, Grenoble, France,
e-mail: simon.clement2@univ-grenoble-alpes.fr, florian.lemarie@inria.fr,
eric.blayo@univ-grenoble-alpes.fr

where initial, boundary and coupling conditions will be specified together with the discretization. The reaction-diffusion equation (1) has a constant viscosity ν_j representing the turbulent vertical mixing and a complex reaction term $\mathrm{i}f$ accounting for the Coriolis effect: U_j are complex and include the horizontal direction of winds and currents (e.g. [5]). The Coriolis parameter f is a constant determined by the latitude of the 1D vertical column considered. There is also a nudging term u_G^j in (1) which pulls the solution towards the geostrophic equilibrium, i.e. a balance between the Coriolis force and the pressure gradient. In this study u_G^a, u_G^o are constant.

The coupling between ocean and atmosphere actually excludes from the computational domains a surface layer $\Omega_{\mathrm{sl}} = [z_{-\frac{1}{2}}, z_{\frac{1}{2}}]$ which is located between the first grid points of the two domains and contains the interface z_0. The size of this surface layer being linked with the discretization in space, we investigate the properties of the semi-discrete in space coupled problem. The solution U_j ($j = o, a$) at the grid point $z_{m+\frac{1}{2}}$ (m is the space index) is denoted $U_{j,m+\frac{1}{2}}$ and we denote by ϕ_j the derivatives in space which are approximated with finite differences:

$$\phi_{j,m} = \frac{1}{h_j}\left(U_{j,m+\frac{1}{2}} - U_{j,m-\frac{1}{2}}\right), \qquad m \neq 0, \tag{2}$$

where h_j is the space step assumed constant in each subdomain. The function $\phi_{j,0}$ will be determined by the boundary conditions involved in the coupling. The semi-discrete in space coupled problem is the following (for $\frac{H_o}{h_o} < m + \frac{1}{2} < \frac{H_a}{h_a}$, where H_j are the size of the spatial domains):

$$(\partial_t + \mathrm{i}f)U_{j,m+\frac{1}{2}} = \nu_j \frac{\phi_{j,m+1} - \phi_{j,m}}{h_j} + \mathrm{i}f u_G^j, \qquad t \in]0,T], \tag{3a}$$

$$U_{j,m+\frac{1}{2}}\Big|_{t=0} = U_0, \qquad U_j\big|_{z=H_j} = U_j^\infty, \qquad t \in]0,T], \tag{3b}$$

$$\nu_a \phi_{a,0} = C_D \left|U_{a,\frac{1}{2}} - U_{o,-\frac{1}{2}}\right| \left(U_{a,\frac{1}{2}} - U_{o,-\frac{1}{2}}\right), \qquad t \in]0,T], \tag{3c}$$

$$\rho_o \nu_o \phi_{o,0} = \rho_a \nu_a \phi_{a,0}, \qquad t \in]0,T], \tag{3d}$$

The initial condition U_0 is chosen as the steady state (derived in Section 3.1) and the geostrophic winds and currents are prescribed as boundary conditions: $U_j^\infty = u_G^j$. The spatial extent of the domains will be considered sufficiently large ($H_j \to \infty$) but all the results can be easily extended to finite domains. The coupling conditions are composed of a quadratic friction law (3c) and of a flux continuity (3d). Those conditions are representative of the ones that can be found in more realistic models: they are simpler, because C_D and ν_j are assumed to be constant instead of depending themselves on $U_{a,\frac{1}{2}} - U_{o,-\frac{1}{2}}$. The densities ρ_j are such that $\frac{\rho_a}{\rho_o} \approx 10^{-3}$.

3 Well-posedness

In this section the focus is on the well-posedness of (3). First the steady state is given, and the existence and uniqueness of solutions are proven in a neighborhood of this steady state. We assume in this section that $f \neq 0$: it can be proven that there is otherwise no bounded steady state of (3).

3.1 Steady state

The derivation of the steady state of (3a) is somewhat similar to [6] and we obtain $\phi^e_{o,-m} = A^e(\lambda^e_o + 1)^m$ and $\phi^e_{a,m} = B^e(\lambda^e_a + 1)^m$ with $\lambda^e_j = \frac{1}{2}\left(\chi_j - \sqrt{\chi_j}\sqrt{\chi_j + 4}\right)$, $\chi_j = \frac{\mathrm{i}f h_j^2}{\nu_j}$. The continuity of the flux (3d) gives $\frac{A^e}{B^e}$ and finding a steady state solution amounts to find $\widetilde{x} \in \mathbb{C}$ such that

$$\widetilde{x} - \left(u^a_G - u^o_G\right) = d|\widetilde{x}|\widetilde{x}, \tag{4}$$

where $d = C_D\left(\frac{\lambda^e_a}{\mathrm{i}f h_a} + \frac{\rho_a}{\rho_o}\frac{\lambda^e_o}{\mathrm{i}f h_o}\right)$ and $\widetilde{x} = A^e\frac{\rho_o\nu_o}{\rho_a C_D}d + \left(u^a_G - u^o_G\right)$. The steady state corresponds to a solution $\widetilde{x}$ whose modulus $|\widetilde{x}|$ is a real and non-negative root of a polynomial:

$$|\widetilde{x}| = \frac{d_R}{2|d|^2} \pm \frac{\sqrt{\zeta + \gamma}}{2} \pm \frac{1}{2}\sqrt{2\zeta - \gamma \pm \frac{2{d_R}^3 - 2d_R|d|^2}{|d|^6\sqrt{\zeta + \gamma}}}, \quad d_R = \Re(d), \tag{5a}$$

$$\gamma = \frac{\left(\frac{s}{2} + \frac{1}{2}\sqrt{s^2 - 4\beta^3}\right)^{\frac{1}{3}}}{3|d|^2} + \frac{\beta}{3|d|^2\left(\frac{s}{2} + \frac{1}{2}\sqrt{s^2 - 4\beta^3}\right)^{\frac{1}{3}}}, \quad \zeta = -\frac{2}{3|d|^2} + \frac{{d_R}^2}{|d|^4}, \tag{5b}$$

$$s = 2 + \left(u^a_G - u^o_G\right)^2\left(72|d|^2 - 108d_R^2\right), \qquad \beta = 1 - 12|d|^2\left(u^a_G - u^o_G\right)^2, \tag{5c}$$

where the first $\pm$ and the third one in (5a) are necessarily the same. For our parameters there is only one combination of $\pm$ in (5a) for which $|\widetilde{x}|$ is real and non-negative. There is hence only one steady solution of (3).

Finally, we recover $\widetilde{x}$ from (4): $\widetilde{x} = \frac{u^a_G - u^o_G}{1 - d|\widetilde{x}|}$, then $A^e = \frac{C_D\rho_a}{\rho_o\nu_o d}\left(\widetilde{x} - (u^a_G - u^o_G)\right)$ and $B^e = \frac{\rho_o\nu_o}{\rho_a\nu_a}A^e$. The steady state U^e_j is given by

$$\begin{aligned} U^e_{o,-m-1/2} &= u^o_G - \frac{\nu_o\lambda^e_o}{\mathrm{i}f h_o}(1 + \lambda^e_o)^m A^e, \\ U^e_{a,m+1/2} &= u^a_G + \frac{\nu_a\lambda^e_a}{\mathrm{i}f h_a}(1 + \lambda^e_a)^m B^e. \end{aligned} \tag{6}$$

Fig. 1 shows that this analysis exactly fits the numerical solution.

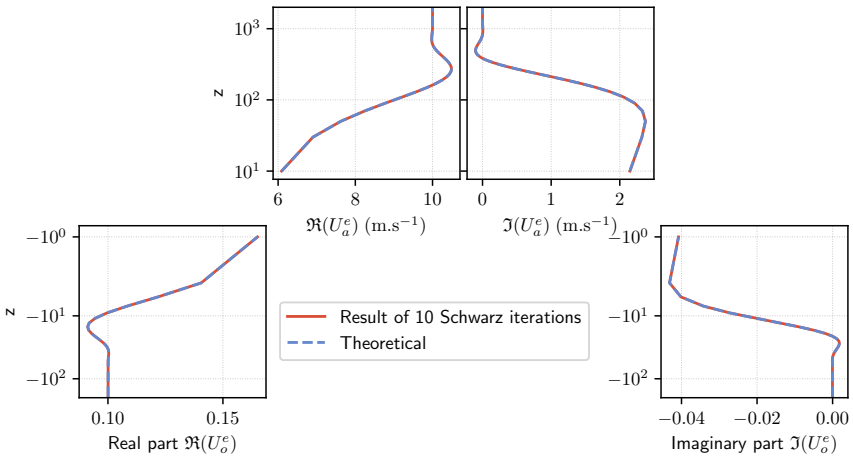


Fig. 1 Stationary solution profile in the ocean (bottom) and the atmosphere (top); A numerical steady state computed with 10 Schwarz iterations is the continuous red lines and the theoretical steady state obtained is displayed with dashed blue lines. Notice that the surface layer is not explicitly computed.

3.2 Existence of solutions of the nonlinear semi-discrete in space problem

The method used by [2] to prove the existence and unicity of a solution in the neighborhood of a steady state can be used to deal with several types of nonlinearities. In particular, we can prove the existence and uniqueness of a solution to the problem (3) close to the steady state (i.e. with initial condition, boundary conditions and nudging terms close to U_0, U_j^∞, u_G^j), thanks to the following steps:

1. The existence of a steady state U^e is discussed in §3.1.
2. The well-posedness of the coupled problem with the linearized transmission conditions is proven in Appendix 6.A of [3].
3. The use of the inverse function theorem can be done in four steps:

   a. concatenate the state vectors in a single vector $\mathbf{U} = \{U_a, U_o, \phi_a|_{z=0}\} \in \mathcal{U}$ where $\mathcal{U} = (L^2([0,T]))^{M_a+M_o+1}$ and M_a, M_o are the number of grid levels in the subdomains. The functions $\phi_a|_{z\neq 0}$ and ϕ_o are not in $\mathbf{U}$ because they can be expressed as linear combinations of elements of $\mathbf{U}$.

   b. Define a mapping $\mathbf{\Phi} : \mathcal{U} \to \mathbf{Y}$ such that

$$\begin{aligned}\mathbf{\Phi}(\mathbf{U}) = \{&(\partial_t + \mathrm{i}f)U_a - \nu_a \partial_z \phi_a - g_a,\quad (\partial_t + \mathrm{i}f)U_o - \nu_o \partial_z \phi_o - g_o,\\ &U_a(H_a,t) - U_a^\infty,\quad U_o(H_o,t) - U_o^\infty,\quad U_a|_{t=0} - U_a^e,\quad U_o|_{t=0} - U_o^e,\\ &\nu_a\ \phi_a|_{z=0} - C_D \left|U_{a,\frac{1}{2}} - U_{o,-\frac{1}{2}}\right| \left(U_{a,\frac{1}{2}} - U_{o,-\frac{1}{2}}\right)\},\end{aligned} \tag{7}$$

where $\partial_z \phi_j$ is to be understood in the finite difference sense. Let us draw some important remarks about $\mathbf{\Phi}$:

- The equation $\phi_{o,0} = \frac{\rho_a \nu_a}{\rho_o \nu_o} \phi_{a,0}$ at interface is implicit: in (7), $\partial_z \phi_o$ at the first grid level is $\frac{\rho_a \nu_a}{\rho_o \nu_o} \phi_{a,0} - \frac{1}{h_o} \left(U_{o,-\frac{1}{2}} - U_{o,-\frac{3}{2}} \right)$.
- The function $\mathbf{\Phi}$ is such that $\mathbf{\Phi}(\mathbf{U}^e) = 0$ where $\mathbf{U}^e$ is the steady state;
- The codomain $\mathbf{Y}$ is

$$\mathbf{Y} = (L^2([0,T]))^{M_a - 1} \times (L^2([0,T]))^{M_o - 1} \times (L^2([0,T]))^3 \times \mathbb{R}^{M_a + M_o}. \quad (8)$$

- Finding $\mathbf{\Phi}^{-1}(y)$ is equivalent to solving the nonlinear semi-discrete problem (3) if the component of y corresponding to the interface condition is zero (the other components correspond to other forcing terms, boundary conditions and initial condition). The idea of the proof is that if $\mathbf{\Phi}$ is invertible around $\mathbf{U}^e$ then the nonlinear semi-discrete problem (3) is invertible. Moreover, the inverse function theorem also tells us that $\mathbf{\Phi}^{-1}$ is continuous: this means that around the equilibrium state, the problem (3) is well-posed: it has a unique solution that depends continuously on the initial data.

c. Prove that $\mathbf{\Phi}$ is C^1 in a neighborhood of $\mathbf{U}^e$. The function $\mathbf{\Phi}$ is linear except for the transmission condition. Besides, the nonlinearity in this transmission condition is the function $x \mapsto |x|x$, which is continuously differentiable in a ball that does not contain zero. It is then straightforward to show that $\mathbf{\Phi}$ is C^1 and that its differential $D\mathbf{\Phi}(\mathbf{U}^e)$ is given by the linearized problem (a rigourous proof that can be directly adapted here is given in [2]).

d. Prove that $D\mathbf{\Phi}(\mathbf{U}^e)$ is an isomorphism: this is where the well-posedness of the coupled problem with linearized transmission conditions intervenes. The differential $D\mathbf{\Phi}(\mathbf{U}^e)$ corresponds indeed to the linearized problem with additional input data.

The next section uses the idea of considering the linearized problem around the steady state in the context of examining the convergence of a SWR algorithm.

4 Convergence analysis

In this section we conduct a convergence analysis of the SWR algorithm applied to the coupled problem (initial and boundary conditions are similar to (3) and omitted):

$$(\partial_t + \mathrm{i}f) U^k_{j,m+\frac{1}{2}} = \nu_j \frac{\phi^k_{j,m+1} - \phi^k_{j,m}}{h_j} + \mathrm{i}f u^j_G \,, \quad (9a)$$

$$\nu_a \phi^k_{a,0} = C_D \left| U^{k-1}_{a,\frac{1}{2}} - U^{k-1}_{o,-\frac{1}{2}} \right| \left(U^{k-1+\theta}_{a,\frac{1}{2}} - U^{k-1}_{o,-\frac{1}{2}} \right), \quad (9b)$$

$$\rho_o \nu_o \phi^k_{o,0} = \rho_a \nu_a \phi^k_{a,0}, \quad (9c)$$

where k is an iteration index and $U^{k-1+\theta}_{a,1/2} = (1-\theta)U^{k-1}_{a,1/2} + \theta U^{k}_{a,1/2}$. Here $\theta > 0$ is a relaxation parameter: the goal of the convergence analysis pursued in this section is to find an adequate value of θ leading to a fast convergence. The time windows is also assumed to be of infinite size: this hypothesis is necessary to carry a Fourier analysis of the convergence; finite time windows were handled at the semi-discrete in time level [1] but the method is not straightforward to extend to discretized equations in space.

The analysis of the SWR algorithm with the nonlinear transmission condition cannot be directly pursued through a Fourier transform. We thus consider the linearization of the problem around a steady state U^e_j, ϕ^e_j defined in (6). Assuming that U^k_j is in a neighborhood of U^e_j, the modulus in (9b) is non-zero and is thus differentiable (close to zero, one could also smooth the modulus). Differences with the steady state at the interface are noted $\delta\phi^k_j = \phi^k_{j,0} - \phi^e_{j,0}$, $\delta U^k_a = U^k_{a,\frac{1}{2}} - U^e_{a,\frac{1}{2}}$ and $\delta U^k_o = U^k_{o,-\frac{1}{2}} - U^e_{o,-\frac{1}{2}}$ (note that we omit the $\frac{1}{2}$ in δU^k_j).

The linearized transmission operator involves the complex conjugate $\overline{\delta U^{k-1}_j}$:

$$\nu_a \delta\phi^k_a = \alpha^e \left(\left(\frac{3}{2} - \theta \right) \delta U^{k-1}_a + \theta\, \delta U^k_a - \frac{3}{2}\delta U^{k-1}_o + \frac{\mathcal{R}^e}{2} \overline{\delta U^{k-1}_a - \delta U^{k-1}_o} \right) \tag{10}$$

with $\alpha^e = C_D \left| U^e_{a,1/2} - U^e_{o,-1/2} \right|$ and $\mathcal{R}^e = \frac{U^e_{a,1/2} - U^e_{o,-1/2}}{\overline{U^e_{a,1/2} - U^e_{o,-1/2}}}$. The relation (10) is used in the convergence analysis instead of (9b).

We now follow [6] to derive a convergence factor of the SWR method applied to the linearized transmission condition. It yields notably that the Fourier transform of $U^k_{a,m+1/2}$ is $\widehat{U}^k_{a,m+1/2} = B_k(\lambda_a + 1)^m$ with $\lambda_j = \frac{1}{2}(\widetilde{\chi}_j - \sqrt{\widetilde{\chi}_j}\sqrt{\widetilde{\chi}_j + 4})$ and $\widetilde{\chi}_j = \frac{(f+\omega)\mathrm{i}h_j^2}{\nu_j}$. We find that the evolution of B_k is:

$$B_{k+1}(\omega) = a_1(\omega)B_k(\omega) + a_2(\omega)\overline{B_k(-\omega)}, \quad \text{where} \quad a_1 = \alpha^e \frac{\frac{3}{2}\mu(\omega) - \theta}{\frac{\nu_a}{h_a}(\lambda_a - \widetilde{\chi}_a) - \alpha^e\theta},$$

$$a_2 = \alpha^e \frac{\frac{\mathcal{R}^e}{2}\overline{\mu(-\omega)}}{\frac{\nu_a}{h_a}(\lambda_a - \widetilde{\chi}_a) - \alpha^e\theta} \quad \text{and} \quad \mu(\omega) = 1 - \frac{\lambda_a - \widetilde{\chi}_a}{\lambda_o}\epsilon\frac{\nu_a h_o}{\nu_o h_a}.$$

Note that the variable $\overline{\mathrm{i}\omega} = -\mathrm{i}\omega$ appears when using the Fourier transform on $\overline{\delta U^{k-1}_a - \delta U^{k-1}_o}$. As a consequence, the convergence factor ξ^q in the linearized quadratic friction case differs from one iteration to another: it is a function of $\frac{\overline{B_{k-1}(-\omega)}}{B_{k-1}(\omega)}$. We need to examine the evolution of both $B_{k+1}(\omega), \overline{B_{k+1}(-\omega)}$:

$$\begin{pmatrix} B(\omega) \\ \overline{B(-\omega)} \end{pmatrix}_{k+1} = \mathbf{M} \begin{pmatrix} B(\omega) \\ \overline{B(-\omega)} \end{pmatrix}_{k}, \qquad \mathbf{M} = \begin{pmatrix} a_1(\omega) & a_2(\omega) \\ \overline{a_2(-\omega)} & \overline{a_1(-\omega).} \end{pmatrix} \tag{11}$$

The singular values (shown in Fig. 2) of $\mathbf{M}$ can be studied instead of the convergence factor. One can see on Fig. 2 that its two singular values $\widetilde{\xi}_1$ and $\widetilde{\xi}_2$ are different for small frequencies, especially around the frequencies f and $-f$.

One can hence expect that for frequencies close to f and $-f$ the "convergence factor" $\sqrt{\frac{|B_k(\omega)|^2+|B_k(-\omega)|^2}{|B_{k-1}(\omega)|^2+|B_{k-1}(-\omega)|^2}}$ will be different from one iteration to another and will be between $\widetilde{\xi_1}$ and $\widetilde{\xi_2}$.

We optimize only the maximum over the frequencies of the largest singular value $\widetilde{\xi_1}$ (see Fig. 3) and find that the optimal value of θ is slightly smaller than 1.5.

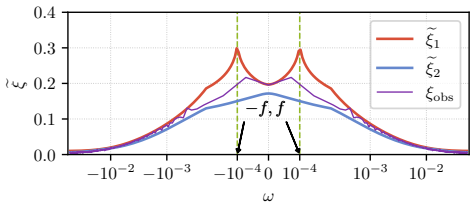


Fig. 2 Singular values $\widetilde{\xi_1}(\omega)$, $\widetilde{\xi_2}(\omega)$ of $\mathbf{M}$. The observed "convergence factor" at the first iteration $\xi_{\text{obs}} = \sqrt{\frac{|B_2(\omega)|^2+|B_2(-\omega)|^2}{|B_1(\omega)|^2+|B_1(-\omega)|^2}}$ is in purple. The numerical validation ξ_{obs} fits the convergence analysis since $\widetilde{\xi_2} \leq \xi_{\text{obs}} \leq \widetilde{\xi_1}$:

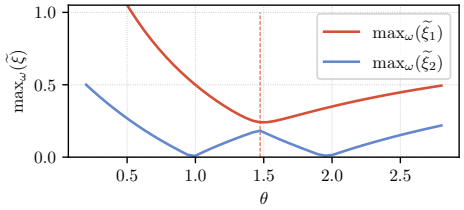


Fig. 3 Singular values $\widetilde{\xi_1}$, $\widetilde{\xi_2}$ of $\mathbf{M}$ maximized over the set of discrete frequencies $\{-\frac{\pi}{\Delta t}, \ldots, \frac{\pi}{T}, 0, \frac{\pi}{T}, \ldots, \frac{\pi}{\Delta t}\}$ as a function of θ. The vertical dashed line highlights the minimum of $\max \widetilde{\xi_1}$. The windows length T is one day and $\Delta t = 60$ s.

It is also seen in Fig. 3 that for $\theta < \frac{1}{2}, \widetilde{\xi_1} > 1$, which means that the SWR algorithm does not converge. On the contrary, for θ larger than $\frac{1}{2}$ both singular values are smaller than one for all the frequencies, and the convergence factor is then bounded by $\widetilde{\xi_1}$.

5 Numerical experiments

Parameters of the numerical experiments are $C_D = 1.2\times10^{-3}$, $h_a = 20$ m, $h_o = 2$ m, $H_o = H_a = 2000$ m. The Coriolis parameter is $f = 10^{-4}$ s^{-1} and the diffusivities are $\nu_a = 1$ m^2 s^{-1}, $\nu_o = 3 \times 10^{-3}$ m^2 s^{-1}. The boundary conditions and nudging terms $U_j^\infty = u_G^j$ are set to constant values of 10 m s^{-1} in the atmosphere and 0.1 m s^{-1} in the ocean, while the initial condition is the steady state $U_0(z) = U_j^e(z)$. SWR is initialized at the interface with a white noise around the interface value of the steady state. Fig. 1 and Fig. 2 show that with those parameters, the theoretical results are coherent with the numerical experiments. Moreover Fig. 4 shows the evolution of the

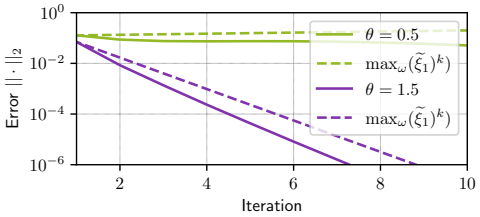


Fig. 4 Evolution of the L^2 norm of the errors for $\theta = \frac{1}{2}$ (green) and $\theta = \frac{3}{2}$ (purple). The singular value $\widetilde{\xi_1}$ gives an upper bound of the error represented by dashed lines.

error for two choices of θ. As it is expected the choice $\theta = \frac{1}{2}$ leads to a convergence rate of approximately 1 whereas a relatively fast convergence is obtained with $\theta = \frac{3}{2}$.

References

1. Arnoult, A., Japhet, C., and Omnes, P. Discrete-time analysis of Schwarz waveform relaxation convergence (2022). ⟨hal-03746438⟩.
2. Chacon-Rebollo, T., Gomez-Marmol, M., and Rubino, S. On the existence and asymptotic stability of solutions for unsteady mixing-layer models. *Discrete Contin. Dyn. Syst.* **34**(2), 421–436 (2014).
3. Clement, S. *Numerical analysis for a combined space-time discretization of air-sea exchanges and their parameterizations*. Phd thesis, UGA (2022). ⟨tel-03822632⟩.
4. Marti, O., Nguyen, S., Braconnot, P., Valcke, S., Lemarié, F., and Blayo, E. A Schwarz iterative method to evaluate ocean–atmosphere coupling schemes: implementation and diagnostics in IPSL-CM6-SW-VLR. *Geosci. Model Dev.* **14**(5), 2959–2975 (2021).
5. Thery, S., Pelletier, C., Lemarié, F., and Blayo, E. Analysis of Schwarz waveform relaxation for the coupled Ekman boundary layer problem with continuously variable coefficients. *Numer. Algorithms* **89**, 1145–1181 (2021).
6. Wu, S.-L. and Al-Khaleel, M. D. Semi-discrete Schwarz waveform relaxation algorithms for reaction diffusion equations. *BIT Numer. Math.* **54**(3), 831–866 (2014).

A Block Jacobi Sweeping Preconditioner for the Helmholtz Equation

Ruiyang Dai

1 Introduction

Solving Helmholtz problems using numerical methods is challenging due to the large, indefinite, and ill-conditioned linear systems that result, which cannot be solved using classical direct or iterative solvers [5]. While optimized Schwarz (OS) methods have been proposed as an alternative, the number of iterations required by Krylov methods increases with the number of subdomains, especially for layer-type domain decompositions [9]. Preconditioners, such as sweeping preconditioners, are necessary when using iterative methods to solve Helmholtz problems. Lately, there has been significant interest in sweeping preconditioners, invented by [3, 4], that achieve quasi-linear asymptotic complexity. Despite their effectiveness, sweeping preconditioners face challenges with parallel scalability due to the inherently sequential nature of their operations, as well as the need to ensure accurate and consistent information transfer between subdomains. These challenges can restrict the use of layer-type domain decompositions.

To address these challenges, recent research has focused on improving parallel performance through new sweeping strategies on checkerboard domain decompositions that can handle more general domain decompositions. Several sweeping algorithms have been proposed that improve parallelism by ensuring consistent transfer among subdomains, such as L-sweeps preconditioners [11], trace transfer-based diagonal sweeping preconditioners [7], and multidirectional sweeping preconditioners [2], with high-order transmission conditions and cross-point treatments [8].

Subdomains in sweeping algorithms can be assigned to Message Passing Interface (MPI) ranks based on rows or columns. This enables parallel application of sweeping algorithms for a single right-hand side. However, these approaches still have limitations, including long preconditioning procedures, waste of computation resources, and relatively high computation costs. To overcome these limitations, the

Ruiyang Dai
Laboratory J.L. Lions, Sorbonne Université, Paris, France, e-mail: ruiyang.dai@upmc.fr

authors propose a block Jacobi sweeping preconditioner that uses block Jacobi matrices to decompose full sweeps into several partial sweeps, which can be thought of as sweeps that operate on a subset of the subdomains. These partial sweeps can be performed concurrently. This approach enhances scalability and makes full use of resources on parallel computer architectures.

2 Notations

Let $\mathbf{i} = (i_1, i_2) \in \mathbb{N}^2$ be a multi-index denoting the subdomain number. We define the discrete l^1 norm by: $|\mathbf{i}|_1 := |i_1| + |i_2|$. We use the convention that two multi-indices $\mathbf{i}$ and $\mathbf{j}$ are equal if and only if $i_1 = j_1$ and $i_2 = j_2$.

Definition 1 The *lexicographic order* on multi-indices is the relation defined by $\mathbf{i} < \mathbf{j}$ if and only if $|\mathbf{i}|_1 < |\mathbf{j}|_1$, or $|\mathbf{i}|_1 = |\mathbf{j}|_1$ and $i_1 < j_1$.

Definition 2 The *lexicographic order* on pair multi-indices $(\mathbf{i}, \mathbf{j}) \in \mathbb{N}^2 \times \mathbb{N}^2$ is the relation defined by $(\mathbf{i}, \mathbf{j}) < (\mathbf{k}, \mathbf{l})$ if and only if $\mathbf{i} < \mathbf{k}$, or $\mathbf{i} = \mathbf{k}$ and $\mathbf{j} < \mathbf{l}$.

We define a function m that maps a pair of multi-indices which are in lexicographic order to natural numbers in a monotonically increasing fashion $m\colon \mathbb{N}^2 \times \mathbb{N}^2 \to \mathbb{N}$, such that $m((1,1),(1,2)) = 1$, $m((1,1),(2,1)) = 2$, $m((1,2),(1,1)) = 3$, etc.

We consider $\Omega \subset \mathbb{R}^2$ be a square domain with boundary $\partial\Omega$, which is given by the union of the scattered boundary $\partial\Omega^{\text{sca}}$ with the external artificial boundaries Γ_i^{∞} for $i = 1, 2, 3, 4$, and its a non-overlapping checkerboard partition, which consists in a lattice of rectangular non-overlapping subdomains $\Omega_{\mathbf{i}}$ with N_1 columns and N_2 rows ($i_1 = 1, \ldots, N_1$, and $i_2 = 1, \ldots, N_2$), that is

$$\overline{\Omega} = \bigcup \overline{\Omega}_{\mathbf{i}}, \quad \text{and} \quad \Omega_{\mathbf{i}} \cap \Omega_{\mathbf{j}} = \varnothing \quad \text{for} \quad \mathbf{j} \neq \mathbf{i}.$$

And we say that $\exists \mathbf{i}$, such that $\Omega^{\text{sca}} \subseteq \Omega_{\mathbf{i}}^{\circ}$ and $\partial\Omega^{\text{sca}} \cap \partial\Omega_{\mathbf{i}} = \varnothing$. The boundary of a subdomain $\Omega_{\mathbf{i}}$ is split into two parts: the exterior part $\partial\Omega_{\mathbf{i}} \cap \Gamma_i^{\infty}$ and the interior part including decomposed interior interfaces $\Gamma_{\mathbf{i},\mathbf{j}} := \partial\Omega_{\mathbf{i}} \cap \partial\Omega_{\mathbf{j}}$ ($\mathbf{j} \neq \mathbf{i}$), and $\Gamma_{\mathbf{i},\mathbf{j}} = \Gamma_{\mathbf{j},\mathbf{i}}$. There are $N_{\text{dom}} = N_1 \times N_2$ subdomains, $N_e = 2N_1N_2 - N_1 - N_2$ interior interfaces. We define the number of diagonal groups $N_g := N_1 + N_2 - 1$.

3 Non-overlapping domain decomposition method

We study the 2D Helmholtz equation in Ω with an absorbing boundary condition on Γ_i^{∞}. For a more detailed description, see [2]. We seek the field $u(\mathbf{x})$ that verifies

$$\begin{cases} (-\Delta - \kappa^2)u = 0, & \text{in } \Omega, \\ (\partial_{\boldsymbol{n}_i} - \mathscr{T})u = 0, & \text{on } \Gamma_i^{\infty}, \\ u = -u^{\text{inc}}, & \text{on } \partial\Omega^{\text{sca}}, \end{cases} \tag{1}$$

where κ is the wavenumber, u^{inc} is the incident wave, $\partial_{\boldsymbol{n}}$ is the exterior normal derivative, and $\mathscr{T}$ is an impedance operator to be defined. We take the convention that the time-dependence of the fields is $e^{-\imath\omega t}$, where ω is the angular frequency and t is the time.

The domain decomposition method consists in considering the N_{dom} local subproblems coupled by the Robin conditions: Seek the field $u_{\mathbf{i}}(\mathbf{x})$ that verifies

$$\begin{cases} (-\Delta - \kappa^2)u_{\mathbf{i}} = 0, & \text{in } \Omega_{\mathbf{i}}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},i}} - \mathscr{T})u_{\mathbf{i}} = 0, & \text{on } \partial\Omega_{\mathbf{i}} \cap \Gamma_i^{\infty}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{j}}} - \mathscr{T})u_{\mathbf{i}} = (-\partial_{\boldsymbol{n}_{\mathbf{j},\mathbf{i}}} - \mathscr{T})u_{\mathbf{j}}, & \text{on } \Gamma_{\mathbf{i},\mathbf{j}}, \forall \mathbf{j} \in D_{\mathbf{i}}, \\ u_{\mathbf{i}} = -u^{\text{inc}}, & \text{on } \partial\Omega_{\mathbf{i}} \cap \partial\Omega^{\text{sca}}, \end{cases} \tag{2}$$

where the set $D_{\mathbf{i}} := \left\{\mathbf{j} \,|\, \mathbf{j} \neq \mathbf{i} \text{ and } \Gamma_{\mathbf{i},\mathbf{j}} \neq \varnothing\right\}$. The paper uses high-order absorbing boundary conditions (HABCs) as transmission conditions, which are effective for both layered-type and checkerboard-type domain decompositions [1, 8]. However, special treatment is required at corners in 2D cases for polygonal domains. Section 4 of the paper employs HABCs, but in the next section, the paper uses less effective boundary conditions based on the basic impedance operator to investigate the algebraic structure of the interface problem for clarity.

4 Interface problem

To derive the interface problem, let's introduce $w_{\mathbf{i}}(\mathbf{x})$ a lifting of the source: Seek $w_{\mathbf{i}}(\mathbf{x})$ that verifies

$$\begin{cases} (-\Delta - \kappa^2)w_{\mathbf{i}} = 0, & \text{in } \Omega_{\mathbf{i}}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},i}} - \mathscr{T})w_{\mathbf{i}} = 0, & \text{on } \partial\Omega_{\mathbf{i}} \cap \Gamma_i^{\infty}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{j}}} - \mathscr{T})w_{\mathbf{i}} = 0, & \text{on } \Gamma_{\mathbf{i},\mathbf{j}}, \forall \mathbf{j} \in D_{\mathbf{i}}, \\ u_{\mathbf{i}} = -u^{\text{inc}}, & \text{on } \partial\Omega_{\mathbf{i}} \cap \partial\Omega^{\text{sca}}, \end{cases} \tag{3}$$

By the linearity of the problem, the field $u_{\mathbf{i}}$ can be decomposed into $v_{\mathbf{i}} + w_{\mathbf{i}}$, where $v_{\mathbf{i}}$ is the field (2) after lifting the sources by (3). We introduce the local scattering operator $\mathscr{S}_{m(\mathbf{j},\mathbf{i}),m(\mathbf{i},\mathbf{k})} : x_{m(\mathbf{i},\mathbf{k})} \to (-\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{j}}} - \mathscr{T})v_{\mathbf{i}}$ where

$$\begin{cases} (-\Delta - \kappa^2)v_{\mathbf{i}} = 0, & \text{in } \Omega_{\mathbf{i}}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},i}} - \mathscr{T})v_{\mathbf{i}} = 0, & \text{on } \partial\Omega_{\mathbf{i}} \cap \Gamma_i^{\infty}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{k}}} - \mathscr{T})v_{\mathbf{i}} = x_{m(\mathbf{i},\mathbf{k})}, & \text{on } \Gamma_{\mathbf{i},\mathbf{k}}, \\ (\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{l}}} - \mathscr{T})v_{\mathbf{i}} = 0, & \text{on } \Gamma_{\mathbf{i},\mathbf{l}}, \forall \mathbf{l} \neq \mathbf{k}, \end{cases} \tag{4}$$

and $\mathbf{j}, \mathbf{k}, \mathbf{l} \in D_{\mathbf{i}}$. Using the linearity of the problem and the above scattering operator, we obtain the interface problem

$$(\partial_{\boldsymbol{n}_{\mathbf{j},\mathbf{i}}} - \mathscr{T})v_{\mathbf{j}} = \sum_{\mathbf{k}\in D_{\mathbf{i}}} \mathscr{S}_{m(\mathbf{j},\mathbf{i}),m(\mathbf{i},\mathbf{k})}(\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{k}}} - \mathscr{T})v_{\mathbf{i}} + (-\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{j}}} - \mathscr{T})w_{\mathbf{i}}, \quad \mathbf{j} \in D_{\mathbf{i}}.$$

We introduce the global scattering matrix $S \in M_{2N_e}(\mathscr{S}_{m(\mathbf{j},\mathbf{i}),m(\mathbf{i},\mathbf{k})})$, the global additional variable vector $g \in M_{2N_e\times 1}(g_{m(\mathbf{j},\mathbf{i})})$, and the global right-hand-side vector $b \in M_{2N_e\times 1}(b_{m(\mathbf{j},\mathbf{i})})$, where

$$g_{m(\mathbf{j},\mathbf{i})} = (+\partial_{\boldsymbol{n}_{\mathbf{j},\mathbf{i}}} - \mathscr{T})v_{\mathbf{j}}, \quad b_{m(\mathbf{j},\mathbf{i})} = (-\partial_{\boldsymbol{n}_{\mathbf{i},\mathbf{j}}} - \mathscr{T})w_{\mathbf{i}}.$$

We obtain that g is the solution to the global matrix

$$(I - S)g = b, \tag{5}$$

or

$$g_{m(\mathbf{j},\mathbf{i})} - \sum_{\mathbf{k}} \mathscr{S}_{m(\mathbf{j},\mathbf{i}),m(\mathbf{i},\mathbf{k})} g_{m(\mathbf{i},\mathbf{k})} = b_{m(\mathbf{j},\mathbf{i})}, \quad \forall \mathbf{j}, \text{ for } \mathbf{i},\mathbf{k} \in D_{\mathbf{j}}. \tag{6}$$

5 Sweeping preconditioner

Let V_i be the $2N_e \times n_i$ matrix $V_i = (e_{m(\mathbf{i},\mathbf{j})})$, with $|\mathbf{i}|_1 = i$ $(i = 2, \ldots, N_g + 1)$, where each $e_{m(\mathbf{i},\mathbf{j})}$ is the $m(\mathbf{i},\mathbf{j})$-th column of the $2N_e \times 2N_e$ identity matrix, and n_i is the number of columns. One has $V_i^\top V_j = \mathbf{0}$, if $i \neq j$. Let $S_{i,j}$ be the $n_i \times n_j$ matrix $S_{i,j} = V_i^\top S V_j$.

Proposition 1 *The upper and lower triangular matrix of the global matrix $(I - S)$ can be decomposed by Gaussian elimination.*

Proof We denote the lower triangular matrix of the global matrix (5) S_L. Consider the following matrix $\prod_i (I - V_i S_{i,i-1} V_{i-1}^\top)$, $\quad i = 3, \ldots, N_g + 1$. Then, $\forall i = 3, \ldots, N_g$, we have

$$\begin{aligned}
&(I - V_i S_{i,i-1} V_{i-1}^\top)(I - V_{i+1} S_{i+1,i} V_i^\top) \\
&= I - V_i S_{i,i-1} V_{i-1}^\top - V_{i+1} S_{i+1,i} V_i^\top + V_i S_{i,i-1} V_{i-1}^\top V_{i+1} S_{i+1,I} V_i^\top \\
&= I - V_i S_{i,i-1} V_{i-1}^\top - V_{i+1} S_{i+1,i} V_i^\top
\end{aligned}$$

The last term at the 2nd line vanishs since $V_{i-1}^\top V_{i+1}$ is null. Hence, we have

$$\prod_i (I - V_i S_{i,i-1} V_{i-1}^\top) = I - \sum_i V_i S_{i,i-1} V_{i-1}^\top = I - S_L.$$

Similarily, we can proof that the upper triangular matrix of the global matrix (5) S_U can be decomposed as

$$I - S_U = \prod_i (I - V_{i-1} S_{i-1,i} V_i^\top), \quad i = N_g + 1, \ldots, 3,$$

which is a series of matrices. □

Next, we present the Symmetric Gauss-Seidel (SGS) sweeping preconditioner P_{SGS}. This matrix can then be rewritten as $P_{\mathrm{SGS}} \approx (I - S_L)(I - S_U)$, We can easily invert the matrix $P_{\mathrm{SGS}}^{-1} = (I - S_U)^{-1}(I - S_L)^{-1}$, with

$$\begin{aligned}(I - S_U)^{-1} &= \prod_i (I + V_{i-1} S_{i-1,i} V_i^\top), \quad i = 3, \ldots, N_g + 1, \\ (I - S_L)^{-1} &= \prod_i (I + V_i S_{i,i-1} V_{i-1}^\top), \quad i = N_g + 1, \ldots, 3.\end{aligned} \tag{7}$$

Observing Eqn. (7), we notice that it consists of a sequential process, in which there are $2(N_g - 1)$ sequential *steps* in total.

6 Block Jacobi sweeping preconditioner

Let W_{Li}, W_{Ui} be the $2N_e \times s_i$ matrix

$$\begin{aligned}W_{Li} &= (e_{m(\mathbf{i},\mathbf{j})}), \quad 2 + (i-1)N_1 \le |\mathbf{i}|_1 \le 2 + iN_1, \\ W_{Ui} &= (e_{m(\mathbf{i},\mathbf{j})}), \quad (1 + N_g) - iN_1 \le |\mathbf{i}|_1 \le (1 + N_g) - (i-1)N_1,\end{aligned}$$

where s_i is the number of columns $(e_{m(\mathbf{i},\mathbf{j})})$. Let S_{Li}, S_{Ui} be the $s_i \times s_i$ matrix

$$S_{Li} = W_{Li}^\top S_L W_{Li}, \quad S_{Ui} = W_{Ui}^\top S_U W_{Ui}.$$

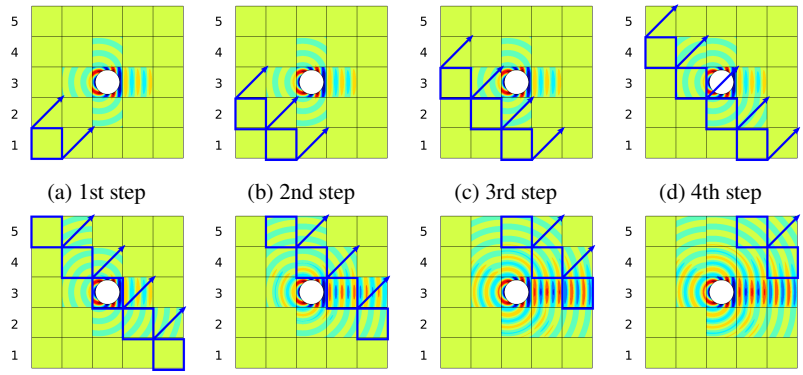
According to the *additive projection processes* [10], the next iterate can be defined as

$$\begin{aligned}g^{(k+1/2)} &= g^{(k)} + \sum_{i=1}^{p} W_{Li}(I_i - S_{Li})^{-1} W_{Li}^\top r^{(k)}, \\ g^{(k+1)} &= g^{(k+1/2)} + \sum_{i=1}^{p} W_{Ui}(I_i - S_{Ui})^{-1} W_{Ui}^\top r^{(k+1/2)}.\end{aligned}$$

$I - S_L$ and $I - S_U$ are quasi-equivalent to p blocks $I_i - S_{Li}$ and p blocks $I_i - S_{Ui}$, respectively, which form the forward and backward block Jacobi preconditioner. The block Jacobi preconditioner can be decomposed into a series of matrices, as stated in Proposition 1. The forward and backward block Jacobi preconditioner consists of the upper and lower block diagonals of $I - S$ and involves $2N_1$ sequential steps. This decomposition enhances the parallel performance of the sweeping preconditioner.

7 Numerical results

In this part, the block Jacobi sweeping preconditioner (BSP) is studied by considering a two-dimensional benchmark with a high-order finite element method and



(a) 1st step (b) 2nd step (c) 3rd step (d) 4th step

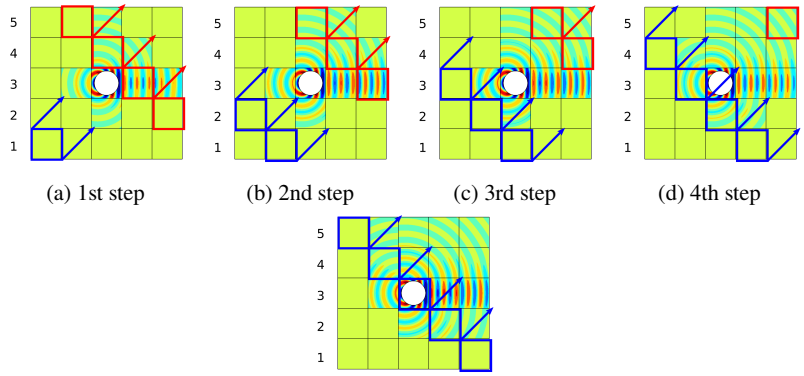
(e) 5th step (f) 6th step (g) 7th step (h) 8th step

Fig. 1 Scattering model in 2D ($k = 2\pi$) with a snapshot of the solution at different steps of the first GMRES iteration using the full sweeping preconditioner. Each row of subdomains is assigned to one MPI rank, and processors are identified by the numbers on the left side. Subdomains processed in parallel are highlighted in blue.

compared to the full sweeping preconditioner (SP). The proposed approaches and the computational results presented in this paper are implemented in parallel by MPI on a single multi-core computer. The linear systems arising from the sub-problems are solved by a sparse direct solver. The mesh generation, mesh decomposition, and post-processing are credited by Gmsh [6]. The parallelism of our approach is realized by assigning subdomains to MPI ranks in a row-based fashion such that the i-th row of the checkerboard partition is processed by rank i.

The test case is a homogeneous scattering problem in free space within a rectangle geometry ($\Omega = [-1.25, 2.50 \cdot N_1 - 1.25] \times [-1.25, 2.50 \cdot N_2 - 1.25]$), which is decomposed into $N_1 \times N_2$ rectangular subdomains. An incident plane wave is generated by a sound-soft circular cylinder of radius equal to 1 which is located at the Origin. On the circular cylinder, the Dirichlet boundary condition $u(\mathbf{x}) = -\exp^{ikx}$ is prescribed at the boundary of the sound-soft scatterer. The Padé-type HABC is prescribed on the exterior boundaries and the interior interfaces used as the absorbing boundary conditions and the transmission boundary conditions, respectively. The compatibility conditions are prescribed at the corners and the cross-points treatment is prescribed at the cross-points. The parameters of the HABC operator are $N_{\text{pade}} = 8$ and $\phi = \pi/3$. The following numerical setting are considered: $P7$ finite elements with 3 elements per wavelength ($h \approx 1/21$).

Figures 1 and 2 show snapshots of the solutions at different steps of forward sweep (sweep starts from the bottom-left corner to the top-right corner) of the 1st GMRES iteration with different sweeping preconditioners. Although the forward sweep in Fig. 1 goes through the whole computational domain from the bottom-left corner to the top-right corner, it takes 8 steps. If we take the backward sweep into account, there are 16 steps of the preconditioning procedure at each iteration. In the second situation, it only takes 5 steps in the forward partial sweeps (see Fig. 2).



(a) 1st step (b) 2nd step (c) 3rd step (d) 4th step

(e) 5th step

Fig. 2 Scattering model in 2D ($k = 2\pi$). Snapshot of the solution at different steps of forward sweep of 1st GMRES iteration with the block Jacobi sweeping preconditioner. The numbers at left side are processors' identities. Each row of subdomains is assigned to one MPI rank. Subdomains processed in parallel have same blue and red color, which represent two partial sweeps.

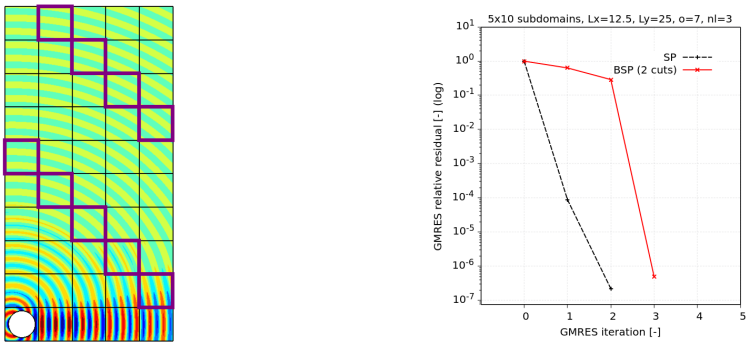


Fig. 3 Scattering model in 2D ($k = 2\pi$). The computational domain is decomposed into $N_1 \times N_2 = 5 \times 10$. Residual history with SP and BSP with two cuts. In this context, two cuts imply that $p = 3$.

Figure 3 shows snapshots of the solutions and residual histories of GMRES with the different preconditioners for the partition $N_1 \times N_2 = 5 \times 10$. All forward/backward (partial) sweeps of these preconditioners start from the bottom-left/top-left to the top-right/bottom-left. The violet boxes indicate the cut location which separates partial sweeps.

The residual histories obtained with the two different preconditioners in Fig. 3, where the relative residual suddenly drops in residual history at the first iteration when a full sweeping preconditioner is used. With the block Jacobi sweeping preconditioner used, it happens at the third iteration, which corresponds to the number of partial sweeps, that is to say, there are two partial sweeps.

Table 1 Scattering model in 2D ($k = 20\pi$). Number of iterations and runtime in seconds with the two different preconditioners for different domain partitions. "ni" stands for the number of iterations, "ns" the number of steps per iteration, and "t" time. The number of MPI ranks is equal to N_2.

$N_1 \times N_2$	SP (ni)	SP (ns)	SP (t)	BSP (ni)	BSP (ns)	BSP (t)
5×5	3	16	32.6 s	3	10	25.4 s
5×10	3	26	49.0 s	4	10	33.0 s
5×15	4	36	90.8 s	6	10	51.3 s
5×20	5	46	147.4 s	8	10	70.9 s

The number of GMRES iterations and the runtime to reach a relative residual 10^{-6} with the two different preconditioners are given in Table 1. The runtime corresponds to the GMRES resolution phase.

References

1. Boubendir, Y., Antoine, X., and Geuzaine, C. A quasi-optimal non-overlapping domain decomposition algorithm for the helmholtz equation. *Journal of Computational Physics* **231**(2), 262–280 (2012).
2. Dai, R., Modave, A., Remacle, J.-F., and Geuzaine, C. Multidirectionnal sweeping preconditioners with non-overlapping checkerboard domain decomposition for helmholtz problems. *Journal of Computational Physics* **453**, 110887 (2022).
3. Engquist, B. and Ying, L. Sweeping preconditioner for the helmholtz equation: hierarchical matrix representation. *Communications on Pure and Applied Mathematics* **64**(5), 697–735 (2011).
4. Engquist, B. and Ying, L. Sweeping preconditioner for the helmholtz equation: moving perfectly matched layers. *Multiscale Modeling & Simulation* **9**(2), 686–710 (2011).
5. Ernst, O. G. and Gander, M. J. Why it is difficult to solve helmholtz problems with classical iterative methods. In: *Numerical Analysis of Multiscale Problems*, 325–363. Springer (2011).
6. Geuzaine, C. and Remacle, J.-F. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* **79**(11), 1309–1331 (2009).
7. Leng, W. and Ju, L. Trace transfer-based diagonal sweeping domain decomposition method for the helmholtz equation: Algorithms and convergence analysis. *Journal of Computational Physics* **455**, 110980 (2022).
8. Modave, A., Royer, A., Antoine, X., and Geuzaine, C. A non-overlapping domain decomposition method with high-order transmission conditions and cross-point treatment for helmholtz problems. *Computer Methods in Applied Mechanics and Engineering* **368**, 113162 (2020).
9. Nataf, F., Rogier, F., and de Sturler, E. Optimal interface conditions for domain decomposition methods. Tech. rep., CMAP Ecole Polytechnique (1994).
10. Saad, Y. *Iterative Methods for Sparse Linear Systems*. SIAM (2003).
11. Taus, M., Zepeda-Núñez, L., Hewett, R. J., and Demanet, L. L-sweeps: A scalable, parallel preconditioner for the high-frequency helmholtz equation. *Journal of Computational Physics* **420**, 109706 (2020).

Optimized Neumann-Neumann Method for the Stokes-Darcy Problem

Marco Discacciati and Jake Robinson

1 Introduction and problem setting

The Stokes-Darcy problem [9, 15] is a good example of multi-physics problem where splitting methods typical of domain decomposition naturally apply. The problem is defined in a computational domain formed by a fluid region Ω_f and a porous-medium region Ω_p that are non-overlapping and separated by an interface Γ. In Ω_f, an incompressible fluid with constant viscosity and density is modelled by the dimensionless Stokes equations:

$$-\boldsymbol{\nabla} \cdot (2\mu_f \nabla^s \mathbf{u}_f - p_f \boldsymbol{I}) = \mathbf{f}_f \,, \qquad \nabla \cdot \mathbf{u}_f = 0 \quad \text{in } \Omega_f \,, \tag{1}$$

where $\mu_f = Re^{-1}$, Re being the Reynolds number, $\mathbf{u}_f$ and p_f are the fluid velocity and pressure, $\boldsymbol{I}$ and $\nabla^s \mathbf{u}_f = \frac{1}{2}(\nabla \mathbf{u}_f + (\nabla \mathbf{u}_f)^T)$ are the identity and the strain rate tensor, and $\mathbf{f}_f$ is an external force. In the porous medium domain Ω_p, we consider the dimensionless Darcy's model:

$$-\nabla \cdot (\boldsymbol{\eta}_p \nabla p_p) = f_p \qquad \text{in } \Omega_p \,, \tag{2}$$

where p_p is the fluid pressure in the porous medium, $\boldsymbol{\eta}_p$ is the permeability tensor, and f_p is an external force. The two local problems are coupled through the classical Beaver-Joseph-Saffman conditions at the interface [1, 14, 17]:

$$\mathbf{u}_f \cdot \mathbf{n} = -(\boldsymbol{\eta}_p \nabla p_p) \cdot \mathbf{n} \quad \text{on } \Gamma, \tag{3}$$

$$-\mathbf{n} \cdot (2\mu_f \nabla^s \mathbf{u}_f - p_f \boldsymbol{I}) \cdot \mathbf{n} = p_p \quad \text{on } \Gamma, \tag{4}$$

$$-((2\mu_f \nabla^s \mathbf{u}_f - p_f \boldsymbol{I}) \cdot \mathbf{n})_\tau = \xi_f (\mathbf{u}_f)_\tau \quad \text{on } \Gamma, \tag{5}$$

Marco Discacciati, Jake Robinson
Department of Mathematical Sciences, Loughborough University, Epinal Way, Loughborough, LE11 3TU, United Kingdom, e-mail: m.discacciati@lboro.ac.uk

where $\xi_f = \alpha_{BJ} \left(\mu_f / (\boldsymbol{\tau} \cdot \boldsymbol{\eta}_p \cdot \boldsymbol{\tau})\right)^{1/2}$, α_{BJ} is the Beavers-Joseph constant, $\mathbf{n}$ denotes the unit normal vector pointing outward of Ω_f, while $(\mathbf{v})_\tau$ indicates the tangential component of any vector $\mathbf{v}$ at Γ. Finally, we impose $\mathbf{u}_f = \mathbf{0}$ on Γ_f^D, $(2\mu_f \nabla^s \mathbf{u}_f - p_f \boldsymbol{I}) \cdot \mathbf{n} = \mathbf{0}$ on Γ_f^N, $p_p = 0$ on Γ_p^D, $\mathbf{u}_p \cdot \mathbf{n}_p = 0$ on Γ_p^N, where $\Gamma_f^D \cup \Gamma_f^N = \partial\Omega_f \setminus \Gamma$ and $\Gamma_p^D \cup \Gamma_p^N = \partial\Omega_p \setminus \Gamma$.

Classical Dirichlet-Neumann type methods [16] for the Stokes-Darcy problem were studied in [7, 9, 10] where it was pointed out that their convergence can be slow for small values of the fluid viscosity and of the porous medium permeability. Robin-Robin methods were then proposed as an alternative [3, 4, 5, 6, 7, 11], and they were analysed in the framework of optimized Schwarz methods in [8, 12, 13].

In this work, we focus on a Neumann-Neumann approach that allows to solve a scalar interface problem like in the case of Dirichlet-Neumann methods. This reduces the number of interface unknowns compared to the system associated with Robin-Robin iterations, and it allows to use preconditioned conjugate gradient (PCG) iterations instead of the more expensive GMRES iterations used in the Robin-Robin context (see, e.g., [8]). However, to define effective Neumann-Neumann methods, the contribution of each subproblem must be suitably weighted. For single-physics problems, this is typically done using algebraic strategies that can take into account coefficient jumps across interfaces (see, e.g., [18]). However, no clear strategies are available for multi-physics problems. In this work, we extend techniques for the analysis of optimized Schwarz methods with the aim of characterizing optimal weighting parameters to define a robust Neumann-Neumann preconditioner.

2 Optimized Neumann-Neumann method

Let α_f and α_p be two positive parameters: $\alpha_f, \alpha_p \in \mathbb{R}$, $\alpha_f, \alpha_p > 0$. The Neumann-Neumann method for the Stokes-Darcy problem considering the normal velocity on Γ as interface variable reads as follows. Given λ^0 on Γ, for $m \geq 1$ until convergence,

1. Find $\mathbf{u}_f^{(m)}$ and $p_f^{(m)}$ such that

$$\begin{aligned} -\boldsymbol{\nabla} \cdot (2\mu_f \nabla^s \mathbf{u}_f^{(m)} - p_f^{(m)} \boldsymbol{I}) = \mathbf{f}_f\,, \quad \nabla \cdot \mathbf{u}_f^{(m)} = 0 \qquad & \text{in } \Omega_f\,, \\ -(\mathbf{n} \cdot (2\mu_f \nabla^s \mathbf{u}_f^{(m)} - p_f^{(m)} \boldsymbol{I}))_\tau = \xi_f (\mathbf{u}_f^{(m)})_\tau \quad & \text{on } \Gamma\,, \\ \mathbf{u}_f^{(m)} \cdot \mathbf{n} = \lambda^{(m-1)} \qquad & \text{on } \Gamma\,. \end{aligned} \tag{6}$$

2. Find $p_p^{(m)}$ such that

$$\begin{aligned} -\nabla \cdot (\boldsymbol{\eta}_p \nabla p_p^{(m)}) = f_p \quad & \text{in } \Omega_p\,, \\ -(\boldsymbol{\eta}_p \nabla p_p^{(m)}) \cdot \mathbf{n} = \lambda^{(m)} \quad & \text{on } \Gamma\,. \end{aligned} \tag{7}$$

3. Compute

$$\sigma^{(m)} = -\mathbf{n} \cdot (2\mu_f \nabla^s \mathbf{u}_f^{(m)} - p_f^{(m)} \boldsymbol{I}) \cdot \mathbf{n} - p_p^{(m)} \quad \text{on } \Gamma\,. \tag{8}$$

4. Find $\mathbf{v}_f^{(m)}$ and $q_f^{(m)}$ such that

$$\begin{aligned} -\boldsymbol{\nabla}\cdot(2\mu_f\nabla^s\mathbf{v}_f^{(m)} - q_f^{(m)}\boldsymbol{I}) = \mathbf{0}\,, \quad \nabla\cdot\mathbf{v}_f^{(m)} = 0 \qquad & \text{in } \Omega_f\,, \\ -(\mathbf{n}\cdot(2\mu_f\nabla^s\mathbf{v}_f^{(m)} - q_f^{(m)}\boldsymbol{I}))_\tau = \xi_f\,(\mathbf{v}_f^{(m)})_\tau & \text{on } \Gamma\,, \\ -\mathbf{n}\cdot(2\mu_f\nabla^s\mathbf{v}_f^{(m)} - q_f^{(m)}\boldsymbol{I})\cdot\mathbf{n} = \sigma^{(m)} & \text{on } \Gamma\,. \end{aligned} \tag{9}$$

5. Find $q_p^{(m)}$ such that

$$\begin{aligned} -\nabla\cdot(\boldsymbol{\eta}_p\nabla q_p^{(m)}) = 0 \qquad & \text{in } \Omega_p\,, \\ q_p^{(m)} = \sigma^{(m)} & \text{on } \Gamma\,. \end{aligned} \tag{10}$$

6. Set

$$\lambda^{(m+1)} = \lambda^{(m)} - (\alpha_f\,(\mathbf{v}_f^{(m)}\cdot\mathbf{n}) + \alpha_p(\boldsymbol{\eta}_p\nabla q_p^{(m)})\cdot\mathbf{n}) \quad \text{on } \Gamma\,. \tag{11}$$

Problems (6), (7), (9) and (10) are supplemented with homogeneous boundary conditions on $\partial\Omega_f \setminus \Gamma$ and $\partial\Omega_p \setminus \Gamma$ as indicated in Sect. 1.

2.1 Convergence analysis and optimization of the parameters

We analyse the Neumann-Neumann method (6)-(11) with the aim of characterizing optimal parameters α_f and α_p. To this purpose, we extend the methodology used to study optimized Schwarz methods for the Stokes-Darcy problem in [8, 12, 13]. Since all the problems are linear, we can study the convergence on the error equation to the zero solution when the forcing terms are $\mathbf{f}_f = \mathbf{0}$ and $f_p = 0$.

We consider the simplified setting where $\Omega_f = \{(x,y) \in \mathbb{R}^2 \colon x < 0\}$, $\Omega_p = \{(x,y) \in \mathbb{R}^2 \colon x > 0\}$, $\Gamma = \{(x,y) \in \mathbb{R}^2 \colon x = 0\}$, and $\mathbf{n} = (1,0)$ and $\boldsymbol{\tau} = (0,1)$. We assume $\boldsymbol{\eta}_p = \mathrm{diag}(\eta_1,\eta_2)$ with constant $\eta_1 \neq \eta_2$, and let $\mathbf{u}_f(x,y) = (u_1(x,y), u_2(x,y))^T$, $\mathbf{v}_f(x,y) = (v_1(x,y), v_2(x,y))^T$. In this setting, the Neumann-Neumann algorithm (6)–(11) becomes: given λ^0 on Γ, for $m \geq 1$ until convergence,

1. Solve the Stokes problem

$$\begin{aligned} -\mu_f \begin{pmatrix} (\partial_{xx}+\partial_{yy})u_1^{(m)} \\ (\partial_{xx}+\partial_{yy})u_2^{(m)} \end{pmatrix} + \begin{pmatrix} \partial_x p_f^{(m)} \\ \partial_y p_f^{(m)} \end{pmatrix} = 0,\ \partial_x u_1^{(m)} + \partial_y u_2^{(m)} = 0,\ \text{ in } (-\infty,0)\times\mathbb{R}\,, \\ -\mu_f\,(\partial_x u_2^{(m)} + \partial_y u_1^{(m)}) = \xi_f\,u_2^{(m)}, \quad u_1^{(m)} = \lambda^{(m)},\ \text{ on } \{0\}\times\mathbb{R}\,. \end{aligned} \tag{12}$$

2. Solve Darcy's problem

$$\begin{aligned} -(\eta_1\,\partial_{xx} + \eta_2\,\partial_{yy})\,p_p^{(m)} = 0 \qquad & \text{in } (0,+\infty)\times\mathbb{R}\,, \\ -\eta_1\partial_x p_p^{(m)} = \lambda^{(m)} & \text{on } \{0\}\times\mathbb{R}\,. \end{aligned} \tag{13}$$

3. Compute

$$\sigma^{(m)} = -2\mu_f\,\partial_x u_1^{(m)} + p_f^{(m)} - p_p^{(m)} \quad \text{on } \{0\}\times\mathbb{R}\,. \tag{14}$$

4. Solve the Stokes problem

$$
\begin{aligned}
-\mu_f \begin{pmatrix} (\partial_{xx}+\partial_{yy})v_1^{(m)} \\ (\partial_{xx}+\partial_{yy})v_2^{(m)} \end{pmatrix} + \begin{pmatrix} \partial_x q_f^{(m)} \\ \partial_y q_f^{(m)} \end{pmatrix} = 0,\ \partial_x v_1^{(m)} + \partial_y v_2^{(m)} = 0\,,\ \text{in } (-\infty,0)\times\mathbb{R}\,, \\
-\mu_f\,(\partial_x v_2^{(m)} + \partial_y v_1^{(m)}) = \xi_f\, v_2^{(m)}\,,\ \text{on } \{0\}\times\mathbb{R}\,, \\
-2\mu_f\,\partial_x v_1^{(m)} + q_f^{(m)} = \sigma^{(m)}\,,\ \text{on } \{0\}\times\mathbb{R}\,.
\end{aligned}
\tag{15}
$$

5. Solve Darcy's problem

$$
\begin{aligned}
-(\eta_1\,\partial_{xx} + \eta_2\,\partial_{yy})\, q_p^{(m)} = 0 \qquad \text{in } (0,+\infty)\times\mathbb{R}\,, \\
q_p^{(m)} = \sigma^{(m)} \ \text{on } \{0\}\times\mathbb{R}\,.
\end{aligned}
\tag{16}
$$

6. Set

$$
\lambda^{(m+1)} = \lambda^{(m)} - (\alpha_f\, v_1^{(m)} + \alpha_p\,\eta_1 \partial_x q_p^{(m)}) \quad \text{on } \{0\}\times\mathbb{R}\,. \tag{17}
$$

For the convergence analysis, we consider the Fourier transform in the direction tangential to the interface (corresponding to the y variable):

$$
\mathcal{F}\colon w(x,y) \mapsto \widehat{w}(x,k) = \int_{\mathbb{R}} e^{-iky} w(x,y)\,dy\,, \qquad \forall\, w(x,y) \in L^2(\mathbb{R}^2)\,,
$$

where k is the frequency variable. We quantify the error in the frequency space between two successive approximations $\widehat{\lambda}^{m+1}$ and $\widehat{\lambda}^{m}$ at Γ and characterize the reduction factor at iteration m for each frequency k. Finally, we identify optimal values of α_f and α_p by minimizing the reduction factor at each iteration over all the relevant Fourier modes.

Proposition 1 *Let $\eta_p = \sqrt{\eta_1\eta_2}$. The reduction factor of algorithm* (12)–(16) *does not depend on the iteration m, and it is given by $|\rho(\alpha_f,\alpha_p,k)|$ with*

$$
\rho(\alpha_f,\alpha_p,k) = 1 - \alpha_p(1+2\,\mu_f\,\eta_p\,k^2) - \alpha_f\,(1+(2\,\mu_f\,\eta_p\,k^2)^{-1})\,. \tag{18}
$$

Proof Following the same steps of the proof of Proposition 3.1 of [8], we find

$$
\widehat{u}_1^{(m)}(x,k) = \left(U_1^{(m)}(k) + \frac{P^{(m)}(k)}{2\mu_f}\,x\right) e^{|k|x}\,, \qquad \widehat{p}_p^{(m)}(x,k) = \Phi^{(m)}(k)\, e^{-\sqrt{\frac{\eta_2}{\eta_1}}|k|x}\,,
$$

and $\widehat{p}_f^{(m)}(x,k) = P^{(m)}(k)\, e^{|k|x}$. The interface conditions $(12)_4$ and $(13)_2$ give $U_1^{(m)}(k) = \widehat{\lambda}^{(m)}$ and $\Phi^{(m)}(k) = \frac{\widehat{\lambda}^{(m)}}{\eta_p|k|}$. Then, using the Fourier transform of (14), we can obtain

$$
\widehat{\sigma}^{(m)} = -(2\mu_f\,|k| + (\eta_p|k|)^{-1})\,\widehat{\lambda}^{(m)}\,.
$$

Proceeding in analogous way, the solutions of problems (15) and (16) become

$$
\widehat{v}_1^{(m)}(x,k) = \left(\overline{P}^{(m)}(k)\,x - \frac{\widehat{\sigma}^{(m)}}{|k|}\right)\frac{e^{|k|x}}{2\mu_f}\,, \qquad \widehat{q}_p^{(m)}(x,k) = \widehat{\sigma}^{m} e^{-\sqrt{\frac{\eta_2}{\eta_1}}|k|x}\,,
$$

and $\widehat{q}_f^{(m)}(x,k) = \overline{P}^m(k)e^{|k|x}$. Substituting into the Fourier transform of (17), we find $\widehat{\lambda}^{(m+1)} = \rho(\alpha_f, \alpha_p, k)\widehat{\lambda}^{(m)}$ with $\rho(\alpha_f, \alpha_p, k)$ defined in (18). □

Using a classical approach in optimized Schwarz methods, we now aim at optimizing the parameters α_f and α_p by minimizing the reduction factor for all the relevant frequencies k with $0 < \underline{k} \leq |k| \leq \overline{k}$, where $\underline{k}$ and $\overline{k}$ are the minimum and maximum relevant frequencies, respectively, with $\underline{k} = \pi/L$ (L being the length of the interface) and $\overline{k} = \pi/h$ (h being the size of the mesh). Since the function $\rho(\alpha_f, \alpha_p, k)$ is even with respect to k, we only consider $k > 0$ without loss of generality, and we proceed to solve the min-max problem

$$\min_{\alpha_f, \alpha_p > 0} \max_{k \in [\underline{k}, \overline{k}]} |\rho(\alpha_f, \alpha_p, k)| . \tag{19}$$

The following result holds.

Proposition 2 *The solution of the min-max problem* (19) *is given by*

$$\begin{aligned} \alpha_f^{NN} &= (2\,\mu_f\,\eta_p\,\underline{k}\,\overline{k})^2\,(\,1 + (2\,\mu_f\,\eta_p\,\underline{k}\,\overline{k})^2 + \mu_f\,\eta_p\,(\underline{k}+\overline{k})^2\,)^{-1}\,, \\ \alpha_p^{NN} &= (\,1 + (2\,\mu_f\,\eta_p\,\underline{k}\,\overline{k})^2 + \mu_f\,\eta_p\,(\underline{k}+\overline{k})^2\,)^{-1}\,. \end{aligned} \tag{20}$$

Moreover, $|\rho(\alpha_f^{NN}, \alpha_p^{NN}, k)| < 1$ *for all* $k \in [\underline{k}, \overline{k}]$*, and, asymptotically, when* $h \to 0$*,*

$$\begin{aligned} \alpha_f^{NN} &= 4\pi^2\mu_f\eta_p\,C_{NN}\,(1 - 2\,L\,C_{NN}\,h) + O(h^2) \\ \alpha_p^{NN} &= L^2(\pi^2\mu_f\eta_p)^{-1}C_{NN}\,h^2 + O(h^3) \\ \rho(\alpha_f^{NN}, \alpha_p^{NN}, \overline{k}) &= -L^2\,C_{NN} + (8\pi^2\mu_f\eta_p L + 4L^3)\,C_{NN}^2\,h + O(h^2)\,, \end{aligned}$$

with $C_{NN} = (4\pi^2\mu_f\,\eta_p + L^2)^{-1}$.

Proof For all $\alpha_f, \alpha_p > 0$, $\lim_{k\to 0}\rho(\alpha_f, \alpha_p, k) = \lim_{k\to\infty}\rho(\alpha_f, \alpha_p, k) = -\infty$, and the function $\rho(\alpha_f, \alpha_p, k)$ has a local maximum at $k^* = (\alpha_f/(\alpha_p\,(2\,\mu_f\,\eta_p)^2))^{1/4}$ where

$$\rho(\alpha_f, \alpha_p, k^*) = 1 - (\sqrt{\alpha_f} + \sqrt{\alpha_p})^2\,. \tag{21}$$

We distinguish two cases.

Case 1: $\sqrt{\alpha_f} + \sqrt{\alpha_p} \geq 1$. In this case, $\rho(\alpha_f, \alpha_p, k) \leq 0$ for all $\underline{k} \leq k \leq \overline{k}$, and $\rho(\alpha_f, \alpha_p, k) = 0$ if $\sqrt{\alpha_f} + \sqrt{\alpha_p} = 1$. Taking $\sqrt{\alpha_f} + \sqrt{\alpha_p} = 1$ would result in a null convergence rate for $k = k^*$, and we could then choose α_f and α_p by imposing $|\rho(\alpha_f, \alpha_p, \underline{k})| = |\rho(\alpha_f, \alpha_p, \overline{k})|$ (which would also ensure that $\underline{k} < k^* < \overline{k}$). This approach leads to $\alpha_p = (1 + 2\,\mu_f\,\eta_p\,\underline{k}\,\overline{k})^{-2}$ and $\alpha_f = (2\,\mu_f\,\eta_p\,\underline{k}\,\overline{k})^2(1 + 2\,\mu_f\,\eta_p\,\underline{k}\,\overline{k})^{-2}$, but, unfortunately, it does not guarantee that $|\rho(\alpha_f, \alpha_p, k)| < 1$ for all $k \in [\underline{k}, \overline{k}]$, which would be true when $1 + 2\,\mu_f\,\eta_p\,\underline{k}\,\overline{k} > \sqrt{2\,\mu_f\,\eta_p}\,(\overline{k} - \underline{k})$.

Case 2: $0 < \sqrt{\alpha_f} + \sqrt{\alpha_p} < 1$. In this case, $\rho(\alpha_f, \alpha_p, k^*) > 0$, and the function $\rho(\alpha_f, \alpha_p, k)$ has two positive zeros

$$k_{1,2} = (1 - \alpha_f - \alpha_p \pm ((1 - \alpha_f - \alpha_p)^2 - 4\,\alpha_f\,\alpha_p)^{1/2})^{1/2} / (4\,\mu_f\,\eta_p\,\alpha_p)^{1/2} \quad ,$$

whose position depends on the values of α_f and α_p. Therefore, we proceed by equioscillation and we look for α_f and α_p such that $-\rho(\alpha_f, \alpha_p, \underline{k}) = \rho(\alpha_f, \alpha_p, k^*)$ and $-\rho(\alpha_f, \alpha_p, \overline{k}) = \rho(\alpha_f, \alpha_p, k^*)$. This gives the values (20). Simple algebraic manipulations permit to verify that, for such values of the parameters, $k^* = (\underline{k}\,\overline{k})^{1/2}$ so that $\underline{k} < k_1 < k^* < k_2 < \overline{k}$. Moreover, $|\rho(\alpha_f, \alpha_p, k)| \le \rho(\alpha_f, \alpha_p, k^*)$ for all $\underline{k} \le k \le \overline{k}$ and, owing to (21), we can conclude that $|\rho(\alpha_f, \alpha_p, k)| < 1$ for all frequencies of interest. □

3 Numerical results

We consider a finite element approximation based on the inf-sup stable $\mathbb{Q}_2 - \mathbb{Q}_1$ Taylor-Hood elements [2] for Stokes, and $\mathbb{Q}_2$ elements Darcy. Denoting by the indices I_f, I_p and Γ the degrees of freedom in Ω_f, Ω_p and on Γ, respectively, the algebraic form of the discrete Stokes-Darcy problem (1)–(5) becomes

$$\begin{pmatrix} A^f_{I_f I_f} & A^f_{I_f \Gamma} & G^f_{I_f} & 0 & 0 \\ A^f_{\Gamma I_f} & A^f_{\Gamma\Gamma} & G^f_{\Gamma} & 0 & C_{fp} \\ (G^f_{I_f})^T & (G^f_{\Gamma})^T & 0 & 0 & 0 \\ 0 & 0 & 0 & A^p_{I_p I_p} & A^p_{I_p \Gamma} \\ 0 & -C^T_{fp} & 0 & A^p_{\Gamma I_p} & A^p_{\Gamma\Gamma} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{f,I_f} \\ \mathbf{u}_{f,\Gamma} \\ \mathbf{p}_f \\ \mathbf{p}_{p,I_p} \\ \mathbf{p}_{p,\Gamma} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_{f,I_f} \\ \mathbf{f}_{f,\Gamma} \\ \mathbf{0} \\ \mathbf{f}_{p,I_p} \\ \mathbf{f}_{p,\Gamma} \end{pmatrix}, \tag{22}$$

where $\mathbf{u}_{f,\Gamma}$ denotes the vector of degrees of freedom of the normal velocity on Γ. The Schur complement system with respect to $\mathbf{u}_{f,\Gamma}$ is

$$(\Sigma_f + \Sigma_p)\,\mathbf{u}_{f,\Gamma} = \mathbf{b}_\Gamma \tag{23}$$

where Σ_f and Σ_p are the symmetric and positive definite matrices (see [7]):

$$\Sigma_f = A^f_{\Gamma\Gamma} - \begin{pmatrix} A^f_{\Gamma I_f} & G^f_\Gamma \end{pmatrix} \begin{pmatrix} A^f_{I_f I_f} & G^f_{I_f} \\ (G^f_{I_f})^T & 0 \end{pmatrix}^{-1} \begin{pmatrix} A^f_{I_f \Gamma} \\ (G^f_\Gamma)^T \end{pmatrix},$$

$$\Sigma_p = \begin{pmatrix} 0 & C_{fp} \end{pmatrix} \begin{pmatrix} A^p_{I_p I_p} & A^p_{I_p \Gamma} \\ A^p_{\Gamma I_p} & A^p_{\Gamma\Gamma} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ C^T_{fp} \end{pmatrix}.$$

Following a classical approach in domain decomposition (see, e.g., [7, 16]), the Neumann-Neumann method (6)–(10) can be equivalently reformulated as a Richardson method for the Schur complement system (23) with preconditioner

$$P = \alpha_f\,\Sigma_f^{-1} + \alpha_p\,\Sigma_p^{-1}. \tag{24}$$

The PCG method with preconditioner P can then be used to solve (23).

We consider the computational domains $\Omega_f = (0, 0.5) \times (1, 1.5)$ and $\Omega_p = (0, 0.5) \times (0.5, 1)$ so that $\Gamma = (0, 0.5) \times \{1\}$, and we choose the forces $\mathbf{f}_f$ and f_p and the boundary conditions in such a way that the Stokes-Darcy problem has analytic solution $\mathbf{u}_f = (\sqrt{\eta_p}, \alpha_{BJ} x)^T$, $p_f = 2\,\mu_f\,(x + y - 1) + (3\,\eta_p)^{-1}$, and $p_p = \eta_p^{-1}(-\alpha_{BJ} x(y-1) + y^3/3 - y^2 + y) + 2\,\mu_f\, x$. The computational meshes are structured and characterized by $h = 0.1 \times 2^{1-j}$, $j = 1, \ldots, 4$, with 11, 21, 41, and 81 interface unknowns, respectively. We consider four configurations of physically significant dimensionless problem parameters (see also [12]): *(a)* $\mu_f = 10$, $\eta_p = 4 \times 10^{-10}$; *(b)* $\mu_f = 1$, $\eta_p = 4 \times 10^{-7}$; *(c)* $\mu_f = 10$, $\eta_p = 4 \times 10^{-9}$; *(d)* $\mu_f = 0.2$, $\eta_p = 2 \times 10^{-7}$.

Table 1 reports the computed values of the optimal parameters α_f^{NN} and α_p^{NN} (20) and the number of CG iterations with preconditioner (24) and without preconditioner (in brackets). For comparison, we indicate also the values of the optimal parameters α_f^{RR} and α_p^{RR} and the number of GMRES iterations obtained with the optimized Schwarz (Robin-Robin) method studied in [8]. (Notice that $\alpha_f^{NN} \sim c_f^0 + c_f^1 h$ and $\alpha_p^{NN} \sim c_p^0 + c_p^1 h$ when $h \to 0$ for suitable constants c_f^0, c_f^1, c_p^0 and c_p^1 that depend on μ_f, η_p and L.)

The number of PCG iterations using optimized parameters α_f^{NN}, α_p^{NN} is almost independent of both the mesh size and of the values of μ_f and η_p.

Moreover, the optimized Neumann-Neumann method performs better than the Robin-Robin method with lower computational cost per iteration. We also observe that, considering the Robin interface conditions $(3.3)_4$ and $(3.4)_4$ in [8] and the values of α_f^{RR} and α_p^{RR} (especially, the large values of α_f^{RR}), the Robin-Robin method actually behaves like a Dirichlet-Robin method with interface condition on

Table 1 Optimal parameters α_f^{NN} and α_p^{NN} and number of PCG iterations, and optimal parameters α_f^{RR} and α_p^{RR} for the Robin-Robin method with corresponding GMRES iterations ($tol = 10^{-9}$).

Case	Mesh	α_f^{NN}	α_p^{NN}	PCG iter	α_f^{RR}	α_p^{RR}	GMRES iter
(a)	h_1	9.97×10^{-12}	$1.00 \times 10^{+0}$	2 (12)	$7.23 \times 10^{+7}$	$6.91 \times 10^{+2}$	4
	h_2	3.99×10^{-11}	$1.00 \times 10^{+0}$	2 (17)	$3.79 \times 10^{+7}$	$1.32 \times 10^{+3}$	4
	h_3	1.60×10^{-10}	$1.00 \times 10^{+0}$	3 (22)	$1.94 \times 10^{+7}$	$2.58 \times 10^{+3}$	4
	h_4	6.38×10^{-10}	9.99×10^{-1}	3 (31)	$9.83 \times 10^{+6}$	$5.09 \times 10^{+3}$	4
(b)	h_1	9.96×10^{-8}	9.98×10^{-1}	3 (12)	$7.24 \times 10^{+4}$	$6.91 \times 10^{+1}$	6
	h_2	3.96×10^{-7}	9.93×10^{-1}	4 (17)	$3.80 \times 10^{+4}$	$1.32 \times 10^{+2}$	6
	h_3	1.55×10^{-6}	9.74×10^{-1}	4 (24)	$1.96 \times 10^{+4}$	$2.55 \times 10^{+2}$	8
	h_4	5.78×10^{-6}	9.06×10^{-1}	5 (30)	$1.03 \times 10^{+4}$	$4.86 \times 10^{+2}$	8
(c)	h_1	9.97×10^{-10}	$1.00 \times 10^{+0}$	3 (12)	$7.23 \times 10^{+6}$	$6.91 \times 10^{+2}$	4
	h_2	3.99×10^{-9}	9.99×10^{-1}	3 (17)	$3.79 \times 10^{+6}$	$1.32 \times 10^{+3}$	4
	h_3	1.59×10^{-8}	9.97×10^{-1}	3 (24)	$1.94 \times 10^{+6}$	$2.57 \times 10^{+3}$	6
	h_4	6.32×10^{-8}	9.90×10^{-1}	4 (30)	$9.87 \times 10^{+5}$	$5.06 \times 10^{+3}$	6
(d)	h_1	2.49×10^{-10}	$1.00 \times 10^{+0}$	2 (12)	$7.23 \times 10^{+5}$	$3.46 \times 10^{+1}$	4
	h_2	9.97×10^{-10}	$1.00 \times 10^{+0}$	3 (17)	$3.79 \times 10^{+5}$	$6.60 \times 10^{+1}$	4
	h_3	3.98×10^{-9}	9.99×10^{-1}	3 (22)	$1.94 \times 10^{+5}$	$1.29 \times 10^{+2}$	6
	h_4	1.59×10^{-8}	9.95×10^{-1}	4 (29)	$9.85 \times 10^{+4}$	$2.54 \times 10^{+2}$	6

the normal velocity $\mathbf{u}_f \cdot \mathbf{n}$ for the Stokes problem. This confirms that condition $(6)_3$ in the Neumann-Neumann algorithm is a valid choice for the Stokes problem.

Finally, the optimal values α_f^{NN}, α_p^{NN} suggest that the preconditioner (24) behaves like $P \approx \Sigma_p^{-1}$. Thus, while Σ_f^{-1} is an effective preconditioner for large values of μ_f and η_p (see [7, 10]), Σ_p^{-1} is a much better choice for small values, which is the case in most applications. This can lead to a Dirichlet-Neumann-type method different from the one in [7, 10] that will be discussed in a future work.

Acknowledgements The first author acknowledges funding by the EPSRC grant EP/V027603/1.

References

1. Beavers, G. S. and Joseph, D. D. Boundary conditions at a naturally permeable wall. *J. Fluid Mech.* **30**, 197–207 (1967).
2. Boffi, D., Brezzi, F., and Fortin, M. *Mixed Finite Element Methods and Applications*. Springer, Berlin and Heidelberg (2013).
3. Caiazzo, A., John, V., and Wilbrandt, U. On classical iterative subdomain methods for the Stokes-Darcy problem. *Comput. Geosci.* **18**, 711–728 (2014).
4. Cao, Y., Gunzburger, M., Hu, X., Hua, F., Wang, X., and Zhao, W. Finite element approximations for Stokes-Darcy flow with Beavers-Joseph interface conditions. *SIAM J. Numer. Anal.* **47**(6), 4239–4256 (2010).
5. Cao, Y., Gunzburger, M., Hua, F., and Wang, X. Coupled Stokes-Darcy model with Beavers-Joseph interface boundary conditions. *Comm. Math. Sci.* **8**(1), 1–25 (2010).
6. Chen, W., Gunzburger, M., Hua, F., and Wang, X. A parallel Robin-Robin domain decomposition method for the Stokes-Darcy system. *SIAM J. Numer. Anal.* **49**(3), 1064–1084 (2011).
7. Discacciati, M. *Domain Decomposition Methods for the Coupling of Surface and Groundwater Flows*. Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Switzerland (2004).
8. Discacciati, M. and Gerardo-Giorda, L. Optimized Schwarz methods for the Stokes-Darcy coupling. *IMA J. Numer. Anal.* **38**(4), 1959–1983 (2018).
9. Discacciati, M., Miglio, E., and Quarteroni, A. Mathematical and numerical models for coupling surface and groundwater flows. *Appl. Numer. Math.* **43**, 57–74 (2002).
10. Discacciati, M. and Quarteroni, A. Convergence analysis of a subdomain iterative method for the finite element approximation of the coupling of Stokes and Darcy equations. *Comput. Visual. Sci.* **6**, 93–103 (2004).
11. Discacciati, M., Quarteroni, A., and Valli, A. Robin-Robin domain decomposition methods for the Stokes-Darcy coupling. *SIAM J. Numer. Anal.* **45**(3), 1246–1268 (2007).
12. Discacciati, M. and Vanzan, T. Optimized Schwarz methods for the time-dependent Stokes-Darcy coupling. Tech. rep. (2022). Submitted.
13. Gander, M. J. and Vanzan, T. On the derivation of optimized transmission conditions for the Stokes-Darcy coupling. In: et al., R. H. (ed.), *Domain Decomposition Methods in Science and Engineering XXV. DD 2018*. Springer (2020).
14. Jäger, W. and Mikelić, A. On the boundary conditions at the contact interface between a porous medium and a free fluid. *Ann. Scuola Norm. Sup. Pisa Cl. Sci.* **23**, 403–465 (1996).
15. Layton, W. L., Schieweck, F., and Yotov, I. Coupling fluid flow with porous media flow. *SIAM J. Num. Anal.* **40**, 2195–2218 (2003).
16. Quarteroni, A. and Valli, A. *Domain Decomposition Methods for Partial Differential Equations*. The Clarendon Press, Oxford University Press, New York (1999).
17. Saffman, P. G. On the boundary condition at the interface of a porous medium. *Stud. Appl. Math.* **1**, 93–101 (1971).
18. Toselli, A. and Widlund, O. *Domain Decomposition Methods – Algorithms and Theory*, *Springer Series in Computational Mathematics*, vol. 34. Springer, Berlin (2005).

Finite Basis Physics-Informed Neural Networks as a Schwarz Domain Decomposition Method

Victorita Dolean, Alexander Heinlein, Siddhartha Mishra, and Ben Moseley

1 Introduction

The success and advancement of machine learning (ML) in fields such as image recognition and natural language processing has lead to the development of novel methods for the solution of problems in physics and engineering. However, algorithms developed in traditional fields of ML usually require a large amount of data, which are difficult to obtain from measurements and/or traditional numerical simulations. Furthermore, such algorithms can be difficult to interpret and can struggle to generalize. To overcome these issues, a new research paradigm has emerged, known as scientific machine learning (SciML) [1, 7], which aims to more tightly combine ML with scientific principles to provide more powerful algorithms.

One such approach are physics-informed neural networks (PINNs) [4, 10], which are designed to approximate the solution to the boundary value problem

$$\begin{aligned} \mathcal{N}[u](\mathbf{x}) &= f(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d, \\ \mathcal{B}_k[u](\mathbf{x}) &= g_k(\mathbf{x}), \quad \mathbf{x} \in \Gamma_k \subset \partial\Omega \end{aligned} \tag{1}$$

where $\mathcal{N}[u](\mathbf{x})$ is a differential operator, u is the solution and $\mathcal{B}_k(\cdot)$ is a set of boundary conditions, such that the solution u is uniquely determined. Note that boundary conditions are to be understood in a broad sense and the $\mathbf{x}$ variable can also include time. In particular, we do not distinguish between initial and boundary conditions.

Victorita Dolean
University of Strathclyde, 26, Richmond Street, G1 1XH Glasgow, UK,
e-mail: work@victoritadolean.com

Alexander Heinlein
Delft University of Technology, Faculty of Electrical Engineering Mathematics & Computer Science, Delft Institute of Applied Mathematics, Mekelweg 4, 2628 CD Delft, Netherlands,
e-mail: a.heinlein@tudelft.nl

Siddhartha Mishra, Ben Moseley
ETH Zürich, Computational and Applied Mathematics Laboratory / ETH AI Center, Rämistrasse 101, 8092 Zürich, Switzerland

The approximation to the solution of (1) is given by a neural network $u(\mathbf{x}, \boldsymbol{\theta})$ (for the sake of simplicity we use the same notation for the solution of the PDE and the neural network) where $\boldsymbol{\theta}$ is a vector of all the parameters of the neural network (i.e., its weights and biases). The network is trained via the loss function

$$\mathcal{L}(\boldsymbol{\theta}) = \underbrace{\frac{\lambda_I}{N_I}\sum_{i=1}^{N_I}(\mathcal{N}[u](\mathbf{x}_i, \boldsymbol{\theta}) - f(\mathbf{x}_i))^2}_{\mathcal{L}_{\text{PDE}}} + \underbrace{\sum_{k=1}^{N_k}\frac{\lambda_B^k}{N_B^k}\sum_{j=1}^{N_B^k}(\mathcal{B}_k[u](\mathbf{x}_j^k, \boldsymbol{\theta}) - g_k(\mathbf{x}_j^k))^2}_{\mathcal{L}_{\text{BC}}}. \tag{2}$$

Here, $\{\mathbf{x}_i\}_{i=1}^{N_I}$ is a set of collocation points sampled in the interior of the domain, $\{\mathbf{x}_j^k\}_{j=1}^{N_B^k}$ is a set of points sampled along each boundary condition, and λ_I and λ_B^k are well-chosen scalar hyperparameters which ensure that the terms in the loss function are well balanced. Intuitively, one can see that the PDE loss tries to ensure that the solution learned by the network obeys the underlying PDE whilst the boundary loss tries to ensure it obeys the boundary conditions.

In practice, the presence of the boundary loss in eq. (2) often slows down training as it can compete with the PDE term [12]. In a slightly different formulation, boundary conditions can instead be enforced exactly as hard constraints by using the neural network as part of a solution ansatz Cu where C is a constraining operator which enforces that the solution explicitly satisfies the boundary conditions [4, 8]. This turns the optimization problem into an unconstrained one, and only the PDE loss from eq. (2) is required to train the PINN. For example, suppose we want to enforce that $u(0) = 0$ when solving a one-dimensional ODE, then the ansatz and constraining operator can be chosen as $[Cu](x, \boldsymbol{\theta}) = \tanh(x)u(x, \boldsymbol{\theta})$. The rationale behind this is that the function $\tanh(x)$ is null at 0, forcing the boundary condition to be obeyed, but non-zero away from 0, allowing the network to learn the solution away from the boundary condition.

Whilst PINNs have proven to be successful for solving many different types of differential equations, they often struggle to scale to problems with larger domains and more complex, multi-scale solutions [8, 11]. This is in part due to the spectral bias of neural networks [9] (their tendency to learn higher frequencies much slower than lower frequencies), and the increasing size of the underlying PINN optimization function. One way to alleviate these scaling issues is to combine PINNs with a domain decomposition method (DDM); by taking a divide-and-conquer approach, one hopes that the large, global optimization problem can be turned into a series of smaller and easier localized problems. In particular, [8] proposed finite basis physics-informed neural networks (FBPINNs) where the global PINN is partitioned into many local networks that are trained to approximate the solution on an overlapping domain decomposition. Related approaches are the deep domain decomposition method (D3M) [5] and the deep-learning-based domain decomposition method (DeepDDM) [6], which combine overlapping Schwarz domain decomposition methods with a PINN-based discretization. Other earlier works on the use

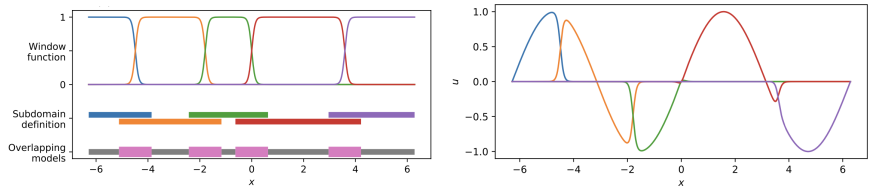


Fig. 1 Local FBPINN subdomains and window functions w_j (left), local solutions u_j (right)

of machine learning and domain decomposition methods include the prediction of the geometrical location of constraints in adaptive FETI-DP and BDDC methods; see [2]. For an overview of the combination of domain decomposition methods and machine learning, see [3].

In this work, we build upon FBPINNs by showing how Schwarz-like additive, multiplicative and hybrid iterative training strategies for FBPINNs can be developed. We present numerical experiments on the influence of these training strategies on convergence and accuracy. We propose and evaluate a preliminary implementation of a coarse space correction for FBPINNs, to further improve their efficiency.

2 Finite basis physics-informed neural networks (FBPINNs)

First we briefly present the FBPINN method introduced by [8] from a DDM perspective. The FBPINN method can be seen as a network architecture that allows for a localization of the network training. Therefore, let us consider a set of collocation points $X = \{\mathbf{x}_i\}_{i=1}^N$ in the global domain Ω and a decomposition into overlapping domains $\Omega = \cup_{j=1}^J \Omega_j$ inducing a decomposition into subsets of collocation points $X_j = \{\mathbf{x}_i^j\}_{i=1}^{N_j}$, $j = 1, \ldots, J$. As usual in overlapping Schwarz methods, $X = \cup_{j=1}^J X_j$ is not disjoint. For each subdomain Ω_j, we denote $\mathcal{N}_j$ the index set of neighboring subdomains, $\Omega_j^\circ = \cup_{l=1}^{\mathcal{N}_j} \Omega_l \cap \Omega_j$ the overlapping subset of Ω_j, and $\Omega_j^{\text{int}} = \Omega_j \setminus \Omega_j^\circ$, the interior part of the domain; let X_j° and X_j^{int} be the corresponding sets of collocation points, and $X^\circ = \cup_{j=1}^J X_j^\circ$ and $X^{\text{int}} = \cup_{j=1}^J X_j^{\text{int}}$. We now define the global network u as the sum of local networks $u_j(\mathbf{x}, \boldsymbol{\theta}_j)$ weighted by window functions ω_j: $u = \sum_{j, \mathbf{x}_i \in \Omega_j} \omega_j u_j$. Here, the local networks have individual network parameters $\boldsymbol{\theta}_j$, and of course, they could simply be evaluated everywhere in $\mathbb{R}^d$. In order to restrict them to their corresponding overlapping subdomains, we multiply them with the window functions, which have the properties $\text{supp}(\omega_i) \subset \Omega_i$ and $\Omega \subset \cup_{j=1}^J \text{supp}(\omega_i)$; the specific definition of ω_i employed here can be found in [8, eq. (14)]. See Fig. 1 for a graphical representation of the overlapping subdomains, their overlapping and interior sets, window functions, and local solutions for a simple one-dimensional example. If we insert the expression for u into eq. (2), we see that the loss function can be written as:

$$\mathcal{L}(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_J) = \frac{1}{N} \sum_{i=1}^{N} \left(\mathcal{N}[C \sum_{j, \mathbf{x}_i \in X_j} \omega_j u_j](\mathbf{x}_i, \boldsymbol{\theta}_j) - f(\mathbf{x}_i) \right)^2 . \tag{3}$$

The contribution to the global loss function can be split into a part coming from interior points and another one from points in the overlap, respectively:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_J) = & \underbrace{\frac{1}{N} \sum_{\mathbf{x} \in X^{\text{int}}} \left(\mathcal{N}[C \sum_{l, \mathbf{x} \in X_l} \omega_l u_l](\mathbf{x}, \boldsymbol{\theta}_l) - f(\mathbf{x}) \right)^2}_{=: \mathcal{L}^{\text{int}}(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_J)} \\ & + \frac{1}{N} \sum_{\mathbf{x} \in X^{\circ}} \left(\mathcal{N}[C \sum_{l, \mathbf{x} \in X_l} \omega_l u_l](\mathbf{x}, \boldsymbol{\theta}_l) - f(\mathbf{x}) \right)^2 . \end{aligned} \tag{4}$$

Note also that, since $X_i^{\text{int}} \cap X_j^{\text{int}} = \emptyset$ for $i \neq j$, the interior contribution can be simplified as follows:

$$\mathcal{L}^{\text{int}}(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_J) = \frac{1}{N} \sum_{j=1}^{J} \sum_{\mathbf{x}_i \in X_j^{\text{int}}} \left(\mathcal{N}[C \omega_j u_j](\mathbf{x}_i, \boldsymbol{\theta}_j) - f(\mathbf{x}_i) \right)^2 .$$

In [8], the authors introduce the notion of *scheduling* which is related to the degree of parallelism one can consider in Schwarz domain decomposition methods. For example in the well-know alternating Schwarz method, local solves take place sequentially, in an alternating manner, with data being exchanged at the interfaces. In the case of the parallel Schwarz method, local solutions are computed simultaneously, but subdomains only have access to interface data at the previous iteration. As is well-known in DDMs, the alternating method convergences in fewer iterations than the parallel method, whereas the second methods allows the concurrent computation of the local solutions; hence, the parallel Schwarz method is often more efficient in a parallel implementation.

In the case of many subdomains, one can define a so-called coloring strategy, i.e., subdomains with the same color are computed in parallel and different colors are processed sequentially. Here, we will consider any possible coloring scheme, allowing for arbitrary combinations of additive and multiplicative coupling. In particular, let us split the set of subdomain indices as follows $\{1, \dots, J\} = \mathcal{A} \cup \mathcal{I}$, such that subdomains Ω_j, $j \in \mathcal{A}$ are allocated the same 'color' which is different than those of the subdomains Ω_j, $j \in \mathcal{I}$. In the case of training FBPINNs, the notion of coloring is replaced by that of *scheduling*, that is, subdomains indexed in $\mathcal{A}$ are considered to be *active* at a given iteration and those indexed in $\mathcal{I}$ are *inactive*. The case when $\mathcal{I} = \emptyset$ corresponds to the fully parallel Schwarz method, whereas the case where only one subdomain is *active* at a time corresponds to a fully alternating Schwarz iteration. Denoting a subdomain Ω_j as *inactive* corresponds to fixing $\boldsymbol{\theta}_j$ during the optimization of $\mathcal{L}(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_J)$.

Algorithm 1 FBPINN training step for each subdomain

if $j \in \mathcal{A}$ (Ω_j is an active domain) **then**

Perform p iterations of gradient descent on θ_j^k (θ_i^k where $i \neq j$ are kept fixed):

$$\theta_j^{k+l} = \theta_j^{k+l-1} - \lambda \nabla_{\theta_j} \mathcal{L}(\theta_1^k, \ldots, \theta_{j-1}^k, \theta_j^{k+l-1}, \theta_{j+1}^k, \ldots, \theta_J^k), l = 1, \ldots, p.$$

Update the solution in the overlapping regions (communicate with neighbours):

$$\forall \mathbf{x} \in \Omega_j^\circ,\ u(\mathbf{x}, \theta_j^{k+p}) \leftarrow \sum_{l, \mathbf{x} \in \Omega_l} \omega_l u_l(\mathbf{x}, \theta_l^{k+p}).$$

end if

The FBPINN training algorithm follows the 'coloring' strategy described above. Let us denote by θ_j^k the parameter values at the k-th training step, and to simplify the presentation, we focus on the case of a first order gradient-based optimizer. If we start from an initial guess θ_j^0, then the training step for each subdomain is given by Algorithm 1. Once all active subdomains have completed one training step, the set $\mathcal{A}$ and $\mathcal{I}$ are updated. This whole procedure is repeated until any stopping criterion, such as a maximum number of iterations or a tolerance for the loss, is met.

Let us note that:

- The gradient updates can be performed in parallel and are fully localized even if the loss function is global; only in the update step are network solutions and network gradients transferred between neighboring subdomains.
- It is not necessary to perform communication in the overlaps (here in orange) at every iteration of gradient descent, but rather every p iterations for a better computational efficiency. The overall convergence can also be affected; cf. Fig.2.
- Unlike in classical domain decomposition methods, in our approach, the global problem is not decomposed into local problems, which can be solved independently. Instead, we always compute gradient updates with respect to the global loss function, and the domain decomposition and hence the localization enters through the window functions in the definition of the architecture of global network.

To illustrate the behavior of this algorithm we will consider a scaling study for its flexible training strategy. In particular, we fix the number of global collocation points and investigate the influence of changing the number of subdomains and the value of p on convergence when solving the simple 1D ODE $\frac{du}{dx} = \cos \omega x$, $u(0) = 0$, with $\omega = 15$. All subdomains are kept active all of the time, and all other FBPINN design choices are kept the same, including window function and local network architecture per subdomain. We only consider the case of relatively large overlap of 70% of the subdomain width, but the results are qualitatively the same for other sizes of overlap. As discussed in [8] and as in classical overlapping Schwarz methods, performance generally improves when increasing the size of the overlap; a systematic investigation is still open.

In Figure 2, we display the convergence of the loss function when the communication between subdomains takes place every $p \in \{1, 10, 100, 1000\}$ epochs. We ob-

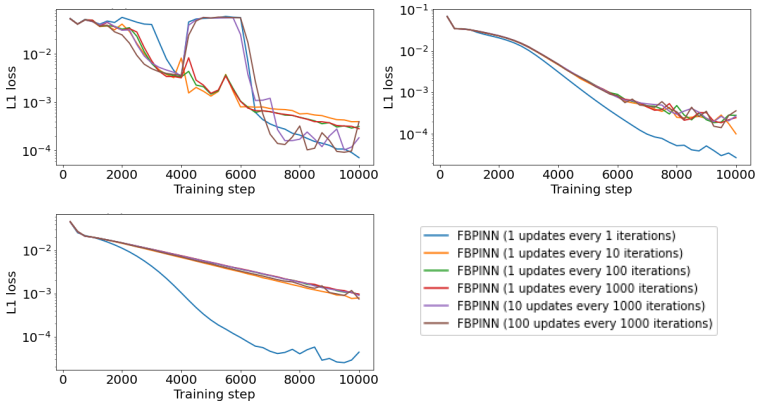


Fig. 2 FBPINN convergence for decompositions into 8 (upper left), 16 (upper right) and 32 (lower left) subdomains and different p values. Each network has 2 layers and 16 hidden units per layer. A total of 3,000 collocation points regularly sampled over the domain are used. For each decomposition, subdomains are regularly spaced, with an overlap of 70% of the subdomain width.

serve that the case of 8 subdomains is rather special since convergence appears rather unstable and there is no option that performs clearly best. As we increase the number of subdomains to 16 and 32 we observe an expected behavior, that is, the convergence rate improves if we communicate solutions and gradients in the overlaps every iteration. Moreover, when increasing the number of subdomains, naturally the global training performs less well, which is well known in domain decomposition as lack of scalability; we observe this behavior for all values of p. This is expected because the method above corresponds to a one-level method (meaning only neighboring subdomains communicate and there is no global exchange of information). Surprisingly, we do not see any clear difference in the convergence depending on p, that is, depending on how often we communicate. Since, in a parallel setting, it is computationally more efficient to communicate less, the results seems to indicate that, if we do not communicate in each step, it is beneficial to communicate as little as possible.

3 Coarse correction

Coarse spaces are instrumental in DDMs, as they ensure the robustness of a given method with respect to the number of subdomains as well as other problem-specific parameters, such as physical properties like frequency for wave problems or conductivity for diffusion type problems. Coarse spaces are often defined based on geometrical information (like a coarser mesh) but more sophisticated coarse spaces can be constructed using spectral information of underlying local problems. When training PINNs, it is not immediately clear how to define a coarse space, that is, a coarse network model, nor how to choose the number of collocation points and parameters of the coarse model. In what follows, we propose and evaluate a preliminary implementation of a coarse space correction for FBPINNs.

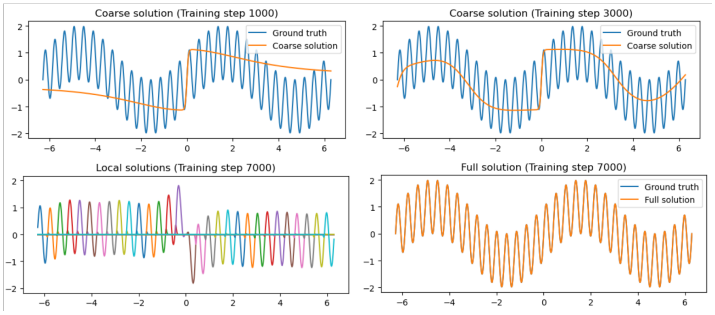


Fig. 3 Coarse correction for FBPINNs. Each subdomain network and the coarse network has 2 layers and 16 hidden units per layer. A total of 500 collocation points regularly sampled over the domain are used to train the coarse network, and 3,000 for the local networks. For the local networks, the subdomains are regularly spaced, with an overlap of 70% of the subdomain width.

In particular, we exploit the spectral bias of neural networks in order to build a coarse correction. This is the well-studied phenomenon that they tend to learn higher frequencies much slower than lower frequencies [9], and similar effects are observed for PINNs [8, 11]. Indeed, this effect is what motivated the use of domain decomposition in FBPINNs. More precisely, we first train a small but global network for enough epochs to learn the low frequency component of the solution; in particular, we employ a coarse network with the same architecture as a single local network. Then, local subdomains are added to approximate missing higher frequency components. The resulting FBPINN solution is given by $u = u_g + \sum_{j,\mathbf{x}_i \in \Omega_j} \omega_j u_j$, where $u_g(\mathbf{x}, \boldsymbol{\theta}_g)$ is the coarse network and $u_j(\mathbf{x}, \boldsymbol{\theta}_j)$ are the local networks. Because of spectral bias, low frequencies are first learned by the coarse network, and a relatively small network is sufficient to approximate the low frequencies. Then the local networks only need to learn the remaining higher frequencies. Since the local models only have to learn a local part of the solution, relatively small local network models are also sufficient.

We will apply these ideas on the simple 1D ODE, $\frac{du}{dx} = \omega_1 \cos(\omega_1 x) + \omega_2 \cos(\omega_2 x)$, $u(0) = 0$, where two frequencies are present in the solution, $u(x) = \sin(\omega_1 x) + \sin(\omega_2 x)$. For our test case, we choose $\omega_1 = 1$ as a lower frequency and $\omega_2 = 15$ as the higher frequency and we decompose the global domain into 30 overlapping subdomains; see Fig. 3. We note, as shown in [8] for an ODE with a single high frequency, solving such a problem with a single PINN requires a high network complexity and large number of iterations. First, we train the global coarse network, u_g, until the lower frequency is learned. We illustrate this progressive process in Fig. 3 where we see that we need roughly 3,000 epochs to identify the lower frequency. Here, we have chosen the number of epochs by hand based on the accuracy of the coarse solution, but in the future, we will work on automating the training of the coarse network. Then, the coarse network is fixed and the local networks are trained to approximate the remaining component of the solution, with all local networks kept active at each training step. As can be seen in Fig. 3, using our proposed approach, the coarse network approximates the coarse component of the solution, and the local subdomain networks approximate the high frequency components on the local subdomains.

4 Conclusions

In this work, we provide first insights on how to incorporate techniques from classical Schwarz domain decomposition methods into the FBPINN method. We show that its algorithmic components can be translated in the language of domain decomposition methods, and the well-established notions of additive, multiplicative and, hybrid Schwarz iterations can be identified through the notion of the flexible scheduling strategies introduced in [8]. Finally, we start exploring the notion of coarse space for FBPINNs. In particular, we train a coarse network to approximate the low frequency components of the solution and then continue by training local networks to approximate the remaining high frequency components. These ideas can be extended in a straightforward way to other, more complex boundary value problems.

References

1. Baker, N., Alexander, F., Bremer, T., Hagberg, A., Kevrekidis, Y., Najm, H., Parashar, M., Patra, A., Sethian, J., Wild, S., Willcox, K., and Lee, S. Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence. Tech. rep., USDOE Office of Science (SC) (United States) (2019).
2. Heinlein, A., Klawonn, A., Lanser, M., and Weber, J. Machine learning in adaptive domain decomposition methods - predicting the geometric location of constraints. *SIAM Journal on Scientific Computing* **41**(6), A3887–A3912 (2019).
3. Heinlein, A., Klawonn, A., Lanser, M., and Weber, J. Combining machine learning and domain decomposition methods for the solution of partial differential equations – a review. *GAMM-Mitteilungen* **44**(1), e202100001 (2021).
4. Lagaris, I. E., Likas, A., and Fotiadis, D. I. Artificial neural networks for solving ordinary and partial differential equations. *IEEE Transactions on Neural Networks* **9**(5), 987–1000 (1998). 9705023.
5. Li, K., Tang, K., Wu, T., and Liao, Q. D3M: A deep domain decomposition method for partial differential equations. *IEEE Access* **8**, 5283–5294 (2019).
6. Li, W., Xiang, X., and Xu, Y. Deep domain decomposition method: Elliptic problems. In: *Mathematical and Scientific Machine Learning*, 269–286. PMLR (2020).
7. Moseley, B. *Physics-informed machine learning: from concepts to real-world applications*. Ph.D. thesis, University of Oxford (2022).
8. Moseley, B., Markham, A., and Nissen-Meyer, T. Finite Basis Physics-Informed Neural Networks (FBPINNs): a scalable domain decomposition approach for solving differential equations. *arXiv preprint arXiv:2107.07871* (2021).
9. Rahaman, N., Baratin, A., Arpit, D., Draxler, F., Lin, M., Hamprecht, F., Bengio, Y., and Courville, A. On the spectral bias of neural networks. In: Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, *Proceedings of Machine Learning Research*, vol. 97, 5301–5310. PMLR (2019).
10. Raissi, M., Perdikaris, P., and Karniadakis, G. E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics* **378**, 686–707 (2019).
11. Wang, S., Wang, H., and Perdikaris, P. On the eigenvector bias of Fourier feature networks: From regression to solving multi-scale PDEs with physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering* **384**, 113938 (2021). 2012.10047.
12. Wang, S., Yu, X., and Perdikaris, P. When and why PINNs fail to train: A neural tangent kernel perspective. *Journal of Computational Physics* **449**, 110768 (2022).

Multigrid Interpretation of a Three-Level Parareal Algorithm

Stephanie Friedhoff, Martin J. Gander, and Felix Kwok

1 Introduction

Parallel-in-time methods, of which parareal [13] and multigrid reduction in time (MGRIT) [3] are well-known examples, are important tools for increasing parallelism beyond traditional spatially parallel methods, see [6, 14] and references therein. As a two-level method, parareal performs the fine but expensive integration independently (and in parallel) over many short time intervals, and it uses a cheap (but coarse) integrator to correct values across time subintervals sequentially. For linear ODE systems, parareal iterates are known to be equivalent to two-level MGRIT ones for a specific choice of initial guess, restriction/prolongation operators and relaxation scheme, cf. [3, 9, 10]. One can thus analyze parareal convergence in two ways: one can make hypotheses on Lipschitz constants and truncation errors, which is typical in the ODE community, cf. [1, 8, 13], or one can use spectral information of all-at-once matrices, as is common in the multigrid community, see [2, 3, 5, 15].

When parareal and MGRIT are used with many time subintervals, the coarse correction step becomes a computational bottleneck. To overcome this, one can parallelize the coarse solution by subdividing the coarse problem and using a coarser level to ensure global communication. For MGRIT, this leads to a multilevel variant [11]; for parareal, a three-level variant has been introduced and analyzed in [12]. In this paper, we show that there is a choice of restriction/prolongation operators and relaxation schemes such that the resulting MGRIT method is equivalent to three-level parareal when applied to linear problems. The existing MGRIT literature can thus add to our understanding of three-level parareal, beyond what is shown in [12].

Stephanie Friedhoff
Bergische Universität Wuppertal, e-mail: friedhoff@math.uni-wuppertal.de

Martin J. Gander
Université de Genève, e-mail: martin.gander@unige.ch

Felix Kwok
Université Laval, e-mail: felix.kwok@mat.ulaval.ca

2 The three-level parareal algorithm

Suppose one wishes to solve the *linear* system of ODEs $u' = \Phi u + f(t)$ with initial conditions $u(0) = u_0$ on the interval $[0, T]$. To obtain the temporal grid for both parareal and MGRIT, we subdivide the interval hierarchically as follows:[1]

- The interval $[0, T]$ is subdivided into p coarsest intervals $\mathcal{I}_i = [T_{i-1}, T_i]$, $i = 1, \ldots, p$, each of length $\Delta T = T/p$;
- Each coarsest interval $\mathcal{I}_i$ is subdivided into m subintervals $\mathcal{I}_{i,j} = [t_{i,j}, t_{i,j+1}]$, $j = 0, 1, \ldots, m-1$, of length $\Delta t = \Delta T/m$;
- Each $\mathcal{I}_{i,j}$ is divided into intervals $[t_{i,j,k}, t_{i,j,k+1}]$ $(0 \le k < n)$ of length $\delta t = \Delta t/n$.

We can now define the following *propagators*, which take an initial value at the beginning of $\mathcal{I}_i$, $\mathcal{I}_{i,j}$ or $\mathcal{I}_{i,j,k}$ and return the solution at the end of the interval:[2]

- F_0 is the action of the fine integrator over one fine time step δt. For a linear problem, we have $F_0 u_{i-1} = \Phi_0 u_{i-1} + f_i$.
- $F = F_0^n$ is the action of the fine integrator over one intermediate time step $\Delta t = n\delta t$. For a linear problem, we have $F u_{i-n} = \Phi_0^n u_{i-n} + \sum_{k=0}^{n-1} \Phi_0^k f_{i-k}$.
- G is the action of the intermediate integrator over one intermediate time step Δt. For a linear problem, we have $G U_{i,j-1} = \Phi_1 U_{i,j-1} + \gamma_{i,j}$.
- H is the action of the coarse integrator over one coarse time step $\Delta T = m\Delta t$. For a linear problem, we have $H Y_{i-1} = \Phi_2 Y_{i-1} + \eta_i$.

The three-level parareal algorithm, as introduced in [12], iterates on the level-1 state variables $U_{i,j}$ and level-2 state variables Y_i as follows:

1. Initialization (with iteration indices appearing as superscripts):

$$\begin{aligned} Y_0^0 &= u_0, & Y_i^0 &= H Y_{i-1}^0 \\ U_{i,0}^0 &= Y_{i-1}^0, & U_{i,j}^0 &= G U_{i,j-1}^0 \qquad (1 \le j \le m), \end{aligned}$$

2. Iteration: for $\nu = 0, 1, 2, \ldots$,

$$U_{i,0}^{\nu+1} = Y_{i-1}^{\nu}, \qquad U_{i,j}^{\nu+1} = F U_{i,j-1}^{\nu} + G U_{i,j-1}^{\nu+1} - G U_{i,j-1}^{\nu} \qquad (1 \le j \le m), \quad (1)$$

$$Y_0^{\nu+1} = u_0, \qquad Y_i^{\nu+1} = U_{i,m}^{\nu+1} + H Y_{i-1}^{\nu+1} - H Y_{i-1}^{\nu}. \quad (2)$$

This method is shown in [12] to converge to the fine solution *in finitely many steps*, i.e., $U_{i,j}^{\nu} = F^{(i-1)m+j} u_0$ for $\nu \ge i(m+1)$, *for any choice of G and H*. Note that this is not a nested iteration, where one needs to iterate U or Y to sufficient accuracy before switching levels; instead, only one parareal step on $U_{i,j}$ is performed before it is used in (2), and one coarse parareal step (2) is performed before the Y_i are used as new initial values in (1).

[1] For ease of explanation, we assume that all subdivisions have equal length, although it is easy to see that similar results hold for non-uniform subdivisions.

[2] To lighten the notation, the time index is only indicated in the variable on which the propagators are applied, and not in the propagators themselves.

Algorithm 1 MGRIT(ℓ, $\tilde{\mathbf{g}}^{(\ell)}$) (in correction form, as defined in [3])

if ℓ is the coarsest level L **then**
  Solve coarse grid system $A_L\mathbf{u}^{(L)} = \tilde{\mathbf{g}}^{(L)}$
else
  Relax on $A_\ell\mathbf{u}^{(\ell)} = \tilde{\mathbf{g}}^{(\ell)}$ using F-relaxation
  Compute and restrict residual using injection: $\tilde{\mathbf{g}}^{(\ell+1)} = R_\ell^{\ell+1}(\tilde{\mathbf{g}}^{(\ell)} - A_\ell\mathbf{u}^{(\ell)})$
  Solve on the next level : MGRIT($\ell+1$, $\tilde{\mathbf{g}}^{(\ell+1)}$)
  Correct: $\mathbf{u}^{(\ell)} \leftarrow \mathbf{u}^{(\ell)} + P_{\ell+1}^{\ell}\mathbf{u}^{(\ell+1)}$
end if

Algorithm 2 MGRIT-FAS(ℓ, $\mathbf{u}^{(\ell)}$, $\mathbf{g}^{(\ell)}$) (as defined in [4])

if ℓ is the coarsest level L **then**
  Solve coarse grid system $A_L(\mathbf{u}^{(L)}) = \mathbf{g}^{(L)}$
else
  Relax on $A_\ell(\mathbf{u}^{(\ell)}) = \mathbf{g}^{(\ell)}$ using F-relaxation to obtain $\mathbf{v}^{(\ell)}$
  Compute FAS right hand side: $\mathbf{g}^{(\ell+1)} = R_\ell^{\ell+1}(\mathbf{g}^{(\ell)} - A_\ell(\mathbf{v}^{(\ell)})) + A_{\ell+1}(R_\ell^{\ell+1}\mathbf{v}^{(\ell)})$
  Solve on the next level : MGRIT-FAS($\ell+1$, $\mathbf{u}^{(\ell+1)}$, $\mathbf{g}^{(\ell+1)}$)
  Correct: $\mathbf{u}^{(\ell)} \leftarrow \mathbf{v}^{(\ell)} + P_{\ell+1}^{\ell}(\mathbf{u}^{(\ell+1)} - R_\ell^{\ell+1}\mathbf{v}^{(\ell)})$
end if

3 Equivalence with the MGRIT V-cycle

The initial value problems that are solved by the propagators can also be written as linear systems of the type $A_\ell\mathbf{u}^{(\ell)} = \mathbf{g}^{(\ell)}$, where

$$A_\ell = \begin{bmatrix} I & & & \\ -\Phi_\ell & I & & \\ & \ddots & \ddots & \\ & & -\Phi_\ell & I \end{bmatrix}.$$

The index ℓ here indicates the level of coarseness of the temporal grid, with $\ell = 0$ being the finest grid, and $\ell = 2$ being the coarsest for a three-level method. Such systems can be solved using the MGRIT V-cycle with F-relaxation algorithm, which can be written in correction form [3] or as a full approximation scheme (FAS) [4], see Algorithms 1 and 2. Here, we consider the special case of $L = 2$, i.e., the three-level algorithm. For the purpose of writing the recurrence, we will index the fine grid (level-0) solution as $u_{i,j,k} \approx u(t_{i,j,k})$. The level-1 vectors will be double indexed as $u_{i,j} \approx u(t_{i,j})$, and level-2 vectors are singly indexed as $u_i \approx u(T_{i-1})$. If injection is used for $P_{\ell+1}^{\ell}$ and $R_\ell^{\ell+1} = (P_{\ell+1}^{\ell})^T$ in Algorithm 2, then one V-cycle of MGRIT-FAS with F-relaxation for solving $A_0\mathbf{u} = f$ updates the iterate $u_{i,j,k}$ as follows:

1. Relax on level 0:

$$v_{i,j,k} = \begin{cases} f_{i,j,k} + \Phi_0 v_{i,j,k-1}, & 1 \le k \le n-1, \quad \forall i,j, \\ u_{i,j,0}, & k = 0. \end{cases}$$

2. Compute FAS right-hand side for level 1:

$$g_{i,j} = \begin{cases} f_{i,j,0} + \Phi_0 v_{i,j-1,n-1} - \Phi_1 u_{i,j-1,0}, & 1 \le j \le m-1, \\ f_{i,0,0} + \Phi_0 v_{i-1,m-1,n-1} - \Phi_1 u_{i-1,m-1,0}, & j = 0. \end{cases}$$

3. Relax on level 1 using initial guess $(\mathbf{u}^{(1)})_{i,j} = v_{i,j,0} = u_{i,j,0}$:

$$v_{i,j} = \begin{cases} f_{i,j,0} + \Phi_0 v_{i,j-1,n-1} + \Phi_1 (v_{i,j-1} - u_{i,j-1,0}), & 1 \le j \le m-1, \\ u_{i,0,0}, & j = 0. \end{cases}$$

4. Compute FAS right-hand side for level 2:

$$g_i = f_{i,0,0} + \Phi_0 v_{i-1,m-1,n-1} + \Phi_1 (v_{i-1,m-1} - u_{i-1,m-1,0}) - \Phi_2 u_{i-1,0,0}.$$

5. Solve the level-2 system:

$$u_i^{\text{new}} = f_{i,0,0} + \Phi_0 v_{i-1,m-1,n-1} + \Phi_1 (v_{i-1,m-1} - u_{i-1,m-1,0}) + \Phi_2 (u_{i-1}^{\text{new}} - u_{i-1,0,0}).$$

6. Correct on level 1 and then on level 0, using injection for both levels: we set for all $1 \le i \le p$

$$u_{i,j,k}^{\text{new}} = \begin{cases} f_{i,j,k} + \Phi_0 v_{i,j,k-1}, & 1 \le k \le n-1, \ \forall\, j, \\ f_{i,j,0} + \Phi_0 v_{i,j-1,n-1} + \Phi_1 (v_{i,j-1} - u_{i,j-1,0}), & k = 0, 1 \le j \le m-1, \end{cases}$$
$$u_{i,0,0}^{\text{new}} = f_{i,0,0} + \Phi_0 v_{i-1,m-1,n-1} + \Phi_1 (v_{i-1,m-1} - u_{i-1,m-1,0}) + \Phi_2 (u_{i-1}^{\text{new}} - u_{i-1,0,0}).$$

We can now prove the following equivalence theorem.

Theorem 1 *For the linear problem $u' = \Phi u + f(t)$, assume that $u_{i,j,k}^0$ satisfies*

$$u_{1,0,0}^0 = u_0, \quad u_{i,0,0}^0 = H u_{i-1,0,0}^0 \ \forall\, i \ge 1, \quad u_{i,j,0}^0 = G u_{i,j-1,0}^0 \ \forall\, j = 1, \ldots, m-1.$$

Then for all $\nu \ge 0$, the three-level MGRIT-FAS V-cycle with F-relaxation and with injection as the prolongation operator is equivalent to three-level parareal via

$$u_{i,j,k}^{\nu+1} = \begin{cases} F_0^k U_{i,j}^\nu, & 1 \le k \le n-1, \quad \forall\, i, j, \\ U_{i,j}^{\nu+1}, & k = 0, 1 \le j \le m-1, \quad \forall\, i \ge 1, \\ Y_{i-1}^{\nu+1}, & j = k = 0, \quad \forall\, i \ge 1. \end{cases}$$

Proof From the initialization conditions, we have for $\nu = 0$ that $u_{i,j,0}^\nu = U_{i,j}^\nu$ for $1 \le j \le m-1$, and $u_{i,0,0}^\nu = Y_{i-1}^\nu$ for all i. We will prove by induction that these two equalities also hold for $\nu \ge 1$. To do so, we rewrite $u_{i,j,k}^{\text{new}}$ in terms of the propagators F_0, F, G and H. The update formula at step 6 leads us to consider three cases:

Case 1 ($k \ne 0$). Step 1 at iteration ν reads

$$u_{i,j,k}^{\text{new}} = v_{i,j,k} = F_0 v_{i,j,k-1} = \cdots = F_0^k v_{i,j,0} = F_0^k U_{i,j}^\nu.$$

Case 2 ($k = 0$, $j \neq 0$). This case is given by step 3, where

$$u^{\text{new}}_{i,j,0} = v_{i,j} = F_0 v_{i,j-1,n-1} + \Phi_1(v_{i,j-1} - U^{\nu}_{i,j-1}) = F_0^n U^{\nu}_{i,j-1} + G v_{i,j-1} - G U^{\nu}_{i,j-1}.$$

Here, we have replaced the difference of Φ_1 by a difference of G, because G is affine. Thus, we have $v_{i,j} = U^{\nu+1}_{i,j}$ for $1 \le j \le m-1$, since both quantities are initialized the same way (we have $v_{i,0} = u_{i,0,0} = Y^{\nu}_{i-1} = U^{\nu+1}_{i,0}$) and satisfy the same recurrence.

Case 3 ($j = k = 0$). Here we have $u^{\text{new}}_{i,0,0} = u^{\text{new}}_i$, so step 5 gives, for $i \ge 2$,

$$\begin{aligned} u^{\text{new}}_i &= f_{i,0,0} + \Phi_0 v_{i-1,m-1,n-1} + \Phi_1(v_{i-1,m-1} - u_{i-1,m-1,0}) + \Phi_2(u^{\text{new}}_{i-1} - u_{i-1,0,0}) \\ &= F_0 v_{i-1,m-1,n-1} + \Phi_1(v_{i-1,m-1} - U^{\nu}_{i-1,m-1}) + \Phi_2(u^{\text{new}}_{i-1} - u_{i-1,0,0}) \\ &= F_0^n U^{\nu}_{i-1,m-1} + G v_{i-1,m-1} - G U^{\nu}_{i-1,m-1} + H u^{\text{new}}_{i-1} - H u_{i-1,0,0} \\ &= U^{\nu+1}_{i-1,m} + H u^{\text{new}}_{i-1} - H Y^{\nu}_{i-2}. \end{aligned}$$

For $i = 1$, we have $u^{\text{new}}_1 = u_0 = Y^{\nu+1}_0$; thus, u^{new}_i and $Y^{\nu+1}_{i-1}$ satisfy the same recurrence with the same initial condition. This leads to $u^{\text{new}}_{i,0,0} = Y^{\nu+1}_{i-1}$ for all i, as claimed. □

We can now use the FAS formulation to deduce the equivalence in classical (correction) form. We define the following operators:

$$E_\ell = I - P^{\ell}_{\ell+1} R^{\ell+1}_{\ell}, \qquad M_\ell = \operatorname{diag}((A_\ell)_{11}, (A_\ell)_{22}, \ldots),$$

where $(A_\ell)_{ii}$ are diagonal blocks of A_ℓ corresponding to the ith subinterval, *starting* with the coarse point and including all the fine points until (but excluding) the next coarse point. In other words, M_ℓ is the block Jacobi smoother for level ℓ, and E_ℓ blanks out the coarse points and retains the fine points when applied to a vector of values at level ℓ. Similar operators were defined in [10], where the authors proved the equivalence between two-level parareal and a geometric multigrid method with block Jacobi smoothing and aggressive coarsening in the FAS setting; however, the blocks in [10] are defined differently, with the coarse points appearing at the end of the block rather than the beginning. We write the change in the solution at step 6 as

$$u^{\text{new}}_{i,j,k} - u_{i,j,k} = \begin{cases} v_{i,j,k} - u_{i,j,k} =: (\Delta \mathbf{u}^{(0)})_{i,j,k}, & k \neq 0, \\ v_{i,j} - u_{i,j,0} =: (\Delta \mathbf{u}^{(1)})_{i,j}, & k = 0,\ j \neq 0, \\ u^{\text{new}}_i - u_{i,0,0} =: (\Delta \mathbf{u}^{(2)})_i, & j = k = 0. \end{cases}$$

To compute $\Delta \mathbf{u}^{(0)}$, note that $v_{i,j,k} - u_{i,j,k} = 0$ when $k = 0$; for $k \neq 0$, we have

$$\begin{aligned} (\Delta \mathbf{u}^{(0)})_{i,j,k} = v_{i,j,k} - u_{i,j,k} &= f_{i,j,k} + \Phi_0(v_{i,j,k-1} - u_{i,j,k-1}) + \Phi_0 u_{i,j,k-1} - u_{i,j,k} \\ &= (\mathbf{f} - A_0 \mathbf{u})_{i,j,k} + \Phi_0(\Delta \mathbf{u}^{(0)})_{i,j,k-1}. \end{aligned}$$

If we move $\Phi_0(\Delta \mathbf{u}^{(0)})_{i,j,k-1}$ to the left and recall the definition of M_0, we get

$$M_0 \Delta \mathbf{u}^{(0)} = E_0(\mathbf{f} - A_0 \mathbf{u}) \implies \Delta \mathbf{u}^{(0)} = M_0^{-1} E_0 \tilde{\mathbf{g}}^{(0)},$$

where $\tilde{\mathbf{g}}^{(0)} = \mathbf{f} - A_0\mathbf{u}$ is the initial residual. This is almost the same as in [10], except the residual is blanked before the smoothing, instead of after. Next, we calculate

$$(\Delta\mathbf{u}^{(1)})_{i,j} = v_{i,j} - u_{i,j,0} = \begin{cases} 0, & j = 0, \\ g_{i,j} + \Phi_1(v_{i,j-1} - u_{i,j-1,0}) + \Phi_1 u_{i,j-1,0} - u_{i,j,0}, & \\ & j \neq 0, \end{cases}$$

which implies

$$M_1\Delta\mathbf{u}^{(1)} = E_1(\mathbf{g}^{(1)} - A_1R_0^1\mathbf{u}) = E_1R_0^1(\mathbf{f}^{(0)} - A_0(\mathbf{u} + \Delta\mathbf{u}^{(0)})).$$

Thus, $\Delta\mathbf{u}^{(1)} = M_1^{-1}E_1\tilde{\mathbf{g}}^{(1)}$, where $\tilde{\mathbf{g}}^{(1)} = R_0^1(\tilde{\mathbf{g}}^{(0)} - A_0\Delta\mathbf{u}^{(0)})$. Finally, we have

$$(\Delta\mathbf{u}^{(2)})_i = u_i^{\text{new}} - u_{i,0,0} = g_i + \Phi_2(u_{i-1}^{\text{new}} - u_{i-1,0,0}) + \Phi_2 u_{i-1,0,0} - u_{i,0,0},$$

which leads to

$$A_2\Delta\mathbf{u}^{(2)} = \mathbf{g}^{(2)} - A_2R_0^2\mathbf{u} = R_1^2(\mathbf{g}^{(1)} - A_1(R_0^1\mathbf{u} + \Delta\mathbf{u}^{(1)})) = R_1^2(\tilde{\mathbf{g}}^{(1)} - A_1\Delta\mathbf{u}^{(1)}).$$

We conclude, by replacing $\Delta\mathbf{u}^{(1)}$ with $M_1^{-1}E_1\tilde{\mathbf{g}}^{(1)}$ in the last step, that

$$\begin{aligned} \mathbf{u}^{\text{new}} - \mathbf{u} &= \Delta\mathbf{u}^{(0)} + P_1^0\Delta\mathbf{u}^{(1)} + P_2^0\Delta\mathbf{u}^{(2)} \\ &= \Delta\mathbf{u}^{(0)} + P_1^0(\Delta\mathbf{u}^{(1)} + P_2^1A_2^{-1}R_1^2(\tilde{\mathbf{g}}^{(1)} - A_1\Delta\mathbf{u}^{(1)})) \\ &= \Delta\mathbf{u}^{(0)} + P_1^0((I - P_2^1A_2^{-1}R_1^2A_1)M_1^{-1}E_1 + P_2^1A_2^{-1}R_1^2)\tilde{\mathbf{g}}^{(1)} \end{aligned}$$

Defining $T = (I - P_2^1A_2^{-1}R_1^2A_1)M_1^{-1}E_1 + P_2^1A_2^{-1}R_1^2$, we continue to calculate

$$\begin{aligned} \mathbf{u}^{\text{new}} - \mathbf{u} &= \Delta\mathbf{u}^{(0)} + P_1^0TR_0^1(\tilde{\mathbf{g}}^{(0)} - A_0\Delta\mathbf{u}^{(0)}) \\ &= (P_1^0TR_0^1 + (I - P_1^0TR_0^1A_0)M_0^{-1}E_0)(\mathbf{f} - A\mathbf{u}) =: \mathcal{P}(\mathbf{f} - A\mathbf{u}). \end{aligned}$$

We conclude that the error propagator reads

$$\mathcal{S} = I - \mathcal{P}A_0 = (I - P_1^0TR_0^1A_0)(I - M_0^{-1}E_0A_0),$$

where the operator T satisfies $I - TA_1 = (I - P_2^1A_2^{-1}R_1^2A_1)(I - M_1^{-1}E_1A_1)$. Note that the preconditioners $\mathcal{P}$ and T can also be written as

$$\mathcal{P} = M_0^{-1}E_0 + P_1^0TR_0^1(I - A_0M_0^{-1}E_0), \qquad T = M_1^{-1}E_1 + P_2^1A_2^{-1}R_1^2(I - A_1M_1^{-1}E_1).$$

We can hence interpret the action of the preconditioner $\mathcal{P}$ as follows:

1. $M_0^{-1}E_0$: Take the fine residual, blank out the coarse points and apply block Jacobi.
2. $I - A_0M_0^{-1}E_0$: Update the residual after relaxation.
3. $P_1^0TR_0^1$: Restrict the new residual, recursively solve the coarse problem, then update the coarse points by injection.

Since T acts the same way but at a coarser level, the action of $\mathcal{P}$ corresponds to exactly one MGRIT V-cycle with F-relaxation, written in correction form.

Remark If one replaces injection with injection plus F-relaxation (like in standard MGRIT), then the equivalent parareal formulation at the νth iteration would be

$$\begin{aligned}
U_{i,0}^{\nu+1/2} &= Y_{i-1}^{\nu}, & U_{i,j}^{\nu+1/2} &= GU_{i,j-1}^{\nu+1/2} + FU_{i,j-1}^{\nu} - GU_{i,j-1}^{\nu} \qquad (1 \le j \le m),\\
Y_0^{\nu+1} &= u_0, & Y_i^{\nu+1} &= U_{im}^{\nu+1/2} + HY_{i-1}^{\nu+1} - HY_{i-1}^{\nu},\\
U_{i,0}^{\nu+1} &= Y_{i-1}^{\nu+1}, & U_{i,j}^{\nu+1} &= GU_{i,j-1}^{\nu+1} + FU_{i,j-1}^{\nu} - GU_{i,j-1}^{\nu} \qquad (1 \le j \le m).
\end{aligned}$$

Note that the term $FU_{i,j-1}^{\nu} - GU_{i,j-1}^{\nu}$ is used twice, but it only needs to be computed once using a fine propagation. The intermediate propagation G, however, needs to be computed twice, since it is applied once to $U_{i,j-1}^{\nu+1/2}$, and another time to $U_{i,j-1}^{\nu+1}$.

4 Numerical example

We present the numerical example in [7], where the advection-diffusion equation $u_t = u_x + \kappa u_{xx}$ with periodic boundary conditions $u(0,t) = u(2,t)$, $u_x(0,t) = u_x(2,t)$ is solved on $t \in (0,4)$, with $\kappa = 1/1024$ (advection-dominated case) and $u(x,0) = e^{-20(x-1)^2}$. We discretize the problem using second order finite difference in space and backward Euler in time, with $\Delta x = 1/20$ and $\delta t = 1/1280$. For two-level parareal, the coarse propagator is backward Euler with $\Delta T = 1/2$ (8 coarse steps with 640 fine steps per coarse step). For three-level parareal, we use an intermediate level with $\Delta t = 1/128$ (10 fine steps per intermediate step), while keeping $\Delta T = 1/2$ for the coarsest level (i.e., 64 intermediate steps per coarse step). In Figure 1, we compare two-level and three-level parareal, both with and without post-smoothing. We compare both the iteration count and the *idealized* running time, as measured by the number of *non-concurrent* backward Euler steps taken at all levels; this cost is normalized by that of sequential time-stepping, so that a cost of 1 means the same cost as sequential time-stepping without parallelization. We see that two-level parareal

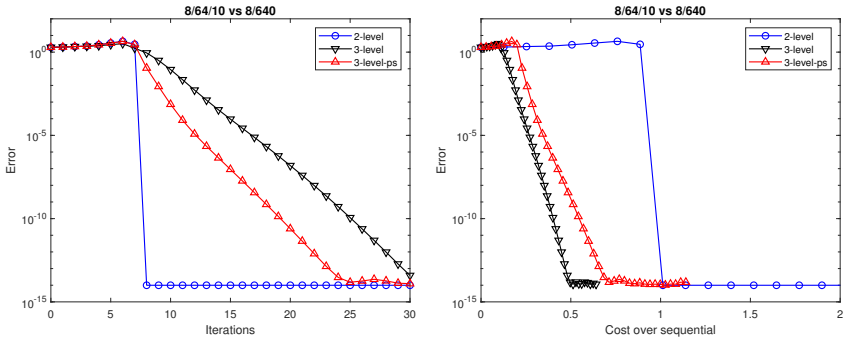


Fig. 1 Left: Iteration count for two-level parareal, and three-level parareal, with and without post-smoothing. Right: Computational cost of the three methods, as measured by the number of backward Euler steps taken, normalized by the cost of sequential time-stepping.

converges to the exact solution in 8 iterations, whereas the three-level variants take many more iterations. However, the three-level iterations are much more parallel and take less time to run than a two-level iteration. In particular, both three-level versions converge with cost much lower than 1; such speedup is not possible for two-level parareal. Finally, although post-smoothing reduces the number of three-level iterations, the higher cost per iteration (two intermediate propagations rather than one) makes it slower than no post-smoothing once the normalized cost is considered.

Acknowledgements F. Kwok acknowledges support from the National Science and Engineering Research Council of Canada (RGPIN- 2021-02595). The work described in this paper is partially supported by a grant from the ANR/RGC joint research scheme sponsored by the Research Grants Council of the Hong Kong Special Administrative Region, China and the French National Research Agency (Project no. A-CityU203/19). This project has received funding from the Federal Ministry of Education and Research and the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955701, Time-X. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Belgium, France, Germany, Switzerland.

References

1. Bal, G. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In: *Domain Decomposition Methods in Science and Engineering XV*, 425–432. Springer (2005).
2. Dobrev, V., Kolev, T., Petersson, N. A., and Schroder, J. B. Two-level convergence theory for multigrid reduction in time (MGRIT). *SIAM J. Sci. Comput.* **39**(5), S501–S527 (2017).
3. Falgout, R. D., Friedhoff, S., Kolev, T. V., MacLachlan, S. P., and Schroder, J. B. Parallel time integration with multigrid. *SIAM J. Sci. Comput.* **36**(6), C635–C661 (2014).
4. Falgout, R. D., Katz, A., Kolev, T. V., Schroder, J. B., Wissink, A., and Yang, U. M. Parallel time integration with multigrid reduction for a compressible fluid dynamics application. Tech. Rep. LLNL-JRNL-663416, Lawrence Livermore National Laboratory (2015).
5. Friedhoff, S. and MacLachlan, S. A generalized predictive analysis tool for multigrid methods. *Numer. Linear Algebra Appl.* **22**(4), 618–647 (2015).
6. Gander, M. J. 50 years of time parallel time integration. In: *Multiple Shooting and Time Domain Decomposition Methods*, 69–113. Springer (2015).
7. Gander, M. J. Five decades of time parallel time integration, and a note on the degradation of the performance of the Parareal algorithm as a function of the Reynolds number. *Oberwolfach Report* (2017).
8. Gander, M. J. and Hairer, E. Nonlinear convergence analysis for the parareal algorithm. In: *Domain Decomposition Methods in Science and Engineering XVII*, 45–56 (2008).
9. Gander, M. J., Kwok, F., and Zhang, H. Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT. *Comput. Visualization Sci.* **19**(3), 59–74 (2018).
10. Gander, M. J. and Vandewalle, S. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.* **29**(2), 556–578 (2007).
11. Hessenthaler, A., Southworth, B. S., Nordsletten, D., Röhrle, O., Falgout, R. D., and Schroder, J. B. Multilevel convergence analysis of multigrid-reduction-in-time. *SIAM J. Sci. Comput.* **42**(2), A771–A796 (2020).
12. Kwok, F. Analysis of a three-level variant of parareal (In preparation, 2022).
13. Lions, J.-L., Maday, Y., and Turinici, G. Résolution d'EDP par un schéma en temps «pararéel». *C. R. Acad. Sci., Ser. I: Math.* **332**(7), 661–668 (2001).
14. Ong, B. W. and Schroder, J. B. Applications of time parallelization. *Computi. Visualization Sci.* **23**(1), 1–15 (2020).
15. Southworth, B. S. Necessary conditions and tight two-level convergence bounds for parareal and multigrid reduction in time. *SIAM J. Matrix Anal. Appl.* **40**(2), 564–608 (2019).

Coupling Dispersive Shallow Water Models by Deriving Asymptotic Interface Operators

José Galaz, Maria Kazolea, and Antoine Rousseau

1 Introduction

We are interested in coupling the linear Green-Naghdi equations (LGNE)

$$\partial_t \zeta + \partial_x V = 0, \tag{1}$$
$$\partial_t V + \partial_x \zeta = \phi, \tag{2}$$
$$-\frac{\mu}{3}\partial_x^2 \phi + \phi = -\frac{\mu}{3}\partial_x^3 \zeta \tag{3}$$

for $x < 0$ and $t \in (0, T)$, with the linear shallow water equations (LSWE)

$$\partial_t \zeta + \partial_x V = 0, \tag{4}$$
$$\partial_t V + \partial_x \zeta = 0 \tag{5}$$

for $x > 0$ and $t \in (0, T)$ with $T > 0$, to represent the 1D propagation of water waves in shallow water. Here ∂_t, ∂_x denote partial derivatives in the time and space variables t, x; $V(t, x)$ and $\zeta(t, x)$ stand for the vertically-averaged velocity and the free-surface level over its state at rest; $\phi(t, x)$ is an auxiliar variable for the elliptic part of the problem; and $\mu > 0$ is the asymptotic parameter characterizing the wave dispersion.

In the nonlinear case, Boussinesq-type equations, such as the Green-Naghdi equations (GNE), have been coupled with the nonlinear shallow water equations (NSWE) to take advantage of their physical-modeling features: the dispersive terms

José Galaz, Antoine Rousseau
INRIA, Team LEMON, Centre Inria d'Université Côte d'Azur, Antenne Montpellier, Bat 5 CC05 017, 860 rue Saint-Priest, 34095 Montpellier Cedex 5 France, e-mail: jose.galaz@inria.fr, antoine.rousseau@inria.fr

Maria Kazolea
INRIA, Team CARDAMOM, INRIA Bordeaux-Sud-Ouest, 200 Avenue de la Vieille Tour, 33405 Talence cedex France e-mail: maria.kazolea@inria.fr

in the GNE can be used to accurately represent the phase and amplitude of waves in the shoaling zone, while shock-capturing well-balanced finite volume schemes for the NSWE can mimic the energy dissipation of wave-breaking and provide a robust handling of vanishing water depths without ad-hoc parametrizations. However, this coupled model has been shown to be unstable unless dissipative terms are added [6].

Since this coupling is a "divide and conquer" type of problem, domain decomposition methods (DDM) can help to obtain further insights. Usually, coupling conditions have been derived for each equation on a case-by-case basis (e.g., [1, 2, 5, 9]). Here we explore a different approach, based on the steps of the derivation of the GNE and NSWE [7, ch. 1 and 5]. First, a DDM of the linearized Euler equations in the discrete level is defined, based on the Neumann-Dirichlet method. Then, recalling that both GNE and NSWE derive from the Euler equations, we derive transmission conditions for the children asymptotic equations by taking the vertical average of the original operators and truncating the resulting expression according to an asymptotic expansion of the velocity potential. We examine this approach in the homogeneous case first, when coupling LGNE with LGNE, and then in the heterogeneous case, coupling the LGNE with the LSWE.

2 A domain decomposition of the free-surface Euler equations

The linear Euler equations for an incompressible fluid and irrotational flow in one horizontal dimension can be formulated as an elliptic problem for the velocity potential $\Phi(t,x,z)$ and two evolution equations for the free-surface function $\zeta(t,x)$ and the trace of the potential at the surface $\psi(t,x)$ respectively (see [7, ch. 1.1.3]). A finite-difference discretization of the equations for $\zeta_i^n = \zeta(t^n, x_i)$ and $\psi_i^n = \psi(t^n, x_i)$, in an uniform grid $x_i = i\Delta x$ and $t^n = n\Delta t$, with $i \in \mathcal{N} = \{1 \dots N_x - 2\}$ and $n = 1, 2, ..,$ is given by

$$\frac{\zeta_i^{n+1} - \zeta_i^n}{\Delta t} + \frac{V_i^n - V_{i-1}^n}{\Delta x} = 0 \quad \text{for } i \in \mathcal{N}, \tag{6}$$

$$\frac{\psi_i^{n+1} - \psi_i^n}{\Delta t} + \zeta_i^n = 0 \quad \text{for } i \in \mathcal{N}, \tag{7}$$

where V_i^n is the discrete vertically-averaged velocity cf. [7, ch. 3.31] given by

$$V_i^n = \sum_{j=1}^{N_z - 1} \frac{\Phi_{i+1,j}^n - \Phi_{i,j}^n}{\Delta x} \Delta z. \tag{8}$$

Boundary conditions at nodes $i = 0$ and $i = N_x - 1$ can be $V_i^n = 0$ and $\zeta_0^n = \zeta_1^n$, $\zeta_{N_x-1}^n = \zeta_{N_x-2}^n$ [8, eqs. (28), (37)]. In equation (8), $\Phi_{i,j}^n = \Phi(t^n, x_i, z_j)$ is the discrete velocity potential computed from

$$\mu\frac{\Phi^n_{i+1,j}+\Phi^n_{i-1,j}-2\Phi^n_{i,j}}{\Delta x^2}+\frac{\Phi^n_{i,j+1}+\Phi^n_{i,j-1}-2\Phi^n_{i,j}}{\Delta z^2}=0$$
$$\text{for } i,j\in\mathcal{N}\times\{1,\dots,N_z-1\} \tag{9}$$

on a grid (x_i, z_j) such that $z_j = j\Delta z - 1$ and $z_{N_z-1} = 0$ and its boundary conditions are $\Phi^n_{i,N_z-1} = \psi^n_i$, $\Phi_{i,0} = \Phi_{i,1}$, $\Phi_{0,j} = \Phi_{1,j}$ and $\Phi_{N_x-1,j} = \Phi_{N_x-2,j}$

We decompose the domain in two components with a vertical interface located at $i = l \in \mathcal{N}$. To do this let $\zeta^{n+1}_{i,s}$, $\psi^{n+1}_{i,s}$, $\Phi^{n+1}_{i,j,s}$, $V^n_{i,s}$ be the values of the unknowns on each subdomain $s = 1, 2$. These variables are computed from equations (6), (7), (8) but with i in $\mathcal{N}_s$ instead of $\mathcal{N}$, with $\mathcal{N}_1 = \{1, .., l-1\}$ and $\mathcal{N}_2 = \{l, \dots, N_x - 2\}$; equation (9) is solved with $i \in \mathcal{N}_1$ for $s = 1$ and $i \in \mathcal{N}_2 \setminus \{l\}$ for $s = 2$. To obtain the same solution as the monodomain problem, equations (6) and (7) are complemented with Dirichlet transmission conditions

$$\begin{aligned}\zeta^{n+1}_{l,1}=\zeta^{n+1}_{l,2}\quad V^{n+1}_{l,1}=V^{n+1}_{l,2},\\ \zeta^{n+1}_{l-1,2}=\zeta^{n+1}_{l-1,1}\ V^{n+1}_{l-1,2}=V^{n+1}_{l-1,1},\end{aligned} \tag{10}$$

while equation (9) uses Neumann and Dirichlet transmission conditions

$$\frac{\Phi^n_{l,j,1}-\Phi^n_{l-1,j,1}}{\Delta x}-\frac{\Delta x}{2\mu\Delta z^2}\left(\Phi^n_{l,j+1,1}+\Phi^n_{l,j-1,1}-2\Phi^n_{l,j,1}\right) \tag{11}$$
$$=\frac{\Phi^n_{l+1,j,2}-\Phi^n_{l,j,2}}{\Delta x}+\frac{\Delta x}{2\mu\Delta z^2}\left(\Phi^n_{l,j+1,2}+\Phi^n_{l,j-1,2}-2\Phi^n_{l,j,2}\right),$$
$$\Phi^n_{l,j,2}=\Phi^1_{l,j,2}, \tag{12}$$

which include an $O(\Delta x)$ term necessary in finite-difference schemes to preserve the monodomain solution [4]. This scheme satisfies

- $\Phi^n_{i,j,*} = \Phi^n_{i,j}$ with $\Phi^n_{i,j,*} = \Phi^n_{i,j,1}$ if $i \le l$ and $\Phi^n_{i,j} = \Phi^n_{i,j,2}$ if $i > l$. This means that the solution $\Phi^n_{i,j,*}$ formed by both subdomains will be equal to the monodomain solution $\Phi^n_{i,j}$
- A parallel or alternating method to solve (9) with (12) will be convergent if $L_1 < L_2$, with $L_1 = l\Delta x$ and $L_2 = (N_x - l)\Delta x$ (see [3, eq. (2.5) with $\theta = 1$] for example).

3 Asymptotic domain-decomposition method

In the first part of this section we drop the time superscript n and introduce the asymptotic-degree superscript $(k) = (1), (2), \dots$. We need an asymptotic expansion $\Phi_{i,j} = \Phi^{(0)}_{i,j} + \mu\Phi^{(1)}_{i,j} + \cdots + \mu^k\Phi^{(k)}_{i,j}$ of the solution to the discrete Laplace equation (9). At first order $\Phi_{i,j} = \Phi^{(0)}_{i,j} + O(\mu)$ which substituted on equation (9) and discarding $O(\mu)$ terms leads to $\Phi^{(0)}_{i,j+1} + \Phi^{(0)}_{i,j-1} - 2\Phi^{(0)}_{i,j} = 0$, whose solution is

$\Phi^{(0)}_{i,j} = \psi_i$ so the first order expansion is

$$\Phi_{i,j} = \psi_i + O(\mu). \tag{13}$$

Similarly, at second order, replacing $\Phi_{i,j} = \Phi^{(0)}_{i,j} + \mu\Phi^{(1)}_{i,j} + O(\mu^2)$ into (9)

$$\frac{\Phi^{(1)}_{i,j+1} + \Phi^{(1)}_{i,j-1} - 2\Phi^{(1)}_{i,j}}{\Delta x^2} = -\frac{\psi_{i+1} + \psi_{i-1} - 2\psi_i}{\Delta x^2}, \tag{14}$$

whose solution gives us the second order expansion

$$\Phi_{i,j} = \psi_i - \mu\frac{\Delta z^2}{2\Delta x^2}(\psi_{i+1} + \psi_{i-1} - 2\psi_i)(j^2 - j - (N_z - 1)(N_z - 2)) + O(\mu^2). \tag{15}$$

Substituting (13) into (8) and using that $(N_z - 1)\Delta z = 1$ one obtains that at first order

$$\frac{\psi_{i+1} - \psi_i}{\Delta x} = V_i + O(\mu). \tag{16}$$

From (15) we can proceed similarly to obtain the second order expansion for V_i

$$V_i = \frac{\psi_{i+1} - \psi_i}{\Delta x} - \nu T\left(\frac{\psi_{i+1} - \psi_i}{\Delta x}\right) + O(\mu^2), \tag{17}$$

where $TV_i = -(V_{i+1} + V_{i-1} - 2V_i)/(3\Delta x^2)$ and $\nu = \mu(1 - \Delta z/2)(1 - \Delta z)$. We can now substitute the first order approximation (16) into (17) and isolate $(\psi_{i+1} - \psi_i)/\Delta x$ to obtain

$$\frac{\psi_{i+1} - \psi_i}{\Delta x} = V_i + \nu T V_i + O(\mu^2). \tag{18}$$

To substitute (18) into (7) let us apply a forward finite-difference in x to equation (7). Introducing the notation $D^+_x f_i = (f_{i+1} - f_i)/\Delta x$, $D^+_z f_j = (f_{j+1} - f_j)/\Delta z$ and $D^-_z f_j = (f_j - f_{j-1})/\Delta z$, and the time superscript n, (8) becomes $V^n_i = \sum_{j=1}^{N_z - 1} D^+_x \Phi^n_{i,j}\Delta z$. Using the asymptotic expansion (18), discarding terms of size $O(\mu^2)$ and rearranging one finally obtains the discrete momentum equation of the LGNE

$$\begin{aligned} \frac{V^{n+1}_i - V^n_i}{\Delta t} + \frac{\zeta^n_{i+1} - \zeta^n_i}{\Delta x} &= \phi^n_i \\ (1 + \nu T)\phi^n_i &= \nu T(D^+_x \zeta^n_i) \end{aligned} \tag{19}$$

with ϕ^n_i an auxiliar variable for the new elliptic problem. And discarding all $O(\mu)$ terms the discrete momentum equation of the LSWE reads

$$\frac{V^{n+1}_i - V^n_i}{\Delta t} + \frac{\zeta^n_{i+1} - \zeta^n_i}{\Delta x} = 0. \tag{20}$$

We can proceed in a similar fashion to derive asymptotic versions of the Neumann boundary condition. To do this let us multiply equation (12) by Δz and sum it up

from $j = 1$ to $j = N_z - 2$. Using the formula for V_i and simplifying, one obtains that (12) can be written as

$$\begin{aligned} V^n_{l-1,1} - D^+_x\Phi^n_{l-1,N_z-1,1}\Delta z - \frac{\Delta x}{2\mu}D^-_z\Phi^n_{l,N_z-1,1} \\ = V^n_{l,2} - D^+_x\Phi^n_{l,N_z-1,2}\Delta z + \frac{\Delta x}{2\mu}D^-_z\Phi^n_{l,N_z-1,2}. \end{aligned} \tag{21}$$

To further simplify the remaining Φ terms we will use that $\partial_z\Phi_{z=0}/\mu = \partial_x V + O(\varepsilon)$ (from Ref. [7, eqs. (1.29) and Proposition 3.35]), and substitute the discrete derivatives to finally obtain $1/\mu D^-_z\Phi_{i,N_z-1} = -D^p_x V_i + O(\varepsilon, \Delta z/\mu, \Delta x^p)$, where $D^p_x V_i = \partial_x V_{|x=x_i} + O(\Delta x^p)$ is a finite-difference operator to be defined. If we replace this back into the equation, use equation (13) written as $D_x\Phi_{i,N_z-1} = V_i + O(\mu)$, and replace D^p_x with the backward finite difference D^-_x in the left-hand-side of the equation and a forward finite difference D^+_x in the right-hand-side, we can write the original Neumann boundary condition as

$$\frac{1}{2}(V^n_{l,1} + V^n_{l-1,1}) - V^n_{l-1,1}\Delta z = \frac{1}{2}(3V^n_{l,2} - V^n_{l+1,2}) - V^n_{l,2}\Delta z. \tag{22}$$

Substituting the auxiliar variable for the LGNE, $\phi^n_{i,s} = D^+_t V^n_{i,s} + D^+_x\zeta^n_{i,s}$.

$$\begin{aligned} \frac{1}{2}(\phi^n_{l,1} + \phi^n_{l-1,1}) - \phi^n_{l-1,1}\Delta z = \frac{1}{2}(3\phi^n_{l,2} - \phi^n_{l+1,2}) - \phi^n_{l,2}\Delta z \\ - \left(\frac{3\Delta x^2}{2}T(D^+_x\zeta^n_l) - (D^+_x\zeta^n_l - D^+_x\zeta^n_{l-1})\Delta z\right). \end{aligned} \tag{23}$$

Summarizing, for the homogeneous case, the domain decomposition of the LGNE reads, for each subdomain $s = 1, 2$,

$$\begin{aligned} \frac{\zeta^{n+1}_i - \zeta^n_i}{\Delta t} + \frac{V^n_i - V^n_{i-1}}{\Delta x} = 0 \text{ for } i \in \mathcal{N}_s \\ \frac{V^{n+1}_i - V^n_i}{\Delta t} + \frac{\zeta^n_{i+1} - \zeta^n_i}{\Delta x} = \phi_i \text{ for } i \in \mathcal{N}_s \\ \phi^n_{i,s} + \nu T\phi^n_{i,s} = \nu T(D^+_x\zeta_i) \quad \text{for } i \in \mathcal{N}_s \\ \phi^n_{0,1} = \phi_{Left} \\ \phi^n_{N_x-1,2} = \phi_{Right} \end{aligned} \tag{24}$$

and at the interface $i = l$ equation (10) holds for $V^n_{l,s}$ and $\zeta^n_{l,s}$, while for $\phi^n_{l,s}$

$$\begin{aligned} \frac{1}{2}(\phi^n_{l,1} + \phi^n_{l-1,1}) - \phi^n_{l-1,1}\Delta z = \frac{1}{2}(3\phi^n_{l,2} - \phi^n_{l+1,2}) - \phi^n_{l,2}\Delta z \\ - \left(\frac{3\Delta x^2}{2}T(D^+_x\zeta^n_l) - (D^+_x\zeta^n_l - D^+_x\zeta^n_{l-1})\Delta z\right) \\ \phi^n_{l,2} = \phi^n_{l,1}. \end{aligned} \tag{25}$$

To test this asymptotic domain decomposition method with an additive iterative scheme we compare the monodomain solution with the DDM solution for $x \in (0, 1)$, $\phi_{Left} = 0$, $\phi_{Right} = -0.5$, $\partial_x^3 \zeta^n = -1$, $\mu = 3$, $\Delta x = 0.05$, $\Delta z = \Delta x^2$. These parameters are convenient to avoid overflow in intermediate calculations in the formula of the analytical solution. Figure 1 (left) shows the L^2 distance between the discrete monodomain solution and each subdomain solution at each iteration for an interface located at $x = 0.4$ where we can see that the DDM diverges.

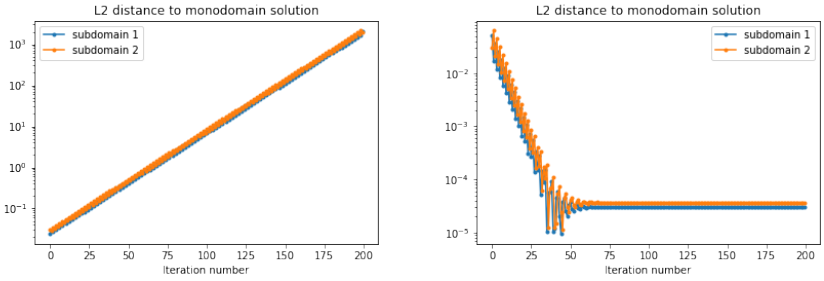


Fig. 1 L2 distance to monodomain of the solution on each subdomain as a function of the iteration number when using (25) and (26) respectively.

To fix this situation we can substract $\phi_{l,1}^n$ from each side of the first equation of (25), use that $\phi_{l,1}^n = \phi_{l,2}^n$, and multiply the equation by $-\frac{2}{\Delta x}$. The coupling conditions become

$$\begin{aligned} \frac{\phi_{l,1}^n - \phi_{l-1,1}^n}{\Delta x} + 2\frac{\Delta z}{\Delta x}\phi_{l-1,1}^n = \frac{\phi_{l+1,2}^n - \phi_{l,2}^n}{\Delta x} + 2\frac{\Delta z}{\Delta x}\phi_{l,2}^n \\ +3\Delta x T(D_x^+\zeta_l^n) - 2\frac{(D_x^+\zeta_l^n - D_x^+\zeta_{l-1}^n)}{\Delta x}\Delta z \\ \phi_{l,2}^n = \phi_{l,1}^n. \end{aligned} \tag{26}$$

As before, Figure 1 (right) shows the L2 distance to the monodomain solution at each iteration of the DDM. In contrast with the previous case now the algorithm converges. This happens because (26) is also consistent at $O(\Delta x)$ with a Neumann boundary condition. To see this notice that $3\Delta x T(D_x^+\zeta_l^n) = O(\Delta x)$ and $(D_x^+\zeta_l^n - D_x^+\zeta_{l-1}^n)\Delta z/\Delta x = O(\Delta z)$, so if $\Delta z = O(\Delta x^2)$, when taking the limit $\Delta x \to 0$, equation (26) will satisfy $\partial_x \phi_1^n = \partial_x \phi_2^n + O(\Delta x)$ at $x = x_l$, with ϕ_s^n the limit of $\phi_{i,s}^n$ when $\Delta x \to 0$. However, solutions of the subdomains are different from the monodomain discrete solution, since the $O(\Delta x)$ terms in equation (26) lead to a linear system that is not equivalent to the monodomain's.

Defining ϕ_i^* as $\phi_i^* = \phi_i^1$ if $i \leq l$ and $\phi_i^* = \phi_i^2$ if $i > l$, we conclude that in the homogeneous case the asymptotic domain decomposition method:

- Similarly to its parent DDM, it also corresponds to the Neumann-Dirichlet method, so the additive scheme will be convergent when $L_1 < L_2$, with $L_1 = l\Delta x$ and $L_2 = (N_x - l)\Delta x$.
- The monodomain solution ϕ_i is different than ϕ_i^*, the solution formed by each subdomain solution. This is because the asymptotic boundary condition includes $O(\Delta x)$ terms that induce a different linear system than the monodomain. Additional $O(\Delta x)$ terms must be added to fix this situation.

The heterogeneous case. Now we want to use (26) to couple the LGNE with the LSWE. To do this we can add the constraint $\phi_i^2 = 0$ for $i \geq l+1$ to impose the LSWE on the right side of the domain, and ϕ_i^1 satisfying the third equation of system (24), to impose the LGNE on the left side of the domain. If we write the Neumann boundary condition (26) as $(\phi_{l,1}^n - \phi_{l-1,1}^n)/\Delta x = (\phi_{l+1,2}^n - \phi_{l,2}^n)/\Delta x + O(\Delta x)$, use the LSWE $\phi_{l+1,2}^n = 0$ and the Dirichlet boundary condition $\phi_{l,2}^n = \phi_{l,1}^n$, we arrive to the condition $\phi_{l,1}^n = \frac{1}{2}\phi_{l-1,1}^n + O(\Delta x^2)$. An interpretation of this formula is that the Neumann-Dirichlet condition becomes a linear interpolation between $\phi_{l-1,1}^n$ and $\phi_{l+1,2}^n = 0$ plus the $O(\Delta x)$ term.

To test this heterogeneous DDM we manufacture the solitary wave of the GNE into the LGNE and LSWE $\zeta(t,x) = sech^2(x-t+3)$, $V(t,x) = \zeta(t,x)/(1+\varepsilon\zeta(t,x))$ with $\varepsilon = 0.2$, for $(x,t) \in (-11,11) \times (-3,3)$, and an interface located at $x = 0$. By definition any change on its shape must be due to the influence of the asymptotic transmission conditions. Figure 2 shows the results for $\Delta x = 0.05, 0.02, 0.01$, when half of the solitary wave has crossed the interface at $t = 0$ and later at $t = 3$. We see that the interface boundary conditions have introduced oscillations and a discontinuity at the interface whose amplitudes grow as Δx decreases. This is similar to the results reported by [6].

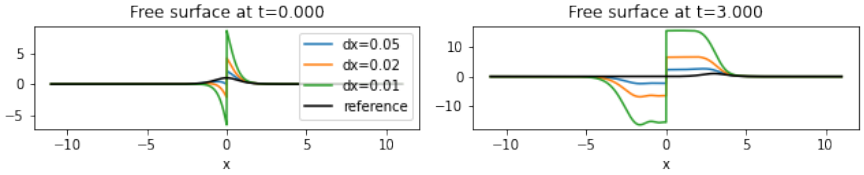


Fig. 2 Comparison of the free surface of a solitary wave calculated with the asymptotic heterogeneous DDM at two time steps and different grids.

4 Conclusions

Transmisison conditions have been derived for the LGNE and LSWE by taking a vertical average and truncating an asymptotic expansion of the transmission conditions of a DDM of the discrete linear free-surface Euler equations. This DDM uses the

Neumann-Dirichlet method on the elliptic problem for the velocity potential. In the homogeneous case, when the LGNE are solved on both sides of the interface, we recover the Neumann-Dirichlet method of the elliptic part of the LGNE, plus an $O(\Delta x)$ term. The method has the same convergence property as its parent method but the $O(\Delta x)$ terms make the limit of the subdomain iterations different from the monodomain solution, even though this was imposed on the parent DDM. Also, using more than 2 subdomains could be handled with a relaxation parameter as in [3, section 4.]. In the heterogeneous case the Neumann-Dirichlet method corresponds to a linear interpolation of the elliptic variable between its last value in the LGNE domain and 0, the condition that defines the LSWE, plus an $O(\Delta x^2)$ term. Numerical results show that this induces an unstable scheme, due to oscillations and discontinuities in the interface that grow in amplitude as Δx decreases. The next steps could be the introduction of a free parameter in the boundary conditions to optimize the convergence of the method, for example through a Robin boundary condition, and the analysis of this approach in the continuous case.

References

1. Besse, C., Coulombel, J.-F., and Noble, P. Discrete transparent boundary conditions for the two-dimensional leap-frog scheme: approximation and fast implementation. *ESAIM: Mathematical Modelling and Numerical Analysis* **55**, S535–S571 (2021).
2. Caldas Steinstraesser, J. G., Cienfuegos Carrasco, R. A., Galaz Mora, J. D., and Rousseau, A. A schwarz-based domain decomposition method for the dispersion equation. *Journal of Applied Analysis & Computations* **8**(3), 859 (2018).
3. Funaro, D., Quarteroni, A., and Zanolli, P. An iterative procedure with interface relaxation for domain decomposition methods. *SIAM Journal on Numerical Analysis* **25**(6), 1213–1236 (1988).
4. Gander, M. J., Halpern, L., and Nataf, F. Optimal schwarz waveform relaxation for the one dimensional wave equation. *SIAM Journal on Numerical Analysis* **41**(5), 1643–1681 (2003).
5. Kazakova, M. and Noble, P. Discrete transparent boundary conditions for the linearized green–naghdi system of equations. *SIAM Journal on Numerical Analysis* **58**(1), 657–683 (2020).
6. Kazolea, M. and Ricchiuto, M. On wave breaking for boussinesq-type models. *Ocean Modelling* **123**, 16–39 (2018).
7. Lannes, D. *The water waves problem: mathematical analysis and asymptotics*, vol. 188. American Mathematical Soc. (2013).
8. Wei, G. and Kirby, J. T. Time-dependent numerical code for extended boussinesq equations. *Journal of waterway, port, coastal, and ocean engineering* **121**(5), 251–261 (1995).
9. Zheng, C., Wen, X., and Han, H. Numerical solution to a linearized kdv equation on unbounded domain. *Numerical Methods for Partial Differential Equations: An International Journal* **24**(2), 383–399 (2008).

Piece-wise Constant, Linear and Oscillatory: a Historical Introduction to Spectral Coarse Spaces with Focus on Schwarz Methods

Martin J. Gander and Laurence Halpern

1 Classical coarse spaces

In 1987, Roy Nicolaides introduced what we would now call a coarse space correction for the conjugate gradient method [30]:

> "In this paper, another way of improving the convergence of conjugate gradients is used. It can be used alone or in conjunction with preconditioners. Used alone, it is at least as efficient as the standard preconditioners on model problems. Used with preconditioning it appears from numerical experiments to give a method considerably better than either used separately–it seems that the approaches are in some sense complementary."

The idea of Nicolaides for an example Poisson problem is to deflate piece-wise constant functions on subdomains from the residual at each CG iteration ("we shall systematically interpret E's columns as being a basis for a subspace of certain slowly varying residual components"). From the quote above we see that he advocates to use this technique together with another preconditioner, realizing the two-level character this provides:

> "The method has something in common with a two-level multigrid scheme, although neither smoothing nor subgrids is explicitly used."

There was however no theoretical understanding yet at this point:

> "No theoretical predictions are available at present on the rate of convergence to be expected with preconditioned versions."

Deflation was also introduced independently by Zdeněk Dostál in [7] under the name of 'preconditioning by projector', and the special case of deflating eigenvectors was studied; see also [8] for a relation to Schur complement preconditioning.

Martin J. Gander
FSMP and Section de Mathématiques, Université de Genève, e-mail: martin.gander@unige.ch

Laurence Halpern
LAGA, Université Sorbonne Paris Nord e-mail: halpern@math.univ-paris13.fr

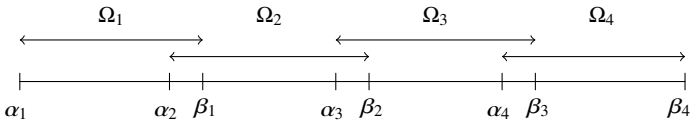


Fig. 1 One-dimensional overlapping domain decomposition.

In order to illustrate the performance of this piece-wise constant coarse space in the context of domain decomposition, we show a numerical experiment for the 1D Laplace problem in $\Omega = (0, 1)$,

$$\partial_{xx} u = 0, \quad u(0) = 0 \text{ and } u(1) = 1,$$

using the parallel Schwarz method introduced by Pierre-Louis Lions [26] at the first international conference on domain decomposition methods (DD1),

$$\begin{aligned} \partial_{xx} u_i^n = 0 \quad \text{in } \Omega_i, \qquad i = 1, .., I, \\ u_i^n(\alpha_i) = u_{i-1}^{n-1}(\alpha_i), \quad u_i^n(\beta_i) = u_{i+1}^{n-1}(\beta_i), \end{aligned} \tag{1}$$

for the decomposition shown in Figure 1 for $I = 4$. When this method is discretized, it is equivalent to Restricted Additive Schwarz (RAS) by Xiao-Chuan Cai and Markus Sarkis [4] for the linear system $A\mathbf{u} = \mathbf{f}$,

$$\mathbf{u}^n := \mathbf{u}^{n-1} + \sum_{i=1}^{I} \tilde{R}^T A_i^{-1} R(\mathbf{f} - A\mathbf{u}^{n-1}), \tag{2}$$

where R_i are restriction matrices of $\mathbf{u}^n$ to the subdomain Ω_i, $A_i := R_i A R_i^T$, and $\tilde{R}_i$ are restriction matrices for a non-overlapping partition; see [15] for more details and the proof of equivalence. In order to combine this with a piece-wise constant coarse correction, we use the $\mathbf{u}^n$ from RAS in (2) and then coarse correct them by computing $\mathbf{u}^n := \mathbf{u}^n + R^T A_c^{-1} R(\mathbf{f} - A\mathbf{u}^n)$, where R is a restriction to the piece-wise constant coarse space functions and $A_c := RAR^T$ is the coarse correction matrix on that space.

We show in Figure 2 the iterates without Krylov acceleration for RAS without and with piece-wise constant coarse correction in the top two rows. We see that the coarse correction indeed changes the iterates, but not by much. To see the true benefit from the coarse correction, we need to use more subdomains. We show the decay of the error for more and more subdomains in Figure 3 (left). We see that indeed with the piece-wise constant coarse space we obtain a scalable method[1], which would be termed "optimal" because of this, but are there better coarse spaces?

Max Dryja and Olof Widlund introduced in the same year as Nicolaides their seminal additive Schwarz method [34] which includes a different coarse space:

> "The first subspace V_0^h, which we also call V^H, is special. It is the space of continuous, piece-wise linear functions on the coarse mesh defined by the substructures Ω_i."

[1] Iteration numbers do not deteriorate when using more and more subdomains.

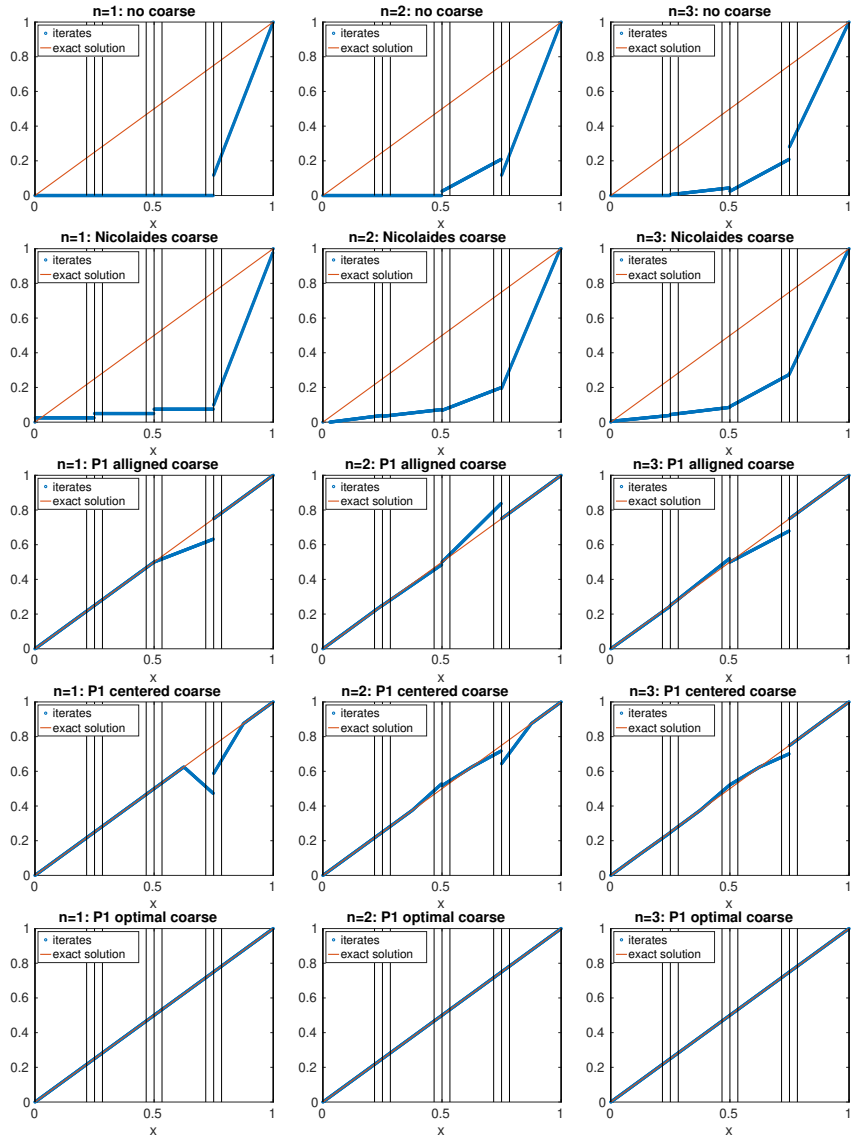


Fig. 2 First three parallel Schwarz iterates without coarse correction (top row), with piece-wise constant coarse correction (second row), with P1 coarse correction aligned with the subdomains (third row), with P1 coarse correction centered in the subdomains (fourth row) and optimal (best possible) coarse correction (last row).

Here the Ω_i correspond to triangles forming a non-overlapping decomposition of the domain, and in contrast to Nicolaides, the coarse functions are linear, not constant, on the subdomains. The results for this coarse space and our model problem are

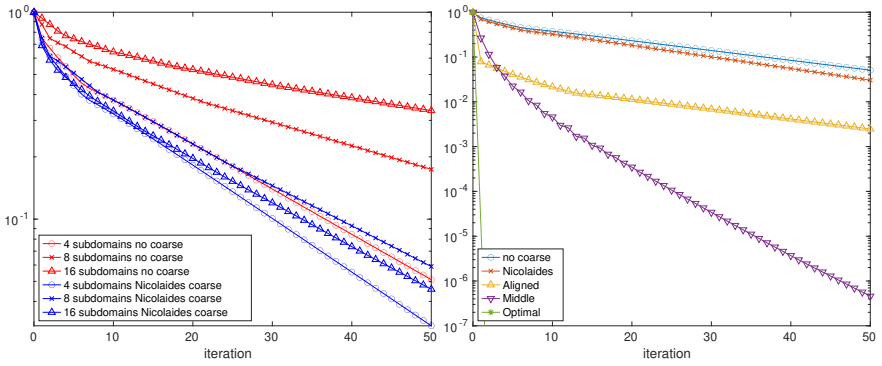


Fig. 3 Error of the parallel Schwarz method with and without piece-wise constant coarse space for increasing number of subdomains (left) and for various coarse spaces and four subdomains (right).

shown in Figure 2 (third row), and we see the coarse space works much better than the Nicolaides coarse space. At the first international conference on domain decomposition methods a year later, Olof Widlund presented an iterative substructuring variant for the piece-wise linear coarse correction on triangles [35], and Max Dryja an extension to three-dimensional problems [9], also in the context of substructuring.

Jan Mandel and Marian Brezina then studied the balancing domain decomposition method in [27]:

> "The Balancing Domain Decomposition (BDD) was introduced by Mandel [1993] by adding a coarse problem to an earlier method of De Roeck and Le Tallec [1991[2]], known as the Neumann-Neumann method . . . "
>
> ". . . a global coarse problem with one or few unknowns for each subdomain . . . "
>
> "The presence of the coarse problem now guarantees that the possibly singular local problems are consistent."

They transformed the bug of the classical Neumann-Neumann method to have floating subdomains with all Neumann conditions around that made the method not well posed into a feature: they determine the constant (in the Laplace case) by a coarse problem, which leads to a piece-wise constant coarse space aligned with the subdomains. The FETI method invented by Charbel Farhat and François-Xavier Roux in [12] also contains naturally the piece-wise constant modes in the projection step as a coarse space; for a theoretical analysis, see [11, 28]. Note that all these coarse spaces were developed independently of the work by Nicolaides.

Max Dryja, Barry Smith and Olof Widlund emphasize in [10] the great importance and challenge of good coarse space constructions:

> " The design, analysis, and implementation of the coarse space problem pose the most challenging technical problems in work of this kind."

They consider several richer coarse spaces than just a constant per substructure and compare them for primal Schur complement substructuring methods. A first

[2] Also at the first international conference on domain decomposition methods!

variant is using piece-wise linear coarse basis functions aligned with triangular substructures, and then additional piece-wise constant edge and face coarse functions are considered, harmonically extended into the subdomains, keeping the vertex functions. For all variants, detailed condition number estimates are provided, and compared to the earlier piece-wise constant coarse space.

We see that all these early coarse spaces were aligned with subdomain boundaries of the domain decomposition method. A generalization of the analysis that permits coarse spaces not aligned with the subdomains, also using ideas from non-overlapping methods, can be found in the book by Andrea Toselli and Olof Widlund [33]:

> " We introduce a shape-regular coarse mesh $\mathcal{T}_H$ on the domain Ω and the finite element space [...] of continuous, piece-wise linear functions on $\mathcal{T}_H$ [...] We stress that the fine mesh $\mathcal{T}$ need not be a refinement of $\mathcal{T}_H$."

Such general coarse spaces were studied at the continuous level in [18] with accurate estimates of the constants involved in the resulting condition number estimate. We show the performance of such a P1 non-aligned coarse space in Figure 2 (fourth row) with coarse points in the middle of the subdomains for our model problem. A comparison of the convergence as a function of the iterations is shown in Figure 3 on the right, where we see that the general position of the coarse points in the middle of the subdomains performs best so far. But is there an even better option?

2 Optimal coarse spaces and spectral approximations

It was first observed in [16] and then analyzed in more detail in [17, 19] that the position of the coarse nodes has indeed an important impact on the performance of the coarse space. For a large scale implementation of various coarse node positionings for Schwarz methods, see [23]. We show in Figure 2 in the last row the performance of a coarse space whose nodes are located to the left and right of the RAS non-overlapping interface. We see that this P1 coarse space transforms the two-level method into a direct solver, the solution is obtained within the subdomains after the coarse correction. This is also visible in the convergence curves in Figure 3 (right). Such coarse spaces are called optimal in the sense of better is not possible, not in the sense of scalable, and the idea is related to the algebraic multigrid construction in [3, 32].

New coarse spaces in domain decomposition methods are approximations of this optimal coarse space; see the Spectral Harmonically Enriched Multiscale (SHEM) coarse space [21, 20] for such a construction in a multiscale context. In higher spatial dimensions, this optimal coarse space simply needs to contain all discrete harmonic functions (functions that solve the homogeneous equation) in each subdomain, and is thus of the size of the number of interface variables of the subdomains. A first approximation for a decomposition into square subdomains is to add the historically successful Q1 functions aligned with each subdomain; see e.g. Figure 4 (left) for one of them. One can then enrich this coarse space by adding harmonically extended

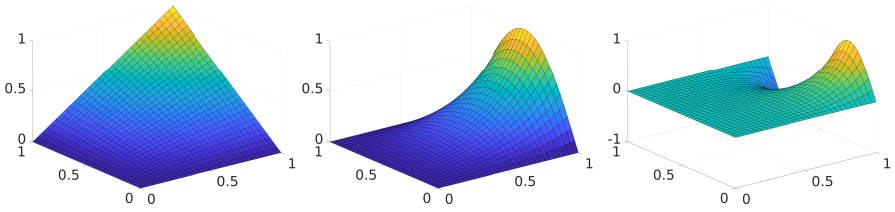
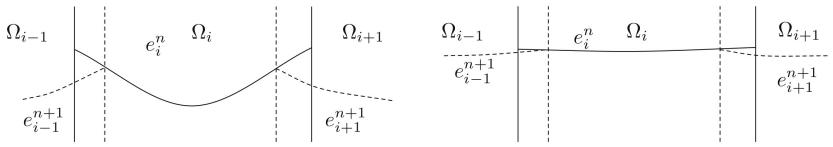


Fig. 4 First Q1 coarse space functions and two spectral enrichments.

sine functions; see Figure 4 (middle and right) to get a spectral coarse space. This construction is not restricted to square subdomains; see [21, 20, 5].

A seemingly different construction of a new coarse space was proposed by Frédéric Nataf, Hua Xiang, Victorita Dolean and Nicole Spillane in [29] for high contrast problems:

> "An effective two-level preconditioner is highly dependent on the choice of the coarse-grid subspace. We will now focus on the choice of the coarse space Z in the context of DDMs for problems of type (1.1) with heterogeneous coefficients."



> "Moreover, a fast decay for this value corresponds to a large eigenvalue of the DtN map, whereas a slow decay corresponds to small eigenvalues of this map because the DtN operator is related to the normal derivative at the interface and the overlap is thin."

From the drawing in their manuscript above, eigenmodes of the Dirichlet-to-Neumann (DtN) map with large eigenvalues will converge fast (left), while eigenmodes with small eigenvalues will converge slowly (right). Hence the idea is to use eigenmodes of the DtN map with small eigenvalues on each subdomain as coarse space. We show in Figure 5 the first four DtN modes for a square subdomain, and also mode 5 and 9. The first four modes look like they span the same space as the four Q1 coarse modes from before. Mode 5 contains a first sine component on the boundary like the enrichment mode in Figure 4 (middle); modes 6-8 (not shown) are similar. Mode 9 contains the second sine mode on the boundary, like the enrichment mode in Figure 4 (right), and modes 10-12 (not shown) are again similar. So the DtN coarse space seems to be related to the SHEM coarse space. This relation becomes even more evident if one uses eigenmodes of the DtN operator computed for each of the four boundaries of the square subdomain separately, since then they coincide with the modes shown in Figure 4 (middle, right)!

A highly successful coarse space, also for high contrast problems, was introduced by Nicole Spillane, Victorita Dolean, Patrice Hauret, Frédéric Nataf,

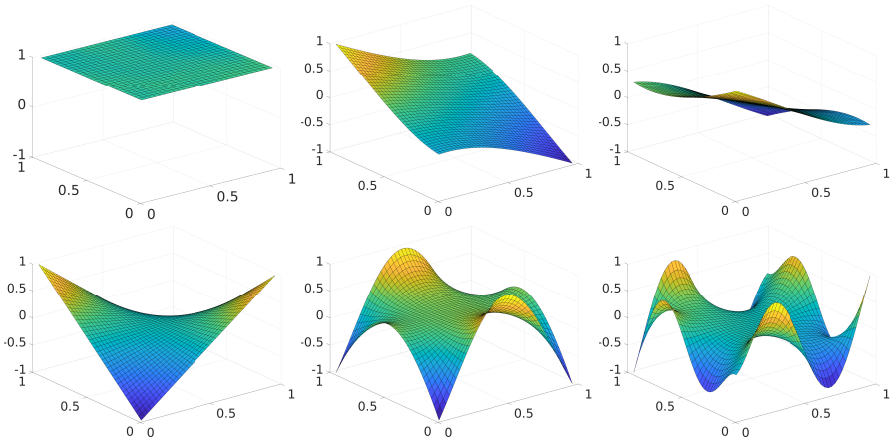


Fig. 5 DtN modes 1-4, 5, and 9.

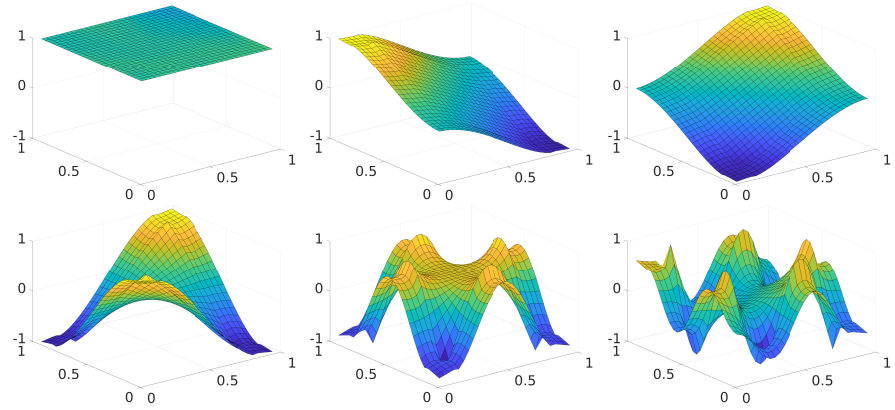


Fig. 6 GenEO modes 1-4, 5, and 9.

Clemens Pechstein and Robert Scheichl in [31], namely GenEO (Generalized Eigenvalue Problems in the Overlaps). The powerful idea of GenEO is to directly improve the Additive Schwarz convergence estimate by adding the corresponding slow modes from the estimate to the coarse space. The modes are also computed in each overlapping subdomain, following [6], by solving the eigenvalue problem

$$B_i \mathbf{u} = \lambda D A_i D \mathbf{u}, \tag{3}$$

where B_i is the Neumann subdomain matrix, A_i is the Dirichlet subdomain matrix, and D is a diagonal weighting matrix representing a partition of unity. We show the first 4 modes, and then also mode 5 and 9 in Figure 6. We see that they are

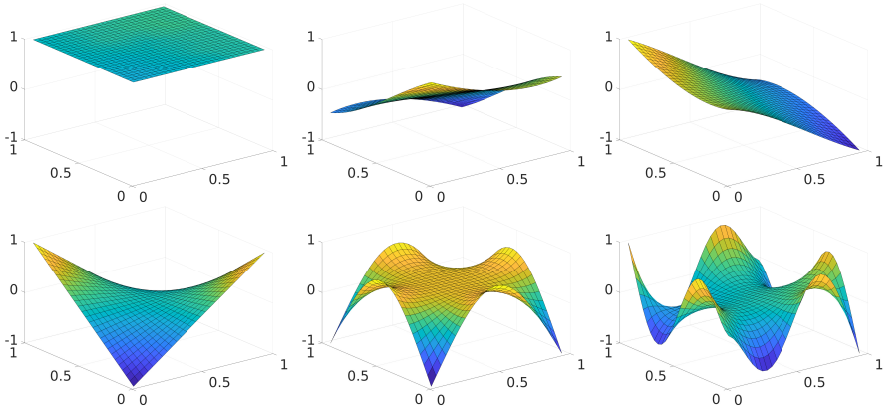


Fig. 7 GenEO modes 1-4, 5, and 9 without the partition of unity.

very similar to the DtN eigenmodes (mode 3, 4 and 9 just need to be multiplied by -1). If we remove the partition of unity in the eigenvalue problem (3), we get the modes shown in Figure 7. These are now very close to the DtN modes (up to multiplications by -1) in Figure 5, and we are working to prove that they in fact span the same coarse space. A comparison of the numerical performance of these coarse spaces can be found in [22]; this comparison was made before these relations were known. In [22], there is also a comparison with the coarse spaces introduced by Juan Galvis and Yalchin Efendiev in [13, 14], which are based on subdomain eigenfunctions in volume and thus not harmonic in the subdomains. Note that such volume eigenvalue coarse spaces have been already introduced for non-overlapping domain decomposition methods by Petter Bjørstad and Piotr Krzyżanowski almost a decade earlier [2], and this in an adaptive fashion (see also [1]):

> "It appears that this paper is the first to propose an adaptive algorithm that can construct an effective coarse space for problems of this kind".

Techniques from multiscale finite element methods were also used to construct coarse spaces for Schwarz methods: the ACMS (Approximate Component Mode Synthesis) coarse space by Alexander Heinlein, Axel Klawonn, Jascha Knepper and Oliver Rheinbach in [25] is using Schur complement eigenvalue problems on subdomain edges in order to construct coarse basis functions. This approach is in the simple Laplace case related to the SHEM enrichment functions shown in Figure 4 in the middle and on the right. The early coarse space from [10] for non-overlapping domain decomposition methods based on piece-wise constant edge (and face) functions became also the basis for a spectrally enriched coarse space under the name adaptive GDSW (Generalized Dryja Smith Widlund) coarse space, see [24], where the authors use for the enrichment Dirichlet to Neumann eigenfunctions at the interfaces, extended harmonically into the subdomains.

3 Conclusions

We gave a short historical and personal introduction to the fascinating research area of coarse space construction for domain decomposition methods. This is currently a very active field of research, and a complete understanding of best coarse spaces in terms of performance even for Laplace problems is only emerging. Corresponding intrinsic coarse space components for Schwarz methods can be found in [5], and their analysis is currently our focus.

References

1. Bjørstad, P. E., Koster, J., and Krzyżanowski, P. Domain decomposition solvers for large scale industrial finite element problems. In: *Applied Parallel Computing. New Paradigms for HPC in Industry and Academia: 5th International Workshop, PARA 2000 Bergen, Norway, June 18–20, 2000 Proceedings 5*, 373–383. Springer (2001).
2. Bjørstad, P. E. and Krzyżanowski, P. A flexible 2-level Neumann-Neumann method for structural analysis problems. In: *International Conference on Parallel Processing and Applied Mathematics*, 387–394. Springer (2001).
3. Brandt, A., McCormick, S., and Ruge, J. Algebraic multigrid (AMG) for automatic algorithm design and problem solution. Tech. rep., Report,. Comp. Studies, Colorado State University, Ft. Collins (1982).
4. Cai, X.-C. and Sarkis, M. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing* **21**(2), 792–797 (1999).
5. Cuvelier, F., Gander, M. J., and Halpern, L. Fundamental coarse space components for Schwarz methods with crosspoints. In: *International Conference on Domain Decomposition Methods XXVI*, 39–50. Springer (2023).
6. Dolean, V., Jolivet, P., and Nataf, F. *An introduction to domain decomposition methods: algorithms, theory, and parallel implementation*. SIAM (2015).
7. Dostál, Z. Conjugate gradient method with preconditioning by projector. *International Journal of Computer Mathematics* **23**(3-4), 315–323 (1988).
8. Dostál, Z. Projector preconditioning and domain decomposition methods. *Applied Mathematics and Computation* **37**(2), 75–81 (1990).
9. Dryja, M. A method of domain decomposition for three-dimensional finite element elliptic problems. In: *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, 43–61. SIAM Philadelphia (1988).
10. Dryja, M., Smith, B. F., and Widlund, O. B. Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. *SIAM journal on numerical analysis* **31**(6), 1662–1694 (1994).
11. Farhat, C., Mandel, J., and Roux, F.-X. Optimal convergence properties of the FETI domain decomposition method. *Computer methods in applied mechanics and engineering* **115**(3-4), 365–385 (1994).
12. Farhat, C. and Roux, F.-X. A method of Finite Element Tearing and Interconnecting and its parallel solution algorithm. *Int. J. Numer. Meth. Engrg.* **32**, 1205–1227 (1991).
13. Galvis, J. and Efendiev, Y. Domain decomposition preconditioners for multiscale flows in high-contrast media. *Multiscale Modeling & Simulation* **8**(4), 1461–1483 (2010).
14. Galvis, J. and Efendiev, Y. Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces. *Multiscale Modeling & Simulation* **8**(5), 1621–1644 (2010).
15. Gander, M. J. Schwarz methods over the course of time. *Electronic transactions on numerical analysis* **31**, 228–255 (2008).

16. Gander, M. J. and Halpern, L. Méthodes de décomposition de domaine. In: *Encyclopédie électronique pour les ingénieurs*. Techniques de l'ingénieur (2012).
17. Gander, M. J., Halpern, L., and Repiquet, K. S. A new coarse grid correction for RAS/AS. In: *Domain Decomposition Methods in Science and Engineering XXI*, 275–283. Springer (2014).
18. Gander, M. J., Halpern, L., and Santugini-Repiquet, K. Continuous analysis of the additive Schwarz method: a stable decomposition in H1. *ESAIM Mathematical Modelling and Numerical Analysis* **49**(3), 365–385 (2011).
19. Gander, M. J., Halpern, L., and Santugini-Repiquet, K. On optimal coarse spaces for domain decomposition and their approximation. In: *International Conference on Domain Decomposition Methods XXIV*, 271–280. Springer (2018).
20. Gander, M. J. and Loneland, A. SHEM: An optimal coarse space for RAS and its multiscale approximation. In: *Domain Decomposition Methods in Science and Engineering XXIII*, 313–321. Springer (2017).
21. Gander, M. J., Loneland, A., and Rahman, T. Analysis of a new harmonically enriched multiscale coarse space for domain decomposition methods. *arXiv preprint arXiv:1512.05285* (2015).
22. Gander, M. J. and Song, B. Complete, optimal and optimized coarse spaces for additive Schwarz. In: *International Conference on Domain Decomposition Methods XXIV*, 301–309. Springer (2018).
23. Gander, M. J. and Van Criekingen, S. New coarse corrections for Optimized Restricted Additive Schwarz using PETSc. In: *International Conference on Domain Decomposition Methods XXV*, 483–490. Springer (2020).
24. Heinlein, A., Klawonn, A., Knepper, J., and Rheinbach, O. An adaptive GDSW coarse space for two-level overlapping Schwarz methods in two dimensions. In: *Domain Decomposition Methods in Science and Engineering XXIV*, 373–382. Springer (2018).
25. Heinlein, A., Klawonn, A., Knepper, J., and Rheinbach, O. Multiscale coarse spaces for overlapping Schwarz methods based on the ACMS space in 2D. *ETNA* **48**, 156–182 (2018).
26. Lions, P.-L. On the Schwarz alternating method. I. In: *First international symposium on domain decomposition methods for partial differential equations*, vol. 1, 42. Paris, France (1988).
27. Mandel, J. and Brezina, M. Balancing domain decomposition: Theory and performance in two and three dimensions. Tech. rep., University of Colorado at Denver (1993).
28. Mandel, J. and Tezaur, R. Convergence of a substructuring method with Lagrange multipliers. *Numerische Mathematik* **73**(4), 473–487 (1996).
29. Nataf, F., Xiang, H., Dolean, V., and Spillane, N. A coarse space construction based on local Dirichlet-to-Neumann maps. *SIAM Journal on Scientific Computing* **33**(4), 1623–1642 (2011).
30. Nicolaides, R. A. Deflation of conjugate gradients with applications to boundary value problems. *SIAM Journal on Numerical Analysis* **24**(2), 355–365 (1987).
31. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., and Scheichl, R. Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. *Numerische Mathematik* **126**(4), 741–770 (2014).
32. Stüben, K. Algebraic multigrid (AMG): experiences and comparisons. *Applied Mathematics and Computation* **13**(3-4), 419–451 (1983).
33. Toselli, A. and Widlund, O. *Domain Decomposition Methods - Algorithms and Theory*, vol. 34. Springer Science & Business Media (2004).
34. Widlund, O. and Dryja, M. An additive variant of the Schwarz alternating method for the case of many subregions. Tech. rep., Department of Computer Science, Courant Institute (1987).
35. Widlund, O. B. Iterative substructuring methods: Algorithms and theory for elliptic problems in the plane. In: *First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Philadelphia, PA*, 113–128. SIAM Philadelphia (1988).

A New Nodal Integration Method for Helmholtz Problems Based on Domain Decomposition Techniques

Martin J. Gander and Niteen Kumar

1 Introduction

Wave field simulations have many applications, from seismology over radiation to acoustics. The Helmholtz equation is used to model many of these phenomena, and several numerical schemes were developed for this, see e.g. [5, 8, 7] and references therein. However, to capture the accurate wave behavior, in general these schemes need very fine meshes, because of the so called pollution effect, see [1]. The fine mesh requirement results in large system matrices with bad condition number, and thus requires a huge computational effort, since Helmholtz problems are notoriously difficult to solve using iterative methods [4]. Also, due to the high condition number, often these schemes have numerical problems for large wave numbers.

We present in this short note a new Nodal Integration Method (NIM) based on domain decomposition techniques for the Helmholtz equation

$$\nabla^2 u(\mathbf{x}) + k^2 u(\mathbf{x}) = f(\mathbf{x}), \tag{1}$$

where $\mathbf{x}$ is the spatial position, k is the wave number, u represents the wave field, typically a pressure perturbation, and f is the source term. NIM is a coarse mesh numerical scheme based on the transverse integration process (TIP) and analytical solutions of the ODEs resulting from TIP [10]. NIM has an edge over other schemes due to the inbuilt semi-analytical approach in the scheme development process, which closely relates the scheme to the physical problem compared to predefined basis-function based methods such as finite-element methods. NIM schemes are related to Trefftz methods [12] going back to Erich Trefftz in 1926 as a counterpart of the classical Ritz method [11] from 1909. Trefftz methods use basis functions that satisfy the homogeneous equations exactly within elements, see also [8] and references therein, whereas NIMs satisfy only one dimensional averaged equations.

Martin J. Gander, Niteen Kumar
Section de Mathématiques, Université de Genève, e-mail: martin.gander@unige.ch, niteen.kumar@unige.ch

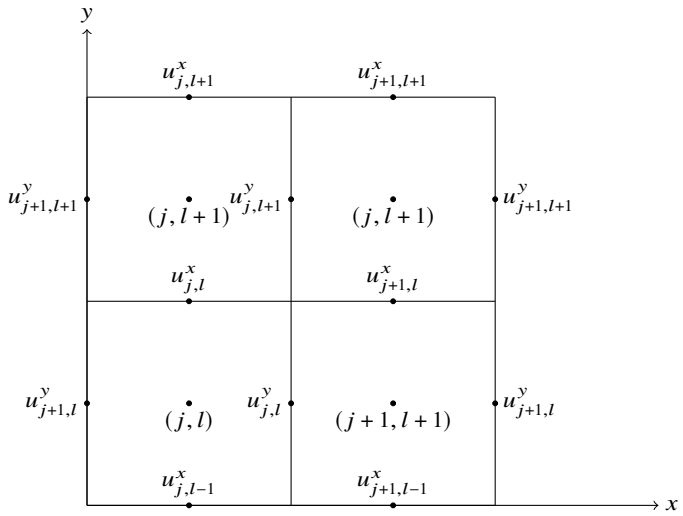


Fig. 1 Arrangement of elements in 2D called nodes in NIM.

The first NIM scheme was developed for simulations in nuclear industry [6], and NIM found its acceptance in other engineering domains as well, due to high accuracy with coarser meshes, see e.g. [9] and references therein. The discretization of PDEs is also often plagued with numerical dispersion, and NIM schemes show minimal dispersion compared to other schemes, see [10], and [2] and references therein for more information about dispersion correction. We propose here a new NIM scheme for the Helmholtz equation to improve the conditioning of the resulting system matrix, and further reduce dispersion. Our new approach uses impedance (or Robin) conditions in its construction, in contrast to the classical Dirichlet and Neumann conditions in earlier NIMs for Helmholtz problems.

2 Classical NIM for the Helmholtz problem

In order to derive the classical NIM scheme for the Helmholtz equation (1) in 2D, the domain is divided into n rectangular elements of size h called nodes, see Figure 1. For each node, a local coordinate system is defined with its origin at the node center. The Helmholtz Equation (1) can be written with reference to node (j, l) as

$$\nabla^2 u_{j,l}(x, y) + k^2 u_{j,l}(x, y) = f_{j,l}(x, y), \quad (x, y) \in \Big(-\frac{h}{2}, \frac{h}{2}) \Big) \times \Big(-\frac{h}{2}, \frac{h}{2} \Big). \quad (2)$$

In NIM, the PDE is first averaged within a node to remove the dependency in one spatial directions, which results in an approximate ODE. This is called the transverse integration process (TIP). To perform the TIP, Equation (2) is averaged using the

operator $\frac{1}{h}\int_{-h/2}^{+h/2} \mathrm{d}x$ in x-direction and the operator $\frac{1}{h}\int_{-h/2}^{+h/2} \mathrm{d}y$ in y-direction. On performing the TIP (averaging) for example in the x-direction,

$$\frac{1}{h}\int_{-h/2}^{+h/2}\left(\frac{\mathrm{d}^2 u_{j,l}(x,y)}{\mathrm{d}x^2}+\frac{\mathrm{d}^2 u_{j,l}(x,y)}{\mathrm{d}y^2}+k^2 u_{j,l}(x,y)=f_{j,l}(x,y)\right)\mathrm{d}x, \tag{3}$$

we get x-averaged ODEs whose solutions are a function of y only as given in equation (4) below. Similarly, performing TIP on equation (2) in the y-direction gives us y-averaged ODEs whose solutions are a function of x only,

$$\frac{\mathrm{d}^2 \overline{u}^x_{j,l}(y)}{\mathrm{d}y^2}+k^2\overline{u}^x_{j,l}(y)=\overline{S}^x_{j,l}(y), \quad \frac{\mathrm{d}^2 \overline{u}^y_{j,l}(x)}{\mathrm{d}x^2}+k^2\overline{u}^y_{j,l}(x)=\overline{S}^y_{j,l}(x). \tag{4}$$

Here the solution variables represent averaged quantities,

$$\overline{u}^x_{j,l}(y):=\frac{1}{h}\int_{-h/2}^{+h/2} u_{j,l}(x,y)\mathrm{d}x, \quad \overline{u}^y_{j,l}(x):=\frac{1}{h}\int_{-h/2}^{+h/2} u_{j,l}(x,y)\mathrm{d}y, \tag{5}$$

and also the source term $f_{j,l}$ was averaged including the remaining transverse term,

$$\overline{S}^x_{j,l}(y):=\frac{1}{h}\int_{-h/2}^{+h/2}\left(f_{j,l}(x,y)-\frac{\partial^2 u_{j,l}(x,y)}{\partial x^2}\right)\mathrm{d}x, \tag{6}$$

$$\overline{S}^y_{j,l}(x):=\frac{1}{h}\int_{-h/2}^{+h/2}\left(f_{j,l}(x,y)-\frac{\partial^2 u_{j,l}(x,y)}{\partial y^2}\right)\mathrm{d}y. \tag{7}$$

After the TIP, the set of approximate ODEs given in Equation (4) is solved analytically within two consecutive nodes, using an appropriate approximation of the source term to make this analytical integration possible (for example a truncated Legendre expansion). After the integration, the two analytical solutions are connected using coupling conditions, classically Dirichlet continuity is imposed by imposing a common (unknown) value, which is then determined imposing Neumann continuity, like in a substructuring domain decomposition method. This results in two three point schemes, one in the x-direction and the other in the y-direction. From these three point schemes, the pseudo source is finally eliminated using constraint conditions, which results in the final set of algebraic equation for the scheme, see [6, 9, 10] for more details, and below for a simple example.

While this NIM scheme for Helmholtz is working, the resulting matrix elements can have a strong dependence on the wave number k. We show in Table 1 an example of the dependence of the system matrix norm on the wave number k of the 2D NIM scheme described above. This strong dependence is numerically not desirable, especially when the mesh resolution is not changed as in our example, there is too much sensitivity with respect to the wave number in this discrete problem.

In order to better understand this strong dependence on the wave number k of the classical NIM system matrix for the Helmholtz equation, we now study in more detail the one dimensional case,

Table 1 Dependence of the system matrix norm on the wave number k for the classical NIM scheme in 2D for the Helmholtz equation.

Wave number (k)	NIM matrix norm (2D-Helmholtz)
150	14800
151	32170
152	214350
153	25180
154	13500

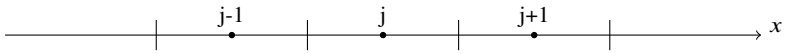


Fig. 2 Arrangement of elements in 1D.

$$\partial_{xx} u_j(x) + k^2 u_j(x) = f_j(x), \quad x \in \Big(-\frac{h}{2}, \frac{h}{2}\Big), \tag{8}$$

see also Figure 2. In one dimension, the TIP is not necessary, except for the right hand side function $f_j(x)$. Here we expand $f_j(x)$ in Legendre polynomials and truncate to the first term, i.e. the constant, which we call S_j. This approximation to a constant term leads to second order accuracy in the scheme. We can then directly solve Equation (8) analytically with $f_j(x)$ replaced by S_j on each node, and using Dirichlet boundary conditions, which are

$$\left.\begin{aligned} \overline{u}^a_j(x)\Big|_{-h/2} &= u_{j-1} \\ \overline{u}^a_j(x)\Big|_{h/2} &= u_j \end{aligned}\right\} \text{for node } j, \qquad \left.\begin{aligned} \overline{u}^a_{j+1}(x)\Big|_{-h/2} &= u_j \\ \overline{u}^a_{j+1}(x)\Big|_{h/2} &= u_{j+1} \end{aligned}\right\} \text{for node } j+1. \tag{9}$$

The analytical solution for node j and $j+1$ is then given by

$$\begin{aligned} \overline{u}^a_j(x) &= \tfrac{2S_j + (-2S_j + k^2(u_j + u_{j-1}))\cos kx \sec\frac{hk}{2} + k^2(u_j - u_{j-1})\csc\frac{hk}{2}\sin kx}{2k^2}, \\ \overline{u}^a_{j+1}(x) &= \tfrac{2S_{j+1} + (-2S_{j+1} + k^2(u_j + u_{j+1}))\cos kx \sec\frac{hk}{2} + k^2(u_{j+1} - u_j)\csc\frac{hk}{2}\sin kx}{2k^2}. \end{aligned} \tag{10}$$

Now in order to connect consecutive nodes, the matching of Neumann traces is imposed, i.e.

$$\left(\frac{d\overline{u}^a_j(x)}{dx}\right)\Big|_{h/2} = \left(\frac{d\overline{u}^a_{j+1}(x)}{dx}\right)\Big|_{-h/2}. \tag{11}$$

This leads to a finite difference like stencil for the unknown Dirichlet values u_j, which contains in its coefficients information about the physical problem that is solved, namely

$$\frac{k}{\sin kh} u_{j+1} - \frac{2k}{\tan kh} u_j + \frac{k}{\sin kh} u_{j-1} = \frac{\tan\frac{hK}{2}}{k}(S_j + S_{j+1}). \tag{12}$$

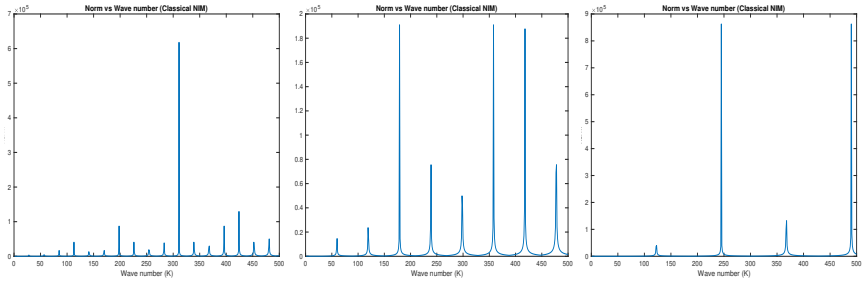


Fig. 3 Norms of the system matrix of the classical Helmholtz NIM from the stencils (12), (14), (15) for varying wave number k and three mesh sizes: 0.1 (left), 0.05 (middle) and 0.025 (right).

To complete the linear system, we have to use on the first node, $j = 1$, and the last node, $j = J$, the original boundary conditions imposed on the problem, which we assume to be of impedance type,

$$\left(-\frac{d\overline{u}_1^a(x)}{dx} + ik\overline{u}_1^a(x)\right)\Bigg|_{-h/2} = 0, \quad \left(\frac{d\overline{u}_J^a(x)}{dx} + ik\overline{u}_J^a(x)\right)\Bigg|_{h/2} = 0. \tag{13}$$

This leads for the first and last NIM matrix equations to the stencils

$$\left(\frac{ik + k^2 \cot hk}{k}\right) u_1 - (k \csc hk) u_2 = \left(\frac{\cot hk - \csc hk}{k}\right) S_2, \tag{14}$$

$$\left(\frac{ik + k^2 \cot hk}{k}\right) u_J - (k \csc hk) u_{J-1} = \left(\frac{\cot hk - \csc hk}{k}\right) S_J. \tag{15}$$

Collecting these stencils in the associated system matrix of the Helmholtz NIM in 1D, and computing its norm, we find the results shown in Figure 3. Clearly the norm is extremely sensitive to the wave number k, and this does not improve when the mesh is refined. We can now also see the reason for this looking at the stencil entries: in the interior stencil in (12), the stencil coefficients contain a division by $\sin kh$, and this quantity becomes zero for $k = \ell\pi/h$, $\ell = 1, 2, \ldots$, which explains the poles in Figure 3 and more generally the sensitivity of the classical Helmholtz NIM matrix norm on the wave number. We can now also explain the reason for this sensitivity: in the construction of the classical Helmholtz NIM, we solved 1D Helmholtz problems on each node, imposing Dirichlet boundary conditions, and if k^2 corresponds to an eigenvalue of the one dimensional Laplacian, then this problem is not well posed, a fact that manifests itself in the division by zero in the stencil coefficients.

3 Derivation of the new NIM scheme

To address the issue of division by zero for some values of k, we must design a new Helmholtz NIM that avoids in its construction the solution of Helmholtz problems with Dirichlet conditions that can become ill-posed. This can be achieved by using impedance conditions instead, like it was proposed in the seminal work of Després and his non-overlapping Schwarz method for Helmholtz problems [3]. We thus replace in the construction of our new Helmholtz NIM the conditions (9) for nodes j and $j+1$ by the conditions

$$\left.\begin{aligned} &\left(-\frac{\partial \overline{u}_j^a(x)}{\partial x} + ik\overline{u}_j^a(x)\right)\Big|_{-h/2} = \sigma_{j-1} \\ &\left(\frac{\partial \overline{u}_j^a(x)}{\partial x} + ik\overline{u}_j^a(x)\right)\Big|_{h/2} = \lambda_j \end{aligned}\right\} \text{for node } j, \tag{16}$$

$$\left.\begin{aligned} &\left(-\frac{\partial \overline{u}_{j+1}^a(x)}{\partial x} + ik\overline{u}_{j+1}^a(x)\right)\Big|_{-h/2} = \sigma_j \\ &\left(\frac{\partial \overline{u}_{j+1}^a(x)}{\partial x} + ik\overline{u}_{j+1}^a(x)\right)\Big|_{h/2} = \lambda_{j+1} \end{aligned}\right\} \text{for node } j+1. \tag{17}$$

Instead of the unknown Dirichlet values u_j in the original Helmholtz NIM, now the unknowns are the impedance traces λ_j and σ_j, which means that we construct directly a right preconditioned system in this new Helmholtz NIM design. The analytical solution of the Helmholtz equation (8) with constant source term S_j and node impedance boundary conditions (16) on node j is

$$\overline{u}_j^a(x) = \frac{2S_j + e^{\frac{-ik(h+2x)}{2}}(-S_j - e^{2ikx}(S_j + ik\lambda_j) - ik\sigma_j}{2k^2}, \tag{18}$$

and similarly we find on node $j+1$

$$\overline{u}_{j+1}^a(x) = \frac{2S_{j+1} + e^{\frac{-ik(h+2x)}{2}}(-S_{j+1} - e^{2ikx}(S_{j+1} + ik\lambda_{j+1}) - ik\sigma_{j+1}}{2k^2}. \tag{19}$$

In order to obtain the new Helmholtz NIM scheme, we use impedance condition matching at the interface,

$$\sigma_{j+1} = \left(-\frac{d\overline{u}_j^a(x)}{dx} + ik\overline{u}_j^a(x)\right)\Big|_{h/2}, \quad \lambda_j = \left(\frac{d\overline{u}_j^a(x)}{dx} + ik\overline{u}_j^a(x)\right)\Big|_{-h/2}. \tag{20}$$

This leads to the new finite difference type stencil

$$\sigma_{j+1} - e^{-ihk}\sigma_j = \left(-\frac{i}{k} + \frac{ie^{-ikh}}{k}\right)S_j, \quad \lambda_j - e^{-ihk}\lambda_{j+1} = \left(-\frac{i}{k} + \frac{ie^{-ikh}}{k}\right)S_{j+1}. \tag{21}$$

For the first and last equation in the system, we need to use again the original boundary conditions in (13), which leads for $j = 1$ to

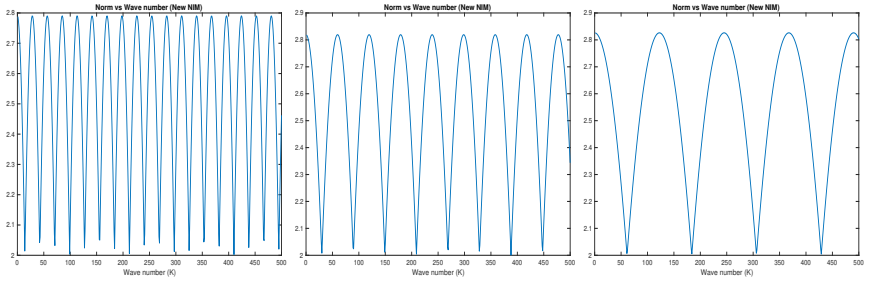


Fig. 4 Norms of the system matrix of the new Helmholtz NIM from the stencils (21), (22), (23), for varying wave number k and three mesh sizes: 0.1 (left), 0.05 (middle) and 0.025 (right).

$$\left(\frac{e^{ihk}(k-1)}{2k}\right)\lambda_1 + \left(\frac{k+1}{2k}\right)\sigma_1 = \left(\frac{-i(k-1)+ie^{ihk}(k-1)}{2k^2}\right)S_1. \tag{22}$$

Similarly the equation on the right boundary, $j = J$, is

$$-\left(\frac{e^{-ihk}(k-1)}{2k}\right)\sigma_J + \left(\frac{k+1}{2k}\right)\lambda_J = \left(\frac{-i(k-1)+ie^{-ihk}(k-1)}{2k^2}\right)S_J. \tag{23}$$

Now we can see from the stencil coefficients in Equation (21) of the new Helmholtz NIM that there is no singularity present any more, and thus the system matrix norms should not have this sensitive dependence on the wave number k any longer. This is confirmed in Figure 4, where we plot the system matrix norm of our new Helmholtz NIM for three different mesh sizes as a function of the wave number k. We see that the norm stays nicely bounded below 3, whereas for the classical NIM the matrix norms we observed were of the order of $1e5$.

4 Conclusions

We presented a new nodal integration method (NIM) based on domain decomposition techniques for the Helmholtz equation. In our new Helmholtz NIM, instead of Dirichlet and Neumann transmission conditions that are usually used in the construction of the NIM, we used impedance (or Robin) transmission conditions. This modification changes the coefficients as well as the resulting system matrix structure, and we observe that the new system matrix has nicely bounded norms for all wave numbers, while the original NIM system matrix norm presented singularities. However, the new system matrix is now twice the size of the old system matrix, since we are solving for the Robin traces as unknowns. We gain stability at the cost of a bigger system matrix. We are currently developing our new Helmholtz NIM in two and three spatial dimensions, and also investigate if it is possible to use impedance conditions without

increasing the system matrix size. We are also studying the dispersion relation properties of our new Helmholtz NIM, and investigate its potential for dispersion correction.

References

1. Babuska, I. M. and Sauter, S. A. Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? *SIAM Journal on numerical analysis* **34**(6), 2392–2423 (1997).
2. Cocquet, P.-H., Gander, M. J., and Xiang, X. Closed form dispersion corrections including a real shifted wavenumber for finite difference discretizations of 2d constant coefficient Helmholtz problems. *SIAM Journal on Scientific Computing* **43**(1), A278–A308 (2021).
3. Després, B. Décomposition de domaine et problème de Helmholtz. *C.R. Acad. Sci. Paris* **1**(6), 313–316 (1990).
4. Ernst, O. G. and Gander, M. J. Why it is difficult to solve Helmholtz problems with classical iterative methods. *Numerical analysis of multiscale problems* 325–363 (2012).
5. Feng, W. Discontinuous Galerkin methods for the Helmholtz equation with large wave number. *SIAM Journal of Numerical analysis* **47**(4), 1–10 (2009).
6. Ferrer, R. M. and Azmy, Y. Y. Error analysis of the nodal integral method for solving the neutron diffusion equation in two-dimensional cartesian geometry. *Nuclear Science and Engineering* **162**(3), 215–233 (2009).
7. Griesmaier, R. and Monk, P. Error analysis for a hybridizable discontinuous Galerkin method for the Helmholtz equation. *Journal of Scientific Computing* **40**(1), 291–310 (2011).
8. Hiptmair, R., Moiola, A., and Perugia, I. Trefftz discontinuous Galerkin methods for acoustic scattering on locally refined meshes. *Applied Numerical Mathematics* **79**(1), 79–91 (2014).
9. Kumar, N., Majumdar, R., and Singh, S. Predictor-Corrector Nodal Integral Method for simulation of high Reynolds number fluid flow using larger time steps in Burgers' equation. *Computers and Mathematics with Applications* **79**(5), 1362–1381 (2020).
10. Kumar, N., Shekar, B., and Singh, S. A nodal integral scheme for acoustic wavefield simulation. *Pure and applied geophysics* **179**(1), 3677–3691 (2022).
11. Ritz, W. Über eine neue Methode zur Lösung gewisser Variationsprobleme der mathematischen Physik. *Journal für die reine und angewandte Mathematik* **135**, 1–61 (1909).
12. Trefftz, E. Ein Gegenstück zum Ritzschen Verfahren. In: *Proc. 2nd Int. Cong. Appl. Mech. Zurich*, 131–137 (1926).

Dirichlet-Neumann and Neumann-Neumann Methods for Elliptic Control Problems

Martin J. Gander and Liu-Di Lu

1 Introduction

Consider the state $y(\mathbf{x})$ governed by the elliptic partial differential equation (PDE)

$$-\text{div}\,(\kappa(\mathbf{x})\nabla y(\mathbf{x})) = u(\mathbf{x}), \quad \mathbf{x} \in \Omega, \qquad y(\mathbf{x}) = 0, \quad \mathbf{x} \in \partial\Omega, \tag{1}$$

where $\Omega \subset \mathbb{R}^n$, $n = 1, 2, 3$ is a bounded domain and $\partial\Omega$ its boundary. Here u is a control variable from an admissible set U_{ad}, which drives the state y to a target state $\hat{y}$. Problem (1) originates from the stationary heat conduction equation. In this setting, $\kappa(\mathbf{x})$ denotes the thermal conductivity of Ω, $y(\mathbf{x})$ is the temperature at a particular position $\mathbf{x}$ and $u(\mathbf{x})$ represents a controlled heat source. The goal is to find the optimal control variable u^* which minimizes the cost functional for $\nu \in \mathbb{R}^+$,

$$J(y, u) = \frac{1}{2}\int_\Omega |y(\mathbf{x}) - \hat{y}(\mathbf{x})|^2 \,\mathrm{d}\mathbf{x} + \frac{\nu}{2}\|u\|^2_{U_{\text{ad}}}, \tag{2}$$

subject to the constraint (1). The term $\frac{\nu}{2}\|u\|^2_{U_{\text{ad}}}$ can be considered as the cost of applying such a control u. It is said that the control is expensive if ν is large. From a mathematical viewpoint, the presence of this term with $\nu \in \mathbb{R}^+$ has a regularizing effect on the optimal control.

The analysis of Domain Decomposition methods (DDMs) for the elliptic PDE (1) is well established, see for instance [12]. Much less is known for DD methods applied to PDE-constrained optimal control problems, see for instance [5, 6]. Although the admissible set U_{ad} is often considered as $L^2(\Omega)$ for such elliptic control problems, a recent study shows that the energy space $H^{-1}(\Omega)$ can also be used for the regularization [10]. Moreover, this space can be expanded with $L^2(0, T; H^{-1}(\Omega))$ to treat parabolic control problems [7]. From an analytical point of view, the first-order optimality system can be simplified to a Poisson type equation by using the energy

Martin J. Gander, Liu-Di Lu
University of Geneva, Switzerland, e-mail: martin.gander@unige.ch, liudi.lu@unige.ch

space $H^{-1}(\Omega)$, whereas a biharmonic type problem still needs to be treated for the usual $L^2(\Omega)$ regularization. Moreover, applications of the energy norm can also be found in electrical engineering, fluid mechanics [9], etc.

Inspired by this approach, we study in this paper DDMs applied to the optimal control problem (1)-(2) using the energy norm. More precisely, we introduce in Section 2 the use of the energy norm H^{-1} for the elliptic control problem, and compare the optimality system with that of the L^2 norm. Although we consider for simplicity an unconstrained control, this can be extended to problems with state or control constraints, see also [13]. We then provide in Section 3 a convergence analysis of the Dirichlet-Neumann (DN) [1] and the Neumann-Neumann (NN) [2] methods applied to the optimality system. Some numerical experiments are given in Section 4, where we conclude with some comments.

2 Regularization: L^2 vs H^{-1}

We assume that both the control u and the target state $\hat{y}$ are in $L^2(\Omega)$, and consider first $U_{\text{ad}} = L^2(\Omega)$ as the set of all feasible controls. Using the Lagrange multiplier approach [13], we get for the first-order optimality system for problem (1)–(2)

$$\begin{aligned} -\text{div}\,(\kappa(\mathbf{x})\nabla y(\mathbf{x})) &= u(\mathbf{x}), \quad \mathbf{x}\in\Omega, & y(\mathbf{x}) &= 0, \quad \mathbf{x}\in\partial\Omega,\\ -\text{div}\,(\kappa(\mathbf{x})\nabla p(\mathbf{x})) &= y(\mathbf{x}) - \hat{y}(\mathbf{x}), \quad \mathbf{x}\in\Omega, & p(\mathbf{x}) &= 0, \quad \mathbf{x}\in\partial\Omega,\\ p(\mathbf{x}) + \nu u(\mathbf{x}) &= 0, \quad \mathbf{x}\in\Omega, \end{aligned} \tag{3}$$

where p is the Lagrange multiplier (or adjoint state). Inserting the third equation of (3) into the first equation, and the result into the second equation, we can rewrite the optimality system (3) with one single variable, for instance, with respect to the state variable y as

$$\begin{aligned} \nu\text{div}\Big(\kappa(\mathbf{x})\nabla\big(\text{div}\,(\kappa(\mathbf{x})\nabla y(\mathbf{x}))\big)\Big) + y(\mathbf{x}) &= \hat{y}(\mathbf{x}), & \mathbf{x}\in\Omega,\\ \text{div}\,(\kappa(\mathbf{x})\nabla y(\mathbf{x})) = y(\mathbf{x}) &= 0, & \mathbf{x}\in\partial\Omega. \end{aligned} \tag{4}$$

In particular, we identify in (4) a biharmonic operator by taking the conductivity $\kappa(\mathbf{x}) = 1$ everywhere over the domain.

We consider now $U_{\text{ad}} = H^{-1}(\Omega)$ in (2) as the set of all feasible controls. As proposed in [10], we can define the norm in $H^{-1}(\Omega)$ by

$$\|u\|^2_{H^{-1}(\Omega)} := \|\sqrt{\kappa}\nabla y\|^2_{L^2(\Omega)}, \tag{5}$$

which is the energy norm. Note that the conductivity κ is positive. On the other hand, following the same reasoning as in the $L^2(\Omega)$ case to derive the optimality system, we obtain

$$-\nu\text{div}\,(\kappa(\mathbf{x})\nabla y(\mathbf{x})) + y(\mathbf{x}) = \hat{y}(\mathbf{x}), \quad \mathbf{x}\in\Omega, \qquad y(\mathbf{x}) = 0, \quad \mathbf{x}\in\partial\Omega. \tag{6}$$

Comparing (6) with the reduced optimality system under L^2 regularization (4), we observe that indeed only a Laplace type operator needs to be solved in (6).

Remark 1 We need to be careful when comparing solutions of the two reduced optimality systems (4) and (6), since we penalize the control in different norms and solve different equations. In the L^2 case, the control can be determined by $u = -\frac{1}{\nu}p$ which is proportional to the adjoint state variable, while it is proportional to the state variable in the H^{-1} case, since $u = \frac{1}{\nu}(\hat{y} - y)$. Furthermore, the solution is less regular in the H^{-1} case as shown in [10].

Remark 2 Depending on the value of ν, (6) is a singularly perturbed PDE. Standard numerical methods can perform poorly, we refer to the monograph [11] for a review of robust numerical methods for such problems. In the recent work [8], the authors use an algebraic multigrid method and a balancing domain decomposition by constraints preconditioner for a finite element discretization to treat the problem (6). They observed that optimal convergence is ensured with $\nu = h^2$, h being the mesh size.

3 Convergence analysis of DD methods

We now provide a convergence analysis for the DN and the NN methods applied to solve the reduced optimality system (6), and then compare with DN and NN methods applied to (4) from [5].

Without loss of generality, the analysis is given under the assumption that the target state $\hat{y} = 0$, meaning that we focus on the error equation related to (6). Moreover, we assume that the conductivity coefficient $\kappa(x) = 1$ everywhere over the domain for the following analysis, although the DN and NN methods are defined for a general $\kappa(x)$. Let us first consider the one-dimensional case with the domain $\Omega = (0, 1)$. We decompose it into two non-overlapping subdomains $\Omega_1 = (0, \alpha)$ and $\Omega_2 = (\alpha, 1)$ with α the interface. We denote by e_i the error in domain Ω_i for $i = 1, 2$.

For the DN method, the error equations for (6) are for iteration index $n = 1, 2, \ldots$,

$$\begin{aligned} &\partial_{xx} e_1^n - \nu^{-1} e_1^n = 0, \quad e_1^n(0) = 0, \quad e_1^n(\alpha) = e_\alpha^{n-1}, \\ &\partial_{xx} e_2^n - \nu^{-1} e_2^n = 0, \quad e_2^n(1) = 0, \quad \partial_x e_2^n(\alpha) = \partial_x e_1^n(\alpha), \end{aligned} \tag{7}$$

with $e_\alpha^n := (1 - \theta) e_\alpha^{n-1} + \theta e_2^n(\alpha)$ and $\theta \in (0, 1)$ a relaxation parameter. We notice that the error equations (7) are similar to the ones in [4, Equation (2.4)] for applying the Dirichlet-Neumann waveform relaxation (DNWR) method to the heat equation. Indeed, after a Laplace transform, the error equations for the DNWR method in the one dimensional case are like (7), where ν^{-1} is replaced by s. For this reason, we follow the same calculations as in [4] and find the convergence factor

$$\rho_{\mathrm{DN}} := \left| 1 - \theta \left[1 + \tanh\left(\sqrt{\nu^{-1}} (1 - \alpha) \right) \coth\left(\sqrt{\nu^{-1}} \alpha \right) \right] \right|. \tag{8}$$

This leads us to the following convergence results.

Theorem 1 *The DN method with $\theta = 1$ applied to Problem* (6) *converges if and only if the interface is closer to the right boundary (i.e., $\alpha > \frac{1}{2}$).*

Proof Taking $\theta = 1$ in (8), we obtain the convergence factor

$$\rho_{\mathrm{DN}} = \tanh\left(\sqrt{\nu^{-1}}(1-\alpha)\right)\coth\left(\sqrt{\nu^{-1}}\alpha\right),$$

that is smaller than 1 if and only if $\alpha > \frac{1}{2}$ which can be seen by studying the function $f(x) = \sinh(1-x)\cosh(x) - \cosh(1-x)\sinh(x)$ for $x \in [0,1]$. □

Theorem 2 *For symmetric subdomains (i.e., $\alpha = \frac{1}{2}$), the convergence of the DN method for Problem* (6) *is linear and is independent of the value of the regularization parameter ν. It converges in two iterations if $\theta = \frac{1}{2}$.*

Proof We just have to take $\alpha = \frac{1}{2}$ in (8) and finds $\rho_{\mathrm{DN}} = |1 - 2\theta|$. □

Theorem 3 *For asymmetric subdomains (i.e., $\alpha \neq \frac{1}{2}$), the DN method converges for Problem* (6) *if and only if*

$$0 < \theta < 2\theta^{\star}_{DN}, \quad \theta^{\star}_{DN} := \frac{1}{1 + \tanh\left(\sqrt{\nu^{-1}}(1-\alpha)\right)\coth\left(\sqrt{\nu^{-1}}\alpha\right)}. \tag{9}$$

Moreover, it converges in two iterations if and only if $\theta = \theta^{\star}_{DN}$.

Proof From the convergence factor (8), the interior part of the absolute value is smaller than 1, since $\theta \in (0,1)$ and $1 + \tanh\left(\sqrt{\nu^{-1}}(1-\alpha)\right)\coth\left(\sqrt{\nu^{-1}}\alpha\right)$ is strictly positive. We then just need to ensure that

$$\theta\left[1 + \tanh\left(\sqrt{\nu^{-1}}(1-\alpha)\right)\coth\left(\sqrt{\nu^{-1}}\alpha\right)\right] < 2,$$

which leads to the inequality in (9). On the other hand, we find directly $\theta^{\star}_{\mathrm{DN}}$ by equating (8) to zero. □

Remark 3 As expected, we find similar results in the symmetric case as for the L^2 regularization. However, we have an optimal relaxation parameter for asymmetric decompositions, which is strictly smaller than 1, whereas a pair of parameters is needed for the L^2 regularization which can be greater than one in some cases, see [5]. This is due to the fact that two transmission conditions need to be considered for a biharmonic type problem.

The error equations for the NN method, for iteration index $n = 1, 2, \cdots$, are

$$\partial_{xx} e_j^n - \nu^{-1} e_j^n = 0, \quad e_1^n(0) = 0, \quad e_2^n(1) = 0, \quad e_j^n(\alpha) = e_{\alpha}^{n-1}, \tag{10}$$

where the transmission condition is given by $e_{\alpha}^n := e_{\alpha}^{n-1} - \theta\left(\psi_1^n(\alpha) + \psi_2^n(\alpha)\right)$ and ψ_j^n satisfies the correction step

$$\partial_{xx}\psi_j^n - \nu^{-1}\psi_j^n = 0, \quad \psi_1^n(0) = 0, \quad \psi_2^n(1) = 0, \quad \partial_{n_j}\psi_j^n(\alpha) = \partial_{n_1}e_1^n(\alpha) + \partial_{n_2}e_2^n(\alpha). \tag{11}$$

Solving (10)-(11) on each domain Ω_j and applying the boundary conditions at $x = 0$ and $x = 1$, we find the solutions with A^n, B^n, C^n, D^n four coefficients to be determined for e_1^n, e_2^n, ψ_1^n and ψ_2^n. Evaluating then e_j^n at $x = \alpha$, and using the transmission condition $e_j^n(\alpha) = e_\alpha^{n-1}$, we can determine the two coefficients A^n, B^n and get

$$e_1^n(x) = e_\alpha^{n-1}\frac{\sinh(\sqrt{\nu^{-1}}x)}{\sinh(\sqrt{\nu^{-1}}\alpha)}, \quad e_2^n(x) = e_\alpha^{n-1}\frac{\sinh\left(\sqrt{\nu^{-1}}(1-x)\right)}{\sinh\left(\sqrt{\nu^{-1}}(1-\alpha)\right)}. \tag{12}$$

Similarly, we evaluate $\partial_{n_j}\psi_j^n$ at $x = \alpha$, and using the transmission condition $\partial_{n_j}\psi_j^n(\alpha) = \partial_{n_1}e_1^n(\alpha) + \partial_{n_2}e_2^n(\alpha)$ with the help of (12), we can determine the remaining two coefficients C^n, D^n and get,

$$\psi_1^n(x) = e_\alpha^{n-1}\frac{\sinh(\sqrt{\nu^{-1}}x)}{\cosh(\sqrt{\nu^{-1}}\alpha)}\left(\coth(\sqrt{\nu^{-1}}\alpha) + \coth(\sqrt{\nu^{-1}}(1-\alpha))\right),$$
$$\psi_2^n(x) = e_\alpha^{n-1}\frac{\sinh\left(\sqrt{\nu^{-1}}(1-x)\right)}{\cosh(\sqrt{\nu^{-1}}(1-\alpha))}\left(\coth(\sqrt{\nu^{-1}}\alpha) + \coth(\sqrt{\nu^{-1}}(1-\alpha))\right).$$

Using finally the definition of the transmission condition e_α^n, we find the convergence factor

$$\begin{aligned}\rho_{\mathrm{NN}} := \Big|1 - \theta\Big(&\tanh(\sqrt{\nu^{-1}}\alpha) + \tanh\left(\sqrt{\nu^{-1}}(1-\alpha)\right)\Big)\\ &\times\Big(\coth(\sqrt{\nu^{-1}}\alpha) + \coth(\sqrt{\nu^{-1}}(1-\alpha))\Big)\Big|.\end{aligned} \tag{13}$$

We obtain the following convergence results.

Theorem 4 *For symmetric subdomains (i.e., $\alpha = \frac{1}{2}$), the convergence of the NN method for Problem* (6) *is linear and is independent of the value of the regularization parameter ν. It converges in two iterations if $\theta = \frac{1}{4}$.*

Proof We just have to take $\alpha = \frac{1}{2}$ in (13) and find $\rho_{\mathrm{NN}} = |1 - 4\theta|$. □

Theorem 5 *For asymmetric subdomains (i.e., $\alpha \neq \frac{1}{2}$), the NN method converges for Problem* (6) *if and only if*

$$0 < \theta < 2\theta_{NN}^\star, \quad \theta_{NN}^\star := \frac{1}{\left(\tanh(\sqrt{\nu^{-1}}\alpha)+\tanh\left(\sqrt{\nu^{-1}}(1-\alpha)\right)\right)\left(\coth(\sqrt{\nu^{-1}}\alpha)+\coth(\sqrt{\nu^{-1}}(1-\alpha))\right)}. \tag{14}$$

Furthermore, it converges in two iterations if and only if $\theta = \theta_{NN}^\star$.

Proof Following the same steps as in the proof of Theorem 3, we obtain the inequality (14), and we find directly $\theta_{\mathrm{NN}}^\star$ by equating (13) to zero. □

Remark 4 As shown in Theorem 3 and in Theorem 5, both the DN and the NN methods converge in two iterations to the exact solution. Moreover, we have a bound for the relaxation parameter θ of each method for which the convergence of the method is guaranteed.

The above analysis can also be extended to the two-dimensional case. More precisely, we assume that the domain Ω is now given by $[0,1] \times [0,1]$, which is then divided into two non-overlapping subdomains $\Omega_1 = (0,\alpha) \times [0,1]$ and $\Omega_2 = (\alpha,1) \times [0,1]$, with the interface at $x_1 = \alpha$ denoted by $\Gamma := \{\alpha\} \times [0,1]$. In addition, we keep the assumption that $\hat{y} = 0$ and $\kappa(x) = 1$. The two-dimensional analysis is often carried out by using a Fourier expansion in one direction, in our case, the x_2 direction $e_i^n(x_1,x_2) = \sum_{k=0}^{\infty} \hat{e}_i(x_1,k)\sin(k\pi x_2)$. In this way, the error function related to $e_i(x_1,x_2)$ passes to $\hat{e}_i(x_1,k)$, and for instance, in the DN case is governed by

$$\begin{aligned} &\partial_{x_1x_1}\hat{e}_1^n - \frac{\nu k^2\pi^2+1}{\nu}\hat{e}_1^n = 0, \quad \hat{e}_1^n(0,k) = 0, \quad \hat{e}_1^n(\alpha,k) = \hat{e}_\alpha^{n-1}, \\ &\partial_{x_1x_1}\hat{e}_2^n - \frac{\nu k^2\pi^2+1}{\nu}\hat{e}_2^n = 0, \quad \hat{e}_2^n(1,k) = 0, \quad \partial_{x_1}\hat{e}_2^n(\alpha,k) = \partial_{x_1}\hat{e}_1^n(\alpha,k), \end{aligned} \tag{15}$$

with $\hat{e}_\alpha^n := (1-\theta)\hat{e}_\alpha^{n-1} + \theta\hat{e}_2^n(\alpha,k)$ and $\theta \in (0,1)$. We observe that (15) has the same structure as in the one-dimensional case (7), where ν^{-1} is replaced by $\frac{\nu k^2\pi^2+1}{\nu}$. Therefore, the same type of reasoning can be applied to analyze this iteration, and we have the following results.

Theorem 6 *For symmetric subdomains (i.e., $\alpha = \frac{1}{2}$), the convergence of the DN and the NN methods for Problem* (6) *are both linear and independent of the value of ν. It converges in two iterations if $\theta = \frac{1}{2}$ for the DN method and $\theta = \frac{1}{4}$ for the NN method.*

Theorem 7 *For asymmetric subdomains (i.e., $\alpha \neq \frac{1}{2}$), the DN method converges for Problem* (6) *whenever*

$$\rho_{DN2d} := \sup_{k\in\mathbb{N}} \left| 1 - \theta \left[1 + \tanh\left(\sqrt{\frac{\nu k^2\pi^2+1}{\nu}}(1-\alpha) \right) \coth\left(\sqrt{\frac{\nu k^2\pi^2+1}{\nu}}\alpha \right) \right] \right| < 1. \tag{16}$$

The NN method converges for Problem (6) *whenever*

$$\begin{aligned} \rho_{NN2d} := \sup_{k\in\mathbb{N}} \Bigg| 1 - \theta \Bigg(&\tanh\left(\sqrt{\frac{\nu k^2\pi^2+1}{\nu}}\alpha\right) + \tanh\left(\sqrt{\frac{\nu k^2\pi^2+1}{\nu}}(1-\alpha) \right) \Bigg) \\ &\cdot \left(\coth\left(\sqrt{\frac{\nu k^2\pi^2+1}{\nu}}\alpha\right) + \coth\left(\sqrt{\frac{\nu k^2\pi^2+1}{\nu}}(1-\alpha) \right) \right) \Bigg| < 1. \end{aligned} \tag{17}$$

4 Numerical experiments

In this section, we provide numerical experiments to illustrate the convergence rate of the DN and the NN methods for Problem (1)-(2) with $\nu = 1$ and $\hat{y} = 0$. Figure 1 (top) shows the one-dimensional convergence behaviour of these two methods for different choices of θ with an asymmetric decomposition $\alpha = \frac{1}{3}$. The best choices of the relaxation parameter are given by $\theta^{\star}_{\mathrm{DN}} \approx 0.355$ and $\theta^{\star}_{\mathrm{NN}} \approx 0.229$. In particular, we observe some divergence behavior in the case of the NN method for $\theta = 0.5$ and $\theta = 0.7$. Indeed, this corresponds to the result in Theorem 5, since these two values are greater than $2\theta^{\star}_{\mathrm{NN}}$ which is the upper bound for the relaxation parameter θ. Furthermore, we observe the convergence to the exact solution in two iterations for a non-symmetric domain decomposition, whereas a three-step convergence is needed for the L^2 regularization [5]. Figure 1 (bottom) presents the behavior of the convergence factors (16) and (17) in the two-dimensional case. The interface here is chosen to be asymmetric $\Gamma = \{\frac{1}{3}\} \times [0, 1]$. We observe good convergence behaviors for some tested relaxation parameters θ. Furthermore, the NN method does not converge for $\theta = 0.5$ and $\theta = 0.7$ as in the one-dimensional case. We obtain that $\rho_{\mathrm{DN2d}} \approx 0.173$ for $\theta^{\star}_{\mathrm{DN2d}} \approx 0.414$ and $\rho_{\mathrm{NN2d}} \approx 0.046$ for $\theta^{\star}_{\mathrm{NN2d}} \approx 0.239$. These two optimal relaxation parameters can also be found by equioscillating the value of the convergence factor both at $k = 0$ and $k \longrightarrow \infty$. Moreover for each method, we find that these optimal relaxation parameters stay very close between the one-dimensional and the two-dimensional case.

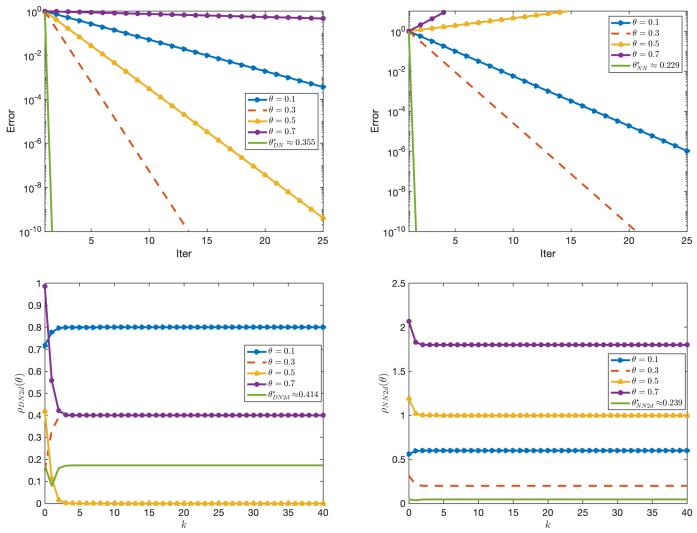


Fig. 1 Error decay in 1D w.r.t. the number of iterations for the DN method (top-left) and the NN method (top-right) with the interface at $\alpha = \frac{1}{3}$. Convergence factors (16) and (17) in 2D w.r.t. the value of $k \in [0, 40]$ for the DN method (bottom-left) and the NN method (bottom-right) with the interface at $\Gamma = \{\frac{1}{3}\} \times [0, 1]$.

To conclude, we presented a convergence analysis of the DN and the NN methods for elliptic optimal control problems using the energy norm for regularization. Only one Poisson type equation needs to be solved, whereas a biharmonic type equation is required for L^2 regularization. Under the energy norm, we found similar results in the symmetric case as for the Poisson problem. Therefore, we can expect similar convergence behavior for many subdomains as presented in [3]. Furthermore, explicit formulations along with an upper bound are also given for the optimal relaxation parameters with a non-symmetric decomposition, for which the methods converge still in two iterations in the one-dimensional case.

References

1. Bjørstad, P. E. and Widlund, O. B. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM Journal on Numerical Analysis* **23**(6), 1097–1120 (1986).
2. Bourgat, J.-F., Glowinski, R., Le Tallec, P., and Vidrascu, M. Variational formulation and algorithm for trace operator in Domain Decomposition calculations. In: *Domain Decomposition Methods*, 3–16. SIAM (1989).
3. Chaouqui, F., Ciaramella, G., Gander, M. J., and Vanzan, T. On the scalability of classical one-level Domain-Decomposition methods. *Vietnam J. Math.* **46**, 1053–1088 (2018).
4. Gander, M. J., Kwok, F., and Mandal, B. Dirichlet-Neumann and Neumann-Neumann waveform relaxation algorithms for parabolic problems. *Electronic Transactions on Numerical Analysis* **45**, 424–456 (2016).
5. Gander, M. J., Kwok, F., and Mandal, B. C. Convergence of substructuring methods for elliptic optimal control problems. In: *Domain Decomposition Methods in Science and Engineering XXIV*, 291–300. Springer International Publishing, Cham (2018).
6. Heinkenschloss, M. A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems. *Journal of Computational and Applied Mathematics* **173**(1), 169–198 (2005).
7. Langer, U., Steinbach, O., Tröltzsch, F., and Yang, H. Space-time finite element discretization of parabolic optimal control problems with energy regularization. *SIAM Journal on Numerical Analysis* **59**(2), 675–695 (2021).
8. Langer, U., Steinbach, O., and Yang, H. Robust discretization and solvers for elliptic optimal control problems with energy regularization. *Computational Methods in Applied Mathematics* **22**(1), 97–111 (2022).
9. Lin, Z., Thiffeault, J.-L., and Doering, C. R. Optimal stirring strategies for passive scalar mixing. *Journal of Fluid Mechanics* **675**, 465–476 (2011).
10. Neumüller, M. and Steinbach, O. Regularization error estimates for distributed control problems in energy spaces. *Mathematical Methods in the Applied Sciences* **44**(5), 4176–4191 (2021).
11. Roos, H.-G., Stynes, M., and Tobiska, L. *Robust Numerical Methods for Singularly Perturbed Differential Equations*. Springer Berlin, Heidelberg, 2 ed. (2008).
12. Smith, B., Bjørstad, P. E., and Gropp, W. *Domain Decomposition, Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press (1996).
13. Tröltzsch, F. *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, vol. 112. Graduate Studies in Mathematics (2010).

An Introduction to Heterogeneous Domain Decomposition Methods for Multi-Physics Problems

Martin J. Gander and Véronique Martin

1 Mono-physics and multi-physics problems

In order to understand research problems in multi-physics, it is instructive to first look at coupling conditions for mono-physics problems, which arise naturally in domain decomposition. To do so, we consider the model problem $\mathcal{L}u := (\eta - \Delta)u = f$ in a domain Ω, with suitable boundary conditions. In this case, the solution u we want to compute is well defined, and domain decomposition methods provide two techniques to couple solutions of such mono-physics problems: the first one comes from the alternating Schwarz method [20], see Figure 1 (left) for the historical domain and its decomposition. The alternating Schwarz method solves for $n = 1, 2, \ldots$

$$\mathcal{L}u_1^n = f \text{ in } \Omega_1, \quad u_1^n = u_2^{n-1} \text{ on } \Gamma_1, \qquad \mathcal{L}u_2^n = f \text{ in } \Omega_2, \quad u_2^n = u_1^n \text{ on } \Gamma_2, \quad (1)$$

starting with some u_2^0. At convergence, this method defines naturally coupling conditions that involve an overlap, namely the two subdomain solutions must satisfy $u_1 = u_2$ on Γ_1 and $u_2 = u_1$ on Γ_2. We thus found the classical *overlapping coupling conditions for second order mono-physics problems*. The second coupling technique comes from historical substructuring methods introduced by Przemieniecki in [19].

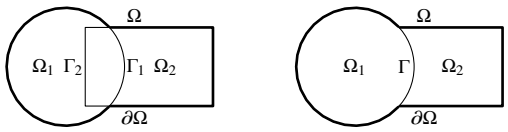


Fig. 1 Left: Schwarz coupling. Right: Przemieniecki or Schur coupling.

Martin J. Gander
Section de Mathématiques, Université de Genève, e-mail: martin.gander@unige.ch

Véronique Martin
UMR CNRS 7352, Université de Picardie Jules Verne, e-mail: veronique.martin@u-picardie.fr

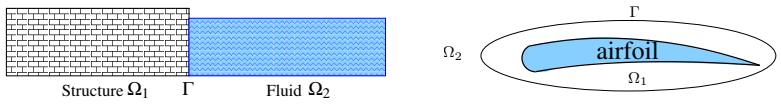


Fig. 2 Left: Fluid-structure interaction. Right: Flow around an airfoil.

For a Schwarz like example, see Figure 1 (right), but now with subdomains that do not overlap, Przemieniecki posed directly the coupled problem with Dirichlet and Neumann conditions,

$$\mathcal{L}u_1 = f \text{ in } \Omega_1, \quad u_1 = u_2 \text{ on } \Gamma, \qquad \mathcal{L}u_2 = f \text{ in } \Omega_2, \quad \partial_n u_2 = \partial_n u_1 \text{ on } \Gamma. \tag{2}$$

He solved it by first assuming the Dirichlet condition to be known, eliminating then the interior unknowns in the subdomains, and finally by imposing the Neumann coupling conditions obtained as equation for the Dirichlet traces, see [12] for more details. The equations on Γ in (2) are the classical *non-overlapping coupling conditions for second order mono-physics problems*. These conditions can also be used to obtain an iterative algorithm, namely the Dirichlet-Neumann method, by solving

$$\mathcal{L}u_1^n = f \text{ in } \Omega_1, \quad u_1^n = u_2^{n-1} \text{ on } \Gamma, \qquad \mathcal{L}u_2^n = f \text{ in } \Omega_2, \quad \partial_n u_2^n = \partial_n u_1^n \text{ on } \Gamma, \tag{3}$$

which however needs a relaxation parameter for convergence, see [3, Section 4.7]. This naturally raises the question if these physical coupling conditions are good to obtain rapid convergence. One could consider for example Robin conditions in (3),

$$(\partial_{n_1} + p)u_1^n = (\partial_{n_1} + p)u_2^{n-1} \text{ on } \Gamma_1, \quad (\partial_{n_2} + p)u_2^n = (\partial_{n_2} + p)u_1^n \text{ on } \Gamma_2, \tag{4}$$

where p can be a number, a function or even an operator as advocated by Lions [18]. One can now use both overlapping and non-overlapping ($\Gamma_1 = \Gamma_2 = \Gamma$) configurations, since the Robin conditions imply the coupling conditions on Γ in (2) at convergence (just take the sum and difference of the Robin conditions). We call the Robin conditions *transmission conditions*, since they must transmit information as effectively as possible for fast convergence, a research field that led to optimized Schwarz methods, see [7] for an introduction.

For Multi-Physics Problems, we have to distinguish two situations. The first one is where the physics is truly different in different regions, as for example in fluid structure interaction, see Figure 2 (left). Here the solution u we want to compute is also well defined, the coupling conditions are given by the physics of the problem along the interface Γ between the fluid and the structure, and only non-overlapping techniques make sense. Once good physical coupling conditions are found, the question is what are good transmission conditions for fast convergence when one solves alternatingly the structure and fluid problems in Ω_1 and Ω_2, and which imply the coupling conditions on Γ at convergence.

The second situation is when in principle we have a mono-physics problem in Ω, but different physical models are used in different regions for computational savings, see Figure 2 (left) for a flow around an airfoil. Here one wants to use an expensive

model only where it is necessary, in Ω_1 close to the airfoil, and far away a cheaper model suffices in Ω_2 to save computation time. The question is then what are good coupling conditions to get close to the expensive solution everywhere. For this one can consider non-overlapping techniques with one interface Γ, or also overlapping ones with two interfaces Γ_1 and Γ_2 at an overlapping distance, see Figure 1. Once good coupling conditions are found, again the question arises on what are good transmission conditions for fast iterative convergence when solving alternatingly on Ω_1 the expensive and on Ω_2 the cheap model, which also imply the good coupling conditions at convergence.

2 Truly multi-physics problems

A typical example of a truly multi-physics problem can be found in [4]:

> "In a second circumstance, one may be obliged to consider truly different models to account for the presence of distinct physical problems within the same global domain. This case is usually indicated as multi-physics or multi-field problem."

The problem considered is the deformation of an artery. The fluid equations for the velocity field $\mathbf{u}$ and the pressure p are

$$\begin{aligned} \rho_{\mathrm{f}}\left(\left.\frac{\partial \boldsymbol{u}}{\partial t}\right|_{x_0}+(\boldsymbol{u}-\boldsymbol{w}^{\mathrm{f}})\cdot\boldsymbol{\nabla}\boldsymbol{u}\right)-\operatorname{div}[\boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{u},p)]&=\boldsymbol{f}_{\mathrm{f}} \quad \text{in } \Omega^{\mathrm{f}}(t),\\ \operatorname{div}\boldsymbol{u}&=0 \quad \text{in } \Omega^{\mathrm{f}}(t),\\ \boldsymbol{u}=\boldsymbol{u}_{\mathrm{in}} \quad \text{on } \Gamma^{\mathrm{in}}(t), \quad \boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{u},p)\cdot\boldsymbol{n}_{\mathrm{f}}&=\boldsymbol{g}_{\mathrm{f}} \quad \text{on } \Gamma^{\mathrm{out}}(t). \end{aligned}$$

The solid equations for the displacement $\mathbf{d}^s$ are

$$\begin{aligned} \rho_{\mathrm{s}}\frac{\partial^2\boldsymbol{d}^{\mathrm{s}}}{\partial t^2}-\operatorname{div}_{|x_0}(\boldsymbol{\sigma}_{\mathrm{s}}(\boldsymbol{d}^{\mathrm{s}}))&=\boldsymbol{f}_{\mathrm{s}} \quad \text{in } \Omega_0^{\mathrm{s}},\\ \boldsymbol{\sigma}_{\mathrm{s}}(\boldsymbol{d}^{\mathrm{s}})\cdot\boldsymbol{n}_{\mathrm{s}}&=\boldsymbol{g}_{\mathrm{s}} \quad \text{on } \partial\Omega_0^{\mathrm{s}}\setminus\Gamma_0, \end{aligned}$$

and the physical coupling conditions are ("Dirichlet" and "Neumann")

$$\begin{aligned} \boldsymbol{x}_t^{\mathrm{s}}=x_0+\lambda=\boldsymbol{x}_t^{\mathrm{f}}, \qquad \boldsymbol{u}\circ\boldsymbol{x}_t^{\mathrm{f}}&=\frac{\partial\lambda}{\partial t},\\ (\boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{u},p)\cdot\boldsymbol{n}_{\mathrm{f}})\circ\boldsymbol{x}_t^{\mathrm{f}}&=-\boldsymbol{\sigma}_{\mathrm{s}}(\boldsymbol{d}^{\mathrm{s}})\cdot\boldsymbol{n}_{\mathrm{s}}, \end{aligned}$$

imposing the matching of the interface displacements from the fluid and solid subdomains, the continuity of the velocities and the normal stresses. The authors propose to use directly these coupling conditions also as transmission conditions and study the following methods:

$$\begin{aligned} &\textit{Dirichlet-Neumann (DN):}\ P_k=P_{DN}=S_{\mathrm{s}}'(\lambda^k),\ \text{for } \alpha_{\mathrm{f}}^k=0,\ \alpha_{\mathrm{s}}^k=1,\\ &\textit{Neumann-Dirichlet (ND):}\ P_k=P_{ND}=S_{\mathrm{f}}'(\lambda^k),\ \text{for } \alpha_{\mathrm{f}}^k=1,\ \alpha_{\mathrm{s}}^k=0,\\ &\textit{Neumann-Neumann (NN):}\ P_k=P_{NN} \text{ with } \alpha_{\mathrm{f}}^k+\alpha_{\mathrm{s}}^k=1,\ \alpha_{\mathrm{f}}^k,\alpha_{\mathrm{s}}^k\neq 0. \end{aligned}$$

Our interest for such multiphysics problems mainly focused on developing transmission conditions for fast convergence. In [8] we developed a non-overlapping optimized Schwarz method for jumping coefficient diffusion problems that converges independently of the mesh parameter, and faster and faster the bigger the jump becomes: the method truly benefits from the multi-physics nature of the problem. In [13] we designed and studied heterogeneous optimized Schwarz methods for coupling Helmholtz and Laplace equations; for more general elliptic problems, see [14, 15], for Stokes-Darcy see [16], and for a new technique to automatically obtain such transmission conditions through probing, see [11].

3 Multi-physics problems for computational savings

The classical mathematical approach for such problems is the technique of *matched asymptotic expansions*. For the model problem of advection diffusion with $a > 0$,

$$\nu\partial_{xx}u + a\partial_x u - \eta u = 0 \quad \text{in } (0,1), \quad u(0) = 0,\ u(1) = 1, \tag{5}$$

a regular expansion for ν small, $u = u_0 + \nu u_1 + \ldots$, gives

$$a\partial_x u_0 - \eta u_0 + \nu(\partial_{xx}u_0 + a\partial_x u_1 - \eta u_1) + \ldots = 0,$$

and therefore by matching terms $u_0(x) = e^{\frac{\eta}{a}(x-1)}$, $u_1(x) = -\frac{\eta^2}{a^3}e^{\frac{\eta}{a}(x-1)}(x-1)$ etc, which can not capture the zero boundary condition at $x = 0$. One thus introduces the stretched variable $x := \epsilon\xi$, $\tilde{u}(\xi) := u(\epsilon\xi)$, which yields $\partial_\xi\tilde{u}(\xi) = \partial_x u(\epsilon\xi)\epsilon$ and

$$\frac{\nu}{\epsilon^2}\partial_{\xi\xi}\tilde{u} + \frac{a}{\epsilon}\partial_x\tilde{u} - \eta\tilde{u} = 0 \quad \Longrightarrow \quad \partial_{\xi\xi}\tilde{u} + a\partial_x\tilde{u} - \nu\eta\tilde{u} = 0,$$

where we multiplied by ϵ and choose $\epsilon := \nu$. A regular expansion for ν small, $\tilde{u} = \tilde{u}_0 + \nu\tilde{u}_1 + \ldots$ now gives for the first term with $\tilde{u}_0(0) = 0$

$$\partial_{\xi\xi}\tilde{u}_0 + a\partial_\xi\tilde{u}_0 = 0 \quad \Longrightarrow \quad \tilde{u}_0(\xi) = C(e^{-a\xi} - 1).$$

The constant C is then determined by asymptotic matching, $\lim_{\xi=\infty}\tilde{u}_0(\xi) = -C = e^{-\frac{\eta}{a}} = \lim_{x=0}u_0(x)$. We obtain the inner and outer solutions $\tilde{u}_0(x) = e^{-\frac{\eta}{a}}(1 - e^{-\frac{ax}{\nu}})$, $u_0(x) = e^{\frac{\eta}{a}(x-1)}$, and the composite solution by summation, and subtraction of the common limit,

$$u_0^a(x) = -e^{-\frac{\eta}{a}}e^{-\frac{ax}{\nu}} + e^{\frac{\eta}{a}(x-1)}.$$

The exact solution of the problem is $u(x) = \frac{e^{\lambda_1 x} - e^{\lambda_2 x}}{e^{\lambda_1} - e^{\lambda_2}}$, with $\lambda_1 := \frac{-a+\sqrt{a^2+4\eta\nu}}{2\nu} = \frac{\eta}{a} - \frac{\eta^2}{a^3}\nu + O(\nu^2)$, $\lambda_2 := \frac{-a-\sqrt{a^2+4\eta\nu}}{2\nu} = -\frac{a}{\nu} - \frac{\eta}{a} + \frac{\eta^2}{a^3}\nu + O(\nu^2)$. We show in Figure 3 (left) the difference between the matched asymptotic solution and the exact one. Using asymptotic analysis, one can show

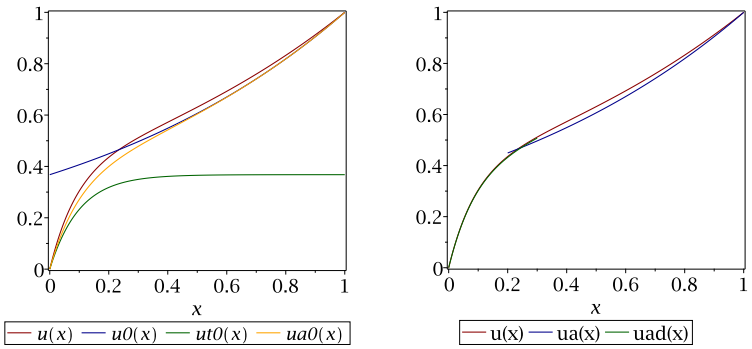


Fig. 3 Example for $a = 1$, $\eta = 1$, $\nu = 0.1$. Left: matched asymptotic expansions. Right: overlapping coupling from [5] with overlap $\gamma_{1,2} = (0.2, 0.3)$.

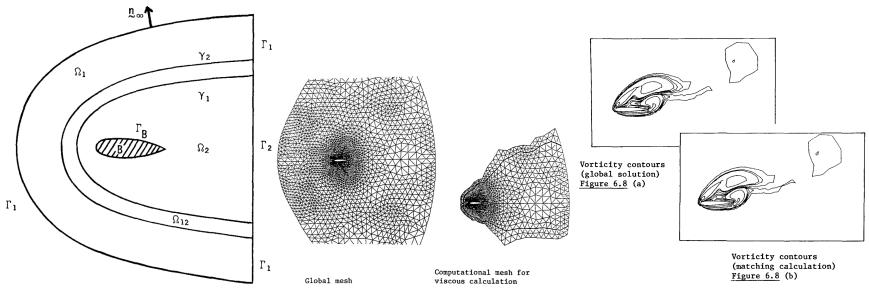


Fig. 4 Navier-Stokes and potential flow coupling from [5].

Proposition 1 (Matched asymptotic expansion) *For ν small, the matched asymptotic expansion approximation satisfies for $x = O(\nu^\alpha)$ the error estimate*

$$\|u - u_0^a\|_{L^\infty(0,x)} = \begin{cases} O(\nu) & 0 < \alpha < 1, \\ O(\nu^\alpha) & \alpha \geq 1. \end{cases} \tag{6}$$

An *optimal control method* for such a coupled solution was given in [5]:

> The main goal of this paper is to present a computational method for the coupling of two distinct mathematical models describing the same physical phenomenon, namely the flow of an *incompressible viscous fluid*. The basic idea is to replace the Navier-Stokes equations by the potential one in those regions where we can neglect the viscous effects and where the vorticity is small.

In order to achieve this coupling, the authors use an overlapping decomposition as in Figure 4 (left), and then impose on the Navier-Stokes equation the velocity $\mathbf{u} = \mathbf{v}$ on the interface γ_1, and on the potential flow the Dirichlet condition $\phi = \psi$ on the interface γ_2. They then determine $\mathbf{v}$ and ψ that minimize the functional

$$J(\mathbf{v}, \psi) := \frac{1}{2} \int_{\Omega_{12}} |\mathbf{u} - \nabla\phi|^2,$$

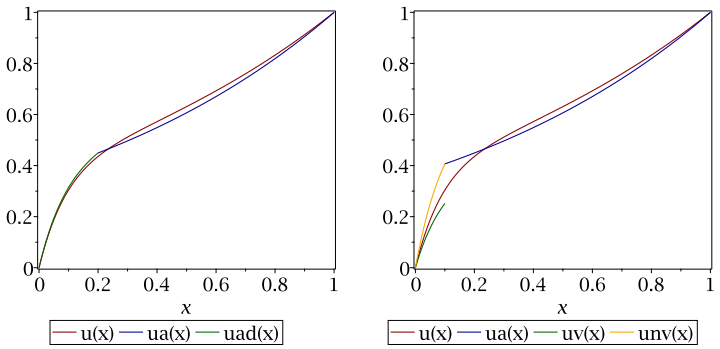


Fig. 5 Example for $a = \eta = 1$, $\nu = 0.1$. Left: χ-method with $\delta = 5$. Right: variational coupling with $L = 0.1$.

where $\mathbf{u}$ solves the Navier-Stokes equation in Ω_2 and ϕ the potential equation in Ω_1. For the model problem of the matched asymptotic expansion we solve for $\gamma_2 < \gamma_1$

$$\nu\partial_{xx}u_{ad} + a\partial_x u_{ad} - \eta u_{ad} = 0 \text{ in } (0,\gamma_1), \quad a\partial_x u_a - \eta u_a = 0 \text{ in } (\gamma_2, 1),$$

$u_{ad}(0) = 0$, $u_a(1) = 1$, and $u_{ad}(\gamma_1) = \psi$ with ψ which minimizes the norm $\|u_{ad} - u_a\|_{L^2(\gamma_1,\gamma_2)}$. We get with this approach the results shown in Figure 3 (right). Using asymptotic analysis, we obtain

Proposition 2 (Advection error estimate, valid for all methods) *At $x = O(\nu^\alpha)$, the advection approximation satisfies the error estimate*

$$u(x) - u_a(x) = \begin{cases} O(\nu) & 0 \le \alpha < 1, \\ O(1) & \alpha \ge 1. \end{cases} \tag{7}$$

Proposition 3 (Optimal control method) *For $\gamma_1 = O(\nu^\alpha)$, $\alpha \ge 0$, the optimal control method satisfies for the advection diffusion approximation the error estimate*

$$\|u - u_{ad}\|_{L^\infty(0,\gamma_1)} = \begin{cases} O(\nu) & 0 \le \alpha < 1, \\ O(\nu^{1-\beta}) & \alpha \ge 1, \gamma_2 = O(\nu^\beta), 0 \le \beta < 1, \\ O(1) & \textit{otherwise}. \end{cases} \tag{8}$$

The χ*-method* from [1] is a different such coupling method. The idea is to add in the advection reaction diffusion equation a cut-function for the diffusion,

$$-\nu\chi_\delta(\Delta u) + \mathbf{a}\cdot\nabla u + bu = f \quad \text{in } \Omega, \quad \chi_\delta(s) := \begin{cases} 0, & |s| \le \delta, \\ s, & |s| > \delta. \end{cases}$$

The authors say: "We remark that the perturbed equation is at least as difficult to solve as the imperturbed equation", but conceptually think it is better to solve the same equation on the entire domain. For our model problem, we obtain the results in Figure 5 (left). Using asymptotic analysis, one can show

Proposition 4 (χ-method) *For $\delta = O(\nu^{-\alpha})$, $\alpha \geq 0$, the $\chi-$method for* (5) *satisfies in the advection diffusion region the error estimate*

$$\|u - u_{ad}\|_{L^\infty(0,\gamma_1)} = \begin{cases} O(\nu) & 0 \leq \alpha < 1, \\ O(\nu^{2-\alpha}) & 1 \leq \alpha \leq 2, \\ O(1) & \alpha > 2. \end{cases} \tag{9}$$

A *non-overlapping DD coupling technique* ($\gamma := \gamma_1 = \gamma_2$) was proposed in [17]:

> "We deal with the coupling of hyperbolic and parabolic systems in a domain Ω divided into two disjoint subdomains Ω^+ and Ω^- [...] The justification of the interface conditions is based on a singular perturbation analysis, that is, the hyperbolic system is rendered parabolic by adding a small artificial "viscosity". As this goes to zero, the coupled parabolic-parabolic problem degenerates into the original one, yielding some conditions at the interface. These we take as interface conditions for the hyperbolic-parabolic problem. Actually, we discuss two alternative sets of interface conditions according to whether the regularization procedure is variational or nonvariational."

For our model problem, the variational condition is $(\nu\partial_x + a)u^v_{ad}(\gamma) = au_a(\gamma)$, and the non-variational condition is $u^{nv}_{ad}(\gamma) = u_a(\gamma)$. We show in Figure 5 (right) a computational result for the variational and non-variational conditions.

Proposition 5 (DD coupling technique) *For $\gamma = O(\nu^\alpha)$, $\alpha \geq 0$, we obtain in the advection diffusion region for the variational and non-variational approach*

$$\|u - u^v_{ad}\|_{L^\infty(0,\gamma)} = \begin{cases} O(\nu) & 0 \leq \alpha < 1, \\ O(\nu^\alpha) & \alpha \geq 1, \end{cases} \quad \|u - u^{nv}_{ad}\|_{L^\infty(0,\gamma)} = \begin{cases} O(\nu) & 0 \leq \alpha < 1, \\ O(1) & \alpha \geq 1. \end{cases} \tag{10}$$

In the PhD thesis [6], a fundamental new *optimization based method* was introduced:

> "L'objectif est alors d'essayer des conditions de transmission adéquates à la frontière de façon à minimiser l'erreur entre la solution du problème de transmission et celle de Navier Stokes complet dans tout le domaine."

The new idea is to find coupling conditions s.t. $||u - u_{\text{approx}}|| \longrightarrow \min$! Based on absorbing boundary condition techniques, this gives variational coupling conditions for our model problem, and non-variational ones for the inverse flow direction ($a < 0$). We introduced in [9, 10] a method based on the factorization $-\nu(\partial_x - \lambda_2)(\partial_x - \lambda_1)u = 0$, $\lambda_1 \geq 0$, $\lambda_2 \leq 0$. The idea consists in first solving the modified advection equation $u'_{\text{ma}} - \lambda_1 u_{\text{ma}} = 0$ on $(\gamma, 1)$ with the boundary condition $u_{\text{ma}}(1) = g$ where g is an approximation at order m of a function of $u(1)$ and $u'(1)$. We solve then the advection-diffusion equation with the boundary condition $(-\nu(u^{\text{fact}}_{\text{ad}})' + \nu\lambda_2 u^{\text{fact}}_{\text{ad}})(\gamma) = au_{\text{ma}}(\gamma)$ and we obtain the following error estimate:

Proposition 6 (Factorization method) *For $\gamma = O(\nu^\alpha)$, $\alpha \geq 0$,*

$$\|u - u^{\text{fact}}_{\text{ad}}\|_{L^\infty(0,\gamma)} = \begin{cases} O(\nu^m) & 0 \leq \alpha < 1, \\ O(\nu^{m+\alpha-1}) & \alpha \geq 1, \end{cases}$$

A further technique using partition of unity methods can be found in [2].

References

1. Brezzi, F., Canuto, C., and Russo, A. A self-adaptive formulation for the euler/navier-stokes coupling. *Computer methods in applied mechanics and engineering* **73**(3), 317–330 (1989).
2. Ciaramella, G. and Gander, M. J. Partition of unity methods for heterogeneous domain decomposition. In: *International Conference on Domain Decomposition Methods XXIV*, 177–186. Springer (2017).
3. Ciaramella, G. and Gander, M. J. *Iterative methods and preconditioners for systems of linear equations*. SIAM (2022).
4. Deparis, S., Discacciati, M., Fourestey, G., and Quarteroni, A. Heterogeneous domain decomposition methods for fluid-structure interaction problems. In: *Domain decomposition methods in science and engineering XVI*, 41–52. Springer (2007).
5. Dinh, Q., Periaux, J., Terrasson, G., and Glowinski, R. On the coupling of incompressible viscous flows and incompressible potential flows via domain decomposition. In: *Tenth International Conference on Numerical Methods in Fluid Dynamics*, 229–234. Springer (1986).
6. Dubach, E. *Contribution à la Résolution des Équations fluides en domaine non borné*. Ph.D. thesis, Université Paris 13 (1993).
7. Gander, M. J. Optimized Schwarz methods. *SIAM Journal on Numerical Analysis* **44**(2), 699–731 (2006).
8. Gander, M. J. and Dubois, O. Optimized Schwarz methods for a diffusion problem with discontinuous coefficient. *Numerical Algorithms* **69**(1), 109–144 (2015).
9. Gander, M. J., Halpern, L., and Martin, V. A new algorithm based on factorization for heterogeneous domain decomposition. *Numerical Algorithms* **73**(1), 167–195 (2016).
10. Gander, M. J., Halpern, L., and Martin, V. Multiscale analysis of heterogeneous domain decomposition methods for time-dependent advection–reaction–diffusion problems. *Journal of Computational and Applied Mathematics* **344**, 904–924 (2018).
11. Gander, M. J., Masson, R., and Vanzan, T. A numerical algorithm based on probing to find optimized transmission conditions. In: *International Conference on Domain Decomposition Methods XXVI* (2021).
12. Gander, M. J. and Tu, X. On the origins of iterative substructuring methods. In: *Domain Decomposition Methods in Science and Engineering XXI*, 597–605. Springer (2014).
13. Gander, M. J. and Vanzan, T. Heterogeneous optimized Schwarz methods for coupling Helmholtz and Laplace equations. In: *International Conference on Domain Decomposition Methods XXIV*, 311–320. Springer (2017).
14. Gander, M. J. and Vanzan, T. Optimized Schwarz methods for advection diffusion equations in bounded domains. In: *European Conference on Numerical Mathematics and Advanced Applications*, 921–929. Springer (2017).
15. Gander, M. J. and Vanzan, T. Heterogeneous optimized Schwarz methods for second order elliptic PDEs. *SIAM Journal on Scientific Computing* **41**(4), A2329–A2354 (2019).
16. Gander, M. J. and Vanzan, T. On the derivation of optimized transmission conditions for the Stokes-Darcy coupling. In: *International Conference on Domain Decomposition Methods XXV*, 491–498. Springer (2020).
17. Gastaldi, F. and Quarteroni, A. On the coupling of hyperbolic and parabolic systems: Analytical and numerical approach. In: *Proceedings of the Third German-Italian Symposium Applications of Mathematics in Industry and Technology*, 123–165. Springer (1989).
18. Lions, P.-L. On the Schwarz alternating method III: a variant for nonoverlapping subdomains. In: *Third international symposium on domain decomposition methods for partial differential equations*, vol. 6, 202–223. SIAM Philadelphia (1990).
19. Przemieniecki, J. S. Matrix structural analysis of substructures. *AIAA Journal* **1**(1), 138–147 (1963).
20. Schwarz, H. A. Über einen Grenzübergang durch alternierendes Verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich* **15**, 272–286 (1870).

Substructuring of Arbitrary Domain Decomposition Methods

Martin J. Gander and Frédéric Nataf

1 History of substructuring in domain decomposition

Substructuring domain decomposition methods and iterative substructuring methods referred originally to a few specific methods, see e.g. [13]. The purpose of this note is to briefly present this historical development, and then to show that in fact all domain decomposition methods with exact subdomain solves can be written in substructured form, see e.g. [4] for two level- and [1] for non-linear Schwarz methods, and this can be beneficial for the run-time of domain decomposition methods when Krylov acceleration is used, because the memory requirements are drastically reduced [3].

In Civil Engineering, HARDY CROSS introduced in 1930 an interesting iterative method for solving structural problems [5], see Figure 1 for the physical intuition he had. The unknowns in the method are the moments m_j at joints, and the method corresponds to a Gauss-Seidel iteration to update one moment after the other, e.g.

> To the mathematically inclined the method will appear as one of solving a series of normal simultaneous equations by successive approximation. From an engineering viewpoint it seems simpler and more useful to think of the solution as if it were a physical occurrence. The beams are loaded or otherwise distorted while the joints are held against rotation; one joint is then allowed to rotate with accompanying distribution of the unbalanced moment at that joint and the resulting moments are carried over to the adjacent joints; then another joint is allowed to rotate while the others are held against rotation; and the process is repeated until all the joints are "eased down" into equilibrium.

Fig. 1 The Hardy Cross method from 1932.

Martin J. Gander
Section de Mathématiques, Université de Genève, e-mail: martin.gander@unige.ch

Frédéric Nataf
Laboratoire J.L. Lions, Université Pierre et Marie Curie, e-mail: nataf@ljll.math.upmc.fr

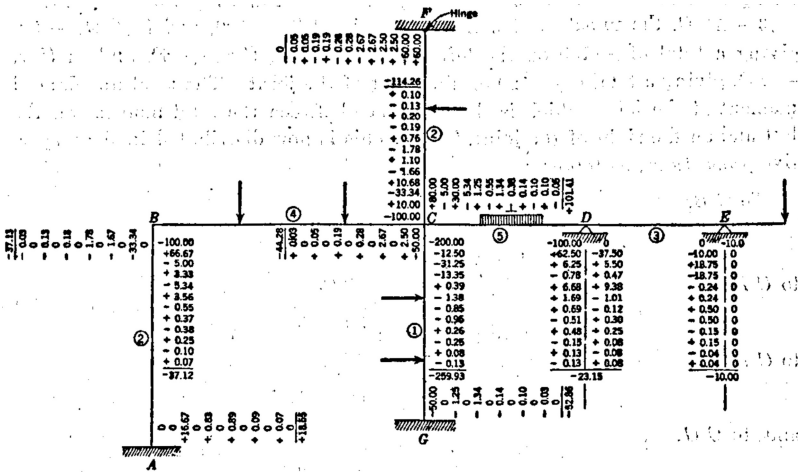


Fig. 2 Example of Hardy Cross from 1932.

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} \Longrightarrow \begin{bmatrix} A_{11} & \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} m_1^{n+1} \\ m_2^{n+1} \end{bmatrix} = \begin{bmatrix} -A_{12} & \\ & \end{bmatrix} \begin{bmatrix} m_1^{n} \\ m_2^{n} \end{bmatrix} + \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}, \quad (1)$$

if there were only two moments in the system. In Figure 2, we show an original example of Hardy Cross with many moments, and how he computed the corrections. He starts with initial moment estimates, e.g. 0 at A, $(0,-100)$ at B etc., and then computes in alternating fashion how moments at joints have to be updated until convergence. The moment corrections are tabulated, and then summed. The Hardy Cross method is therefore an iterative method, and one has to know how beams (subdomains) react to loads to execute it, the beams themselves are not simulated.

In Aerospace Engineering, Janusz Przemieniecki introduced in 1963 a substructuring method where now also the substructures (subdomains) must be simulated [12], see also Figure 3:

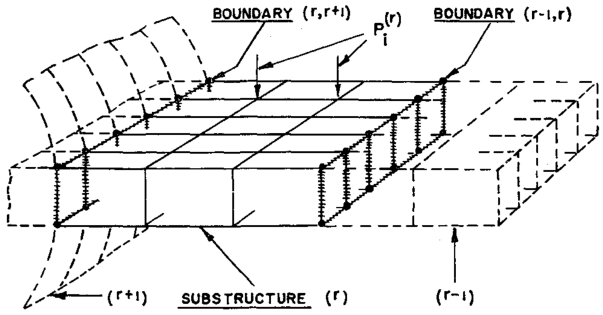


Fig. 3 Substructures of Przemieniecki from 1963.

> "The necessity for dividing a structure into substructures arises either from the requirement that different types of analysis have to be used on different components, or because the capacity of the digital computer is not adequate to cope with the analysis of the complete structure."

We see that the motivation is now quite different from Hardy Cross, including local solvers and distributed computing, and Przemieniecki describes his methods as follows:

> "In the present method each substructure is first analyzed separately, assuming that all common boundaries with adjacent substructures are completely fixed: these boundaries are then relaxed simultaneously and the actual boundary displacements are determined from the equations of equilibrium of forces at the boundary joints. The substructures are then analyzed separately again under the action of specified external loading and the previously determined boundary displacements."

In the notation of Przemieniecki, from the finite element system for the entire structure $KU = P$, unknowns are reordered into interior subdomain unknowns ('i'), and interface unknowns ('b' for 'boundary'),

$$\begin{bmatrix} K_{bb} & K_{bi} \\ K_{ib} & K_{ii} \end{bmatrix} \begin{bmatrix} U_b \\ U_i \end{bmatrix} = \begin{bmatrix} P_b \\ P_i \end{bmatrix}. \tag{2}$$

Fixing the interface unknowns U_b, one obtains for the interior unknowns $U_i = K_{ii}^{-1}(P_i - K_{ib}U_b)$. Introducing this into the equations for interface unknowns yields

$$(K_{bb} - K_{bi}K_{ii}^{-1}K_{ib})U_b = P_b - K_{bi}K_{ii}^{-1}P_i, \tag{3}$$

which is simply the Schur complement system. This system is then solved by a direct method by Przemieniecki, and once the interface values are known, the substructures can be computed, see the above quote. The method is therefore not iterative.

Wlodzimierz Proskurowski and Olof Widlund then introduced in 1976 a new Schur complement technique for capacitance matrix methods [11]:

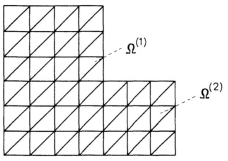
> "This new formulation leads to well-conditioned capacitance matrix equations which can be solved quite efficiently by the conjugate gradient method."

The key point here is that now the Schur complement system is solved by a Krylov method, not by a direct method, and thus the method is iterative.

Soon thereafter, in 1982, Max Dryja, the first winner of the Olof Widlund prize in domain decomposition, then introduced the seminal idea of preconditioning this Schur complement system in substructuring domain decomposition [7]:

> "The system is solved by generalized conjugate gradient method with $K^{1/2}$ as the preconditioning."

Max Dryja used an L-shaped domain Ω decomposed into two rectangles $\Omega_1 := (0, a_1) \times (0, b_2)$ and $\Omega_1 := (a_1, a_2) \times (0, b_1)$, see Figure 4, and the key matrix K here is the discrete Laplacian operator on the subdomain interface, whose square root allowed Max Dryja to get a condition number estimate which does not depend on the mesh size!



$$\gamma_0(K^{1/2}y,y)_{R^p} \leqq (Cy,y)_{R^p} \leqq \gamma_1(K^{1/2}y,y)_{R^p}$$

γ_0 *and* γ_1 *are the following constants*

$$\gamma_0 = r/(1+(1+r^2)^{1/2}), \qquad \gamma_1 = (1+2r(1+2\sigma))\,b_2/(4r\,b_1)$$
$$r = (32)^{1/2}(a_2-a_1)/b_1.$$

Fig. 4 Max Dryja inventing preconditioned iterative substructuring in 1982.

Bruno Després introduced in 1991 in his seminal PhD thesis [6] on what we call now optimized Schwarz methods a substructured formulation of a non-overlapping Schwarz method with Robin transmission conditions for Helmholtz problems; we quote directly from his PhD thesis in French:

$$\begin{aligned}
&(-\Delta-\omega^2)u_k^{n+1} = f && \text{dans } \Omega_k,\\
&(\frac{\partial}{\partial\nu_k}+i\omega)u_k^{n+1} = (-\frac{\partial}{\partial\nu_j}+i\omega)u_j^n && \text{sur } \Sigma_{kj}, \forall j,\\
&(\frac{\partial}{\partial\nu_k}+i\omega)u_k^{n+1} = 0 && \text{sur } \Gamma_k.
\end{aligned}$$

Notons $x = (x_{kj})$ le vecteur constitué des quantités de transmissions.

$$x_{kj} = \frac{1}{h}\hat{p}_{jk} + i\omega\hat{T}u_{jk}.$$

Le système précédent peut alors se mettre sous la forme

$$(I - \Pi T)x = g. \tag{7.36}$$

La matrice I est la matrice identité.

La matrice T permet de calculer les quantités de transmissions rentrantes à partir des quantités de transmissions sortantes et d'une équation de Helmholtz discrétisée à l'intérieur de chaque élément.

- **L'algorithme de décomposition de domaine qui résoud l'équation (7.36) n'est autre que $x^{n+1} = g + \Pi T x^n$.**

Equation (7.36) is the Schwarz substructured system, and the last equation is Schwarz for the first time written as an iteration on interface unknowns x^n only.

Only three years later, Frédéric Nataf, François Rogier and Eric De Sturler introduced substructured overlapping optimized Schwarz methods [10]. They considered a domain decomposition into strips, see Figure 5, and an optimized Schwarz method which can be made nil-potent, a groundbreaking result they prove in two ways, as seen directy from their manuscript:

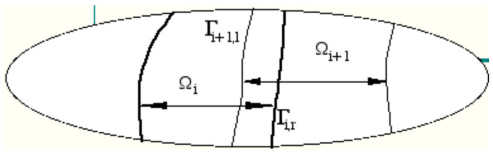


Fig. 5 Strip decomposition considered for substructured optimized Schwarz.

$$\begin{array}{c} \mathcal{L}(u_i^{n+1}) = f \text{ in } \Omega_i \\ (\frac{\partial}{\partial \vec{n}_{i,l}} - \Lambda_{i,l})(u_i^{n+1}) = (\frac{\partial}{\partial \vec{n}_{i,l}} - \Lambda_{i,l})(u_{i-1}^{n}) \text{ on } \Gamma_{i,l} \ (2 \le i \le N) \\ (\frac{\partial}{\partial \vec{n}_{i,r}} - \Lambda_{i,r})(u_i^{n+1}) = (\frac{\partial}{\partial \vec{n}_{i,r}} - \Lambda_{i,r})(u_{i+1}^{n}) \text{ on } \Gamma_{i,r} \ (1 \le i \le N-1) \\ \mathcal{C}(u_i^{n+1}) = g \text{ on } \partial\Omega \cap \partial\Omega_i \end{array} \tag{3}$$

Proposition 2.4 *The Schwarz algorithm (3) achieves convergence in N iterations, where N is the number of subdomains.*

We give two proofs. The first one is direct. The second one is based on an interpretation of (3) as an algorithm for unknowns defined on the boundaries of the subdomains. It is an introduction to the Schur method.

The second proof uses a substructured formulation of the Schwarz method, using the letter $h_{i,r \text{ or } l}$ for the interface unknowns, which gives in their manuscript

From (3), we have for $n \ge 1$

$$\begin{array}{ll} h_{2,l}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{2,l}} - \Lambda_{2,l})(S_1(h_{1,r}^n, 0, 0) + S_1(0, f, g)) \\ h_{3,l}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{3,l}} - \Lambda_{3,l})(S_2(h_{2,l}^n, 0, 0, 0) + S_2(0, h_{2,r}^n, 0, 0) + S_2(0, 0, f, g)) \\ & \vdots \\ h_{N,l}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{N,l}} - \Lambda_{N,l})(S_{N-1}(h_{N-1,l}^n, 0, 0, 0) + S_{N-1}(0, h_{N-1,r}^n, 0, 0) + S_{N-1}(0, 0, f, g)) \\ h_{N-1,r}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{N-1,r}} - \Lambda_{N-1,r})(S_N(h_{N,l}^n, 0, 0) + S_N(0, f, g)) \\ h_{N-2,r}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{N-2,r}} - \Lambda_{N-2,r})(S_{N-1}(h_{N-1,l}^n, 0, 0) + S_{N-1}(0, h_{N-1,r}^n, 0, 0) + S_{N-1}(0, 0, f, g)) \\ & \vdots \\ h_{1,r}^{n+1} & = (\frac{\partial}{\partial \vec{n}_{1,r}} - \Lambda_{1,r})(S_2(h_{2,l}^n, 0, 0) + S_2(0, h_{2,r}^n, 0, 0) + S_2(0, 0, f, g)) \end{array}$$

The authors even show the Schur complement like system to which the substructured overlapping Schwarz method at the limit corresponds, which is simply obtained by taking the limit in the above system as the iteration index n goes to infinity, see also (4) and (5) below. For a first substructured optimized Schwarz method with coarse correction, see [9].

The key ideas for substructuring a domain decomposition method are therefore to decompose, like in all domain decomposition methods, the domain of computation Ω into subdomains Ω_j, which were historically non-overlapping for substructuring. The domain decomposition iteration is then reformulated as an iteration on interface unknowns only, which were historically moments, then Dirichlet traces, and then Robin or more generalized traces. The resulting interface systems are solved by iteration, historically Gauss-Seidel, then by Conjugate Gradients, possibly with a preconditioner for Dirichlet coupling, and finally by Schwarz iterations.

2 General concepts and examples

Domain decomposition methods for linear problems can all be written as stationary iterations of the form (see e.g. [13, Section 1.3 and 1.4], and also the examples below)

$$\mathbf{u}^{n+1} = \mathbf{u}^n + M^{-1}(\mathbf{f} - A\mathbf{u}^n), \tag{4}$$

where $\mathbf{u}^n$ can be interface values or subdomain volume solutions, and M represents the domain decomposition method, which can contain also a coarse space. This

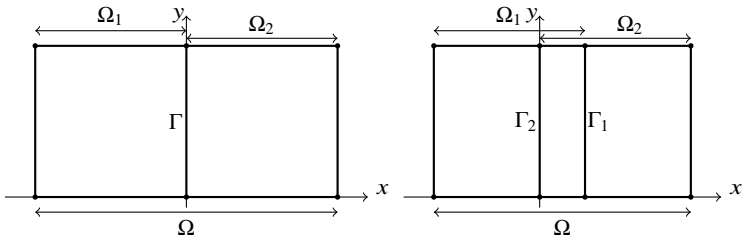


Fig. 6 Non-overlapping decomposition (left) and overlapping one (right).

iteration can be accelerated by a Krylov method: one then solves the DD iterative system (4) at the fixed point using a Krylov method. The system at the fixed point, i.e., when $n \to \infty$ and thus $\mathbf{u}^{n+1}$ and $\mathbf{u}^n$ cancel, is simply the preconditioned system

$$M^{-1}A\mathbf{u} = M^{-1}\mathbf{f}. \tag{5}$$

Solving this with a Krylov method gives much better convergence than the stationary domain decomposition iteration (4), since the error $\mathbf{e}^n := \mathbf{u} - \mathbf{u}^n$ for (4) satisfies

$$\mathbf{e}^{n+1} = \mathbf{e}^n - M^{-1}A\mathbf{e}^n = (I - M^{-1}A)^{n+1}\mathbf{e}^0, \tag{6}$$

and a Krylov method finds a much better residual polynomial than $(I - M^{-1}A)^{n+1}$,

$$\mathbf{e}^{n+1} = p_{n+1}(M^{-1}A)\mathbf{e}^0, \tag{7}$$

with $p_{n+1}(M^{-1}A)$ much smaller than $(I - M^{-1}A)^{n+1}$. For example Conjugate Gradients minimizes the energy norm $||\mathbf{e}^n||_{M^{-1/2}AM^{-1/2}}$, and GMRES minimizes the residual $||M^{-1}(\mathbf{f} - A\mathbf{u}^n)||_2$, see e.g. [2, Chapter 4.1].

Note that this same idea of acceleration also applies to non-linear problems: to accelerate a non-linear domain decomposition iteration $\mathbf{u}^{n+1} = G(\mathbf{u}^n)$ (or any non-linear fixed point iteration), one simply solves the fixed point equation $F(\mathbf{u}) := \mathbf{u} - G(\mathbf{u}) = 0$ by Newton's method, which is called non-linear preconditioning [8].

We now show several examples on how domain decomposition iterations can be substructured and then accelerated by Krylov methods. We start with the Dirichlet-Neumann method for a Poisson problem and two subomains, as shown in Figure 6 on the left. The method solves alternatingly Dirichlet and Neumann problems,

$$\begin{array}{llll} \Delta u_1^n = f & \text{in } \Omega_1, & \Delta u_2^n = f & \text{in } \Omega_2, \\ u_1^n = u_\Gamma^{n-1} & \text{on } \Gamma, & \partial_x u_2^n = \partial_x u_1^n & \text{on } \Gamma, \\ u_1^n = g & \text{on } \partial\Omega \cap \partial\Omega_1, & u_2^n = g & \text{on } \partial\Omega \cap \partial\Omega_2, \end{array} \tag{8}$$

and uses a relaxation to update the Dirichlet transmission condition,

$$u_\Gamma^n = \theta u_\Gamma^{n-1} + (1-\theta)u_2^n(\Gamma). \tag{9}$$

Using the Dirichlet to Neumann operator DtN, and the Neumann to Dirichlet operator NtD, we can write this method in substructured form, namely

$$u_\Gamma^n = \theta u_\Gamma^{n-1} + (1-\theta)\mathrm{NtD}_2(f, g, \mathrm{DtN}_1(f, g, u_\Gamma^{n-1})). \tag{10}$$

To use Krylov acceleration, we would solve this iteration at the fixed point using a Krylov method. By linearity the iteration at the fixed point yields the linear system

$$(I - \mathrm{NtD}_2(0, 0, \mathrm{DtN}_1(0, 0, \cdot))u_\Gamma = \mathrm{NtD}_2(f, g, \mathrm{DtN}_1(f, g, 0)). \tag{11}$$

Similarly, the Neumann-Neumann method for this example would be

$$\begin{array}{llll} \Delta u_i^n = f & \text{in } \Omega_i, & \Delta\psi_i^n = 0 & \text{in } \Omega_i, \\ u_i^n = u_\Gamma^{n-1} & \text{on } \Gamma, & \partial_{n_i}\psi_i^n = \partial_{n_1}u_1^n + \partial_{n_2}u_2^n & \text{on } \Gamma, \\ u_i^n = g & \text{on } \partial\Omega \cap \Omega_i, & \psi_i^n = 0 & \text{on } \partial\Omega \cap \Omega_i, \end{array} \tag{12}$$

with the interface updating relaxation

$$u_\Gamma^n = u_\Gamma^{n-1} - \theta(\psi_1(\Gamma) + \psi_2(\Gamma)). \tag{13}$$

This iteration can be written in the substructured form,

$$u_\Gamma^n = u_\Gamma^{n-1} - \theta \sum_{i=1}^{2} \mathrm{NtD}_i \left(\sum_{j=1}^{2} \mathrm{DtN}_j(f, g, u_\Gamma^{n-1}) \right), \tag{14}$$

and the solution can again be accelerated by solving with a Krylov method the Neumann-Neumann system at the fixed point,

$$\sum_{i=1}^{2} \mathrm{NtD}_i \left(\sum_{j=1}^{2} \mathrm{DtN}_j(0, 0, \cdot) \right) u_\Gamma = -\sum_{i=1}^{2} \mathrm{NtD}_i \left(\sum_{j=1}^{2} \mathrm{DtN}_j(f, g, 0) \right). \tag{15}$$

Finally, a Schwarz method for this problem and the overlapping decomposition in Figure 6 (right) would be

$$\begin{array}{llll} \Delta u_1^n = f & \text{in } \Omega_1, & \Delta u_2^n = f & \text{in } \Omega_2, \\ u_1^n = u_2^{n-1} & \text{on } \Gamma_1, & u_2^n = u_1^n & \text{on } \Gamma_2, \\ u_1^n = g & \text{on } \partial\Omega \cap \partial\Omega_1, & u_2^n = g & \text{on } \partial\Omega \cap \partial\Omega_2. \end{array}$$

To obtain a substructured formulation, we introduce the interface unknowns $\lambda^n := u_2^n|_{\Gamma_1}$, and then obtain the substructured iteration

$$\lambda^n = \mathrm{DD}_{21}(f, g, \mathrm{DD}_{12}(f, g, \lambda^{n-1})),$$

where DD_{ij} is the name for the subdomain solves and Dirichlet traceing. This iteration can again be accelerated by applying a Krylov method to the preconditioned substructured system

$$(I - \mathrm{DD}_{21}(0,0,\mathrm{DD}_{12}(0,0,\cdot))\lambda = \mathrm{DD}_{21}(f,g,\mathrm{DD}_{12}(f,g,0)).$$

For a more general substructured formulation of Schwarz methods, see [1, 3, 4].

3 Conclusions

We have seen that classical iterative domain decomposition methods can all be written in substructured form, and iterations in substructured form or in volume form are equivalent, provided exact subdomain solvers are used, see e.g. [1, 4]. Krylov acceleration in substructured form is cheaper with Krylov methods that do not have short recurrences (e.g. GMRES), because then the Krylov vectors to be stored are only of the dimension of the interfaces, not the volume unknowns [3]. It is easy to generate a substructured domain decomposition method from a volume one, one just has to apply restrictions and prolongations with interface data, see e.g. [1].

References

1. Chaouqui, F., Gander, M. J., Kumbhar, P. M., and Vanzan, T. Linear and nonlinear substructured Restricted Additive Schwarz iterations and preconditioning. *Numerical Algorithms* **91**(1), 81–107 (2022).
2. Ciaramella, G. and Gander, M. J. *Iterative methods and preconditioners for systems of linear equations*. SIAM (2022).
3. Ciaramella, G., Gander, M. J., Van Vriekingen, S., and Vanzan, T. A performance comparison of classical volume and new substructured one- and two-level Schwarz methods in PETSc. In: *Domain Decomposition Methods in Science and Engineering XXVII*. Springer (2023).
4. Ciaramella, G. and Vanzan, T. Substructured two-grid and multi-grid domain decomposition methods. *Numerical Algorithms* **91**(1), 413–448 (2022).
5. Cross, H. Analysis of continuous frames by distributing fixed-end moments. *Transactions of the American Society of Civil Engineers* **96**(1), 1–10 (1932).
6. Després, B. *Méthodes de décomposition de domaine pour la propagation d'ondes en régime harmonique. Le théorème de Borg pour l'équation de Hill vectorielle*. Ph.D. thesis, Paris 9 (1991).
7. Dryja, M. A capacitance matrix method for Dirichlet problem on polygon region. *Numerische Mathematik* **39**(1), 51–64 (1982).
8. Gander, M. J. On the origins of linear and non-linear preconditioning. In: *Domain Decomposition Methods in Science and Engineering XXIII*, 153–161. Springer (2017).
9. Japhet, C. *Conditions aux limites artificielles et décomposition de domaine: Méthode OO2 (Optimisé d'ordre 2). Application à la résolution de problèmes en mécanique des fluides*. Ph.D. thesis, CMAP, Ecole Polytechnique, Paris (1997).
10. Nataf, F., Rogier, F., and de Sturler, E. Optimal interface conditions for domain decomposition methods. Tech. rep., CMAP, Ecole Polytechnique, Paris (1994).
11. Proskurowski, W. and Widlund, O. On the numerical solution of Helmholtz's equation by the capacitance matrix method. *Mathematics of Computation* **30**(135), 433–468 (1976).
12. Przemieniecki, J. S. Matrix structural analysis of substructures. *AIAA Journal* **1**(1), 138–147 (1963).
13. Toselli, A. and Widlund, O. *Domain Decomposition Methods - Algorithms and Theory*, vol. 34. Springer (2004).

Spectral Q1-Based Coarse Spaces for Schwarz Methods

Martin J. Gander and Serge Van Criekingen

1 Introduction

The Q1 coarse space [7, 8] is based on coarse Q1 bilinear finite element functions on rectangular elements which are here the subdomains. Hence the coarse grid points are placed (in 2-D) around each cross point of the non-overlapping decomposition. It was studied by the authors in [9] and [10], together with several of its variants, in the context of the Restricted Additive Schwarz (RAS) method [4] with optimized Robin-type transmission conditions [11]. Encouraging numerical results were obtained in that the resulting method, implemented in PETSc [1, 2, 3], showed computing times competitive with multigrid approaches on a 2-D Laplace test case, for both symmetric and non-symmetric (i.e., with advection) problems. Among the different options invesitigated in [10], the so-called Half_Q1 (see also [6]) appeared most promising, in that it halves the coarse space dimension compared to Q1 by using a selected combination of its basis functions, while causing only a moderate increase in iteration count, resulting in our best observed computing times. We therefore pursue here the investigation around this Half_Q1 coarse space and, more generally, Q1-based spectral coarse spaces [5], that is, coarse spaces based on the study of the eigenvectors of the underlying iteration operator, in our case the RAS iteration operator. Note however that we here do not compute a spectrum specific to each problem (as for instance in [5]): we define our coarse spaces based on the observation of the eigenmodes of the non-overlapping symmetric Laplace test case and hope that the resulting method will apply succesfully to a broader set of problems, as was the

Martin J. Gander
University of Geneva, e-mail: martin.gander@unige.ch

Serge Van Criekingen
Institut du Développement et des Ressources en Informatique Scientifique (IDRIS), CNRS, Université Paris-Saclay, F-91403, Orsay, France and Université Paris-Saclay, UVSQ, CNRS, CEA, Maison de la Simulation, 91191, Gif-sur-Yvette, France, e-mail: serge.van.criekingen@idris.fr

case in [10] adding overlap and advection. Note that we here restrict our analysis to homogeneous Dirichlet boundary conditions.

As already pointed out in [10], the largest two eigenvalues (i.e. closest to 1 in modulus) of the RAS iteration operator for the 2-D symmetric Laplace model problem appear to be equal in modulus and of opposite signs, while the corresponding eigenvectors appear to be one continuous (for the positive eigenvalue) and one discontinuous (for the negative one). We display these eigenmodes[1] in Fig. 1 for various square domain decompositions in the algebraically non-overlapping case (RAS then reduces to Block Jacobi), as obtained using the `SLEPc` [12] eigenproblem companion package to `PETSc`, with the traditional 5-point finite difference discretization. These modes appear to be piecewise `Q1` functions.

The `Q1` basis functions at a cross point will be denoted by q_1, q_2, q_3, q_4 (i.e., bilinear with value 1 at the cross point and 0 at the other corners of the subdomain - see Fig. 2a), the four of them building up a "hat" around the cross point. (Note that we do not need to solve eigenproblems to use these `Q1` functions.) The `Half_Q1` coarse space is based on the observation of the 2×2 eigenmodes (Fig. 1a and 1b): these modes appear to be particular combinations of the `Q1` basis functions at the cross point, namely $q_1 + q_2 + q_3 + q_4$ (the "hat" itself) and $q_1 - q_2 + q_3 - q_4$. The `Half_Q1` coarse space is therefore obtained by taking these 2 combinations as basis functions, thus with 2 basis functions per cross point instead of 4 in the `Q1` case. This is equivalent to taking the combinations $q_1 + q_4$ and $q_2 + q_3$ at each cross point.

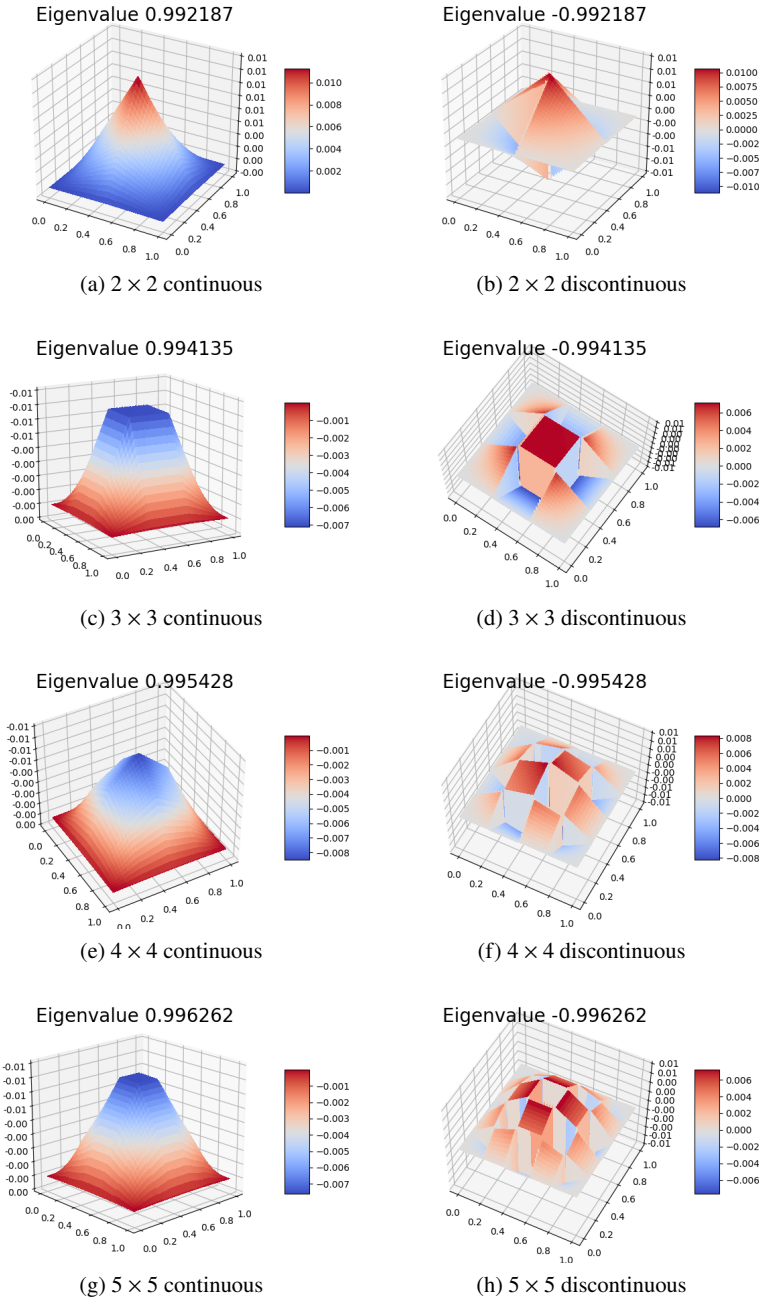
By construction, the `Half_Q1` space contains the first two eigenmodes of the non-overlapping RAS iteration operator in the case of a 2×2 decomposition. In turn, taking one of these first two RAS eigenmodes as initial guess of a coarse corrected (i.e., two-level) RAS iteration process, we obtain convergence at iteration 1 using the `Half_Q1` coarse space (with square subdomains and a non-overlapping decomposition). For more than 2×2 subdomains, convergence at iteration 1 does not hold, but it still holds (with the same restrictions) for square decompositions using the `Q1` coarse space, and it is moreover possible to define a `Half_Q1`$^+$ coarse space, larger than `Half_Q1` but smaller than `Q1`, so as to include the first two modes, i.e., so that this convergence at iteration 1 is verified. This will be described in section 2.

Another new `Q1`-based coarse space, named `Checkerboard`, is introduced in section 3. Based on the first two modes of the decomposition considered (not only the 2×2 one), it can be applied to non-square decompositions.

2 The `Half_Q1`$^+$ coarse space

The `Half_Q1`$^+$ coarse space is built by adding a minimal number of extra basis functions to the `Half_Q1` coarse space so as to contain the first two eigenmodes of the RAS iteration operator. It is meant to be smaller than the `Q1` coarse space.

[1] The problem solved here in thus $Gx = \lambda x$ where $G := I - \left(\sum_{j=1}^{J} \tilde{R}_j^T \, A_j^{-1} \, \tilde{R}_j\right) A$, and R_j are restriction operators to the J non-overlapping subdomains decomposing the global domain.



(a) 2×2 continuous (b) 2×2 discontinuous

(c) 3×3 continuous (d) 3×3 discontinuous

(e) 4×4 continuous (f) 4×4 discontinuous

(g) 5×5 continuous (h) 5×5 discontinuous

Fig. 1 Eigenmodes of the non-overlapping RAS iteration operator corresponding to the two largest eigenvalues in modulus for the 2×2 to 5×5 decompositions, for a global 256x256 fine mesh resolution.

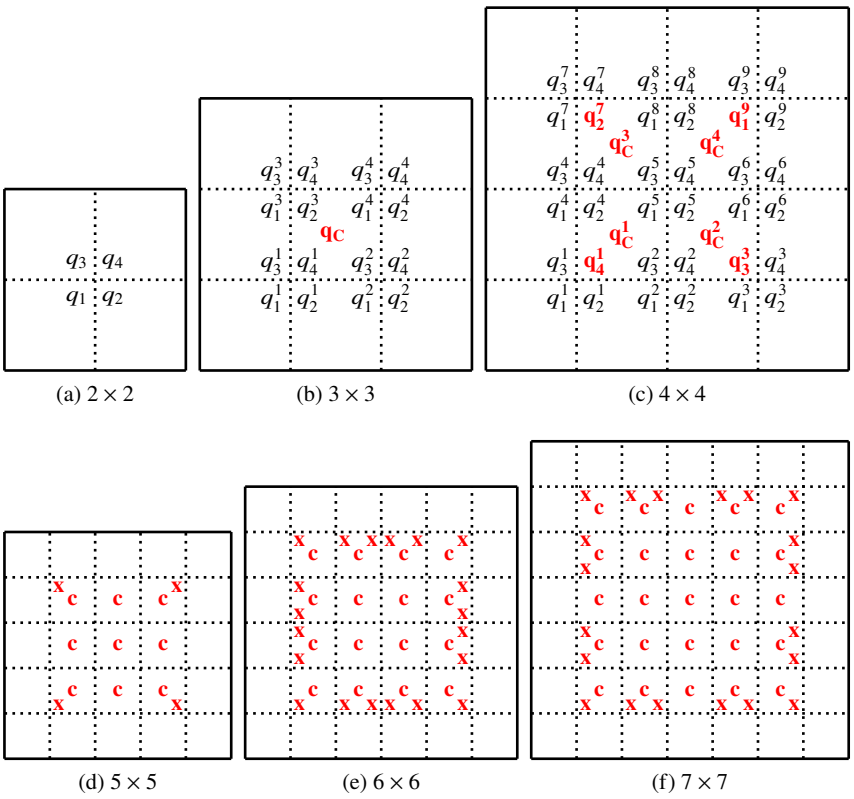
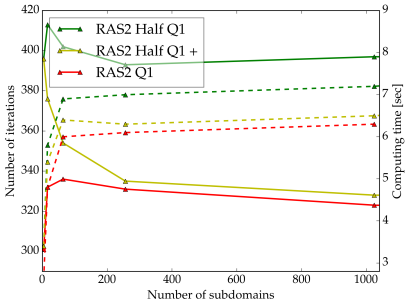


Fig. 2 In red, basis functions to be added to the `Half_Q1` coarse space to obtain the `Half_Q1`$^+$ one, for various decompositions. $q_1^i, q_2^i, q_3^i, q_4^i$ are the `Q1` basis functions at cross point i and $\mathbf{q_C}$ represents a constant function in the considered subdomain. For the 5×5 to 7×7 decompositions, only a schematic view is given, with c representing a constant and x a `Q1` basis function.

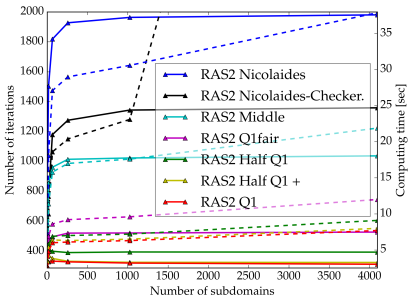
For the 2×2 decomposition, the `Half_Q1`$^+$ coarse space is the same as the `Half_Q1` one. This is not the case anymore for the 3×3 decomposition: starting from one of the first two modes of the RAS iteration operator (Figs. 1c and 1d), convergence of the `Half_Q1` coarse-corrected RAS iteration process is not obtained at iteration 1, while it is the case with `Q1`. But what is missing in `Half_Q1` to achieve convergence at iteration 1? Observing Figs. 1c and 1d, one can intuitively infer that adding a constant coarse function in the central subdomain to the `Half_Q1` coarse space will greatly improve convergence. Our numerical implementation showed that this is actually sufficient to obtain convergence at iteration 1. The `Half_Q1`$^+$ coarse space is thus obtained from `Half_Q1` by adding one single constant coarse function in the central subdomain ($\mathbf{q_C}$ in Fig. 2b) and is of size 9 (8 for `Half_Q1` and 16 for `Q1`).

For the 4×4 decomposition, the first two RAS modes are given in Figs. 1e and 1f. In this case, the minimal function set we found to add to `Half_Q1` to resolve the first two modes is made out of one constant coarse function on each inner subdomain as well as the extra `Q1` basis functions located at the four "inner corners" one subdomain away from the boundary, namely q_4^1, q_3^3, q_2^7, q_1^9 in Fig. 2c. Thus, for this decomposition, `Half_Q1`$^+$ is of size 26 (18 for `Half_Q1` and 36 for `Q1`).

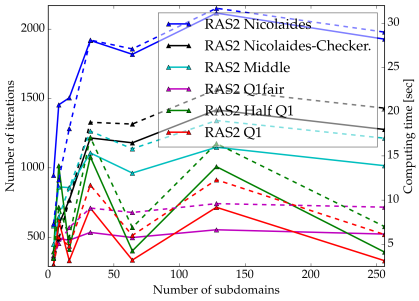
We pursued our investigations for larger $N \times N$ decompositions and observed that the extra basis functions to be added to the `Half_Q1` coarse space to build `Half_Q1`$^+$ remain of two types, namely constants on each non-boundary subdomain and extra `Q1` basis functions located one subdomain away from the boundary, as described schematically in Figs. 2d, 2e and 2f. Note that for N odd, the extra basis functions on the "middle" subdomain on each side (one subdomain away from the boundary) appear not to be needed (see Figs. 2d and 2f). However, these appear to be needed in the case $N = 11, 15, 19, 23, \ldots$ This was tested numericallly up to $N = 50$, i.e., 2500 subdomains. Note that, while the size of `Q1` and `Half_Q1` asymptotically grow as $4N^2$ and $2N^2$ respectively, the size of `Half_Q1`$^+$ grows as $3N^2$.



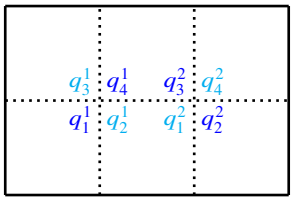
(a) Square decompositions up to 32×32.



(b) Square decompositions up to 64×64.

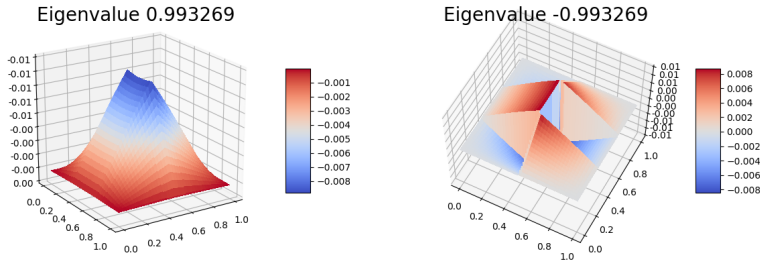


(c) With 4×2, 8×4 and 16×8 decompositions.



(d) 3×2 `Checkerboard`

Fig. 3 (a) to (c): Weak scaling experiment for overlapping RAS2 (256x256 fine mesh per subdomain) for various decompositions. Solid: number of iterations, dashed: computing times. (d): 3×2 `Checkerboard` coarse basis function definition.



(a) 3×2 continuous (b) 3×2 discontinuous

Fig. 4 Eigenmodes of the non-overlapping RAS iteration operator corresponding to the two largest eigenvalues in modulus in the case of 3×2 subdomains, for a global 256x256 fine mesh resolution.

Once defined, the `Half_Q1`$^+$ coarse space can be used in a general context: weak scaling experiment results for RAS with overlap 1 (in the `PETSc` sense, i.e, algebraic overlap of 2) are given in Fig. 3a. Starting from a random initial guess, the number of iterations and computing times necessary to bring the relative tolerance below 1.e-8 are given. In terms of iterations, `Half_Q1`$^+$ tends to behave asymptotically like `Q1`, while using only $3N^2$ coarse functions instead of $4N^2$. In terms of computing time, `Half_Q1`$^+$ yields scalable timings very close to `Q1` and better than `Half_Q1`.

Note that the `Half_Q1`$^+$ coarse space is not defined in the general rectangular case since then even the `Q1` coarse space does not resolve the first two non-overlapping RAS eigenmodes. This is the case for instance for the 3×2 subdomain case whose eigenmodes are depicted in Figs. 4a and 4b. A close observation of these plots reveals that what appears as a horizontal edge at $y = .5$ (assuming three subdomains in x and two in y) is in fact slightly curved, giving an intuitive explanation to the non-inclusion of these modes into the `Q1` coarse space.

3 The `Checkerboard` coarse space

The `Checkerboard` coarse space is based on the first two modes of the decomposition considered, not only the 2×2 one as in the `Half_Q1` case. It is defined for square and rectangular decompositions, and contains 2 modes. For the 3×2 case and as illustrated in Fig. 3d these two modes are $q_1^1 + q_4^1 + q_3^2 + q_2^2$ and $q_3^1 + q_2^1 + q_1^2 + q_4^2$. This definition comes from the observation of Fig. 4a and 4b. Starting from one of these two modes, we observed that the `Checkerboard` coarse space gives the exact same iterates as `Half_Q1` but with 2 coarse functions instead of 4.

For the 3×3 case, the two `Checkerboard` modes are defined to be (using the numbering in Fig. 2b and not including the constants) $q_1^1+q_2^2+q_3^3+q_4^4+q_4^1+q_3^2+q_2^3+q_1^4$ for the first mode and the sum of the 8 other q_j^i for the second mode. This comes from the observation of Figs. 1c and 1d. It again produces the same iterates as `Half_Q1` but with 2 coarse functions instead of 8.

For the 4×4 case, the observation of Figs. 1e and 1f leads us to define the first `Checkerboard` coarse basis functions as (using the numbering in Fig. 2c and grouping the q_j^i functions by subdomain) $q_1^1 + (q_2^2 + q_1^3) + (q_4^1 + q_3^2 + q_2^4 + q_1^5) + (q_4^3 + q_2^6) + (q_3^4 + q_1^7) + (q_4^5 + q_3^6 + q_2^8 + q_1^9) + (q_4^7 + q_3^8) + q_4^9$ and the second one as made out of the other q_j^i. Here these two `Checkerboard` coarse functions do not produce the same iterates as the `Half_Q1` coarse functions (18 in this case). This is not surprising since no scalability can be achieved with only two coarse functions.

Nevertheless, it is still possible to obtain a scalable - at least in terms of iterations - two-level method based on the two `Checkerboard` functions, by adding the constant function in each subdomain, yielding a coarse space of size N^2+2 that will be named `Nicolaides-Checkerboard` since it is the same as the Nicolaides coarse space [13] but with two extra basis functions. Fig. 3b presents the same weak scaling experiment as Fig. 3a, but extended up to 4096 cores and also to other coarse spaces defined in [10], namely `Middle` (classical coarse space with one coarse point in the middle of each subdomain) and `Q1_fair` (same number of coarse mesh points as `Q1`, but equally distributed in space). The two extra `Checkerboard` functions yield a major improvement to the Nicolaides coarse space in terms of number of iterations, and this improvement is scalable in that it remains as effective when increasing the number of subdomains. In terms of computing time, the new coarse space appears not scalable above 1024 cores: the coarse solve (performed here with a parallel direct solver) remains a challenge, the two extra functions implying the whole domain.

Fig. 3c includes non-square decompositions up to 16 subdomains in one direction. These appear to require more iterations (and computing time) than their square counterparts. For the `Half_Q1` coarse space, this can be related to the absence of affine modes for non-square subdomains pointed out in [6].

4 Conclusions

We introduced two new `Q1`-based coarse spaces. Firstly, the `Half_Q1`$^+$ coarse space is built from `Half_Q1` (thus from the first two RAS modes of the 2×2 decomposition) so as to contain the first two RAS modes of the considered (square) decomposition while using a minimal set of coarse functions in order to remain smaller than `Q1`. It was shown to behave asymptotically like `Q1` in terms of number of iterations, but using $3N^2$ coarse functions instead of $4N^2$. Secondly, the `Checkerboard` coarse space is built as the first two RAS modes of the decomposition considered and can be defined for square and rectangular decompositions. Combined with Nicolaides into the `Nicolaides-Checkerboard` coarse space, it yields a significant improvement in terms of number of iterations. Its scalability in time is still under investigation.

Acknowledgements This work was performed using HPC resources from GENCI-IDRIS.

References

1. Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E. M., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., Kong, F., Kruger, S., May, D. A., McInnes, L. C., Mills, R. T., Mitchell, L., Munson, T., Roman, J. E., Rupp, K., Sanan, P., Sarich, J., Smith, B. F., Zampini, S., Zhang, H., Zhang, H., and Zhang, J. PETSc Web page. https://petsc.org/ (2022).
2. Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E. M., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., Kong, F., Kruger, S., May, D. A., McInnes, L. C., Mills, R. T., Mitchell, L., Munson, T., Roman, J. E., Rupp, K., Sanan, P., Sarich, J., Smith, B. F., Zampini, S., Zhang, H., Zhang, H., and Zhang, J. PETSc/TAO users manual. Tech. Rep. ANL-21/39 - Revision 3.18, Argonne National Laboratory (2022).
3. Balay, S., Gropp, W., McInnes, L. C., and Smith, B. Efficient management of parallelism in object oriented numerical software libraries. In: Arge, E., Bruaset, A. M., and Langtangen, H. P. (eds.), *Modern Software Tools in Scientific Computing*, 163–202. Birkhäuser Press (1997).
4. Cai, X.-C. and Sarkis, M. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comp.* **21**(2), 239–247 (1999).
5. Ciaramella, G. and Vanzan, T. On the asymptotic optimality of spectral coarse spaces. In: *Domain Decomposition Methods in Science and Engineering XXVI*, Lecture Notes in Computational Science and Engineering, 181–188. Springer-Verlag (2021).
6. Cuvelier, F., Gander, M. J., and Halpern, L. Fundamental coarse space components for schwarz methods with crosspoints. In: *Domain Decomposition Methods in Science and Engineering XXVI*, Lecture Notes in Computational Science and Engineering, 39–50. Springer-Verlag (2021).
7. Dubois, O., Gander, M., Loisel, S., St-Cyr, A., and Szyld, D. The optimized Schwarz methods with a coarse grid correction. *SIAM J. Sci. Comp.* **34**(1), A421–A458 (2012).
8. Gander, M., Halpern, L., and Santugini, K. A new coarse grid correction for RAS/AS. In: *Domain Decomposition Methods in Science and Engineering XXI*, Lecture Notes in Computational Science and Engineering, 275–284. Springer-Verlag (2014).
9. Gander, M. and Van Criekingen, S. New coarse corrections for restricted additive Schwarz using PETSc. In: *Domain Decomposition Methods in Science and Engineering XXV*, Lecture Notes in Computational Science and Engineering, 483–490. Springer-Verlag (2019).
10. Gander, M. and Van Criekingen, S. Coarse corrections for Schwarz methods for symmetric and non-symmetric problems. In: *Domain Decomposition Methods in Science and Engineering XXVI*, Lecture Notes in Computational Science and Engineering, 589–596. Springer-Verlag (2021).
11. Gander, M. J. Optimized Schwarz methods. *SIAM J. Numer. Anal.* **44**(2), 669–731 (2006).
12. Hernandez, V., Roman, J. E., and Vidal, V. SLEPc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Software* **31**(3), 351–362 (2005).
13. Nicolaides, R. A. Deflation of conjugate gradients with applications to boundary value problems. *SIAM Journal on Numerical Analysis* **24**, 355–365 (1987).

Optimized Schwarz Methods for Stokes-Darcy Flows: the Brinkman Equations

Martin J. Gander, Yiying Wang, and Yingxiang Xu

1 Introduction

The Brinkman equations model a combination of Darcy's law and the Navier-Stokes equations, see [3]. They describe the incompressible viscous flow of a fluid in complex porous media with a high-contrast permeability coefficient such that the flow is dominated by Darcy in porous media regions and by Stokes in fluid regions, which naturally defines a decomposition of the domain by the physics of the problem, see for example [4, 7].

Let $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$) be a bounded convex domain with Lipschitz boundary $\partial\Omega$; the Brinkman model for the unknown velocity vector function $\mathbf{u}\colon \overline{\Omega} \to \mathbb{R}^d$, the scalar pressure function $p : \overline{\Omega} \to \mathbb{R}$ and some given force term $\mathbf{f}\colon \overline{\Omega} \to \mathbb{R}^d$ is

$$\begin{aligned} -\nu\Delta\mathbf{u} + \frac{\nu}{\kappa}\mathbf{u} + \nabla p &= \mathbf{f} \quad \text{in } \Omega, \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{in } \Omega, \\ \mathbf{u} &= \mathbf{g} \quad \text{on } \partial\Omega, \end{aligned} \tag{1}$$

where ν denotes the viscosity and κ is the permeability coefficient of the porous media which occupies the domain Ω.

We present here a new non-overlapping Schwarz method [5, 10] for solving the Brinkman equation (1) with fully-coupled Robin-like transmission conditions [1, 12]. We derive a general expression for the iteration operator, and study the corresponding min-max problems for local approximations to optimize performance using

Martin J. Gander
Section de Mathématiques, Université de Genève, Switzerland, e-mail: martin.gander@unige.ch

Yiying Wang
School of Mathematics, Jilin University, P. R. China, e-mail: yiyingw20@mails.jlu.edu.cn

Yingxiang Xu
School of Mathematics and Statistics, Northeast Normal University, P. R. China,
e-mail: yxxu@nenu.edu.cn

asymptotic analysis, which we also illustrate with numerical results. For the sake of simplicity, we will consider the 2-D case with $\mathbf{g} = 0$ and $\nu = 1$ (we can always scale the solution with ν) in two spatial dimensions.

2 Iteration operator of a non-overlapping Schwarz algorithm

We consider (1) on a bounded domain Ω in $\mathbb{R}^2$ formed by two non overlapping subregions: the porous medium Ω_1, the fluid domain Ω_2, separated by an interface Γ. We split the domain Ω into two subdomains determined by the porous media: $\overline{\Omega} = \overline{\Omega}_1 \cup \overline{\Omega}_2$, and the permeability coefficient κ is a corresponding piecewise constant function. On the two subdomains Ω_j ($j = 1, 2$), we use a parallel Schwarz iteration with generic Robin-like transmission conditions,

$$\begin{aligned} -\Delta \mathbf{u}_j^n + \nabla p_j^n + \kappa_j^{-1}\mathbf{u}_j^n &= \mathbf{f} \qquad \text{in } \Omega_j, \\ \nabla \cdot \mathbf{u}_j^n &= 0 \qquad \text{in } \Omega_j, \\ \mathbf{u}_j^n &= \mathbf{g} \qquad \text{on } \partial\Omega_j \setminus \Gamma, \\ \sigma_j^n \cdot \mathbf{n_j} + S_j \mathbf{u}_j^n &= \sigma_{3-j}^{n-1} \cdot \mathbf{n_j} + S_j \mathbf{u}_{3-j}^{n-1} \quad \text{on } \Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2, \end{aligned} \tag{2}$$

where $\sigma_j^n := \nabla \mathbf{u}_j^n - p_j^n I$ is the stress tensor [4, 7, 2] in domain Ω_j (I represents the 2×2 identity matrix), $\mathbf{n}_j$ is the outward normal vector, S_j is a general 2×2-matrix of linear operators, and n is the iteration index of the Schwarz algorithm.

Subdomain solutions: In order to study solutions of (2), we consider a model problem on the infinite plane $\Omega = \mathbb{R}^2$, with the two subdomains $\Omega_1 = \mathbb{R} \times (-\infty, 0)$, and $\Omega_2 = \mathbb{R} \times (0, \infty)$.

We use the Fourier transform in the x (horizontal) variable for the error equations of (2), i.e. $\mathbf{f} = \mathbf{g} = 0$. In Fourier space, the PDE in Ω_j becomes an ODE in y (for each fixed frequency k),

$$-\begin{pmatrix} -k^2\hat{\mathbf{e}}_j^n + \dfrac{d^2\hat{\mathbf{e}}_j^n}{dy^2} \end{pmatrix} + \begin{pmatrix} ik\hat{\eta}_j^n \\ \dfrac{d\hat{\eta}_j^n}{dy} \end{pmatrix} + \kappa_j^{-1}\hat{\mathbf{e}}_j^n = 0 \qquad \text{in } \Omega_j, \tag{3}$$

$$ik\hat{e}_{j,1}^n + \frac{d\hat{e}_{j,2}^n}{dy} = 0 \qquad \text{in } \Omega_j, \tag{4}$$

$$\hat{\mathbf{e}}_j^n \to 0 \qquad \text{when } |y| \to \infty, \tag{5}$$

$$\hat{\sigma}_j^n \cdot \mathbf{n}_j + \hat{S}_j \hat{\mathbf{e}}_j^n = \hat{\sigma}_{j'}^{n-1} \cdot \mathbf{n}_j + \hat{S}_j \hat{\mathbf{e}}_{j'}^{n-1} \text{ on } \Gamma,\ j' = 3 - j, \tag{6}$$

where $\hat{\eta}_j := \hat{p}|_{\Omega_j} - \hat{p}_j^n$ and $\hat{\mathbf{e}}_j^n := \hat{\mathbf{u}}|_{\Omega_j} - \hat{\mathbf{u}}_j^n = (\hat{e}_{j,1}^n, \hat{e}_{j,2}^n)^T$. Here $\hat{e}_{j,1}^n$ and $\hat{e}_{j,2}^n$ denote the horizontal component and the vertical component of $\hat{\mathbf{e}}_j^n$.

As in [8] and [6], we seek solutions using the ansatz

$$\mathbf{E}_j^n := \begin{pmatrix} \hat{\mathbf{e}}_j^n \\ \hat{\eta}_j^n \end{pmatrix}(y) = \Phi_j^n e^{\xi y}.$$

This leads to a system for Φ_j^n, namely

$$\begin{pmatrix} k^2 + \kappa_j^{-1} - \xi^2 & 0 & ik \\ 0 & k^2 + \kappa_j^{-1} - \xi^2 & \xi \\ ik & \xi & 0 \end{pmatrix} \Phi_j^n = 0. \tag{7}$$

In order to get a non-trivial solution to system (7), a necessary and sufficient condition is that the matrix is singular, which leads to four possible values for ξ, $\xi_1 = |k|$, $\xi_2 = \lambda_1$, $\xi_3 = -|k|$, and $\xi_4 = -\lambda_2$, with $\lambda_j := \sqrt{k^2 + \kappa_j^{-1}}$.

The solutions of (7) are linear combinations of four terms,

$$\mathbf{E}_j^n = \sum_{m=1}^{4} \boldsymbol{\gamma}_{j,m}^n \Phi_m e^{\xi_m y},$$

where $((\Phi_m)_{1 \le m \ge 4})$ are the eigenvectors (corresponding to the eigenvalue 0), associated with each of the ξ_m,

$$\Phi_1 = \begin{pmatrix} -ik \\ -|k| \\ \kappa_1^{-1} \end{pmatrix}, \quad \Phi_2 = \begin{pmatrix} \lambda_1 \\ -ik \\ 0 \end{pmatrix}, \quad \Phi_3 = \begin{pmatrix} -ik \\ |k| \\ \kappa_2^{-1} \end{pmatrix}, \quad \Phi_4 = \begin{pmatrix} \lambda_2 \\ ik \\ 0 \end{pmatrix}.$$

Due to the condition (5), $\xi_1, \xi_2 \ge 0$ and $\xi_3, \xi_4 \le 0$, only two terms are possible in the expression of $\mathbf{E}_j^n$ in each of the subdomain Ω_j, and we obtain for the subdomain errors

$$\mathbf{E}_1^n(y) = \sum_{m=1}^{2} \boldsymbol{\gamma}_{1,m}^n \Phi_m e^{\xi_m y}, \qquad \mathbf{E}_2^n(y) = \sum_{m=3}^{4} \boldsymbol{\gamma}_{2,m}^n \Phi_m e^{\xi_m y}. \tag{8}$$

Iteration operator: To obtain the iteration operator, we need to apply the transmission conditions (6) to (8). Using the horizontal component of equation (3), we can simplify the error in the pressure as

$$\hat{\eta}_j^n = \frac{i}{k}((k^2 + \kappa_j^{-1})\hat{\mathbf{e}}_j^n - \frac{d^2\hat{\mathbf{e}}_j^n}{dy^2}) \cdot (1,0)^T. \tag{9}$$

Inserting the gradient of $\hat{\mathbf{e}}_j^n$ and (9) into the transmission condition

$$\nabla\hat{\mathbf{e}}_j^n \cdot \mathbf{n}_j - \hat{\eta}_j^n I \cdot \mathbf{n}_j + \hat{S}_j\hat{\mathbf{e}}_j^n = \nabla\hat{\mathbf{e}}_{j'}^{n-1} \cdot \mathbf{n}_j - \hat{\eta}_{j'}^{n-1} I \cdot \mathbf{n}_j + \hat{S}_j\hat{\mathbf{e}}_{j'}^{n-1},$$

and using that the normal vector $\mathbf{n}_j = (0, (-1)^{j-1})^T$ at the interface, we obtain

$$\begin{aligned} M\frac{d^2\hat{\mathbf{e}}_1^n}{dy^2} + \frac{d\hat{\mathbf{e}}_1^n}{dy} + P_1\hat{\mathbf{e}}_1^n + \hat{S}_1\hat{\mathbf{e}}_1^n &= M\frac{d^2\hat{\mathbf{e}}_2^{n-1}}{dy^2} + \frac{d\hat{\mathbf{e}}_2^{n-1}}{dy} + P_2\hat{\mathbf{e}}_2^{n-1} + \hat{S}_1\hat{\mathbf{e}}_2^{n-1}, \\ -M\frac{d^2\hat{\mathbf{e}}_2^n}{dy^2} - \frac{d\hat{\mathbf{e}}_2^n}{dy} - P_2\hat{\mathbf{e}}_2^n + \hat{S}_2\hat{\mathbf{e}}_2^n &= -M\frac{d^2\hat{\mathbf{e}}_1^{n-1}}{dy^2} - \frac{d\hat{\mathbf{e}}_1^{n-1}}{dy} - P_1\hat{\mathbf{e}}_1^{n-1} + \hat{S}_1\hat{\mathbf{e}}_1^{n-1}, \end{aligned} \tag{10}$$

with $M := \begin{pmatrix} 0 & 0 \\ \frac{i}{k} & 0 \end{pmatrix}$, and $P_j := \begin{pmatrix} 0 & 0 \\ \frac{\lambda_j^2}{ik} & 0 \end{pmatrix}$, $j = 1, 2$. Using (8), we then get

$$\hat{\mathbf{e}}_1^n(y) = M_{12}e^{\xi_{12}y}\boldsymbol{\gamma}_{12}^n\,, \quad \hat{\mathbf{e}}_2^n(y) = M_{34}e^{\xi_{34}y}\boldsymbol{\gamma}_{34}^n\,, \text{ with}$$

$$M_{12} := \begin{pmatrix} -ik & \lambda_1 \\ -|k| & -ik \end{pmatrix}, \quad e^{\xi_{12}y} := \begin{pmatrix} e^{\xi_1 y} & 0 \\ 0 & e^{\xi_2 y} \end{pmatrix}, \quad \boldsymbol{\gamma}_{12}^n := \begin{pmatrix} \gamma_{1,1}^n \\ \gamma_{1,2}^n \end{pmatrix},$$

$$M_{34} := \begin{pmatrix} -ik & \lambda_2 \\ |k| & ik \end{pmatrix}, \quad e^{\xi_{34}y} := \begin{pmatrix} e^{\xi_3 y} & 0 \\ 0 & e^{\xi_4 y} \end{pmatrix}, \quad \boldsymbol{\gamma}_{34}^n := \begin{pmatrix} \gamma_{2,3}^n \\ \gamma_{2,4}^n \end{pmatrix}.$$

Therefore (10) becomes $\widehat{H}_{11}\boldsymbol{\gamma}_{12}^n = widehatH_{12}\boldsymbol{\gamma}_{34}^{n-1}$, $\widehat{H}_{22}\boldsymbol{\gamma}_{34}^n = \widehat{H}_{21}\boldsymbol{\gamma}_{12}^{n-1}$, where, with $\frac{dM_{12}e^{\xi_{12}y}\boldsymbol{\gamma}_{12}^n}{dy} = M_{12}\xi_{12}e^{\xi_{12}y}\boldsymbol{\gamma}_{12}^n$ and $\frac{d^2M_{12}e^{\xi_{12}y}\boldsymbol{\gamma}_{12}^n}{dy} = M_{12}\xi_{12}^2e^{\xi_{12}y}\boldsymbol{\gamma}_{12}^n$, we have

$$\begin{aligned} \widehat{H}_{11} &= MM_{12}\begin{pmatrix} \xi_1^2 & 0 \\ 0 & \xi_2^2 \end{pmatrix} + M_{12}\begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{pmatrix} + P_1M_{12} + \hat{S}_1M_{12}, \\ \widehat{H}_{12} &= MM_{34}\begin{pmatrix} \xi_3^2 & 0 \\ 0 & \xi_4^2 \end{pmatrix} + M_{34}\begin{pmatrix} \xi_3 & 0 \\ 0 & \xi_4 \end{pmatrix} + P_2M_{34} + \hat{S}_1M_{34}, \\ \widehat{H}_{22} &= -MM_{34}\begin{pmatrix} \xi_3^2 & 0 \\ 0 & \xi_4^2 \end{pmatrix} - M_{34}\begin{pmatrix} \xi_3 & 0 \\ 0 & \xi_4 \end{pmatrix} - P_2M_{34} + \hat{S}_2M_{34}, \\ \widehat{H}_{21} &= -MM_{12}\begin{pmatrix} \xi_1^2 & 0 \\ 0 & \xi_2^2 \end{pmatrix} - M_{12}\begin{pmatrix} \xi_1 & 0 \\ 0 & \xi_2 \end{pmatrix} - P_1M_{12} + \hat{S}_2M_{12}. \end{aligned}$$

Assuming that $\widehat{H}_{11}$ and $\widehat{H}_{22}$ are invertible ,we thus get for the error coefficients $\boldsymbol{\gamma}_{12}^n$ the recurrence relation $\boldsymbol{\gamma}_{12}^n = \widehat{H}_{11}^{-1}\widehat{H}_{12}\widehat{H}_{22}^{-1}\widehat{H}_{21}\boldsymbol{\gamma}_{12}^{n-2}$. Hence, the convergence factor of the error in the Schwarz method (2) is determined by the spectral radius of the *iteration operator* given by

$$\widehat{H}(k, \hat{S}_1, \hat{S}_2) := \widehat{H}_{11}^{-1}\widehat{H}_{12}\widehat{H}_{22}^{-1}\widehat{H}_{21}. \tag{11}$$

Convergence factor: In order to ensure fast convergence of the Schwarz algorithm for all possible frequencies $k \in \mathbb{R}$, we have to choose operators $\hat{S}_j$ $(j = 1, 2)$ that make the convergence factor small [9, 5, 2]. The convergence factor is

$$\rho_{\text{OSM}}(k, \hat{S}_1, \hat{S}_2) := \rho(\widehat{H}(k, \hat{S}_1, \hat{S}_2)) < 1, \tag{12}$$

where $\rho(\widehat{H})$ is the spectral radius of $\widehat{H}$ for a fixed k and $\hat{S}_j$ $(j = 1, 2)$.

Optimal operators: The symbols $\hat{S}_j$ (or equivalently operators S_j) are still free to be chosen at this point. It is possible to make the right hand of the transmission conditions (10) vanish, and to obtain an algorithm that converges in two iterations, if we choose

$$\hat{S}_1^* := -MM_{34}\begin{pmatrix}\xi_3^2 & 0\\ 0 & \xi_1^2\end{pmatrix}M_{34}^{-1} - M_{34}\begin{pmatrix}\xi_3 & 0\\ 0 & \xi_4\end{pmatrix}M_{34}^{-1} - P_2,$$
$$\hat{S}_2^* := MM_{12}\begin{pmatrix}\xi_1^2 & 0\\ 0 & \xi_1^2\end{pmatrix}M_{12}^{-1} + M_{12}\begin{pmatrix}\xi_1 & 0\\ 0 & \xi_2\end{pmatrix}M_{12}^{-1} + P_1,$$

and a lengthy calculation permits to simplify the preceding expressions, yielding

$$\hat{S}_1^* = \begin{pmatrix}|k| + \lambda_2(k) & \frac{ik}{|k|}\lambda_2(k)\\ \frac{-ik}{|k|}\lambda_2(k) & \frac{\lambda_2(k)}{|k|}(|k| + \lambda_2(k))\end{pmatrix},\ \hat{S}_2^* = \begin{pmatrix}|k| + \lambda_1(k) & \frac{-ik}{|k|}\lambda_1(k)\\ \frac{ik}{|k|}\lambda_1(k) & \frac{\lambda_1(k)}{|k|}(|k| + \lambda_1(k))\end{pmatrix}. \tag{13}$$

Some terms in these operators are not polynomials in ik, and thus the corresponding operators $S_j = \mathcal{F}_x^{-1}(\hat{S}_j)$ in real space are nonlocal in x, which is not convenient for implementations, since it requires convolution computations.

3 Optimized Schwarz methods and asymptotic performance

We would therefore like to approximate $\hat{S}_j^*$ by local operators that still give very fast convergence of the Schwarz iteration. The idea is to find local operators $\hat{S}_j$ that minimize the convergence factor (12) uniformly over a relevant range of frequencies, which leads to the min-max problem

$$\min_{\hat{S}_j} \quad \left(\max_{k\in[k_{\min},k_{\max}]} \rho_{\mathrm{OSM}}(k, \hat{S}_1, \hat{S}_2)\right). \tag{14}$$

Although the problem we considered before is a continuous model on the infinite plane, the range of frequencies can be bounded by incorporating information about the actual discretized problem we intend to solve. In (14), $k_{\min}$ can in general be negative, but when the optimized $\hat{S}_j$ lead to an even convergence factor in k, as we will see later, we can equivalently assume $k_{\min} > 0$. Thus the minimal frequency component of the solution can be estimated by $k_{\min} = \frac{\pi}{L}$ for an interface of length L, and $k_{\max} = \frac{\pi}{h}$ with grid spacing h, see for example [9, 6].

Let $\hat{S}_j$ $(j = 1, 2)$ keep the sign, symmetry and parity of the optimal operator in (13), and let us denote $\hat{S}_j := \begin{pmatrix}\mathcal{S}_{11}^j(k) & \mathcal{S}_{12}^j(k)\\ \mathcal{S}_{21}^j(k) & \mathcal{S}_{22}^j(k)\end{pmatrix}$, with $\mathcal{S}_{21}^j(k) = -\mathcal{S}_{12}^j(k)$ $(j = 1, 2)$. We first study properties of $\widehat{H}(k, \hat{S}_1, \hat{S}_2)$ for these $\hat{S}_j$, which can be obtained by a lengthy technical computation that will appear elsewhere [11].

Lemma 1 *Assuming that* $\mathcal{S}_{12}^j = ik \cdot f_{0j}(k)$ *and* $\mathcal{S}_{ll}^j$ *and* $f_{0j}(k)$ $(l, j = 1, 2)$ *are even functions of* k*, then* $\widehat{H}(k, \hat{S}_1, \hat{S}_2)$ *is always of the form*

$$\widehat{H}(k, \hat{S}_1, \hat{S}_2) = \begin{pmatrix} f_1(k) & ig_1(k) \\ ig_2(k) & f_2(k) \end{pmatrix},$$

where $f_l(k)$ *are even functions and* $g_l(k)$ *are odd functions of* k *for* $k \in \mathbb{R}$ $(l = 1, 2)$*. Furthermore, this implies that*

- the eigenvalues of $\widehat{H}(k, \hat{S}_1, \hat{S}_2)$ *are even functions of* k *for* $k \in \mathbb{R}$*,*

- the optimized problem in (14) *is equivalent to restricting* $k \in \mathbb{R}^+$*.*

Now, we derive optimized Robin-like transmission conditions for continuous and discontinuous coefficients κ.

The continuous case, $\kappa_1 = \kappa_2 := \kappa$: In this case, $\lambda_j(k) = \sqrt{k^2 + \frac{1}{\kappa}} = \lambda(k)$, and we introduce the structurally consistent approximations replacing k by the constant p for the non-local terms k and $\lambda_j(k)$, c.f. (13),

$$\hat{S}_1^c := \begin{pmatrix} p + \lambda(p) & \frac{ik}{p}\lambda(p) \\ \frac{-ik}{p}\lambda(p) & \frac{\lambda(p)}{p}(p + \lambda(p)) \end{pmatrix}, \quad \hat{S}_2^c := \begin{pmatrix} p + \lambda(p) & \frac{-ik}{p}\lambda(p) \\ \frac{ik}{p}\lambda(p) & \frac{\lambda(p)}{p}(p + \lambda(p)) \end{pmatrix},$$

with one free parameter p, where $p, k > 0$ (using superscript c for continuous to distinguish from the following discontinuous case). With $\hat{S}_j = \hat{S}_j^c$, the convergence factor $\rho_{\text{OSM}}(k, \hat{S}_1, \hat{S}_2)$ only depends on k and p, so we denote it by $\rho_{\text{OSM}}(k, p)$. A lengthy computation shows that $\widehat{H}_{11}^{-1}\widehat{H}_{12} = \widehat{H}_{22}^{-1}\widehat{H}_{21}$, and we obtain the following property to choose the maximum of the two eigenvalues of $\widehat{H}$.

Lemma 2 *The eigenvalues* $\mu^{\pm}(k, p)$ *of* $\widehat{H}(k, \hat{S}_1^c, \hat{S}_2^c)$ *are always positive, and*

$$sign(\mu^+(k, p) - \mu^-(k, p)) = sign(p - k).$$

From Fig.1 (left), we find that the optimized parameter p^* for continuous κ is characterized by an equioscillation property: $\rho_{\text{OSM}}(k_{\min}, p^*) = \rho_{\text{OSM}}(k_{\max}, p^*)$.

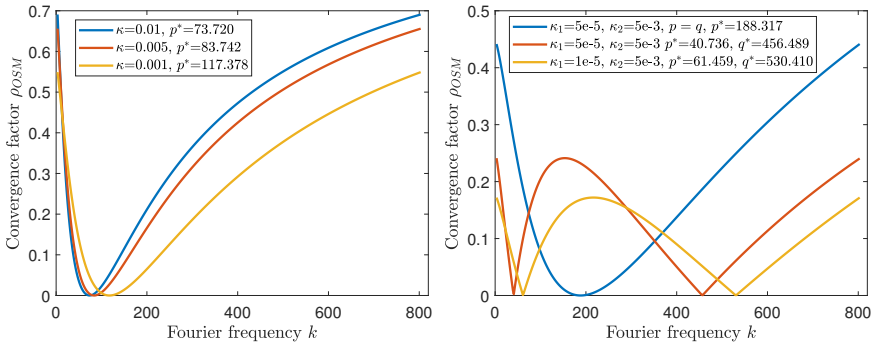


Fig. 1 Left: Optimized $\rho_{\text{OSM}}(k, p)$ for $\kappa_1 = \kappa_2$. Right: Optimized $\rho_{\text{OSM}}(k, p, q)$ for $\kappa_1 \neq \kappa_2$.

Theorem 1 *The optimized parameter p^* solved from $\mu^+(k_{\min}, p) = \mu^-(k_{\max}, p)$ is*

$$p^* \sim C_p h^{-\frac{1}{2}}, \quad C_p := \sqrt{\frac{L^2 - D^2}{4\kappa \frac{D^2}{L}}}, \quad D := \pi\sqrt{\kappa} - \sqrt{L^2 + \kappa\pi^2},$$

when $k_{\min} = \frac{\pi}{L}$ and $k_{\max} = \frac{\pi}{h}$. Furthermore, the asymptotic convergence factor of the resulting one-side optimized Schwarz method is

$$\min_{k \in [k_{\min}, k_{\max}]} \rho_{\text{OSM}}(k, p^*) \sim 1 - C \cdot h^{\frac{1}{2}}, \quad C := \frac{4C_p}{\pi}.$$

Proof We make the ansatz $p^* := C_p \cdot h^{-\frac{1}{2}}$. Expanding for small h, we obtain

$$\mu^+(k_{\min}, p^*) = 1 - \frac{L(L^2 - D^2)}{C_p \kappa \pi D^2}\sqrt{h} + O(h), \text{ where } D = \pi\sqrt{\kappa} - \sqrt{L^2 + \kappa\pi^2},$$
$$\mu^-(k_{\max}, p^*) = 1 - \frac{4C_p}{\pi}\sqrt{h} + O(h)$$

Solving $\mu^+(k_{\min}, p) = \mu^-(k_{\max}, p)$ asymptotically then determines p^*. □

The discontinuous case, $\kappa_1 \neq \kappa_2$: For p and q ($p, q > 0$) two free parameters, we introduce the structurally consistent approximations (superscript d for discontinuous)

$$\hat{S}_1^d = \begin{pmatrix} p + \lambda_2(p) & \frac{ik}{p}\lambda_2(p) \\ \frac{-ik}{p}\lambda_2(p) & \frac{\lambda_2(p)}{p}(p + \lambda_2(p)) \end{pmatrix}, \quad \hat{S}_2^d = \begin{pmatrix} q + \lambda_1(q) & \frac{-ik}{q}\lambda_1(q) \\ \frac{ik}{q}\lambda_1(q) & \frac{\lambda_1(q)}{q}(q + \lambda_1(q)) \end{pmatrix},$$

and study the associated convergence factor $\rho_{\text{OSM}}(k, p, q)$ numerically. We show in Fig.1 (right) that

- the optimized parameters p^* and q^* are characterized by an equioscillation property: $\rho_{\text{OSM}}(k_{\min}, p^*, q^*) = \rho_{\text{OSM}}(k_{\max}, p^*, q^*) = \rho_{\text{OSM}}(\bar{k}, p^*, q^*)$,

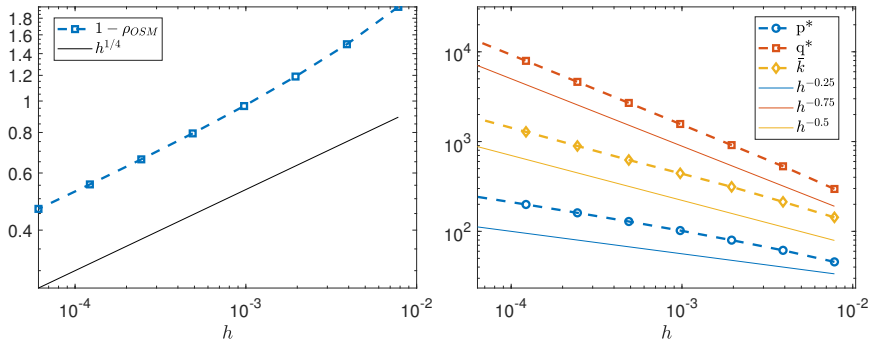


Fig. 2 Log–log plot of the convergence factor and the optimized parameters for $\kappa_1 = 10^{-5}$ and $\kappa_2 = 5 \times 10^{-3}$.

- high contrast κ leads to fast convergence,
- the two parameters (two-sided Robin) give better convergence than the one parameter (one sided Robin) case earlier.

Numerically, we observe in Fig. 2 that the asymptotic performance is given by

$$p^* = C_1 h^{-\frac{1}{4}},\ q^* = C_2 h^{-\frac{3}{4}}, \bar{k} = C_3 h^{-\frac{1}{2}}, \quad \rho_{\mathrm{OSM}}(k, p^*, q^*) = 1 - C_0 h^{\frac{1}{4}} + O(h^{\frac{1}{2}}), \tag{15}$$

where C_0, C_1, C_2 and C_3 are constants.

Acknowledgements This work was supported by scholarship from China Scholarship Council and the Swiss National Science Foundation.

References

1. Benamou, J. A domain decomposition method with coupled transmission conditions for the optimal control of systems governed by elliptic partial differential equations. *SIAM J. Numer. Anal.* **33**(6), 2401–2416 (1996).
2. Blayo, E., Cherel, D., and Rousseau, A. Towards optimized Schwarz methods for the Navier-Stokes equations. *J. Sci. Comput.* **66**(1), 275–295 (2016).
3. Brinkman, H. A calculation of the viscous force exerted by a flowing fluid on a dense swarm of particles. *Flow Turbul. Combust.* **1**, 27–34 (1949).
4. Discacciati, M. and Gerardo-Giorda, L. Optimized Schwarz methods for the Stokes-Darcy coupling. *IMA J. Numer. Anal.* **38**(4), 1959–1983 (2018).
5. Gander, M. J. Optimized Schwarz methods. *SIAM J. Numer. Anal.* **44**(2), 699–731 (2006).
6. Gander, M. J. and Dubois, O. Optimized Schwarz methods for a diffusion problem with discontinuous coefficient. *Numer. Algorithms* **69**(1), 109–144 (2015).
7. Gander, M. J. and Vanzan, T. On the derivation of optimized transmission conditions for the Stokes-Darcy coupling. In: *Domain decomposition methods in science and engineering XXV*, *Lect. Notes Comput. Sci. Eng.*, vol. 138, 491–498. Springer, Cham (2020).
8. Halpern, L. Artificial boundary conditions for incompletely parabolic perturbations of hyperbolic systems. *SIAM J. Math. Anal.* **22**(5), 1256–1283 (1991).
9. Japhet, C. and Nataf, F. The best interface conditions for domain decomposition methods: absorbing boundary conditions. In: *Absorbing boundaries and layers, domain decomposition methods*, 348–373. Nova Sci. Publ., Huntington, NY (2001).
10. Lions, P.-L. On the Schwarz alternating method. III. A variant for nonoverlapping subdomains. In: *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989)*, 202–223. SIAM, Philadelphia, PA (1990).
11. M.J. Gander, Y.Y. Wang, and Y.X. Xu, M. Analysis of optimized Schwarz methods for the Brinkman equations. *in preparation* .
12. Xu, Y. and Chen, X. Optimized Schwarz methods for the optimal control of systems governed by elliptic partial differential equations. *J. Sci. Comput.* **79**(2), 1182–1213 (2019).

Parareal Algorithms for the Cahn-Hilliard Equation

Gobinda Garai and Bankim C. Mandal

1 Introduction

In this work we are interested in designing time parallel algorithm for the Cahn-Hilliard (CH) equation. The CH equation

$$\begin{aligned} \frac{\partial u}{\partial t} &= \Delta f(u) - \varepsilon^2 \Delta^2 u \quad \text{for } (x,t) \in \Omega(\subset \mathbb{R}) \times (0,T], \\ \frac{\partial u}{\partial n} &= \frac{\partial(\Delta u)}{\partial n} = 0 \quad \text{for } (x,t) \in \partial\Omega \times (0,T], \\ u(x,0) &= u_0(x) \quad \text{for } x \in \Omega, \end{aligned} \tag{1}$$

is a prototype to display the evolution of a binary melted alloy below the critical temperature; see [2, 3]. The nonlinear function $f(u)$ satisfies $f(u) = F'(u)$, where $F(u) = 0.25(u^2-1)^2$ is the homogeneous free energy. As the solution u of (1) takes values in $[-1,1]$, the function $f(u)$ becomes Lipschitz with Lipschitz constant 2. The solution of (1) involves two different dynamics, one being the phase separation which is rapid in time and phase regions are separated by the interface of width $\varepsilon(0 < \varepsilon \ll 1)$. Another is phase coarsening which is slower in time, during this stage the solution lean towards an equilibrium state which reduces the internal energy. The energy associated with the CH equation is

$$\mathscr{E}(u) := \int_\Omega \left(F(u) + \frac{\varepsilon^2}{2} |\nabla u|^2 \right) dx,$$

known as the Ginzburg-Landau free energy functional. The energy functional $\mathscr{E}(u)$ and total mass $\int_\Omega u$ satisfy the following

Gobinda Garai, Bankim C. Mandal
School of Basic Sciences (Mathematics), Indian Institute of Technology Bhubaneswar, India,
e-mail: gg14@iitbbs.ac.in, bmandal@iitbbs.ac.in

$$\frac{d}{dt}\mathscr{E}(u) \leq 0, \qquad \frac{d}{dt}\int_{\Omega} u = 0.$$

And the energy minimization and mass conservation property of (1) is expected to be preserved by numerical method. To deal with that, Eyre proposed an unconditionally gradient stable scheme in [4, 5]. The idea is to split the homogeneous free energy $F(u)$ into the sum of a convex and a concave term, and then treat the convex term implicitly and the concave term explicitly to obtain a nonlinear approximation for the CH equation (1) in 1D as:

$$u_j^{n+1} - u_j^n = \Delta t A(u_j^{n+1})^3 - \Delta t A u_j^n - \varepsilon^2 \Delta t A^2 u_j^{n+1}, \tag{2}$$

where Δt is the time step and A is the discrete Laplacian and the scheme is $O(\Delta t + \Delta x^2)$ accurate [4, 5]. To get a linear approximation of (1) the term $(u_j^{n+1})^3$ in (2) is rewritten as $(u_j^n)^2 u_j^{n+1}$ to get the following

$$u_j^{n+1} - u_j^n = \Delta t A(u_j^n)^2 u_j^{n+1} - \Delta t A u_j^n - \varepsilon^2 \Delta t A^2 u_j^{n+1}, \tag{3}$$

which is also an unconditionally gradient stable scheme and has the same accuracy as the nonlinear scheme (2), see [4]. This is known as linearly stabilized splitting scheme (LSS). We also use the following semi-implicit Euler (SIE) approximation of (1)

$$u_j^{n+1} - u_j^n = \Delta t A(u_j^n)^3 - \Delta t A u_j^n - \varepsilon^2 \Delta t A^2 u_j^{n+1}, \tag{4}$$

though it is not a physically relevant approximation as the scheme is not gradient stable [5]. The solution of (1) involves long time dynamics, namely phase coarsening stage, thus the CH equation (1) needs to be simulated over long time window to get the solution. Therefore it is of great importance to develop efficient time parallel method for (1) to speed-up the computation. To achieve this we construct the Parareal methods [9] for (1). The Parareal method has been successfully applied to: fluid-structure interaction in [6], Navier-Stokes equation in [7], molecular-dynamics in [1]. The main objective of this work is to adapt the Parareal algorithm for the CH equation (1) and study the convergence behaviour.

We introduce the Parareal algorithm in one spatial dimension for the CH equation in Section 2. In Section 3 we discuss stability and convergence property of the Parareal method. To illustrate our theoretical findings, the accuracy and robustness of the proposed algorithms, we present the numerical results in Section 4.

2 Parareal method

To solve the following system of ODEs

$$\frac{du}{dt} = f(u), \quad u(0) = u_0, \quad t \in (0, T], \tag{5}$$

Lions et al. proposed the Parareal algorithm in [9], where $f\colon \mathbb{R}^+ \times \mathbb{R}^d \to \mathbb{R}^d$ is Lipschitz. The method constitutes of the following strategy: first a non-overlapping

decomposition of time domain $(0,T]$ into N smaller subintervals of uniform size, i.e., $(0,T] = \cup_{n=1}^{N}[T_{n-1},T_n]$ with $T_n - T_{n-1} = \Delta T = T/N$, secondly one divides each time slice $[T_{n-1},T_n]$ into J smaller time slices with $\Delta t = \Delta T/J$, then a fine propagator F which is expensive but accurate, and a coarse propagator G which is cheap but may be inaccurate are assigned to compute the solution in fine grid and coarse grid, respectively. Then the Parareal algorithm for (5) starts with an initial approximation U_n^0 at T_n's, obtained by the coarse operator G and solve for $k = 0,1,\ldots$

$$\begin{aligned} U_0^{k+1} &= u_0, \\ U_{n+1}^{k+1} &= G(T_{n+1},T_n,U_n^{k+1}) + F(T_{n+1},T_n,U_n^k) - G(T_{n+1},T_n,U_n^k), \end{aligned} \tag{6}$$

where $S(T_{n+1},T_n,U_n^k)$ provides solution at T_{n+1} by taking the initial solution U_n at T_n for the k-th iteration for $S = F$ or G. The Parareal solution U_{n+1}^{k+1} converges towards the fine resolution in finite step. To get a practical parallel algorithm we should have $k \ll N$.

Now to employ discrete Parareal method for the CH equation (1) we discretize (1) as shown earlier and denote U_n^k as $u(jh,T_n), j = 1,2,\ldots,N_x$ in (6) for k-th iteration, where h is spatial mesh size and N_x is number of nodes in spatial domain. We fix the fine propagator F to be the LSS scheme (3) in (6). For the coarse operator G in (6) we consider the following three choices:

(i) The coarse propagator G is given by the LSS scheme in (3).

(ii) The coarse propagator G is given by the SIE scheme in (4).

(iii) The coarse propagator G is given by the implicit scheme of the heat equation

$$u_t = 2\Delta u, \tag{7}$$

which is a linearization of (1) with respect a constant solution and then truncate the fourth order derivative term as ε is small.

The third choice of coarse operator is interesting as the equation (7) does not represent the underlying physics of the equation (1). Here we study the convergence behaviour of the Parareal algorithm corresponding to the coarse operators (ii) and (iii). The coarse operator corresponding to (ii) and (iii) can be written as

$$G_{\mathrm{SI}}(U) = \left(I + \varepsilon^2 \Delta T A^2\right)^{-1} \left(U + \Delta T A f(U)\right), \quad U \in \mathbb{R}^{N_x}, \tag{8a}$$

$$G_{\mathrm{IH}}(U) = \left(I - 2\Delta T A\right)^{-1} U, \quad U \in \mathbb{R}^{N_x} \tag{8b}$$

respectively, and $A = \frac{1}{h^2}[1 \ -2 \ \ 1] \in \mathbb{R}^{N_x \times N_x}$ with $A(1,2) = 2 = A(N_x,N_x-1)$ is the discrete Laplacian with homogeneous Neumann boundary conditions.

3 Stability and convergence result

In this section, we discuss the stability and convergence issues related to the coarse operators in (8). We start with a few auxiliary results.

Lemma 1 (Growth of coarse operators)

The coarse operators in (8) *satisfy the growth condition*

$$\| G_{SI}(U) \| \leq \| U \|, \quad \forall U \in \mathbb{R}^{N_x} \tag{9a}$$

$$\| G_{IH}(U) \| \leq \| U \|, \quad \forall U \in \mathbb{R}^{N_x}. \tag{9b}$$

Proof The eigenvalues of A are $\lambda_p = \frac{2}{h^2}\left\{\cos\left(\frac{(p-1)\pi}{N_x-1}\right)-1\right\}, p = 1,\cdots,N_x$. Clearly, λ_p's are distinct and satisfy $\lambda_p \leq 0, \forall p$. By taking norm on (8a) and using Lipschitz condition on f we get $\| G_{\mathrm{SI}}(U) \| \leq \max_{\lambda_p}\left|\frac{1+2\Delta T\lambda_p}{1+\varepsilon^2\Delta T\lambda_p^2}\right| \| U \|$. Now the function $g(x) = \frac{1-2\Delta Tx}{1+\varepsilon^2\Delta Tx^2} \leq 1$, $\forall x \geq 0$. Hence, we have (9a). Now $\| (I-2\Delta TA)^{-1} \| = \frac{1}{\min_{\lambda_p}\{1-2\Delta T\lambda_p\}} = 1$. Then by taking norm on (8b) we have (9b). □

Lemma 2 (Lipschitz property of G)

The coarse operators in (8) *satisfy the Lipschitz condition*

$$\| G_{SI}(T_{n+1},T_n,U) - G_{SI}(T_{n+1},T_n,V) \| \leq \| U-V \|, \quad \forall U,V \in \mathbb{R}^{N_x} \tag{10a}$$

$$\| G_{IH}(T_{n+1},T_n,U) - G_{IH}(T_{n+1},T_n,V) \| \leq \| U-V \|, \quad \forall U,V \in \mathbb{R}^{N_x}. \tag{10b}$$

Proof The results are straight forward. □

Lemma 3 (Local truncation error (LTE) differences)

Let $F(T_{n+1},T_n,U)$ be the fine operator in (3)*. For any coarse operators among $G_{SI}(T_{n+1},T_n,U), G_{IH}(T_{n+1},T_n,U)$ in* (8)*, the following LTE differences hold*

$$F(T_{n+1},T_n,U) - G_{SI}(T_{n+1},T_n,U) = c_2(U)\Delta T^2 + c_3(U)\Delta T^3 + \cdots, \tag{11a}$$

$$F(T_{n+1},T_n,U) - G_{IH}(T_{n+1},T_n,U) = c_1'(U)\Delta T + c_2'(U)\Delta T^2 + \cdots, \tag{11b}$$

where $c_j(U)$, $c_{j'}'(U)$ are continuously differentiable function for $j = 2,3,\ldots$, $j' = 1,2,\ldots$

Proof Let $S(T_{n+1},T_n,U)$ be the exact solution of (1). Since F and G_{SI} have LTE of $O(\Delta T^2)$, we have

$$\begin{aligned} &F(T_{n+1},T_n,U) - G_{\mathrm{SI}}(T_{n+1},T_n,U) \\ &\quad = F(T_{n+1},T_n,U) - S(T_{n+1},T_n,U) + S(T_{n+1},T_n,U) - G_{\mathrm{SI}}(T_{n+1},T_n,U) \\ &\quad = \widetilde{c_2}(U)\Delta T^2 + \widetilde{c_3}(U)\Delta T^3 + \cdots + \widehat{c_2}(U)\Delta T^2 + \widehat{c_3}(U)\Delta T^3 + \cdots \\ &\quad = c_2(U)\Delta T^2 + c_3(U)\Delta T^3 + \cdots. \end{aligned}$$

Similarly one can obtain LTE differences for G_{IH}. □

Theorem 1 (Stability) *Let $G_{SI}(T_{n+1},T_n,U_n)$ be the coarse operator in* (8a)*, then the corresponding Parareal method is stable, i.e., for each n and k, there exist a constant C such that*

$$\| U_{n+1}^{k+1} \| \leq \| u_0 \| + C\Delta T^2 (n+1) \left(\max_{0\leq j\leq n} \| U_j^k \| \right).$$

Proof Taking norm in the correction step (6) we have

$$\begin{aligned} \| U_{n+1}^{k+1} \| &\leq \| G_{\mathrm{SI}}(T_{n+1},T_n,U_n^{k+1}) \| + \| F(T_{n+1},T_n,U_n^k) - G_{\mathrm{SI}}(T_{n+1},T_n,U_n^k) \| \\ &\leq \| U_n^{k+1} \| + C\Delta T^2 \| U_n^k \|, \end{aligned} \tag{12}$$

where in the 2nd inequality we use (9a) and (11a). Taking the sum over n on the recurrence relation (12) we have

$$\| U_{n+1}^{k+1} \| - \| U_0^{k+1} \| \leq C\Delta T^2 \sum_{j=0}^{n} \| U_j^k \| \leq C\Delta T^2 (n+1) \left(\max_{0\leq j\leq n} \| U_j^k \| \right).$$

Now using $U_0^{k+1} = u_0$ we get the stated result. □

Theorem 2 (Stability) *Let $G_{IH}(T_{n+1},T_n,U_n)$ be the coarse operator in* (8b)*, then the corresponding Parareal method is stable, i.e., for each n and k, there exist a constant C such that*

$$\| U_{n+1}^{k+1} \| \leq \| u_0 \| + C\Delta T (n+1) \left(\max_{0\leq j\leq n} \| U_j^k \| \right).$$

Proof Proof can be obtained by following Theorem 1. □

Theorem 3 (Convergence) *Let $F(T_{n+1},T_n,U_n)$ be the fine operator in* (3) *and $G_{SI}(T_{n+1},T_n,U_n)$ be the coarse operator in* (8a)*. The propagator F and G_{SI} satisfy LTE differences* (11a) *and G_{SI} satisfies Lipschitz condition* (10a)*, then the corresponding Parareal method satisfies the following error estimation*

$$\| U(T_n) - U_n^k \| \leq \frac{C_3'}{C_1'} \frac{(C_1'\Delta T^2)^{k+1}}{(k+1)!} \prod_{j=0}^{k} (n-j),$$

where the constants C_1', C_3' are related to LTE.

Theorem 4 (Convergence) *Let $F(T_{n+1},T_n,U_n)$ be the fine operator in* (3) *and $G_{IH}(T_{n+1},T_n,U_n)$ be the coarse operator in* (8b)*. The propagator F and G_{IH} satisfy LTE differences* (11b) *and G_{IH} satisfies Lipschitz condition* (10b)*, then the corresponding Parareal method satisfies the following error estimation*

$$\| U(T_n) - U_n^k \| \leq \frac{C_3''}{C_1''} \frac{(C_1''\Delta T)^{k+1}}{(k+1)!} \prod_{j=0}^{k} (n-j),$$

where the constants C_1'', C_3'' are related to LTE.

The proof of Theorems 3 & 4 is followed by the argument of the proof of Theorem 1 in [8].

4 Numerical illustration

We now show numerical experiments of Parareal method for (1) corresponding to three different coarse operators. We consider the random initial condition for (1). The Parareal error is measured in $L^\infty(0,T;L^2)$, and we fix the spatial domain $\Omega = (0,2)$.

4.1 $F =$ LSS, $G =$ LSS

We first run the numerical experiments of Parareal method corresponding to fine and coarse operator as LSS scheme (3). We plot the error curves for short as well as long time window on the left panel in Figure 1 with $\varepsilon^2 = 0.01$, $J = 40$ and $h = 1/64$. The method converges in four iterations to the fine solution of temporal accuracy $O(10^{-4})$ for different T. For $T = 1$ we can see that the Parareal method 20 times faster than the serial method on single processor. To see the dependency of the parameter ε, we plot the error curve on the right in Figure 1 for different ε by fixing $T = 1$, $N = 80$, $J = 40$. We observe that the method behaves similar irrespective of any choice of ε. On the left of the Figure 2 we plot error curves for more refined solution for $T = 1$, $N = 80$, $J = 40$, $\varepsilon^2 = 0.01$. We observe that the convergence is independent of mesh parameters.

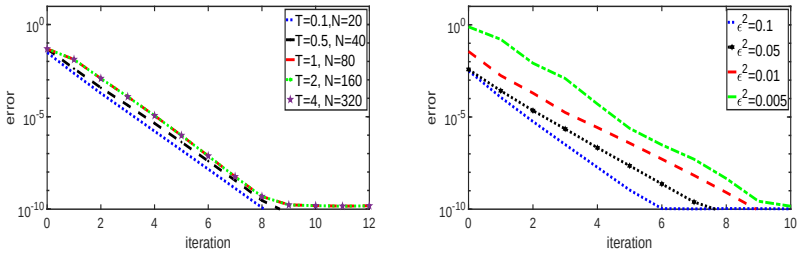


Fig. 1 On the left: different T and N; On the right: different choice of ε.

4.2 $F =$ LSS, $G =$ SIE

Now we run experiments of Parareal method corresponding to fine operator as LSS scheme (3) and coarse operator as SIE scheme (4). We plot the error curves on the right panel in Figure 2 for short as well as long time window with the parameters $\varepsilon^2 = 0.01$, $h = 1/64$, $J = 40$. Ignoring the cost of computing the coarse operator, it is visible that a reasonable speed up is possible; for example to get the solution at

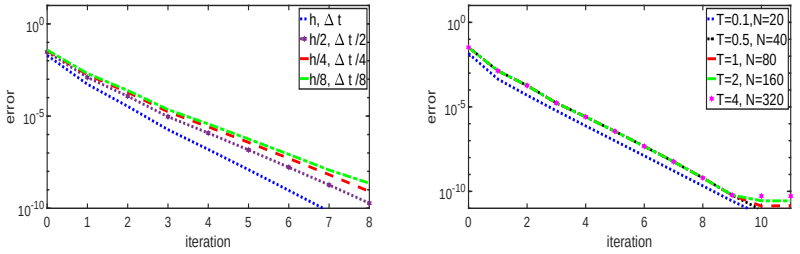


Fig. 2 On the left: different $h, \Delta t$ for LSS; On the right: different T, N for SIE.

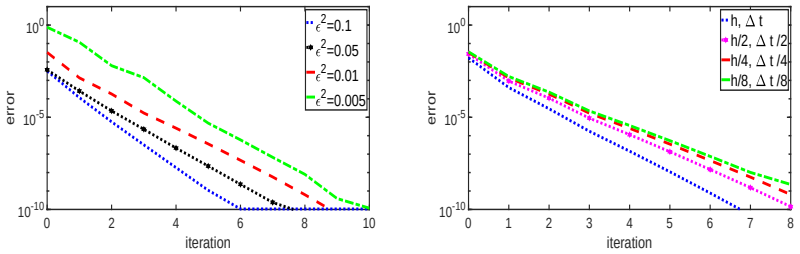


Fig. 3 On the left: different ε; On the right: different $h, \Delta t$.

$T = 4$ with an accuracy of $O(10^{-4})$ the method needs four iterations and this implies that the solution can be obtained 80 times faster than serial method on a single processor. We plot the error curves on the left in Figure 3 for different ε by taking $T = 1, N = 80, J = 40$ and we see that the convergence is independent of the choice of ε. On the right of the Figure 3 we plot error curves for more refined solution for $T = 1$, $N = 80$, $J = 40$, $\varepsilon^2 = 0.01$. We see that the convergence is independent of mesh parameters.

4.3 $F =$ LSS, $G =$ IH

We finally take the fine operator as LSS scheme (3) and coarse operator as implicit scheme of (7).We plot the error curves on the left in Figure 4 for short as well as long time window with the parameters $\varepsilon^2 = 0.01$, $h = 1/64$, $J = 40$ and small ΔT. We observe the convergence but it is not immediate. Even if we take reasonably large ΔT we obtain convergence but with very less speed up, see on the right of Figure 4. Even though the heat equation (7) as coarse operator provide solution we need further investigation to obtain the speed up.

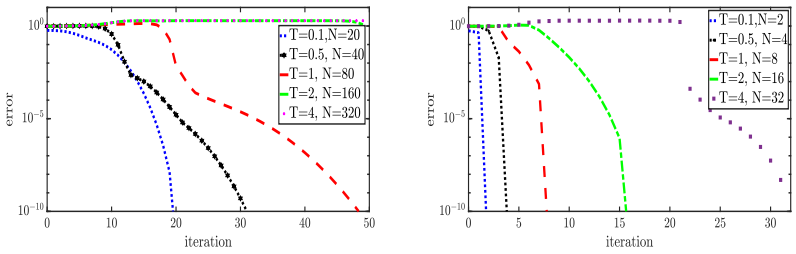


Fig. 4 On the left: small ΔT; On the right: large ΔT.

5 Conclusions

We formulated and studied the Parareal methods for the CH equation in 1D. We gave stability and convergence estimates of the Parareal method for different choices of coarse operator. Lastly we presented numerical experiments for all the proposed algorithms.

Acknowledgements Authors would like to thank the CSIR (File No:09/1059(0019)/2018-EMR-I) and DST-SERB (File No: SRG/2019/002164) for the financial assistance and IIT Bhubaneswar for research facility.

References

1. Baffico, L., Bernard, S., Maday, Y., Turinici, G., and Zérah, G. Parallel-in-time molecular-dynamics simulations. *Physical Review E* **66**(5), 057701 (2002).
2. Cahn, J. W. On spinodal decomposition. *Acta Metall* **9**(9), 795–801 (1961).
3. Cahn, J. W. and Hilliard, W. Free energy of a nonuniform system. i. interfacial free energy. *J. Chem. Phys.* **28**(2), 258–267 (1958).
4. Eyre, D. J. Unconditionally gradient stable time marching the Cahn-Hilliard equation. In: *Computational and mathematical models of microstructural evolution (San Francisco, CA, 1998)*, *Mater. Res. Soc. Sympos. Proc.*, vol. 529, 39–46. MRS, Warrendale, PA (1998).
5. Eyre, D. J. An unconditionally stable one-step scheme for gradient systems. *Unpublished article* (1998).
6. Farhat, C. and Chandesris, M. Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid–structure applications. *International Journal for Numerical Methods in Engineering* **58**(9), 1397–1434 (2003).
7. Fischer, P. F., Hecht, F., and Maday, Y. A parareal in time semi-implicit approximation of the navier-stokes equations. In: *Domain decomposition methods in science and engineering*, 433–440. Springer (2005).
8. Gander, M. J. and Hairer, E. Nonlinear convergence analysis for the parareal algorithm. In: *Domain decomposition methods in science and engineering XVII*, 45–56. Springer (2008).
9. Lions, J.-L., Maday, Y., and Turinici, G. A" parareal" in time discretization of pde's. *Comptes Rendus De L Academie Des Sciences Serie I-Mathematique* **332**(7), 661–668 (2001).

A Parallel Space-Time Finite Element Method for the Simulation of an Electric Motor

Peter Gangl, Mario Gobrial, and Olaf Steinbach

1 Introduction

Shape and topology optimization of electrical machines [3] as well as the optimal control [4] subject to parabolic evolution equations require an efficient solution of the direct simulation problem which is forward in time, and in most cases also of the adjoint problem which is backward in time. Space-time discretization methods [8] are therefore a method of choice to solve the overall system at once, and also to allow for adaptive refinements and a parallel solution simultaneously in space and time. In the case of a fixed spatial domain the numerical analysis of a space-time finite element method was given, e.g., in [7], see also the review article [8] and the references given therein. In this note we present an extension of this approach in order to simulate an electric motor where one part, the rotor, is rotating in time, while the stator is fixed. In addition to the stator and the rotor we have to include an air domain which is non-conducting. Hence we have to deal with an elliptic-parabolic interface problem for the eddy current approximation of the Maxwell system in two space dimensions. In this paper we present its space-time finite element discretization and provide first numerical results using parallel solution strategies in order to handle the overall system in the space-time domain. More details on the numerical analysis of the proposed method and further numerical results will be given in our forthcoming paper [2].

This paper is structured as follows: In Section 2 we describe the mathematical model and its space-time variational formulation. Unique solvability is based on the Babuška–Nečas theorem, i.e., on injectivity and surjectivity of the operator which is

Peter Gangl
Johann Radon Institute for Computational and Applied Mathematics, Altenberger Straße 69, 4040 Linz, Austria, e-mail: peter.gangl@oeaw.ac.at

Mario Gobrial, Olaf Steinbach
Institute of Applied Mathematics, TU Graz, Steyrergasse 30, 8010 Graz, Austria
e-mail: gobrial@math.tugraz.at, o.steinbach@tugraz.at

associated to the bilinear form of the variational formulation. The space-time finite element discretization is given in Section 3, which also provides an a priori error estimate, i.e., Cea's lemma, for the numerical solution. Numerical results are given in Section 4, and finally we provide some conclusions and comment on ongoing work.

2 Mathematical model and space-time variational formulation

To model the electromagnetic fields in a rotating electric machine, we consider the eddy current approximation of the Maxwell equations,

$$\operatorname{curl}_y H(y,t) = j(y,t), \quad \operatorname{curl}_y E(y,t) = -\partial_t B(y,t), \quad \operatorname{div}_y B(y,t) = 0,$$

subject to the constitutive equations

$$B(y,t) = \mu(y)H(y,t) + M(y,t),\ j(y,t) = j_i(y,t) + \sigma(y)\Big[E(y,t) + v(y,t) \times B(y,t)\Big],$$

with the material dependent magnetic permeability μ, the electric conductivity σ, and an impressed electric current j_i. Moreover, M is the magnetization which vanishes outside permanent magnets. For a reference point $x \in \mathbb{R}^3$ we consider the trajectory $y(t) = \varphi(t,x)$, where the deformation φ is assumed to be bijective and sufficiently regular for all $t \in (0,T)$, satisfying $\varphi(0,x) = x$. Here, $T > 0$ is a given time horizon. Finally, we introduce the velocity $v(y,t) = \frac{d}{dt} y(t)$. In addition we consider appropriate boundary and initial conditions to be specified.

When using the vector potential ansatz $B(y,t) = \operatorname{curl}_y A(y,t)$, and following the standard approach to consider a spatially two-dimensional reference domain $\Omega \subset \mathbb{R}^2$ describing the cross section of the electric machine, this gives an evolution equation to find $u(y,t)$ as third component of $A = (0,0,u)^\top$ such that

$$\sigma(y)\,\frac{d}{dt} u(y,t) - \operatorname{div}_y[\nu(y)\nabla_y u(y,t)] = j_i(y,t) - \operatorname{div}_y[\nu(y)M^\perp(y,t)] \tag{1}$$

is satisfied in the space-time domain

$$Q := \Big\{(y,t) \in \mathbb{R}^3 : y = \varphi(t,x) \in \Omega(t),\ x \in \Omega \subset \mathbb{R}^2,\ t \in (0,T)\Big\}.$$

Note that in (1) we use the reluctivity $\nu = 1/\mu$, and the total time derivative

$$\frac{d}{dt} u(y,t) := \partial_t u(y,t) + v(y,t) \cdot \nabla_y u(y,t).$$

Moreover, $M^\perp = (-M_2, M_1)^\top$ is the perpendicular of the first two components of the magnetization M. In addition to (1) we consider homogeneous Dirichlet boundary conditions $u = 0$ on $\Sigma := \partial\Omega \times (0,T)$, and the homogeneous initial condition $u(x,0) = 0$ whenever $\sigma(x) > 0$ is satisfied for $x \in \Omega$.

The electric motor consists of a rotor in $\Omega_r(t)$, the stator in Ω_s, and the air domain Ω_{air} which is non-conducting, i.e., $\sigma = 0$ in Ω_{air}. This shows that we can formulate (1) as an elliptic-parabolic interface problem. While the stator domain Ω_s is fixed in time, i.e., $v \equiv 0$, the rotating subdomain $\Omega_r(t)$ can be described, when using polar coordinates, as

$$x = r \begin{pmatrix} \cos\varphi \\ \sin\varphi \end{pmatrix}, \quad y(t) = \varphi(t,x) = r \begin{pmatrix} \cos(\varphi + \alpha t) \\ \sin(\varphi + \alpha t) \end{pmatrix} \in \Omega_r(t),\ t \in (0,T),$$

with $\alpha > 0$ describing the velocity

$$v(y,t) = \frac{d}{dt} y(t) = \alpha r \begin{pmatrix} -\sin(\varphi + \alpha t) \\ \cos(\varphi + \alpha t) \end{pmatrix} = \alpha \begin{pmatrix} -y_2 \\ y_1 \end{pmatrix}.$$

The variational formulation of the parabolic-elliptic interface problem (1) is to find $u \in X$ such that

$$\begin{aligned} b(u,z) &:= \int_0^T \int_{\Omega(t)} \left[\sigma \frac{d}{dt} u\, z + \nu \nabla_y u \cdot \nabla_y z \right] dy\, dt \\ &= \int_0^T \int_{\Omega(t)} \left[j_i\, z + \nu M^{\perp} \cdot \nabla_y z \right] dy\, dt \end{aligned} \tag{2}$$

is satisfied for all $z \in Y$, where $Y := L^2(0,T; H_0^1(\Omega(t)))$ and

$$X := \left\{ u \in Y : \sigma \frac{d}{dt} u \in Y^*,\ u(x,0) = 0 \text{ for } x \in \Omega : \sigma(x) > 0 \right\}.$$

Related norms are given by

$$\|z\|_Y^2 := \int_0^T \int_{\Omega(t)} \nu\, |\nabla_y z|^2\, dy\, dt, \quad \|u\|_X^2 := \|u\|_Y^2 + \|w_u\|_Y^2,$$

where $w_u \in Y$ is the unique solution of the variational formulation

$$\int_0^T \int_{\Omega(t)} \nu \nabla_y w_u \cdot \nabla_y z\, dy\, dt = \int_0^T \int_{\Omega(t)} \sigma \frac{d}{dt} u\, z\, dy\, dt \quad \text{for all } z \in Y. \tag{3}$$

The bilinear form $b(\cdot,\cdot)$ as defined in (2) is bounded and satisfies an inf-sup stability condition, see [2, 7], i.e., for all $u \in X$ and $z \in Y$ there holds

$$|b(u,z)| \le \sqrt{2}\, \|u\|_X \|z\|_Y, \qquad \frac{1}{\sqrt{2}} \|u\|_X \le \sup_{0 \neq z \in Y} \frac{b(u,z)}{\|z\|_Y}.$$

Moreover, the bilinear form $b(\cdot,\cdot)$ is surjective, i.e., for any $z \in Y$ there exists a $u_z \in X$ such that $b(u_z, z) > 0$ is satisfied, see [2]. Hence, all assumptions of the Babuška–Nečas theorem [1, 5] are satisfied, i.e. we conclude unique solvability of the space-time variational formulation (2).

3 Space-time finite element discretization

For the space-time finite element discretization of the variational formulation (2) we introduce conforming finite dimensional spaces $X_h \subset X$ and $Y_h \subset Y$ where we assume as in the continuous case $X_h \subset Y_h$. For our specific purpose we even consider $X_h = Y_h := S_h^1(Q_h) \cap X = \mathrm{span}\{\varphi_k\}_{k=1}^M$ as the space of piecewise linear and continuous basis functions φ_k which are defined with respect to some admissible locally quasi-uniform decomposition $\mathcal{T}_h = \{\tau_\ell\}_{\ell=1}^N$ of the space-time domain Q into shape-regular simplicial finite elements τ_ℓ of mesh size h_ℓ, see, e.g., [6].

The Galerkin space-time finite element variational formulation of (2) reads to find $u_h \in X_h$, such that

$$b(u_h, z_h) = \int_0^T \int_{\Omega(t)} \Big[j_i \, z_h \, + \, \nu \, M^\perp \cdot \nabla_y z_h \Big] \, dy \, dt \quad \text{for all } z_h \in Y_h. \tag{4}$$

Unique solvability of (4) is based on the discrete inf-sup stability condition

$$\frac{1}{\sqrt{2}} \, \|u_h\|_{X_h} \leq \sup_{0 \neq z_h \in Y_h} \frac{b(u_h, z_h)}{\|z_h\|_Y} \quad \text{for all } u_h \in X_h,$$

which follows as in the continuous case [2, 7], but makes use of the discrete norm

$$\|u_h\|_{X_h}^2 := \|u_h\|_Y^2 + \|w_{u_h h}\|_Y^2 \leq \|u_h\|_Y^2 + \|w_{u_h}\|_Y^2 = \|u_h\|_X^2,$$

where $w_{u_h h} \in Y_h$ is the unique solution of the variational formulation

$$\langle \nu \nabla_y w_{u_h h}, \nabla_y z_h \rangle_{L^2(Q)} = \langle \sigma \frac{d}{dt} u_h, z_h \rangle_Q \quad \text{for all } z_h \in Y_h. \tag{5}$$

As in [7] we can then derive Cea's lemma,

$$\|u - u_h\|_{X_h} \leq 3 \inf_{z_h \in X_h} \|u - z_h\|_X,$$

from which we conclude optimal order of convergence when assuming sufficient regularity for the solution. In particular we obtain linear convergence in the space-time mesh size h when assuming $u \in H^2(Q)$, see [2, 7].

4 Numerical results

As numerical example we consider an electric motor, where both the rotor and the stator are made of iron, with 16 magnets and 48 coils, see Fig. 1.
The motor is pulled up in time, where the rotation of the rotating parts, i.e., the rotor, the magnets and the air around the magnets, is already considered within the mesh for a 90 degree rotation. The time component is treated as the third spatial dimension

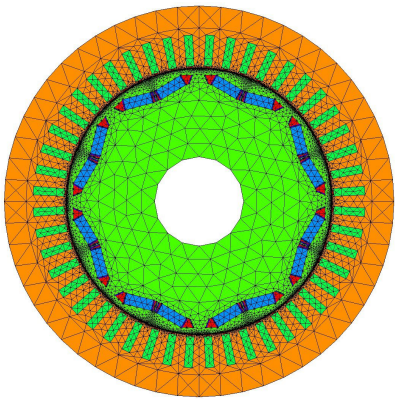


Fig. 1 The unstructured mesh of the bottom of the motor, showing the different materials.

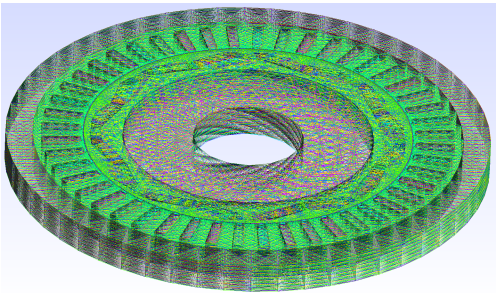


Fig. 2 The full space-time cylinder of the motor for a 90 degree rotation with 333, 288 nodes and 1, 978, 689 elements. The time is treated as the third spatial component divided into 30 time slices. The rotating parts are already considered within the mesh.

with a time span $(0, T)$, $T = 0.015$ seconds. Moreover, 30 time slices are inserted in order to have a good temporal resolution, where the mesh is differently unstructured at every time $t \in (0, T)$, see Figure 2.
The electric motor consists of isotropic materials, hence we choose the linear material parameters as given in Table 1.

Table 1 Material parameters

material	σ	ν
air	0	$10^7/(4\pi)$
coils	$5.8 \cdot 10^7$	$10^7/(4\pi)$
magnets	10^6	$10^7/(4\pi)$
iron	10^7	$10^7/(20400\pi)$

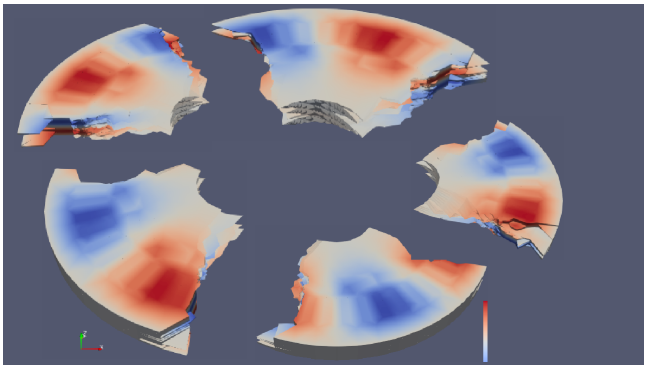


Fig. 3 Space-time mesh decomposition into 5 subdomains.

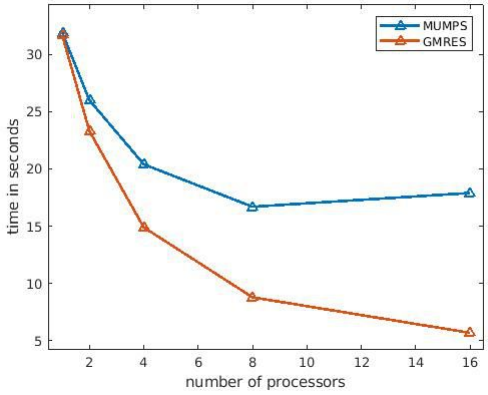


Fig. 4 The computation times for solving the linear system using the MUMPS solver and GMRES solver.

We solve the resulting linear system in parallel, using a mesh decomposition method provided by the finite element library Netgen/NGSolve[1], see Fig. 3. For our purpose, MPI parallelization is used, however the computations are done on one computer with 384 GiB RAM and two Intel Xenon Gold 5218 CPU's.
We use MUMPS[2] supported by PETSc[3], to solve the linear system. Figure 4 shows the time for solving the linear system in relation to the number of processors used for the parallel computation. For comparison, the GMRES solver provided by PETCs is used to solve the same linear systems with an error tolerance of 10^{-6} and a maximum of 1000 iterations.
The solution of the Galerkin space-time finite element formulation (4) for the time span $(0, T)$ with $T = 0.015$ is not sufficient to visualize, since in this short time the solution is close to zero due to the zero initial condition. Instead, one may consider

[1] https://www.ngsolve.org

[2] http://mumps-solver.org

[3] https://petsc.org

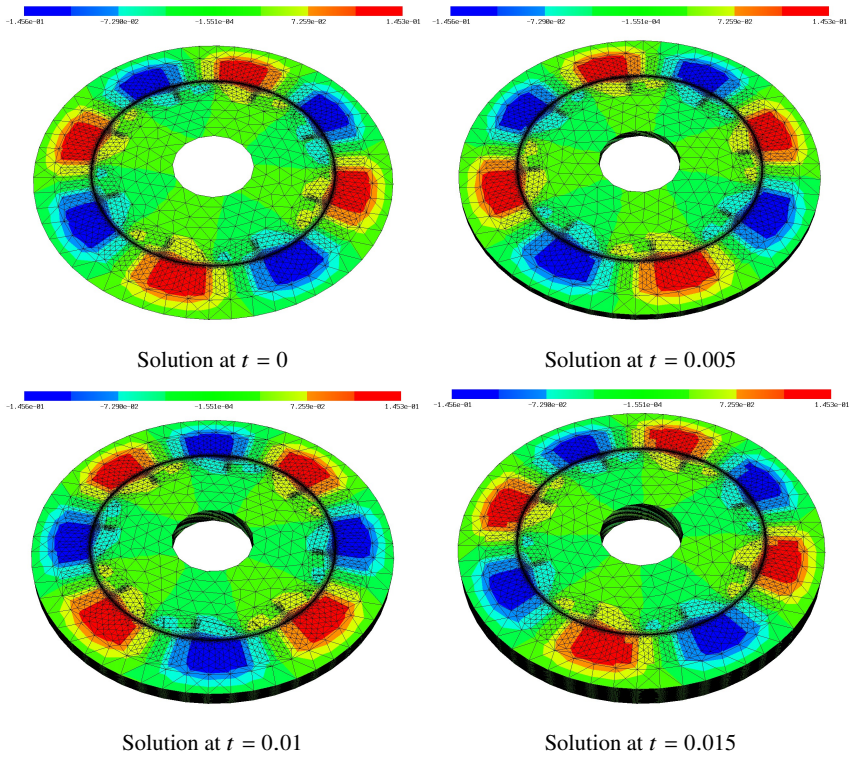


Fig. 5 The solution of the static problem visualized on different time slices.

periodic conditions $u(x, T) = u(x, 0)$ for $x \in \Omega$, see [2]. Here, we also present the results for the quasi-static problem using $\sigma = 0$ for all material regions at any time, see Fig. 5 for the solution at different time slices. Note that this corresponds to the problem of magnetostatics which is widely used in practical applications together with time-stepping methods.

5 Conclusions

In this note we have described a space-time finite element discretization of an elliptic-parabolic interface problem to model an electric motor. The computed electromagnetic fields can be used to compute other characteristic quantities such as the torque and iron losses in order to optimize the shape and the topology of electric machines. Instead of initial conditions and a linear description of the involved materials one can easily include periodic conditions in time and a nonlinear material model. Although we have provided first results for a parallel solution of the resulting linear

system of algebraic equations also using mesh decomposition algorithms, further work is required in the design of more efficient solution strategies using appropriate preconditioning and domain decomposition methods.

Acknowledgements This work has been supported by the Austrian Science Fund (FWF) under the Grant Collaborative Research Center TRR361/F90: CREATOR Computational Electric Machine Laboratory. P. Gangl acknowledges the support of the FWF project P 32911. We would like to thank U. Iben, J. Fridrich, I. Kulchytska-Ruchka, O. Rain, D. Scharfenstein, and A. Sichau (Robert Bosch GmbH, Renningen, Germany) for the cooperation and fruitful discussions during this work.

References

1. Babuška, I. and Aziz, A. K. *The Mathematical Foundations of the Finite Element Method with Applications to Partial Differential Equations*. Academic Press (1972).
2. Gangl, P., Gobrial, M., and Steinbach, O. A space-time finite element method for the simulation of rotating electric machines. Bericht 2022/9 (2022).
3. Gangl, P. and Krenn, N. Topology optimization of a rotating electric machine by the topological derivative. RICAM Report 2022/31 (2022).
4. Langer, U., Steinbach, O., Tröltzsch, F., and Yang, H. Unstructured space-time finite element methods for optimal control of parabolic equations. *SIAM J. Sci. Comput* **43**, 744–771 (2021).
5. Nečas, J. Sur une méthode pour résoudre les équations aux dérivées partielles du type elliptique, voisine de la variationnelle. *Ann. Scuola Norm. Sup. Pisa* (4), 305 – 326 (1962).
6. Neumüller, M. and Karabelas, E. Generating admissible space-time meshes for moving domains in (d+1) dimensions. *Langer, U., Steinbach, O. (eds.) Space-Time Methods. Applications to Partial Differential Equations, Radon Series on Computational and Applied Mathematics* **25**, 185–206 (2019).
7. Steinbach, O. Space-time finite element methods for parabolic problems. *Comput. Meth. Appl. Math.* **15**, 551–566 (2015).
8. Steinbach, O. and Yang, H. Space-time finite element methods for parabolic evolution equations: Discretization, a posteriori error estimation, adaptivity and solution. *Langer, U., Steinbach, O. (eds.) Space-Time Methods. Applications to Partial Differential Equations, Radon Series on Computational and Applied Mathematics* **25**, 207–248 (2019).

Reynolds-Blended Weights for BDDC in Applications to Incompressible Flows

Martin Hanek, Jakub Šístek, and Marek Brandner

1 Introduction

We investigate the applicability of the Balancing Domain Decomposition by Constraints (BDDC) method to numerical solution of problems of incompressible flows. In particular, we use BDDC to solving linear systems with a nonsymmetric matrix arising from discretization of the Navier–Stokes equations by the finite element method.

The BDDC method was introduced by Dohrmann in [1] for the Poisson problem and linear elasticity. The underlying theory for the condition number bound of $O\left(\log^2\left(1+H/h\right)\right)$ was presented by Mandel and Dohrmann in [5]. By discretizing and linearizing the Navier-Stokes equations, we get saddle-point systems with nonsymmetric matrices. An application of the BDDC method to nonsymmetric matrices arising from advection-diffusion problems was presented by Tu and Li [9], where the method was formulated without building and solving an explicit coarse problem. Finding explicit coarse basis functions and forming an explicit coarse problem of BDDC was presented by Yano for nonsymmetric problems arising from the Euler equations in [10]. A three-level extension of BDDC was presented by Tu [8], while a general multilevel method was introduced and analysed for symmetric positive definite problems by Mandel et al. [6]. We have extended the multilevel BDDC method

Martin Hanek
Institute of Mathematics of the Czech Academy of Sciences, Žitná 25, Prague, Czech Republic,
Czech Technical University in Prague, Technická 4, Prague, Czech Republic,
e-mail: martin.hanek@fs.cvut.cz

Jakub Šístek
Institute of Mathematics of the Czech Academy of Sciences, Žitná 25, Prague, Czech Republic,
e-mail: sistek@math.cas.cz

Marek Brandner
University of West Bohemia in Pilsen, Faculty of Applied Sciences, Univerzitní 22, Pilsen, Czech Republic, e-mail: brandner@kma.zcu.cz

to nonsymmetric matrices in [3]. A theoretically supported approach for handling continuous pressure in the Stokes problem was introduced in [4].

An important building block of BDDC as well as other nonoverlapping domain decomposition methods is the choice of weights used for averaging a discontinuous solution at the interface between subdomains. Standard types of weights include an arithmetic average (also known as cardinality scaling), or weighted average based on diagonal entries of subdomain matrices. In [3], we have also presented a novel averaging operator tailored to Navier-Stokes equations. The main idea behind it is using the current approximation of velocity for preferring information opposite the flow. Due to the similarity of this idea with numerical methods for convection dominated flows, we called this choice as the upwind scaling.

In this contribution, we present a modification of the upwind scaling. While the upwind scaling is superior for flows at higher Reynolds numbers, the simple arithmetic scaling tends to perform better for flows at lower Reynolds numbers. For this reason, we 'blend' the arithmetic and upwind scalings with the ratio based on the local Reynolds number, and we call the proposed method as *Reynolds-blended (Re-blended)* weights.

The rest of the paper is organized as follows. In Section 2, we recall the basics of iterative substructuring and BDDC for the nonsymmetric saddle-point systems arising from the finite element method (FEM). The new weights are proposed in Section 3. Section 4 presents results of numerical experiments showing the benefits of the Re-blended weights, while Section 5 is devoted to the summary.

2 FEM and BDDC for Navier-Stokes equations

We consider a stationary incompressible flow in a bounded three-dimensional domain $\Omega \subset \mathbb{R}^3$ with its boundary $\partial\Omega$ consisting of two disjoint parts $\partial\Omega_D$ and $\partial\Omega_N$, governed by the Navier-Stokes equations (see e.g. [2]),

$$(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu\Delta\boldsymbol{u} + \nabla p = \boldsymbol{f} \quad \text{in } \Omega, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0 \quad \text{in } \Omega, \tag{2}$$

where $\boldsymbol{u}$ is the velocity vector of the fluid, ν is the kinematic viscosity of the fluid, p is the kinematic pressure, and $\boldsymbol{f}$ is the vector of body forces. In addition, we consider the following boundary conditions: prescribed velocity on $\partial\Omega_D$ and $-\nu(\nabla\boldsymbol{u})\boldsymbol{n} + p\boldsymbol{n} = \boldsymbol{0}$ on $\partial\Omega_N$, with $\boldsymbol{n}$ being the unit outer normal vector of $\partial\Omega$.

We consider Taylor-Hood Q2-Q1 elements, and after substituting linear combinations of the basis functions, we get the following system of algebraic equations

$$\begin{bmatrix} \nu\mathbf{A} + \mathbf{N}(\mathbf{u}) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}. \tag{3}$$

Details can be found in [3].

System (3) is nonlinear due to the matrix $\mathbf{N}(\mathbf{u})$, and we consider the Picard iteration for its linearization. This leads to solving a sequence of linear systems of equations in the form

$$\begin{bmatrix} \nu\mathbf{A} + \mathbf{N}(\mathbf{u}^p) & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}^{p+1} \\ \mathbf{p}^{p+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}. \tag{4}$$

Linear system (4) is solved by means of iterative substructuring (see, e.g., [7]). In order to use the BDDC method, we decompose the solution domain Ω into N nonoverlapping subdomains. Then we reduce the system (4) to the interface to get

$$S \begin{bmatrix} \mathbf{u}_\Gamma \\ \mathbf{p}_\Gamma \end{bmatrix} = g, \tag{5}$$

where S is the Schur complement of the interior unknowns and g is the reduced right-hand side.

Problem (5) is solved by the BiCGstab method using one step of BDDC as the preconditioner. In each action of the BDDC preconditioner, a coarse problem and independent subdomain problems are solved. Before solving it in each iteration, we need to set-up the preconditioner. This is performed by solving two saddle-point systems

$$\begin{bmatrix} S_i & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} \Psi_i \\ \Lambda_i \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} \qquad \begin{bmatrix} S_i^T & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} \Psi_i^* \\ \Lambda_i^T \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{6}$$

where S_i is the Schur complement with respect to the interface of the i-th subdomain, C_i is the matrix defining coarse degrees of freedom, which has as many rows as is the number of coarse degrees of freedom defined at the subdomain. The solution Ψ_i is the matrix of *coarse basis functions* with every column corresponding to one coarse unknown on the subdomain. These functions are equal to one in one coarse degree of freedom, and they are equal to zero in the remaining local coarse unknowns. The solution Ψ_i^* is the matrix of *adjoint coarse basis functions* which is needed for nonsymmetric problems as was shown in [10]. The coarse problem matrix is assembled in the setup of the BDDC preconditioner as $S_C = \sum_{i=1}^N R_{Ci}^T \Psi_i^{*T} S_i \Psi_i R_{Ci}$.

One step of the BDDC preconditioner $M_{BDDC} : r^l \to u_\Gamma^l$ proceeds as follows:

$$r_i^l = W_i R_i r^l$$

coarse problem

$$r_C^l = \sum_{i=1}^N R_{Ci}^T \Psi_i^{*T} r_i^l$$

$$S_C u_C = r_C^l$$

$$u_{Ci} = \Psi_i r_{Ci}^l u_C$$

subdomain problems

$$\begin{bmatrix} S_i & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} u_i \\ \lambda \end{bmatrix} = \begin{bmatrix} r_i^l \\ 0 \end{bmatrix}$$

$$u_\Gamma^l = \sum_{i=1}^N R_i^T W_i (u_i + u_{Ci}),$$

where R_i is an operator restricting a global interface vector to the i-th subdomain, R_{Ci} is the restriction of the global vector of coarse unknowns to those present at the i-th subdomain, and matrix W_i applies weights to satisfy the partition of unity, which will be elaborated in the next section. Details of the application of this method to Navier-Stokes equations can be found in [3].

3 Weight operators

Let us now discuss several particular choices of the matrix of weights W_i. An important class of these matrices is represented by diagonal matrices

$$W_i = \begin{pmatrix} W_{iN}^1 & & \\ & W_{iN}^2 & \\ & & \ddots \end{pmatrix}, \tag{7}$$

where W_{iN}^k denotes the weight matrix for the unknowns in the k-th (with respect to the subdomain interface) node of the i-th subdomain. These matrices differ for nodes with just velocity unknowns and those containing also a pressure unknown ordered after the velocity ones. For example, in 3D the former and latter looks respectively as

$$W_{iN}^k = \begin{pmatrix} w_i^k & & \\ & w_i^k & \\ & & w_i^k \end{pmatrix}, \qquad W_{iN}^k = \begin{pmatrix} w_i^k & & & \\ & w_i^k & & \\ & & w_i^k & \\ & & & \frac{1}{N_S} \end{pmatrix}, \tag{8}$$

where N_S is the number of subdomains sharing the node.

A general scheme for constructing these matrices satisfying the partition of unity can be described in the following way. Every subdomain first generates a nonnegative weight $\widetilde{w}_i^k$. These values are then shared with all neighbouring subdomains, and the normalized weight w_i^k satisfying the partition of unity is obtained by dividing the local weight with the sum of contributions from all neighbours,

$$w_i^k = \frac{\widetilde{w}_i^k}{\sum_{j=1}^{N_S} \widetilde{w}_j^k}. \tag{9}$$

The first type of weights is based on the cardinality (*card*) of the set of subdomains sharing the node. Hence, $\widetilde{w}_i^k = 1$, and

$$w_i^k = \frac{1}{N_S}. \tag{10}$$

For example, the weight is simply $w_i^k = 1/2$ if the node is shared by two subdomains.

The second type of weights was introduced in [3], and it is inspired by numerical schemes for flow problems, namely by upwinding. The underlying idea is that for dominant advection, it should be beneficial to consider the subdomain from which the fluid flows with a higher weight than for the one where the node is a part of an inflow boundary.

More specifically, these *upwind* weights are based on the inner product of the vector of velocity at the k-th interface node $\boldsymbol{u}^k$ and the unit vector of the outer normal to the i-th subdomain boundary $\boldsymbol{n}_i^k$, therefore

$$p_i^k = \frac{\boldsymbol{u}^k \cdot \boldsymbol{n}_i^k}{\left\|\boldsymbol{u}^k\right\|_2}.$$

The values of the p_i^k are from the interval $[-1, 1]$. To derive a nonnegative weight, these values are mapped to the interval $[0, 1]$ by taking $\widetilde{w}_i^k = \frac{p_i^k+1}{2}$, which is used for all velocity unknowns. More details, such as the discrete construction of $\boldsymbol{n}_i^k$, can be found in [3].

The third type is the new approach obtained by linear interpolation of the previous two weights. For this method, we choose a critical Reynolds number Re_C, and then the resulting Reynolds-blended ($\mathrm{Re}_{\mathrm{blended}}$) weight is defined according to the local Reynolds number $\mathrm{Re}_{\mathrm{loc}} = |\boldsymbol{u}^k|L/\nu$ as

$$\widetilde{w}_i^k = \begin{cases} \widetilde{w}_{\mathrm{card}}^k & \text{for } \mathrm{Re}_{\mathrm{loc}} \le 1, \\ \dfrac{\mathrm{Re}_{\mathrm{loc}}}{\mathrm{Re}_C}\,\widetilde{w}_{\mathrm{upwind}}^k + \left(1 - \dfrac{\mathrm{Re}_{\mathrm{loc}}}{\mathrm{Re}_C}\right)\widetilde{w}_{\mathrm{card}}^k & \text{for } 1 < \mathrm{Re}_{\mathrm{loc}} < \mathrm{Re}_C, \\ \widetilde{w}_{\mathrm{upwind}}^k & \text{for } \mathrm{Re}_{\mathrm{loc}} \ge \mathrm{Re}_C. \end{cases} \tag{11}$$

Here L corresponds to the characteristic length of the problem. Thus for small local Reynolds numbers, the scaling behaves as cardinality weights and for high Reynolds number as upwind weights depending on the chosen critical Reynolds number Re_C. Note that these weights are updated after each nonlinear iteration.

4 Numerical results

In this section, we compare the behaviour of the 2-level BDDC method for different types of interface weights described in Section 3, namely the cardinality scaling (*card*), *upwind*, and the proposed $\mathrm{Re}_{\mathrm{blended}}$ weights. We assume two problems, namely the lid-driven cavity and the backward facing step problems. First we look at the cavity problem. We consider unit cube with unit velocity on the top wall as in [2]. For $\mathrm{Re}_{\mathrm{blended}}$, we consider two critical Reynolds numbers, $\mathrm{Re}_C = 100$ and $\mathrm{Re}_C = 200$. For these simulations, the number of subdomains is 125 with 8 elements per subdomain edge. The decomposed solution domain can be seen in Fig. 1. For this problem, Reynolds number is defined as $\mathrm{Re} = |\boldsymbol{u}_{\mathrm{top}}|L/\nu$, where $|\boldsymbol{u}_{\mathrm{top}}| = 1$ is the velocity at the lid, and $L = 1$ is the cube size. We compare $\mathrm{Re} = 1$ and $\mathrm{Re} = 200$

monitoring the number of nonlinear iterations, the minimal, maximal, and mean number of linear iterations over all nonlinear iterations, the mean setup time of the BDDC preconditioner, the mean time for the Krylov subspace method with the mean time for one linear iteration, the mean time for one nonlinear iteration, and the time for all nonlinear iterations.

The computations are performed on the *Karolina* supercomputer at the IT4I National Supercomputing Centre in Ostrava, Czech Republic. The computational nodes are equipped with two 64-core AMD 7H12 2.6 GHz processors, and 256 GB RAM. The values are presented in Tables 1 and 2.

Table 1 Re = 1. Number of nonlinear iterations, number of linear iterations (minimal, maximal, and mean), mean setup time, time for the BiCGstab iterations, time for one linear iteration, time for one nonlinear iteration, and the total time for solving the nonlinear problem.

weights type	nonl	linear solve			time [s]			
		min	max	mean	setup	BiCGstab iter (one iter)	nonl	total
card	4	13.5	13.5	**13.5**	4.50	4.07 (0.30)	8.57	32.28
upwind	4	13.5	18.5	**17.3**	4.53	5.20 (0.30)	9.73	38.92
Re-blended ($\mathrm{Re}_C = 100$)	4	13.5	13.5	**13.5**	4.58	4.09 (0.30)	8.67	34.68
Re-blended ($\mathrm{Re}_C = 200$)	4	13.5	13.5	**13.5**	4.51	4.10 (0.30)	8.61	34.44

Table 2 Re = 200. Number of nonlinear iterations, number of linear iterations (minimal, maximal, and mean), mean setup time, time for the BiCGstab iterations, time for one linear iteration, time for one nonlinear iteration, and the total time for solving the nonlinear problem.

weights type	nonl	linear solve			time [s]			
		min	max	mean	setup	BiCGstab iter (one iter)	nonl	total
card	29	14	91.5	**86.2**	5.01	27.99 (0.32)	33.0	1517.2
upwind	29	14	85.5	**34.2**	4.99	10.79 (0.32)	15.78	1029.1
Re-blended ($\mathrm{Re}_C = 10$)	29	14	85.5	**34.0**	4.99	10.76 (0.32)	15.75	1028.6
Re-blended ($\mathrm{Re}_C = 100$)	29	14	89	**137.8**	4.99	12.15 (0.32)	17.14	1069.5
Re-blended ($\mathrm{Re}_C = 200$)	29	14	137.5	**58.1**	5.05	18.22 (0.31)	23.27	1240.44

From Tables 1 and 2, we can see that for small Reynolds numbers, the cardinality weight is slightly more efficient and for the high Reynolds number the same stands for the upwind weight. The critical Reynolds weight seems to benefit from both depending on the Reynolds number. For Re = 1, it inclines to the cardinality and for Re = 200 to the upwind weight.

Let us now explore the effect of the Reynolds-blended weight on the backward facing step problem. This problem was investigating in [2] in 2D. The solution domain is shown in Fig. 1 with prescribed unit inlet velocity, zero velocity on the top and bottom walls, and symmetry boundary condition on the side walls. With the x-axis aligned with the flow, the step occurs for $x = 1$, where the height changes from 1 to 2. The length of the domain is 5, and its width is 1. The solution domain consists of 37 thousand elements which correspond to 978 thousand unknowns. The mesh is decomposed into 32 subdomains using a vertical partitioner, which cuts the domain along the x direction (see Fig. 1). The Reynolds number is defined as $\mathrm{Re} = |\boldsymbol{u}_{\mathrm{inlet}}|L/\nu$, where $|\boldsymbol{u}_{\mathrm{inlet}}| = 1$ is the input velocity, and $L = 1$ is the size of the narrow part.

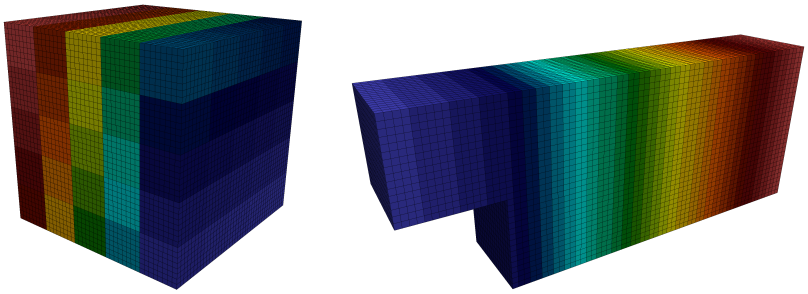


Fig. 1 Decomposed solution domain for the cavity problem (left) and for the backward facing step problem (right).

We set the critical Reynolds number for our new weight to 20 and plot the mean number of linear iterations and the average time for solving one linearized problem (4) depending on Reynolds number in Fig. 2.

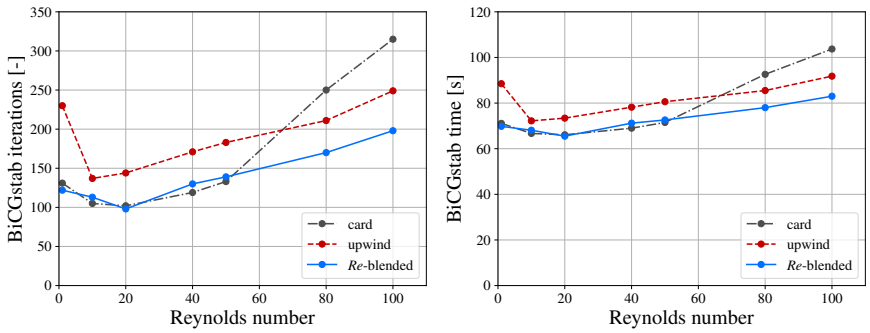


Fig. 2 Number of BiCGstab iterations (left) and average time for solving one linearized problem (right) for different Reynolds numbers for cardinality, upwind, and Re-blended operators.

From these plots we can see that up to a certain Re, cardinality performs better while for larger Re, the upwind is more effective. The Reynolds-blended weight operator with a suitably chosen critical Reynolds number Re_C provides the best results for almost every Re, and therefore it again combines advantages of cardinality and upwind weight operator. Interestingly, it even outperforms the upwind weight operator. This positive effect is attributed to the fact that the blending based on the local Reynolds number $\mathrm{Re}_{\mathrm{loc}}$ reduces the effect of upwinding in zones with reduced velocity such as in boundary layers.

5 Conclusions

We have presented a new scaling operator for the BDDC method in applications to saddle-point linear systems arising from discretization of the Navier-Stokes equa-

tions. It can be seen as a correction of the recent upwind operator when applied to flows with low Reynolds numbers, for which arithmetic scaling is superior. We have compared the relevant weight operators on the cavity and the backward facing step problems. The results demonstrate the intended behaviour of the new scaling, namely mimicking the arithmetic averaging for low Re and the upwind scaling for high Re. Although our simulations show promising results for the considered small and moderate Reynolds numbers, for larger Re some kind of stabilization of the discretization would be needed. Investigating the performance of the new method for other flow problems and the choice of the Re_C parameter will be a matter of our future research.

Acknowledgements This research was supported by the Czech Science Foundation through grants GAČR 20-01074S and GAČR 23-06159S, by the Czech Academy of Sciences through RVO:67985840, and by the Czech Technical University in Prague through RVO:12000. Computational time on the Karolina supercomputer has been provided thanks to the support of the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90140).

References

1. Dohrmann, C. R. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.* **25**(1), 246–258 (2003).
2. Elman, H. C., Silvester, D. J., and Wathen, A. J. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York (2005).
3. Hanek, M., Šístek, J., and Burda, P. Multilevel BDDC for incompressible Navier-Stokes equations. *SIAM J. Sci. Comput.* **42**(6), C359–C383 (2020).
4. Li, J. and Tu, X. A nonoverlapping domain decomposition method for incompressible Stokes equations with continuous pressures. *SIAM J. Numer. Anal.* **51**(2), 1235–1253 (2013).
5. Mandel, J. and Dohrmann, C. R. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numer. Linear Algebra Appl.* **10**(7), 639–659 (2003).
6. Mandel, J., Sousedík, B., and Dohrmann, C. R. Multispace and multilevel BDDC. *Computing* **83**(2-3), 55–85 (2008).
7. Toselli, A. and Widlund, O. B. *Domain Decomposition Methods—Algorithms and Theory*, *Springer Series in Computational Mathematics*, vol. 34. Springer-Verlag, Berlin (2005).
8. Tu, X. Three-level BDDC in three dimensions. *SIAM J. Sci. Comput.* **29**(4), 1759–1780 (2007).
9. Tu, X. and Li, J. A balancing domain decomposition method by constraints for advection-diffusion problems. *Commun. Appl. Math. Comput. Sci* **3**(1), 25–60 (2008).
10. Yano, M. *Massively Parallel Solver for the High-Order Galerkin Least-Squares Method*. Master's thesis, Massachusests Institute of Technology (2009).

Neural Network Interface Condition Approximation in a Domain Decomposition Method Applied to Maxwell's Equations

Tobias Knoke, Sebastian Kinnewig, Sven Beuchler, and Thomas Wick

1 Introduction

The time-harmonic Maxwell equations are of great interest in current research fields, e.g., [7, 8, 10, 14, 16]. As their numerical solution is challenging due to their ill-posed nature, e.g., [2], suitable techniques need to be applied. The most prominent technique in literature is based on domain decomposition techniques [5, 17]. The work of Hiptmair [10] can only be applied for the problem in the time domain (i.e., the well-posed problem).

In this work, we design a proof of concept to approximate the interface operator with the help of a feedforward neural network [3, 9, 12]. To this end, a two-domain problem is designed, which is then trained by exchanging data from a modern finite element library deal.II [1] and the well-known PyTorch [15] library. Our main aim is to showcase that our approach is feasible and can be a point of departure for detailed future investigations. An extended version with more technical details and additional computations is [13].

The outline of this work is as follows: In Section 2 we introduce the time-harmonic Maxwell's equations and our notation. Next, in Section 3, domain decomposition and neural network approximations are introduced. Afterward, we address in detail the training process in Section 4. In Section 5, numerical tests demonstrate our proof of concept.

Tobias Knoke, Sebastian Kinnewig, Sven Beuchler, Thomas Wick
Leibniz University Hannover, Institute of Applied Mathematics, Welfengarten 1, 30167 Hannover, Germany, e-mail: tobias.knoke@stud.uni-hannover.de,
{beuchler,kinnewig,thomas.wick}@ifam.uni-hannover.de
and Cluster of Excellence PhoenixD (Photonics, Optics, and Engineering - Innovation Across Disciplines), Leibniz Universität Hannover, Germany

2 Equations

Let $\Omega \subset \mathbb{R}^2$ (here dimension 2, but usually we deal with dimension 3 in Maxwell's equations) be a bounded domain with sufficiently smooth boundary Γ. The latter is partitioned into $\Gamma = \Gamma^{\infty} \cup \Gamma^{\text{inc}}$. Furthermore, the time-harmonic Maxwell equations are then defined as follows: Find the electric field E such that

$$\begin{cases} \operatorname{curl}\left(\mu^{-1}\operatorname{curl}(E)\right) - \omega^2 E & = 0 & \text{in } \Omega \\ \mu^{-1}\gamma^t(\operatorname{curl}(E)) - i\omega\gamma^T(E) & = 0 & \text{on } \Gamma^{\infty} \\ \gamma^T(E) & = \gamma^T\left(E^{\text{inc}}\right) & \text{on } \Gamma^{\text{inc}}, \end{cases} \tag{1}$$

where $E^{\text{inc}}\colon \mathbb{R}^2 \to \mathbb{C}^2$ is some given incident electric field, $\omega > 0$ is the wave number, μ is the relative permeability and i denotes the imaginary number. For the weak form and corresponding definitions, we seek $E \in H(\operatorname{curl}, \Omega) := \{v \in \mathcal{L}^2(\Omega) \mid \operatorname{curl}(v) \in \mathcal{L}^2(\Omega)\}$. The traces $\gamma^t\colon H(\operatorname{curl}, \Omega) \to H_{\times}^{-1/2}(\operatorname{div}, \Gamma)$ and $\gamma^T : H(\operatorname{curl}, \Omega) \to H_{\times}^{-1/2}(\operatorname{curl}, \Gamma)$ are defined by

$$\gamma^t(v) = n \times v \quad \text{and} \quad \gamma^T(v) = n \times (v \times n),$$

where $n \in \mathbb{R}^2$ is the normal vector of Ω, $H_{\times}^{-1/2}(\operatorname{div}, \Gamma) := \{v \in H^{-1/2}(\Gamma) \mid v \cdot n = 0, \operatorname{div}_{\Gamma} v \in H^{-1/2}(\Gamma)\}$ is the space of well-defined surface divergence fields and $H(\operatorname{curl}, \Gamma) := \{v \in H^{-1/2}(\Gamma) \mid v \cdot n = 0,\ \operatorname{curl}_{\Gamma}(v) \in H^{-1/2}(\Gamma)\}$ is the space of well-defined surface curls. System (1), as well as its weak form (not shown here), is called time-harmonic, because the time dependence can be expressed by $e^{i\omega\tau}$, where $\tau \geq 0$ denotes the time.

For the implementation with the help of a Galerkin finite element method (FEM), we need the discrete weak form. Based on the De-Rham cohomology, we need to choose our basis functions out of the Nédélec space $\mathcal{N}_h^p(\Omega)$. For the description of the Nédélec space we refer to [4]. The discrete form is given by, find $E_h \in \mathcal{N}_h^p(\Omega)$ such that

$$\begin{aligned} \int_{\Omega} \mu^{-1}\operatorname{curl}(E_h)\operatorname{curl}(\Phi_h) - \omega^2 E_h \Phi_h \,\mathrm{d}x + \int_{\Gamma^{\infty}} i\omega\gamma^T(E_h)\,\gamma^T(\Phi_h)\,\mathrm{d}s \\ = \int_{\Gamma^{\infty}} \gamma^T(E_h^{inc})\gamma^T(\Phi_h)\,\mathrm{d}s \;\forall \Phi_h \in \mathcal{N}_h^p(\Omega). \end{aligned} \tag{2}$$

For a more in-depth derivation of equations (1) and their discretization see [14].

3 Numerical approach

3.1 Domain decomposition

Since the solution of the Maxwell equation system (1) is challenging, we apply a non-overlapping domain composition method (DDM) in which the domain is divided into subdomains as follows

$$\Omega = \bigcup_{i=0}^{n_{\text{dom}}} \Omega_i \quad \text{with}$$
$$\Omega_i \cap \Omega_j = \varnothing \quad \forall i \neq j,$$

in such a way, that every subdomain Ω_i becomes small enough, so it can be handled with a direct solver. The global solution of the electric field E is obtained via an iterative method, where we solve the time-harmonic Maxwell's equations on each subdomain with suitable interface conditions between the different subdomains. So we obtain a solution E_i^k for every subdomain Ω_i, where k denotes the k-th iteration step. The initial interface condition is given by

$$g_{ji}^{k=0} := -\mu^{-1}\gamma_i^t\left(\text{curl}\left(E_i^{k=0}\right)\right) - ikS\left(\gamma_i^T\left(E_i^{k=0}\right)\right) = 0, \tag{3}$$

where S describes the surface operator [6]. Please note that in ikS, i denotes the imaginary number, while as subscript, i is an index. Afterwards, the electric-field E_i^{k+1} is computed at each step by solving the following system

$$\begin{cases} \text{curl}\left(\mu^{-1}\text{curl}\left(E_i^{k+1}\right)\right) - \omega^2 E_i^{k+1} & = 0 & \text{in } \Omega_i \\ \mu^{-1}\gamma_i^t\left(\text{curl}\left(E_i^{k+1}\right)\right) - i\omega\gamma_i^T\left(E_i^{k+1}\right) & = 0 & \text{on } \Gamma_i^{\infty} \\ \gamma_i^T\left(E_i^{k+1}\right) & = \gamma_i^T\left(E_i^{\text{inc}}\right) & \text{on } \Gamma_i^{\text{inc}} \\ \mu^{-1}S\left(\gamma_i^t\left(\text{curl}\left(E_i^{k+1}\right)\right)\right) - i\omega\gamma_i^T\left(E_i^{k+1}\right) & = g_{ji}^k & \text{on } \Sigma_{ij}, \end{cases} \tag{4}$$

where $\Sigma_{ij} = \Sigma_{ji} := \partial\Omega_i \cap \partial\Omega_j$ denotes the interface of two neighbouring elements and the interface condition is updated by

$$g_{ji}^{k+1} = -\mu^{-1}\gamma_i^t\left(\text{curl}\left(E_i^{k+1}\right)\right) - ikS\left(\gamma_i^T\left(E_i^{k+1}\right)\right) = -g_{ij}^k - 2ikS\left(\gamma_i^T\left(E_i^{k+1}\right)\right). \tag{5}$$

In case of success we obtain $\lim_{k\to\infty} E_i^k = E|_{\Omega_i}$, but this convergence depends strongly on the chosen surface operator S (see [5, 6]).

3.2 Neural network approximation

Since the computation of a good approximation of S is challenging, we examine a new approach in which we attempt to approximate this operator with the help of a neural network (NN). For a first proof of concept, we choose a prototype example and explore whether at all an NN can approximate the values on the interfaces. As it is not feasible to compute the exact surface operator S, we aim to compute g_{ij}^{k+l}, $l > 0$ with an NN, where we use g_{ij}^{k} and E_i^{k+1} as input. Another benefit of this approach is that we can generate easily a training data set from a classical domain decomposition method, as described in section 4.3. For simplicity, we choose $S = \mathbb{1}$ inside of our classical domain decomposition method. Hence, the advantage of this approach is that the interface condition can be updated without recomputing the system (4) at each step, rising hope to reducing the computational cost.

4 Neural network training

In this section, we describe the training process. Besides the mathematical realization, we also need to choose the software libraries. For computing the time-harmonic Maxwell equations with the finite element method (FEM), we utilize deal.II [1]. The neural network is trained with PyTorch [15].

4.1 Decomposing the domain

Before we construct the NN, we choose the domain, the decomposition and the grid on which the system (4) is solved to obtain the training values, because they will influence the size of the network. The domain in our chosen example, given by $\Omega = (0,1) \times (0,1)$ is divided into two subdomains $\Omega_0 = (0,1) \times (0,0.5)$ and $\Omega_1 = (0,1) \times (0.5,1)$, see Figure 1 and the grid (obtained from two times uniform refinement) on which the FEM is applied is a mesh of 32×32 elements with quadratic Nédélec elements.

Hence, 32 elements with each 4 degrees of freedom (dofs) are located on the interface in both subdomains. We evaluate the interface condition and the solution on each dof and use the values as the input and the target of the NN. Therefore the input contains $4 \cdot \dim(g_{ij}) + 4 \cdot \dim(E_i) = 16$ values and the output consists of $4 \cdot \dim(g_{ji}) = 8$ values and we obtain 32 input-target pairs with one computation.

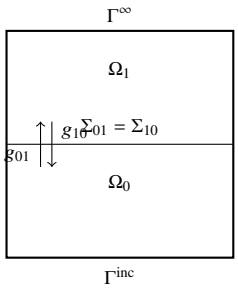


Fig. 1 Visualization of the domain Ω with the chosen decomposition

4.2 Neural network construction

Regarding the previous considerations, we need an input layer with 16 neurons and an output layer with 8 neurons. Furthermore, we use one hidden layer with 500 neurons. Our tests revealed, that this is a sufficent and reasonable size for our purpose, since it leads to more effective networks in terms of error minimization and training duration than other sizes we tested (e.g. 50, 100 and 300 neurons in the hidden layer). The activation functions used are the sigmoid function given by $f(x) = (1 + e^{-x})^{-1}$ in the hidden layer, which turned out to be the most effective among those we tested (e.g. $\tanh(x)$, $\log\left((1 + e^{-x})^{-1}\right)$ and $\max(0, x) + \min(0, e^x - 1)$) and the identity in the other layers. Moreover, we apply separate networks N_{01} and N_{10} of the same shape for both interface conditions g_{01} and g_{10}, since it turned out that they are approximated differently fast and accurately.

4.3 Training

To obtain enough training data, we vary the boundary condition E^{inc} and create a set of training values and a set of test values to control the network during the training and avoid overfitting. The training set and the test set are generated by the boundary values that are displayed in Table 1.

Table 1 Boundary values for generating the training set and the test set

E^{inc} for the training set		E^{inc} for the test set
$\begin{pmatrix} e^{\frac{-(x-0.7)^2}{0.008}} \\ 0 \end{pmatrix}$	$\begin{pmatrix} \cos(\pi^2 y) + \sin(\pi^2 x)i \\ \sin(\pi^2 y) + 0.5\cos(\pi^2 x)i \end{pmatrix}$	$\begin{pmatrix} e^{\frac{-(x-0.5)^2}{0.003}} \\ 0 \end{pmatrix}$
$\begin{pmatrix} e^{\frac{-(x-0.2)^2}{0.002}} \\ 1 \end{pmatrix}$	$\begin{pmatrix} \sin(\pi^2 x) + \sin(\pi^2 x)i \\ \sin(\pi^2 y) + 0.5\cos(\pi^2 x)i \end{pmatrix}$	$\begin{pmatrix} \cos(\pi^2 y) + \sin(\pi^2 x)i \\ \cos(\pi^2 y) + 0.5\cos(\pi^2 x)i \end{pmatrix}$
$\begin{pmatrix} e^{\frac{-(x-0.7)^2}{0.003}} \\ 1 \end{pmatrix}$	$\begin{pmatrix} \sin(\pi^2 x) + \sin(\pi^2 x)i \\ \sin(\pi^2 x) + 0.5\cos(\pi^2 x)i \end{pmatrix}$	
$\begin{pmatrix} e^{\frac{-(x-0.8)^2}{0.003}} \\ \sin(\pi^2 x) \end{pmatrix}$	$\begin{pmatrix} \cos(\pi^2 y) + \sin(\pi^2 x)i \\ \cos(\pi^2 x) + 0.5\cos(\pi^2 x)i \end{pmatrix}$	
$\begin{pmatrix} e^{\frac{-(x-0.5)^2}{0.003}} \\ \cos(\pi^2 x) \end{pmatrix}$	$\begin{pmatrix} \cos(\pi^2 x) + \sin(\pi^2 x)i \\ \cos(\pi^2 y) + 0.5\cos(\pi^2 x)i \end{pmatrix}$	

Since we choose 10 different boundary values for the training set and 2 for the test set and each of them generates a set of 32 training/test values (one per element on the interface), we obtain all in all a set of $32 \cdot 10 = 320$ input-target pairs (each with with a total of $16 + 8 = 24$ values) for the training and a test set of $32 \cdot 2 = 64$ input-target pairs for both networks. To keep the computations simple, we choose a small wave number $\omega = \epsilon\frac{2\pi}{3}$, where ϵ denotes the relative permittivity, and compute the sets with the iterative DDM in 4 steps. Afterwards we use the results $\left(g_{ij}^1, E_i^2\right)$

and g^3_{ji} as the input and the targets to train our NNs with the application of the mean squared error as the loss function and the Adam algorithm [11] as the optimizer. The network N_{01} is trained with the learning rate 10^{-5}. The initial training error 3.12 and the test error 5.87 are reduced to $1.7 \cdot 10^{-4}$ and $3 \cdot 10^{-3}$ after 29 843 steps of the optimization method. At N_{10} the initial training error 0.72 and the test error 1.28 are reduced to $3 \cdot 10^{-4}$ and $4 \cdot 10^{-3}$ after 20 326 steps with learning rate 10^{-5} and after further training with learning rate 10^{-6} in 3706 steps, we finally achieve the training error $2.9 \cdot 10^{-4}$ and the test error $3 \cdot 10^{-3}$.

5 Numerical tests

In this section, we apply the implemented and trained NNs for different numerical examples. For the first example, we choose the following boundary condition

$$E^{\text{inc}}(x, y) = \begin{pmatrix} \cos\left(\pi^2 (y - 0.5)\right) + \sin\left(\pi^2 x\right) i \\ \cos\left(\pi^2 y\right) + 0.5 \sin\left(\pi^2 x\right) i \end{pmatrix},$$

and compute the first interface conditions g^1_{10} and g^1_{01} and the solutions E^1_1 and E^1_0 by solving (4) and (5) once. Afterwards, these values are passed on to the networks N_{01}

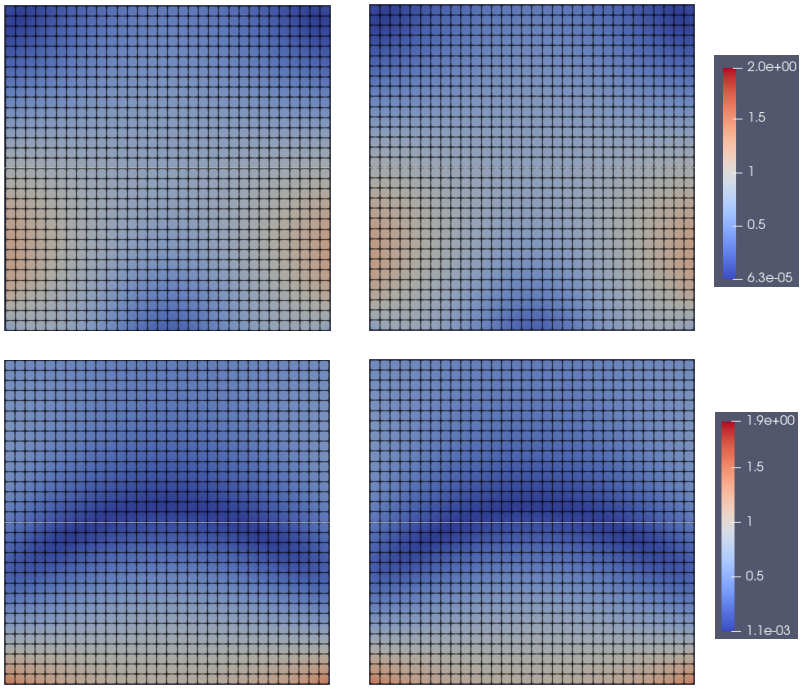


Fig. 2 First example: Real part (above) and imaginary part (below) of the NN solution (left) and the DDM solution (right)

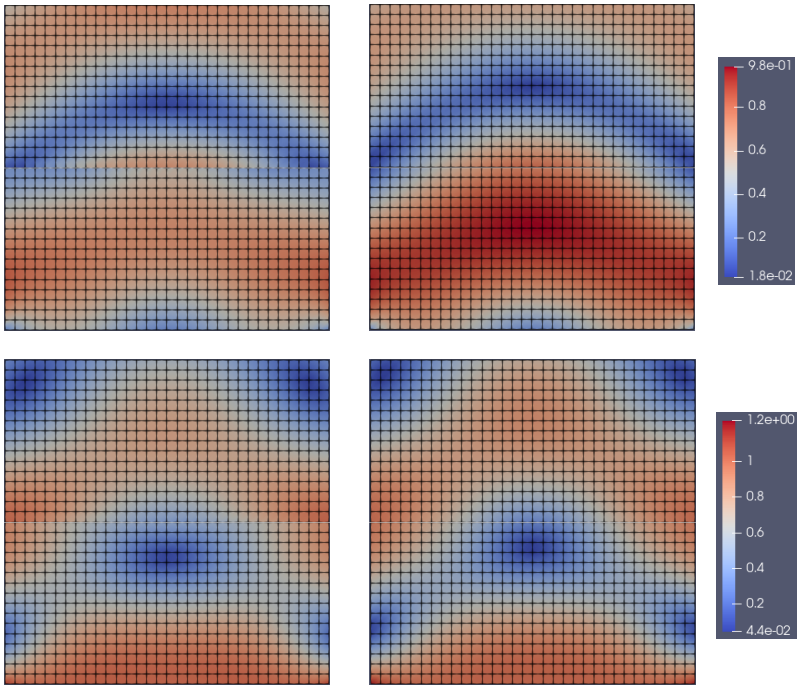


Fig. 3 Second example: Real part (above) and imaginary part (below) of the NN solution (left) and the DDM solution (right)

and N_{10}. The output they return is then handled as our new interface condition which we use to solve system (4) one more time. With that, we obtain the final solution. Moreover, we compute the same example with the DDM in 4 steps. The results that are displayed in Figure 2 show excellent agreement.

As a second example, we increase the wave number, which leads to a more complicated problem. Therefore we repeat the same computation with $\omega = \epsilon\pi$ and leave the other parameters (especially the networks) unchanged. In contrast to the previous example, the results that are displayed in Figure 3 show differences. While the imaginary part is still well approximated, the real part of the NN solution differs significantly from the DDM solution and shows a discontinuity on the interface.

6 Conclusion

In this contribution, we provided a proof of concept and feasibility study for a neural network approximation of the interface conditions in domain decomposition. Analyzing our numerical tests, it can be inferred that the approach works for two subdomains. Ongoing work is the extension to more subdomains.

Acknowledgements This work is funded by the Deutsche Forschungsgemeinschaft (DFG) under Germany's Excellence Strategy within the Cluster of Excellence PhoenixD (EXC 2122, Project ID 390833453).

References

1. Arndt, D., Bangerth, W., Feder, M., Fehling, M., Gassmöller, R., Heister, T., Heltai, L., Kronbichler, M., Maier, M., Munch, P., Pelteret, J.-P., Sticko, S., Turcksin, B., and Wells, D. The `deal.II` library, version 9.4. *Journal of Numerical Mathematics* **30**(3), 231–246 (2022).
2. Beuchler, S., Kinnewig, S., and Wick, T. *Parallel domain decomposition solvers for the time harmonic Maxwell equations*, *Lecture Notes in Computational Science and Engineering*, vol. 145, 615–622. Springer (2023).
3. Bishop, C. M. *Pattern recognition and machine learning*. Springer (2006).
4. Demkowicz, L., Kurtz, J., Pardo, D., Paszynski, M., Rachowicz, W., and Zdunek, A. *Computing with HP-Adaptive Finite Elements, Vol. 2: Frontiers Three Dimensional Elliptic and Maxwell Problems with Applications*. Chapman & Hall/CRC, 1st ed. (2007).
5. Dolean, V., Gander, M. J., and Gerardo-Giorda, L. Optimized Schwarz methods for Maxwell's equations. *SIAM J. Sci. Comput.* **31**(3), 2193–2213 (2009).
6. El Bouajaji, M., Thierry, B., Antoine, X., and Geuzaine, C. A quasi-optimal domain decomposition algorithm for the time-harmonic maxwell's equations. *Journal of Computational Physics* **294**, 38–57 (2015).
7. Faustmann, M., Melenk, J. M., and Parvizi, M. $\mathcal{H}$-matrix approximability of inverses of FEM matrices for the time-harmonic Maxwell equations. *Advances in Computational Mathematics* **48**(5) (2022).
8. Henneking, S. and Demkowicz, L. A numerical study of the pollution error and dpg adaptivity for long waveguide simulations. *Computers & Mathematics with Applications* **95**, 85–100 (2021).
9. Higham, C. F. and Higham, D. J. Deep learning: An introduction for applied mathematicians. *SIAM review* **61**(4), 860–891 (2019).
10. Hiptmair, R. Multigrid method for maxwell's equations. *SIAM Journal on Numerical Analysis* **36**(1), 204–225 (1998).
11. Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization (2017). 1412.6980.
12. Kinnewig, S., Kolditz, L., Roth, J., and Wick, T. *Numerical Methods for Algorithmic Systems and Neural Networks*. Hannover : Institutionelles Repositorium der Leibniz Universität Hannover, Lecture Notes. Institut für Angewandte Mathematik, Leibniz Universität Hannover (2022, https://doi.org/10.15488/11897).
13. Knoke, T., Kinnewig, S., Beuchler, S., Demircan, A., Morgner, U., and Wick, T. Domain decomposition with neural network interface approximations for time-harmonic maxwell's equations with different wave numbers (2023). ArXiv:2303.02590.
14. Monk, P. *Finite Element Methods for Maxwell's Equations*. Oxford Science Publications (2003).
15. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems 32*, 8024–8035. Curran Associates, Inc. (2019). URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.
16. Schöberl, J. A posteriori error estimates for maxwell equations. *Mathematics of Computation* **77**(262), 633–649 (2008).
17. Toselli, A. and Widlund, O. *Domain decomposition methods - algorithms and theory*. Volume 34 of Springer Series in Computational Mathematics. Springer, Berlin, Heidelberg (2005).

Learning Adaptive FETI-DP Constraints for Irregular Domain Decompositions

Axel Klawonn, Martin Lanser, and Janine Weber

1 Introduction

Adaptive, that is, problem-dependent coarse spaces provide a robust condition number estimate and thus a robust convergence behavior for FETI-DP (Finite Element Tearing and Interconnecting - Dual Primal) and BDDC (Balancing Domain Decomposition by Constraints) methods for highly heterogeneous model problems; see, e.g., [7, 10] for a condition number indicator and a related proof for a specific adaptive coarse space in two spatial dimensions. In general, the setup of an adaptive coarse space usually requires the solution of local eigenvalue problems on edges, faces, or local parts of the domain decomposition interface. Even though the setup and the solution of these eigenvalue problems can be parallelized in a parallel implementation, it can take up the largest part of the overall time to solution, especially for three-dimensional problems. Thus, in [2], we have proposed to train a supervised classification model in form of a dense feedforward neural network to make an a priori decision, which of the eigenvalue problems are actually necessary for a robust FETI-DP coarse space. By testing our approach for different realistic heterogeneous model problems as, e.g., arising from a dual-phase steel in solid mechanics, we have shown that it is possible to drastically reduce the number of necessary eigenvalue problems while still maintaining the robustness of the iterative solver.

In [6], we have extended these results by directly learning the adaptive edge constraints themselves. Hence, we have trained different regression neural network models to compute an a priori approximation of the first $k \in \mathbb{N}$ adaptive edge constraints, which are then used to enhance the classic FETI-DP method. In particular, this approach does not require the setup or the solution of any eigenvalue problems at all. In [6], we have trained the regression neural network models exclusively with

Axel Klawonn, Martin Lanser, and Janine Weber
Department of Mathematics and Computer Science, University of Cologne, Weyertal 86-90, 50931 Köln, Germany, e-mail: axel.klawonn@uni-koeln.de, martin.lanser@uni-koeln.de, janine.weber@uni-koeln.de, url: http://www.numerik.uni-koeln.de and Center for Data and Simulation Science, University of Cologne, url: http://www.cds.uni-koeln.de

training data obtained from straight edges and consequently evaluated the trained network for test problems based on a regular domain decomposition only. Note that the approach of learning the adaptive constraints in an offline phase is in general of interest if a number of problems of the same class has to be solved, for example, diffusion or elasticity problems with different material coefficient distributions.

In this paper, we extend our idea from [6] by training regression neural networks which can be applied to both, regular domain decompositions as well as irregular decompositions as obtained by METIS [4]. To generalize our approach to arbitrary edge structures, we also train the network models with training data obtained from irregular edges and, additionally, with a set of randomized coefficient distributions. We provide numerical results for different heterogeneous stationary diffusion problems in two spatial dimensions for both, regular and irregular domain decompositions, and the adaptive coarse space from [10, 11].

2 Test problem and adaptive FETI-DP

As a test problem, we consider a stationary diffusion problem in two spatial dimensions

$$\begin{aligned} -\operatorname{div}(\rho \nabla u) &= 1 \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{1}$$

where $\rho\colon \Omega := [0,1]\times[0,1] \to \mathbb{R}$ denotes a heterogeneous coefficient function. Its weak formulation is discretized with piecewise linear conforming finite elements.

In this paper, we consider a hybrid, adaptive FETI-DP method which uses supervised machine learning to setup a robust and efficient coarse space. Thus, we decompose our domain $\Omega \subset \mathbb{R}^2$ into a number of nonoverlapping subdomains Ω_i, $i = 1, \dots, N$. Due to space limitations, we refrain from explaining the classic, that is, the non-adaptive FETI-DP method in detail. For a detailed description of the classic FETI-DP method, we refer to, e.g., [9]. Let us note that in our implementation, we always choose the vertices of the subdomains as primal variables. Additionally, we implement adaptive, that is, problem-dependent edge constraints to enhance the robustness of our methods; see the following discussion. For the remainder of the paper, we denote by $\mathcal{E}_{ij}$ the edge shared by the two neighboring subdomains Ω_i and Ω_j.

The classic FETI-DP condition number bound using exclusively primal vertex constraints is only robust under fairly restrictive assumptions on the coefficient function ρ; see, for example, [8]. Thus, we enhance the FETI-DP method with a very specific adaptive coarse space which was originally introduced in [10, 11].

Here, the main idea is to add selected eigenvectors to the coarse space, which are obtained from the solution of the following generalized local eigenvalue problem for each edge $\mathcal{E}_{ij}$: find $w_{ij} \in \left(\ker S_{ij}\right)^{\perp}$ such that

$$\langle P_{D_{ij}} v_{ij},\ S_{ij} P_{D_{ij}} w_{ij} \rangle = \mu_{ij} \langle v_{ij},\ S_{ij} w_{ij} \rangle \quad \forall\, v_{ij} \in \left(\ker S_{ij}\right)^{\perp}. \tag{2}$$

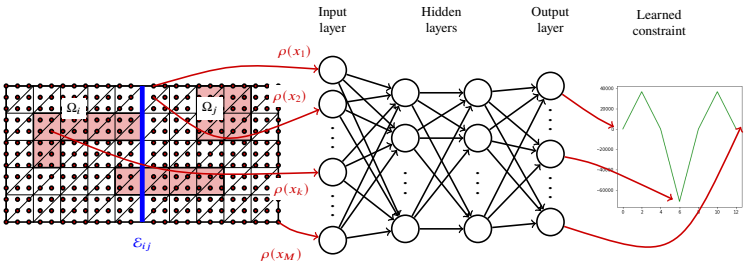


Fig. 1 Visualization of our network models N_l and $\widetilde{N}_l$, $l \leq 3$. As input data for the neural network, we use samples of the coefficent function for the two neighboring subdomains of an edge **(left)**. Here, dark red corresponds to a high coefficient and white corresponds to a low coefficient. The output of the network is a discretized egde constraint **(right)**. Figure taken from [6, Fig. 1].

Here, $S_{ij} = \text{diag}(S^{(i)}, S^{(j)})$ denotes a local Schur complement matrix with $S^{(i)}$ and $S^{(j)}$ being the Schur complements of $K^{(i)}$ and $K^{(j)}$, respectively, and $P_{D_{ij}} = B^T_{D,\mathcal{E}_{ij}} B_{\mathcal{E}_{ij}}$ is a local jump operator, with $B_{D,\mathcal{E}_{ij}} = \left(B^{(i)}_{D,\mathcal{E}_{ij}},\ B^{(j)}_{D,\mathcal{E}_{ij}}\right)$ being a local submatrix of $\left(B^{(i)}_D,\ B^{(j)}_D\right)$ obtained by exclusively taking the rows corresponding to the edge $\mathcal{E}_{ij}$; see [10] for more details. The matrix $B_{\mathcal{E}_{ij}}$ is obtained by taking the same rows from $\left(B^{(i)},\ B^{(j)}\right)$. We assume that R eigenvectors w^r_{ij}, $r = 1, ..., R$, belong to eigenvalues which are larger than a user-defined tolerance TOL and then enhance the FETI-DP coarse space with the edge constraint vectors

$$(c^r_{ij})^T := B_{D,\mathcal{E}_{ij}} S_{ij} P_{D_{ij}} w^r_{ij},\ r = 1, ..., R \tag{3}$$

using projector preconditioning; see [7, Sections 3,5] for more details. In the following we refer to the constraint vectors as constraints. When enhancing the FETI-DP coarse space with these adaptive constraints one can prove a robust condition number bound, which exclusively depends on the user-defined tolerance TOL and some geometrical constants; see, e.g., [7, Theorem 5.1]. On the one hand, this ensures a robust convergence behavior of the resulting FETI-DP algorithm, but, as a drawback, one has to setup and solve the eigenvalue problems in Eq. (2) for all edges belonging to the interface of our domain decomposition. Hence, in [6], we have proposed a hybrid FETI-DP method which uses a supervised regression model to directly learn approximations of the adaptive edge constraints resulting from Eq. (3) such that the solution of any eigenvalue problems is not necessary.

3 Learning coarse constraints in adaptive FETI-DP

The aim of our work is to compute discrete approximations of the first k adaptive edge constraints resulting from the local eigenvalue problem in Eq. (2) and to use the learned constraints to enhance the classic FETI-DP coarse space; see [6]. In partic-

ular, for each of the first k adaptive edge constraints, we train a separate regression neural network model that we denote by $N_l,\ l \leq k$. In the following, we always consider $k = 3$ and thus, train 3 different network models $N_l, l \leq 3$, to obtain 3 discrete approximations of the constraints resulting from the first 3 eigenmodes; see also [6]. As explained in more detail in [6], we additionally train separate neural network models for edges which have direct contact to the Dirichlet boundary $\partial\Omega_D$ of the domain and for edges without any contact to $\partial\Omega_D$ since both cases result in different edge constraints due to the influence of the Dirichlet boundary condition on the local Schur complement matrices S_{ij} in Eq. (2). To distinguish between these different network models, we denote the respective regression networks for edges with direct contact to $\partial\Omega_D$ by $\widetilde{N}_l$, $l \leq 3$; see also [6].

As input data for all neural network models $N_l,\ l \leq 3$, we use a mesh-independent image representation of the underlying coefficient function ρ within the two subdomains Ω_i and Ω_j adjacent to the edge $\mathcal{E}_{ij}$. The concrete details of the computation of this image representation are described in [2] such that, in the following, we only briefly sketch the main idea. First, we compute an auxiliary grid of points which we denote by *sampling grid* and which is independent of the finite element grid. Then, we evaluate the coefficient function ρ for each of these *sampling points* within the sampling grid and use the corresponding ρ values as input data for the neural networks. In order to make sure that the input data always have the same length and a consistent structure, we define a concrete order within our sampling grid and encode sampling points with the dummy value -1 if they fall outside the two neighboring subdomains for a given edge $\mathcal{E}_{ij}$. Let us note that this is especially relevant for irregular decompositions of the domain as obtained by METIS [4]. In particular, all trained network models N_l and $\widetilde{N}_l,\ l \leq 3$, share the same input data and only differ by their output data in order to define the concrete regression tasks. As specific output data for the different network models, we use discrete values of the adaptive edge constraints resulting from the local edge eigenvalue problems in Eq. (2). For the training of the l-th network N_l, we hence use a discretized version of the respective edge constraint resulting from the eigenvector w_{ij}^l belonging to the eigenvalue μ_{ij}^l. All in all, we use 3200 sampling points as input data for the neural networks and 19 output nodes, that is, 19 discrete values to approximate the adaptive edge constraints. In principle, the output space of our networks corresponds to an edge length defined by $H/h = 20$. However, in order to be able to evaluate the trained network for different finite element discretizations, we use an interpolation technique to generalize our approach to different mesh sizes and thus only use the number of 19 degrees of freedom for each edge as the basis for the interpolation. In case we want to apply the approximated constraints for finer or coarser finite element meshes, we linearly interpolate the obtained regression values by using the finite element mesh points as the interpolation points and the finite element basis functions as interpolation basis. An exemplary visualization of our network models N_l and $\widetilde{N}_l$ is given in Fig. 1.

Other than in [6], where we have trained and tested the different network models exclusively for regular edges, in this paper, we generalize these results also to irregular decompositions obtained by METIS [4]. Therefore, we train the different

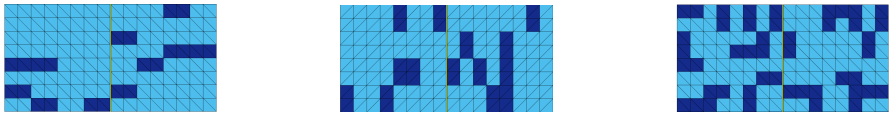


Fig. 2 Examples of three different randomly distributed coefficient functions obtained by using the same randomly generated coefficient for a horizontal (**left**) or vertical (**middle**) stripe of a maximum length of four finite element pixels, as well as by pairwise superimposing (**right**).

networks N_l and $\widetilde{N}_l$, $l \leq 3$, with both, regular and irregular edges. Additionally, in contrast to [6], we do not train the networks with our manually constructed set of coefficient distributions that we have denoted by *smart training data* in [6], but use a set of randomized coefficient distributions. In [3], we have shown that it is possible to achieve comparable accuracy results for the classification model as defined in [3] when using randomized training data with a slight structure compared to the smart training data. Considering these results and with regard to better expected generalization properties in three spatial dimensions, here, we have decided to also train our regression neural networks with randomized coefficient distributions. Three exemplary randomized coefficient distributions where we have additionally controlled the ratio of high versus low coefficient values are shown in Fig. 2. To obtain the entire set of training and validation data, we have generated various randomized coefficient distributions and combined them with pairs of subdomains adjacent to both, straight edges and edges resulting from the respective decompositions obtained by METIS. In particular, to generate the input and output data for the networks, we have used a regular decomposition of the unit square into 4×4 subdomains and a mesh size defined by $H/h \in \{10, 20, 40\}$ as well as the corresponding irregular decompositions obtained by METIS. All in all, this results in 4800 training and validation data configurations. In all coefficient configurations, we always set the high coefficient to $\rho_1 = 1e6$ and the low coefficient to $\rho_2 = 1$. For the selection of the necessary adaptive constraints, we always choose the tolerance $TOL = 100$.

Finally, for each of the network models N_l and $\widetilde{N}_l$, $l \leq 3$, we train a separate dense feedforward regression neural network [1] with 4 hidden layers and 50 neurons per layer. For each layer, we use the ReLU activation function and 20% dropout for each layer. For the optimization process, we have chosen the stochastic gradient descent (SGD) method using the Adam optimizer [5] with its default parameters, the initial learning rate of 0.001, and a batch size of 32. As loss function, we use the MSE (mean squared error) between the true adaptive and the predicted constraint vectors at the output grid points. For the final model, we obtain a MSE of $9.77e$-03 for the training data and a MSE of $4.62e$-02 for the validation data.

4 Numerical results

In this section, we provide numerical results for our proposed hybrid FETI-DP method using the approximated edge constraints as learned by the neural networks in direct comparison with the adaptive coarse space from [10].

To test our approach, we consider both, a regular decomposition and an irregular METIS decomposition of basically the same test problem. In both cases, the underlaying problem is a heterogeneous stationary diffusion problem, which we have already used in [6, Sect. 3]; see also Fig. 3 for a visualization. Only the underlying finite element discretization differs in both cases. Let us remark that this test configuration was of course not included in the training or validation data used for the training of the networks. For the test case with regular subdomains, we decompose our domain $\Omega = [0, 1]^2$ into 4×4 square subdomains and use a regular finite element mesh defined by $H/h = 10$. We choose all vertices as primal variables and consider a coefficient contrast of $\rho_1/\rho_2 = 1e6$. In particular, we compare the robustness of the resulting coarse space when implementing our trained edge constraints to the adaptive coarse space from [10] and the condition and iteration numbers from [6, Sect. 3] where we have trained the regression networks exclusively with training data from straight edges. Note again that in this paper, we train the networks with both, training data from straight edges and from irregular edges resulting from a decomposition by METIS. In Fig. 4 (top), we show the two adaptive edge constraints resulting from the local eigenvalue problem in Eq. (2) for the tolerance $TOL = 100$, that is, the ground truth as well as the learned approximations from our regression neural networks. As we can see from Fig. 4 (top), for an exemplary straight edge $\mathcal{E}_{ij}$ between two floating subdomains, both approximations using either just straight edges for the training or using both, straight and irregular edges, result in quantitatively similar approximations of the two adaptive edge constraints. Using the approximated edge constraints in our hybrid FETI-DP method leads to an iteration number of 14 and a condition number estimate of 35.5 when training the network with straight edges only while training the network with both straight and irregular edges results in an iteration number of 17 and a condition number estimate of 57.9; see also Table 1. In particular, both approximate coarse spaces result in robust condition number estimates independent of the coefficient contrast and using both, straight and irregular edges for the training of the network models provides qualitatively similar results as we have obtained in [6].

To test the performance of our approach with a METIS decomposition, we consider a decomposition of the unit square into 4×4 irregular subdomains computed by METIS [4] and we choose 3200 finite elements for each subdomain; see also Fig. 3 (right). Again, we consider a coefficient contrast of $\rho_1/\rho_2 = 1e6$. We evaluate our regression neural networks N_l and $\widetilde{N}_l$, $l \leq 3$, trained with straight and irregular edges for all 34 irregular edges resulting from the domain decomposition obtained by METIS in Fig. 3 (right), and integrate the learned edge constraints into the FETI-DP coarse space. The resulting iteration number and condition number estimate are given in Table 1, where we also show the corresponding values for the adaptive coarse space from [10] and the tolerance $TOL = 100$. As we can observe from Table 1,

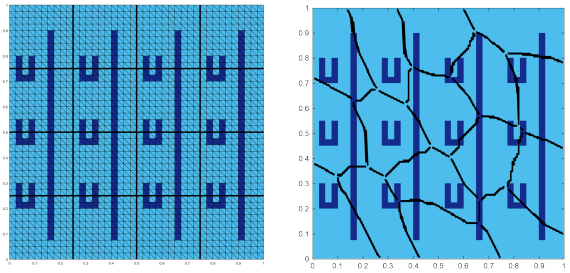


Fig. 3 Heterogeneous test problem: Stationary diffusion problem, coefficient contrast $1e6$, $\Omega = [0,1]^2$ decomposed into 4×4 subdomains. **Left:** Regular decomposition, mesh size defined by $H/h = 10$. **Right:** Irregular decomposition computed by METIS with 3200 FEs per subdomain.

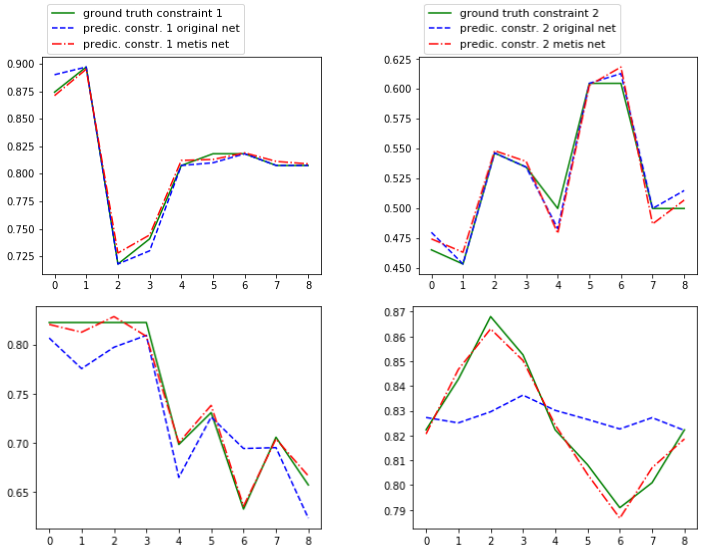


Fig. 4 Results for a straight edge of the regular decomposition (**top row**) and for an exemplary edge of the irregular decomposition (**bottom row**) of the test problem; see Fig. 3. Green, solid line: ground truth for the tolerance TOL = 100. Blue, dashed line: prediction as obtained by the neural networks in [6]. Red, dashed-dotted line: prediction as obtained by the neural networks trained with both, straight and irregular edges. See Table 1 for the resulting condition and iteration numbers.

using the learned constraints leads to a condition number estimate of 67.64 that is clearly independent of the coefficient contrast and in a quantitatively similar order of magnitude as the respective condition number for the adaptive FETI-DP coarse space. Thus, the learned coarse space seems to serve as a good approximation of the respective adaptive FETI-DP coarse space. Furthermore, in Fig. 4 (bottom), we show the learned constraints as well as the ground truth for an exemplary edge within the irregular decomposition. We can see that the learned constraints when training the networks with both, straight and irregular edges, are quantitatively similar to the ground truth. However, evaluating our networks from [6], which were only trained

Table 1 Condition number estimates (cond) and iteration numbers (iter) for the adaptive FETI-DP coarse space and the hybrid coarse spaces as learned by the regression neural networks for the coefficient distributions in Fig. 3. We denote by *METIS nets* the neural networks that are trained with both, straight and irregular edges.

	Regular decomposition		**METIS decomposition**	
	iter	**cond**	**iter**	**cond**
Classic FETI-DP	55	32443	79	375020
Adaptive FETI-DP	10	2.81	19	3.32
Learned constraints from [6]	14	35.56	41	7055.95
Learned constraints from METIS nets	17	57.97	26	67.64

with straight edges, for an irregular edge, provides a relatively poor approximation of the constraints. Note again that the setup of the learned coarse space does not require the solution of any eigenvalue problems at all and the training of the different network models can be executed in parallel and in an apriori offline phase. In particular, in this work, we have shown that it is possible to generalize our results from [6] also to non-straight edges as, e.g., resulting from METIS [4].

References

1. Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning*. MIT press Cambridge (2016).
2. Heinlein, A., Klawonn, A., Lanser, M., and Weber, J. Machine Learning in Adaptive Domain Decomposition Methods - Predicting the Geometric Location of Constraints. *SIAM J. Sci. Comput.* **41**(6), A3887–A3912 (2019).
3. Heinlein, A., Klawonn, A., Lanser, M., and Weber, J. Machine Learning in Adaptive FETI-DP - A Comparison of Smart and Random Training Data. In: *Domain decomposition methods in science and engineering XXV*, *Lect. Notes Comput. Sci. Eng.*, vol. 138, 218–226. Springer (2020).
4. Karypis, G. and Kumar, V. METIS: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0. http://www.cs.umn.edu/ metis (2009).
5. Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2015). Http://arxiv.org/abs/1412.6980.
6. Klawonn, A., Lanser, M., and Weber, J. Learning Adaptive Coarse Basis Functions of FETI-DP (2022). TR series, Center for Data and Simulation Science, University of Cologne, Germany, Vol. 2022-2. http://kups.ub.uni-koeln.de/id/eprint/62001. Submitted for publication in 06/2022.
7. Klawonn, A., Radtke, P., and Rheinbach, O. A comparison of adaptive coarse spaces for iterative substructuring in two dimensions. *Electron. Trans. Numer. Anal.* **45**, 75–106 (2016).
8. Klawonn, A., Rheinbach, O., and Widlund, O. B. An analysis of a FETI-DP algorithm on irregular subdomains in the plane. *SIAM J. Numer. Anal.* **46**(5), 2484–2504 (2008).
9. Klawonn, A. and Widlund, O. B. Dual-primal FETI methods for linear elasticity. *Comm. Pure Appl. Math.* **59**(11), 1523–1572 (2006).
10. Mandel, J. and Sousedík, B. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Comput. Methods Appl. Mech. Engrg.* **196**(8), 1389–1399 (2007).
11. Mandel, J., Sousedík, B., and Šístek, J. Adaptive BDDC in three dimensions. *Math. Comput. Simulation* **82**(10), 1812–1831 (2012).

Adaptive Three-Level BDDC Using Frugal Constraints

Axel Klawonn, Martin Lanser, and Janine Weber

1 Introduction

The convergence rate of both the FETI-DP (Finite Element Tearing and Interconnecting - Dual Primal) and the BDDC (Balancing Domain Decomposition by Constraints) domain decomposition methods strongly depend on the spectrum, i.e., the eigenvalues of the preconditioned system [7, 2]. To obtain a robust condition number estimate which is independent of the coefficient or material distribution, several adaptive coarse spaces have been developed which rely on the solution of local eigenvalue problems and use selected eigenvectors to enhance the coarse space; see, e.g., [9]. Besides the robustness of the considered domain decomposition method, its computational efficiency and parallel scalability is also of major interest. However, for an increasing number of subdomains, the exact solution of the coarse problem can become a scaling bottleneck within a parallel implementation. For the BDDC method, the coarse problem has the same structure as the original problem. Thus, it is straightforward to apply the BDDC preconditioner recursively either once or several times to the coarse problem, leading to a three-level [14] or a multilevel BDDC method [1, 10].

In the present work, we combine the three-level BDDC approach from [14] with the choice of adaptive constraints from [9] and, other than in the adaptive multilevel BDDC method in [12, 13], additionally with the frugal constraints from [3]. Since the computation of the frugal edge constraints is fairly cheap, we aim to reduce the computational effort of the adaptive three-level BDDC method by replacing the eigenvalue problems either on the subdomain or on the subregion level by frugal constraints while still retaining a satisfactory convergence behavior. We compare the robustness of the resulting different three-level BDDC methods using frugal

Axel Klawonn, Martin Lanser, and Janine Weber
Department of Mathematics and Computer Science, University of Cologne, Weyertal 86-90, 50931 Köln, Germany, e-mail: axel.klawonn@uni-koeln.de, martin.lanser@uni-koeln.de, janine.weber@uni-koeln.de, url: http://www.numerik.uni-koeln.de and
Center for Data and Simulation Science, University of Cologne, url: http://www.cds.uni-koeln.de

and/or adaptive edge constraints on either the second and/or the third level for different heterogeneous stationary diffusion problems with high contrasts in two spatial dimensions.

2 Problem and three-level BDDC

We consider a stationary diffusion problem in two spatial dimensions, i.e., the weak formulation of

$$\begin{aligned} -\operatorname{div}(\rho\nabla u) &= 1 \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{1}$$

Here, $\rho\colon \Omega := [0,1]\times[0,1] \to \mathbb{R}$ denotes the coefficient function and in Section 4, we will consider various heterogeneous coefficient functions ρ.

In this paper, we numerically investigate different coarse spaces and approximate solvers for the BDDC domain decomposition method. Thus, we decompose the domain Ω into $N \in \mathbb{N}$ nonoverlapping subdomains Ω_i, $i = 1,\dots,N$. For each of these subdomains, we then compute a conforming finite element triangulation and compute local stiffness matrices $K^{(i)}$ and local load vectors $f^{(i)}$, $i = 1,\dots,N$. Due to space limitations, we refrain from explaining the classic two-level BDDC method in detail and focus instead on the description of the different approximate coarse spaces and a specific adaptive BDDC coarse space. For a detailed description of the two-level BDDC method, we refer to [2].

In a parallel implementation of the two-level BDDC method, the exact solution of the coarse problem in form of the primally coupled Schur complement matrix $\widetilde{S}_{\Pi\Pi}^{-1}$ can become a scaling bottleneck; see, e.g., [4, 5] for related parallel numerical experiments in a linear and a nonlinear framework, respectively. One possible approach to delay the related scaling bottleneck is to apply the BDDC preconditioner recursively and to compute only an approximation $\widehat{S}_{\Pi\Pi}^{-1}$ of the coarse problem $\widetilde{S}_{\Pi\Pi}^{-1}$, leading to a three-level BDDC method [14]. Here, the main idea is to introduce a third level of the domain decomposition by additionally decomposing the domain Ω into a number of nonoverlapping subregions Ω^j, $j = 1,\dots,\overline{N}$. In particular, each of the subregions Ω^j comprises a given number of subdomains Ω_i, $i = 1,\dots,N_j$. Then, all primal variables on the second level are again partitioned into inner, primal, and dual variables on the subregion level. We denote the respective index sets on the subregion level by $\overline{I}$, $\overline{\Pi}$, and $\overline{\Delta}$, respectively; see also Fig. 1 for an exemplary visualization. On the subregion level, analogously to the subdomain level, all inner and dual variables are eliminated, leading to a primal Schur complement system on the third level which is, generally, of much smaller size than the respective system on the second level depending on the number of subdomains per subregion. Hence, only a smaller coarse problem on the third level has to be solved compared to the classic, i.e., the two-level BDDC method. A complete mathematical description as well as the related theory and a condition number bound for the three-level BDDC method for stationary diffusion problems can be found in [14]. Parallel numerical results and a weak scaling study for the three-level BDDC method in comparison with other

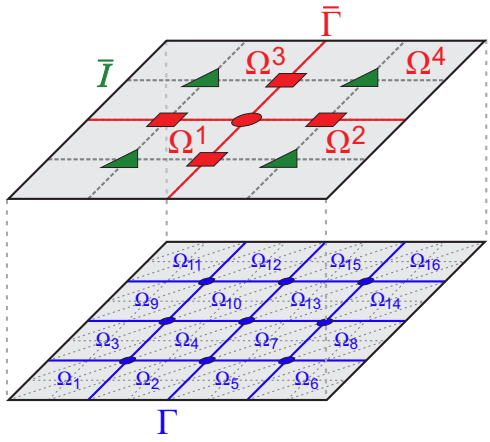


Fig. 1 Example of a three-level domain decomposition into 16 regular subdomains (bottom) and 4 regular subregions (top). We mark in blue the interface Γ between subdomains and in red the interface $\overline{\Gamma}$ between subregions. Primal nodes $\overline{\Pi}$ on the third level and primal nodes Π on the second level are visualized as red and blue circles, respectively. Inner or dual nodes on the third level, i.e., $\overline{I}$ or $\overline{\Delta}$, are visualized as green triangles or red squares, respectively. Figure taken from [15, Fig. 5.1].

approximate coarse solvers can, e.g., be found in [5]. Let us note that besides adding a third level of the domain decomposition, also further additional levels can be added leading to a recursive multilevel BDDC method; see, e.g., [1, 10].

3 Adaptive three-level BDDC combined with frugal edge constraints

In general, we are interested in BDDC coarse spaces which can efficiently be computed on a parallel computer and which are, preferably, robust for different heterogeneous problems. Unfortunately, the classic condition number bounds both for the FETI-DP and the BDDC method are only independent of the coefficient contrast under fairly restrictive assumptions on the coefficient distribution; see, e.g., [7, 8, 11] for a closer discussion for FETI-DP and BDDC, respectively. A similar theoretical condition number bound has also been derived for the three-level BDDC method; see [14]. As a remedy, different adaptive, i.e., problem-dependent coarse spaces have been developed. In the following, we will focus on a specific adaptive coarse space strategy which has been originally introduced in [9]. Here, the main idea is to solve a local generalized eigenvalue problem for each edge $\mathcal{E}_{ij}$ between two subdomains Ω_i and Ω_j which is of the general form: find $w_{ij} \in \left(\ker\, S_{ij}\right)^{\perp}$ such that

$$\langle P_{D_{ij}} v_{ij},\ S_{ij} P_{D_{ij}} w_{ij} \rangle = \mu_{ij} \langle v_{ij},\ S_{ij} w_{ij} \rangle \quad \forall v_{ij} \in \left(\ker\, S_{ij}\right)^{\perp}. \tag{2}$$

Here, $S_{ij} = \begin{pmatrix} S^{(i)} & \\ & S^{(j)} \end{pmatrix}$ is a local Schur complement matrix where $S^{(i)}$ and $S^{(j)}$ are the Schur complements of $K^{(i)}$ and $K^{(j)}$, respectively, and $P_{D_{ij}} = B^T_{D,\mathcal{E}_{ij}} B_{\mathcal{E}_{ij}}$ is a local jump operator; see [9] for more details. Assuming that R eigenvectors w^r_{ij}, $r = 1, \ldots, R$ belong to eigenvalues which are larger than a user-defined tolerance TOL, we then enhance the BDDC coarse space with the edge constraints

$$B_{D,\mathcal{E}_{ij}} S_{ij} P_{D_{ij}} w^r_{ij}, \quad r = 1, \ldots, R, \tag{3}$$

with $B_{D,\mathcal{E}_{ij}} = \left(B^{(i)}_{D,\mathcal{E}_{ij}},\ B^{(j)}_{D,\mathcal{E}_{ij}} \right)$ being a local submatrix of $\left(B^{(i)}_D,\ B^{(j)}_D \right)$ obtained by taking the rows corresponding to the edge $\mathcal{E}_{ij}$. In particular, for two-dimensional problems and primal subdomain vertices, enhancing the BDDC coarse space with these adaptive constraints leads to a robust condition number which exclusively depends on the chosen tolerance TOL and some geometrical constants; see [6].

In order to benefit both from the robustness of the described adaptive coarse space as well as from the increased parallel scalability of a three-level BDDC method, we combine both approaches and also implement adaptive edge constraints on the subregion level within a three-level BDDC method. To compute the local eigenvalue problem for edges between two neighboring subregions Ω^i and Ω^j, both the local Schur complement matrices $S^{(i)}$ and $S^{(j)}$ as well as the local jump operator $P_{D_{ij}}$ in Eq. (2) are replaced by recursive versions with respect to the primal variables on the subregion level. This leads to an adaptive three-level BDDC approach; see also [12, 13] for previous work on adaptive multilevel BDDC. Due to the implementation of adaptive constraints within each level, here, a robust condition number estimate can be obtained.

As a drawback, we have to set up and solve local eigenvalue problems on both the subdomain and the subregion level, which can be computationally expensive, especially for three-dimensional problems. Hence, we propose a modified approach of the adaptive multilevel approach presented in [12, 13] by replacing the eigenvalue problems either on the second and/or the third level by frugal edge constraints as introduced in [3]. The resulting frugal edge constraints serve as a low-dimensional approximation of the adaptive coarse space defined in [9] and have been shown to be robust for a range of different heterogeneous coefficient or material distributions both in two and three spatial dimensions; see [3] for detailed experiments. For two-dimensional stationary diffusion problems, the frugal edge constraints on the subdomain level are defined as follows. We denote by $\omega(x)$ the support of the finite element basis functions associated with a finite element node $x \in (\Omega_i \cup \Omega_j)$. Then, for each x on $\partial\Omega_i$ or $\partial\Omega_j$, respectively, we compute $\widehat{\rho}^{(i)}(x) = \max\limits_{y \in \omega(x) \cap \Omega_i} \rho(y)$ and $\widehat{\rho}^{(j)}(x) = \max\limits_{y \in \omega(x) \cap \Omega_j} \rho(y)$. We define $v^{(l)}_{E_{ij}}$ on $\partial\Omega_l$ for $l = i, j$ by

$$v^{(l)}_{E_{ij}}(x) := \begin{cases} \widehat{\rho}^{(l)}(x), & x \in \partial\Omega_l \backslash \Pi^{(l)}, \\ 0, & x \in \Pi^{(l)} \end{cases} \tag{4}$$

and $v_{E_{ij}}^T := (v_{E_{ij}}^{(i)T}, -v_{E_{ij}}^{(j)T})$; see also Fig. 2 for an exemplary illustration. Here, $\Pi^{(l)}$ denotes the index set of all local primal variables. Finally, we obtain the frugal edge constraint by $c_{E_{ij}} := B_{D,\mathcal{E}_{ij}} S_{ij} P_{D_{ij}} v_{E_{ij}}$ in direct analogy to the adaptive edge constraints in Eq. (3). Let us note that on the subregion level, the subdomains take over the role of the finite elements on the subdomain level and thus, the ρ-coefficient is not uniquely defined for each subdomain, i.e., each element on the third level. Hence, for the additional construction of frugal edge constraints on the subregion level, we use the stiffness, i.e., the diagonal entries of the local subregion Schur complement matrices instead of using the maximum ρ-coefficient to construct $v_{E_{ij}}^{(l)}(x)$ in Eq. (4).

In Table 1, we summarize the different coarse spaces presented here as well as their main benefits and drawbacks. The overall goal of this paper is to combine the different BDDC methods discussed above to benefit from the robustness of the presented adaptive constraints, a reduced computational effort when using frugal constraints, and the increased parallel scalability of a three-level BDDC method.

In the following, we consider four different possibilities of how to combine the presented BDDC methods from Table 1. In particular, we consider three-level BDDC with:

i) Frugal constraints on the second and the third level.
ii) Frugal constraints on the second and adaptive constraints on the third level.
iii) Adaptive constraints on the second and the third level.
iv) Adaptive constraints on the second and frugal constraints on the third level.

Furthermore, variants i) and ii) can be slightly modified by using the stiffness instead of the maximum ρ-coefficient values for the construction of the frugal subdomain edge constraints. We denote these variants by i a/b) and ii a/b), respectively, in the experiments in Section 4.

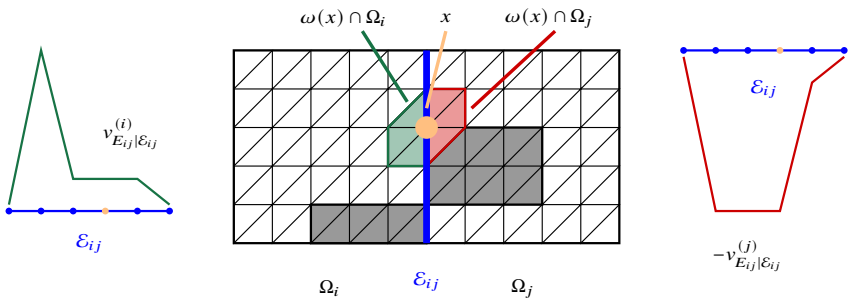


Fig. 2 Visualization of the construction of a frugal edge constraint in two dimensions for a given heterogeneous coefficient distribution. **Left/Right:** Maximum coefficient per finite element node of $\mathcal{E}_{ij}$ with respect to Ω_i and Ω_j, respectively, for the coefficient distribution in the middle. **Middle:** Exemplary heterogeneous coefficient distribution for two neighboring subdomains Ω_i and Ω_j sharing the edge $\mathcal{E}_{ij}$. High coefficients are marked in grey and low coefficients are marked in white. Figure taken from [15, Fig. 6.1].

Table 1 Non-exhaustive overview of benefits and drawbacks of the different BDDC algorithms considered in this paper.

Coarse space	Benefits	Drawbacks
Adaptive	Theoretical proof of robustness	Expensive setup
Frugal	Cheap setup	Limited robustness
3-Level, Classic	Increased parallel scalability	Robust only for moderate heterogeneities

4 Numerical results

In this section, we compare different BDDC methods using varying coarse spaces for different heterogeneous stationary diffusion problems in two dimensions. All shown computations were performed using MATLAB and a transformation-of-basis approach to implement the different coarse space enhancements. For all presented results, we choose all vertices as primal variables and we always use ρ-scaling unless explicitly mentioned otherwise. In Fig. 3, we show three different heterogeneous coefficient distributions which we use to evaluate the robustness of the four presented BDDC methods. Here, we always consider $\rho = 1e6$ in the dark blue pixels and $\rho = 1$ otherwise. Note that for the coefficient distribution in Fig. 3 (right), the ratio H/h has to be a multiple of five. Hence, we choose $H/h = 20$ for this case.

In Table 2 (top), we compare the iteration numbers and condition number estimates for the coefficient distribution in Fig. 3 (right) with coefficient jumps along and across both, subdomain and subregion edges. As we can observe from the results in Table 2, implementing adaptive constraints on both levels, i.e., algorithm iii) leads to the lowest iteration counts and lowest condition number estimates. However, all other BDDC variants using frugal edge constraints on the second and/or the third level also show condition numbers which are independent of the coefficient contrast and thus are robust. In particular, using frugal constraints on the subdomain level and computing adaptive constraints exclusively on the subregion level, i.e., algorithm ii), delivers results which are fairly similar to the fully adaptive approach. Hence, in this

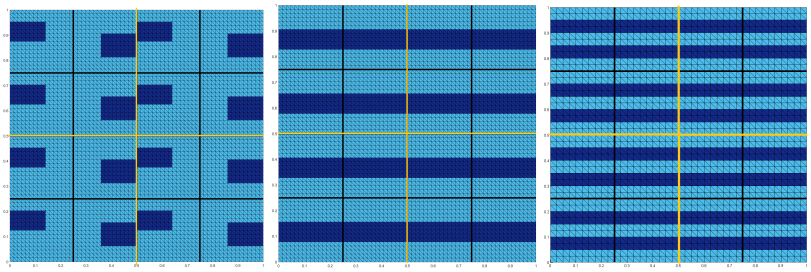


Fig. 3 Examples of three different heterogeneous coefficient functions. We set $\rho = 1e6$ in the dark blue pixels and $\rho = 1$ elsewhere. **Left:** Shifted boxes of a high coefficient with jumps along and across vertical edges; see Table 2 (top). **Middle:** One straight channel of a high coefficient crossing each vertical edge; see Table 2 (middle). **Right:** Two straight channels of a high coefficient crossing each vertical edge; see Table 2 (bottom).

Table 2 Iteration numbers (it) and condition numbers (cond) for a stationary diffusion problem with heterogeneous coefficient distributions as in Fig. 3. Decomposition of the domain into 4×4 subdomains and 2×2 subregions.

	2^{nd} **level**	3^{rd} **level**	**it**	**cond**
Shifted boxes; see Fig. 3 (left). $H/h = 16$**.**				
i a)	frugal, stiffness	frugal, stiffness	24	108.59
i b)	**frugal, rho-max**	**frugal, stiffness**	**18**	**35.58**
ii a)	frugal, stiffness	adaptive	19	24.78
ii b)	**frugal, rho-max**	**adaptive**	**18**	**30.11**
iii)	**adaptive**	**adaptive**	**18**	**21.73**
iv)	adaptive	frugal, stiffness	25	65.73
One straight channel; see Fig. 3 (middle). $H/h = 16$**.**				
i a)	frugal, stiffness	frugal, stiffness	17	11.11
i b)	**frugal, rho-max**	**frugal, stiffness**	**19**	**36.87**
ii a)	frugal, stiffness	adaptive	15	11.11
ii b)	**frugal, rho-max**	**adaptive**	**14**	**33.97**
iii)	**adaptive**	**adaptive**	**16**	**34.53**
iv)	adaptive	frugal, stiffness	20	35.93
Two straight channels; see Fig. 3 (right). $H/h = 20$**.**				
i a)	frugal, stiffness	frugal, stiffness	41	19 376
i b)	**frugal, rho-max**	**frugal, stiffness**	**42**	**54 582**
ii a)	frugal, stiffness	adaptive	34	19 355
ii b)	**frugal, rho-max**	**adaptive**	**28**	**55 009**
iii)	**adaptive**	**adaptive**	**22**	**42.92**
iv)	adaptive	frugal, stiffness	26	141.02

case, variant ii) would be our favored approach since it requires only the solution of smaller eigenvalue problems on the subregion level whereas the construction of frugal constraints on the subdomain level is computationally cheap.

For the coefficient distribution in Fig. 3 (middle) which is symmetric with respect to all edges and which has only a single channel crossing each subdomain edge, the numerical results for frugal and adaptive edge constraints are even more similar; see Table 2 (middle). This can be interpreted as an indicator that for this specific case, the computed frugal constraint is indeed a good approximation of the corresponding adaptive constraint. This will be further investigated in future research.

For the coefficient distribution in Fig. 3 (right) where we have more coefficient jumps along and across the subdomain and subregion edges, only adaptive three-level BDDC, i.e., algorithm iii) is robust with respect to the coefficient contrast; see Table 2 (bottom). However, also the remaining variants which use three-level BDDC with frugal constraints show satisfactory iteration numbers, indicating that we obtain only a few outliers within the spectrum of the preconditioned system.

As a conclusion, for completely arbitrary coefficient distributions with numerous jumps along and across the subdomain and subregion edges, only adaptive three-level BDDC ensures a robust condition number independent of the coefficient contrast. However, for rather moderate heterogeneities, also replacing the eigenvalue problems on either the second or the third level by frugal edge constraints can deliver a robust algorithm. With respect to computational efficiency, variant ii), i.e., frugal constraints

on the subdomain level and adaptive constraints on the subregion level would be our favored choice due to the smaller size of the eigenvalue problems exclusively on the subregion level. For future research, we plan to fully integrate all proposed approaches into our parallel BDDC software and to test it more extensively with respect to parallel scalability for both, two- and three-dimensional problems.

References

1. Badia, S., Martín, A. F., and Principe, J. Multilevel Balancing Domain Decomposition at Extreme Scales. *SIAM J. Sci. Comput.* **38**(1), C22–C52 (2016).
2. Dohrmann, C. R. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.* **25**(1), 246–258 (2003).
3. Heinlein, A., Klawonn, A., Lanser, M., and Weber, J. A Frugal FETI-DP and BDDC Coarse Space for Heterogeneous Problems. *Electr. Trans. Numer. Anal.* **53**, 562–591 (2020).
4. Klawonn, A., Lanser, M., and Rheinbach, O. Nonlinear BDDC Methods with Approximate Solvers. *Electron. Trans. Numer. Anal.* **49**, pp. 244–273 (2018).
5. Klawonn, A., Lanser, M., Rheinbach, O., and Weber, J. Preconditioning the coarse problem of BDDC methods - Three-level, algebraic multigrid, and vertex-based preconditioners. *Electron. Trans. Numer. Anal.* **51**, 432–450 (2019).
6. Klawonn, A., Radtke, P., and Rheinbach, O. A comparison of adaptive coarse spaces for iterative substructuring in two dimensions. *Electron. Trans. Numer. Anal.* **45**, 75–106 (2016).
7. Klawonn, A., Rheinbach, O., and Widlund, O. B. An analysis of a FETI-DP algorithm on irregular subdomains in the plane. *SIAM J. Numer. Anal.* **46**(5), 2484–2504 (2008).
8. Mandel, J., Dohrmann, C. R., and Tezaur, R. An algebraic theory for primal and dual substructuring methods by constraints. *Appl. Numer. Math.* **54**(2), 167–193 (2005).
9. Mandel, J. and Sousedík, B. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Comput. Methods Appl. Mech. Engrg.* **196**(8), 1389–1399 (2007).
10. Mandel, J., Sousedík, B., and Dohrmann, C. R. Multispace and multilevel BDDC. *Computing* **83**(2-3), 55–85 (2008).
11. Mandel, J. and Tezaur, R. On the convergence of a dual-primal substructuring method. *Numer. Math.* **88**(3), 543–558 (2001).
12. Sousedík, B. and Mandel, J. On adaptive-multilevel BDDC. In: *Domain decomposition methods in science and engineering XIX*, *Lect. Notes Comput. Sci. Eng.*, vol. 78, 39–50. Springer (2011).
13. Sousedík, B., Šístek, J., and Mandel, J. Adaptive-multilevel BDDC and its parallel implementation. *Computing* **95**(12), 1087–1119 (2013).
14. Tu, X. Three-level BDDC in two dimensions. *Internat. J. Numer. Methods Engrg.* **69**(1), 33–59 (2007).
15. Weber, J. *Efficient and robust FETI-DP and BDDC methods–Approximate coarse spaces and deep learning-based adaptive coarse spaces*. Ph.D. thesis, Universität zu Köln (2022). Online available at http://kups.ub.uni-koeln.de/id/eprint/55179.

Efficient Adaptive Elimination Strategies in Nonlinear FETI-DP Methods in Combination with Adaptive Spectral Coarse Spaces

Axel Klawonn and Martin Lanser

1 Introduction

Nonlinear domain decomposition methods (DDMs) are based on a decomposition of a discretized nonlinear partial differential equation instead of applying a linear DDM to the tangential systems in a Newton-type iteration. The advantages are a faster convergence and an improved ratio of local work to communication, at least for many problems. We focus on nonlinear FETI-DP (Finite Element Tearing and Interconnecting - Dual Primal) methods here, which build a class of nonlinear two-level approaches. These methods can have a (partially) nonlinear coarse level and the integrated nonlinear right-preconditioner is based on a partial elimination of arbitrary degrees of freedom collected in an index set E. In [2], it was shown that the combination of nonlinear FETI-DP with an adaptive coarse space [9], which was implemented with a transformation of basis approach, improves the convergence. Additionally, in [6] the concept of choosing an index set E adaptively based on the residual was investigated. Finally, in [3], both ideas are combined to a nonlinear FETI-DP algorithm iterating in the transformed space, which we abbreviate with NL-FETI-DP-XT here. Additionally, also in [3], an efficient implementation iterating in the original finite element space is suggested using local saddle point problems [8] instead of an explicit transformation of basis; this method is abbreviated by NL-FETI-DP-X here. For the latter approach, modifications have to be made to the elimination set E. We will compare different strategies to modify E and finally suggest and numerically test a completely new and more efficient and robust strategy for NL-FETI-DP-X based on an approximation of the transformed residual.

Axel Klawonn, Martin Lanser
Department of Mathematics and Computer Science, Division of Mathematics, University of Cologne, Weyertal 86-90, 50931 Cologne, Germany, e-mail: axel.klawonn@uni-koeln.de, martin.lanser@uni-koeln.de

2 Nonlinear FETI-DP

Let us briefly recall the unified framework of nonlinear FETI-DP methods. For a detailed description, we refer to [5]. Throughout this article, we assume that we have a computational domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, which is divided into N nonoverlapping subdomains Ω_i, i.e., $\Omega = \bigcup_{i=1}^{N} \Omega_i$. Each subdomain is the union of finite elements and the associated finite element spaces are denoted by $W^{(i)}$. We denote the product space of all finite element spaces as $W = W^{(1)} \times \cdots \times W^{(N)}$. In FETI-DP methods, we partition all variables into interior (I), dual (Δ), and primal (Π) variables, where only continuity in the primal variables is prescribed and continuity in the dual variables is enforced by Lagrange multipliers λ iteratively. Therefore, we further introduce a subspace $\widetilde{W} \subset W$ of all finite element functions from W that are continuous in the primal variables. A simple choice of primal variables are subdomain vertices. More advanced strategies are based on enforcing continuity in certain weighted averages over the degrees of freedoms of an edge or face. The weights can, for example, be computed adaptively by solving localized eigenvalue problems related to edges. This approach results in provably robust linear FETI-DP methods; see, e.g., [7, 9]. For nonlinear FETI-DP methods, the adaptive coarse space can be computed using the tangential matrix linearized in the initial value; see [2]. We use this specific adaptive coarse space in all computations in this article.

For completeness, we also introduce the subspace $\widehat{W} \subset W$, which contains all finite element functions that are continuous across the complete interface and it holds $\widehat{W} \subset \widetilde{W} \subset W$. Let us introduce the primal assembly operator $\check{R}^T : W \to \widetilde{W}$ and the nonlinear function $K(u) : W \to W$ obtained by a finite element discretization of a nonlinear partial differential equation. Let us note that $K(u)$ is not necessarily continuous on the interface.

As it was shown in [4], finding the solution of the fully assembled finite element problem is equivalent to solving the nonlinear FETI-DP saddle point system

$$A(\tilde{u}, \lambda) = \begin{bmatrix} \widetilde{K}(\tilde{u}) + \check{R}^T B^T \lambda - \check{R}^T f \\ B\check{R}\tilde{u} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tilde{u} \in \widetilde{W},\ \widetilde{K}(\tilde{u}) := \check{R}^T K(\check{R}\tilde{u}) \in \widetilde{W}. \quad (1)$$

This system is the basis for all nonlinear FETI-DP methods. Here, the linear constraints $B\check{R}\tilde{u} = 0$ together with Lagrange multipliers $\lambda \in V := \text{range}(B)$ enforce continuity in all dual variables.

To implement arbitrary coarse constraints, as, e.g., adaptive constraints, one can use a transformation of basis approach. The underlying idea is to transform the complete system into a space W_T, where all primal constraints are again pointwise constraints and can be enforced by a simple assembly operator as before. A transformation matrix $T : W_T \to W$ with orthonormal rows, that is, with $T^T T = I$, can be computed for all coarse spaces based on edge and face averages; see [3] for details. Then, the transformed nonlinear FETI-DP saddle point system writes

$$A_T(\tilde{u}_T, \lambda) = \begin{bmatrix} \widetilde{K}_T(\tilde{u}_T) + \check{R}^T T^T B^T \lambda - \check{R}^T T^T f \\ BT\check{R}\tilde{u}_T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \widetilde{K}_T(\tilde{u}_T) := \check{R}^T T^T K(T\check{R}\tilde{u}). \tag{2}$$

As introduced in [5], we use a nonlinear right-preconditioner $M_T(\tilde{u}_T, \lambda)$ that is nonlinear in $\tilde{u}_T$ and linear in λ; see [5] for some desirable properties of M_T. Instead of $A_T(\tilde{u}_T, \lambda) = 0$, we now solve $A_T(M_T(\tilde{u}_T, \lambda)) = 0$ with a Newton-Krylov method. Following [5], the application of a nonlinear right-preconditioner can be interpreted as a (partial) nonlinear elimination process, where different choices of M_T lead to different elimination sets. With this interpretation, it is obvious to divide the overall set of variables into two different subsets E and L, where E contains all variables that should be nonlinearly eliminated by the preconditioner M_T, and L contains the remaining variables in which will be linearized.

After an appropriate rearrangement, we can split Eq. (2) according to the subsets E and L. We can write the nonlinear saddle point system (Eq. (2)) as

$$A_T(\tilde{u}_{T,E}, \tilde{u}_{T,L}, \lambda) = \begin{bmatrix} A_{T,E}(\tilde{u}_{T,E}, \tilde{u}_{T,L}, \lambda) \\ A_{T,L}(\tilde{u}_{T,E}, \tilde{u}_{T,L}, \lambda) \\ BT\check{R}\tilde{u}_T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

With the application of the nonlinear right-preconditioner, we now aim to eliminate all variables $\tilde{u}_E$, which correspond to the subset E. Thus, our preconditioner is implicitly defined by solving the nonlinear equation

$$A_{T,E}(M_{T,\tilde{u}_{T,E}}(\tilde{u}_{T,L}, \lambda), \tilde{u}_{T,L}, \lambda) = 0, \tag{3}$$

where we have $M_T(\tilde{u}_{T,E}, \tilde{u}_{T,L}, \lambda) := (M_{T,\tilde{u}_{T,E}}(\tilde{u}_L, \lambda), \tilde{u}_L, \lambda)$, since, by construction, M_T is linear in $\tilde{u}_{T,L}$ and λ. After we have computed the nonlinear preconditioner M_T by solving Eq. (3) with Newton's method, we obtain the nonlinear Schur complement system

$$\begin{bmatrix} A_{T,L}(M_{T,\tilde{u}_{T,E}}(\tilde{u}_{T,L}, \lambda), \tilde{u}_{T,L}, \lambda) \\ BT\check{R}\tilde{u}_T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

This can be solved with the traditional Newton-Krylov-FETI-DP approach; see [5]. Putting it all together, in each of these (outer) Newton iterations, M_T has to be recomputed, which is typically done by an inner Newton iteration.

Both Newton loops iterate in the transformed space, that is, all outer Newton updates $\delta\tilde{u}_T$ and inner Newton updates $\delta\tilde{u}_{T,E}$ have to be projected back to the original space after convergence. As in linear FETI-DP methods, the explicit usage of T leads to denser linear systems and thus, especially in three dimensions using rich coarse spaces, to a higher memory demand and a slower time to solution. As in the linear case, it is possible to reformulate nonlinear FETI-DP in the original nodal space using local saddle point systems and some further tricks; see [3] for details. It is possible to get rid of the matrix T in all computations of nonlinear FETI-DP without changing the nonlinear and linear convergence. Unfortunately, the additional

assumption has to be made that T has the block structure

$$T = \begin{bmatrix} T_E & 0 \\ 0 & T_L \end{bmatrix};$$

see [3] for details. To enforce this, all primal edges or faces, i.e., edges or faces with at least one primal constraint, have to be either included in E or L completely. In contrast, iterating in the transformed space, E can be chosen arbitrarily. In this article, we discuss different strategies of how to choose an appropriate E adaptively and compare nonlinear FETI-DP in the iterating in the nodal space (NL-FETI-DP-X) with nonlinear FETI-DP iterating in the transformed space (NL-FETI-DP-XT).

3 Adaptive selection of E

NL-FETI-DP-XT allows for completely arbitrary elimination sets E. For the more efficient NL-FETI-DP-X we will fulfill the necessary assumption on T by either choosing an edge to be part of E or, respectively, L completely, i.e., we will not split any edge. More precisely, in theory, it is sufficient to not split faces or edges which carry primal constraints. The adaptive selection of E used in this article is a modification of the procedure suggested in [6]. This heuristic strategy is based on the assembled nonlinear residual and is inspired by [1].

We first define the nonlinear residual in the k-th outer Newton iteration

$$r^{(k)} := K(u^{(k)}) + B^T \lambda^{(k)} - f = K(T\check{R}\tilde{u}_T^{(k)}) + B^T \lambda^{(k)} - f$$

and the assembled and transformed residual by

$$r_T^{(k)} := R^T T^T r^{(k)},$$

where $R^T : W \to \widehat{W}$ assembles all degrees of freedom on the interface. We now eliminate all variables, where the residual is comparably high, that is, if for the i-th component $r_{T,i}^{(k)}$ of $r_T^{(k)}$ the inequality

$$r_{T,i}^{(k)} \geq \rho_E ||r_T^{(k)}||_\infty$$

holds, the i-th degree of freedom is eliminated. That means, the index i is added to the elimination set E. Let us remark that we only describe the scalar case here. A procedure for systems of equations with more degrees of freedom in each physical node, e.g., elasticity problems, is discussed in [6] and out of the scope of this article. Here, $\rho_E < 1$ is a user defined parameter and smaller values immediately result in larger elimination sets E. To avoid single and isolated physical points in the elimination set, δ_E layers of finite element nodes surrounding E are added to the preselected set E. This is comparable to the procedure of selecting an overlap of a nonoverlapping subdomain. The resulting E can immediately be used within NL-FETI-DP-XT

and we denote this procedure to find E by **basic strategy**. Nonetheless, the resulting E can not be used in NL-FETI-DP-X, where we are not allowed to split edges. We suggest two different strategies to overcome this issue.

Strategy 1: After choosing E with the basic algorithm, all edges which have a nonempty intersection with E are added to E completely. Around these edges, δ_E layers of finite element nodes belonging to the interior of the adjacent subdomains are also added to E. We refer to Fig. 1 for a visualization of this strategy.

A disadvantage of strategy 1 and the basic approach is the need for computing T which is used to compute $r_T^{(k)}$. We will introduce a third strategy avoiding T, since, in the efficient implementation NL-FETI-DP-X, T is not necessary at all. Only the rows of T^T belonging to primal variables are usually known. Sorting the variables properly, we have

$$T^T = \begin{bmatrix} T_{\Pi\Pi}^T & T_{\Pi B}^T \\ T_{B\Pi}^T & T_{BB}^T \end{bmatrix}$$

and only the block $C := \begin{bmatrix} T_{\Pi\Pi}^T & T_{\Pi B}^T \end{bmatrix}$ is available. Let us note that $T_{\widehat{B},:}^T = [I\ 0]$, where $\widehat{B} := [I\ \widehat{\Delta}]$ and $\widehat{\Delta}$ is the index set of all dual variables belonging to edges which do not carry primal constraints. Therefore, we have

$$r_{T,\widehat{B}}^{(k)} = \left(R^T r^{(k)}\right)\Big|_{\widehat{B}}$$

which can be computed without knowing T. For the computation of the primal part $r_{T,\Pi}^{(k)}$ solely C is necessary. Only the part related to the dual part of the primal edges $r_{T,\Delta\setminus\widehat{\Delta}}^{(k)}$ cannot be computed without using T^T. In NL-FETI-DP-X, all edges carrying primal constraints have to be either completely part of E or, respectively, L. Assuming to have an appropriate coarse space and that all important information about the primal edges are transformed to the coarse space and thus to the vector $r_{T,\Pi}^{(k)}$, we can rely on $r_{T,\Pi}^{(k)}$ for the decision if an edge, which is part of $\Delta \setminus \widehat{\Delta}$, is chosen to be part of E or L. We therefore suggest the following modification.

Strategy 2: Choose the initial set E applying the basic algorithm to $\bar{r}_T^{(k)} := [r_{T,\widehat{B}}^{(k)T},\ r_{T,\Pi}^{(k)T},\ 0_{\Delta\setminus\widehat{\Delta}}]$ instead of $r_T^{(k)}$. Then proceed as in Strategy 1.

4 Problem and numerical results

We consider the nonlinear problem

$$\begin{aligned} -\alpha\Delta_p u &= 1 && \text{in } \Omega, \\ u &= 0 && \text{on } \partial\Omega, \end{aligned} \tag{4}$$

with the scaled p-Laplace operator $\alpha\Delta_p u := \operatorname{div}(\alpha|\nabla u|^{p-2}\nabla u)$. Within this article, we use $p = 4$ and a coefficient function $\alpha\colon \Omega \to \mathbb{R}$ with jumps. Moreover, we always

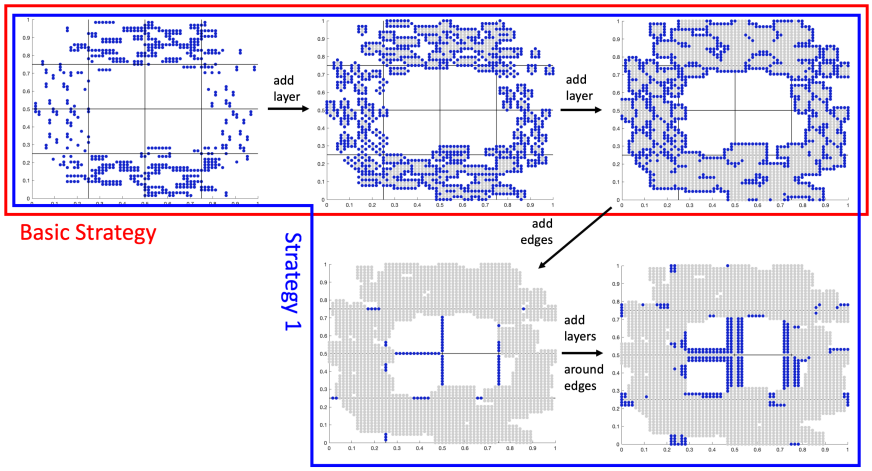


Fig. 1 Illustration of Strategy 1 to compute the elimination set E with $\delta_E = 2$. The starting point or initial set E is obtained based on the residual. In Strategy 1, first two layers are added, then all necessary edges are included in E, and finally an overlap of δ_E layers is added in the interior of the subdomains adjacent to those edges. The result of the basic algorithm can solely be used in NL-FETI-DP-XT.

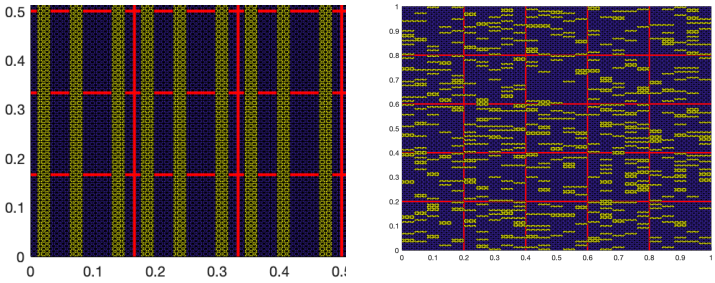


Fig. 2 Coefficient distributions and domain decompositions used in the numerical computations. **Left:** Channels with $\alpha = 1e3$ crossing material with $\alpha = 1$, zoomed in to a quarter of the unit square. **Right:** Randomly generated distribution with $\alpha = 1e6$ in the small yellow stripes.

use the unit square $\Omega = [0, 1] \times [0, 1]$ as the computational domain, a discretization with piecewise linear finite elements, and a structured domain decomposition into square subdomains. We consider two different coefficient distributions, which can be found in Fig. 2. We always choose $u^{(0)}(x, y) = x(1 - x)y(1 - y)$ as initial value in all computations.

For all edges we compute the adaptive coarse constraints introduced in [9] using the first linearized system and a tolerance of $tol = 10$ for the localized eigenvalue problems. For linear problems, there is a provable condition number bound of $N_{\mathcal{E}}^2 \cdot tol$ for FETI-DP using this coarse space, where $N_{\mathcal{E}}$ is the maximum number of edges

Table 1 Results for model problems with randomly generated coefficients and channels; always using vertices plus adaptive edge constraints; **outer it.** gives the total number of global Newton iterations and in brackets the number of Newton-Krylov-FETI-DP steps used for stability is shown; **inner it.** gives the number of inner Newton iterations summed up over the outer Newton iterations; **PCG it.** gives the number of PCG iterations summed up over the outer Newton iterations; $|E_{avg}|$ gives the average size of the elimination set in percentage of the number of degrees of freedom; **NL-FETI-DP-X** and **NL-FETI-DP-XT** stand for the adaptive selection of the elimination set; **NL-FETI-DP-2** stands for eliminating all variables, i.e., $E = [B\ \Pi]$; **NK-FETI-DP** stands for Newton-Krylov-FETI-DP. The best results are marked in bold.

p-Laplace random; see Fig. 2 (right) $p = 4$; $H/h = 25$; 25 subdomains; $tol = 10$							
method	**Strategy**	δ_E	ρ_E	$\|E\|_{avg}$	**outer it.**	**inner it.**	**PCG it. (sum)**
NL-FETI-DP-2	-	-	-	37.5%	8(5)	57	142
NK-FETI-DP	-	-	-	-	15(15)	-	284
NL-FETI-DP-XT	basic	2	0.01	6.9%	13(10)	36	254
NL-FETI-DP-XT	basic	5	0.01	33.3%	**4(0)**	**53**	81
NL-FETI-DP-X	1	2	0.01	11.1%	11(2)	74	200
NL-FETI-DP-X	1	5	0.01	41.1%	**4(0)**	55	78
NL-FETI-DP-X	2	2	0.01	8.3%	13(6)	68	237
NL-FETI-DP-X	2	5	0.01	38.7%	**4(0)**	**53**	**77**
p-Laplace channels; see Fig. 2 (left) $p = 4$; $H/h = 32$; 36 subdomains; $tol = 10$							
method	**Strategy**	δ_E	ρ_E	$\|E\|_{avg}$	**outer it.**	**inner it.**	**PCG it. (sum)**
NL-FETI-DP-2	-	-	-	71.4%	7(2)	43	80
NK-FETI-DP	-	-	-	-	19(19)	-	237
NL-FETI-DP-XT	basic	2	0.01	5.2%	16(12)	61	197
NL-FETI-DP-XT	basic	5	0.01	7.5%	14(9)	101	180
NL-FETI-DP-X	1	2	0.01	20.2%	6(0)	53	72
NL-FETI-DP-X	1	5	0.01	43.8%	**4(0)**	**45**	**54**
NL-FETI-DP-X	2	2	0.01	19.9%	6(0)	53	73
NL-FETI-DP-X	2	5	0.01	43.8%	**4(0)**	**45**	**54**

a subdomain can have; see [7] for the proof. In our computations, the outer Newton iteration is stopped if a relative reduction of 10^{-5} of the globally assembled residual is reached. The inner iteration is stopped, if the inner Newton update is smaller than 10^{-5} in the l_2-norm. Let us finally note that, for stability reasons, we will always switch to a Newton-Krylov-FETI-DP approach, if no further reduction of the residual is reached in the outer loop. We never switch back to nonlinear FETI-DP.

We always compute the average size of the elimination set E to give a rough estimate on the computational cost of the elimination process in the inner loop. We therefore compute the value

$$|E|_{avg} := \frac{1}{N_o} \sum_{k=1}^{N_o} \frac{|E^{(k)}|}{N_{dof}} \cdot 100\%,$$

where $|E^{(k)}|$ is the number of degrees of freedom in the elimination set of the k-th outer iteration, N_{dof} is the number of total degrees of freedom, and N_o the number of outer iterations. Let us remark that $|E^{(k)}| = 0$ for each Newton-Krylov iteration and thus $|E|_{avg}$ can be small if many Newton-Krylov steps have to be made.

The results for both model problems can be found in Table 1. It can be observed that NL-FETI-DP-X can compete with NL-FETI-DP-2 in terms of nonlinear and linear convergence; at least if appropriate elimination sets are chosen. Let us remark that NL-FETI-DP-X has in all setups less than 44% of the local computational cost. Additionally, both approaches outperform classical NK-FETI-DP. Strategies 1 and 2 have only been introduced in order to implement the theoretical need for edges not being split up in the efficient implementation of NL-FETI-DP-X. Nonetheless, splitting edges, which happens often in the basic strategy used in NL-FETI-DP-XT, actually deteriorates the performance, which was not expected. The most efficient Strategy 2, which does not need T explicitly, is competitive to Strategy 1 and therefore it is our suggestion to use this approach in NL-FETI-DP-X. Of course more tests and also three dimensional problems have to be investigated in the future.

References

1. Gong, S. and Cai, X.-C. A nonlinear elimination preconditioned inexact Newton method for heterogeneous hyperelasticity. *SIAM Journal on Scientific Computing* **41**(5), S390–S408 (2019).
2. Heinlein, A., Klawonn, A., and Lanser, M. Adaptive nonlinear domain decomposition methods with an application to the p-Laplacian. *SIAM Journal on Scientific Computing* **0**(0), S152–S172 (0).
3. Klawonn, A. and Lanser, M. Nonlinear FETI-DP domain decomposition methods: On the efficient implementation of arbitrary coarse spaces and nonlinear elimination sets (2022). In preparation.
4. Klawonn, A., Lanser, M., and Rheinbach, O. Nonlinear FETI-DP and BDDC methods. *SIAM J. Sci. Comput.* **36**(2), A737–A765 (2014).
5. Klawonn, A., Lanser, M., Rheinbach, O., and Uran, M. Nonlinear FETI-DP and BDDC methods: a unified framework and parallel results. *SIAM J. Sci. Comput.* **39**(6), C417–C451 (2017).
6. Klawonn, A., Lanser, M., and Uran, M. Adaptive nonlinear elimination in nonlinear FETI-DP methods. Preprint, Universität zu Köln (2021).
7. Klawonn, A., Radtke, P., and Rheinbach, O. A comparison of adaptive coarse spaces for iterative substructuring in two dimensions. *Electron. Trans. Numer. Anal.* **45**, 75–106 (2016).
8. Klawonn, A. and Widlund, O. B. Dual-Primal FETI methods for linear elasticity. *Communications on Pure and Applied Mathematics* **59**(11), 1523–1572 (2006).
9. Mandel, J. and Sousedík, B. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Comput. Methods Appl. Mech. Engrg.* **196**(8), 1389–1399 (2007).

On the Use of Hybrid Coarse-Level Models in Multilevel Minimization Methods

Alena Kopaničáková

1 Introduction

We consider the following minimization problem:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}), \tag{1}$$

where $f\colon \mathbb{R}^n \to \mathbb{R}$ is a bounded, twice continuously differentiable objective function and $n \in \mathbb{N}$ is typically very large. Our goal is to minimize (1) using a nonlinear multilevel minimization (NMM) method, e.g., MG-OPT [11] or RMTR [7]. The main idea behind NMM methods is to employ a hierarchy of so-called coarse-level objective functions, denoted by $\{f^\ell\}_{\ell=1}^L$, where $L > 1$. These functions are typically obtained by exploring the structure of the underlying minimization problem, e.g., by discretizing the underlying infinite-dimensional problem with a varying discretization parameter. During the solution process, the functions $\{f^\ell\}_{\ell=1}^L$ are utilized in order to construct the search-directions for the minimization problem at hand in a computationally efficient manner.

The overall efficiency of NMM methods relies on the ability of the coarse-level objective functions $\{f^\ell\}_{\ell=1}^L$ to approximate the function f well. Indeed, the convergence theory of the majority of NMM methods requires that the local behavior of the coarse-level objective functions is at least first-order coherent with the local behavior of f. The coherence is commonly ensured by employing the so-called τ-correction [1], which corrects the coarse-level objective function f^ℓ in an additive manner. Although this approach is almost universally employed in the multilevel literature, other approaches were also considered, e.g., a second-order additive correction approach [7, 12], or Galerkin-based coarse-level models [7, 9]. In this work, we explore techniques from the surrogate-based/multi-fidelity optimization [4] in order to construct the first-order coherent coarse-level models in the context of NMM methods. In particular, we discuss how to correct functions $\{f^\ell\}_{\ell=1}^L$ using additive, multiplicative, and hybrid approaches.

Alena Kopaničáková
Brown University, USA, e-mail: alena.kopanicakova@brown.edu

2 Nonlinear multilevel minimization framework

In this work, we minimize (1) using the NMM method. To this aim, we consider a hierarchy of L levels. Each level $\ell = 1, \dots, L$ is associated with some model $h^\ell : \mathbb{R}^{n^\ell} \to \mathbb{R}$, where we assume that $h^{\ell-1}$ is computationally cheaper to minimize than h^ℓ and that $n^{\ell-1} < n^\ell$. As we will discuss in Section 3, the models $\{h^\ell\}_{\ell=1}^L$ are constructed during the minimization process by correcting the objective functions $\{f^\ell\}_{\ell=1}^L$ by taking into account the knowledge of the current iterate. Through this work, we assume that $h^L := f^L := f$. Transfer of the data between different levels of the multilevel hierarchy is performed using the prolongation operator $\mathbf{I}_\ell^{\ell+1} : \mathbb{R}^{n^\ell} \to \mathbb{R}^{n^{\ell+1}}$, and the restriction operator $\mathbf{R}_{\ell+1}^\ell : \mathbb{R}^{n^{\ell+1}} \to \mathbb{R}^{n^\ell}$, where $\mathbf{R}_{\ell+1}^\ell = (\mathbf{I}_\ell^{\ell+1})^T$. Moreover, we also employ the projection operator $\mathbf{P}_{\ell+1}^\ell : \mathbb{R}^{n^{\ell+1}} \to \mathbb{R}^{n^\ell}$ to transfer iterates from the level $\ell+1$ to ℓ. The operator $\mathbf{P}_{\ell+1}^\ell$ is constructed such that $\mathbf{x}^\ell = \mathbf{P}_{\ell+1}^\ell(\mathbf{I}_\ell^{\ell+1}\mathbf{x}^\ell)$, for any $\mathbf{x}^\ell \in \mathbb{R}^{n^\ell}$.

Using the aforementioned definitions, we now describe a generic NMM method in the form of a V-cycle, summarized in Algorithm 1. During the description, we use a superscript to denote the level and a subscript to denote the iteration index. Starting from the finest level, $\ell = L$, and initial guess $\mathbf{x}_0^\ell$, the NMM method performs μ_s nonlinear smoothing steps to approximately minimize model h^ℓ. The choice of the nonlinear smoother depends on the particular choice of the NMM method. For instance, one can employ a first-order method equipped with a line-search or trust-region globalization strategy if a variant of multilevel line-search or trust-region method is considered. The outcome of this minimization process, iterate $\mathbf{x}_{\mu_s}^\ell$, is then used to construct a coarse-level model $h^{\ell-1}$ and initial guess $\mathbf{x}_0^{\ell-1} = \mathbf{P}_\ell^{\ell-1}\mathbf{x}_{\mu_s}^\ell$. This process is repeated recursively until the coarsest level is reached.

On the coarsest level, $\ell = 1$, an NMM method approximately minimizes h^ℓ using μ_c steps of a nonlinear solution strategy, giving rise to $\mathbf{x}_*^\ell$. Afterwards, the prolongated coarse-level correction $\mathbf{s}_{\mu_s+1}^{\ell+1} := \mathbf{I}_\ell^{\ell+1}(\mathbf{x}_*^\ell - \mathbf{x}_0^\ell)$ is used to update the current iterate $\mathbf{x}_{\mu_s}^{\ell+1}$ on level $\ell+1$. However, before this update is performed, the correction $\mathbf{s}_{\mu_s+1}^\ell$ has to undergo some convergence control. The type of convergence control again depends on the particular type of the NMM method. For example, if the multilevel trust-region method is used, then $\mathbf{s}_{\mu_s+1}^{\ell+1}$ is required to provide a decrease in $h^{\ell+1}$ to be accepted by the algorithm. If a variant of a line-search method is used, then an appropriate step size has to be determined. In the end, the algorithm performs μ_s post-smoothing steps, starting from $\mathbf{x}_{\mu_s+1}^{\ell+1}$ and giving rise to $\mathbf{x}_*^{\ell+1}$. This process is again repeated on all levels until the finest level is reached.

3 Construction of coarse-level models

On each level ℓ, the NMM methods minimize the model $h^\ell : \mathbb{R}^{n^\ell} \to \mathbb{R}$ approximately. The result of this minimization, the iterate $\mathbf{x}_*^\ell$, is then used to construct the search

Algorithm 1 NMM($\ell,\ h^\ell,\ \mathbf{x}_0^\ell$)

Require: $\ell \in \mathbb{N}, h^\ell : \mathbb{R}^{n^\ell} \to \mathbb{R},\ \mathbf{x}_0^\ell \in \mathbb{R}^{n^\ell}$ and $\mu_s, \mu_c \in \mathbb{N}$
1: $\mathbf{x}_{\mu_s}^\ell$ = Nonlinear_smoothing($h^\ell,\ \mathbf{x}_0^\ell,\ \mu_s$)
2: Construct $h^{\ell-1}$ using $\mathbf{x}_{\mu_s}^\ell$, and $\nabla h^\ell(\mathbf{x}_{\mu_s}^\ell)$
3: **if** $\ell = 2$ **then**
4: $\quad \mathbf{x}_*^{\ell-1}$ = Nonlinear_solve($h^{\ell-1},\ \mathbf{P}_\ell^{\ell-1}\mathbf{x}_{\mu_s}^\ell,\ \mu_c$)
5: **else**
6: $\quad \mathbf{x}_*^{\ell-1}$ = NMM($\ell - 1, h^{\ell-1},\ \mathbf{P}_\ell^{\ell-1}\mathbf{x}_{\mu_s}^\ell$)
7: **end if**
8: $\mathbf{x}_{\mu_s+1}^\ell$ = Convergence_control($h^\ell, \mathbf{x}_{\mu_s}^\ell, \boldsymbol{I}_{\ell-1}^\ell(\mathbf{x}_*^{\ell-1} - \mathbf{P}_\ell^{\ell-1}\mathbf{x}_{\mu_s}^\ell)$)
9: $\mathbf{x}_*^\ell$ = Nonlinear_smoothing($h^\ell,\ \mathbf{x}_{\mu_s+1}^\ell,\ \mu_s$)
10: **return** $\mathbf{x}_*^\ell$

direction for the minimization on the next finer level. As a consequence, the overall efficiency of NMM methods depends on the capabilities of the models $\{h^\ell\}_{\ell=1}^L$ to approximate f as accurately as possible.

Given an initial guess $\mathbf{x}_0^\ell = \mathbf{P}_{\ell+1}^\ell \mathbf{x}_{\mu_s}^{\ell+1}$, the model h^ℓ is constructed during each V-cycle by correcting the function f^ℓ, such that the following condition holds:

$$\nabla h^\ell(\mathbf{x}_0^\ell) = \mathbf{R}_{\ell+1}^\ell \nabla h^{\ell+1}(\mathbf{x}_{\mu_s}^{\ell+1}). \tag{2}$$

This ensures that h^ℓ and $h^{\ell+1}$ are locally first-order coherent and that the following relation holds: $\langle \nabla h^\ell(\mathbf{x}_0^\ell), \mathbf{s}^\ell \rangle = \langle \nabla h^{\ell+1}(\mathbf{x}_{\mu_s}^{\ell+1}), \mathbf{I}_\ell^{\ell+1}\mathbf{s}^\ell \rangle$. In this work, we discuss three different approaches for constructing models $\{h^\ell\}_{\ell=1}^L$, namely additive, multiplicative and hybrid. Our discussion considers only the first-order coherent models, constructed using the Taylor approximation of the associated correction function. However, models enforcing higher-order coherency as well as different approximations of the correction function could also be considered.

3.1 An additive approach

Using the additive approach, the coarse-level model $h_{\text{add}}^\ell : \mathbb{R}^{n^\ell} \to \mathbb{R}$ is obtained by correcting the low-cost function f^ℓ as follows

$$h_{\text{add}}^\ell(\mathbf{x}^\ell) = f^\ell(\mathbf{x}^\ell) + \gamma_{\text{add}}^\ell(\mathbf{x}^\ell), \tag{3}$$

where the additive correction function $\gamma_{\text{add}}^\ell : \mathbb{R}^{n^\ell} \to \mathbb{R}$ accounts for the difference between the value of f^ℓ and the fine-level model $h^{\ell+1}$, i.e.,

$$\gamma_{\text{add}}^\ell(\mathbf{x}^\ell) := h^{\ell+1}(\mathbf{I}_\ell^{\ell+1}\mathbf{x}^\ell) - f^\ell(\mathbf{x}^\ell). \tag{4}$$

Unfortunately, the evaluation of $\gamma_{\text{add}}^{\ell}$ at any given $\mathbf{x}^{\ell}$ requires an evaluation of the fine-level model $h^{\ell+1}$ at $\mathbf{I}_{\ell}^{\ell+1}\mathbf{x}^{\ell}$. As a consequence, numerical computations involving h_{add}^{ℓ} are computationally more demanding than computations performed using $h^{\ell+1}$ directly. To ease the computational burden, we evaluate $\gamma_{\text{add}}^{\ell}$ exactly only at the initial coarse-level iterate $\mathbf{x}_0^{\ell} = \mathbf{P}_{\ell}^{\ell+1}\mathbf{x}_{\mu_s}^{\ell+1}$. Thus, we impose

$$\gamma_{\text{add}}^{\ell}(\mathbf{x}_0^{\ell}) := h^{\ell+1}(\mathbf{x}_{\mu_s}^{\ell+1}) - f^{\ell}(\mathbf{x}_0^{\ell}),$$

only at $\mathbf{x}_0^{\ell}$. For any other $\mathbf{x}^{\ell}$, we approximate the correction function $\gamma_{\text{add}}^{\ell}$ by means of the first-order Taylor approximation, defined around $\mathbf{x}_0^{\ell}$ as follows

$$\tilde{\gamma}_{\text{add}}^{\ell}(\mathbf{x}^{\ell}) = \gamma_{\text{add}}^{\ell}(\mathbf{x}_0^{\ell}) + \langle \nabla\gamma_{\text{add}}^{\ell}(\mathbf{x}_0^{\ell}),\ \mathbf{x}^{\ell} - \mathbf{x}_0^{\ell}\rangle.$$

Replacing $\gamma_{\text{add}}^{\ell}$ with $\tilde{\gamma}_{\text{add}}^{\ell}$ in (3) gives rise to

$$h_{\text{add}}^{\ell}(\mathbf{x}^{\ell}) := f^{\ell}(\mathbf{x}^{\ell}) + h^{\ell+1}(\mathbf{x}_{\mu_s}^{\ell+1}) - f^{\ell}(\mathbf{x}_0^{\ell}) + \langle \nabla\gamma_{\text{add}}^{\ell}(\mathbf{x}_0^{\ell}), \mathbf{x}^{\ell} - \mathbf{x}_0^{\ell}\rangle, \tag{5}$$

where

$$\nabla\gamma_{\text{add}}^{\ell}(\mathbf{x}_0^{\ell}) := \mathbf{R}_{\ell+1}^{\ell}\nabla h^{\ell+1}(\mathbf{x}_{\mu_s}^{\ell+1}) - \nabla f^{\ell}(\mathbf{x}_0^{\ell}). \tag{6}$$

Note, the quantity $h^{\ell+1}(\mathbf{x}_{\mu_s}^{\ell+1}) - f^{\ell}(\mathbf{x}_0^{\ell})$ enforces zeroth-order coherence between $h^{\ell+1}$ and h_{add}^{ℓ} at $\mathbf{x}_{\mu_s}^{\ell+1}$ and $\mathbf{x}_0^{\ell}$, respectively, i.e., $h_{\text{add}}^{\ell}(\mathbf{x}_0^{\ell}) = h^{\ell+1}(\mathbf{x}_{\mu_s}^{\ell+1})$. However, this term does not affect the evaluation of the derivatives of h_{add}^{ℓ}, and therefore it is often neglected in practice. We also point out that the term $\nabla\gamma_{\text{add}}^{\ell}(\mathbf{x}_0^{\ell})$, known in the multilevel literature as τ-correction, ensures that condition (2) holds.

3.2 A multiplicative approach

Optimization methods that exploit multiple fidelities often employ multiplicative correction functions [4]. In this case, the low-cost approximation f^{ℓ} associated with level ℓ is made coherent with the model $h^{\ell+1}$ as follows:

$$h_{\text{mult}}^{\ell}(\mathbf{x}^{\ell}) = \gamma_{\text{mult}}^{\ell}(\mathbf{x}^{\ell}) f^{\ell}(\mathbf{x}^{\ell}). \tag{7}$$

Here, the multiplicative correction function $\gamma_{\text{mult}}^{\ell} \colon \mathbb{R}^{n^{\ell}} \to \mathbb{R}$ is given as

$$\gamma_{\text{mult}}^{\ell}(\mathbf{x}^{\ell}) := \frac{h^{\ell+1}(\mathbf{I}_{\ell}^{\ell+1}\mathbf{x}^{\ell}) + \kappa}{f^{\ell}(\mathbf{x}^{\ell}) + \kappa}, \tag{8}$$

where $\kappa \approx \epsilon$ ensures numerical stability as the value of $f^{\ell}(\mathbf{x}^{\ell})$ approaches zero.

Similar to the additive approach, evaluating $\gamma_{\text{mult}}^{\ell}$ precisely at all coarse-level iterates is computationally expensive. Therefore, we impose (8) only at $\mathbf{x}_0^{\ell}$, i.e.,

$$\gamma^{\ell}_{\text{mult}}(\mathbf{x}^{\ell}_0) := \frac{h^{\ell+1}(\mathbf{x}^{\ell+1}_{\mu_s}) + \kappa}{f^{\ell}(\mathbf{x}^{\ell}_0) + \kappa},$$

where we explored that $\mathbf{x}^{\ell+1}_{\mu_s} = \mathbf{I}^{\ell+1}_{\ell}\mathbf{x}^{\ell}_0$. At any other iterate $\mathbf{x}^{\ell}$, we approximate $\gamma^{\ell}_{\text{mult}}$ by means of the first-order Taylor approximation, defined around $\mathbf{x}^{\ell}_0$ as

$$\tilde{\gamma}^{\ell}_{\text{mult}}(\mathbf{x}^{\ell}) = \gamma^{\ell}_{\text{mult}}(\mathbf{x}^{\ell}_0) + \langle \nabla\gamma^{\ell}_{\text{mult}}(\mathbf{x}^{\ell}_0),\ \mathbf{x}^{\ell} - \mathbf{x}^{\ell}_0\rangle. \tag{9}$$

Replacing $\gamma^{\ell}_{\text{mult}}$ with $\tilde{\gamma}^{\ell}_{\text{mult}}$ in (7) then gives rise to the first-order coherent model

$$h^{\ell}_{\text{mult}}(\mathbf{x}^{\ell}) := \tilde{\gamma}^{\ell}_{\text{mult}}(\mathbf{x}^{\ell})\ \ f^{\ell}(\mathbf{x}^{\ell}). \tag{10}$$

The numerical evaluation of $\tilde{\gamma}^{\ell}_{\text{mult}}$ amounts to

$$\tilde{\gamma}^{\ell}_{\text{mult}}(\mathbf{x}^{\ell}) := \frac{h^{\ell+1}(\mathbf{x}^{\ell+1}_{\mu_s}) + \kappa}{f^{\ell}(\mathbf{x}^{\ell}_0) + \kappa} + \langle \nabla\gamma^{\ell}_{\text{mult}}(\mathbf{x}^{\ell}_0),\ \mathbf{x}^{\ell} - \mathbf{x}^{\ell}_0\rangle,$$

where $\nabla\gamma^{\ell}_{\text{mult}}(\mathbf{x}^{\ell}_0)$ is given by

$$\nabla\gamma^{\ell}_{\text{mult}}(\mathbf{x}^{\ell}_0) := \frac{1}{f^{\ell}(\mathbf{x}^{\ell}_0) + \kappa}\left(\mathbf{R}^{\ell}_{\ell+1}\nabla h^{\ell+1}(\mathbf{x}^{\ell+1}_{\mu_s})\right) - \frac{h^{\ell+1}(\mathbf{x}^{\ell+1}_{\mu_s}) + \kappa}{(f^{\ell}(\mathbf{x}^{\ell}_0) + \kappa)^2}\nabla f^{\ell}(\mathbf{x}^{\ell}_0).$$

Straightforward calculations show that model h^{ℓ}_{mult}, defined by (10), is zeroth-order and first-order coherent with $h^{\ell+1}$ at $\mathbf{x}^{\ell}_0$ and $\mathbf{x}^{\ell+1}_{\mu_s}$, respectively.

3.3 A hybrid approach

From a computational point of view, additive and multiplicative approaches are comparable. However, their behavior is very different. The additive approach adds new terms to f^{ℓ}, which can be interpreted as uniform translation (zeroth-order), and rotation (first-order) of the function graph; see also Fig. 1. In contrast, the multiplicative approach introduces skewing, which might not be desirable if f and f^{ℓ} are in good agreement, at least locally. However, if functions f^{ℓ} and f are not in good agreement, then additional skewing can be beneficial [3], e.g., if the polynomial order of f is higher than the polynomial order of f^{ℓ}. Moreover, multiplication of f^{ℓ} with $\tilde{\gamma}^{\ell}_{\text{mult}}$ can introduce new minima on level ℓ, where $\ell < L$. For instance, let us suppose that f^{ℓ} is a second-order polynomial. Its multiplication with $\tilde{\gamma}^{\ell}_{\text{mult}}$ increases the order of the polynomial, i.e., we obtain a model h^{ℓ}_{mult} which is quartic and has, in general, more minima than quadratic function.

In general, it is not known a priori whether the additive or the multiplicative model is more suitable for a given optimization problem. To overcome this difficulty, a hybrid approach [6] can be employed. A coarse-level model h^{ℓ}_{mix} is then obtained

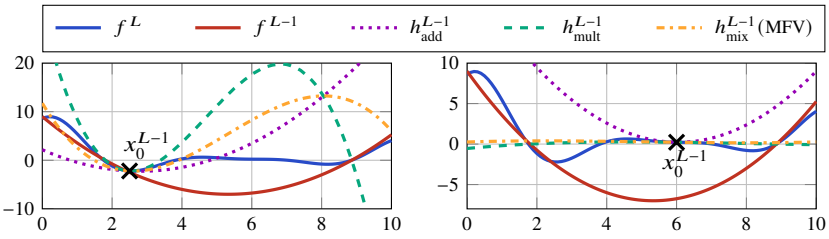


Fig. 1 Coarse-level models constructed around $x_0^{L-1} = 2.5$ and $x_0^{L-1} = 6.0$.

as a convex combination of the additive h^ℓ_{add} and the multiplicative h^ℓ_{mult} models, i.e.,

$$h^\ell_{\text{mix}}(\mathbf{x}^\ell) := w^\ell_{\text{add}}\, h^\ell_{\text{add}}(\mathbf{x}^\ell) + w^\ell_{\text{mult}}\, h^\ell_{\text{mult}}(\mathbf{x}^\ell), \tag{11}$$

where $w^\ell_{\text{add/mult}} \in \mathbb{R}$ and $w^\ell_{\text{add}} + w^\ell_{\text{mult}} = 1$. In order to maximize the approximation properties of h^ℓ_{mix}, the weights $w^\ell_{\text{add}}, w^\ell_{\text{mult}}$ have to be chosen carefully. Below, we describe two different strategies for selecting the values w^ℓ_{add} and w^ℓ_{mult}.

3.3.1 Matching function values (MFV) at the previously evaluated fine-level iterate

Following [2], the weights $w^\ell_{\text{add}}, w^\ell_{\text{mult}}$ can be selected by matching the function value at the previously evaluated fine-level iterate, denoted by $\mathbf{x}_p^{\ell+1}$, as in

$$w^\ell_{\text{add}} = \frac{h^{\ell+1}(\mathbf{x}_p^{\ell+1}) - h^\ell_{\text{mult}}(\mathbf{x}_0^\ell)}{h^\ell_{\text{add}}(\mathbf{x}_0^\ell) - h^\ell_{\text{mult}}(\mathbf{x}_0^\ell)} \qquad \text{and} \qquad w^\ell_{\text{mult}} = 1 - w^\ell_{\text{add}}. \tag{12}$$

From a computational point of view, evaluating (12) is cheap as $h^{\ell+1}(\mathbf{x}_p^{\ell+1})$ is readily available, for instance from the $\mu_s - 1$ pre-smoothing step performed on level $\ell + 1$.

3.3.2 Bayesian updating approach

To maximize the approximation properties of h^ℓ_{mix}, it might be beneficial to take into account the history of the d^ℓ previously evaluated fine-level iterates [3]. Therefore, we consider the dataset $\mathcal{D}^\ell = \{(h^{\ell+1}(\mathbf{x}_p^{\ell+1}), h^\ell_{\text{add}}(\mathbf{P}^\ell_{\ell+1}\mathbf{x}_p^{\ell+1}), h^\ell_{\text{mult}}(\mathbf{P}^\ell_{\ell+1}\mathbf{x}_p^{\ell+1})\}_{p=1}^{d^\ell}$, where each sample contains the function value of $h^{\ell+1}$ at $\mathbf{x}_p^{\ell+1}$, as well as the function values of the coarse-level models $h^\ell_{\text{add/mult}}$ obtained at $\mathbf{P}^\ell_{\ell+1}\mathbf{x}_p^{\ell+1}$. In this work, we construct $\mathcal{D}^\ell$ by taking into account the last d^ℓ iterates which were transferred from level $\ell + 1$ to level ℓ. For example, if $d^\ell = 3$, then $\mathcal{D}^\ell$ is constructed by taking into account the iterate $\mathbf{x}_p^{\ell+1} = \mathbf{x}_{\mu_s}^{\ell+1}$, obtained as a result of the pre-smoothing step during the previous three V-cycles. For simplicity, we use the notation $d^\ell = \infty$ to denote all previous V-cycles.

Having constructed the dataset $\mathcal{D}^\ell$, we can now employ the Bayesian posterior updating approach [3] to determine the values of $w^\ell_{\text{add/mult}}$. Starting from $w^\ell_{\text{add/mult}} = 0.5$, the weights are updated every time the model h^ℓ is constructed as follows:

$$w^\ell_{\text{add/mult}} = \frac{w^\ell_{\text{add/mult}} \psi^\ell_{\text{add/mult}}}{w^\ell_{\text{mult/add}} \psi^\ell_{\text{mult/add}} + w^\ell_{\text{add/mult}} \psi^\ell_{\text{add/mult}}}. \tag{13}$$

The model likelihoods $\psi^\ell_{\text{add/mult}}$ in (13) are evaluated as

$$\psi^\ell_{\text{add/mult}} = \left(2\pi\sigma^2_{\text{add/mult}}\right)^{-d^\ell/2} \exp(-d^\ell/2), \tag{14}$$

and the maximum likelihood estimator of the model variance is given by

$$\sigma^2_{\text{add/mult}} = \frac{1}{d^\ell} \sum_{p=1}^{d^\ell} (h^{\ell+1}(\mathbf{x}^{\ell+1}_p) - h^\ell_{\text{add/mult}}(\mathbf{P}^\ell_{\ell+1}\mathbf{x}^{\ell+1}_p)). \tag{15}$$

4 Numerical results and discussion

In this section, we investigate the influence of different coarse-level models on the performance of the NMM method using numerical examples from the field of supervised learning, namely classification using ResNets [8]. Given a dataset $\mathcal{S} = \{(\mathbf{z}_s, \mathbf{c}_s)\}_{s=1}^{n_s}$, where $\mathbf{z}_s \in \mathbb{R}^{n_{in}}$ and $\mathbf{c}_s \in \mathbb{R}^{n_{out}}$, our goal is to find parameters $\mathbf{x} \in \mathbb{R}^n$ of a ResNet, defined as $\text{RN}\colon \mathbb{R}^{n_{in}} \times \mathbb{R}^n \to \mathbb{R}^{n_{out}}$, by solving the following minimization problem:

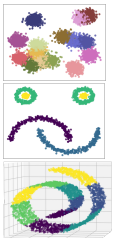
$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) := \frac{1}{n_s} \sum_{s=1}^{n_s} g(\text{RN}(\mathbf{z}_s, \mathbf{x}), \mathbf{c}_s), \tag{16}$$

where g denotes the cross-entropy loss function.

Since (16) is a non-convex function, we choose the NMM method to be a variant of the RMTR method [7]. The multilevel hierarchy and transfer operators are constructed by leveraging the fact that the ResNet can be interpreted as a forward Euler discretization of an ordinary differential equation; see [10, 5] for details. Here, we construct a hierarchy of ResNets by uniformly refining a ResNet with three layers three times. Fig. 4 demonstrates the number of effective gradient evaluations[1] of the RMTR method with respect to different coarse-level models for three different datasets.

As we can observe, the choice of the coarse-level model has a significant impact on the overall efficiency of the multilevel method. For all three examples, hybrid approaches outperform purely additive and multiplicative ones. In terms of hybrid

[1] The number of effective gradient evaluations is obtained as $\sum_{\ell=1}^{L} 2^{\ell-L} W_\ell C_L$, where C_L represents a cost associated with an evaluation of the gradient on the level L, W_ℓ describes a number of gradient evaluations performed on a level ℓ, and $2^{\ell-L}$ is a coarsening factor in 1D.



Model/Example	Blobs	Smiley	Spiral
h_{add}	29 ± 5.3%	676 ± 11.2%	203 ± 12.3%
h_{mult}	32 ± 6.1%	485 ± 15.1%	153 ± 15.9%
$h_{\text{mix}}(w = 0.5)$	38 ± 4.8%	404 ± 10.3%	297 ± 11.3%
$h_{\text{mix}}(\text{MFV})$	25 ± 4.2%	352 ± 6.5%	**123 ± 7.1**%
$h_{\text{mix}}(d^\ell = 5)$	25 ± 3.4%	514 ± 6.3%	197 ± 6.8%
$h_{\text{mix}}(d^\ell = 20)$	**24 ± 2.9**%	471 ± 7.7%	156 ± 7.4%
$h_{\text{mix}}(d^\ell = \infty)$	25 ± 3.8%	**301 ± 6.9**%	126 ± 9.9%

Fig. 2 *Left:* Blobs, Smiley, and Spiral datasets (*Top* to *Down*). Each class is illustrated by different color. *Right:* The average number of effective gradient evaluations of the RMTR method (4 levels). Averages are obtained from 5 independent runs.

models, we observe that the Bayesian approach performs similar, or superior to MFV, especially if all prior fine-level iterates are considered ($d^\ell = \infty$).

Given our (limited) numerical experience, we believe that employing hybrid, and possibly other types of novel coarse-level models, provides a promising future direction for improving the efficiency and the reliability of NMM methods.

Acknowledgements This work was supported by the Swiss National Science Foundation under the project "Multilevel training of DeepONets — multiscale and multiphysics applications" (grant no. 206745).

References

1. Brandt, A. Multi-level adaptive solutions to boundary-value problems. *Mathematics of computation* **31**(138), 333–390 (1977).
2. Eldred, M. and Dunlavy, D. Formulations for surrogate-based optimization with data fit, multifidelity, and reduced-order models. In: *11th AIAA/ISSMO multidisciplinary analysis and optimization conference*, 7117 (2006).
3. Fischer, C. C., Grandhi, R. V., and Beran, P. S. Bayesian-Enhanced Low-Fidelity Correction Approach to Multifidelity Aerospace Design. *AIAA Journal* **56**(8), 3295–3306 (2018).
4. Forrester, A. I. and Keane, A. J. Recent advances in surrogate-based optimization. *Progress in aerospace sciences* **45**(1-3), 50–79 (2009).
5. Gaedke-Merzhäuser, L., Kopaničáková, A., and Krause, R. Multilevel minimization for deep residual networks. *ESAIM. Proceedings and Surveys* **71**, 131 (2021).
6. Gano, S. E., Renaud, J. E., and Sanders, B. Hybrid variable fidelity optimization by using a kriging-based scaling function. *Aiaa Journal* **43**(11), 2422–2433 (2005).
7. Gratton, S., Sartenaer, A., and Toint, P. L. Recursive Trust-Region Methods for Multiscale Nonlinear Optimization. *SIAM Journal on Optimization* **19**(1), 414–444 (2008).
8. He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778 (2016).
9. Ho, C. P., Kočvara, M., and Parpas, P. Newton-type multilevel optimization method. *Optimization Methods and Software* 1–34 (2019).
10. Kopaničáková, A. and Krause, R. Globally Convergent Multilevel Training of Deep Residual Networks. *SIAM Journal on Scientific Computing* (0), S254–S280 (2022).
11. Nash, S. G. A multigrid approach to discretized optimization problems. *Optimization Methods and Software* **14**(1-2), 99–116 (2000).
12. Yavneh, I. and Dardyk, G. A Multilevel Nonlinear Method. *SIAM Journal on Scientific Computing* **28**(1), 24–46 (2006).

Nonlinear Schwarz Preconditioning for Quasi-Newton Methods

Hardik Kothari

1 Introduction

In this work, we consider a nonlinear preconditioning strategy for Quasi-Newton (QN) methods. QN methods are a class of root-finding methods, where the full Jacobian is replaced with an approximation. In the context of this work, we consider secant methods, which take into account a variable number of secant equations at each nonlinear iteration. These types of methods are mostly used if the Jacobian of the nonlinear system is expensive to evaluate, requires more storage, or is simply unavailable. Such scenarios are often encountered while solving coupled multiphysics problems that require higher-order discretization, inverse problems, optimal control problems, training of deep neural networks, etc.

To this aim, we consider the following abstract nonlinear minimization problem:

$$\text{Find } x^* \in \mathcal{V} \text{ that minimizes} \quad \Psi(x), \tag{1}$$

where $\Psi\colon \mathcal{V} \to \mathbb{R}$ denotes a bounded, twice continuously differentiable objective function. The objective function Ψ is obtained by a finite element (FE) discretization of a nonlinear optimization problem, and $\mathcal{V}$ denotes some FE space. To solve (1), we can consider the first-order optimality condition for the function $\Psi(x)$, and then a nonlinear iterative method can be employed to find the root of the nonlinear equation $F(x^*) = 0$, where $F\colon \mathcal{V} \to \mathcal{V}'$ is defined as $F(\cdot) \equiv \nabla\Psi(\cdot)$. We also note that the Hessian of the objective function $\nabla^2\Psi$ is equivalent to the Jacobian F'. Depending on the properties of the objective function Ψ, multiple approaches can be considered to solve (1), for example, Newton's method and its variants; nonlinear Krylov methods; secant methods [11].

Among all these methods, Newton's method is one of the most popular methods to solve such problems due to its locally quadratic convergence property. However, its convergence might suffer if the objective function is highly nonlinear with locally stiff or unbalanced nonlinearities and/or if the initial guess is far from a local

Hardik Kothari
Università della Svizzera italiana, Switzerland, e-mail: hardik.kothari@usi.ch

minimizer. In recent years, some nonlinear preconditioning strategies have been developed to accelerate the convergence of Newton's method, e.g.: Additive Schwarz Preconditioned Inexact Newton (ASPIN) [1]; Nonlinear Elimination Preconditioned Inexact Newton (NEPIN) [2]; Restricted Additive Schwarz Preconditioned Exact Newton (RASPEN) [5]. Similarly, in the context of optimization methods, nonlinear preconditioning strategies have been considered to improve the convergence of a nonlinear Krylov method [3] and a quasi-Newton (QN) method [4]. To the best of our knowledge, unlike the ASPIN, NEPIN, and RASPEN methods, the nonlinear domain decomposition-based preconditioners have not yet been considered for nonlinear Krylov methods and QN methods.

In this work, we apply the nonlinear Schwarz preconditioning strategies to accelerate the convergence of the standard QN methods. We explore the "left" and "right" nonlinear preconditioning strategies and discuss the necessary modifications to the QN framework. Finally, we examine the efficiency of the preconditioned QN methods by means of some numerical experiments.

2 Preconditioned Quasi-Newton methods

In this section, we discuss QN methods, nonlinear restricted additive Schwarz (NRAS) methods, and how to nonlinearly precondition QN methods.

Quasi-Newton methods: Quasi-Newton (QN) methods are quite popular in the optimization community, especially when the Hessian of the underlying minimization problem is unavailable or very expensive to evaluate. In QN methods, the evaluation of the Hessian is replaced by its low-rank approximation. This low-rank approximation of the Hessian is carried out using a secant condition. At each iteration, the approximation of the Hessian B is constructed using the information between subsequent iterations. The approximate Hessian, $B^{(k+1)}$, satisfies the secant equation

$$B^{(k+1)} s^{(k)} = y^{(k)}, \tag{2}$$

where $s^{(k)} = x^{(k+1)} - x^{(k)}$ and $y^{(k)} = F(x^{(k+1)}) - F(x^{(k)})$. As the secant equation is not sufficient to uniquely determine the matrix B, additional constraints have to be imposed on B, which gives rise to different variants of the QN methods. In this work, we consider two types of multi-secant methods, namely the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, and the Andersen acceleration (AA) method. As one of the motivations of this work is reducing the memory footprint of the algorithm, the limited-memory variant of the BFGS method (L-BFGS), and of the AA method, becomes a natural choice. These methods utilize only the m pairs of the vectors $\{s^{(i)}, y^{(i)}\}_{i=k-m}^{k-1}$ from the m most recent iterations to construct the approximate Hessian. We note that the original AA method is not proposed in the context of the optimization but its interpretation as a QN method is established in [6, 12]. The approximate Hessians obtained by the L-BFGS method and the type-I AA method (AA-I) at an iterate $k+1$ can be written in a compact matrix format in the following manner:

$$\begin{aligned} &\text{(L-BFGS)} \quad B^{(k+1)} = B_0 - \begin{bmatrix} B_0 S_k & Y_k \end{bmatrix} \begin{bmatrix} S_k B_0 S_k & L_k \\ L_k^\top & -D_k \end{bmatrix}^{-1} \begin{bmatrix} S_k^\top B_0 \\ Y_k^\top \end{bmatrix}, \\ &\text{(AA-I)} \qquad B^{(k+1)} = I + (Y_k - S_k)(S_k^\top S_k)^{-1} S_k^\top. \end{aligned} \tag{3}$$

Here, $S_k := [s^{(k-m)}, \ldots, s^{(k-1)}]$, $Y_k := [y^{(k-m)}, \ldots, y^{(k-1)}]$, L_k and D_k denote the strictly lower triangular, and the diagonal part of matrix $S_k^\top Y_k$, B_0 denotes some initial Hessian approximation. In order to find the search direction $p^{(k)}$, we need the inverse of the approximate Hessians, which is generally obtained using the Sherman–Morrison–Woodbury formula. To accelerate the convergence speed of these methods, we propose to precondition the QN methods with an NRAS method.

Nonlinear Restricted Additive Schwarz Methods: We consider a decomposition of the domain Ω into n non-overlapping domains $\{\Omega_i\}_{i=1}^n$ and overlapping domains as $\{\Omega_i^\delta\}_{i=1}^n$, such that $\Omega_i \subset \Omega_i^\delta$, here δ denotes the size of the overlap. The FE spaces associated with the overlapping domains are defined as $\{\mathcal{V}_i^\delta\}_{i=1}^n$, $\mathcal{V}_i^\delta \subset \mathcal{V}$. On these overlapping subspaces, we define the restriction and prolongation operators as $R_i^\delta \colon \mathcal{V} \to \mathcal{V}_i^\delta$ and $P_i^\delta \colon \mathcal{V}_i^\delta \to \mathcal{V}$, respectively. We note that for $\delta = 0$, the overlapping decomposition degenerates to a non-overlapping decomposition, i.e., $\Omega_i = \Omega_i^0$. The prolongation operator on the non-overlapping subspaces is termed as *restricted* prolongation operator, $P_i^0 \colon \mathcal{V}_i^0 \to \mathcal{V}$. The overlapping and the non-overlapping decomposition of the subspaces ensures that the partition of unity is satisfied, e.g., $\sum_{i=1}^n P_i^0 R_i^\delta = I$.

Now, we can define a local nonlinear minimization problem restricted to each overlapping subspace as follows. For a given initial guess $x_i^{(0)} = R_i^\delta x^{(k)}$:

$$\text{Find } x_i^* \in \mathcal{V}_i^\delta \text{ that minimizes} \quad \Psi_i^\delta(x_i). \tag{4}$$

Here, $\Psi_i^\delta \colon \mathcal{V}_i^\delta \to \mathbb{R}$ is the restriction of the objective function Ψ to the subspace $\mathcal{V}_i^\delta$. Once the minimization problem is approximately solved on each subdomain, the global iterate is updated in the following manner

$$x^{(k+1)} = x^{(k)} + \alpha^{(k)} \sum\nolimits_{i=1}^n P_i^0 (x_i^* - R_i^\delta x^{(k)}). \tag{5}$$

We note that the problem (4) is solved on the overlapping subdomains, but the correction is accepted only on the non-overlapping part. Furthermore, to construct a two-level variant of the NRAS method, we define a coarse space $\mathcal{V}_0 \subset \mathcal{V}$ and the restriction and the prolongation operators $R_0 \colon \mathcal{V}' \to \mathcal{V}_0'$ and $P_0 \colon \mathcal{V}_0 \to \mathcal{V}$, where $P_0^\top = R_0$. Also, we define a projection operator $\Pi_0 \colon \mathcal{V} \to \mathcal{V}_0$ to transfer the primal variables to the coarse level. The objective function on the coarse level is defined as $\Psi_0 \colon \mathcal{V}_0 \to \mathbb{R}$, which denotes a discretization of the function Ψ on the space $\mathcal{V}_0$. The coarse space plays an important role in the NRAS method, as it allows global communication between the subdomains and ensures the scalability of the algorithm. In this work, the coarse-level objective function is defined in the spirit of the full approximation scheme (FAS) or the MG-Opt method [10]. The coarse-level function is constructed by adding a first-order consistency term, which is also called

a "defect" in the context of FAS. Thus, the optimization problem on the coarse level is defined as follows. For an initial guess $x_0^{(0)} = \Pi_0 x^{(k)}$:

$$\text{Find } x_0^* \text{ that minimizes} \quad \hat{\Psi}_0(x_0) := \Psi_0(x_0) + \langle \delta g_0, x_0 \rangle \tag{6}$$

where $\delta g_0 = R_0 \nabla\Psi(x^{(k)}) - \nabla\Psi_0(\Pi_0 x^{(k)})$ denotes the first-order consistency term. Additionally, we employ a multiplicative variant of the coarse-level update, where we first approximately solve the problem on the coarse level and bootstrap the initial guess on the subdomains using the approximate solution from the coarse level. The update step for the two-level NRAS is given as follows:

$$x^{(k+1)} = x^{(k)} + \hat{\alpha} T_0(x^{(k)}) + \alpha \sum_{i=1}^{n} P_i^0 (x_i^* - R_i^\delta (x^{(k)} + \hat{\alpha} T_0(x^{(k)}))) \tag{7}$$

where $T_0(x^{(k)}) = P_0(x_0^* - \Pi_0 x^{(k)})$ denotes the coarse-level correction. We note that in (7), $\hat{\alpha}$ and α are computed using a line-search method, while x_i^* and x_0^* denote the approximate solutions of problems (4) and (6), respectively.

Nonlinear preconditioning: In this section, we discuss strategies to nonlinearly preconditioned quasi-Newton methods. Recall, we seek $x^* \in \mathcal{V}$ such that $F(x^*) = 0$. A nonlinear preconditioner G of the residual function F is defined such that the preconditioner G approximates the inverse of the residual i.e., $G \approx F^{-1}$. Practically, it is not possible to obtain such a preconditioning operator G explicitly but, generally, such an operator can be defined implicitly as a fixed-point nonlinear iterative scheme, given as $x = G(x)$. The operator, G, can be applied to the nonlinear residual as either a "left" or a "right" preconditioner, which gives rise to two different nonlinearly preconditioned residuals

$$\mathcal{F}_L(x) = G_L(F(x)) = x - G(x), \qquad \mathcal{F}_R(x) = F(G_R(x)) = F(G(x)). \tag{8}$$

We remark that the left preconditioning operator is not equivalent to a fixed-point nonlinear iterative method $G_L \neq G$, while the right preconditioning operator is a fixed-point iteration scheme $G_R = G$. The ASPIN and RASPEN methods are the "left" preconditioned methods, where the nonlinear residual is first computed using a fixed-point method, and Newton's method is used to solve the equation $\mathcal{F}_L(x) = 0$. The NEPIN method [2], nonlinear FETI-DP and BDDC methods [7] are considered to be the "right" preconditioned methods.

We define generic iterations for both types of preconditioning strategies. The iteration for the preconditioned QN method can be achieved by replacing the residual F with the preconditioned residual given as $\mathcal{F}_{L/R}$. For a given initial iterate $x^{(k)}$, we first compute $x^{(+)}$ using a NRAS method, i.e., $x^{(+)} = G(x^{(k)})$. Once the preconditioning step has been carried out, we can define the iteration for the "left-preconditioned" QN method as,

$$x^{(k+1)} = x^{(k)} - \alpha^{(k)} \left(B_L^{(k)}\right)^{-1} \mathcal{F}_L(x^{(k)}), \quad \text{where } \mathcal{F}_L(x^{(k)}) = x^{(k)} - G(x^{(k)}). \tag{9}$$

Algorithm 1: Nonlinearly Preconditioned QN method

Data: $F : \mathcal{V} \to \mathcal{V}'$, $x^{(0)} \in \mathcal{V}$, $k \leftarrow 0$
Result: $x^{(k)}$
while $\|F(x^{(k)})\| \geqslant \epsilon_{rtol}\|F(x^{(0)})\|$ **do**
  For given $x^{(k)}$, compute the preconditioned residual $\mathcal{F}_{L/R}(x^{(k)})$
  Compute direction using L-BFGS/AA-I approximation of preconditioned Hessian $p_{L/R}^{(k)} \leftarrow -\left(B_{L/R}^{(k)}\right)^{-1}\mathcal{F}_{L/R}x^{(k)}$
  Find $\alpha^{(k)}$ using a line-search algorithm
  Update iterate: $x^{(k+1)} \leftarrow x^{(k)} + \alpha^{(k)}p_L^{(k)}$ or $x^{(k+1)} \leftarrow x^{(+)} + \alpha^{(k)}p_R^{(k)}$
  Compute $s_{L/R}^{(k)}$ as in (12) and $y_{L/R}^{(k)}$ using (13)
  Update the history of secant pairs $\{s_{L/R}^{(k)}, y_{L/R}^{(k)}\}$
  Update $k \leftarrow k+1$

The update process for the "right-preconditioned" QN method differs from the "left-preconditioning" approach. The iteration for the "right-preconditioned" QN method is given as

$$x^{(k+1)} = x^{(+)} - \alpha^{(k)}\left(B_R^{(k)}\right)^{-1}\mathcal{F}_R(x^{(k)}), \quad \text{where } \mathcal{F}_R(x^{(k)}) = F(G(x^{(k)})). \tag{10}$$

In (9) and (10), we compute $\alpha^{(k)}$ using a line-search method. We note that, the operator G can be explicitly given by (5) and (7) for one-level and two-level NRAS preconditioner, respectively. Here, $B_L^{(k)}$ and $B_R^{(k)}$ denote the approximation of the "left" and "right" preconditioned Hessians, respectively. The QN method aims to approximate the Hessian of the underlying optimization function utilizing a set of vectors $\{s^k, y^k\}$. As we have preconditioned the QN method, we also have to change the underlying secant equation and corresponding secant pairs. The corresponding secant equations for the "left" and the "right" preconditioned systems are now given as

$$B_L^{(k+1)}s_L^{(k)} = y_L^{(k)}, \qquad B_R^{(k+1)}s_R^{(k)} = y_R^{(k)}. \tag{11}$$

From (9) and (10), it is clear that $s_{L/R}^{(k)}$ at each iteration are defined as corrections, which are given as

$$s_L^{(k)} = x^{(k+1)} - x^{(k)}, \qquad s_R^{(k)} = x^{(k+1)} - x^{(+)}. \tag{12}$$

Now, we focus our attention on the computation of $y_{L/R}^{(k)}$, which are defined as the difference between the preconditioned residuals

$$y_L^{(k)} = \mathcal{F}_L(x^{(k+1)}) - \mathcal{F}_L(x^{(k)}), \qquad y_R^{(k)} = F(x^{(k+1)}) - F(x^{(+)}). \tag{13}$$

We note that for the "right" preconditioning approach, the nonlinear preconditioner can be simplified as $\mathcal{F}_R(x^{(k)}) = F(G(x^{(k)})) = F(x^{(+)})$, and the iteration in (10) can be further simplified as

$$x^{(k+1)} = x^{(+)} - \alpha^{(k)} \left(B_R^{(k)}\right)^{-1} F(x^{(+)}).$$

This update process can be interpreted as a half iteration, while the first half of the iteration is the preconditioning step $x^{(+)} = G(x^{(k)})$. Hence, the "right-preconditioned" QN method should only construct the approximation of the Hessian for the second half of the iteration.

A sketch of the nonlinearly preconditioned quasi-Newton method is provided in Algorithm 1.

3 Numerical experiments

We investigate the performance of the nonlinearly preconditioned QN method through some numerical experiments. To this aim, we consider a domain $\Omega = (0,1)^2$ with the boundary Γ. The boundary Γ is decomposed into four parts: top ($\Gamma_t = [0,1] \times \{1\}$), bottom ($\Gamma_b = [0,1] \times \{0\}$), left ($\Gamma_l = \{0\} \times [0,1]$) and right ($\Gamma_r = \{1\} \times [0,1]$). We use the discretize-then-optimize approach, where the discretization is done with the first-order FE method using a mesh with 200×200 quadrilateral elements. The coarse level is also constructed with the same approach, where a mesh with 10×10 elements is employed for discretization.

Minimal surface: This experiment aims to find the surface of the minimal area described by the function u by solving the following minimization problem:

$$\begin{aligned} &\min_{u \in H^1(\Omega)} \Psi_M(u) = \int_\Omega \sqrt{(1 + \|\nabla u\|^2)}\, dx, \\ &\text{subject to } \begin{cases} u = -0.5 \sin(2\pi x_2) \text{ on } \Gamma_l, \;\; u = 0.5 \sin(2\pi x_2) \text{ on } \Gamma_r, \\ u = -0.5 \sin(2\pi x_1) \text{ on } \Gamma_b, \;\; u = 0.5 \sin(2\pi x_1) \text{ on } \Gamma_t. \end{cases} \end{aligned} \tag{14}$$

Setup for the solution methods: As we aim to study the behavior of the preconditioned QN method, we use a fixed configuration of the TL-NRAS method. The overlap for all experiments is prescribed as $\delta = 2$, and the domain Ω is decomposed into 8 subdomains. The partitioning of the mesh is carried out using the METIS library. The preconditioned QN is terminated if one of these conditions is satisfied: $\|F(x^{(k)})\| < 10^{-7}$ or $\|F(x^{(k)})\| < 10^{-6}\|F(x^{(0)})\|$. The subdomain solvers in the TL-NRAS method employ Newton's method, which terminates if $\|F_i(x_i^{(k)})\| < 10^{-10}$ or $\|F_i(x_i^{(k)})\| < 10^{-1}\|F_i(x_i^{(0)})\|$ is satisfied. On the coarse level, we also employ Newton's method, which terminates if $\|F_0(x_0^{(k)})\| < 10^{-12}$ or $\|F_0(x_0^{(k)})\| < 10^{-10}\|F_0(x_0^{(0)})\|$ is satisfied, also the maximum number of iterations is set to 5. We note, Newton's method can be replaced by a multigrid preconditioned Jacobian-free Newton-Krylov method [9] to reduce the memory requirement of the overall methodology. We employ backtracking line-search algorithm with strong Wolfe condition [11, Alg. 3.1, Eq. (3.7)], with $c_1 = 10^{-4}, c_2 = 0.99$, and the value of ρ is chosen to be 0.5 for global solvers and 0.1 for subdomain solvers. The ex-

Table 1 Number of iterations and the time to solution for the L-BFGS method and the TL-NRAS preconditioned QN methods. (L)/(R) denote left/right preconditioning.

Memory	m = 1		m = 3		m = 5		m = 7		m = 10	
	Time (s)	# Iter	Time (s)	# Iter	Time (s)	# Iter	Time (s)	# Iter	Time (s)	# Iter
L-BFGS	698.16	643	720.01	642	699.53	646	702.62	679	536.40	513
L-BFGS (L)	*301.37*	*25*	288.69	23	296.82	24	288.61	23	300.00	25
L-BFGS (R)	426.21	36	296.99	22	*278.68*	*20*	273.99	*19*	*272.45*	*20*
AA-I (L)	350.60	30	285.56	22	284.20	22	287.44	23	284.43	22
AA-I (R)	374.47	36	*274.40*	*22*	281.74	23	*269.98*	21	281.21	22

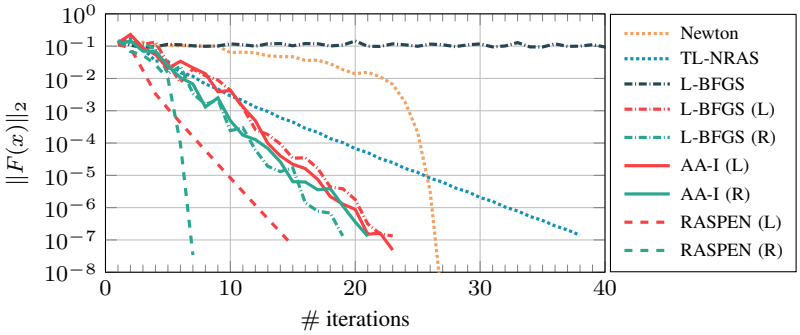


Fig. 1 Convergence history of the L-BFGS method, Newton's method, TL-NRAS method, and TL-NRAS preconditioned QN and RASPEN methods. The QN methods are configured to use the last 7 secant pairs.

periments are carried out using MATLAB on a system with an Intel Core i9-9880H processor, and 16 GB of memory.

Convergence study: In order to study the convergence behavior of the preconditioned QN method, the "left" and the "right" preconditioned variants of the L-BFGS methods and the AA-I method are considered. For this numerical experiment, we store m pairs of secant vectors, where $m \in \{1, 3, 5, 7, 10\}$. Table 1 depicts the time to solution and the required number of iterations to satisfy the termination criterion for different solution methods and different values of m. We have included only preconditioned AA-I method in our study.⋆ From Table 1, it is clear that the preconditioned QN methods outperform the standard L-BFGS method both in terms of the number of iterations and the computational time. Regardless of the number of stored secant pairs, the preconditioned L-BFGS methods and AA-I methods are two times faster than the L-BFGS method. The preconditioned AA-I methods and the preconditioned L-BFGS methods have comparable performance. While the "right" preconditioned L-BFGS methods outperform all other methods if more pairs of secant pairs are used. Figure 1 depicts the convergence history of the preconditioned QN methods, the two-level NRAS method, and Newton's method. We can observe that the preconditioned

⋆ The AA-I method requires factorization of $S_k^\top Y_k$, which is not possible if the successive pairs of $\{s^{(k)}, y^{(k)}\}$ are very similar. To avoid such issues, one can construct the pairs in such a way that successive $s^{(k)}$ are orthogonal [12].

QN method outperforms Newton's method and the L-BFGS method. Also, we can see that the TL-NRAS method has linear convergence, and by employing a QN method as an outer solver we can reduce the number of required iterations in half. Specifically for the left-preconditioning, in comparison with the RASPEN methods the preconditioned QN method shows only mild deterioration in the convergence.

From the performed experiments, we can conclude that the proposed domain decomposition-based preconditioning strategy is quite robust both in the case of the L-BFGS method and the type-I AA method. This works provides a promising future direction for problems when memory is a limiting factor, for example for solving the phase field fracture problems [8] or for the training of deep neural networks.

Acknowledgements This work is supported by the Swiss National Science Foundation (SNSF) and the Deutsche Forschungsgemeinschaft for their through the project SPP 1962 "Stress-Based Methods for Variational Inequalities in Solid Mechanics: Finite Element Discretization and Solution by Hierarchical Optimization" [186407]. We also acknowledge the support of Platform for Advanced Scientific Computing through the project FraNetG: Fracture Network Growth and SNSF through the project ML2 - Multilevel and Domain Decomposition Methods for Machine Learning [197041].

References

1. Cai, X.-C. and Keyes, D. E. Nonlinearly Preconditioned Inexact Newton Algorithms. *SIAM Journal on Scientific Computing* **24**(1), 183–200 (2002).
2. Cai, X.-C. and Li, X. Inexact Newton Methods with Restricted Additive Schwarz Based Nonlinear Elimination for Problems with High Local Nonlinearity. *SIAM Journal on Scientific Computing* **33**(2), 746–762 (2011).
3. De Sterck, H. and Howse, A. Nonlinearly Preconditioned Optimization on Grassmann Manifolds for Computing Approximate Tucker Tensor Decompositions. *SIAM Journal on Scientific Computing* **38**(2), A997–A1018 (2016).
4. De Sterck, H. and Howse, A. J. Nonlinearly preconditioned L-BFGS as an acceleration mechanism for alternating least squares with application to tensor decomposition. *Numerical Linear Algebra with Applications* **25**(6), e2202 (2018).
5. Dolean, V., Gander, M. J., Kheriji, W., Kwok, F., and Masson, R. Nonlinear Preconditioning: How to Use a Nonlinear Schwarz Method to Precondition Newton's Method. *SIAM Journal on Scientific Computing* **38**(6), A3357–A3380 (2016).
6. Fang, H. and Saad, Y. Two classes of multisecant methods for nonlinear acceleration. *Numerical Linear Algebra with Applications* **16**(3), 197–221 (2009).
7. Klawonn, A., Lanser, M., Rheinbach, O., and Uran, M. Nonlinear FETI-DP and BDDC Methods: A Unified Framework and Parallel Results. *SIAM Journal on Scientific Computing* **39**(6), C417–C451 (2017).
8. Kopaničáková, A., Kothari, H., and Krause, R. Nonlinear field-split preconditioners for solving monolithic phase-field models of brittle fracture. *Comput. Methods Appl. Mech. Engrg.* **403**, 115733 (2023).
9. Kothari, H., Kopaničáková, A., and Krause, R. A Multigrid Preconditioner for Jacobian-free Newton –Krylov Methods. In: *Domain Decomposition Methods in Science and Engineering XXVI*, 365–372 (2022).
10. Nash, S. A multigrid approach to discretized optimization problems. *Optimization Methods and Software* **14**(1), 99–116 (2000).
11. Nocedal, J. and Wright, S. *Numerical Optimization*. Springer (2000).
12. Zhang, J., O'Donoghue, B., and Boyd, S. Globally Convergent Type-I Anderson Acceleration for Nonsmooth Fixed-Point Iterations. *SIAM Journal on Optimization* **30**(4), 3170–3197 (2020).

Nonlinear Schwarz Preconditioning for Nonlinear Optimization Problems with Bound Constraints

Hardik Kothari, Alena Kopaničáková, and Rolf Krause

1 Introduction

We consider a Lipschitz domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, and a triangulation $\mathcal{T}$ on Ω. Now, we define $\mathcal{V} = \text{span}\{\phi_p\}_{p \in \mathcal{N}}$ as a Finite Element (FE) space, where $\mathcal{N}$ denotes a set of nodes of the mesh $\mathcal{T}$. Furthermore, we introduce the feasible set $\mathcal{F} = \{v \in \mathcal{V} \mid \underline{\psi} \leqslant v \leqslant \overline{\psi}\}$, where $\underline{\psi}, \overline{\psi}$ denote the component-wise lower bound and upper bound, respectively.

We consider the following abstract nonlinear minimization problem:

$$\text{Find } v^* = \arg\, \min_{v \in \mathcal{F}} f(v), \tag{1}$$

where $f \colon \mathcal{V} \to \mathbb{R}$ denotes a bounded, twice-Lipschitz-continuously-differentiable objective function. Problems of this type arise in numerous applications, such as contact mechanics [15], or fracture mechanics [12, 13].

Under certain assumptions on the function f, the minimization problem (1) can be equivalently rewritten as a nonlinear complementarity problem (NCP). As the first-order optimality conditions for (1) are given as: Find $v \in \mathcal{V}$ such that

$$\begin{aligned} &\nabla f(v) \leqslant 0, \quad v - \overline{\psi} \leqslant 0, \quad \langle \nabla f(v), v - \overline{\psi} \rangle = 0, \quad \forall v \in \mathcal{F} \setminus \underline{\mathcal{F}}, \\ &\nabla f(v) \geqslant 0, \quad \underline{\psi} - v \leqslant 0, \quad \langle \nabla f(v), \underline{\psi} - v \rangle = 0, \quad \forall v \in \mathcal{F} \setminus \overline{\mathcal{F}}, \end{aligned} \tag{2}$$

where $\underline{\mathcal{F}} = \{v \in \mathcal{F} \mid v = \underline{\psi}\}$ and $\overline{\mathcal{F}} = \{v \in \mathcal{F} \mid v = \overline{\psi}\}$ denote boundaries of the feasible set $\mathcal{F}$. Standard approaches for solving such minimization problems include penalty/augmented Lagrangian methods, interior-point methods, or active-set methods; see [16] for a detailed overview. In this work, we focus our attention on Newton-based active-set methods, namely the semismooth Newton method, and the

Hardik Kothari, Rolf Krause
Università della Svizzera italiana, Switzerland, e-mail: hardik.kothari@usi.ch,rolf.krause@usi.ch

Alena Kopaničáková
Brown University, USA, e-mail: alena.kopanicakova@brown.edu
Università della Svizzera italiana, Switzerland, e-mail: alena.kopanicakova@usi.ch

sequential quadratic programming (SQP) Newton method. Although the active-set methods are fairly efficient, their convergence tends to deteriorate due to three main factors: inability to detect an active-set sufficiently fast; strong and highly unbalanced nonlinearities; and ill-conditioning of the problem.

In the context of unconstrained nonlinear problems, nonlinear additive Schwarz preconditioners have been demonstrated to accelerate the convergence of the Newton methods; see for example Additive Schwarz Preconditioned Inexact Newton (ASPIN) [6], Restricted Additive Schwarz Preconditioned Exact Newton (RASPEN) [8], or Nonlinear Elimination Preconditioned Inexact Newton (NEPIN) [7] methods. In this work, we aim to extend a class of Schwarz preconditioned Newton methods to solve constrained nonlinear optimization problems. To the best of our knowledge, there have been only a few attempts to employ Schwarz methods to solve variational inequalities, for instance [1, 2, 3, 11]. In this work, we introduce a two-level nonlinear additive Schwarz preconditioner for the Newton-SQP method, which ensures that the subdomain and coarse-level corrections remain in the feasible set.

2 Nonlinear preconditioning

We define a residual function $F\colon \mathcal{V} \to \mathcal{V}'$ as the gradient of the original objective function, i.e., $F(\cdot) \equiv \nabla f(\cdot)$. Let G be a nonlinear preconditioner of the residual function F, such that in some sense G denotes an approximate inverse of the nonlinear function F, i.e., $G \approx F^{-1}$. Now, we can define nonlinearly-preconditioned residual function as $\mathscr{F}(v) := F(G(v))$. This preconditioner is used to define a nonlinearly-preconditioned variational inequality problem as follows: Find $v \in \mathcal{V}$ such that

$$\begin{aligned} &\mathscr{F}(v) \leqslant 0, \quad v - \overline{\psi} \leqslant 0, \quad \langle \mathscr{F}(v), v - \overline{\psi} \rangle = 0, \quad \forall v \in \mathcal{F} \setminus \underline{\mathcal{F}}, \\ &\mathscr{F}(v) \geqslant 0, \quad \underline{\psi} - v \leqslant 0, \quad \langle \mathscr{F}(v), \underline{\psi} - v \rangle = 0, \quad \forall v \in \mathcal{F} \setminus \overline{\mathcal{F}}, \end{aligned} \tag{3}$$

where the solution of (3) is the same as the solution of (1) and (2). Please note that the operator G is used as a "right-preconditioner", since this type of preconditioning does not change the original nonlinear system, and it also avoids the transformation of bound constraints into general inequality constraints. Generally, the preconditioner G can be constructed implicitly as a fixed-point iteration, i.e., $v^{(k+1)} = G(v^{(k)})$. In this work, we construct G using a variant of the nonlinear Restricted Additive Schwarz (NRAS) method, termed as NRAS-B method. Contrary to the standard NRAS method, the NRAS-B method ensures that the bound constraints are not violated by taking the preconditioning step. Thus, the preconditioner G produces an iterate that remains in the feasible set, i.e., $v^{(k)} \in \mathcal{F}$, for all $k = 1, 2, \ldots$

NRAS-B method: We consider a decomposition of the FE space $\mathcal{V}$ into n overlapping and non-overlapping subspaces, denoted by $\{\mathcal{V}_i\}_{i=1}^n$ and $\{\widetilde{\mathcal{V}}_i\}_{i=1}^n$, respectively. The overlap between the subspaces is controlled by the variable δ, defined as a multiple of the mesh-width h of the underlying mesh $\mathcal{T}$. On these subspaces, we

define the standard restriction operator $R_i \colon \mathcal{V} \to \mathcal{V}_i$, and the prolongation operator $P_i \colon \mathcal{V}_i \to \mathcal{V}$, where $R_i^\top = P_i$. Similarly, we define a restricted prolongation operator $\widetilde{P}_i \colon \widetilde{\mathcal{V}}_i \to \mathcal{V}$ such that $\sum_{i=1}^n \widetilde{P}_i R_i = I$.

Utilizing the aforementioned decomposition and the transfer operators, we now define the constrained nonlinear optimization problem on each subspace as follows. For a given initial guess $v_i^{(0)} \in \mathcal{F}_i$, where $v_i^{(0)} \hookleftarrow R_i v^{(k)}$:

$$\text{Find } v_i^* = \arg\, \min_{v_i \in \mathcal{F}_i} f_i(v_i), \tag{4}$$

where $f_i \colon \mathcal{V}_i \to \mathbb{R}$ denotes a restriction of the function f to the subspace $\mathcal{V}_i$. The feasible set associated with the subspace $\mathcal{V}_i$ is given by $\mathcal{F}_i = \{v_i \in \mathcal{V}_i \mid \underline{\psi}_i \leqslant v_i \leqslant \overline{\psi}_i\}$. Here, we point out that the local minimization problems (4) are solved on overlapping subspaces. However, the global iterate $v^{(k)}$ is updated using the corrections associated with the non-overlapping subspaces, as in

$$v^{(k+1)} = v^{(k)} + \alpha \sum_{i=1}^{n} \widetilde{P}_i (v_i^* - R_i v^{(k)}), \tag{5}$$

where v_i^* denotes a solution of (4) and α is computed using a line-search strategy. Solving (4) and update rule (5) comprise an iteration of the NRAS-B method.

Two-level NRAS-B method: The convergence of additive Schwarz methods is known to deteriorate with an increasing number of subdomains. In order to achieve algorithmic scalability, it is essential to ensure global information transfer through a coarse space. In the context of constrained minimization problems, constructing a coarse space is not a trivial task, as one has to ensure that the prolongated corrections from the coarse level provide a sufficient decrease in the objective function f, and the updated iterate remains in the feasible set. We construct a coarse-level objective function $f_0 \colon \mathcal{V}_0 \to \mathbb{R}$, where $\mathcal{V}_0$ denotes a coarse space $\mathcal{V}_0 \subset \mathcal{V}$, and $\mathcal{T}_0$ denotes a mesh associated with the FE space $\mathcal{V}_0$. The transfer of information between the coarse level and the original problem is ensured by the prolongation operator $P_0 \colon \mathcal{V}_0 \to \mathcal{V}$ and the restriction operator $R_0 \colon \mathcal{V}' \to \mathcal{V}_0'$, where $R_0 = P_0^\top$. Moreover, we also employ the projection operator $\mathit{\Pi}_0 \colon \mathcal{V} \to \mathcal{V}_0$ in order to transfer primal quantities to the coarse level.

In the context of nonlinear multilevel methods, several approaches for constructing the coarse-level feasible set $\mathcal{F}_0 = \{v_0 \in \mathcal{V}_0 \mid \underline{\psi}_0 \leqslant v_0 \leqslant \overline{\psi}_0\}$ are considered in the literature [9, 10, 14]. Here, we utilize constraint-projection rules from [9] and construct $\underline{\psi}_0, \overline{\psi}_0$ in a component-wise manner as

$$\begin{aligned} (\underline{\psi}_0)_t &= (v_0)_t + \max_{j \in \mathcal{N} \cap (\mathring{\omega}_0)_t} [(\underline{\psi} - v^{(k)})_j], \\ (\overline{\psi}_0)_t &= (v_0)_t + \min_{j \in \mathcal{N} \cap (\mathring{\omega}_0)_t} [(\overline{\psi} - v^{(k)})_j], \end{aligned} \tag{6}$$

where the symbol $(\cdot)_t$ denotes the value of a function associated with the t-th node of the mesh. The support of the basis function $(\phi_0)_t$ is denoted by $(\omega_0)_t$. Now, we

Algorithm 1: RASPN-B method

Data: $f : \mathcal{V} \to \mathbb{R}$, $\underline{\psi} \in \mathcal{V}$, $\overline{\psi} \in \mathcal{V}$, $v^{(0)} \in \mathcal{F}$, $k \leftarrow\!\shortmid 0$
Result: $v^{(k)}$

while $\|[\nabla f(v^{(k)})]_{\mathcal{F}}\| \geqslant \epsilon_{atol}$ **do**
- For given $v^{(k)}$, find $v^{(+)}$ by using a step of NRAS-B or TL-NRAS-B method
- Assemble gradient and Hessian: $g \leftarrow\!\shortmid \nabla f(v^{(+)})$, $H \leftarrow\!\shortmid \nabla^2 f(v^{(+)})$
- Find $s^{(k)}$ by solving the following constrained quadratic optimization problem
  $\min_{s^{(k)}} Q(s^{(k)}) := {}^1\!/_2\langle Hs^{(k)}, s^{(k)}\rangle + \langle g, s^{(k)}\rangle,$ s. t. $\underline{\psi} - v^{(+)} \leqslant s^{(k)} \leqslant \overline{\psi} - v^{(+)}$
- Find $\alpha^{(k)}$ using a line-search algorithm
- Update the iterate: $v^{(k+1)} \leftarrow\!\shortmid v^{(+)} + \alpha^{(k)} s^{(k)}$, $k \leftarrow\!\shortmid k+1$

can define the optimization problem on the coarse level as follows. For a given initial guess $v_0^{(0)} \in \mathcal{F}_0$, where $v_0^{(0)} \leftarrow\!\shortmid \mathit{\Pi}_0 v^{(k)}$:

$$\text{Find } v_0^* = \arg\min_{v_0 \in \mathcal{F}_0} \hat{f}_0(v_0). \tag{7}$$

Please note that minimization problem (7) is defined using an augmented coarse-level objective function $\hat{f}_0$, defined as

$$\hat{f}_0(v_0) = f_0(v_0) + \langle R_0 \nabla f(v^{(k)}) - \nabla f_0(\mathit{\Pi}_0 v^{(k)}), v_0\rangle, \tag{8}$$

where $v^{(k)}$ denotes the current iterate on the fine level. By adding the first-order consistency term to the objective function f_0, we ensure that the gradient of the augmented objective function $\hat{f}_0$ at the first iterate is the restricted fine-level gradient.

We follow an inverted V-cycle approach, where a coarse-level update step is followed by a single step of NRAS-B iteration, i.e., iterate $v^{(k)}$ is updated as follows:

$$\begin{aligned} v^{(k+1/2)} &= v^{(k)} + \hat{\alpha} P_0(v_0^* - \mathit{\Pi}_0 v^{(k)}), \\ v^{(k+1)} &= v^{(k+1/2)} + \alpha \sum_{i=1}^{n} \widetilde{P}_i(v_i^* - R_i v^{(k+1/2)}). \end{aligned} \tag{9}$$

The symbol v_0^* in (9) denotes the solution of the coarse-level minimization problem (7), while v_i^* is the solution of the subproblem (4) associated with the i-th subspace. The step sizes $\hat{\alpha}$ and α are again obtained using a line-search algorithm. Combining solutions of (7) and (4) with update rule (9), we can define an iteration of the TL-NRAS-B method.

Nonlinearly-preconditioned Newton SQP method: Finally, we provide a brief description of the nonlinearly-preconditioned Newton-SQP method for bound-constrained optimization problems. As summarized in Alg. 1, the method consists of two main phases. First, we invoke a step of the NRAS-B/TL-NRAS-B method in order to obtain an updated iterate $v^{(k)}$. Later, we construct a quadratic model Q, which is minimized subject to the bound constraints with the aim of obtaining a new

search direction $s^{(k)}$. In contrast to standard preconditioned Newton methods [5, 7], the minimization of the quadratic model is subjected to pointwise constraints, which ensures that the updated iterates remain in the feasible set $\mathcal{F}$. We note that the right-preconditioning can also be interpreted as a multiplicative or a composite solver [4].

3 Numerical experiments

In this section, we investigate the performance of nonlinear Schwarz preconditioners using two constrained minimization problems, namely the ignition and the minimal surface problems. Both numerical examples are defined on a domain $\Omega := (0,1)^2$ with boundary $\Gamma = \partial\Omega$, which is decomposed into four parts: $\Gamma_l = \{0\} \times [0,1]$, $\Gamma_r = \{1\} \times [0,1]$, $\Gamma_b = [0,1] \times \{0\}$ and $\Gamma_t = [0,1] \times \{1\}$. The discretization is performed using a mesh consisting of 120×120 uniform quadrilaterals which are further decomposed into triangular elements. In the case of two-level methods, we also employ a coarser mesh with 30×30 elements in each direction.

Ignition: We minimize a variant of the Bratu problem, given as:

$$\begin{aligned} &\min_{u \in H^1(\Omega)} \; f_I(u) := \frac{1}{2}\int_\Omega \|\nabla u\|^2 - (ue^u - e^u)\,dx - \int_\Omega f(x)u\,dx, \\ &\text{subject to } \; \underline{\psi}(x) \leqslant u \leqslant \overline{\psi}(x), \quad \text{a.e. in } \Omega, \qquad u = 0, \quad \text{on } \Gamma, \end{aligned} \tag{10}$$

where $f(x) = (9\pi^2 + e^{(x_1^2 - x_1^3)\sin(3\pi x_2)}(x_1^2 - x_1^3) + 6x_1 - 2)\sin(3\pi x_1)$. The bounds are given as $\underline{\psi}(x) = 0.2 - 8(x_1 - 7/16)^2 - 8(x_2 - 7/16)^2$ and $\overline{\psi}(x) = 0.5$.

Minimal Surface: This experiment aims to find the minimal surface described by a function u by solving the following minimization problem:

$$\begin{aligned} &\min_{u \in H^1(\Omega)} f_M(u) = \int_\Omega \sqrt{(1 + \|\nabla u\|^2)}\,dx, \\ &\text{subject to } \begin{cases} \underline{\psi}(x) \leqslant u \leqslant \overline{\psi}(x) \;\text{ a.e. in } \Omega. \\ u = -0.3\sin(2\pi x_2) \text{ on } \Gamma_l, \;\; u = 0.3\sin(2\pi x_2) \text{ on } \Gamma_r, \\ u = -0.3\sin(2\pi x_1) \text{ on } \Gamma_b, \;\; u = 0.3\sin(2\pi x_1) \text{ on } \Gamma_t, \end{cases} \end{aligned} \tag{11}$$

where the lower bound is prescribed as $\underline{\psi}(x) = 0.25 - 8(x_1 - 0.7)^2 - 8(x_2 - 0.7)^2$ and the upper bound is $\overline{\psi}(x) = 8(x_1 - 0.3)^2 - 8(x_2 - 0.3)^2 - 0.4$.

Setup of the solution strategies: For all numerical experiments, we prescribe the overlap $\delta = 3$ and obtain the decomposition into the subdomains using the library METIS. All considered solution methods terminate if $\|[\nabla f]_{\mathcal{F}}\| \leqslant 10^{-8}$, where $[\nabla f]_{\mathcal{F}} = \mathcal{P}_{\mathcal{F}}(x - \nabla f(x)) - x$ denotes the projected gradient. Here, the symbol $\mathcal{P}_{\mathcal{F}}$ denotes the projection onto the feasible set $\mathcal{F}$. Contrary to the traditional nonlinear RAS methods, the subdomain solvers, coarse-level solvers, and constrained quadratic minimization solvers are terminated using a fairly strict termination crite-

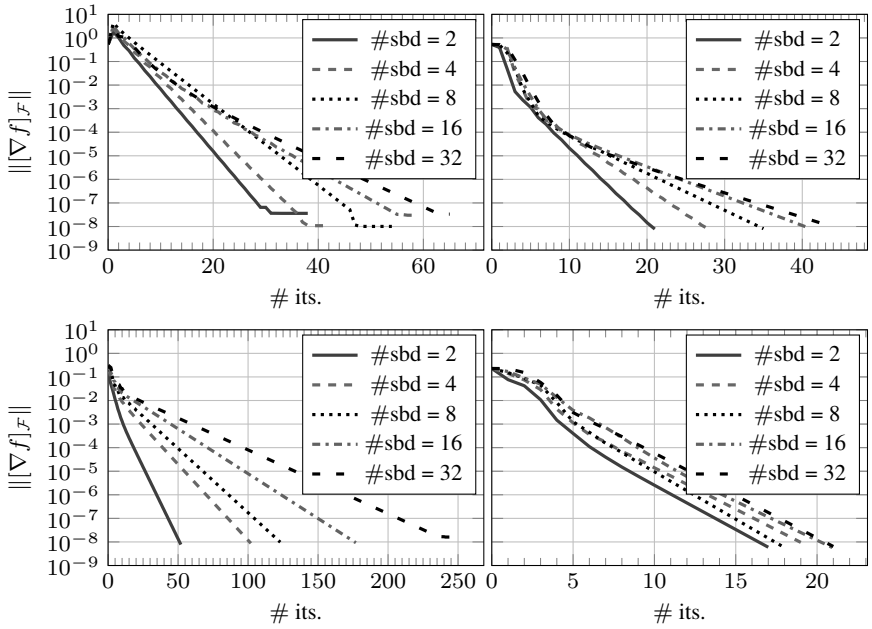


Fig. 1 Convergence history of NRAS-B (Left) and TL-NRAS-B (Right) methods for the ignition problem (Top) and the minimal surface (Bottom) problem. The experiments are performed with an increasing number of subdomains (sbd).

rion, i.e., they terminate if $\|[\nabla f]_{\mathcal{F}}\| \leqslant 10^{-11}$. Moreover, we employ a line-search method with the Armijo condition for computing the step size in all inner and outer solvers. The local and the coarse-level solvers for the NRAS-B/TL-NRAS-B methods employ the Newton-SQP method. On coarse levels, simply restriction of the objective function f_I and f_M is used to construct f_0.

Comparison between NRAS-B and TL-NRAS-B methods: The comparison is performed with respect to an increasing number of subdomains. As we can observe from Fig. 1, the standard NRAS-B method requires more iterations to satisfy the termination criterion than the TL-NRAS-B method. Due to the strict termination criterion, we notice that the NRAS-B method stagnates before reaching the termination criterion, while the TL-NRAS-B method manage to converge to the desired tolerance irrespective of number of subdomains. We also notice that the NRAS-B method requires more iterations with an increasing number of subdomains for both problems. For the TL-NRAS-B, we observe scalable convergence with respect to the number of subdomains for the minimal surface problem. However, for the ignition problem, the number of iterations grows with the number of subdomains. This can be attributed to the fact that the coarse grid is not able to represent the constraint function associated with this particular problem sufficiently well. Hence, the coarse-level nonlinear problems become over-constrained, which amounts to small coarse-grid corrections and insufficient global information transfer.

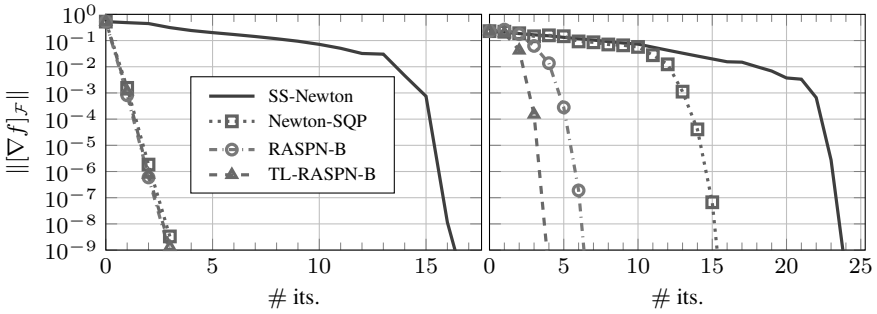


Fig. 2 Convergence history of semismooth Newton (SS-Newton), Newton-SQP, RASPN-B, and TL-RASPN-B methods for the ignition problem (Left) and minimal surface problem (Right).

Comparing RASPN-B method with other methods: In this section, we compare the performance of the NRAS-B and TL-NRAS-B preconditioned Newton methods with the semismooth Newton and Newton-SQP methods. In this study, the NRAS-B and TL-NRAS-B methods employ 16 subdomains. We note that the semismooth Newton method linearizes the nonlinearity of the problem and constraints simultaneously, while the Newton-SQP method first linearizes the nonlinearity of the problem and at each Newton iteration a QP problem is solved with constraints.

From Fig. 2, we can see that the Newton-SQP method preconditioned with NRAS-B and TL-NRAS-B method outperforms the semismooth Newton method for both examples. As the ignition problem is only mildly nonlinear, preconditioned Newton methods as well as the Newton-SQP method manage to satisfy the termination criterion in only 4 iterations. In the case of a minimal surface problem, which is more nonlinear, the benefit of preconditioning the Newton method is more evident. The RASPN-B and TL-RASPN-B methods converge in 7 and 4 iterations, respectively. In comparison, the Newton-SQP and semismooth-Newton methods require 16 and 24 iterations to converge, respectively.

4 Conclusion

In this work, we presented a nonlinear additive Schwarz preconditioning method for bound-constrained nonlinear optimization problems. The scalability of the method is enhanced by introducing a coarse level with the first-order consistent objective function and the constraints restricted from the fine level. The developed Schwarz methods are then employed as a right preconditioner for the Newton-SQP method. Our numerical results demonstrate that the proposed preconditioners enhance the convergence of the Newton-SQP method and outperform standard active-set Newton methods. We also show that the two-level preconditioner is algorithmically scalable if a coarse space captures the constraints from the fine level sufficiently well.

Acknowledgements H.K. and R.K. thank the Swiss National Science Foundation (SNSF) and the Deutsche Forschungsgemeinschaft for their support through the project SPP 1962 “Stress-Based Methods for Variational Inequalities in Solid Mechanics: Finite Element Discretization and Solution by Hierarchical Optimization" [186407]. A.K. gratefully acknowledges the support of the SNSF under the project "Multilevel training of DeepONets - multiscale and multiphysics applications" [206745]. Additionally, authors acknowledge the support of Platform for Advanced Scientific Computing through the project “FraNetG: Fracture Network Growth" and SNSF through the project ML2 - Multilevel and Domain Decomposition Methods for Machine Learning" [197041].

References

1. Badea, L. and Krause, R. One-and two-level Schwarz methods for variational inequalities of the second kind and their application to frictional contact. *Numerische Mathematik* **120**(4), 573–599 (2012).
2. Badea, L., Tai, X.-C., and Wang, J. Convergence rate analysis of a multiplicative Schwarz method for variational inequalities. *SIAM Journal on Numerical Analysis* **41**(3), 1052–1073 (2003).
3. Badea, L. and Wang, J. An additive Schwarz method for variational inequalities. *Mathematics of Computation* **69**(232), 1341–1354 (2000).
4. Brune, P. R., Knepley, M. G., Smith, B. F., and Tu, X. Composing scalable nonlinear algebraic solvers. *SIAM Review* **57**(4), 535–565 (2015).
5. Cai, X.-C. Nonlinear overlapping domain decomposition methods. In: *Domain Decomposition Methods in Science and Engineering XVIII*, 217–224. Springer (2009).
6. Cai, X.-C. and Keyes, D. E. Nonlinearly Preconditioned Inexact Newton Algorithms. *SIAM Journal on Scientific Computing* **24**(1), 183–200 (2002).
7. Cai, X.-C. and Li, X. Inexact Newton methods with restricted additive Schwarz based nonlinear elimination for problems with high local nonlinearity. *SIAM Journal on Scientific Computing* **33**(2), 746–762 (2011).
8. Dolean, V., Gander, M. J., Kheriji, W., Kwok, F., and Masson, R. Nonlinear Preconditioning: How to Use a Nonlinear Schwarz Method to Precondition Newton’s Method. *SIAM Journal on Scientific Computing* **38**(6), A3357–A3380 (2016).
9. Gelman, E. and Mandel, J. On multilevel iterative methods for optimization problems. *Mathematical Programming* **48**(1), 1–17 (1990).
10. Gratton, S., Mouffe, M., Toint, P. L., and Weber-Mendonça, M. A recursive ℓ_∞-trust-region method for bound-constrained nonlinear optimization. *IMA Journal of Numerical Analysis* **28**(4), 827–861 (2008).
11. Gross, C. and Krause, R. A new class of non–linear additively preconditioned trust–region strategies: Convergence results and applications to non-linear mechanics (2009). Available as INS Preprint No. 904.
12. Kopaničáková, A., Kothari, H., and Krause, R. Nonlinear field-split preconditioners for solving monolithic phase-field models of brittle fracture. *Computer Methods in Applied Mechanics and Engineering* **403**, 115733 (2023).
13. Kopaničáková, A. and Krause, R. A recursive multilevel trust region method with application to fully monolithic phase-field models of brittle fracture. *Computer Methods in Applied Mechanics and Engineering* **360**, 112720 (2020).
14. Kopaničáková, A. and Krause, R. A Multilevel Active-Set Trust-Region (MASTR) Method for Bound Constrained Minimization. In: *Domain Decomposition Methods in Science and Engineering XXVI*, 355–363. Springer (2022).
15. Kothari, H. and Krause, R. A generalized multigrid method for solving contact problems in lagrange multiplier based unfitted finite element method. *Computer Methods in Applied Mechanics and Engineering* **392**, 114630 (2022).
16. Nocedal, J. and Wright, S. *Numerical Optimization*. Springer (2000).

Domain Decomposition Solvers for Operators with Fractional Interface Perturbations

Miroslav Kuchta

1 Introduction

Mathematical models featuring interaction of physical systems across a common interface describe numerous phenomena in engineering, environmental sciences and medicine. Here the large variations in material coefficients or wide ranges of temporal/spatial scales at which the phenomena can be studied demand parameter-robust solution algorithms. In [3, 4] such algorithms were recently developed for Darcy-Stokes and Biot-Stokes models by establishing uniform stability of the respective problems in (non-standard) parameter-dependent norms. In particular, the authors show that in order to attain robustness, mass conservation at the interface Γ of the porous domain Ω must be accounted for in the functional setting, leading to control of the porous pressure p in the norm $\|p\|_\Omega$ such that

$$\|p\|_\Omega^2 = \|K^{1/2}\nabla p\|_{0,\Omega}^2 + \|\mu^{-1/2}p\|_{-1/2,\Gamma}^2. \tag{1}$$

Here $\|\cdot\|_{k,D}$ denotes the standard norm of Sobolev space $H^k(D)$ on domain D. The coefficients $K, \mu > 0$ are due to material properties, namely the permeability of the porous medium and the fluid viscosity.

By operator preconditioning, the choice of norm (1) yields a Riesz map preconditioner $b \mapsto x$ defined by solving the problem

$$-K\Delta_\Omega x + \mu^{-1}(-\Delta_\Gamma)^{-1/2}x = b. \tag{2}$$

Note that the operator in (2) contains a bulk part $-\Delta_\Omega$ and an interface part $(-\Delta_\Gamma)^{-1/2}$, which, from the point of view topological dimension of the underlying domains, can be viewed as a lower order *perturbation*.

Miroslav Kuchta
Simula Research Laboratory, Kristian Augusts gate 23, 0164 Oslo e-mail: miroslav@simula.no

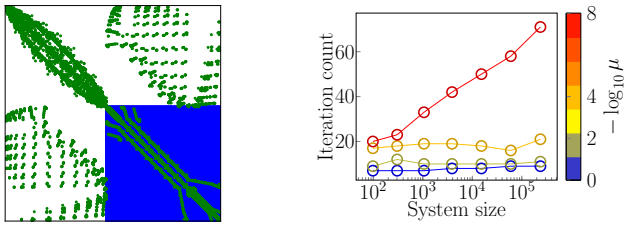


Fig. 1 (Left) Sparsity pattern of the operator in (2) on $\Omega = (0,1)^3$ with $\Gamma \subset \partial\Omega$. Interface perturbation leads to dense block (in blue) which is challenging for sparse LU solvers. (Right) Number of PCG iterations under mesh refinement when solving (2) on $\Omega = (0,1)^2$ with $\Gamma \subset \partial\Omega$ and AMG [8] preconditioner. In both case case $K = 1$ and the problems are discretized by continuous linear Lagrange ($\mathbb{P}_1$) elements.

Efficiency of the block-diagonal Darcy/Biot-Stokes preconditioners [3, 4] hinges on performant solvers for (2). However, the problem might not be amenable to standard (generic, black-box) approaches especially in case when the fractional interface perturbation becomes dominant. We illustrate this behavior in Figure 1 where (2) is solved by preconditioned conjugate gradient (PCG) method with algebraic multigrid (AMG) preconditioner. Indeed, the number of iterations increases with the weight of the perturbation term and, worryingly, for large enough values mesh-independence is lost.

Non-overlapping domain decomposition (DD) is a solution methodology which has been successfully applied to number of challenging problems including coupled multiphysics systems e.g. [5, 7, 11]. A key component of the method are then the algorithms for the problems arising at the interface which can be broadly divided into two categories. In FETI or BDDC variants (see e.g. [1] and references therein) the solvers utilize suitable auxiliary problems on the *subdomains*. To develop tailored solvers for operators with fractional interface perturbation we here follow an alternative approach [1] and address the problem directly at the *interface*. In particular, we shall construct preconditioners for the resulting Steklov-Poincaré operators using sums of fractional order interfacial operators which include contribution due to the DD and the perturbation (which is only localized at the interface).

2 Domain decomposition solvers

We shall consider solvers for (2) in a more general setting. To this end, let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ be a bounded domain with Lipschitz boundary $\partial\Omega$, and $\Gamma \subseteq \partial\Omega$. Moreover, let $V = V(\Omega), Q = Q(\Gamma)$ be a pair of Hilbert spaces with V', Q' being their respective duals and let $R\colon V \to Q'$ be a restriction operator. For $b \in V'$ we are then interested in solving

$$\mathcal{A}x = b \text{ in } V' \quad \text{with} \quad \mathcal{A} = A_\Omega + \gamma R' B_\Gamma^{-1} R, \tag{3}$$

where $\gamma \geq 0$ and $A_\Omega \colon V \to V'$ is some symmetric operator coercive on V while $B_\Gamma \colon Q \to Q'$ is assumed to induce an inner product on Q. Note that the norm operator in (2) is a special case of (3) with $V = H_0^1(\Omega)$, $Q = H^{1/2}(\Gamma)$, R the trace operator and $A_\Omega = -K\Delta_\Omega$ while $B_\Gamma = (-\Delta_\Gamma)^{1/2}$.

To formulate our non-overlapping domain decomposition approach for (3), we follow [1] and decompose $V = V_0 \oplus V_\Gamma$ where $V_0 = \{v \in V; Rv = 0\}$. Assuming that V_Γ can be identified with Q we observe that the operator $\mathcal{A}$ takes a block structure

$$\mathcal{A} = \left(\begin{array}{c|c} A_\Omega^{00} & A_\Omega^{0i} \\ \hline A_\Omega^{i0} & A_\Omega^{ii} \end{array}\right) + \gamma \left(\begin{array}{c|c} 0 & 0 \\ \hline 0 & \tilde{B}_\Gamma^{-1} \end{array}\right), \tag{4}$$

which we exploit to design a preconditioner for $\mathcal{A}$. Specifically, under the assumption that A_Ω^{00} is invertible, let us define the DD preconditioner

$$\mathcal{B} = \left(\begin{array}{c|c} I_\Omega^{00} & -(A_\Omega^{00})^{-1} A_\Omega^{0i} \\ \hline 0 & I_\Omega^{ii} \end{array}\right) \left(\begin{array}{c|c} A_\Omega^{00} & 0 \\ \hline 0 & S_\Gamma \end{array}\right)^{-1} \left(\begin{array}{c|c} I_\Omega^{00} & 0 \\ \hline -A_\Omega^{i0}(A_\Omega^{00})^{-1} & I_\Omega^{ii} \end{array}\right). \tag{5}$$

Here $I_\Omega^{00} \colon V_0 \to V_0$ and $I_\Omega^{ii} \colon V_\Gamma \to V_\Gamma$ are identity operators on the respective subspaces while S_Γ is spectrally equivalent to the DD Schur complement/Steklov-Poincaré operator $S_\Gamma^* = A_\Omega^{i0}(A_\Omega^{00})^{-1} A_\Omega^{0i} + A_\Omega^{ii} + \gamma \tilde{B}_\Gamma^{-1}$. We note that preconditioner (5) preserves symmetry of the original problem (3) as we target PCG solvers. However, with Krylov methods which do not require symmetry a more efficient triangular variant of the preconditioner is sufficient.

Our main contribution is an observation that for problems with interface perturbations, the Schur complement approximation S_Γ in (5) takes the form

$$S_\Gamma = L_A^s + \gamma L_B^t, \tag{6}$$

for some constants $s, t \in \mathbb{R}$ and symmetric, positive-definite operators L_A, L_B depending on the regularity of $\mathcal{A}$, the restriction operator and the perturbation. In particular, the structure of the preconditioner reflects the two contributions to the Schur complement; the decomposition $V = V_0 \oplus V_\Gamma$ applied to operator A_Ω yields L_A^s while L_B^t is due to the perturbation.

Motivated by the initial example (2) we shall in the following focus on problems for which $-1 < s, t < 1$ and L_A, L_B are spectrally equivalent to $L = -\Delta_\Gamma + I_\Gamma$. However, we highlight that the operators might in general differ by their boundary conditions (which for L_A reflect boundary conditions on $\partial\Omega \setminus \Gamma$ imposed on V in (3)).

Assuming that $(A_\Omega^{00})^{-1}$ can be efficiently computed, the main challenge for scalability of preconditioner (5) is an efficient realization of (an approximate) inverse of (6). Upon discretization, the operators L_A^s, L_B^t can be approximated by eigenvalue

factorization[1]. However, this approach suffers from cubic scaling. For the specific case of $L^{1/2}$ a more efficient strategy with improved scaling is applied in [1] based on the Lanczos process while, more recently, [9] proves that rational approximations (RA) lead to non-overlapping DD methods with linear scaling. Building on this observation to obtain order optimal solvers for the perturbed problem (3) we follow [6] where rational approximations[2] were developed for Riesz maps with norms induced by sum operator $\alpha L^s + \beta L^t$ with $\alpha, \beta \geq 0$. In particular, this setting fits our Schur complement operator (6) if constant material properties *and* suitable boundary conditions are prescribed on $\mathcal{A}$ in (3).

3 Model problem

We shall illustrate performance of the domain decomposition preconditioner (5) using a model interface-perturbed problem: Find $x \in V = H^1(\Omega)$ such that

$$K(-\Delta_\Omega + I_\Omega)x + \gamma(-\Delta_\Gamma + I_\Gamma)^t x = b \text{ in } V', \tag{9}$$

where $K > 0$, $\gamma \geq 0$ and $-1 < t < 1$. Here $\Omega = (0,1)^d$, $d = 2, 3$ and $\Gamma = \partial\Omega$. We note that this choice maximizes the size of the interface. At the same times, it enables the RA-favorable setting of $L_A = L$, $L_B = L$, $L = -\Delta_\Gamma + I_\Gamma$ in (6). Following [1] the DD Schur complement of the operator $A_\Omega = K(-\Delta_\Omega + I_\Omega)$ in (3) is spectrally equivalent to fractional operator $KL^{1/2}$. In turn we apply preconditioner (5) with $S_\Gamma = KL^{1/2} + \gamma L^t$. However, for simplicity, we shall fix K, here $K = 3$, and we only investigate the effect of perturbation strength.

In the numerical experiments we consider H^1-conforming finite element spaces $V_h \subset V$ constructed in terms of $\mathbb{P}_1$ elements. Consequently, the matrix realization

[1] For $L\colon Q \to Q'$ let L_h be the matrix realization of the operator in the basis of some finite dimensional approximation space Q_h, $n = \dim Q_h$. Moreover, let M_h be the mass matrix, i.e. matrix realization of the inner product of the Lebesgue space L^2 on Q_h. Assuming L is symmetric and positive definite, the factorization $L_h U_h = M_h U_h \Lambda_h$, $U_h^T M_h U_h = \mathrm{Id}$ holds where Λ_h is a diagonal matrix of eigenvalues while the corresponding eigenvectors constitute the columns of matrix U_h. We then define

$$L_h^s = (M_h U_h)\Lambda_h^s (M_h U_h)^T. \tag{7}$$

Note that for $L = (-\Delta + I)$ and $f \in Q$ represented in Q_h by interpolant with coefficient vector $f_h \in \mathbb{R}^n$ the function $f_h \mapsto f_h \cdot L_h^s f_h$ represents an approximation of the square of the Sobolev norm $\|f\|_s^2$.

[2] Referring to the definitions in (7) the RA construct approximate solutions $x \in Q$ satisfying $\alpha L^s x + \beta L^t x = b$, $b \in Q'$ in the finite dimensional space Q_h via a solution operator

$$c_0 M_h^{-1} + \sum_{k=1}^{m} c_i (L_h + p_k M_h)^{-1}. \tag{8}$$

Here, $c_i \in \mathbb{R}$ and $p_i \geq 0$ are respectively the residues and the poles of the rational approximation f_{RA} to function $f : x \to (\alpha x^s + \beta x^t)^{-1}$. Importantly, the number of poles m does not depend on the dimensionality of Q_h and is instead determined by the accuracy ϵ_{RA} of the RA, i.e. $\|f - f_{\mathrm{RA}}\| \leq \epsilon_{\mathrm{RA}}$. We refer to [6, 9] and references therein for more details.

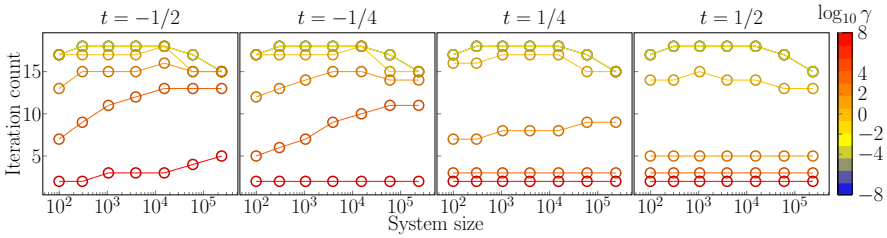


Fig. 2 PCG iterations when solving (9) on $\Omega = (0,1)^2$ and preconditioner (5) with $S_\Gamma = KL^{1/2} + \gamma L^t$. Problem is discretized by $\mathbb{P}_1$ elements. Blocks of the preconditioner are here computed exactly.

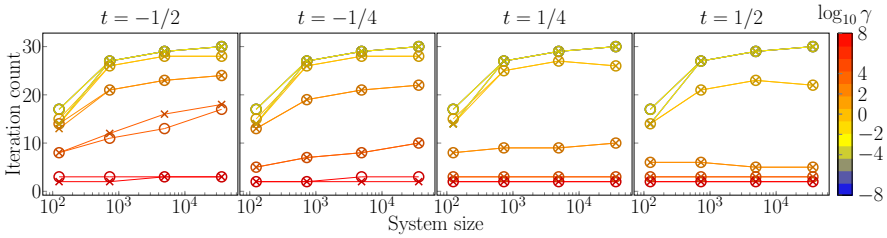


Fig. 3 PCG iterations when solving (9) on $\Omega = (0,1)^3$ and preconditioner (5) with $S_\Gamma = KL^{1/2} + \gamma L^t$. Problem is discretized by $\mathbb{P}_1$ elements. Leading block of preconditioner is computed exactly. Results with realization of the Schur complement preconditioner by RA with tolerance $\epsilon_{\text{RA}} = 10^{-14}$ are depicted by (○) markers while (×) markers correspond to definition via the eigenvalue problem (7).

of the fractional interface perturbation reads $\gamma T_h^T L_h^t T_h$ where matrix L_h^t is defined in (7) and T_h is a discrete trace operator such that $T_h\phi = \sum_{j=1}^n l_j(\phi|_\Gamma)\psi_j$ for any $\phi \in V_h$ and ψ_j, l_j, $j = 1, \cdots, n$ being respectively the basis functions and degrees of freedom (point evaluations) of the discrete trace space $V_{\Gamma,h} = Q_h$ built likewise using $\mathbb{P}_1$ elements.

The linear systems due to discretization of (9) shall be solved by PCG solver using our DD preconditioner (5) which now requires inverse of the linear system due to $KL_h^{1/2} + \gamma L_h^t$. Here we shall either apply the eigenvalue realization (7) (which allows for closed form evaluation of the exact inverse) or the approximate inverse due to RA, see (8). To put focus on the Schur complement action of blocks $(A_\Omega^{00})^{-1}$ in (5), that is, in the diagonal and triangular factors of $\mathcal{B}$, will be computed (exactly) by LU factorization. For results with approximate inverse of A_Ω^{00} we refer to Remark 1. Finally, the PCG solver is always started from 0 initial vector and terminates upon reducing the preconditioned residual norm by factor 10^{10}.

We summarize performance of the DD preconditioner in Figure 2 and Figure 3 which consider (9) with $\Omega = (0,1)^2$ and $\Omega = (0,1)^3$ respectively. It can be seen that the PCG convergence is in general bounded in mesh size, fractionality t and the perturbation strength γ. Important for the scalability of (5) is the observation that iteration counts with RA realization of the Schur complement preconditioner

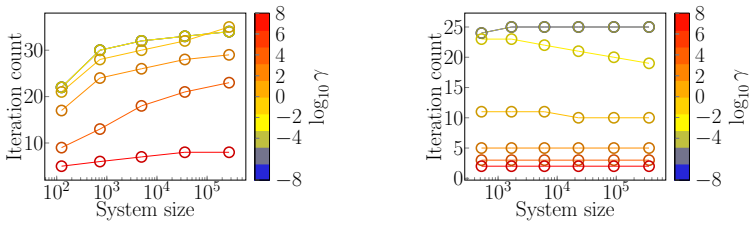


Fig. 4 PCG iterations for computing the inverse of fractional perturbed operators by preconditioner (5). (Left) The operator is (9) with $t = -1/2$ and $\Omega = (0,1)^3$. The operator $\mathcal{A}$ (4) and the preconditioner are evaluated using RA. On the finest refinement level $\dim V_{\Gamma,h} = 24 \cdot 10^3$. (Right) Operator (10) is considered with $S_\Gamma = KL^{-1/2} + \gamma I_\Gamma$ in the Schur complement (6). In both cases $\Gamma = \partial\Omega$.

practically match the exact inverse of S_Γ. We remark that the chosen tolerance of $\epsilon_{\text{RA}} = 10^{-14}$ yields roughly $m = 20$ poles in (8). The computation setup in $3d$ then leads to linear systems with < 6200 unknowns at the interface.

Remark 1 (Evaluation of the operator in (9)*)*

In numerical experiments shown in Figure 2 and Figure 3 the operator $\mathcal{A}$ in (9) utilized the eigenvalue decomposition (7) for L_h^t. This realization restricts the size of Γ or $\dim Q_h$ that are computationally tractable. However, action of the perturbation can instead be computed via RA leading to evaluation of $\mathcal{A}$ with optimal complexity and enabling large scale problems. In Figure 4 we revisit (9) with $t = -1/2$, $\Omega = (0,1)^3$ and RA used both for the operator and the preconditioner (5). Moreover, to illustrate performance when the preconditioner blocks are inexact, all instances of $(A_\Omega^{00})^{-1}$ shall here, for simplicity, be approximated by a single V-cycle of AMG [8]. The number of PCG iterations then appears to be bounded in the mesh size and the parameter γ. As before, $\mathbb{P}_1$ elements were used for discretization.

Remark 2 (Application to $\boldsymbol{H}$*(div)-elliptic problem)* Preconditioners (5) are not limited to H^1-elliptic problems. To illustrate this fact we consider $V = \boldsymbol{H}(\text{div}, \Omega)$, $\Omega = (0,1)^2$ and a variational problem induced by bilinear form due to operator $\mathcal{A}$

$$\langle \mathcal{A}\boldsymbol{u}, \boldsymbol{v}\rangle = \int_\Omega K(\boldsymbol{u}\cdot\boldsymbol{v} + \nabla\cdot\boldsymbol{u}\nabla\cdot\boldsymbol{v}) + \gamma \int_{\partial\Omega} \boldsymbol{u}\cdot\boldsymbol{\nu}\boldsymbol{v}\cdot\boldsymbol{\nu} \quad \forall \boldsymbol{u}, \boldsymbol{v} \in V. \tag{10}$$

We observe that $\mathcal{A}$ falls under the template problem (3). In order to apply the domain-decomposition preconditioner we then require a preconditioner for the DD Schur complement due to $(A_\Omega^{00})^{-1}$ where A_Ω^{00} is here the operator $K(I - \nabla\nabla\cdot)$ on $\boldsymbol{H}_0(\text{div}, \Omega)$. Motivated by [2], we shall to this end consider the operator $L_A = KL^{-1/2}$ so that S_Γ in (5) is defined as $S_\Gamma = KL^{-1/2} + \gamma I_\Gamma$. For numerical experiments the system is discretized by lowest order Brezzi-Douglas-Marini elements which lead to the discrete trace space $V_{\Gamma,h} = Q_h$ of discontinuous piecewise-linear functions on trace mesh Γ_h. Robustness of the resulting preconditioner is shown in Figure 4.

4 Darcy-Stokes preconditioning

We finally apply the proposed non-overlapping DD solvers to realize preconditioners for the coupled Darcy-Stokes model with Darcy problem in the primal form [7]. That is, assuming bounded domains $\Omega_S, \Omega_D \subset \mathbb{R}^d$, $d = 2, 3$ sharing a common interface Γ (cf. Figure 5) we seek to find the Stokes velocity $\boldsymbol{u}_S$, the Stokes pressure p_S and the Darcy pressure p_D such that

$$\begin{aligned} -\nabla \cdot \sigma(\boldsymbol{u}_S, p_S) = \boldsymbol{f}_S \text{ and } \nabla \cdot \boldsymbol{u}_S = 0 \qquad & \text{in } \Omega_S, \\ -\nabla \cdot K \nabla p_D = f_D \qquad & \text{in } \Omega_D, \\ \boldsymbol{u}_S \cdot \boldsymbol{\nu} + K \nabla p_D \cdot \boldsymbol{\nu} = 0 \qquad & \text{on } \Gamma, \\ -\boldsymbol{\nu} \cdot \sigma(\boldsymbol{u}_S, p_s) \cdot \boldsymbol{\nu} - p_D = 0 \qquad & \text{on } \Gamma, \\ -P_{\boldsymbol{\nu}} \left(\sigma(\boldsymbol{u}_S, p_S) \cdot \boldsymbol{\nu}_S\right) - \alpha \mu K^{-1/2} P_{\boldsymbol{\nu}} \boldsymbol{u}_S = 0 \qquad & \text{on } \Gamma, \end{aligned} \tag{11}$$

where $P_{\boldsymbol{\nu}}$ is the tangential trace operator $P_{\boldsymbol{\nu}} \boldsymbol{u} = \boldsymbol{u} - (\boldsymbol{u} \cdot \boldsymbol{\nu}) \boldsymbol{\nu}$ and $\sigma(\boldsymbol{u}, p) = \mu \nabla \boldsymbol{u} - p\mathrm{Id}$. In addition to the previously introduced coefficients K, $\mu > 0$ the model also includes the Beavers-Joseph-Saffman parameter $\alpha \geq 0$. The system (11) is closed by prescribing suitable boundary conditions to be discussed shortly.

We consider (11) with a parameter-robust block diagonal preconditioner [4]

$$\mathcal{B} = \mathrm{diag}\left(-\mu\Delta + \alpha\mu K^{-1/2} P_{\boldsymbol{\nu}}' P_{\boldsymbol{\nu}}, \mu^{-1} I, -K\Delta + \mu^{-1}(-\Delta_\Gamma)^{-1/2}\right)^{-1}. \tag{12}$$

Observe that both the first and the final block in (12) are of the form of the interface-perturbed operators (3). However, for simplicity we shall here set $\alpha = 0$ and only focus on the pressure preconditioner. In particular, to efficiently approximate (2) we shall perform few PCG iterations with the DD preconditioner (5) using $S_\Gamma = KL^{1/2} + \mu^{-1} L^{-1/2}$ in the Schur complement. We note that the interface operator is thus identical to the one utilized in robust preconditioning of mixed Darcy-Stokes model [10].

To illustrate performance of the preconditioner (12) we consider (11) in a $3d$ domain pictured in Figure 5 and set[3] $K = 10^{-2}$, $\mu = 10^{-4}$. Using discretization by $\mathbb{P}_2$-$\mathbb{P}_1$-$\mathbb{P}_2$ elements the linear system is solved by preconditioned Flexible GMRes (FGMRes). The Darcy-Stokes preconditioner is then realized by applying single AMG V-cycle for the Stokes blocks while the Riesz map of the Darcy pressure (2) is approximated by PCG solver using (5) and running with a relative tolerance of 10^{-4}. The DD preconditioner uses RA with tolerance $\epsilon_{\mathrm{RA}} = 10^{-14}$ and AMG for the leading block in (5). With this setup the scalability study summarized in Figure 5 reveals that the proposed solver is order optimal.

[3] Due to computational demands we did not perform parameter-robustness study for $d = 3$. However, with a $2d$ version of the geometry in Figure 5 we observe that (5) with RA approximation of the Schur complement leads to mesh- and parameter-independent Krylov iterations. In particular, $S_\Gamma = KL^{1/2} + \mu^{-1} L^{-1/2}$ leads to K, μ and h bounded iterations when solving (2) with PCG. We omit these results from our presentation due to spatial limitations.

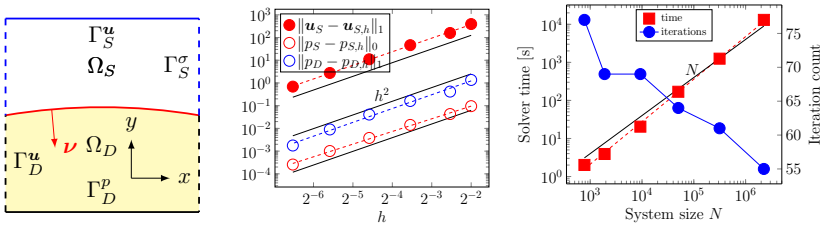


Fig. 5 (Left) Computation domain is obtained by extrusion of the pictured geometry. The interface Γ, being part of a circle arc, is curved. No-slip and traction conditions are prescribed on $\Gamma_S^{\boldsymbol{u}}$ and Γ_S^{σ} respectively. Darcy pressure is prescribed on Γ_D^{p}. (Center) Error convergence study performed using the $3d$ setup. With $\mathbb{P}_2$-$\mathbb{P}_1$-$\mathbb{P}_2$ elements, optimal quadratic rates are observed in all the variables. (Right) Solver time (including preconditioner setup and FGMRes runtime) scales linearly with the problem size.

Acknowledgements The author is grateful to prof. Ludmil T. Zikatanov (Penn State) and prof. Kent-André Mardal (University of Oslo) for stimulating discussions on non-overlapping domain decomposition which inspired the presented approach. This work received support from the Norwegian Research Council grant 303362.

References

1. Arioli, M., Kourounis, D., and Loghin, D. Discrete fractional Sobolev norms for domain decomposition preconditioning. *IMA Journal of Numerical Analysis* **33**(1), 318–342 (2013).
2. Babuška, I. and Gatica, G. N. On the mixed finite element method with Lagrange multipliers. *Numerical Methods for Partial Differential Equations: An International Journal* **19**(2), 192–210 (2003).
3. Boon, W. M., Hornkjøl, M., Kuchta, M., Mardal, K.-A., and Ruiz-Baier, R. Parameter-robust methods for the Biot–Stokes interfacial coupling without Lagrange multipliers. *Journal of Computational Physics* **467**, 111464 (2022).
4. Boon, W. M., Koch, T., Kuchta, M., and Mardal, K.-A. Robust monolithic solvers for the Stokes–Darcy problem with the Darcy equation in primal form. *SIAM Journal on Scientific Computing* **44**(4), B1148–B1174 (2022).
5. Boon, W. M. A parameter-robust iterative method for Stokes-Darcy problems retaining local mass conservation. *ESAIM: M2AN* **54**(6), 2045–2067 (2020).
6. Budisa, A., Hu, X., Kuchta, M., Mardal, K.-A., and Zikatanov, L. Rational approximation preconditioners for multiphysics problems. *arXiv preprint arXiv:2209.11659* (2022).
7. Discacciati, M. and Gerardo-Giorda, L. Optimized Schwarz methods for the Stokes–Darcy coupling. *IMA Journal of Numerical Analysis* **38**(4), 1959–1983 (2018).
8. Falgout, R. D., Jones, J. E., and Yang, U. M. Pursuing scalability for Hypre's conceptual interfaces. *ACM Trans. Math. Softw.* **31**(3), 326–350 (2005).
9. Harizanov, S., Lirkov, I., and Margenov, S. Rational approximations in robust preconditioning of multiphysics problems. *Mathematics* **10**(5), 780 (2022).
10. Holter, K. E., Kuchta, M., and Mardal, K.-A. Robust preconditioning of monolithically coupled multiphysics problems. *arXiv preprint arXiv:2001.05527* (2020).
11. Vassilev, D., Wang, C., and Yotov, I. Domain decomposition for coupled Stokes and Darcy flows. *Computer Methods in Applied Mechanics and Engineering* **268**, 264–283 (2014).

Optimized Schwarz Methods for Isogeometric Analysis

Lahcen Laayouni, Ahmed Ratnani, and Abdessadek Rifqui

1 Introduction

Isogeometric Analysis (IGA) is a novel computational technique for solving partial differential equations (PDEs) first introduced by Hughes et al, see [6]. It integrates computer-aided design (CAD) and simulation. In IGA, a geometric model created within a CAD environment is used as the basis for analysis, and B-splines or non-uniform rational B-splines (NURBS) are employed as basis functions. IGA offers a new type of refinement strategy, in addition to the traditional mesh refinement (h-refinement) and p-refinement in Finite Element Analysis (FEA), namely k-refinement, which allows for changing the smoothness of the basis functions. The aim of IGA is to improve the accuracy and efficiency of simulation by using CAD models directly in the analysis process. In Section 2.1 we give a brief description of the B-spline functions. For an extensive overview on the approximation theory based on IGA, see [2].

Domain decomposition methods (DDM) are based on dividing the domain into subdomains which leads to solve small local problems. The classical Schwarz methods use Dirichlet boundary conditions at the artificial interfaces, see [8], while the Optimized Schwarz Methods (OSM) use Robin ($\partial_n u + \lambda u$) or higher order boundary conditions at the artificial interfaces. The challenge is to find the optimal value of the parameter λ, this latter can be solved by virtue of Fourier transform, see [4] for more details. Rather than relying on the existing literature on DDM for IGA as described in [3], we adopt an approach that enforces C^{-1} smoothness of the B-spline in the interface condition. For a more comprehensive understanding of it, please

Abdessadek Rifqui, Ahmed Ratnani
MSDA-Modeling Simulation and Data Analysis, Mohammed VI Polytechnic University, e-mail: abdessadek.rifqui@um6p.ma, ahmed.ratnani@um6p.ma

Lahcen Laayouni
School of Science and Engineering, Al Akhawayn University, Avenue Hassan II, 53000 P.O. Box 1630, Ifrane, Morocco e-mail: L.Laayouni@aui.ma

refer to [1]. For our analysis, we consider Algebraic Optimized Schwarz methods (AOSM) which mimic OSM algebraically.

Our approach involves combining IGA and AOSM to solve partial differential equations with complex geometries. The efficiency of the resulting algorithm is due to the robustness of AOSM/OSM and the flexibility of IGA.

2 IGA analysis and algebraic optimized Schwarz methods

For our analysis we need to introduce B-spline and algebraic optimized Schwarz methods.

2.1 B-spline based IGA

Let m and p be two positive integers, and Ξ be a set of non-decreasing real numbers such that $\xi_1 \leq \xi_2 \leq \ldots \leq \xi_{m+p+1}$. The ξ_j's are called the *knots*, the set Ξ is the knot vector, and the interval $[\xi_j, \xi_{j+1})$ is the j-th knot span. Note that if ξ_j is repeated $k > 1$ times in the knot vector (i.e. $\xi_j = \xi_{j+1} = \ldots = \xi_{j+k-1}$), ξ_j is a multiple knot of multiplicity k with no corresponding knot span; otherwise, it is a simple knot if ξ_j appears only once (or $k = 1$). A knot vector is said to be *uniform* if its knots are uniformly spaced; otherwise, it is called a *nonuniform* knot vector. A knot vector is considered to be *open* if its first and last knots have multiplicity $p + 1$. The interval (ξ_1, ξ_{m+p+1}) is called the *patch*. The maximum multiplicity allowed is $p + 1$.

Once a knot vector is available, the B-spline basis functions can be defined recursively, beginning with the first order, $p = 0$ (piecewise constant)

$$N_j^0(\xi) := \chi_{[\xi_j, \xi_{j+1})} = \begin{cases} 1, & \text{if } \xi_j \leq \xi < \xi_{j+1}, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

For $p \geq 1$,

$$N_j^p(\xi) := \begin{cases} \dfrac{\xi - \xi_j}{\xi_{j+p} - \xi_j} N_j^{p-1}(\xi) + \dfrac{\xi_{j+p+1} - \xi}{\xi_{j+p+1} - \xi_{j+1}} N_{j+1}^{p-1}(\xi), & \text{if } \xi_j \leq \xi < \xi_{j+p+1}, \\ 0, \quad \text{otherwise.} \end{cases} \tag{2}$$

we adopt the convention $\dfrac{0}{0} = 0$ in (2).

According to (2), all B-spline functions are to be (i) non-negative, (ii) have a local support in $[\xi_j, \xi_{j+p+1}]$ (compact support) for all $j = 1, \ldots, m$, (iii) form a *partition of unity*, and (iv) be linear independent, as shown in [9]. The basis functions of order p, in general, have $p - k$ continuous derivatives $\mathscr{C}^{p-k}$ across knot ξ_j. When the multiplicity of a knot value is exactly p, the basis at that knot is interpolatory. If the multiplicity of a basis is $p + 1$, it can result the basis become discontinuous in the $\mathscr{C}^{-1}$ space. In Figure 1, we present an example of cubic basis functions generated by $p = 3$ from the uniform open knot vector $\Xi = \{0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 6, 6, 6\}$.

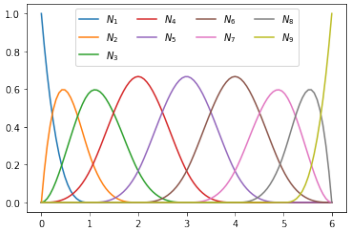


Fig. 1 Cubic basis functions formed from $\Xi = \{0, 0, 0, 0, 1, 2, 3, 4, 5, 6, 6, 6, 6\}$.

2.2 Algebraic optimized Schwarz methods

Descritizing PDEs using IGA analysis leads to solve linear systems of the form

$$Au = f, \tag{3}$$

where A is a block banded matrix of size $n \times n$ given by

$$A = \begin{bmatrix} A_{11} & A_{12} & & \\ A_{21} & A_{22} & A_{23} & \\ & A_{32} & A_{33} & A_{34} \\ & & A_{43} & A_{44} \end{bmatrix}, \tag{4}$$

where A_{ij} are blocks of size $n_i \times n_j$, $i, j = 1, \ldots, 4$, and $n = \sum_i n_i$. For a two-subdomain decomposition with overlap we have $n_1 \gg n_2$ and $n_4 \gg n_3$. To illustrate this decomposition let us solve the Poison equation in $\Omega = \mathbb{R} \times (0, 1)$ with homogeneous Dirichlet at the boundary conditions. We discretize the continuous operator on a grid with an interval of size h in both the x and y directions and we assume that $h = 1/(N+1)$ so that there are N degrees of freedom in y-direction. For instance the stiffness matrix obtained when we discretize with the finite element method using piecewise linear functions, and using the subdomains $\Omega_1 = (-\infty, h) \times (0, 1)$ and $\Omega_2 = (0, +\infty) \times (0, 1)$, leading to the decomposition

$$A = \left[\begin{array}{c|c|c|c} A_{11} & A_{12} & O & O \\ \hline A_{21} & A_{22} & A_{23} & O \\ \hline O & A_{32} & A_{33} & A_{34} \\ \hline O & O & A_{43} & A_{44} \end{array}\right] = \left[\begin{array}{ccc|c|c|c} \ddots & \ddots & \ddots & & & \\ & -I & J & -I & & \\ \hline & & -I & J & -I & \\ \hline & & & -I & J & -I \\ \hline & & & & -I & J \quad -I \\ & & & & & \ddots \quad \ddots \quad \ddots \end{array}\right], \tag{5}$$

where I is the $N \times N$ identity matrix and J is the $N \times N$ tridiagonal $J = \text{tridiag}(-1, 4, -1)$. We have in this case $n_2 = n_3 = N$. The Algebraic Optimized Schwarz methods are iterative methods [5, Section 2, page 4], and the optimized restricted additive and multiplicative Schwarz methods are defined by

$$T_{\text{ORAS}} = I - \sum_{i=1}^{2} \tilde{R}_i^T \tilde{A}_i^{-1} R_i A, \quad \text{and} \quad T_{\text{ORMS}} = \prod_{i=2}^{1} (I - \tilde{R}_i^T \tilde{A}_i^{-1} R_i A), \tag{6}$$

where the restriction operators with overlap are $R_1 = [I\ O]$ and $R_2 = [O\ I]$, of size $(n_1 + n_2 + n_3) \times n$ and $(n_2 + n_3 + n_4) \times n$ respectively, using the prolongations $\tilde{R}_i^T$ without the overlap, which are defined as

$$\tilde{R}_1 = \begin{bmatrix} I & O \\ O & O \end{bmatrix} \quad \text{and} \quad \tilde{R}_2 = \begin{bmatrix} O & O \\ O & I \end{bmatrix},$$

having the same order as the matrices R_i, and where the identity in $\tilde{R}_1$ is of order $n_1 + n_2$ and that in $\tilde{R}_2$ is of order $n_3 + n_4$. The matrices $\tilde{A}_i$ are defined by

$$\tilde{A}_1 = \begin{bmatrix} A_{11} & A_{12} & & \\ A_{21} & A_{22} & A_{23} & \\ & A_{32} & A_{33} + D_1 & \end{bmatrix}, \qquad \tilde{A}_2 = \begin{bmatrix} A_{22} + D_2 & A_{23} & & \\ A_{32} & A_{33} & A_{34} & \\ & & A_{43} & A_{44} \end{bmatrix}, \tag{7}$$

for which the transmission blocks D_1 and D_2 have to be determined for fast convergence. It has been shown in [5, Theorem 3.2] that the asymptotic convergence factor of AOSM depends on the product of the two norms

$$\| (I + D_1 B_{33})^{-1} [D_1 B_{12} - A_{34} B_{13}] \|, \ \| (I + D_2 B_{11})^{-1} [D_2 B_{32} - A_{21} B_{31}] \|. \tag{8}$$

The blocks B_{ij} depend on the inverses A_{11}^{-1} and A_{44}^{-1} which are expensive to calculate. Minimizing the linear part of equation (8) on matrices D_1 and D_2 within the spaces $\mathcal{S}_1$ and $\mathcal{S}_2$ with distinct sparsity patterns leads to various forms of AOSM. The $O0s$ approach uses a scalar α_i in $D_i = \alpha_i I$, while the O0 method employs a general diagonal matrix D_i and the O2 scheme uses a general tridiagonal matrix D_i. The optimal method, i.e., $D_1 = -A_{34} A_{44}^{-1} A_{43}$ and $D_2 = -A_{21} A_{11}^{-1} A_{12}$, converges in two iterations [5].

3 IGA approximation of transmission conditions

3.1 AOSM approximations of D_1 and D_2

The challenge in approximating the transmission blocks D_1 and D_2 is to capture efficiently the sparsity of the related matrices. In Figure 2 we present different sparsity patterns for the model problem $-\Delta u = f$ in a square domain $\Omega = (0, 1)^2$ for an IGA discretization with 32×32 elements with respect to B-spline degrees $p = 4,\ 5,\ 6$. Because of the structure of the matrices we need to use adapted algorithms which capture efficiently the sparsity of the transmissions blocks D_1 and D_2. For this purpose we introduce a new method, which we call O_{p+1}, that consists in approximating the blocks D_1 and D_2 using $2p + 1$ diagonals, where p is B-spline degree.

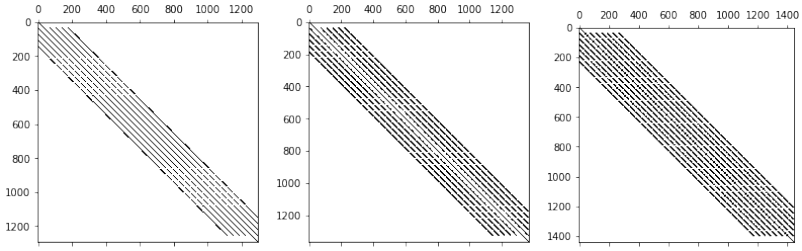


Fig. 2 The sparsity pattern of stiffness matrix in 2D with number of elements 32×32 with respect to spline polynomial degree $p = 4, 5, 6$, and we allows maximum regularity $k = 1$ at the internal knots.

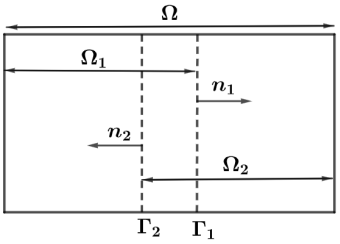


Fig. 3 Domain decomposition into two overlapping subdomains.

3.2 Optimized Schwarz methods for IGA

In this section we consider the Poisson equation

$$\begin{cases} -\Delta u = f, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases} \tag{9}$$

in a square domain $\Omega = (0,1)^2$ with Dirichlet boundary conditions. We decompose the domain Ω into two overlapping subdomains $\Omega_1 = (0,\alpha) \times (0,1)$ and $\Omega_2 = (\beta,1) \times (0,1)$, see Figure 3. The size of the overlap is defined by $\delta = \alpha - \beta$, where $\alpha \geq \beta$ allowing $\alpha = \beta$ for non-overlapping decomposition.

The parallel Schwarz method, introduced by P. Lions, 1990 [7], equipped with Robin boundary conditions for the model problem and the decomposition is

$$\begin{cases} \begin{cases} -\Delta u_1^{n+1} & = f, \text{ in } \Omega_1 = (0,\alpha) \times (0,1), \\ u_1^{n+1} & = 0, \text{ on } \partial\Omega_1 \\ \left(\partial_{n_1} + \lambda_1\right) u_1^{n+1} & = \left(\partial_{n_1} + \lambda_1\right) u_2^n, \text{ on } \Gamma_1 = \{\alpha\} \times (0,1), \end{cases} \\ \begin{cases} -\Delta u_2^{n+1} & = f, \ in \ \Omega_2 = (\beta,1) \times (0,1), \\ u_2^{n+1} & = 0, \text{ on } \partial\Omega_2 \\ \left(\partial_{n_2} + \lambda_2\right) u_2^{n+1} & = \left(\partial_{n_2} + \lambda_2\right) u_1^n, \ on \ \Gamma_2 = \{\beta\} \times (0,1). \end{cases} \end{cases} \tag{10}$$

The OSM is based on finding the optimal parameter set (λ_1, λ_2) that yields a rapid convergence, M. Gander [4] provides an explicit formulas for λ_1 and λ_2 based on Fourier analysis for the model problem $(\eta - \Delta)u = f$. But in our case, no formulas are found yet. Thus, we relied on a numerical approximation supposing that $\lambda_1 = \lambda_2 = \lambda$, then conducting a grid search over a subset of λ to find the best value.

4 Numerical experiments

For our numerical experiments we consider the model problem (9) with two-overlapping decomposition as described before. We allow the parameter $\delta = \alpha - \beta$ to be zero for a non-overlapping decomposition. First we illustrate the performance of the new method O_{p+1} compared the optimal method, O0, O0s, and O2, for the methods labeled "Nonoverlapping" and "Overlapping" correspond to the nonoverlapping block Jacobi and RAS methods respectively (for further details, consult[5, Section 2.1]), see Figures 4, 5. Because of the banded sparsity of the matrices, the optimal method does not converge in two iterations as it is known, see [5, page 10, Proposition 4.4]. The algorithm O_{p+1} has similar behavior as the optimal algorithm. In table 1, we show the number of iterations taken by various methods when used as iterative solvers and as preconditioners for GMRES in order to achieve a residual of 10^{-8}. We can see that AOSMs work well combined with the IGA method, outperforming the classical Schwarz methods.

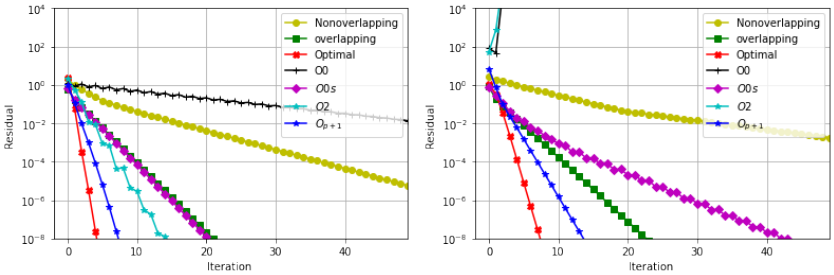


Fig. 4 Convergence history of Additive (7) AOSM with respect to $p = 4, 5$.

In Tables 2 and 3 we present the numerical experiments and the behavior of L^2-norms for the parallel algorithm (10) using isogeometric analysis. We show results for overlapping and non-overlapping decompositions, with the exact solution $u(x, y) = x\,(1 - x)\,y\,(1 - y)$, and $\lambda_1 = \lambda_2 = 0.075$.

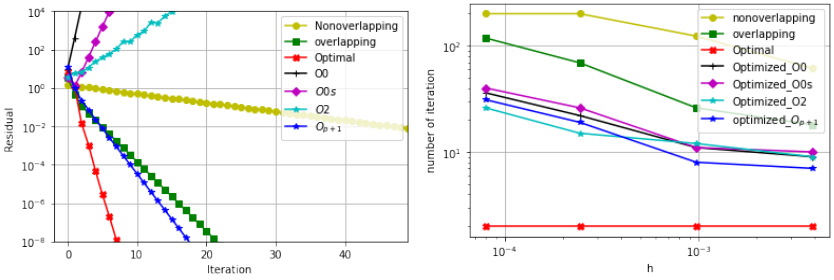


Fig. 5 Left: Convergence history of additive (7) with respect to $p = 6$. Right: The asymptotic behaviors of all methods with respect to h and $p = 2$.

Table 1 Number of iterations to attends a residual of 10^{-8} for: Additive (7) AOSM+IGA used as iterative method: top 32 elements, bottom 64 element in each direction (left), additive (7) AOSM+IGA used as preconditioner method: top 32 elements, bottom 64 element in each direction (right).

degree	Nonoverlap	Overlap	Optimal	O0	O0s	O2	O_{p+1}
1	61	29	2	50	29	NC	NC
2	75	26	2	11	10	12	8
3	87	28	7	28	24	13	9
4	92	24	6	167	24	16	9
5	111	26	9	NC	49	NC	16
6	132	25	9	NC	NC	NC	20
1	122	48	2	187	153	NC	NC
2	135	47	2	122	21	97	9
3	148	43	2	89	29	72	11
4	147	45	5	33	44	17	11
5	146	41	6	41	41	NC	12
6	187	63	4	NC	32	NC	22

degree	Nonoverlap	Overlap	Optimal	O0	O0s	O2	O_{p+1}
1	22	13	2	34	17	NC	NC
2	25	11	2	7	7	8	5
3	26	12	4	12	9	8	5
4	26	12	4	55	10	10	6
5	27	13	5	NC	33	NC	8
6	29	13	5	NC	NC	NC	9
1	23	16	2	68	49	NC	NC
2	27	18	2	73	15	51	3
3	20	19	2	40	15	45	3
4	22	12	3	34	22	8	4
5	25	17	3	21	16	NC	5
6	30	26	2	NC	12	NC	3

Table 2 L^2-norm without overlap after 10 iterations with respect to the number of element 16×16 (left), and 32×32 (right) for OSM method.

degree	$\|u - u_1^h\|_{L^2(\Omega_1)}$	$\|u - u_2^h\|_{L^2(\Omega_2)}$
2	2.80753e-07	3.13938e-06
3	3.82578e-07	1.66811e-06
4	4.89011e-07	1.12886e-06
5	5.38786e-07	8.99693e-07
6	5.71149e-07	7.84679e-07

degree	$\|u - u_1^h\|_{L^2(\Omega_1)}$	$\|u - u_2^h\|_{L^2(\Omega_2)}$
2	5.52967e-07	8.66084e-07
3	5.87442e-07	7.20705e-07
4	6.05761e-07	6.65119e-07
5	6.08555e-07	6.47844e-07
6	6.41174e-07	6.28697e-07

Table 3 L^2-norm with overlap $\delta = 0.2$ after 10 iterations with respect to the number of elements 16×16 (left), and 32×32 (right) for OSM method.

degree	$\|u - u_1^h\|_{L^2(\Omega_1)}$	$\|u - u_2^h\|_{L^2(\Omega_2)}$
2	2.32116e-09	2.32095e-09
3	3.95799e-08	3.88673e-08
4	3.08054e-08	3.08057e-08
5	1.53245e-12	4.18834e-12
6	7.09721e-10	3.29485e-08

degree	$\|u - u_1^h\|_{L^2(\Omega_1)}$	$\|u - u_2^h\|_{L^2(\Omega_2)}$
2	6.62172e-09	8.50223e-10
3	7.10352e-09	7.10352e-09
4	2.85556e-08	3.08303e-08
5	2.40483e-08	2.4419e-08
6	3.0926e-08	1.34111e-09

Concluding remarks

We presented an algebraic computational technique for solving a model problem that has been discetized using IGA. Our numerical experiments suggest that AOSM are well-suited for IGA. However, we found that the methods O0, O0s, and O2 are not effective in capturing the sparsity of IGA matrices, resulting in deteriorating performance. On the other hand, the O_{p+1} method efficiently captures the sparsity of the matrices. Our simulations of OSM for the model problem are encouraging for further analysis of OSM with IGA.

Acknowledgements The authors are thankful to the Chair of Multiphysics and HPC led by Mohammed VI Polytechnic University and sponsored by OCP.

References

1. Bercovier, M. and Soloveichik, I. Overlapping non matching meshes domain decomposition method in isogeometric analysis. *arXiv preprint arXiv:1502.03756* (2015).
2. Da Veiga, L. B., Buffa, A., Sangalli, G., and Vázquez, R. Mathematical analysis of variational isogeometric methods. *Acta Numerica* **23**, 157–287 (2014).
3. Da Veiga, L. B., Cho, D., Pavarino, L. F., and Scacchi, S. Overlapping schwarz methods for isogeometric analysis. *SIAM Journal on Numerical Analysis* **50**(3), 1394–1416 (2012).
4. Gander, M. J. Optimized schwarz methods. *SIAM Journal on Numerical Analysis* **44**(2), 699–731 (2006).
5. Gander, M. J., Loisel, S., and Szyld, D. B. An optimal block iterative method and preconditioner for banded matrices with applications to pdes on irregular domains. *SIAM Journal on Matrix Analysis and Applications* **33**(2), 653–680 (2012).
6. Hughes, T. J., Cottrell, J. A., and Bazilevs, Y. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering* **194**(39-41), 4135–4195 (2005).
7. Lions, P.-L. On the schwarz alternating method. iii: a variant for nonoverlapping subdomains. In: *Third international symposium on domain decomposition methods for partial differential equations*, vol. 6, 202–223. SIAM Philadelphia (1990).
8. Quarteroni, A. and Valli, A. *Domain decomposition methods for partial differential equations*. BOOK. Oxford University Press (1999).
9. Schumaker, L. *Spline functions: basic theory*. Cambridge university press (2007).

An Alternating Approach for Optimizing Transmission Conditions in Algebraic Schwarz Methods

Martin J. Gander, Lahcen Laayouni, and Daniel B. Szyld

1 Introduction

Approximating transmission conditions is very important for Optimized Schwarz Methods (OSM) [2]. For the Algebraic Optimized Schwarz Method (AOSM) [4], approximations need to be done purely algebraically, leading to a challenging minimization problem. A first approach we proposed is to use SPAI [6] to approximate certain intermediate inverses [3]. The resulting method does however not capture the classical behavior of optimized Schwarz methods. In [5] another approach is explored using low-rank approximations, see also [4] for approximate factorization techniques, and [1, 7] for algebraically formulated transmission conditions. We propose here a new approach, based on an alternating method. In section 2 we describe two variants of the alternating method used to approximate the transmission blocks needed in AOSM: a theoretical one using exact inverse information, and a more practical one using SPAI approximations. In section 3 we present numerical evidence to support our findings.

Martin J. Gander
Section de Mathématiques, University of Geneva, Switzerland, e-mail: martin.gander@unige.ch

Lahcen Laayouni
School of Science and Engineering, Al Akhawayn University, Avenue Hassan II, 53000 P.O. Box 1630, Ifrane, Morocco, e-mail: L.Laayouni@aui.ma

Daniel B. Szyld
Department of Mathematics, Temple University (038-16), 1805 N. Broad Street, Philadelphia, Pennsylvania 19122-6094, USA, e-mail: szyld@temple.edu

2 The alternating algorithm to approximate transmission blocks

To describe the alternating algorithm, we consider linear systems of the form

$$Au = f,$$

where the $n \times n$ matrix A usually comes from finite element or finite difference discretizations of a partial differential equation. We further assume that A has a block banded shape of the form

$$A = \begin{bmatrix} A_{11} & A_{12} & & \\ A_{21} & A_{22} & A_{23} & \\ & A_{32} & A_{33} & A_{34} \\ & & A_{43} & A_{44} \end{bmatrix}, \tag{1}$$

with A_{ij} blocks of size $n_i \times n_j$, $i, j = 1, \ldots, 4$, and $n = \sum_i n_i$. The structure of the matrix A corresponds to a two-subdomain decomposition where we assume that $n_1 \gg n_2$ and $n_4 \gg n_3$, i.e. $n_2 + n_3$ is related to the overlap size. For generalizations to more subdomains, see [4, Section 6]. The iteration operators corresponding to the additive and the multiplicative AOSM are given by

$$T_{\text{ORAS}} = I - \sum_{i=1}^{2} \tilde{R}_i^T \tilde{A}_i^{-1} R_i A, \quad \text{and} \quad T_{\text{ORMS}} = \prod_{i=1}^{2} (I - \tilde{R}_i^T \tilde{A}_i^{-1} R_i A), \tag{2}$$

where the classical restriction operators are $R_1 := [I\ O]$ and $R_2 := [O\ I]$, which have order $(n_1 + n_2)n$ and $(n_3 + n_4)n$. The transpose of these operators, R_i^T, are prolongation operators, and $\tilde{R}_i^T$ are RAS-variants thereof, see [4] for more details. The matrices $\tilde{A}_i$ are defined by

$$\tilde{A}_1 = \begin{bmatrix} A_{11} & A_{12} & \\ A_{21} & A_{22} & A_{23} \\ & A_{32} & A_{33} + D_1 \end{bmatrix}, \qquad \tilde{A}_2 = \begin{bmatrix} A_{22} + D_2 & A_{23} & \\ A_{32} & A_{33} & A_{34} \\ & A_{43} & A_{44} \end{bmatrix}, \tag{3}$$

for which the transmission blocks D_1 and D_2 have to be determined for fast convergence. It has been shown in [4, Theorem 3.2] that the asymptotic convergence factor of AOSM depends on the product of the two norms

$$\| (I + D_1 B_{33})^{-1} [D_1 B_{12} - A_{34} B_{13}] \|_2, \ \| (I + D_2 B_{11})^{-1} [D_2 B_{32} - A_{21} B_{31}] \|_2. \tag{4}$$

The goal is to find D_1 and D_2 to minimize the norms in (4), where the B matrices are given by

$$\begin{bmatrix} B_{31} \\ B_{32} \\ B_{33} \end{bmatrix} := \begin{bmatrix} A_{11} & A_{12} & \\ A_{21} & A_{22} & A_{23} \\ & A_{32} & A_{33} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix}, \quad \begin{bmatrix} B_{11} \\ B_{12} \\ B_{13} \end{bmatrix} := \begin{bmatrix} A_{22} & A_{23} & \\ A_{32} & A_{33} & A_{34} \\ & A_{43} & A_{44} \end{bmatrix}^{-1} \begin{bmatrix} I \\ 0 \\ 0 \end{bmatrix}. \tag{5}$$

This implies that

$$B_{13} = -A_{44}^{-1} A_{43} B_{12} \quad \text{and} \quad B_{31} = -A_{11}^{-1} A_{12} B_{32}. \tag{6}$$

Substituting B_{13} and B_{31} into (4), we obtain for the convergence factor estimates

$$\begin{aligned} &\| \left(I + D_1 B_{33}\right)^{-1} \left(D_1 + A_{34} A_{44}^{-1} A_{43}\right) B_{12} \|_2, \\ &\qquad \| \left(I + D_2 B_{11}\right)^{-1} \left(D_2 + A_{21} A_{11}^{-1} A_{12}\right) B_{32} \|_2. \end{aligned} \tag{7}$$

The optimal choice for the transmission matrices making the norms vanish is therefore

$$D_{1,\text{opt}} = -A_{34} A_{44}^{-1} A_{43} \qquad \text{and} \qquad D_{2,\text{opt}} = -A_{21} A_{11}^{-1} A_{12}, \tag{8}$$

which requires however components of the expensive inverses of the large matrices A_{11} and A_{44} and is thus not very practical.

2.1 Alternating algorithm with exact blocks B_{ij}

We start by describing the new alternating algorithm to compute simple diagonal approximations to the optimal $D_{1,\text{opt}}$ in (8) (the algorithm for approximations to $D_{2,\text{opt}}$ is analogous):

Initialization: Set $D_{1,0} := -A_{34} \tilde{A}_{44}^{-1} A_{43}$, where $\tilde{A}_{44}^{-1}$ is a diagonal SPAI approximation of A_{44}^{-1}. Due to the sparsity of A_{34} and A_{43} and the SPAI approximation, $D_{1,0}$ is diagonal and almost constant on the diagonal, except for the two endpoints. For this reason we consider constant diagonal matrices $D_{1,m}$ for $m \geq 1$.

Iteration: For iteration index $m = 1, 2, \ldots$, compute

$$\begin{aligned} p_m &:= \text{argmin}_{p \in \mathbb{R}} \| \left(I + D_{1,m-1} B_{33}\right)^{-1} \left(p\, I + A_{34} A_{44}^{-1} A_{43}\right) B_{12} \|_2; \\ D_{1,m} &:= p_m I; \end{aligned} \tag{9}$$

In (9), we use the exact inverse of the block A_{44}, and we do so also for the blocks B_{12} and B_{33}. The calculation of these blocks is very expensive which makes this first approach expensive. In the next subsection we will present a more practical approach using SPAI approximations for these blocks. Thus the cost in evaluating (9) is reduced significantly.

The minimization problems in (9) are scalar problems for $p \in \mathbb{R}$, but we can obtain tridiagonal and pentadiagonal alternating approximation algorithms by replacing pI in the algorithm above by matrices with tridiagonal and pentadiagonal matrices with constant diagonals leading to 3 and 5 degrees of freedom, respectively. We will use the name Alternating SPAI(1) for diagonal approximations, Alternating SPAI(3) for tridiagonal ones, and Alternating SPAI(5) for pentadiagonal ones.

We next investigate how the alternating algorithm converges to the minimum obtained by globally minimizing the norm in (7). We consider the model prob-

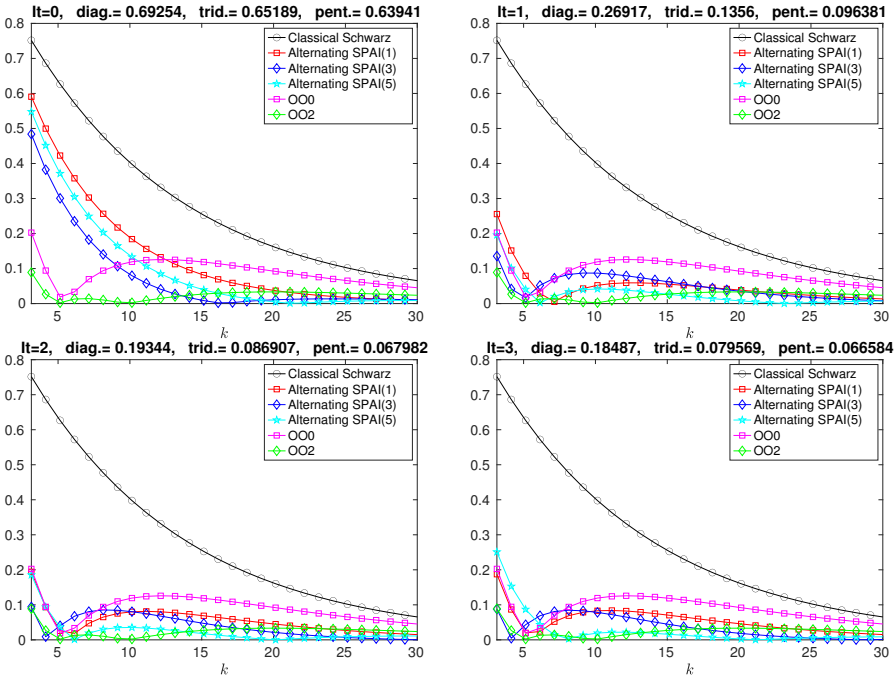


Fig. 1 From top left to bottom right: convergence factor estimates for the initial approximation with SPAI, and then the first three iterations of the new alternating approach.

lem $-\Delta u = f$ in a square domain $\Omega = (0,1)^2$, discretized using standard centered finite differences with mesh size $h = \frac{1}{N+1}$ for $N = 2^5$. We decompose the domain into two equal overlapping subdomains in the x direction with overlap $3h$. In order to visualize the convergence and compare the convergence factor estimates obtained by the alternating method with the convergence factors of the OO0 and OO2 OSM algorithms from [2], we plot them in Fourier space as function of the Fourier variable k in the y direction, see [3] for more details. We show in Figure 1 the results for the initial approximation with SPAI, and then the first 3 iterations of our new alternating algorithm. We see that for the SPAI initial guess, the behavior of the diagonal, tridiagonal and pentadiagonal methods is not like for OSM, their convergence for low frequencies, k small, is more like for the classical Schwarz method. This is consistent with the analysis presented in [3]. With the first correction of our new alternating procedure however, we can see a great improvement for low frequency behavior, the methods obtained from the alternating procedure now behave like OSM. The second and third iterations give further improvements.

In Figure 2, we show on the left the maximum of the two norms in (4) for the first 8 iterations of the alternating algorithm. The algorithm converges very rapidly to the global minimization of the norm (4) shown in Figure 2 on the right.

Iteration	Diagonal	Tridiagonal	Pentadiagonal
0	0.69254	0.651890	0.639410
1	0.26917	0.135600	0.096381
2	0.19344	0.086907	0.067982
3	0.18487	0.079569	0.066584
4	0.18395	0.078645	0.066586
5	0.18385	0.078135	0.066578
6	0.18384	0.078124	0.066579
7	0.18384	0.078123	0.066579
8	0.18384	0.078123	0.066579

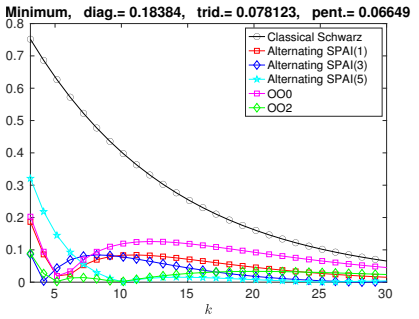


Fig. 2 Left: Maximum of the two norms in (4) for the first 8 iterations of the alternating algorithm. Right: Convergence factors for the global minimization of the norm.

2.2 Alternating algorithm using SPAI approximations for B_{ij}

The alternating algorithm described above requires the calculation of subblocks of A_{11}^{-1} and A_{44}^{-1} and the resulting blocks B_{ij} which is expensive. We now consider SPAI approximations $\tilde{B}_{ij}$ for the blocks B_{ij} and we modify the minimization problem in (9) of the alternating algorithm to

$$p_m = \underset{p\in\mathbb{R}}{\operatorname{argmin}} \| \left(I + D_{1,m-1}\tilde{B}_{33}\right)^{-1} \left(p\,\tilde{B}_{12} - A_{34}\tilde{B}_{13}\right) \|_2. \tag{10}$$

This step thus does no longer require to calculate the inverses A_{11}^{-1} and A_{44}^{-1}, and the modified alternating algorithm requires to compute approximations of the blocks B_{ij} only once.

In Figure 3 we present the behavior of the convergence factor corresponding to each method with respect to the fill-in[1] used in the SPAI approximations for the blocks B_{ij} after 8 iterations. On the top left, we used a diagonal SPAI approximation, and we see that this is not enough for the alternating procedure to improve the low frequency behavior toward OSM. On the top right we used a tridiagonal SPAI approximation and we see that this also does not suffice. In order to obtain good low frequency behavior like OSM, we need to use sufficient fill-in in the SPAI approximations for B_{ij}, as we see in the bottom left and right panels of Figure 3. Note that this is a one time approximation and because of the nature of the SPAI algorithm we can approximate the columns one by one independently, and thus in parallel. In the numerical experiments section we present a comparison between sequential and parallel estimations of B_{ij}.

For the minimization of the linear problems involved in the alternating algorithm we used the Nelder-Mead algorithm implemented in `fminsearch` in Matlab. In the numerical experiments we show that minimizing the norm globally takes more time compared to the time if we minimize 8 linear problems associated to 8 iterations to obtain convergence of the alternating algorithm.

[1] Here, i fill-in means i fill-in entries per column are allowed.

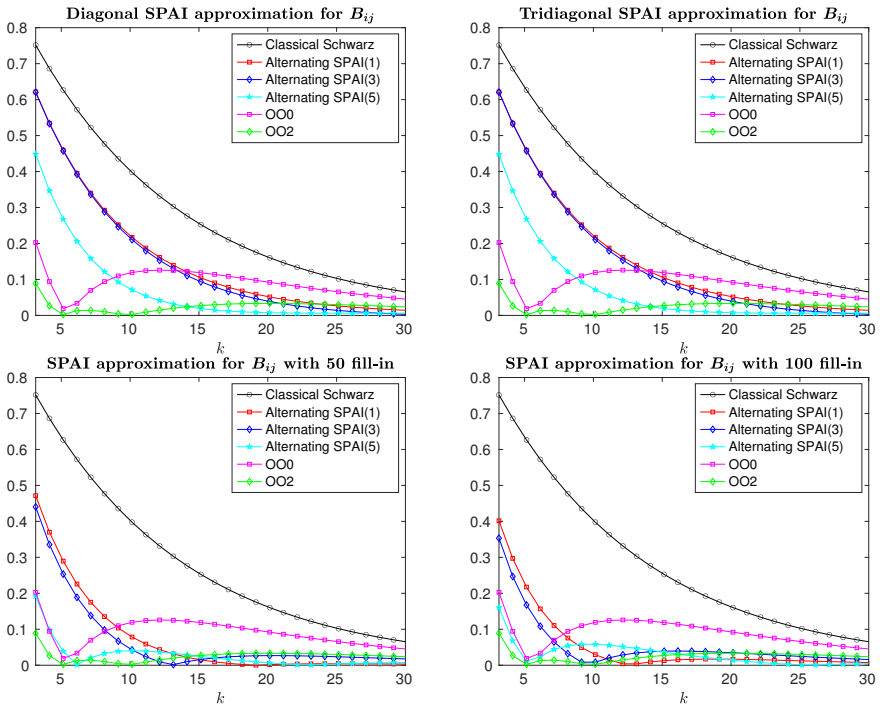


Fig. 3 The behavior of the convergence factor with respect to the fill-in used in the SPAI approximation of the blocks B_{ij} after 3 iterations.

Note that this minimization process can be performed offline, it is independent of the solution process when the Schwarz method is running, and "alternating" here refers to the optimization process, not to the Schwarz method, which can run in parallel or alternating fashion.

3 Numerical experiments

For our numerical experiments we consider the advection-reaction-diffusion equation,

$$\eta u - \nabla \cdot (a \nabla u) + b \cdot \nabla u = f,$$

where $a = a(x, y) > 0$, $b = [b_1(x, y), b_2(x, y)]^T$, $\eta = \eta(x, y) \geq 0$, with

$$b_1 = y - \frac{1}{2}, \quad b_2 = -x + \frac{1}{2}, \quad \eta = x^2 \cos(x + y)^2, \quad a = 1 + (x + y)^2 e^{x-y}.$$

We decompose the unit square domain $\Omega = (0, 1) \times (0, 1)$ into two subdomains $\Omega_1 = (0, \beta) \times (0, 1)$ and $\Omega_2 = (\alpha, 1) \times (0, 1)$, where $0 < \alpha \leq \beta < 1$. Using

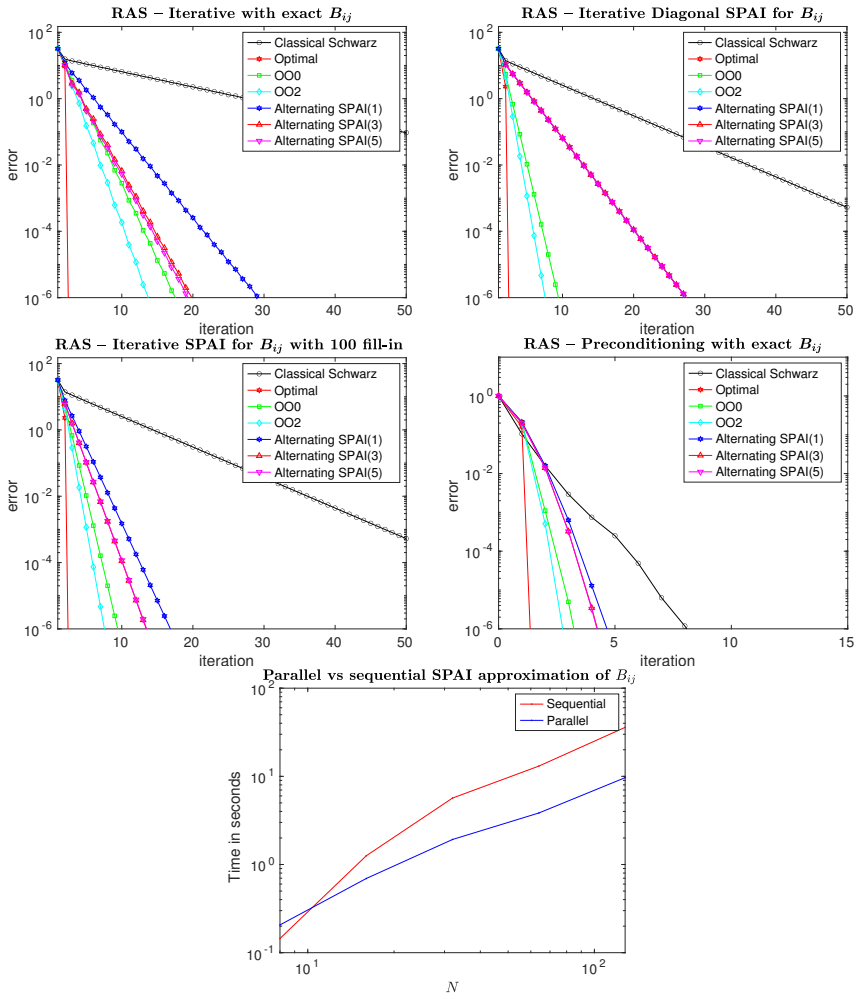


Fig. 4 Convergence of the various methods for the advection-reaction-diffusion model problem. Top left: exact B_{ij}. Top right: diagonal SPAI approximations for B_{ij}. Middle left: SPAI approximations for B_{ij} with 100 fill-in. Middle right: All methods used as preconditioners. Bottom: computational time to compute the corresponding B_{ij} sequentially and in parallel.

a finite difference method, the corresponding matrix A is of size 1024×1024, with a decomposition into two subdomains where the blocks A_{11}, A_{12}, A_{21}, and A_{22} are of size 480×480, 480×32, 32×480, and 32×32 respectively.

In Figure 4 we present the error as a function of the iteration index for the various methods based on the alternating technique. We compare these methods again with OO0, OO2, and also the optimal Schwarz method obtained with the choice (8). We see on the top left in Figure 4 that the alternating SPAI methods are optimized Schwarz methods if we use the exact values of B_{ij}. For alternating

SPAI(1) in the top right in Figure 4, convergence is not as good, but we need only $0.005034 \times 8 = 0.0403$ seconds to calculate the parameter p where 8 is the number of iterations for the alternating algorithm to converge to the minimum. In contrast, we need 4.152525 seconds to calculate the same value of the parameter p if we globally minimize the norm in (4). Using more fill-in in the SPAI approximation, rapid convergence can be recovered, see the bottom-left of Figure 4. This is more expensive, but one can calculate the SPAI approximations for the blocks B_{ij} in parallel. For instance the time needed to calculate the blocks B_{ij} for a 100 fill-in without using `parfor`, in Matlab, is 2.540780 seconds, while with `parfor` we need only 0.005207 seconds.

4 Concluding remarks

We proposed an alternating SPAI technique to minimize the convergence factor estimate for the algebraic optimized Schwarz methods from [4]. By alternating between terms involved in the convergence factor estimate, we reduce the minimization process to solve linear problems instead of non-linear ones. The required time to calculate the parameters of AOSM is thus reduced drastically, but we have also shown that one still needs quite accurate SPAI estimates of the terms in the convergence factor estimate for AOSM in order to obtain good optimized parameters.

Acknowledgements The authors would like to thank the reviewers for their valuable feedback which helped to improve the quality of this manuscript. The first author was supported by the Swiss National Science Foundation.

References

1. Gander, M., Halpern, L., Magoulès, F., and Roux, F.-X. Analysis of patch substructuring methods. *International Journal of Applied Mathematics and Computer Science* **17**(3), 395–402 (2007).
2. Gander, M. J. Optimized schwarz methods. *SIAM Journal on Numerical Analysis* **44**(2), 699–731 (2006).
3. Gander, M. J., Laayouni, L., and Szyld, D. B. Sparse approximate inverse (spai) based transmission conditions for optimized algebraic schwarz methods. In: *Domain Decomposition Methods in Science and Engineering XXVI*, 399–406. Springer (2023).
4. Gander, M. J., Loisel, S., and Szyld, D. B. An optimal block iterative method and preconditioner for banded matrices with applications to pdes on irregular domains. *SIAM Journal on Matrix Analysis and Applications* **33**(2), 653–680 (2012).
5. Gander, M. J. and Outrata, M. Optimized schwarz methods with data-sparse transmission conditions. In: *Domain Decomposition Methods in Science and Engineering XXVI*, 471–478. Springer (2023).
6. Grote, M. J. and Huckle, T. Parallel preconditioning with sparse approximate inverses. *SIAM Journal on Scientific Computing* **18**(3), 838–853 (1997).
7. Roux, F.-X., Magoules, F., Series, L., and Boubendir, Y. Approximation of optimal interface boundary conditions for two-lagrange multiplier feti method. In: *Domain Decomposition Methods in Science and Engineering*, 283–290. Springer (2005).

FETI-DP Algorithms for 2D Biot Model with Discontinuous Galerkin Discretization

Pilhwa Lee

1 Introduction

Poroelasticity, *i.e.*, elasticity of porous media with permeated Darcy flow, pioneered by Biot [7, 8] has been used broadly in geoscience [27] and biomechanics [1, 12, 14] among many others. The difficulties for solving the linear elasticity and incompressible flow problems also arise in solving the poroelastic problem, and there have been diverse mathematical formulations and discretizations. When a continuous Galerkin approach was formulated with mixed finite elements with three-fields of displacement, Darcy flow flux, and pressure [26], the main numerical difficulties are elastic locking and non-physical oscillatory pressure profiles. There have been some new methods for dealing with these difficulties; for example, continuous Galerkin with non-standard three-fields of displacement, pressure, and volumetric stress [24], discontinuous Galerkin formulations [15, 27] with standard three-fields as well as non-conforming mixed finite elements [9, 31]. When lowest-order finite elements are applied, stabilizing terms should be added [6, 11, 28] to satisfy the inf-sup condition [10, 16].

In this paper, we propose a numerical scheme for solving the Biot model with three-fields linear poroelasticity. We consider a discontinuous Galerkin discretization, *i.e.*, the displacement and Darcy flow flux discretized as piecewise continuous in P_1 elements, and the pore pressure as piecewise constant in the P_0 space with a stabilizing term. The emerging formulation is a saddle-point problem, and more specifically, a twofold saddle-point problem. This indefinite system is computationally challenging with slow convergence in iterative methods. It is necessary to incorporate relevant preconditioners for saddle-point problems [5, 23].

FETI-DP algorithms transform indefinite problems to positive definite interface problems of Lagrangian multipliers for subdomains and a primal problem for the

Pilhwa Lee
Department of Mathematics, Morgan State University, 1700 E. Cold Spring Lane, Baltimore, MD, USA, e-mail: Pilhwa.Lee@morgan.edu

coarse space [13]. They have been applied for linear elasticity [21, 25] and incompressible Stokes flows [18, 22, 29, 30] as saddle-point problems. There are theoretical bounds for the condition numbers in the preconditioned systems independent to partitioned subdomains. We show numerical scalability of FETI-DP algorithm preconditioned by Dirichlet preconditioner for the three-fields Biot model discretized with stabilized $P_1 - P_1 - P_0$.

2 Linear poroelastic model

Poroelastic models describe the interaction of fluid flows and deformable elastic porous media saturated in the fluid. Let $\boldsymbol{u}$ be the elastic displacement, p be the pore-pressure. We assume that the permeability is homogeneous: $\mathbf{K} = \kappa\mathbf{I}$. Denote z as the Darcy volumetric fluid flux. The quasi-static Biot model reads as:

$$-(\lambda+\mu)\nabla(\nabla\cdot\mathbf{u}) - \mu\nabla^2\mathbf{u} + \alpha\nabla p = \mathbf{f}, \tag{1}$$

$$\mathbf{K}^{-1}z + \nabla p = \mathbf{b}, \tag{2}$$

$$\frac{\partial}{\partial t}(\alpha\nabla\cdot\boldsymbol{u} + c_0 p) + \nabla\cdot z = g. \tag{3}$$

The first equation is the moment conservation. The second equation is Darcy's law. The third equations is the mass conservation equation. For simplicity, we neglect the effects of gravity acceleration. In the above equations, $\boldsymbol{f}$ is the body force of the solid, $\boldsymbol{b}$ is the body force of the fluid, g is a source or sink term, $c_0 > 0$ is the constrained specific storage coefficient, α is the Biot-Willis constant which is close to 1. λ and μ are the first and second Lamé parameters, respectively.

We consider $\Omega \subset \mathbb{R}^2$ as a bounded domain. For the ease of presentation, we consider mixed partial Neumann and partial Dirichlet boundary conditions in this paper. Specifically, the boundary $\partial\Omega$ is divided into the following:

$$\partial\Omega = \Gamma_\mathrm{d} \cup \Gamma_\mathrm{t} \quad \text{and} \quad \partial\Omega = \Gamma_\mathrm{p} \cup \Gamma_\mathrm{f},$$

where Γ_d and Γ_t are for displacement and stress boundary conditions; Γ_p and Γ_f are for pressure and flux boundary conditions. Accordingly, the boundary conditions are the following:

$$\boldsymbol{u} = \mathbf{0} \quad \text{on } \Gamma_\mathrm{d}, \qquad (\sigma(\boldsymbol{u}) - \alpha p\mathbf{I})\cdot\boldsymbol{n} = \mathbf{t} \quad \text{on } \Gamma_\mathrm{t}, \tag{4}$$

$$p = 0 \quad \text{on } \Gamma_\mathrm{p}, \qquad z\cdot\boldsymbol{n} = g_2 \quad \text{on } \Gamma_\mathrm{f}, \tag{5}$$

where $\sigma(\boldsymbol{u})$ is the deviatoric stress. For simplicity, the Dirichlet conditions are assumed to be homogeneous.

3 Formulation of the Biot model as a saddle-point problem

3.1 Discrete formulation: $\mathbf{P_1 - P_1 - P_0}$

We apply the finite element method with domains normally shaped as triangles in $\mathbb{R}^2$. Let $\mathcal{T}_h$ be a partition of Ω into non-overlapping elements K. We denote by h the size of the largest element in $\mathcal{T}_h$. On the given partition $\mathcal{T}_h$ we apply the following finite element spaces [6]:

$$\boldsymbol{V}_h := \{\boldsymbol{u}_h \in (C^0(\Omega))^2 : \mathbf{u}_h|K \in \mathbf{P}_1(K)\ \forall K \in \mathcal{T}^h, \boldsymbol{u}_h = 0 \text{ on } \Gamma_{\mathrm{d}}\}, \tag{6}$$

$$\boldsymbol{W}_h := \{\boldsymbol{z}_h \in (C^0(\Omega))^2 : \mathbf{z}_h|K \in \mathbf{P}_1(K)\ \forall K \in \mathcal{T}^h, \boldsymbol{z}_h \cdot \boldsymbol{n} = 0 \text{ on } \Gamma_{\mathrm{f}}\}, \tag{7}$$

$$Q_h := \{p_h : p_h|K \in \mathbf{P}_0(K)\ \forall K \in \mathcal{T}^h\}. \tag{8}$$

The problem is to find $(\boldsymbol{u}_h^n, \boldsymbol{z}_h^n, p_h^n) \in \boldsymbol{V}_h \times \boldsymbol{W}_h \times Q_h$ such that

$$\begin{cases} a(\boldsymbol{u}_h^n, \boldsymbol{v}_h) - (p_h^n, \nabla \cdot \boldsymbol{v}_h) = (\boldsymbol{f}^n, \boldsymbol{v}_h) + (\boldsymbol{t}^n, \boldsymbol{v}_h)_{\Gamma_{\mathrm{t}}},\ \forall \boldsymbol{v}_h \in \boldsymbol{V}_h \\ (\mathbf{K}^{-1}\boldsymbol{z}_h^n, \boldsymbol{w}_h) - (p_h^n, \nabla \cdot \boldsymbol{w}_h) = (\boldsymbol{b}^n, \boldsymbol{w}_h),\ \forall \boldsymbol{w}_h \in \boldsymbol{W}_h \\ (\nabla \cdot \boldsymbol{u}_{\Delta t,h}^n, q_h) + \frac{1}{\alpha}(\nabla \cdot \boldsymbol{z}_h^n, q_h) + \frac{c_0}{\alpha}(p_h^n, q_h) + J(p_{\Delta t,h}^n, q_h) = \frac{1}{\alpha}(g^n, q_h), \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall q_h \in Q_h \end{cases} \tag{9}$$

where

$$J(p, q) = \delta_{\mathrm{STAB}} \sum_K \int_{\partial K \setminus \partial\Omega} h_{\partial K}[p][q]ds$$

is a stabilizing term [11], $p_{\Delta t,h}^n = (p_h^n - p_h^{n-1})/\Delta t$, and $\boldsymbol{u}_{\Delta t,h}^n = (\boldsymbol{u}_h^n - \boldsymbol{u}_h^{n-1})/\Delta t$. The finite element discretization will lead to a twofold saddle-point problem of the following form:

$$\begin{bmatrix} A_{\boldsymbol{u}} & 0 & B_1^T \\ 0 & A_{\boldsymbol{z}} & B_2^T \\ B_1 & B_2 & -A_p \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_h \\ z_h \\ p_h \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \boldsymbol{f}_3 \end{bmatrix}. \tag{10}$$

4 FETI-DP formulation for Biot model with discontinuous pressure field

In the algorithm of FETI-DP [13], the domain Ω is decomposed to N nonoverlapping subdomains Ω_i. Each subdomain is with the diameter in the order of H, and the neighboring subdomains are matched across the subdomain interface, $\Gamma = (\cup \partial\Omega_i) \setminus \partial\Omega$.

4.1 FETI-DP algorithm for Biot model: interior and interface spaces

We decompose the discrete displacement space $\boldsymbol{V}$, Darcy flow flux space $\boldsymbol{W}$ into interior and interface spaces ($\mathbf{V} = \mathbf{V}_\mathrm{I} \oplus \mathbf{V}_\Gamma,\ \mathbf{W} = \mathbf{W}_\mathrm{I} \oplus \mathbf{W}_\Gamma$). The discontinuous pressure space Q is decomposed to the constant space Q_0 with the constant pressures on each subdomain and interior space Q_I which has the average zero over each subdomain. Here $\boldsymbol{V}_\mathrm{I}$, $\boldsymbol{W}_\mathrm{I}$, and Q_I are the direct sums of subdomain interior spaces, and $\mathbf{V}_\mathrm{I} = \oplus_{i=1}^{N} \mathbf{V}_\mathrm{I}^i,\ \mathbf{W}_\mathrm{I} = \oplus_{i=1}^{N} \mathbf{W}_\mathrm{I}^i,\ Q_\mathrm{I} = \oplus_{i=1}^{N} Q_\mathrm{I}^i$.

4.2 FETI-DP algorithm for Biot model: primal and dual variables

The interface space $\mathbf{V}_\Gamma$ is further decomposed to primal and dual spaces:

$$\boldsymbol{V}_\Gamma = \boldsymbol{V}_\Delta \oplus \boldsymbol{V}_\Pi = (\oplus_{i=1}^{N} \boldsymbol{V}_\Delta^i) \oplus \boldsymbol{V}_\Pi, \tag{11}$$

where $\boldsymbol{V}_\Pi$ is the continuous, coarse level, and primal space. $\boldsymbol{V}_\Delta$ is the direct sum of independent subdomain dual interface spaces $\boldsymbol{V}_\Delta^i$ [29]. Similarly $\boldsymbol{W}_\Gamma$ is decomposed to $\boldsymbol{W}_\Delta$ and $\boldsymbol{W}_\Pi$.

Let us represent $\boldsymbol{u}$ and z together as $\boldsymbol{U} = (\boldsymbol{u}, z) \in \boldsymbol{V} \times \boldsymbol{W}$. The problem turns out to find $(\boldsymbol{u}_\mathrm{I}, z_\mathrm{I}, p_\mathrm{I}, \boldsymbol{u}_\Pi, z_\Pi, \boldsymbol{u}_\Delta, z_\Delta, p_0) \in \boldsymbol{V}_\mathrm{I} \times \boldsymbol{W}_\mathrm{I} \times Q_\mathrm{I} \times \boldsymbol{V}_\Pi \times \boldsymbol{W}_\Pi \times \boldsymbol{V}_\Delta \times \boldsymbol{W}_\Delta \times Q_0$ such that

$$\begin{bmatrix} A_{\mathrm{II}} & B_{\mathrm{II}}^T & A_{\Pi\mathrm{I}}^T & A_{\Delta\mathrm{I}}^T & 0 \\ B_{\mathrm{II}} & 0 & B_{\mathrm{I}\Pi} & B_{\mathrm{I}\Delta} & 0 \\ A_{\Pi\mathrm{I}} & B_{\mathrm{I}\Pi}^T & A_{\Pi\Pi} & A_{\Delta\Pi}^T & B_{0\Pi}^T \\ A_{\Delta\mathrm{I}} & B_{\mathrm{I}\Delta}^T & A_{\Delta\Pi} & A_{\Delta\Delta} & B_{0\Delta}^T \\ 0 & 0 & B_{0\Pi} & B_{0\Delta} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_\mathrm{I} \\ p_\mathrm{I} \\ \boldsymbol{U}_\Pi \\ \boldsymbol{U}_\Delta \\ p_0 \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_\mathrm{I} \\ 0 \\ \boldsymbol{f}_\Pi \\ \boldsymbol{f}_\Delta \\ 0 \end{bmatrix}. \tag{12}$$

4.3 FETI-DP algorithm for Biot model: Schur complement

A Schur complement operator $\tilde{S}$ is defined in the following:

$$\begin{bmatrix} A_{\mathrm{II}} & B_{\mathrm{II}}^T & A_{\Pi\mathrm{I}}^T & 0 & A_{\Delta I}^T \\ B_{\mathrm{II}} & 0 & B_{\mathrm{I}\Pi} & 0 & B_{\mathrm{I}\Delta} \\ A_{\Pi\mathrm{I}} & B_{\mathrm{I}\Pi}^T & A_{\Pi\Pi} & B_{0\Pi}^T & A_{\Delta\Pi}^T \\ 0 & 0 & B_{0\Pi} & 0 & B_{0\Delta} \\ A_{\Delta\mathrm{I}} & B_{\mathrm{I}\Delta}^T & A_{\Delta\Pi} & B_{0\Delta}^T & A_{\Delta\Delta} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_\mathrm{I} \\ p_\mathrm{I} \\ \boldsymbol{U}_\Pi \\ p_0 \\ \boldsymbol{U}_\Delta \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \tilde{S}\boldsymbol{U}_\Delta \end{bmatrix}. \tag{13}$$

We introduce Lagrange multiplier λ and the jump operator B_Δ to enforce the continuity of $\boldsymbol{U}_\Delta$ across Γ [22]:

$$\begin{bmatrix} \tilde{S} & B_\Delta^T \\ B_\Delta & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_\Delta \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_\Delta^* \\ 0 \end{bmatrix}. \tag{14}$$

The problem is reduced to find $\lambda \in \Lambda = B_\Delta \boldsymbol{U}_\Delta$ such that

$$B_\Delta \tilde{S}^{-1} B_\Delta^T \lambda = B_\Delta \tilde{S}^{-1} \boldsymbol{f}_\Delta^*. \tag{15}$$

This is solved by Preconditioned Conjugate Gradient (PCG).

4.4 Dirichlet preconditioner

We define a Schur complement operator, the discrete Harmonic $H_\Delta^{(i)}$ on Ω_i as follows:

$$\begin{bmatrix} A_{\mathrm{II}}^{(i)} & A_{\mathrm{I}\Delta}^{(i)} \\ A_{\Delta I}^{(i)} & A_{\Delta\Delta}^{(i)} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_{\mathrm{I}}^{(i)} \\ \boldsymbol{U}_\Delta^{(i)} \end{bmatrix} = \begin{bmatrix} 0 \\ H_\Delta^{(i)} \boldsymbol{U}_\Delta^{(i)} \end{bmatrix}. \tag{16}$$

The Dirichlet preconditioner is formulated in the following:

$$M_{\lambda,D}^{-1} = B_{\Delta,D} H_\Delta B_{\Delta,D}^T, \tag{17}$$

where $B_{\Delta,D}$ is a scaled operator obtained from B_Δ by the scaling factor $1/N_x$ with N_x as the number of subdomains sharing each node x in the interface Γ. H_Δ is the direct sum of $H_\Delta^{(i)}$ [30].

5 Numerical experiments

A test problem is formulated with $\alpha = 1$, $c_0 = 0$, $\Omega = [0,1]^2$, and $t \in [0, 0.25]$:

$$\begin{aligned} -(\lambda+\mu)\nabla(\nabla\cdot\boldsymbol{u}) - \mu\nabla^2\boldsymbol{u} + \nabla p &= 0, \\ \mathbf{K}^{-1} z + \nabla p &= 0, \\ \nabla\cdot(\boldsymbol{u}_t + z) &= g_1. \end{aligned} \tag{18}$$

The involving initial and boundary conditions are the following:

$$\begin{cases} \boldsymbol{u} = \mathbf{0} \quad \text{on } \partial\Omega = \Gamma_{\mathrm{d}}, \\ z\cdot\boldsymbol{n} = g_2 \quad \text{on } \partial\Omega = \Gamma_{\mathrm{f}}, \\ \boldsymbol{u}(\boldsymbol{x},0) = 0, \boldsymbol{x} \in \Omega, \\ p(\boldsymbol{x},0) = 0, \boldsymbol{x} \in \Omega. \end{cases} \tag{19}$$

We consider the following analytic solution:

$$
\begin{aligned}
\boldsymbol{u} &= \frac{-1}{4\pi(\lambda+2\mu)} \begin{bmatrix} \cos(2\pi x)\sin(2\pi y)\sin(2\pi t) \\ \sin(2\pi x)\cos(2\pi y)\sin(2\pi t) \end{bmatrix}, \\
z &= -2\pi k \begin{bmatrix} \cos(2\pi x)\sin(2\pi y)\sin(2\pi t) \\ \sin(2\pi x)\cos(2\pi y)\sin(2\pi t) \end{bmatrix}, \\
p &= \sin(2\pi x)\sin(2\pi y)\sin(2\pi t),
\end{aligned} \tag{20}
$$

and derive the compatible source term of g_1.

5.1 Numerical implementation

In the implementation of finite elements, we use a finite element library, libMesh [19]. We apply triangular elements with 3 nodes. For domain partitioning, we apply ParMETIS [17]. Krylov subspace iterative main solver of Preconditioned Conjugate Gradient (PCG) and FETI-DP algorithms are based on PETSc [3, 2, 4] and KSPFETIDP and PCBDDC classes within PETSc [32]. The initial guess is zero and the stopping criterion is set to be 10^{-8}, the reduction of the residual norm. The stabilizing factor is $\delta_{\text{STAB}} = 100$, the time-stepping is $dt = 0.00625$, and Young's modulus $E = 1000$ Pa. In each test, we count the iteration of the FETI-DP solver.

5.2 Scalability of FETI-DP algorithms

Scalability of FETI-DP preconditioning for the Biot model is tested with increasing number of subdomains N. The subdomain size H/h is set with 8, 12, or 16. In the first case, ($\nu = 0.3, k = 10^{-2}$) of compressible elasticity and permeable Darcy flow is tested with Dirichlet preconditioner. As shown in Table 1, FETI-DP iteration numbers

Table 1 Scalability of the FETI-DP algorithms with Dirichlet preconditioner for the saddle-point problem of Biot model. Iteration counts for increasing number of subdomains N. Fixed $\delta_{\text{STAB}} = 100$, $dt = 0.00625$, and $E = 1000$.

	$H/h = 8$		$H/h = 12$		$H/h = 16$	
	$\nu = 0.3$ $k = 10^{-2}$	$\nu = 0.4999$ $k = 10^{-7}$	$\nu = 0.3$ $k = 10^{-2}$	$\nu = 0.4999$ $k = 10^{-7}$	$\nu = 0.3$ $k = 10^{-2}$	$\nu = 0.4999$ $k = 10^{-7}$
N	iteration	iteration	iteration	iteration	iteration	iteration
2×2	4	9	4	10	4	11
3×3	5	9	5	12	5	14
4×4	5	9	5	12	5	17
5×5	5	10	5	13	5	15
6×6	5	11	5	13	5	16
7×7	5	12	5	15	5	16
8×8	5	12	5	13	5	17

are bounded when subdomains were increased from 2×2 to 8×8. In the second case, ($\nu = 0.4999, k = 10^{-7}$) of almost incompressible elasticity and less permeable Darcy flow is tested with Dirichlet preconditioner. FETI-DP iteration numbers are larger than the first case, but still bounded while subdomains are increased from 2×2 to 8×8, showing no issues of elastic locking. This is consistent with a theoretical scalability of FETI-DP for almost incompressible elasticity [20].

6 Conclusion

We have explored the scalability of the FETI-DP algorithms for the 2D Biot model. Upon numerical scalabilities of compressible elasticity with Darcy's flow as well as almost incompressible elasticity with limited Darcy's flow, it remains to test parameter robustness, possibly in the presence of heterogeneity of parameters. Overall, the numerical results are a foundation for further advancement of scalable FETI-DP / BDDC preconditioners for poroelastic large deformation.

Acknowledgements The author gives thanks to Dr. Mingchao Cai for the introduction of Biot models and invaluable discussions. The author was partly supported by NSF DMS-1831950, and the virtual attendance to DD27 conference by Penn State University NSF travel fund.

References

1. Badia, S., Quaini, A., and Quarteroni, A. Coupling Biot and Navier-Stokes equations for modelling fluid-poroelastic media interaction. *J Comput Phys* **228**, 7986–8014 (2009).
2. Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., Kong, F., Kruger, S., May, D. A., McInnes, L. C., Mills, R. T., Mitchell, L., Munson, T., Roman, J. E., Rupp, K., Sanan, P., Sarich, J., Smith, B. F., Zampini, S., Zhang, H., Zhang, H., and Zhang, J. PETSc/TAO users manual. Tech. Rep. ANL-21/39 - Revision 3.19, Argonne National Laboratory (2023).
3. Balay, S., Abhyankar, S., Adams, M. F., Benson, S., Brown, J., Brune, P., Buschelman, K., Constantinescu, E. M., Dalcin, L., Dener, A., Eijkhout, V., Faibussowitsch, J., Gropp, W. D., Hapla, V., Isaac, T., Jolivet, P., Karpeev, D., Kaushik, D., Knepley, M. G., Kong, F., Kruger, S., May, D. A., McInnes, L. C., Mills, R. T., Mitchell, L., Munson, T., Roman, J. E., Rupp, K., Sanan, P., Sarich, J., Smith, B. F., Zampini, S., Zhang, H., Zhang, H., and Zhang, J. PETSc Web page. https://petsc.org/ (2023). URL https://petsc.org/.
4. Balay, S., Gropp, W., McInnes, L., and Smith, B. *Modern Software Tools in Scientific Computing*, chap. Efficient management of parallelism in object oriented numerical software libraries, 163–202. Birkhäuser Press (1997).
5. Benzi, M., Golub, G., and Liesen, J. Numerical solution of saddle point problems. *Acta Numerica* **14**, 1–137 (2005).
6. Berger, L., Bordas, R., Kay, D., and Tavener, S. Stabilized lowest-order finite element approximation for linear three-field poroelasticity. *SIAM J Sci Comput* **37**, A2222–A2245 (2015).
7. Biot, M. General theory of three-dimensional consolidation. *J Appl Phys* **12**, 155–164 (1941).
8. Biot, M. Theory of elasticity and consolidation for a porous anisotropic solid. *J Appl Phys* **26**, 182–185 (1955).

9. Boffi, D., Botti, M., and Di Pietro, D. A nonconforming high-order method for the Biot problem on general meshes. *SIAM J Sci Comput* **38**, A1508–A1537 (2016).
10. Brezzi, F. and Fortin, M. *Mixed and hybrid finite element methods*, *Springer Series in Computational Mathematics*, vol. 15. Springer-Verlag, New York (1991).
11. Burman, E. and Hansbo, P. A unified stabilized method for stokes' and darcy's equations. *J Comput Appl Math* **198**, 35–51 (2007).
12. Chapelle, D., Gerbeau, J.-F., Sainte-Marie, J., and Vignon-Clementel, I. A poroelastic model valid in large strains with applications to perfusion in cardiac modeling. *Comput Mech* **46**, 91–101 (2010).
13. Farhat, C., Lesoinne, M., LeTallec, P., Pierson, K., and Rixen, D. FETI-DP: a dual-primal unified FETI method-part I: A faster alternative to the two-level FETI method. *Int J Numer Engng* **50**, 1523–1544 (2001).
14. Heath Richardson, S., Gao, H., Cox, J., Janiczek, R., Griffith, B., Berry, C., and Luo, X. A poroelastic immersed finite element framework for modeling cardiac perfusion and fluid-structure interaction. *Int J Numer Method Biomed Eng* **37**, e3446 (2021).
15. Hong, Q. and Kraus, J. Parameter-robust stability of classical three-field formulation of biot's consolidation model. *Elec. Trans. Numer. Anal.* **48**, 202–226 (2018).
16. Howell, J. and Walkington, N. Inf-sup conditions for twofold saddle point problems. *Numer Math* **118**, 663–693 (2011).
17. Karypis, G. *Encyclopedia of Parallel Computing*, chap. METIS and ParMETIS, 1117–1124. Springer, New York (2011).
18. Kim, H. and Lee, C.-O. A neumann-dirichlet preconditioner for a FETI-DP formulation of the two-dimensional stokes problem with mortar methods. *SIAM J Sci Comput* **28**, 1133–1152 (2006).
19. Kirk, B., Peterson, J., Stogner, R., and Carey, G. F. libmesh: a c++ library for parallel adaptive mesh refinement/coarsening simulations. *Engineering with Computers* **22**, 237–254 (2006).
20. Klawonn, A., Rheinbach, O., and Wohlmuth, B. Dual-primal iterative substructuring for almost incompressible elasticity. In: *Domain decomposition methods in science and engineering*, *Lecture Notes in Computational Science and Engineering*, vol. 16. International Conference on Domain Decomposition Methods, Springer, Heidelberg (2005).
21. Klawonn, A. and Widlund, O. Dula-primal FETI methods for linear elasticity, communications on pure and applied mathematics. *Communications on Pure and Applied Mathematics* **59**, 1523–1572 (2006).
22. Li, J. A dual-primal FETI method for incompressible stokes equations. *Numer. Math.* **102**, 257–275 (2005).
23. Mardal, K. and Winther, R. Preconditioning discretizations of systems of partial differential equations. *Numer Lin Alg Appl* **18**, 1–40 (2011).
24. Oyarzua, R. and Ruiz-Baier, R. Locking-free finite element methods for poroelasticity. *SIAM J Numer Anal* **54**, 2951–2973 (2016).
25. Pavarino, L. and Scacchi, S. Isogeometric block FETI-DP preconditioners for the stokes and mixed linear elasticity systems. *Comput Methods Appl Mech Engrg* **310**, 694–710 (2016).
26. Phillips, P. and Wheeler, M. A coupling of mixed and continuous galerkin finite-element methods for poroelasticity I: the continuous in time case. *Comput Geosci* **11**, 131–144 (2007).
27. Phillips, P. and Wheeler, M. A coupling of mixed and discontinuous galerkin finite-element methods for poroelasticity. *Comput Geosci.* **12**, 417–435 (2008).
28. Rodrigo, C., Hu, X., Ohm, P., Adler, J., Gaspar, F., and Zikatanov, L. New stabilized discretizations for poroelasticity and the stokes' equations. *Comput Methods Appl Mech Engrg* **341**, 467–484 (2018).
29. Tu, X. and Li, J. A unified FETI-DP approach for incompressible stokes equations. *Internat J Numer Methods Engrg* **94**, 128–149 (2013).
30. Tu, X. and Li, J. A FETI-DP type domain decomposition algorithm for three-dimensional incompressible stokes equations. *SIAM J Numer Anal* **53**, 720–742 (2015).
31. Yi, S.-Y. A coupling of nonconforming and mixed finite element methods for Biot's consolidation model. *Numer Methods Partial Differ Equ* **29**, 1749–1777 (2013).
32. Zampini, S. PCBDDC: a class of robust dual-primal methods in PETSc. *SIAM J Sci Comput* **38**, S282–S306 (2016).

Linear, Super-Linear and Combined Fourier Heat Kernel Convergence Estimates for Schwarz Waveform Relaxation

Martin J. Gander and Véronique Martin

1 Introduction

We are interested in solving the heat equation $\partial_t u - \tilde{\nu}\partial^2_{xx} u = f$ on $(-L, L) \times (0, T)$, with an initial condition and with Dirichlet boundary conditions. We will use a Schwarz Waveform Relaxation (SWR) method and want to study the convergence of this algorithm. More precisely our goal is to understand the influence of T and L on the convergence. We therefore study the equation on an adimentionalized domain

$$\begin{aligned} \mathcal{L}u &:= \frac{\partial u}{\partial t} - \nu \frac{\partial u}{\partial x^2} = f \quad \text{on } (-1,1) \times (0,1), \\ u(-1,\cdot) &= g_{-1}, \\ u(1,\cdot) &= g_1, \\ u(\cdot,0) &= u_0, \end{aligned} \tag{1}$$

where $\nu = \frac{\tilde{\nu} T}{L^2} > 0$. Then it suffices to study the influence of ν on the convergence speed of the algorithm.

We will consider the SWR algorithm with Dirichlet boundary conditions ($\delta > 0$ is the overlap)

$$\begin{array}{llll} \mathcal{L}u_1^k = f & \text{on } (-1,\delta) \times (0,1), & \mathcal{L}u_2^k = f & \text{on } (0,1) \times (0,1), \\ u_1^k(\delta,\cdot) = u_2^{k-1}(\delta,\cdot) & \text{on } (0,1), & u_2^k(0,\cdot) = u_1^k(0,\cdot) & \text{on } (0,1), \\ u_1^k(\cdot,0) = u_0 & \text{on } (-1,\delta), & u_2^k(\cdot,0) = u_0 & \text{on } (0,1), \\ u_1^k(-1,\cdot) = g_{-1} & \text{on } (0,1), & u_2^k(1,\cdot) = g_1 & \text{on } (0,1). \end{array} \tag{2}$$

Martin J. Gander
Section de Mathématiques, Université de Genève, Suisse, e-mail: martin.gander@unige.ch

Véronique Martin
UMR CNRS 7352, Université de Picardie Jules Verne, Amiens, France, e-mail: veronique.martin@u-picardie.fr

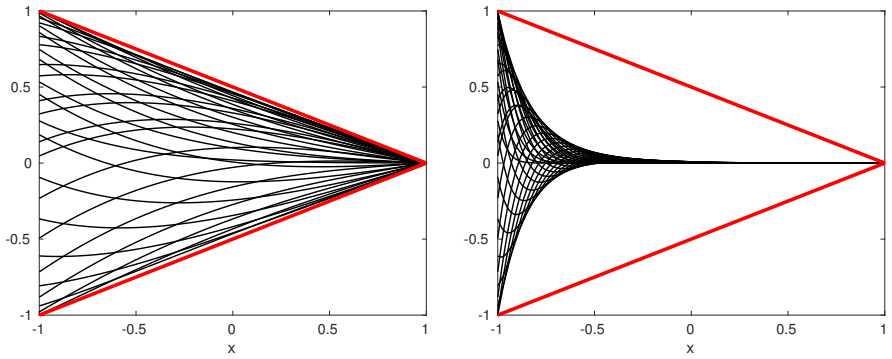


Fig. 1 Solution (in black) at several time steps of the heat equation (1) when $g_{-1}(t) = \sin(3\pi t)$, $g_1(t) = 0$ for $\nu = 10$ (left) or $\nu = 0.1$ (right). In red the bound given by Lemma 1.

The error $e_j^k := u - u_j^k$, $j = 1, 2$ satisfies by linearity again the same algorithm (2) but with homogeneous data, i.e. $f = 0$, $u_0 = 0$, $g_{-1} = g_1 = 0$.

In Sections 2 and 3 we recall convergence results proved using the maximum principle and we give numerical illustrations to understand the domain of validity of each convergence bound. Then we explain in Section 4 how the Fourier transform is usually used to measure the convergence speed of the algorithm and we discuss this strategy when it is applied to a stationary or to an unstationary equation. We end in Section 5 with numerical results to summarize the different regimes of convergence depending on the value of L and T (or equivalently on the value of ν).

2 Linear bound due to the maximum principle

In [4] a theorem is proved which gives a linear bound for the error corresponding to algorithm (2). It relies on

Lemma 1 *If u is solution of the heat equation* (1) *with $u_0 = 0$, $f = 0$ then*

$$\|u(x,\cdot)\|_\infty \le ((1-x)\|g_{-1}\|_\infty + (x+1)\|g_1\|_\infty)/2, \ -1 \le x \le 1,$$

where $\|g\|_\infty = \sup_{t\in[0,1]} |g(t)|$.

Note that this bound does not depend on the value of ν. If ν is large then u tends to satisfy $\partial_{xx} u \simeq 0$ and then u tends to be linear. The bound is sharp in this case. However, if ν is small the solution tends to decay rapidly away from the boundary and is close to 0 except near $x = -1$ and $x = 1$ where boundary layers appear. The bound is not sharp in this case. In Figure 1 we show examples of the solution of the heat equation in these two cases. Using Lemma 1, the following theorem is proved in [4]:

Theorem 1 *The error of algorithm* (2) *satisfies for any $k \ge 1$*

$$\|e_1^k(0,\cdot)\|_\infty \le \left(\frac{1-\delta}{1+\delta}\right)^k \|e_1^0(0,\cdot)\|_\infty.$$

We expect this bound to be sharp for small spatial domains or large time, corresponding to the case of a large value of ν.

3 Superlinear bound

In [4] a superlinear bound is proved for the error of the algorithm (2):

Theorem 2 *The error in algorithm* (2) *satisfies for any* $k \geq 1$ *the superlinear bound*

$$\|e_1^k(0,\cdot)\|_\infty \leq erfc\left(\frac{k\delta}{2\sqrt{\nu}}\right)\|e_1^0(0,\cdot)\|_\infty,$$

where $erfc(x) = \frac{2}{\sqrt{\pi}}\int_x^{+\infty} e^{-t^2}dt$ *is the complementary error function.*

The proof (see [4]) consists in comparing $e_1^k(0,\cdot)$ and $\overline{e}_1^k(0,\cdot)$ where $\overline{e}_1^k$ is defined on an infinite spatial domain by

$$\begin{cases} \mathcal{L}\overline{e}_1^k = 0 & \text{on } (-\infty,\delta)\times(0,1), \\ \overline{e}_1^k(\cdot,0) = 0 & \text{on } (-\infty,\delta), \\ \overline{e}_1^k(\delta,t) = \max\limits_{0\leq\tau\leq t}|e_2^{k-1}(\delta,\tau)| & \text{on } (0,1), \\ \lim\limits_{x\to-\infty}\overline{e}_1^k(x,t) = 0 & \text{on } (0,1). \end{cases}$$

Using the maximum principle we have

$$|e_1^k(0,t)| \leq \overline{e}_1^k(0,t) = \int_0^t \|e_2^{k-1}(\delta,\cdot)\|_{L^\infty(0,\tau)}K(\delta,t-\tau)d\tau,$$

where the last equality is obtained since in the infinite domain $(-\infty,\delta)$ the solution $\overline{e}_1^k(0,t)$ can be computed using the heat kernel $K(x,t) = \frac{x}{2\sqrt{\pi}}\frac{e^{-\frac{x^2}{4t}}}{t^{3/2}}$. The result is then obtained by induction. In Figure 2 we compare e_1^k and $\overline{e}_1^k$. We can see that for a large value of ν the superlinear bound is not sharp (due to the fact that $\overline{e}_1^k$ is computed on an infinite spatial domain) while for a small ν a boundary layer has appeared and the superlinear bound gives a sharper estimate than the linear bound.

4 Analysis using Fourier arguments

While in [4] and [5] the SWR for the heat equation were studied using arguments coming from the PDE analysis, in [6] a method is proposed to use the Fourier transform to obtain the convergence factor of a Schwarz algorithm for the stationary convection-diffusion equation, and this technique was rapidly also applied to a time dependent equation in [2], namely the heat equation.

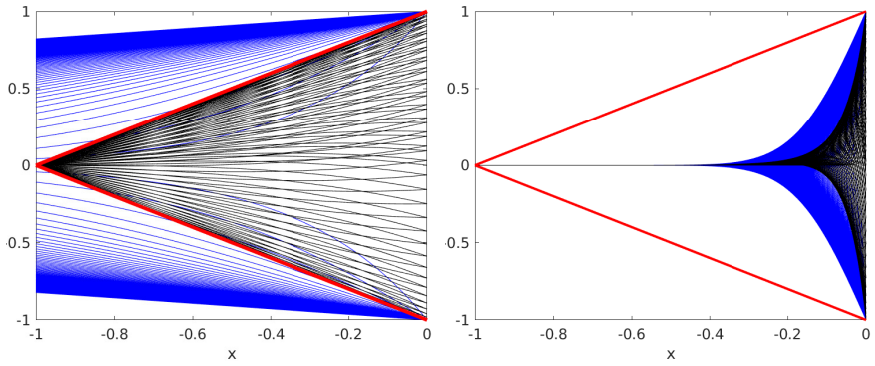


Fig. 2 Comparison of $e_1^k(\cdot,t)$ (in black) and $\overline{e}_1^k(\cdot,t)$ (in blue) for several values of t. Here $\delta = 0$ and $\nu = 10$ (left) or $\nu = 0.01$ (right). In red the bound given by Lemma 1.

In the infinite spatial domain $\mathbb{R}$ and infinite time domain $\mathbb{R}^+$, the strategy in the time dependent case consists in solving algorithm (2) for the errors in Laplace variables. If $s := \sigma + i\omega$, $\sigma, \omega \in \mathbb{R}$ let $\hat{f}(s) = \int_0^{+\infty} f(t)e^{-st}dt$, $\Re(s) \geq \alpha$ be the Laplace transform of the function $f \in L^1(\mathbb{R})$ such that $|f(t)| \leq Ce^{\alpha t}$, $C > 0$ and α constants.

We first obtain

$$\hat{e}_1^k(x,s) = \alpha^k e^{\sqrt{\frac{s}{\nu}}x} \quad \text{and} \quad \hat{e}_2^k(x,s) = \beta^k e^{-\sqrt{\frac{s}{\nu}}x}.$$

We suppose that the algorithm for the error is initialized with $e_2^0(\delta,t) = g(t)$. By induction using the Dirichlet boundary conditions we obtain

$$\hat{e}_1^k(0,s) = \rho(s)^{(2k-1)}\hat{g}(s),$$

where $\rho(s) := e^{-\sqrt{\frac{s}{\nu}}\delta}$ is the convergence factor of the algorithm.

This formula seems to say that $\rho(\sigma + i\omega)$ explains the convergence behavior of the single frequency ω. We will see in the next subsections that this is true for a stationary problem like the screened Laplace equation. However the situation is more complex for an unstationary problem like the heat equation. To understand this point, let us back-transform the previous formula to obtain

$$e_1^k(0,t) = \int_0^t g(t-\tau)K((2k-1)\frac{\delta}{\sqrt{\nu}},\tau)d\tau, \tag{3}$$

where $K(x,t) = \frac{x}{2\sqrt{\pi}}\frac{e^{-\frac{x^2}{4t}}}{t^{3/2}}$ is the heat kernel. We see that the error is expressed as a convolution between the heat kernel and g.

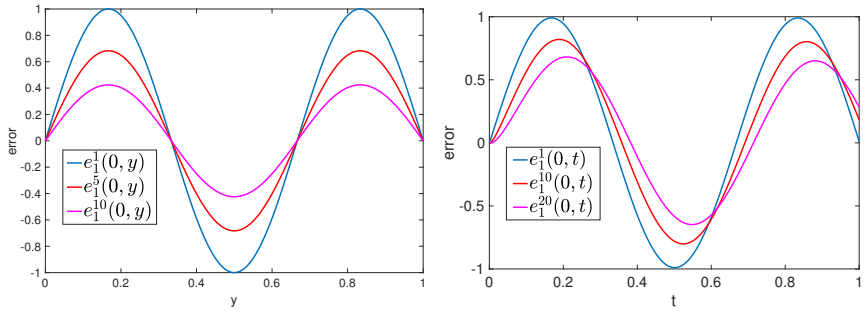


Fig. 3 On the left, errors $e_1^k(0,y)$ for the screened equation at iterations $k=1$, $k=5$ and $k=10$ when the first guess is $e_2^0(\delta,y)=\sin(3\pi y)$. On the right, errors $e_1^k(0,t)$ for the heat equation at iterations $k=1$, $k=10$ and $k=20$ when the first guess is $e_2^0(\delta,t)=\sin(3\pi t)$.

4.1 Using Fourier arguments is different for time dependent and stationary problems

To understand the difference between the stationary case and the unstationary one, we first consider the screened Laplace equation $\tilde{\mathcal{L}}u := \eta u - \triangle u = f$ in $\Omega := \mathbb{R}^2$, with $\eta > 0$. If the domain Ω is split into the two overlapping subdomains $\Omega_1 := (-\infty,\delta)\times\mathbb{R}$ and $\Omega_2 := (0,+\infty)\times\mathbb{R}$, where $\delta > 0$ is the overlap parameter, then the classical Schwarz algorithm (for the errors) solves for iteration index $k=1,\ldots$

$$\begin{array}{ll} \tilde{\mathcal{L}}e_1^k = 0 & \text{on } (-\infty,\delta)\times\mathbb{R}, \\ e_1^k(\delta,\cdot) = e_2^{k-1}(\delta,\cdot) & \text{on } \mathbb{R}, \end{array} \qquad \begin{array}{ll} \tilde{\mathcal{L}}e_2^k = 0 & \text{on } (0,+\infty)\times\mathbb{R}, \\ e_2^k(0,\cdot) = e_1^k(0,\cdot) & \text{on } \mathbb{R}. \end{array} \tag{4}$$

If the initial error is a pure sine signal on the interface, $e_2^0(\delta,y) := \sin(\lambda y)$, then the errors for each iteration $k=1,2,\ldots$ can be obtained by a direct computation to be

$$e_1^k(0,y) = e^{-(2k-1)\delta\sqrt{\eta+\lambda^2}}\sin(\lambda y) =: \tilde{\rho}(\lambda)^{2k-1}\sin(\lambda y),$$

which means that at each iteration the initial sine error is contracted by the convergence factor $\tilde{\rho}(\lambda)$. This result is consistent with the definition of the convergence factor in [1] which was obtained by a Fourier transform in the y direction with Fourier variable ω.

In Figure 3 left, we can see the errors $e_1^k(0,y)$ at iterations $k=1$, $k=5$ and $k=10$. The initial sine is contracted as the iterations grow as predicted by the previous formula. Let us see what happens for the heat equation (Figure 3, right): a sine is introduced ($e_2^0(\delta,t) := \sin(\lambda t)$) and we can see now that the sine is not just contracted anymore, it is also transported! We can use the formula (3) to understand that $e_1^k(0,t)$ is not anymore a sine function. We need therefore a more detailed analysis which is given in the next section.

4.2 Analysis for the heat equation

If the Fourier analysis were relevant for the heat equation, then introducing a pure sine frequency in the algorithm would give a pure sine frequency at any iteration $k > 0$, which is not the case as we saw in the previous subsection. A better understanding of the behavior of the pure sine frequency can be obtained from the following theorem, proved in [3].

Theorem 3 *Let $T = +\infty$ and $L = +\infty$. If the Schwarz Waveform Relaxation algorithm* (2) *is initialized with the pure sine frequency $e_2^0(\delta, t) = \sin(\lambda t)$, then the error is given by*

$$e_1^k(0,t) = |\rho(\lambda)|^{2k-1} \sin\left(\lambda t - (2k-1)\delta\sqrt{\frac{\lambda}{2\nu}}\right) + z_2\left((2k-1)\frac{\delta}{\sqrt{\nu}}, t; \lambda\right),$$

where z_2 satisfies for large frequency λ

$$z_2\left((2k-1)\frac{\delta}{\sqrt{\nu}}, t; \lambda\right) = \frac{1}{\lambda} K\left((2k-1)\frac{\delta}{\sqrt{\nu}}, t\right) + O\left(\frac{1}{\lambda^3}\right),$$

and for large iteration k

$$\left\| z_2\left((2k+1)\frac{\delta}{\sqrt{\nu}}, \cdot; \lambda\right)\right\|_{L^\infty(0,+\infty)} \sim \left(\frac{2k-1}{2k+1}\right)^2 \left\| z_2\left((2k-1)\frac{\delta}{\sqrt{\nu}}, \cdot; \lambda\right)\right\|_{L^\infty(0,+\infty)}.$$

An analogous result also holds for e_2^k.

This theorem states that if you introduce a pure sine frequency as the initial guess, then along the iterations the error becomes a sine which is contracted by ρ but which is also translated. In addition the sine is distorted by a term proportional to the heat kernel K.

5 Numerical results

In this section we illustrate the previous results with numerical experiments. An implicit scheme in time is used to discretize the heat equation. The spatial and time discretization parameters are $\Delta x = \Delta t = \frac{2}{2001}$ and the overlap is $\delta = 5\Delta x$. We will consider two values for ν so that we will obtain the different behaviors described in the previous sections.

We first consider the value $\nu = 1000$ which corresponds to a small spatial domain, or large time. The initial guess is the pure sine $e_2^0(\delta, t) = \sin(25t)$. In Figure 4 left, we show the error at $x = \delta$ as a function of t at iterations $k = 0$ and $k = 100$. We see that the sine is exactly contracted, which we can understand using Figure 4 right: ν is so large that the error is linear and the convergence is dictated by the maximum

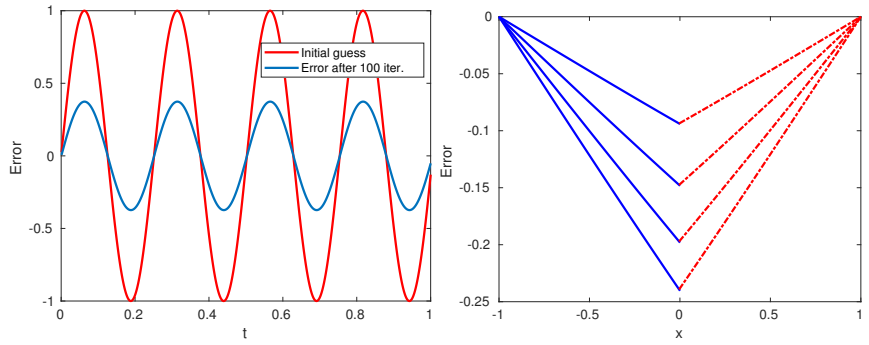


Fig. 4 $\nu = 1000$. On the left, the error $e_1^k(0,t)$ as a function of t. On the right, the error $e_1^k(x, t = 0.9965)$ as a function of x at iterations 1, 20, 50 and 100.

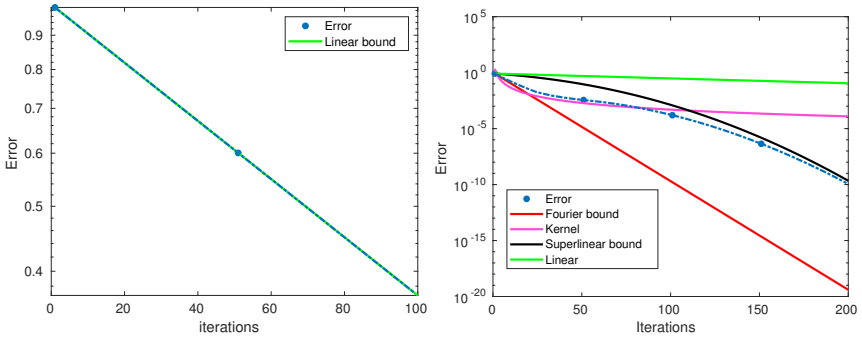


Fig. 5 Error $\|e_1^k(0,\cdot)\|_\infty$ as function of the Schwarz Waveform Relaxation iterations k. On the left $\nu = 1000$, on the right $\nu = 0.05$.

principle described in Section 2. This result is confirmed in Figure 5, left, where we show the norm of the error versus the iterations: we exactly obtain the linear bound described in Section 2.

We then consider the value $\nu = 0.05$ which corresponds to a large spatial domain, or small time interval. The initial guess is the pure sine $e_2^0(\delta,t) = \sin(50t)$.

In Figure 6 we show the solution as a function of x at $t = 0.1$. We see that now the boundary conditions at $x = \pm 1$ do not influence the solution and the behavior is not linear anymore. In Figure 7 we show the error $e_1^k(\delta,t)$ as a function of t for iterations $k = 10$, $k = 30$ and $k = 45$. We see that the initial guess $\sin(50t)$ is not only contracted, it is also translated and transformed by the heat kernel. In Figure 5 right, we show the error as a function of the iterations. The convergence is first guided by the Fourier convergence factor. For later iterations however, the heat kernel we observed in Figure 7 becomes dominant for the convergence mechanism of the Schwarz Waveform Relaxation algorithm. Then the heat kernel leaves the time domain and the superlinear regime dominates.

We have thus shown that for time dependent problems, Fourier analysis techniques can be applied to study Schwarz Waveform Relaxation algorithms, but care must be

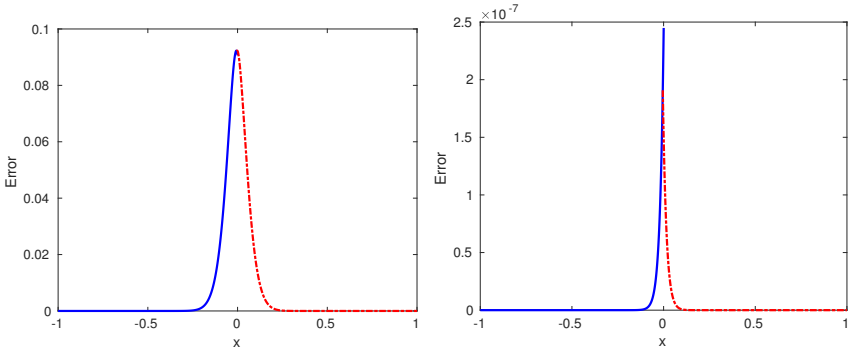


Fig. 6 $\nu = 0.05$. Error $e_1^k(x, t = 0.1)$ as a function of x at iterations 10 and 50.

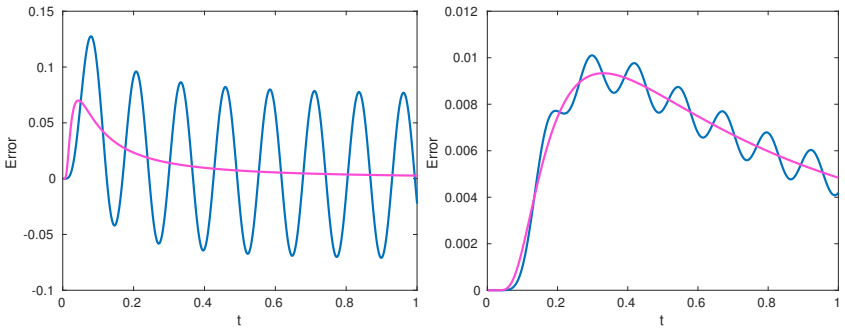


Fig. 7 $\nu = 0.05$. Error $e_1^k(0, t)$ as a function of t from left to right at iterations $k = 10$ and $k = 30$. In magenta the heat kernel term $\frac{1}{\lambda} K\left((2k-1)\frac{\delta}{\sqrt{\nu}}, t\right)$.

taken due to the evolution nature of the problem: Fourier modes initially still contract for diffusion problems away from the initial conditions as expected, but eventually heat kernel components dominate and change the convergence behavior.

References

1. Gander, M. J. Optimized schwarz methods. *SIAM Journal on Numerical Analysis* **44**(2), 699–731 (2006).
2. Gander, M. J. and Halpern, L. Méthodes de relaxation d'ondes (SWR) pour l'équation de la chaleur en dimension 1. *C. R. Math. Acad. Sci. Paris* **336**(6), 519–524 (2003).
3. Gander, M. J. and Martin, V. Why fourier mode analysis in time is different when studying schwarz waveform relaxation. *Accepted for publication in Journal of Computational Physics* (2023).
4. Gander, M. J. and Stuart, A. M. Space-time continuous analysis of waveform relaxation for the heat equation. *SIAM J. Sci. Comput.* **19**(6), 2014–2031 (1998).
5. Giladi, E. and Keller, H. B. Space-time domain decomposition for parabolic problems. *Numer. Math.* **93**(2), 279–313 (2002). URL https://doi.org/10.1007/s002110100345.
6. Japhet, C., Nataf, F., and cois Rogier, F. The optimized order 2 method: Application to convection–diffusion problems. *Future Generation Computer Systems* **18**(1), 17–30 (2001). I. High Performance Numerical Methods and Applications. II. Performance Data Mining: Automated Diagnosis, Adaption, and Optimization.

Cyclic and Chaotic Examples in Schwarz-Preconditioned Newton Methods

Conor McCoid and Martin J. Gander

1 Introduction

In [8] we showed the limits of using Newton's method to accelerate domain decomposition methods in 1D. Methods such as ASPIN [1], RASPEN [4], and MSPIN [7] that use additive (A), restricted additive (RA), and multiplicative (M) Schwarz (S) methods, respectively, to precondition (P) exact (E) or inexact (I) Newton's method (N) cannot always be relied upon to converge, though few examples of divergent, cyclic or chaotic behaviour have been presented to the community. The goal of this paper is to expand the space of counterexamples, showing the extent to which chaotic and cycling behaviour can occur in these methods and that these *SP*N methods are not robust to the pitfalls of Newton's method.

We begin by defining the particular method considered, the alternating Schwarz-preconditioned Newton's method (AltSPN) [8] in a general dimension:

$$(1)\ \begin{cases} F(x,u_1,Du_1,D^2u_1)=0,\ x\in\Omega_1,\\ u_1(\partial\Omega)=h,\\ u_1(\Gamma_1)=\gamma_n, \end{cases} \qquad (2)\ \begin{cases} F(x,u_2,Du_2,D^2u_2)=0,\ x\in\Omega_2,\\ u_2(\partial\Omega)=h,\\ u_2(\Gamma_2)=u_1(\Gamma_2), \end{cases}$$

$$(3)\ \begin{cases} J(u_1)\cdot g_1=0,\\ g_1(\partial\Omega)=0,\\ g_1(\Gamma_1)=I, \end{cases} \qquad (4)\ \begin{cases} J(u_2)\cdot g_2=0,\\ g_2(\partial\Omega)=0,\\ g_2(\Gamma_2)=g_1(\Gamma_2), \end{cases}$$

$$(5)\ \gamma_{n+1}=\gamma_n-(G'(\gamma_n)-I)^{-1}(G(\gamma_n)-\gamma_n)=\gamma_n-(g_2(\Gamma_1)-I)^{-1}(u_2(\Gamma_1)-\gamma_n),$$

where Du and D^2u represent the partial derivatives of $u(x)$ of orders 1 and 2, respectively, $J(u)$ is the Jacobian of F evaluated for the function $u(x)$, Γ_1 is the boundary

Conor McCoid
Université Laval, e-mail: conor.mccoid.1@ulaval.ca

Martin J. Gander
Université de Genève, e-mail: martin.gander@unige.ch

of Ω_1 that lies in Ω_2, and Γ_2 the same for Ω_2. Steps (1) and (3) are solved in the first subdomain Ω_1 while steps (2) and (4) are solved in the second, Ω_2.

The operators g_1 and g_2 are the derivatives of u_1 and u_2, respectively, with respect to γ. Since at $\partial\Omega$ u_1 does not depend on γ, the derivative there is zero. Likewise, since $u_1(\Gamma_1) = \gamma$, $g_1(\Gamma_1)$ is the identity operator. In a discrete setting they are represented by matrices.

The function $G(\gamma) := u_2(\Gamma_1)$ is the fixed point iteration of alternating Schwarz. It represents one iteration of alternating Schwarz using γ as the boundary value of $u_1(\Gamma_1)$. Step (5) applies Newton's method to $G(\gamma)$, thereby accelerating the convergence of the fixed point iteration.

Note that AltSPN possesses identical convergence behaviour to RASPEN, and in fact one may consider RASPEN to be the "gluing together" of two instances of AltSPN, one starting from the first subdomain and another starting from the second, see Section 2 of [4]. See also [2] for substructuring of RASPEN akin to that done here. ASPIN then behaves similarly, though without the restriction in the overlap and using inexact Newton in place of exact Newton [6]. Multiplicative Schwarz is identical to alternating Schwarz in the linear case [5], and so these counterexamples are of interest for MSPIN as well.

As explained in [8] Newton's method displays chaotic or cycling behaviour if it displays both oscillatory convergence and oscillatory divergence in a given domain. In 1D this was found to happen when the fixed point iteration to be accelerated ran parallel to the functions

$$g_C(x) = C\,\mathrm{sign}(x-x^*)\sqrt{|x-x^*|}+x,$$

where $g(x^*) = x^*$ and $C \in \mathbb{R}$ [8]. In higher dimensions the class of functions indicating this behaviour greatly increases and becomes difficult to summarize, but notably if one applies the square root operator to each coordinate of $|x-x^*|$ then the higher dimensional equivalent of $g_C(x)$ remains an indicator of this behaviour.

Newton's method in any dimension generally requires globalization techniques, such as line searches or trust regions, to guarantee convergence [3, Ch. 6]. In [8] we developed a Newton-like method in 1D with guaranteed convergence, with assumptions on the function usually satsified by SP*N methods. The counterexamples presented here show a need for a generalization of this method into 2D. However, it is not clear how to perform this generalization. We refer to [8] for further discussion.

2 Cycling counterexamples in 1D

We seek a larger space of examples where cycling occurs for AltSPN. To do so, we employ optimization techniques. First we must find a functional that takes a nonlinearity and returns a measure of the chaos that results from applying AltSPN.

We consider the set of problems

$$u''(x) + f(u(x)) = 0, \quad x \in (-1,1) \tag{1}$$

with homogeneous Dirichlet boundary conditions. The function $f(x)$ satsifies $f(0) = 0$ so that $u(x) = 0$ is the solution to the ODE.

The counterexample presented in [8],

$$u''(x) - \sin(\mu u(x)) = 0, \quad u(\pm 1) = 0,$$

makes use of the antisymmetry in $G(\gamma)$ to achieve its cycles. We wish the same for all nonlinearities $f(x)$ in our space of counterexamples.

Proposition 1 *If $f(x)$ is antisymmetric then $G(\gamma)$ is antisymmetric.*

Proof Suppose $\hat{u}_1(x)$ solves step (1) of alternating Schwarz with $u_1(\beta) = \gamma$. Then

$$-\hat{u}_1'' + f(-\hat{u}_1) = f(\hat{u}_1) - f(\hat{u}_1) = 0.$$

Thus, $-\hat{u}_1(x)$ solves step (1) of alternating Schwarz with $u_1(\beta) = -\gamma$.

By the same logic, if $\hat{u}_2(x)$ solves step (2) of alternating Schwarz with $u_2(\alpha) = u_1(\alpha)$ then $-\hat{u}_2(x)$ solves step (2) with $u_2(\alpha) = -u_1(\alpha)$. Thus, $-\hat{u}_2(\beta) = -G(\gamma) = G(-\gamma)$. □

We therefore restrict our search to nonlinearities which are antisymmetric. It is then sufficient for the Newton-Raphson acceleration, represented by $G_{NR}(\gamma)$, to cross the line $y = -\gamma$ for cycles to exist. In the steps of AltSPN, $G_{NR}(\gamma_n)$ is equal to γ_{n+1} found in step (5).

Given that $f(x)$ is antisymmetric it can be decomposed into a Fourier series consisting solely of sinusoids, which may be truncated for our purposes:

$$f(x) = \sum_{k=1}^{N} c_k \sin(\pi k x). \tag{2}$$

Thus, the functional to optimize takes a set $\{c_k\}_{k=1}^N \in \mathbb{R}^N$, passes it through $f(x) \in C(-1,1)$ and $G(\gamma) \in C(\mathbb{R})$ to arrive at a measure for the chaos of the system. We will represent this functional as $L : \mathbb{R}^N \to \mathbb{R}$. There are many ways to define $L(\{c_k\})$, but we shall use

$$L(\{c_k\}) := \|G_{NR}(\gamma) + \gamma\|. \tag{3}$$

Thus, $L(\{c_k\}) = 0$ if and only if $G_{NR}(\gamma) = -\gamma$, and AltSPN cycles between γ and $-\gamma$ for all values of γ.

It is well-known that there always exists a region around the root of a function where Newton-Raphson converges, assuming continuity of $G''(\gamma)$ and $G'(\gamma) \neq 1$. We expect then to find only local minimizers of $L(\{c_k\})$.

We use a gradient descent with line search to optimize the functional L, with restrictions $\gamma \in [-2,2]$ and $\{c_k\}_{k=1}^5 \in \mathbb{R}^5$, with starting condition $c_1 = 1$, $c_i = 0$ for $i = 2,\ldots,5$. The gradient is computed through a centered finite difference stencil in $\mathbb{R}^5$.

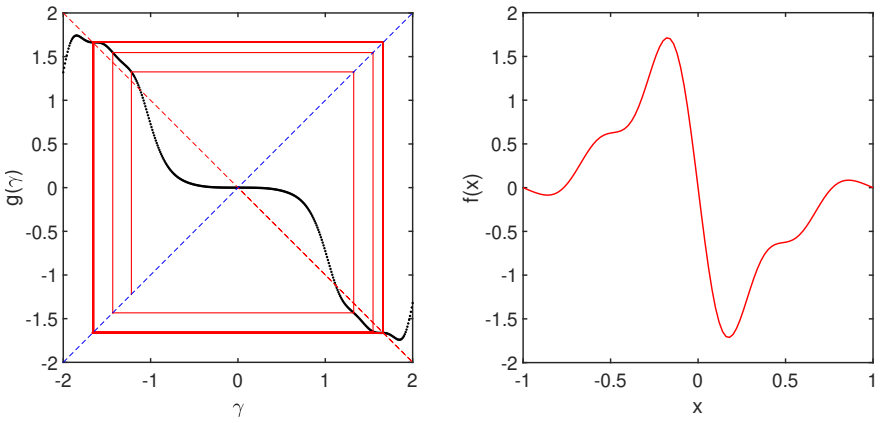


Fig. 1 A counterexample found through optimizing the functional L, equation (3). (Left) AltSPN falls into a stable 2-cycle, as represented by the path of the red lines; the cycle's basin of attraction is most values within $[-2,-1)\cup(1,2]$. (Right) The function $f(x)$ for this counterexample, the sum of five sinusoids.

The coefficients c_k have greater effect with higher k, so that $\partial L/\partial c_5 > \partial L/\partial c_1$. As such, the stencil's step size diminishes with k. The line search takes search direction $\mathbf{p} = \nabla L$ and tests $\{c_k\} + h\mathbf{p}$ for $h = 1$. If $L(\{c_k\} + h\mathbf{p}) > L(\{c_k\})$, then the search is repeated with $h \to 0.5h$, until $L(\{c_k\} + h\mathbf{p}) \leq L(\{c_k\})$.

The domain is discretized with 100 equally spaced points with a 3-point finite difference Laplacian. There is an overlap of 0.4 between the domains, which are symmetric about $x = 0$. This choice of overlap is taken from [8]. We find an exceptional counterexample using this methodology, as presented in Figure 1.

3 Chaotic counterexamples in 2D

We now seek counterexamples in 2D. That is, we seek $f : \mathbb{R} \to \mathbb{R}$ such that using AltSPN to solve

$$\mathscr{L}u(x,y) + f(u(x,y)) = u_{xx}(x,y) + u_{yy}(x,y) + f(u(x,y)) = 0, \quad x,y \in (-1,1)$$

with homogeneous Dirichlet boundary conditions results in cycling behaviour. The two domains are split along the x-axis, so that the first domain is $x \in (-1,\alpha)$, $y \in (-1,1)$ and the second is $x \in (-\alpha,1)$, $y \in (-1,1)$.

The problem with finding cyclic counterexamples in 2D is an issue of dimensionality. If a given direction gives stable cycles, there is no guarantee that every orthogonal direction to these cycles is sufficiently stable to allow these cycles to continue indefinitely. That is, if any point on the cycle is a saddle point, then numerical error can easily eject the iterates from the path of the cycles. The criteria

for numerical cycles is then necessarily stronger than in 1D: Not only must a cycle exist and be stable, there must exist a region around the cycle that is also stable.

As before, $f(x)$ is chosen to be antisymmetric so that $G(\gamma)$ is antisymmetric. Proposition 1 applies to the higher dimensional case by replacing all relevant scalar objects (γ, $G(\gamma)$) with their corresponding vectors (γ, $G(\gamma)$). The decomposition of $f(x)$ into sinusoids and the definition of the functional $L(\{c_k\})$ remain unchanged.

If γ is the discretization of a sinusoid then $G(\gamma) = c\gamma$, where $c \in \mathbb{R}$, up to numerical error. This can be seen by transforming the solution into a Fourier series in the y variable. Suppose

$$\begin{cases} \frac{\partial^2}{\partial x^2} u_1(x,y) + \frac{\partial^2}{\partial y^2} u_1(x,y) + f(u_1(x,y)) = 0, \quad x \in (-1,\alpha), \quad y \in (-1,1), \\ u_1(-1,y) = u_1(x,\pm 1) = 0, \\ u_1(\alpha,y) = C\sin(\pi m y), \end{cases}$$

where $f(x)$ is antisymmetric and can therefore be expressed in the form of equation (2). Use as an ansatz $u_1(x,y) = g(x)\sin(\pi m y)$. Take the Fourier transform of the equation:

$$\begin{aligned} 0 =& \int_{-1}^{1} (g''(x) - m^2\pi^2 g(x))\sin(\pi m y)\sin(\pi k y) + f(g(x)\sin(\pi m y))\sin(\pi k y) dy \\ =& (g''(x) - m^2\pi^2 g(x))\delta_{m,k} C_m + \int_{-1}^{1} \sum_{j=1}^{N} c_j \sin(\pi j g(x)\sin(\pi m y))\sin(\pi k y) dy \\ =& \delta_{m,k}\mathscr{L} g(x) + \sum_{j=1}^{N} c_j \int_{-1}^{1} \sum_{n=0}^{\infty} \frac{(\pi j g(x))^{2n+1}}{(2n+1)!} \sin(\pi m y)^{2n+1} \sin(\pi k y) dy \\ =& \delta_{m,k}\mathscr{L} g(x) + \sum_{j=1}^{N} c_j \sum_{n=0}^{\infty} \frac{(\pi j g(x))^{2n+1}}{(2n+1)!} S(2n+1), \end{aligned}$$

where $S(n)$ is the integral of $\sin(\pi m y)^n \sin(\pi k y)$ over $(-1,1)$. We can find a recursive formula for $S(2n+1)$:

$$\begin{aligned} S(2n+1) =& -\frac{1}{\pi k}\cos(\pi k y)\sin(\pi m y)^{2n+1}\Big|_{-1}^{1} \\ &+ \int_{-1}^{1} \frac{m}{k}(2n+1)\cos(\pi k y)\cos(\pi m y)\sin(\pi m y)^{2n} dy \\ =& (2n+1)\frac{m}{k}\Big(-\frac{1}{\pi k}\cos(\pi m y)\sin(\pi m y)^{2n}\sin(\pi k y)\Big|_{-1}^{1} \\ &+ \frac{m}{k} 2n \int_{-1}^{1} \sin(\pi k y)\left(-\sin(\pi m y)^{2n+1} + \cos(m\pi y)^2 \sin(\pi m y)^{2n-1}\right) dy \Big) \\ =& (2n+1)(2n)\frac{m^2}{k^2}\int_{-1}^{1} \sin(\pi m y)^{2n-1}\sin(\pi k y) - 2\sin(\pi m y)^{2n+1}\sin(\pi k y) dy \end{aligned}$$

$$S(2n+1) = (2n+1)(2n)\frac{m^2}{k^2}\left(S(2n-1) - 2S(2n+1)\right)$$
$$= \frac{(2n+1)(2n)\frac{m^2}{k^2}}{1+2(2n+1)(2n)\frac{m^2}{k^2}} S(2n-1) = C(m,k,n)S(1),$$

where $C(m,k,n)$ is a constant that depends on m, k and n. The value of $S(1)$ is zero unless $k = m$. This proves there exists a function $\tilde{f}_m(x)$ such that

$$\mathscr{L}g(x) + \tilde{f}_m(g(x)) = 0.$$

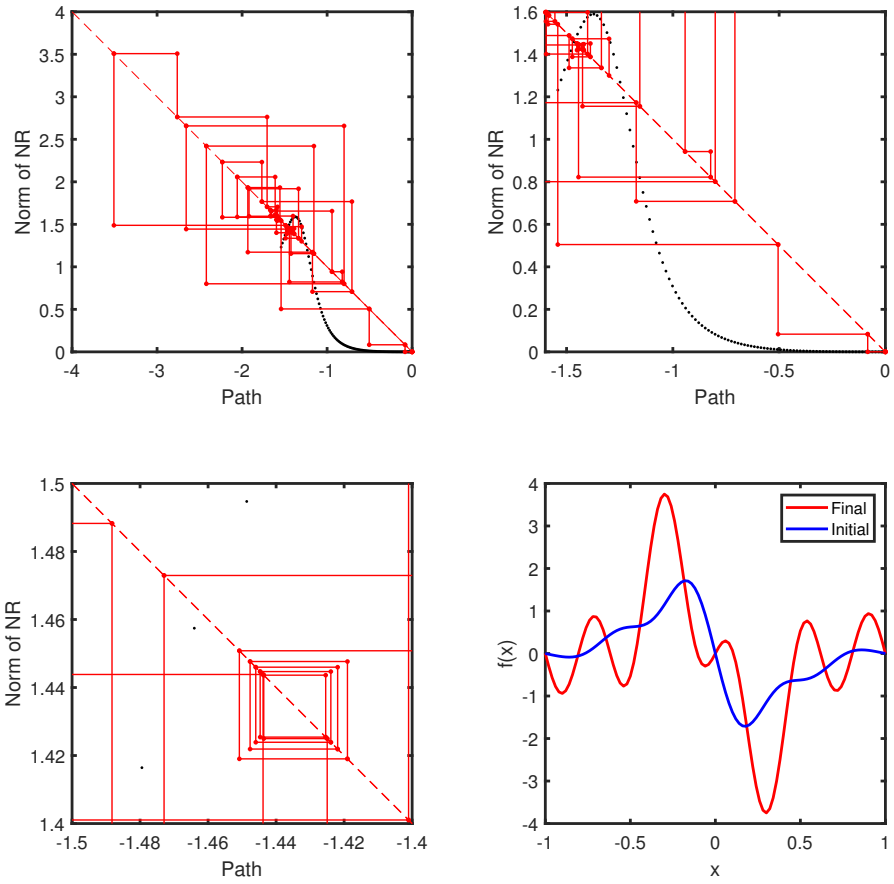


Fig. 2 A chaotic AltSPN sequence in 2D, at three resolutions (top row, bottom left). The solid red lines indicate the path of the fixed point iteration $G_{NR}(\gamma)$. The nonlinearity (bottom right) is found through optimization on the functional L, equation (3). The AltSPN sequence is seeded using a sine wave $\sin(\pi y)$ as boundary condition on the first subdomain, but quickly diverges from this pathway.

Thus, if $g(x)$ satisfies this ODE with boundary conditions $g(-1) = 0$ and $g(\alpha) = C$ then $u_1(x,y)$ solves the PDE. Since $u_1(x,y)$ in this form satisfies the boundary conditions it is the solution to this step of alternating Schwarz. The solution on the second domain, $u_2(x,y)$, then also has the same form by a symmetry argument, and its value at $x = -\alpha$ is a sinusoid of the same period. Therefore, $G(\gamma) = c\gamma$.

As a direct consequence of this, starting with any single sine wave as boundary conditions provides a single parameter pathway for the function $G : \mathbb{R}^N \to \mathbb{R}^N$. We take advantage of this fact and use $c\sin(\pi y)$ as the boundary condition for the first subdomain of AltSPN, varying c between -0.5 and 0. To seed the optimization we use the previously obtained counterexample nonlinearity $f(x)$ from the 1D case. The same optimization method is used. 20 equally spaced points are used in each direction with a 5-point stencil Laplacian.

The resulting nonlinearity does not admit a stable cycle but highlights the chaos that can result from using AltSPN. Figure 2 gives this nonlinearity and an example of a sequence generated by AltSPN. The sequence begins on the sine wave path described previously. It then approaches a nearby stable cycle, having departed from the strict sine wave path to one that closely resembles sine waves (bottom left of the figure). However, the cycle is not numerically stable and is ultimately abandoned.

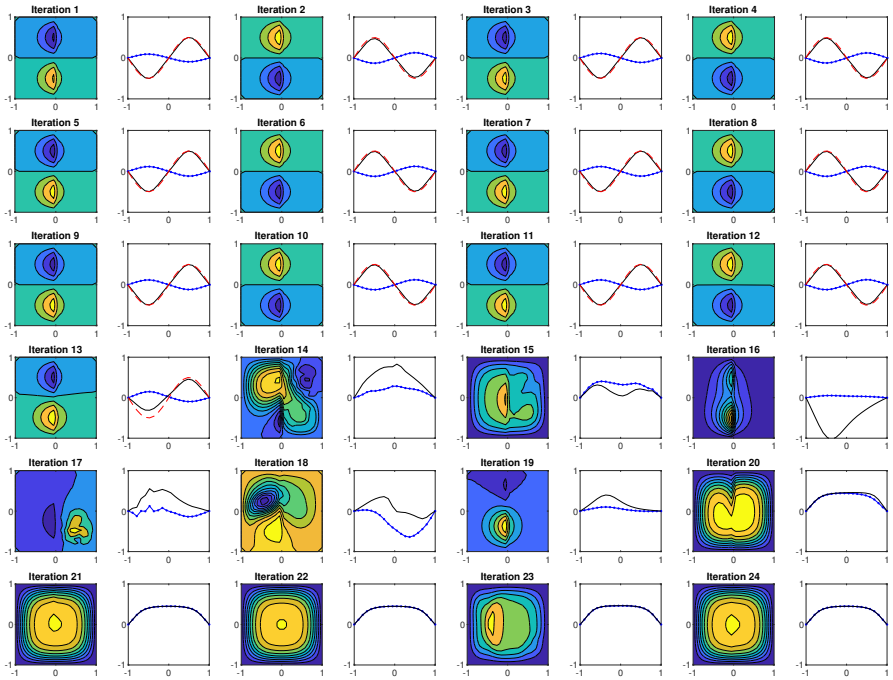


Fig. 3 The first 24 iterations of the chaotic AltSPN sequence. The left figure of each iteration shows the overall solution at that step combined from the two subdomains, and the right of each shows the resulting $G(\gamma)$ in blue and the AltSPN result in black. For comparison, the sine wave $0.5\sin(\pi y)$ is plotted in red for the cycling regime, with sign that alternates with each iteration.

As it leaves this pathway it descends into a choatic regime (top left). It eventually ejects onto a convergent pathway (top right). It is possible these cycles exist on saddlepoints, stable along some pathways but unstable along others. A small numerical error will then shunt the AltSPN sequence away from the cycle, either to a divergent (top left) or convergent (top right) pathway.

Figure 3 provides snapshots of the solution at each of the iterates. The first 24 iterations of AltSPN are shown, giving the nearly cycling regime (iterations 1 to 12) and part of the chaotic regime (13 to 24). The cycling regime is very nearly a cycle of sine waves, as seen by the comparable sine waves in red. A small change from these sine waves in iteration 13 causes chaos to take over until relative stability in iterations 20 through 24. From there, AltSPN eventually converges.

These results show that acceleration cannot be used without consequences for all Schwarz algorithms. As with standard Newton-Raphson, there exist problems for which the sequence diverges, cycles or behaves chaotically.

References

1. Cai, X.-C. and Keyes, D. E. Nonlinearly preconditioned inexact Newton algorithms. *SIAM Journal on Scientific Computing* **24**(1), 183–200 (2002).
2. Chaouqui, F., Gander, M. J., Kumbhar, P. M., and Vanzan, T. Linear and nonlinear substructured restricted additive schwarz iterations and preconditioning. *Numerical Algorithms* **91**(1), 81–107 (2022).
3. Dennis Jr, J. E. and Schnabel, R. B. *Numerical methods for unconstrained optimization and nonlinear equations*. SIAM (1996).
4. Dolean, V., Gander, M. J., Kheriji, W., Kwok, F., and Masson, R. Nonlinear preconditioning: How to use a nonlinear Schwarz method to precondition Newton's method. *SIAM Journal on Scientific Computing* **38**, 3357–3380 (2016).
5. Gander, M. J. Schwarz methods over the course of time. *Electronic transactions on numerical analysis* **31**, 228–255 (2008).
6. Gander, M. J. On the origins of linear and non-linear preconditioning. In: *Domain Decomposition Methods in Science and Engineering XXIII*, 153–161. Springer (2017).
7. Liu, L. and Keyes, D. E. Field-split preconditioned inexact Newton algorithms. *SIAM Journal on Scientific Computing* **37**, A1388–A1409 (2015).
8. McCoid, C. and Gander, M. J. Cycles in Newton-Raphson preconditioned by Schwarz (ASPIN and its cousins). In: *Domain Decomposition Methods in Science and Engineering XXVI*, *Lecture Notes in Computational Science and Engineering*, vol. 145. Springer Cham (2023).

Global-Local Forward Models within Bayesian Inversion for Large Strain Fracturing in Porous Media

Nima Noii, Thomas Wick, and Amirreza Khodadadian

1 Introduction

Phase-field fracture models are employed to capture failure and cracks in structures, alloys, and poroelastic media. The coupled model is based on solving the elasticity equation and an Allen-Cahn-type phase-field equation. In hydraulic fracture, a Darcy-type equation is solved to capture the pressure profile. Solving this coupled system of equations is computationally expensive. Indeed, to provide an accurate estimation (compared to the measurement) a very fine mesh profile is required. Of course, the time-dependent and nonlinear nature of the problem gives rise to more complexity. Another challenge is related to the computational, mechanical, and geomechanical material parameters. They have an essential effect on the simulations; however, many of them can not be estimated experimentally.

In [11], we used the Bayesian inversion to identify the parameters based on hydraulic fractures of porous media. A fracture response is realized through a phase-field equation [2] (based on the seminal work [3]). But that work is limited to small deformations. In the current study, we extend [11] towards a *large strain* formulation [1, 7].

In consequence, the main objective is to utilize non-intrusive global-local models [4] that are originally based on non-overlapping domain decomposition [12] to significantly reduce the computational cost in Bayesian inversion. In extension to our prior work, we introduce an adoption of the hydraulic phase-field fracture formulation of a material that undergoes large deformation in poroelastic media. Finally, ensemble Kalman filters are employed for the proposal adaption in Bayesian inversion to identify the mechanical material parameters once the multiscale approach is used to solve the forward model.

2 Framework for failure mechanics in hydraulic fracture

Let us assume $\mathcal{B} \subset \mathbb{R}^{\delta}$ is the solid computational domain (here $\delta = 2$) with its surface boundary $\partial\mathcal{B}$ and time $t \in \mathcal{T} = [0, T]$. The given boundary-value problem (BVP)

Thomas Wick, Amirreza Khodadadian
Leibniz University Hannover, Institute of Applied Mathematics and Cluster of Excellence PhoenixD, Welfengarten 1, 30167 Hannover, Germany,
e-mail: {thomas.wick,khodadadian}@ifam.uni-hannover.de
Nima Noii
Leibniz University Hannover, Institute of Continuum Mechanics, An der Universität 1, 30823 Garbsen, Germany e-mail: noii@ikm.uni-hannover.de

is a coupled multi-field system for the fluid-saturated porous media of the fracturing material. Since we are dealing in large strain setting, it is required to define the mapping between the referential position $\mathbf{X}$ towards spatial description $\mathbf{x}$ based on the motion $\boldsymbol{\varphi}$ of point P at time t, see Figure 1. The media can be formulated based on a coupled three-field system. At material points $\mathbf{x} \in \mathcal{B}$ and time $t \in \mathcal{T}$, the BVP solution indicates the deformation field $\boldsymbol{\varphi}(\mathbf{x}, t)$ of the solid, the fluid pressure field $p(\mathbf{x}, t)$, and the phase-field fracture variable d can be represented by

$$\boldsymbol{\varphi}\colon \begin{cases} \mathcal{B}\times\mathcal{T} \to \mathcal{R}^{\delta} \\ (\mathbf{X}, t) \mapsto \mathbf{x} = \boldsymbol{\varphi}(\mathbf{X}, t) \end{cases} \quad p\colon \begin{cases} \mathcal{B}\times\mathcal{T} \to \mathcal{R} \\ (\mathbf{X}, t) \mapsto p(\mathbf{X}, t) \end{cases} \quad d\colon \begin{cases} \mathcal{B}\times\mathcal{T} \to [0,1] \\ (\mathbf{X}, t) \mapsto d(\mathbf{X}, t) \end{cases} \tag{1}$$

Here, $d(\mathbf{x}, t) = 0$ and $d(\mathbf{x}, t) = 1$ are referred to as the unfractured and completely fractured parts of the material, respectively. The coupled BVP is formulated through three specific primary fields to illustrate the hydro-poro-elasticity of fluid-saturated porous media by

$$\text{Global Primary Fields}\colon \mathfrak{U} := \{\boldsymbol{\varphi}, p, d\}. \tag{2}$$

2.1 Elastic contribution

The elastic density function is formulated through a Neo-Hookean strain energy function for a compressible isotropic elastic solid

$$W_{\text{elas}}(\mathbf{F}, d) = g(d)\, \psi_{\text{elas}}(\mathbf{F}) \quad \text{with} \quad \psi_{\text{elas}}(\mathbf{F}) = \frac{\mu}{2}\left[(\mathbf{F}:\mathbf{F} - 3) + \frac{2}{\beta}(J^{-\beta} - 1)\right], \tag{3}$$

such that the shear modulus μ and the parameter $\beta := \beta(\nu) = 2\nu/(1-2\nu)$ with the Poisson number $\nu < 0.5$ are used. Here, the material deformation gradient of the solid denoted by $\mathbf{F}(\mathbf{X}) := \nabla\boldsymbol{\varphi}(\mathbf{X}, t) = \text{Grad}\boldsymbol{\varphi}$ with the Jacobian $J := \det[\mathbf{F}] > 0$ augmented with the symmetric right Cauchy-Green tensor $\mathbf{C} = \mathbf{F}^T\mathbf{F}$ is used; for details the reader is referred to [1, 11]. We note that the quadratic function $g(d) = (1-d)^2 + \kappa$ is denoted as a degradation function, with $\kappa \approx 10^{-8}$ that is chosen as a sufficiently small quantity. According to the classical Terzaghi theorem, the constitutive modeling results in the additive split of the stress tensor $\mathbf{P}$ to effective mechanical contribution and fluid part as

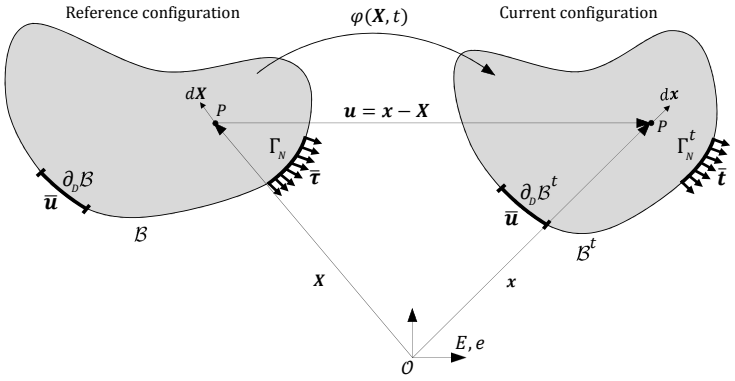


Fig. 1 Setup of the notation for the configuration and motion of the continuum body $\boldsymbol{\varphi}(\mathbf{X}, t)$. The initial position $\mathbf{X}$ in the undeformed configuration $\mathcal{B}$ toward the current position $\mathbf{x}$ in the spatial configuration $\mathcal{B}^t$ for the solid material undergoing finite strain.

$$\mathbf{P}(\mathbf{F}, p, d) := \frac{\partial W_{\text{elas}}}{\partial \mathbf{F}} = g(d)\mathbf{P}_{eff}(\mathbf{F}) - BpJ\mathbf{F}^{-T} \quad \text{with} \quad \mathbf{P}_{eff} = \mu\left[\mathbf{F} - J^{-\beta}\mathbf{F}^{-T}\right]. \tag{4}$$

Here, the first Piola-Kirchoff stress tensor $\mathbf{P}$ is derived from the first-order derivative of the pseudo-energy density function W_{elas} given in (3). Thus, the balance of linear momentum for the multi-field system prescribed through body force $\overline{\mathbf{b}}$ reads

$$\boxed{\text{Div}\,\mathbf{P}(\mathbf{F}, p, d) + \overline{\mathbf{b}} = \boldsymbol{0}.} \tag{5}$$

2.1.1 Fluid contribution

The fluid volume flux vector $\mathcal{F}$ is described through the negative direction of the gradient of the fluid pressure ∇p and permeability based on Darcy-type fluid's

$$\mathcal{F} := -\mathbf{K}(\mathbf{F}, d)\ \nabla p. \tag{6}$$

Here, the second-order permeability tensor $\mathbf{K}(\mathbf{F}, d)$, following [7], is additively decomposed into the permeability tensor into a Darcy-type flow for the unfractured porous medium $\boldsymbol{K}_{Darcy}$ and Poiseuille-type flow in a completely fractured material $\boldsymbol{K}_{frac}$ by

$$\begin{aligned} \boldsymbol{K}(\mathbf{F}, d) &= \boldsymbol{K}_{\text{Darcy}}(\mathbf{F}) + d^{\zeta}\boldsymbol{K}_{\text{frac}}(\mathbf{F}), \\ \boldsymbol{K}_{\text{Darcy}}(\mathbf{F}) &= \tfrac{K}{\eta_F} J\mathbf{C}^{-1}, \\ \boldsymbol{K}_{\text{frac}}(\mathbf{F}) &= K_c\ \omega^2\ J\left[\mathbf{C}^{-1} - \mathbf{C}^{-1}\mathbf{N} \otimes \mathbf{C}^{-1}\mathbf{N}\right]. \end{aligned} \tag{7}$$

Here, K_D is the isotropic intrinsic permeability of the pore space, K_c is the spatial permeability in the fracture, η_F is the dynamic fluid viscosity, and $\zeta \geq 1$ is a permeability transition exponent. Following [7], the so-called crack aperture (or the crack opening deformation) defined through $\omega = (\lambda_{\perp} - 1)h_e$ in terms of the stretch orthogonal to the crack surface $\lambda_{\perp}^2 = \nabla d \cdot \nabla d / \nabla d \cdot \mathbf{C}^{-1} \cdot \nabla d$ and the characteristic element length h_e. Also, $\mathbf{N} = \nabla d / |\nabla d|$ denotes the outward unit normal to the fracture surface, h_e is the characteristic discretization size, and $\mathbf{I}$ is an identity tensor. Thus, following [7, 1], the fluid equation involve pressure files read

$$\boxed{\frac{\dot{p}}{M} + B\dot{J} - \bar{r}_F + \text{Div}[\mathcal{F}] = 0\ .} \tag{8}$$

2.1.2 Fracture contribution

The crack driving state function in the regularized sense conjugate to crack phase-field denoted as $D(\boldsymbol{\varphi}, d, \mathbf{x})$ for every point $\mathbf{x}$ in domain act as a driving force for the fracture evolution state reads

$$D(\boldsymbol{\varphi}, d, \mathbf{x}) := \frac{2l}{G_c}(1 - \kappa)\psi_{\text{elas}}(\mathbf{F}). \tag{9}$$

Here, G_c is the Griffith's critical elastic energy release rate, and $l = 2h_e$ is the regularization term. Following [6], the local evolution of the crack phase-field equation in the given domain $\mathcal{B}$ results in the third Euler-Lagrange differential system as

$$\boxed{(1 - d)\mathcal{H} - [d - l^2\Delta d] = \eta\dot{d} \quad in\ \mathcal{B},} \tag{D}$$

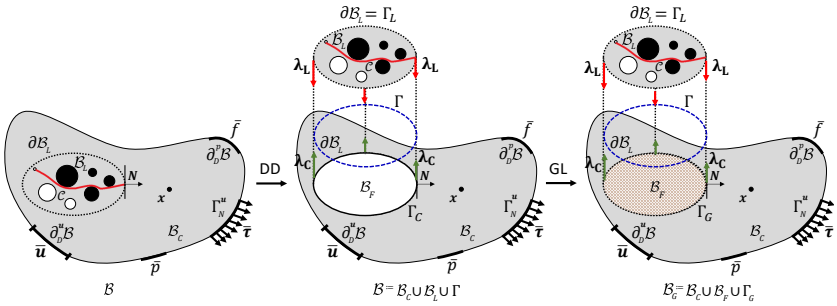


Fig. 2 Configuration and loading setup of the single-scale BVP (left). Middle/right: global-local configuration, by the fictitious domain $\mathcal{B}_F$ through filling the gap between $\mathcal{B}_C$ and $\mathcal{B}_L$ with a same constitutive modeling and discretization of $\mathcal{B}_C$ such that its unification is a so-called global domain $\mathcal{B}_G := \mathcal{B}_C \cup \Gamma_G \cup \mathcal{B}_F$.

augmented by the homogeneous Neumann boundary condition that is $\nabla d \cdot \mathbf{n} = 0$ on $\partial\mathcal{B}$, with the maximum absolute value for the crack driving state $\mathcal{H} = \max_{s\in[0,t]} D(\boldsymbol{\varphi}) \geq 0$ to avoid irreversibly. For different approach see [8]. Thus, following our recent work [11], the variational formulations for the three PDEs for the coupled poroelastic media of the fracturing material are

$$
\begin{aligned}
\mathcal{E}_{\varphi}(\mathfrak{U}, \delta\boldsymbol{\varphi}) &= \int_{\mathcal{B}} \Big[\mathbf{P} : \nabla\delta\boldsymbol{\varphi} - \bar{\mathbf{b}} \cdot \delta\boldsymbol{\varphi} \Big] dV - \int_{\partial_N \mathcal{B}} \bar{\boldsymbol{\tau}} \cdot \delta\boldsymbol{\varphi} \; dA = 0 \,, \\
\mathcal{E}_{p}(\mathfrak{U}, \delta p) &= \int_{\mathcal{B}} \Big[\Big(\frac{1}{M}(p - p_n) + B(J - J_n) - \Delta t \; \bar{r}_F \Big) \delta p + (\Delta t \; \mathbf{K} \; \nabla p) \cdot \nabla\delta p \Big] dV \\
&\quad + \int_{\partial_N \mathcal{B}} \bar{f} \; \delta p \; dA = 0 \,, \\
\mathcal{E}_{d}(\mathfrak{U}, \delta d) &= \int_{\mathcal{B}} \Big[\Big(2\psi_c \; d + 2(d-1)\mathcal{H} \Big) \delta d + 2\psi_c \; l^2 \; \nabla d \cdot \nabla\delta d \Big] dV = 0 \,.
\end{aligned}
\tag{10}
$$

This set of equation is now written in the abstract form through $\mathtt{SS}(\mathfrak{U})$.

3 Multiscale modeling via a non-intrusive global-local method

The previously introduced system of equations for single-scale analysis in (10) for the coupled problem of poroelasticity and fracture is further extended towards the global-local (GL) method now. Following [1, 5], the GL formulation is rooted in domain decomposition (e.g., [12]) by distinguishing the original domain into coarse and fine discretizations, see Figure 2. To couple the domains, namely global and local domains, we have introduced an additional auxiliary interface denoted as Γ between two disjoint domains in poroelastic media (see [1]), and thus corresponding unknown fields, see Figure 2. These additional fields are the interface deformation $\boldsymbol{\varphi}_\Gamma(\mathbf{x}, t)$ and pressure $p_\Gamma(\mathbf{x}, t)$ on auxiliary interface and their corresponding traction forces $\{\boldsymbol{\lambda}^{\boldsymbol{\varphi}}_L, \boldsymbol{\lambda}^{\boldsymbol{\varphi}}_C\}$ and $\{\lambda^{p}_L, \lambda^{p}_C\}$ that are introduced as Lagrange multipliers. These results in a set of coupling equations at the interface by

$$
\begin{cases}
\boldsymbol{\varphi}_L(\mathbf{X}, t) = \boldsymbol{\varphi}_\Gamma(\mathbf{X}, t) & \text{at } \mathbf{X} \in \Gamma_L, \\
\boldsymbol{\varphi}_G(\mathbf{X}, t) = \boldsymbol{\varphi}_\Gamma(\mathbf{X}, t) & \text{at } \mathbf{X} \in \Gamma_G, \\
\boldsymbol{\lambda}^{\boldsymbol{\varphi}}_L(\mathbf{X}, t) + \boldsymbol{\lambda}^{\boldsymbol{\varphi}}_C(\mathbf{X}, t) = \boldsymbol{0} & \text{at } \mathbf{X} \in \Gamma,
\end{cases}
\quad \text{and} \quad
\begin{cases}
p_L(\mathbf{X}, t) = p_\Gamma(\mathbf{X}, t) & \text{at } \mathbf{X} \in \Gamma_L, \\
p_G(\mathbf{X}, t) = p_\Gamma(\mathbf{X}, t) & \text{at } \mathbf{X} \in \Gamma_G, \\
\lambda^{p}_L(\mathbf{X}, t) + \lambda^{p}_C(\mathbf{X}, t) = 0 & \text{at } \mathbf{X} \in \Gamma.
\end{cases}
\tag{11}
$$

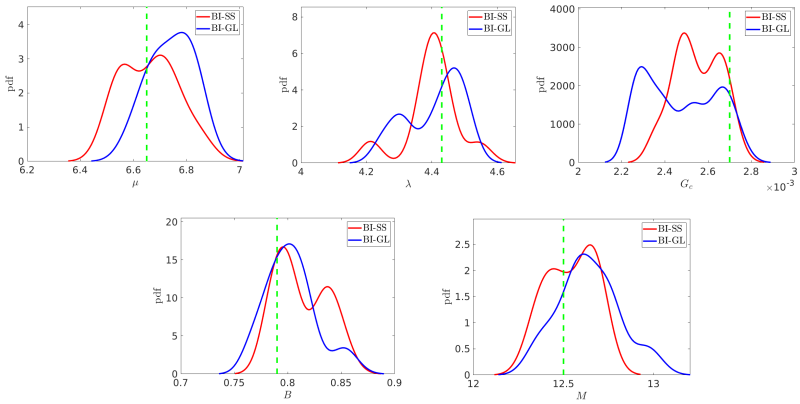


Fig. 3 The pdf of posterior density of the material parameters using the BI-GL and BI-SS approaches for fracture. The true values are shown with a dashed green line.

Now, the multi-physics problem for the global-local approach is described through eleven primary fields to characterize the hydro-poro-elasticity of fluid-saturated porous media at finite strains by

$$\text{Extended Primary Fields}: \mathfrak{P} := \{\boldsymbol{\varphi}_G, \boldsymbol{\varphi}_L, p_G, p_L, d_L, \boldsymbol{\lambda}_C^{\mathbf{u}}, \boldsymbol{\lambda}_L^{\mathbf{u}}, \lambda_C^p, \lambda_L^p, \mathbf{u}_\Gamma, p_\Gamma\} \ . \quad (12)$$

Herein, a global constitutive model behaves as a poroelastic response, abbreviated as E(elastic)-P(pressure), which is augmented with a *single local domain* and behaves as a poroelastic material with fracture response, abbreviated as E(elastic)-P(pressure)-D(damage). The resulting final algorithm is based on our prior work [1, 11].

4 Bayesian inversion for parameter estimation

In this study, we use MCMC (Markov chain Monte Carlo) techniques to identify the material parameters in the hydraulic porous medium phase-field fracture setting. The latter is solved with the previously described GL approach. In general, we can employ the following probabilistic model to update the available prior information according to the forward model (here considers the phase-filed fracture) and a reference observation (arising from measurement, or a synthetic observation). First, we introduce the following statistical model

$$\mathbb{M} = \mathcal{P}(\boldsymbol{x}, \chi) + \varepsilon. \quad (13)$$

Here $\mathbb{M}$ refers to the reference observation arising from the experimental data (a measured value) and $\mathcal{P}$ considers to the model response related to χ a set of d-dimensional material parameters. Furthermore, $\boldsymbol{x} \in \mathbb{R}^\delta$ and ε indicates the measurement error. It is assumed to have Gaussian independent and identically distributed error $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$, having the parameter σ^2. Since $\mathcal{P}$ in (13) is a model response which results in our computation, such that in our presented model can be approximated through

$$\texttt{signle-scale:} \quad \mathcal{P} \approx \mathcal{P}^{\mathrm{SS}} \quad \text{or} \quad \texttt{global-local:} \quad \mathcal{P} \approx \mathcal{P}^{\mathrm{GL}},$$

corresponds to equations (SS) and (GL), respectively. Thus, (13) becomes as

$$\mathcal{M} = \mathcal{P}^\bullet(\Theta) + \varepsilon, \quad \text{with} \quad \bullet \in \{\mathrm{SS},\mathrm{GL}\}. \quad (14)$$

Despite the simplicity of the Metropolis-Hastings algorithm, it is not suitable for complicated cases, specifically when several parameters should be estimated (multi-dimensional domains). In this study, we use MCMC with ensemble-Kalman filter, see for a detailed discussion [11]. The ensemble Kalman filter (EnKF) indicates the error covariance matrix by a large random ensemble of model observations. Here, to achieve a reliable estimation of posterior density, a Kalman gain is computed using the mean and the covariance of the prior density and the cross-covariance between material parameters and observations. Using an ensemble-Kalman filter, we adopt the proposal density with $\chi^{\star} = \chi^{j-1} + \Delta\chi$, where $\Delta\theta$ is the jump of Kalman-inspired proposal. Afterwards, we update the candidate via $\Delta\chi = \mathcal{K}\left(y^{j-1} + s^{j-1}\right)$. The Kalman gain is computed by $\mathcal{K} = C_{\theta M}\left(C_{MM} + \mathcal{R}\right)^{-1}$, where $C_{\theta M}$ is the covariance matrix between the unknowns and the model response, C_{MM} denotes the covariance matrix of the PDE-based model, and $\mathcal{R}$ is the measurement noise covariance matrix [13]. Moreover, y^{j-1} is the residual of candidates w.r.t the model and $s^{j-1} \sim \mathcal{N}(0, \mathcal{R})$ relates to the density of measurement. Denoting `obs` as an observation, $y^{j-1} = \texttt{obs} - f(\theta^{j-1})$. We refer the reader to [9] for more details and the codes.

Thus, we are now able to use Bayesian inversion to identify the fracking process using multiscale approach material parameters that cannot be measured with usual techniques.

5 Numerical example

In this section, we investigate a numerical test with the main goal that Bayesian inversion yields accurate parameter identifications at a cheap cost of the governing global-local phase-field solver. The mechanical and geomechanical descriptoion of the parameters is given in [10]. In the following, a BVP is applied to the square plate shown in Figure 4. The geometry and boundary conditions are from [1]. The single-scale (SS) model results considering the phase-field and pressure are given in Figure 5. Then, we employ our global-local approach, with findings shown in Figure 6. Figure 7 shows the load-displacement curve for both approachs, indicating the accuracy of the GL approach. Finally, the computational costs of both approaches using the Bayesian setting is given in Table 1, denoting the significant efficiency of the domain decomposition technique.

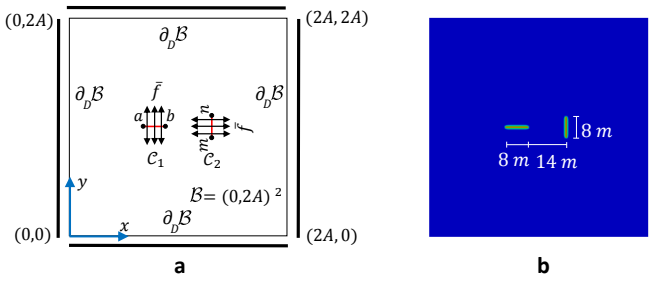


Fig. 4 Joining of two cracks driven by fluid volume injection. (a) Geometry and boundary conditions; and (b) described crack phase-field d as a Dirichlet boundary conditions at $t = 0\ s$.

6 Conclusion

In this study, we extended a global-local (GL) approach for phase-field fracture as the PDE-based model with Bayesian inversion. We applied the proposed idea to hydraulic fracturing within poromechanics concepts, for materials undergoing large deformations. For our numerical example, Bayesian inversion using GL is 20 times faster than the signle-scale model, while the accuracy is similar.

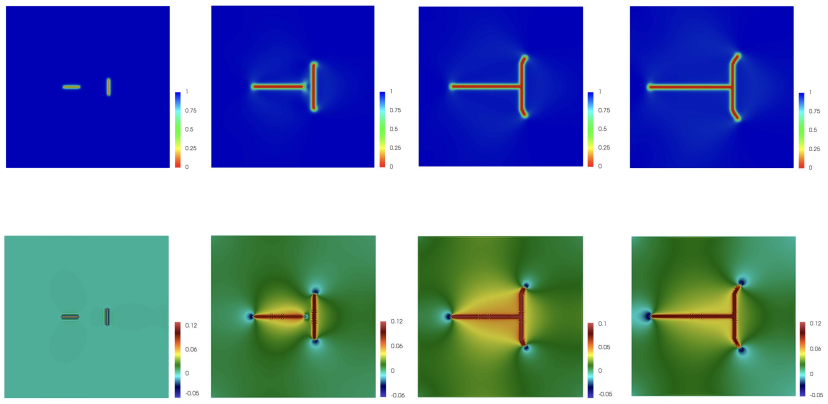


Fig. 5 The evolution of the phase-field (first line) and pressure (second line) for different fluid injection time, i.e., $t \in [0.1, 10, 15, 20]$ seconds using SS model.

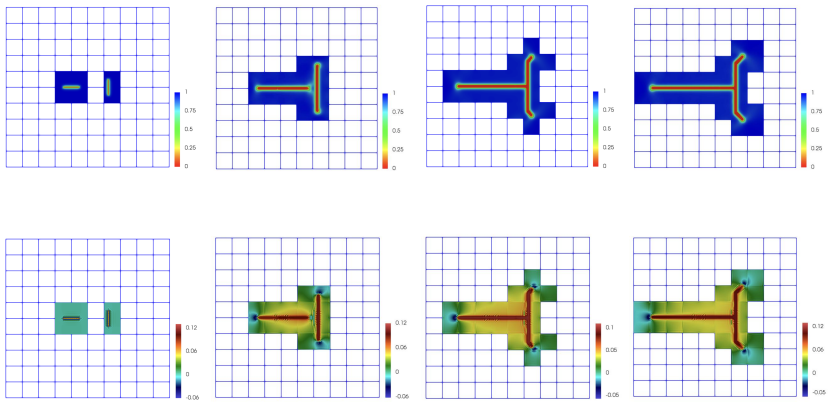


Fig. 6 The evolution of the phase-field (first line) and pressure (second line) for different fluid injection time, i.e., $t \in [0.1, 10, 15, 20]$ seconds using GL model.

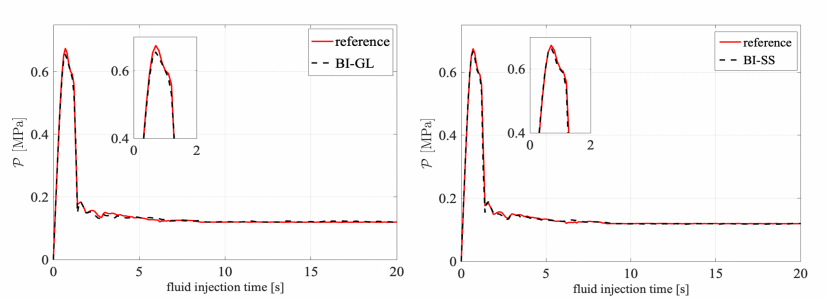


Fig. 7 A comparison between the maximum pressure obtained by the true values (the reference observation) and the mean value of posterior density of BI-GL (left) and BI-SS (right).

Table 1 A comparison between the computational costs of BI-SS and BI-GL approaches for hydraulic fracture. The unit is given in seconds.

Model	min T	max T	mean T	$\sum T$	ratio T
BI – SS	5 645	5 767	5 704	1.14×10^6	19.47
BI – GL	277	296.2	287.1	5.75×10^4	–

Acknowledgements N. Noii acknowledges the Priority Program Deutsche Forschungsgemeinschaft DFG-SPP 2020 within its second funding phase. T. Wick and A. Khodadadian acknowledge the DFG under Germany Excellence Strategy within the Cluster of Excellence PhoenixD (EXC 2122, Project ID 390833453.

References

1. Aldakheel, F., Noii, N., Wick, T., and Wriggers, P. A global–local approach for hydraulic phase-field fracture in poroelastic media. *Computers & Mathematics with Applications* **91**, 99–121 (2021).
2. Bourdin, B., Francfort, G., and Marigo, J.-J. Numerical experiments in revisited brittle fracture. *Journal of the Mechanics and Physics of Solids* **48**(4), 797–826 (2000).
3. Francfort, G. and Marigo, J.-J. Revisiting brittle fracture as an energy minimization problem. *Journal of the Mechanics and Physics of Solids* **46**(8), 1319–1342 (1998).
4. Gendre, L., Allix, O., Gosselet, P., and Comte, F. Non-intrusive and exact global/local techniques for structural problems with local plasticity. *Computational Mechanics* **44**, 233–245 (2009).
5. Gerasimov, T., Noii, N., Allix, O., and De Lorenzis, L. A non-intrusive global/local approach applied to phase-field modeling of brittle fracture. *Advanced Modeling and Simulation in Engineering Sciences* **5**(1), 14 (2018).
6. Miehe, C., Hofacker, M., Schänzel, L.-M., and Aldakheel, F. Phase field modeling of fracture in multi-physics problems. Part II. Coupled brittle-to-ductile failure criteria and crack propagation in thermo-elastic–plastic solids. *Computer Methods in Applied Mechanics and Engineering* **294**, 486–522 (2015).
7. Miehe, C. and Mauthe, S. Phase field modeling of fracture in multi-physics problems. part iii. crack driving forces in hydro-poro-elasticity and hydraulic fracturing of fluid-saturated porous media. *Computer Methods in Applied Mechanics and Engineering* **304**, 619–655 (2016).
8. Noii, N., Fan, M., Wick, T., and Jin, Y. A quasi-monolithic phase-field description for orthotropic anisotropic fracture with adaptive mesh refinement and primal–dual active set method. *Engineering Fracture Mechanics* **258**, 108060 (2021).
9. Noii, N., Khodadadian, A., Ulloa, J., Aldakheel, F., Wick, T., Francois, S., and Wriggers, P. Bayesian inversion with open-source codes for various one-dimensional model problems in computational mechanics. *Archives of Computational Methods in Engineering* **29**(6), 4285–4318 (2022).
10. Noii, N., Khodadadian, A., and Wick, T. Bayesian inversion for anisotropic hydraulic phase-field fracture. *Computer Methods in Applied Mechanics and Engineering* **386**, 114118 (2021).
11. Noii, N., Khodadadian, A., and Wick, T. Bayesian inversion using global-local forward models applied to fracture propagation in porous media. *International Journal for Multiscale Computational Engineering* **20**(3) (2022).
12. Toselli, A. and Widlund, O. *Domain decomposition methods-algorithms and theory*, vol. 34. Springer Science & Business Media (2004).
13. Zhang, J., Vrugt, J. A., Shi, X., Lin, G., Wu, L., and Zeng, L. Improving Simulation Efficiency of MCMC for Inverse Modeling of Hydrologic Systems with a Kalman-Inspired Proposal Distribution. *Water Resources Research* **56**(3), e2019WR025474 (2020).

On Algebraic Bounds for POSM and MRAS

Martin J. Gander and Michal Outrata

1 Introduction and preliminaries

We consider the Poisson equation as our model problem, i.e.,

$$\Delta u = f \quad \text{in } \Omega := (-a, a) \times (0, b) \quad \text{and} \quad u = g \quad \text{on } \partial\Omega, \tag{1}$$

where f and g are given. We decompose Ω into two subdomains $\Omega_1 := (-a, L/2) \times (0, b)$ and $\Omega_2 := (-L/2, a) \times (0, b)$ with interfaces Γ_1 and Γ_2, overlap $O := (-L/2, L/2) \times (0, b)$ (if $L > 0$) and complements $\Theta_2 := \Omega \backslash \Omega_1$ and $\Theta_1 := \Omega \backslash \Omega_2$. Creating an equidistant mesh on Ω with mesh size h, we denote by $N_r + 1$ the number of grid rows and $N_c + 1$ the number of grid columns, see Figure 1. We also define the one-grid-column-prolonged subdomains $\Omega_1^h := (-a, L/2 + h) \times (0, b)$ and $\Omega_2^h := (-L/2 - h, a) \times (0, b)$ and also their interfaces $\Gamma_1^h := (L/2 + h) \times (0, b)$ and $\Gamma_2^h := (-L/2 - h) \times (0, b)$. We discretize (1) with a finite difference scheme, obtaining the block tridiagonal system matrix

$$\begin{bmatrix} A_{\Theta_1} & A_{\Theta_1,\Gamma_2} & & & \\ A_{\Gamma_2,\Theta_1} & A_{\Gamma_2} & A_{\Gamma_2,O} & & \\ & A_{O,\Gamma_1} & A_O & A_{O,\Gamma_1} & \\ & & A_{\Gamma_1,O} & A_{\Gamma_1} & A_{\Gamma_1,\Theta_2} \\ & & & A_{\Theta_2,\Gamma_1} & A_{\Theta_2} \end{bmatrix}. \tag{2}$$

We follow the notation of [3, Section 6.1] and introduce the *parallel optimized Schwarz method* (POSM) with the transmission operators $\mathcal{P}_{\Gamma_1} = \mathcal{P}_{\Gamma_2} = p\mathcal{I}$ and $\mathcal{Q}_{\Gamma_1} = \mathcal{Q}_{\Gamma_2} = \mathcal{I}$ acting on the Dirichlet and Neumann data along the interfaces. Hence POSM is given by the iteration operator $\mathcal{T} : (u_1^{(n-1)}, u_2^{(n-1)}) \mapsto (u_1^{(n)}, u_2^{(n)})$, where $u_1^{(n)}, u_2^{(n)}$ are given as the solutions of the subdomain problems

Martin J. Gander, Michal Outrata
University of Geneva, e-mail: martin.gander@unige.ch, michal.outrata@unige.ch

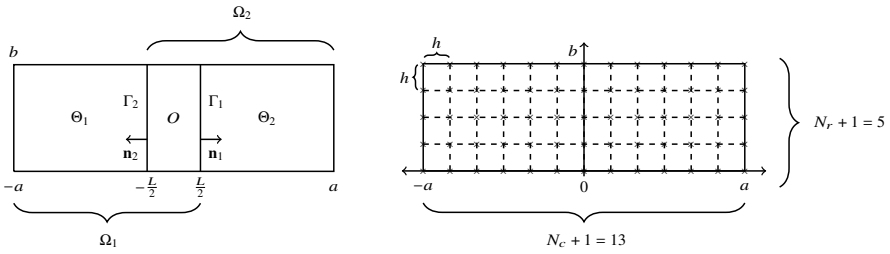


Fig. 1 The physical domain (left), and the discrete mesh (right).

$$\begin{aligned} \Delta u_i^{(n)} &= f \quad \text{in } \Omega_i, \quad u_i^{(n)} = g \quad \text{on } \partial\Omega_i \backslash \Gamma_i, \\ \mathbf{n}_i \cdot \nabla u_i^{(n)} + p u_i^{(n)} &= \mathbf{n}_i \cdot \nabla u_j^{(n-1)} + p u_j^{(n-1)} \quad \text{on } \Gamma_i, \end{aligned} \qquad \text{for } i, j = 1, 2,\ |i - j| = 1.$$

The convergence factor of POSM (see [1, Proposition 2]) as a function of $a, b, L/2$ and the Fourier mode $k \in \mathbb{N}$ is given by

$$\frac{\frac{k\pi}{b}\coth\left(\frac{k\pi}{b}(a - L/2)\right) - p}{\frac{k\pi}{b}\coth\left(\frac{k\pi}{b}(a + L/2)\right) + p} \cdot \frac{\sinh\left(\frac{k\pi}{b}(a - L/2)\right)}{\sinh\left(\frac{k\pi}{b}(a + L/2)\right)}. \tag{3}$$

Writing (2) in its *augmented form* and modifying the interface block rows we get

$$A_{\text{aug}} := \begin{bmatrix} \tilde{A}_{\Omega_1} & \tilde{A}_{\Omega_1,\Omega_2} \\ \tilde{A}_{\Omega_2,\Omega_1} & \tilde{A}_{\Omega_2} \end{bmatrix} := \begin{bmatrix} A_{\Theta_1} & A_{\Theta_1,\Gamma_2} & & & & & & \\ A_{\Gamma_2,\Theta_1} & A_{\Gamma_2} & A_{\Gamma_2,O} & & & & & \\ & A_{O,\Gamma_2} & A_O & A_{O,\Gamma_1} & & & & \\ & & A_{\Gamma_1,O} & \tilde{A}_{\Gamma_1} & & & \tilde{A}_{\Gamma_1,\Gamma_1} & A_{\Gamma_1,\Theta_2} \\ A_{\Gamma_2,\Theta_1} & \tilde{A}_{\Gamma_2,\Gamma_2} & & & \tilde{A}_{\Gamma_2} & A_{\Gamma_2,O} & & \\ & & & & A_{O,\Gamma_2} & A_O & A_{O,\Gamma_1} & \\ & & & & & A_{\Gamma_1,O} & A_{\Gamma_1} & A_{\Gamma_1,\Theta_2} \\ & & & & & & A_{\Theta_2,\Gamma_1} & A_{\Theta_2} \end{bmatrix}, \tag{4}$$

where we introduced the discrete transmission conditions in the last block row of $[A_{\Omega_1}\ A_{\Omega_1,\Omega_2}]$ and the first block row of $[A_{\Omega_2,\Omega_1}\ A_{\Omega_2}]$, which are now given by

$$\tilde{A}_{\Gamma_1} := A_{\Gamma_1} + D,\ \tilde{A}_{\Gamma_1,\Gamma_1} := -D \quad \text{and} \quad \tilde{A}_{\Gamma_2} := A_{\Gamma_2} + D,\ \tilde{A}_{\Gamma_2,\Gamma_2} := -D.$$

We are interested in the subdomain version of the *modified restricted additive Schwarz* (MRAS[1], see [2]), defined by its iteration matrix T,

$$T = I - \sum_{i=1}^{2} R_{\Omega_i}^T \tilde{A}_{\Omega_i}^{-1} R_{\Omega_i} \tilde{A}_{\text{aug}} \quad \text{with } R_{\Omega_1} = [I_{\Omega_1}\ 0_{\Omega_2}],\ R_{\Omega_2} = [0_{\Omega_1}\ I_{\Omega_2}]. \tag{5}$$

[1] MRAS was introduced in the so-called *globally deferred correction form*, where we iterate on the global solution unknowns, in contrast to iterating on the subdomain solution unknowns here. This is but a technicality and hence we keep the name; the equivalence is shown in [3, Section 6.1, 6.2].

Notice that the interface block structure of MRAS does *not* match the one in [3, Algorithm 2] but the transmission matrix D is chosen to get fast convergence, analogously to the parameter p in POSM. Setting

$$E_{\Gamma_2}^{\Omega_1} := \left[0_{\Theta_1} I_{\Gamma_2} 0_O 0_{\Gamma_1}\right]^T,\ E_{\Gamma_1}^{\Omega_1} := \left[0_{\Theta_1} 0_{\Gamma_2} 0_O I_{\Gamma_1}\right]^T,\ E_{\Theta_1}^{\Omega_1} := \left[A_{\Gamma_2,\Theta_1} 0_{\Gamma_2} 0_O 0_{\Gamma_1}\right]^T,$$
$$E_{\Gamma_2}^{\Omega_2} := \left[I_{\Gamma_2} 0_O 0_{\Gamma_1} 0_{\Theta_2}\right]^T,\ E_{\Gamma_1}^{\Omega_2} := \left[0_{\Gamma_2} 0_O I_{\Gamma_1} 0_{\Theta_2}\right]^T,\ E_{\Theta_2}^{\Omega_2} := \left[0_{\Gamma_2} 0_O 0_{\Gamma_1} A_{\Theta_2,\Gamma_1}\right]^T,$$

we can write

$$\tilde{A}_{\Omega_i} = A_{\Omega_i} + E_{\Gamma_i}^{\Omega_i} D \left(E_{\Gamma_i}^{\Omega_i}\right)^T, \quad i = 1, 2,$$

and formulate a convergence result for MRAS, analogue to [2, Theorem 3.2].

Theorem 1 ([2, Section 3])
The MRAS iteration matrix T in (5) *has the structure*

$$T = \begin{bmatrix} 0 & K \\ L & 0 \end{bmatrix}, \quad \begin{aligned} K &:= A_{\Omega_1}^{-1} E_{\Gamma_1}^{\Omega_1} \left[I + D(A_{\Omega_1}^{-1})_{\Gamma_1,\Gamma_1}\right]^{-1} \left(-D(E_{\Gamma_1}^{\Omega_2})^T + (E_{\Theta_2}^{\Omega_2})^T\right), \\ L &:= A_{\Omega_2}^{-1} E_{\Gamma_2}^{\Omega_2} \left[I + D(A_{\Omega_2}^{-1})_{\Gamma_2,\Gamma_2}\right]^{-1} \left(-D(E_{\Gamma_2}^{\Omega_1})^T + (E_{\Theta_1}^{\Omega_1})^T\right). \end{aligned} \tag{6}$$

Moreover, the asymptotic convergence factor of MRAS is bounded by

$$\begin{aligned} &\sqrt{\|M_1 B_1\|_2 \cdot \|M_2 B_2\|_2}, \\ M_1 &:= \left[I + D(A_{\Omega_1}^{-1})_{\Gamma_1,\Gamma_1}\right]^{-1} \left(-D - A_{\Gamma_1,\Theta_2} A_{\Theta_2}^{-1} A_{\Theta_2,\Gamma_1}\right),\ B_1 := (A_{\Omega_2}^{-1})_{\Gamma_1,\Gamma_2}, \\ M_2 &:= \left[I + D(A_{\Omega_2}^{-1})_{\Gamma_2,\Gamma_2}\right]^{-1} \left(-D - A_{\Gamma_2,\Theta_1} A_{\Theta_1}^{-1} A_{\Theta_1,\Gamma_2}\right),\ B_2 := (A_{\Omega_1}^{-1})_{\Gamma_2,\Gamma_1}. \end{aligned} \tag{7}$$

Due to the symmetry of the model problem and the method we have $B := B_1 = B_2$ and $M := M_1 = M_2$, which in turn simplifies the bound in (7) *to $\|MB\|_2$.*

2 Analysis of the MRAS bound and its reformulation

First, we recall the sine series expansion in the y direction $\mathcal{F}_y$, so that we have

$$u(x, y) = \sum_{k=1}^{+\infty} \mathcal{F}_y u(x, k) \sin\left(\frac{k\pi}{b} y\right) \equiv \sum_{k=1}^{+\infty} \hat{u}(x, k) \sin\left(\frac{k\pi}{b} y\right),$$

with[2] $\mathcal{F}_y u := \int_0^b u(x, y) \sin(k\pi y/b) \mathrm{d}y$. Next, we factor out $(A_{\Omega_1}^{-1})_{\Gamma_1,\Gamma_1}$ and $(A_{\Omega_2}^{-1})_{\Gamma_2,\Gamma_2}$ on the left from $M_{1,2}$, so that instead of (7) we focus on the asymptotically equivalent

[2] Using the sine series relies on the Dirichlet boundary conditions (BCs) along $\{y = 0\}$ and $\{y = b\}$ in (1); for different BCs see [4].

$$MB := \underbrace{\left[\left((A_{\Omega_1}^{-1})_{\Gamma_1,\Gamma_1}\right)^{-1} + D\right]^{-1}}_{(T^{\text{Denom}})^{-1}} \underbrace{\left(-D - A_{\Gamma_1,\Theta_2} A_{\Theta_2}^{-1} A_{\Theta_2,\Gamma_1}\right)}_{T^{\text{Numer}}} \underbrace{(A_{\Omega_2}^{-1})_{\Gamma_1,\Gamma_2}\left((A_{\Omega_2}^{-1})_{\Gamma_2,\Gamma_2}\right)^{-1}}_{T^{\text{Over}}}. \quad (8)$$

The *key* question is whether the bound (7), which now becomes $\|MB\|$, is the discrete analogue of (3) – piece by piece. Linking each of the blocks in (8) to a discrete linear operator with a continuous counterpart, we analyze it using the Fourier series expansion. Taking $\mathbf{b} \in \mathbb{R}^{N_r-1}$ and interpolating it to a function $\gamma\colon \Gamma_1^h \to \mathbb{R}$, the following problems are equivalent up to the FD discretization:

$$A_{\Omega_1}\mathbf{u} = -\frac{1}{h^2} E_{\Gamma_1}^{\Omega_1}\mathbf{b} \qquad \text{and} \qquad \begin{aligned} \Delta u &= 0 \quad \text{in } \Omega_1^h, \\ u &= 0 \quad \text{on } \partial\Omega_1^h \backslash \Gamma_1^h, \quad \text{and} \quad u = \gamma \quad \text{on } \Gamma_1^h. \end{aligned} \quad (9)$$

Defining the solution operator by $\mathcal{S}_1(\gamma) = u\big|_{\Gamma_1}$ where u is the solution of (9), we have (up to the FD discretization) the equivalence of the linear operators $-1/h^2(A_{\Omega_1}^{-1})_{\Gamma_1,\Gamma_1}$ and $\mathcal{S}_1$. To calculate $\mathcal{S}_1$ we expand in the y variable using $\mathcal{F}_y$, simplifying the continuous problem in (9) to the semi-discrete problem

$$\begin{gathered} \left(\partial_{xx} - \left(\tfrac{k\pi}{b}\right)^2\right)\hat{u}(x,k) = 0 \quad \text{for } x \in (-a, L/2+h) \text{ and } k \in \mathbb{N}, \\ \hat{u}(-a,k) = 0 \quad \text{and} \quad \hat{u}(L/2+h,k) = \hat{\gamma}(k) \quad \text{for } k \in \mathbb{N}, \end{gathered} \quad (10)$$

and denote by $\hat{\mathcal{S}}_1 := \mathcal{F}_y \mathcal{S}_1$ the Fourier symbol of $\mathcal{S}_1$. A direct calculation yields

$$\hat{u}(x,k) = \frac{\sinh\left(\frac{k\pi}{b}(a+x)\right)\hat{\gamma}(k)}{\sinh\left(\frac{k\pi}{b}(a+L/2+h)\right)}, \quad \hat{\mathcal{S}}_1\hat{\gamma}(k) = \frac{\sinh\left(\frac{k\pi}{b}(a+L/2)\right)}{\sinh\left(\frac{k\pi}{b}(a+L/2+h)\right)}\hat{\gamma}(k).$$

Therefore, the eigenvalues of the linear operator $-1/h^2(A_{\Omega_1}^{-1})_{\Gamma_1,\Gamma_1}$ approximate the modes $k = 1, \ldots, N_r - 1$ of $\hat{\mathcal{S}}_1$ given above, as we see in Figure 2. The rest of the blocks in (8) are summarized in Table 1 and illustrated in Figure 2, see [4] for detailed calculations. We see that the approximation is very accurate for the low-frequency modes but not quite accurate for the high-frequency ones. If D diagonalizes in the same basis as the rest of the blocks and we denote its eigenvalues by $\delta_1, \ldots, \delta_{N_r-1}$, then the eigenvalues of $T^{\text{Denom}}, T^{\text{Numer}}, T^{\text{Over}}$ approximate certain discrete (truncated) Fourier symbols we present in Table 2 and illustrate in Figure 3. We see that the inaccuracy on the high frequencies is still present. More importantly, comparing Table 2 with (3) shows that the contraction factor due to the domain overlap in (3) matches exactly θ_k for each k, i.e., the one due to the continuous representation of T^{Over}. However, this is clearly *not* the case for the contraction factor due to the transmission condition induced by D. The ratio η_k/ζ_k shows that choosing $\delta_k = p$ (the naive choice) is *not* the correct one (see [4] for more details) and we continue by reformulating Theorem 1 to reflect also the transmission part of (3).

Table 1 The blocks and corresponding linear operators (LO) from (8).

block	discrete LO	continuous LO	Fourier symbol
$(A_{\Omega_1}^{-1})_{\Gamma_1,\Gamma_1}$	$-\frac{1}{h^2}(A_{\Omega_1}^{-1})_{\Gamma_1,\Gamma_1}$	$\mathcal{S}_1 : \gamma \mapsto u\big\vert_{\Gamma_1}$	$\hat{\mathcal{S}}_1 = \frac{\sinh\left(\frac{k\pi}{b}(a+L/2)\right)}{\sinh\left(\frac{k\pi}{b}(a+L/2+h)\right))}$
$A_{\Gamma_1,\Theta_2}A_{\Theta_2}^{-1}A_{\Theta_2,\Gamma_1}$	$-h^2 A_{\Gamma_1,\Theta_2}A_{\Theta_2}^{-1}A_{\Theta_2,\Gamma_1}$	$\mathcal{S}_2 : \gamma \mapsto u\big\vert_{\Gamma_1}$	$\hat{\mathcal{S}}_2 = \frac{\sinh\left(\frac{k\pi}{b}(a-L/2-h)\right)}{\sinh\left(\frac{k\pi}{b}(a-L/2))\right)}$
$(A_{\Omega_2}^{-1})_{\Gamma_1,\Gamma_2}$	$-\frac{1}{h^2}(A_{\Omega_2}^{-1})_{\Gamma_1,\Gamma_2}$	$\mathcal{S}_3 : \gamma \mapsto u\big\vert_{\Gamma_1}$	$\hat{\mathcal{S}}_3 = \frac{\sinh\left(\frac{k\pi}{b}(a-L/2)\right)}{\sinh\left(\frac{k\pi}{b}(a+L/2+h)\right))}$
$(A_{\Omega_2}^{-1})_{\Gamma_2,\Gamma_2}$	$-\frac{1}{h^2}(A_{\Omega_2}^{-1})_{\Gamma_2,\Gamma_2}$	$\mathcal{S}_4 : \gamma \mapsto u\big\vert_{\Gamma_2}$	$\hat{\mathcal{S}}_4 = \frac{\sinh\left(\frac{k\pi}{b}(a+L/2)\right)}{\sinh\left(\frac{k\pi}{b}(a+L/2+h)\right))}$

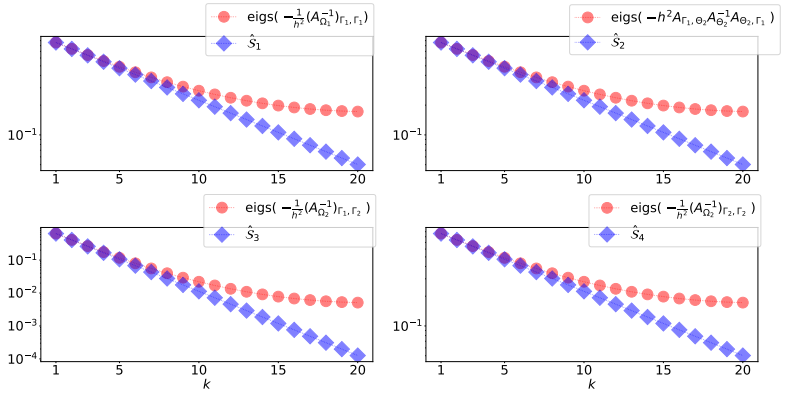


Fig. 2 Results obtained for the parameters $a = b = 1$, $L = 2h$, $N_r = 22$.

Table 2 The matrices and their corresponding (truncated) Fourier symbols.

$(T^{\text{Denom}})^{-1}$	T^{Numer}	T^{Over}
$\eta_k := \delta_k - \frac{1}{h^2}\frac{\sinh\left(\frac{k\pi}{b}(a+L/2+h)\right)}{\sinh\left(\frac{k\pi}{b}(a+L/2)\right)}$	$\zeta_k := -\delta_k + \frac{1}{h^2}\frac{\sinh\left(\frac{k\pi}{b}(a-L/2-h)\right)}{\sinh\left(\frac{k\pi}{b}(a-L/2))\right)}$	$\theta_k := \frac{\sinh\left(\frac{k\pi}{b}(a-L/2)\right)}{\sinh\left(\frac{k\pi}{b}(a+L/2))\right)}$
$\left(\overline{T}^{\text{Denom}}\right)^{-1}$	$\overline{T}^{\text{Numer}}$	$\overline{T}^{\text{Over}}$
$\overline{\eta}_k := -\frac{1}{h}\frac{k\pi}{b}\coth\left(\frac{k\pi}{b}(a+L/2)\right) - \lambda_k$	$\overline{\zeta}_k := -\frac{1}{h}\frac{k\pi}{b}\coth\left(\frac{k\pi}{b}(a-L/2)\right) + \lambda_k$	$\overline{\theta}_k := \frac{\sinh\left(\frac{k\pi}{b}(a-L/2)\right)}{\sinh\left(\frac{k\pi}{b}(a+L/2))\right)}$

The main tool used to obtain Theorem 1 is the Sherman-Morrison-Woodbury formula for the inverse of a low-rank updated matrix, here the update was the corner block D. We now show that using the same formula for a slightly different block gives the "correct" result. We split the interface blocks as in [3, Section 5.2] and write $A_{\Gamma_1} = A_{\Gamma_1}^L + A_{\Gamma_1}^R$ and $A_{\Gamma_2} = A_{\Gamma_2}^L + A_{\Gamma_2}^R$ so that we have

$$\begin{aligned} -h(A_{\Gamma_1,O}\mathbf{u}_O + A_{\Gamma_1}^L\mathbf{u}_{\Gamma_1}) \approx u_x\big\vert_{\Gamma_1}, \quad -h(A_{\Gamma_1,\Theta_2}\mathbf{u}_{\Theta_2} + A_{\Gamma_1}^R\mathbf{u}_{\Gamma_1}) \approx -u_x\big\vert_{\Gamma_1}, \\ -h(A_{\Gamma_2,O}\mathbf{u}_O + A_{\Gamma_2}^R\mathbf{u}_{\Gamma_2}) \approx -u_x\big\vert_{\Gamma_2}, \quad -h(A_{\Gamma_2,\Theta_1}\mathbf{u}_{\Theta_1} + A_{\Gamma_2}^L\mathbf{u}_{\Gamma_2}) \approx u_x\big\vert_{\Gamma_2}. \end{aligned} \tag{11}$$

This is natural for FD and FEM discretizations. Using the so-called *ghost point trick* we get $A_{\Gamma_1}^L = A_{\Gamma_1}^R = \frac{1}{2}A_{\Gamma_1}$, $A_{\Gamma_2}^L = A_{\Gamma_2}^R = \frac{1}{2}A_{\Gamma_2}$. Adopting this we rewrite $\tilde{A}_{aug}$ as

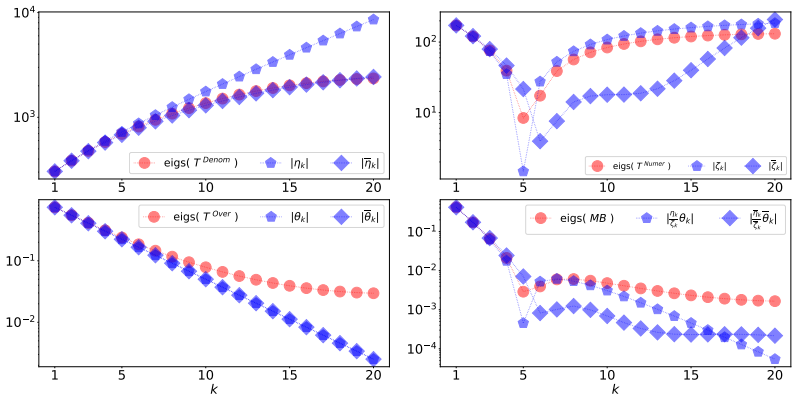


Fig. 3 Results obtained with $a = b = 1$, $L = 2h$, $N_r = 21$ and $D = \operatorname{diag}\left(\pi^2/h\right)$.

$$\overline{A}_{\text{aug}} := \begin{bmatrix} A^L_{\Omega_1} + \overline{A}_{\Omega_1} & \tilde{A}_{\Omega_1,\Omega_2} \\ \tilde{A}_{\Omega_2,\Omega_1} & A^R_{\Omega_1} + \overline{A}_{\Omega_2} \end{bmatrix} := \begin{bmatrix} A_{\Theta_1} & A_{\Theta_1,\Gamma_2} & & & & & & \\ A_{\Gamma_2,\Theta_1} & A_{\Gamma_2} & A_{\Gamma_2,O} & & & & & \\ & A_{O,\Gamma_2} & A_O & A_{O,\Gamma_1} & & & & \\ & & A_{\Gamma_1,O} & A^L_{\Gamma_1} + \overline{A}_{\Gamma_1} & & & \tilde{A}_{\Gamma_1,\Gamma_1} & A_{\Gamma_1,\Theta_2} \\ A_{\Gamma_2,\Theta_1} & \tilde{A}_{\Gamma_2,\Gamma_2} & & & A^R_{\Gamma_2} + \overline{A}_{\Gamma_2} & A_{\Gamma_2,O} & & \\ & & & & A_{O,\Gamma_2} & A_O & A_{O,\Gamma_1} & \\ & & & & & A_{\Gamma_1,O} & A_{\Gamma_1} & A_{\Gamma_1,\Theta_2} \\ & & & & & & A_{\Theta_2,\Gamma_1} & A_{\Theta_2} \end{bmatrix},$$

with the transmission conditions kept the same as in (4) but reorganized with

$$\overline{A}_{\Gamma_1} := A^R_{\Gamma_1} + D, \quad \text{and} \quad \overline{A}_{\Gamma_2} := A^L_{\Gamma_2} + D.$$

As a result, the Sherman-Morrison-Woodbury formula is now used for $\left(A^L_{\Omega_1} + \overline{A}_{\Omega_1}\right)^{-1}$ and $\left(A^R_{\Omega_1} + \overline{A}_{\Omega_2}\right)^{-1}$ and analogously to [2, Lemma 3.1, Theorem 3.2] we obtain Theorem 2 (we take advantage of the symmetry, for the general case see [4]).

Theorem 2 *The MRAS iteration matrix T in* (5) *can also be written as*

$$\overline{T} = \begin{bmatrix} 0 & \overline{K} \\ \overline{L} & 0 \end{bmatrix}, \quad \text{with}$$

$$\overline{K} := \left(A^L_{\Omega_1}\right)^{-1} E^{\Omega_1}_{\Gamma_1} \left(\left(\left(A^L_{\Omega_1}\right)^{-1} \right)_{\Gamma_1,\Gamma_1} \right)^{-1} \left[\left(\left(\left(A^L_{\Omega_1}\right)^{-1} \right)_{\Gamma_1,\Gamma_1} \right)^{-1} + \overline{A}_{\Gamma_1} \right]^{-1} \left(-D(E^{\Omega_2}_{\Gamma_1})^T + (E^{\Omega_2}_{\Theta_2})^T \right),$$

$$\overline{L} := \left(A^R_{\Omega_2}\right)^{-1} E^{\Omega_2}_{\Gamma_2} \left(\left(\left(A^R_{\Omega_2}\right)^{-1} \right)_{\Gamma_2,\Gamma_2} \right)^{-1} \left[\left(\left(\left(A^R_{\Omega_2}\right)^{-1} \right)_{\Gamma_2,\Gamma_2} \right)^{-1} + \overline{A}_{\Gamma_2} \right]^{-1} \left(-D(E^{\Omega_1}_{\Gamma_2})^T + (E^{\Omega_1}_{\Theta_1})^T \right).$$

Moreover, the asymptotic convergence factor of POSM is bounded by

$$\|\overline{MB}\|_2, \quad \text{where} \tag{12}$$

$$\begin{aligned}\overline{M} &:= \left(\overline{T}^{\,\mathrm{Denom}}\right)^{-1}\overline{T}^{\,\mathrm{Numer}} = \left[\left((\left(A^L_{\Omega_1}\right)^{-1})_{\Gamma_1,\Gamma_1}\right)^{-1} + \overline{A}_{\Gamma_1}\right]^{-1}\left(\left(A^R_{\Gamma_1} - A_{\Gamma_1,\Theta_2}A^{-1}_{\Theta_2}A_{\Theta_2,\Gamma_1}\right) - \overline{A}_{\Gamma_1}\right),\\ \overline{B} &:= \overline{T}^{\,\mathrm{Over}} = ((\left(A^R_{\Omega_2}\right)^{-1})_{\Gamma_1,\Gamma_2}\left((\left(A^R_{\Omega_2}\right)^{-1})_{\Gamma_2,\Gamma_2}\right)^{-1}.\end{aligned} \tag{13}$$

Focusing on the first block in (13), we take $\mathbf{b} \in \mathbb{R}^{N_r-1}$ and interpolating it to a function $\gamma\colon \Gamma_1 \to \mathbb{R}$, the following problems are equivalent up to the FD discretization:

$$A^L_{\Omega_1}\mathbf{u} = -\tfrac{1}{h}E^{\Omega_1}_{\Gamma_1}\mathbf{b} \qquad \text{and} \qquad \begin{aligned}\Delta u &= 0 \quad \text{in } \Omega_1,\\ u &= 0 \quad \text{on } \partial\Omega_1\backslash\Gamma_1, \quad \text{and} \quad \mathbf{n}_1\cdot\nabla u = \gamma \quad \text{on } \Gamma_1.\end{aligned} \tag{14}$$

Setting $\overline{\mathcal{S}}_2(\gamma) = u\big|_{\Gamma_1}$, where u is the solution of (14) we have the equivalence (up to the FD discretization) of $-1/h(A^{-1}_{\Omega_1})_{\Gamma_1,\Gamma_1}$ and $\overline{\mathcal{S}}_2$. Considering

$$\begin{aligned}\left(\partial_{xx} - \left(\tfrac{k\pi}{b}\right)^2\right)\hat{u}(x,k) = 0 \quad &\text{for } x \in (-a, L/2+h) \text{ and } k \in \mathbb{N},\\ \hat{u}(-a,k) = 0 \quad \text{and} \quad \hat{u}_x(L/2+h,k) = \hat{\gamma}(k) \quad &\text{for } k \in \mathbb{N},\end{aligned} \tag{15}$$

we set $\hat{\overline{\mathcal{S}}}_1 := \mathcal{F}_y\overline{\mathcal{S}}_1$ and a direct calculation yields the solution of (15) and $\hat{\overline{\mathcal{S}}}_1$ as

$$\hat{u}(x,k) = \frac{\sinh\left(\frac{k\pi}{b}(a+x)\right)}{\frac{k\pi}{b}\cosh\left(\frac{k\pi}{b}(a+L/2)\right)}, \quad \hat{\overline{\mathcal{S}}}_1\hat{\gamma}(k) = \frac{\sinh\left(\frac{k\pi}{b}(a+L/2)\right)}{\frac{k\pi}{b}\cosh\left(\frac{k\pi}{b}(a+L/2)\right)}\hat{\gamma}(k).$$

Therefore, the eigenvalues of $-1/h((A^L_{\Omega_1})^{-1})_{\Gamma_1,\Gamma_1}$ approximate the first N_r-1 modes of $\mathcal{F}_y\overline{\mathcal{S}}_1$ with better accuracy in high-frequencies than we observed with $\mathcal{S}_1$, see Figure 2 and Figure 4. For the other blocks see Table 3 and Figure 4. If $-\overline{A}^R_{\Gamma_1}$ diagonalizes in the Fourier discrete basis with eigenvalues $\lambda_1, \ldots, \lambda_{N_r-1}$, then the eigenvalues of $\overline{T}^{\,\mathrm{Denom}}, \overline{T}^{\,\mathrm{Numer}}, \overline{T}^{\,\mathrm{Over}}$ approximate certain discrete (truncated) Fourier symbols, presented in Table 2 and Figure 3. Notice that at the discrete level we have $MB = \overline{MB}$, i.e., the difference is in the *representation* of the bound (blue markers in Figure 3) as we changed *only* the block organization in the Sherman-Morrison-Woodbury formula. Comparing Table 2 with (3), we get the link between λ_k (and hence also δ_k) and the Robin parameter p in (3). Calculating the optimal p now directly translates to the optimal choice of D by

$$pI = -hW^T\left(A^R_{\Gamma_1} + D\right)W \quad \text{, i. e.,} \quad D = -\frac{p}{h}I - A^R_{\Gamma_1}.$$

Table 3 The blocks and corresponding linear operators (LO) from (8).

block	discrete LO	continuous LO	Fourier symbol
$(\left(A^{L}_{\Omega_1}\right)^{-1})_{\Gamma_1,\Gamma_1}$	$-\frac{1}{h}(\left(A^{L}_{\Omega_1}\right)^{-1})_{\Gamma_1,\Gamma_1}$	$\overline{\mathcal{S}}_1 : \gamma \mapsto u\vert_{\Gamma_1}$	$\hat{\overline{\mathcal{S}}}_1 = \frac{1}{\frac{k\pi}{b}\coth\left(\frac{k\pi}{b}(a+L/2)\right)}$
$\overline{A}^{R}_{\Gamma_1} - A_{\Gamma_1,\Theta_2}A^{-1}_{\Theta_2}A_{\Theta_2,\Gamma_1}$	$-h\left(\overline{A}^{R}_{\Gamma_1} - A_{\Gamma_1,\Theta_2}A^{-1}_{\Theta_2}A_{\Theta_2,\Gamma_1}\right)$	$\overline{\mathcal{S}}_2 : \gamma \mapsto \mathbf{n}_1 \cdot \nabla u\vert_{\Gamma_1}$	$\hat{\overline{\mathcal{S}}}_2 = \frac{k\pi}{b}\coth\left(\frac{k\pi}{b}(a-L/2)\right)$
$(\left(A^{R}_{\Omega_2}\right)^{-1})_{\Gamma_1,\Gamma_2}$	$-\frac{1}{h}(\left(A^{R}_{\Omega_2}\right)^{-1})_{\Gamma_1,\Gamma_2}$	$\overline{\mathcal{S}}_3 : \gamma \mapsto u\vert_{\Gamma_1}$	$\hat{\overline{\mathcal{S}}}_3 = \frac{\sinh\left(\frac{k\pi}{b}(a-L/2)\right)}{\frac{k\pi}{b}\cosh\left(\frac{k\pi}{b}(a+L/2)\right)}$
$(\left(A^{R}_{\Omega_2}\right)^{-1})_{\Gamma_2,\Gamma_2}$	$-\frac{1}{h}(\left(A^{R}_{\Omega_2}\right)^{-1})_{\Gamma_2,\Gamma_2}$	$\overline{\mathcal{S}}_4 : \gamma \mapsto u\vert_{\Gamma_2}$	$\hat{\overline{\mathcal{S}}}_4 = \frac{\sinh\left(\frac{k\pi}{b}(a+L/2)\right)}{\frac{k\pi}{b}\cosh\left(\frac{k\pi}{b}(a+L/2)\right)}$

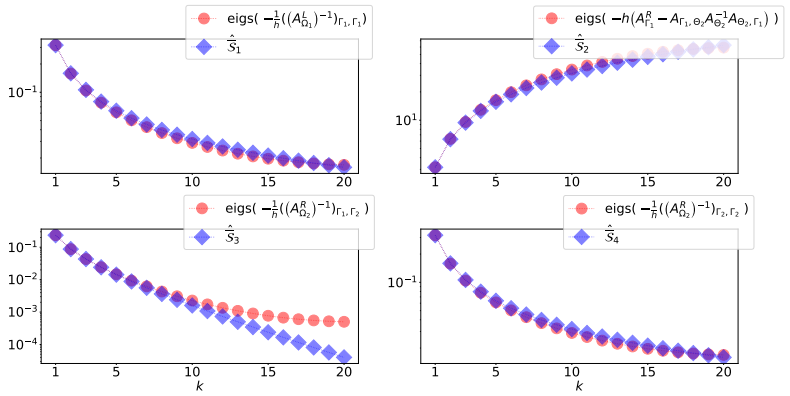


Fig. 4 Results obtained for the parameters $a = b = 1$, $L = 2h$, $N_r = 21$.

References

1. Gander, M. J. On the influence of geometry on optimized Schwarz methods. *SeMA Journal* **53**(1), 71–78 (2013).
2. Gander, M. J., Loisel, S., and Szyld, D. B. An optimal block iterative method and preconditioner for banded matrices with applications to PDEs on irregular domains. *SIAM Journal on Matrix Analysis and Applications* **33**(2), 653–680 (2012).
3. Gander, M. J. and Zhang, H. A class of iterative solvers for the Helmholtz equation: factorizations, sweeping preconditioners, source transfer, single layer potentials, polarized traces, and optimized Schwarz methods. *SIAM Review* **61**(1), 3–76 (2019).
4. Outrata, M. *Schwarz methods, Schur complements, preconditioning and numerical linear algebra*. Ph.D. thesis, University of Geneva, Math Department (2022).

Hierarchical LU Preconditioning for the Time-Harmonic Maxwell Equations

Maryam Parvizi, Amirreza Khodadadian, Sven Beuchler, and Thomas Wick

1 Introduction

The Maxwell system describes the behaviour of electromagnetic fields. Nédélec's edge finite element method is an efficient discretization technique to solve this equation which involves the curl-curl operator numerically [13, 10]. Although the matrix arising from the linear system is sparse, direct solvers cannot solve the problem with linear complexity. In the presence of quasi-uniform meshes, the memory requirement of $O(N^{4/3})$ and computational time of $O(N^2)$ are expected where the problem size is N [11].

Matrices with full rank often can be approximated using low-rank matrices; but, it is not always applicable. Thus, it is desirable to present a block-wise partitioning of the matrix and approximate appropriately chosen (using an admissibility condition) blocks by their low-rank decompositions. Hierarchical matrices ($\mathcal{H}$-matrices), [7], are block wise low-rank matrices that allow us to represent dense matrices with data sparse approximations and the logarithmic-linear storage complexity, i.e., $\mathcal{O}(Nm\log(N))$, where m is a parameter that controls the accuracy of the approximation.

Besides the storage complexity reduction, another application of the $\mathcal{H}$-matrix approximations is to use them as a preconditioner to solve the system directly, or to use them as preconditioner to reduce the number of iterations in Krylov-based iterative solvers based on matrix-vector multiplication, e.g., GMRES [15]. In the time-harmonic case, the system matrix may become indefinite and ill-conditioned, in particular for high frequency cases. In this regime, the usual factorization methods such as incomplete LU (iLU) do not lead to reliable results and converge to the exact solution poorly. Then it is very difficult to design Galerkin discretizations [12] and efficient iterative solvers [4] (see also [9] for recent studies).

Maryam Parvizi, Amirreza Khodadadian, Sven Beuchler, Thomas Wick
Leibniz University Hannover, Institute of Applied Mathematics, Welfengarten 1, 30167 Hannover, Germany e-mail: {parvizi,khodadadian,beuchler,thomas.wick}@ifam.uni-hannover.de
and Cluster of Excellence PhoenixD (Photonics, Optics, and Engineering - Innovation Across Disciplines), Leibniz Universität Hannover, Germany

In this paper, we study the effect of applying hierarchical LU decompositions, i.e., $\mathcal{H}-LU$ decompositions, as a preconditioner to solve the Maxwell equations using iterative solvers. The idea of using $\mathcal{H}$ matrices for the curl-curl operators and magnetostatic model problems was introduced in [3]. A preconditioner based on the $\mathcal{H}$ matrices was used in [2] and [14] for solving the Maxwell equations in the low-frequency regime. In [5], the authors showed that the inverse of the Galerkin matrix corresponding to the FEM discretization of time-harmonic Maxwell equations can be approximated by an $\mathcal{H}$-matrix and proved the exponential convergence in the maximum block-rank. This approximation can be used to prove the existence of $\mathcal{H}$-LU factorization without frequency restriction.

In this work, in addition to addressing the advantage of iterative hierarchical preconditioning, we exploit the influence of applying an inverse of $\mathcal{H}$-matrix approximations as a preconditioner to solve the linear systems directly. Our numerical tests also include studies on the influence of the wave number. As observed, for both low and high frequency materials using $\mathcal{H}-LU$ factorization will lead to a fast and accurate convergence of the iterative solvers.

The paper is organized as follows. In Section 2, we briefly introduce the Maxwell system and present the Nédélec's finite element discretization. The $\mathcal{H}$-matrices and how to compute the inverse of an $\mathcal{H}$-matrix approximation for the Galerkin system matrix are explained in Section 3. We also present an algorithm to compute the $\mathcal{H}-LU$ approximation of a matrix, and use it as a preconditioner to solve the system directly. In Section 4, numerical results are presented to substantiate the efficiency of the hierarchical matrix as a direct and iterative solver. Finally, the conclusions are given in Section 5.

2 The Maxwell equations

Denoting $\hat{\mathcal{H}}$ the magnetic field intensity, $\mathcal{E}$ the electric field density for the domain $\Omega \subset \mathbb{R}^d$ $(d = 2, 3)$, we have the time-harmonic Maxwell equations as

$$\nabla \cdot (\beta \hat{\mathcal{H}}) = 0 \quad \text{in } \Omega \times \mathcal{I}, \tag{1a}$$

$$\nabla \cdot (\alpha \mathcal{E}) = \rho \quad \text{in } \Omega \times \mathcal{I}, \tag{1b}$$

$$\left(\alpha \frac{\partial}{\partial t} + \chi\right) \mathcal{E} - \nabla \times \hat{\mathcal{H}} = \mathcal{F} \quad \text{in } \Omega \times \mathcal{I}, \tag{1c}$$

$$\beta \frac{\partial}{\partial t} \hat{\mathcal{H}} + \nabla \times \mathcal{E} = 0 \quad \text{in } \Omega \times \mathcal{I}, \tag{1d}$$

where $\mathcal{F}$ is the applied electrical force, ρ is the charge density, and $\mathcal{I} = (0, T]$ is the time interval. In addition, α and β are the dielectric and magnetic permeabilities, and χ is the conductivity constant. Considering an arbitrary frequency ω, with respect to time, the electric and magnetic fields can be represented as follows

$$\mathcal{E}(x, t) = e^{-i\omega t} \boldsymbol{E}(x), \tag{2a}$$

$$\hat{\mathcal{H}}(x, t) = e^{-i\omega t} \boldsymbol{H}(x), \tag{2b}$$

$$\mathcal{F}(x, t) = e^{-i\omega t} \boldsymbol{F}(x). \tag{2c}$$

Replacing (2a) and (2b) into (1c) and (1d), we obtain

$$-\nabla \times \boldsymbol{H} - i\omega\zeta \boldsymbol{E} = \mathbf{F}(x) \quad \text{in } \Omega, \tag{3a}$$

$$\nabla \times \boldsymbol{E} - i\omega\beta \boldsymbol{H} = 0 \qquad \text{in } \Omega, \tag{3b}$$

where $\zeta := \alpha + i\chi/\omega$. Also, we consider a perfect conduction boundary condition (we surround Ω by a perfect bounded material i.e., $E \times \boldsymbol{n} = 0$). Therefore, we have the following second-order operator for (3)

$$\boldsymbol{\mathcal{L}E} := \nabla \times (\beta^{-1}\nabla \times \boldsymbol{E}) - \kappa \boldsymbol{E} = \boldsymbol{J}_S \quad \text{in } \Omega, \tag{4}$$

where $\kappa := \omega^2\zeta$ and $\boldsymbol{J}_S := -i\omega \boldsymbol{F}$.

2.1 Discretization by edge elements

To present a Galerkin weak formulation for (4), we denote by $\boldsymbol{L}^2(\Omega)$ as the space of vector field with three entries from $L^2(\Omega)$, i.e.,

$$\boldsymbol{L}^2(\Omega) := \left\{\mathbf{U} = (U_1, U_2, U_3) \colon U_i \in L^2(\Omega),\ i = 1, 2, 3\right\},$$

with $\langle \cdot, \cdot \rangle$ as the inner product on this space, and continue with defining the following space

$$\boldsymbol{H}(\mathrm{curl}, \Omega) := \left\{\mathbf{U} \in \boldsymbol{L}^2(\Omega) \colon \nabla \times \mathbf{U} \in \boldsymbol{L}^2(\Omega)\right\},$$

equipped with the norm

$$\|\mathbf{U}\|^2_{\boldsymbol{H}(\mathrm{curl},\Omega)} := \|\mathbf{U}\|^2_{\boldsymbol{L}^2(\Omega)} + \|\nabla \times \mathbf{U}\|^2_{\boldsymbol{L}^2(\Omega)}.$$

Considering homogeneous Dirichlet boundary conditions, the space $\boldsymbol{H}_0(\mathrm{curl}, \Omega) \subset \boldsymbol{H}(\mathrm{curl}, \Omega)$ is introduced as follows

$$\boldsymbol{H}_0(\mathrm{curl}, \Omega) := \{\mathbf{U} \in \boldsymbol{L}^2(\Omega) \colon \nabla \times \mathbf{U} \in \boldsymbol{L}^2(\Omega),\ \mathbf{U} \times \mathbf{n} = 0 \text{ on } \Gamma\}.$$

Then, the weak formulation for (4) can be written as : find $\boldsymbol{E} \in \boldsymbol{H}_0(\mathrm{curl}, \Omega)$

$$a(\boldsymbol{E}, \Phi) := \langle \nabla \times \boldsymbol{E}, \nabla \times \Phi \rangle - \kappa \langle \boldsymbol{E}, \Phi \rangle = \langle \boldsymbol{J}_S, \Phi \rangle \qquad \forall\, \Phi \in \boldsymbol{H}_0(\mathrm{curl}, \Omega). \tag{5}$$

Here, we should note that κ is not an eigenvalue of the operator $\nabla \times \nabla \times$ [13, Corollary 4.19], in addition we have $\kappa \neq 0$, and we set $\beta = 1$. The existence of the unique solution for the variational formulation (5) is proven in [10, Thm. 5.2], and the following *a priori* estimate is obtained

$$\|\boldsymbol{E}\|_{\boldsymbol{H}(\mathrm{curl},\Omega)} \leq C^* \|\boldsymbol{J}_S\|_{L^2(\Omega)}, \tag{6}$$

where C^* depends on Ω as well as κ.

For the discretization, we consider quasi-uniform mesh simplices $\mathcal{T} = \{T_1, \ldots, T_{N_\mathcal{T}}\}$, where $T_j \in \mathcal{T}$ are open elements and denote $h := \max_{T_j \in \mathcal{T}} \mathrm{diam}(T_j)$. We assume $\mathcal{T}$ is a Ciarlet-regular mesh, i.e., it does not contain any hanging nodes. We also assume there

is $\gamma > 0$ such that $\operatorname{diam}(T_\ell) \le \gamma\,|T_\ell|^{1/3}$ for all $T_\ell \in \mathcal{T}$. In order to present the Galerkin FEM for (5), we consider lowest order Nédélec's curl-conforming elements of the first kind, i.e.,

$$\begin{aligned}\mathbf{V}_h &:= \{\mathbf{v}_h \in \boldsymbol{H}(\operatorname{curl},\Omega)\colon \mathbf{v}_h|_T \in \mathcal{N}_0(T) \quad \forall\, T \in \mathcal{T}\},\\ \mathbf{V}_{h,0} &:= \mathbf{V}_h \cap \boldsymbol{H}_0(\operatorname{curl},\Omega),\end{aligned}$$

where for all $T \in \mathcal{T}_h$, the lowest order local Nédélec space of the first kind $\mathcal{N}_0(T)$ is defined as [13]

$$\mathcal{N}_o(T) = \{\boldsymbol{a} + \boldsymbol{b} \times \mathbf{x}\colon \boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^3, \quad \mathbf{x} \in T\}.$$

We denote $\mathcal{Y}_h := \{\Phi_1, \ldots, \Phi_N\}$ as a basis with N as the dimension of $\mathbf{V}_{h,0}$. This basis is uniquely defined by the property $\sigma(\Phi_i, e_j) = \delta_{ij}$, where e_j denotes an interior edge of the mesh and $\sigma(p, e)$ is the line integral of the tangential component of p along e. Then, the Galerkin FEM for (5) is given as: Find $\boldsymbol{E}_h \in \mathbf{V}_{h,0}$ such that

$$a(\boldsymbol{E}_h, \Phi_h) = \langle \boldsymbol{J}_S, \Phi_h \rangle \quad \forall \Phi_h \in \mathbf{V}_{h,0}. \tag{7}$$

This is equivalent to solve the following system

$$\mathbf{A}x = b, \quad \text{where} \quad \mathbf{A} = [\mathbf{A}_{ij}]_{i,j=1}^{N} \quad \text{with} \quad \mathbf{A}_{ij} := a(\Phi_i, \Phi_j), \quad \Phi_j,\, \Phi_i \in \mathcal{Y}_h, \tag{8}$$

and the right hand side vector b is defined as $b_j := \left\langle \boldsymbol{J}_S, \Phi_j \right\rangle$, $j \in \{1, 2, \ldots, N\}$.

3 $\mathcal{H}$-matrices and $\mathcal{H}$-matrix arithmetic

$\mathcal{H}$-matrices are defined based on a partition P generated from a clustering algorithm that allows us to determine which blocks can be approximated by low-rank matrices or are small [7].

Applying $\mathcal{H}$-matrix approximations allows us to store large matrices in the format of low-rank block-wise matrices which could lead to logarithmic-linear storage complexity provided that a proper method is used to define the hierarchical structure that results in the final block-wise format of the matrix. In the following lemma from [5], it is shown that the inverse of the Galerkin matrix $\mathbf{A}$ (8) can be approximated using an $\mathcal{H}$-matrix, and proven that this approximation converges exponentially with respect to the maximum block rank to $\mathbf{A}$.

Definition 1 *[$\mathcal{H}$-matrices] A matrix $\mathbf{B}_{\mathcal{H}} \in \mathbb{C}^{N\times N}$ is called an $\mathcal{H}$-matrix, if for every admissible block (τ, σ), we have the following factorization*

$$\mathbf{B}_{\mathcal{H}}|_{\tau\times\sigma} = \mathbf{X}_{\tau\sigma}\mathbf{Y}_{\tau\sigma}^{H},$$

of rank r where $\mathbf{X}_{\tau\sigma} \in \mathbb{C}^{\tau\times r}$ and $\mathbf{Y}_{\tau\sigma} \in \mathbb{C}^{\sigma\times r}$.

In order to use the inverse of the $\mathcal{H}$-matrix approximation of $\mathbf{A}$ as a preconditioner, first, we need to find an $\mathcal{H}$-matrix approximation for $\mathbf{A}$, then we obtain the inverse using the iterative method of Schulz [8].

Lemma 1 ([5]) *Let $\boldsymbol{A}$ be the Galerkin matrix defined in* (8)*. Then, there exists an $\mathcal{H}$-matrix approximation $\mathbf{B}_{\mathcal{H}}$ with the maximum block rank r (rank of all the blocks of $\mathbf{B}_{\mathcal{H}}$ is either smaller than or equal to r) such that*

$$\left\|\mathbf{A}^{-1} - \mathbf{B}_{\mathcal{H}}\right\|_2 \leq \bar{C} h^{-1} e^{-c\left(r^{1/4}/\ln r\right)},$$

where $\bar{C}$ and c are constants depending only on material parameters, and the properties of Ω.

In the definition of $\mathcal{H}$-matrices, the low-rank blocks are determined based on the concept of η-admissibility defined in [7]. In the following, the mathematical definition of an $\mathcal{H}$-matrix is given.

Although computing the inverse of the $\mathcal{H}$-matrix approximation of the Galerkin system matrix leads to logarithmic-linear complexity, the computational cost to solve the linear system directly is still too high. Thus, we need to use another alternative to reduce the numerical cost such as the $\mathcal{H} - LU$ factorization. In the following, we present an algorithm from [1, Sec. 2.9], and use it as a preconditioner to solve the linear systems.

Algorithm 1 $\mathcal{H} - LU$ decomposition and application in preconditioning of a simple iterative solver for (7).

1) Compute the $\mathcal{H}$-matrix approximation of $\mathbf{A}$, i.e., $\mathbf{A}_{\mathcal{H}}$

2) Compute the $\mathcal{H}$- matrix LU decomposition of $\mathbf{A}_{\mathcal{H}}$ as follows

1) **for** $j = 1, \ldots, N$

for $k = 1, \ldots, j-1$

Solve the system $\sum_{i=1}^{k} L_{ji}\, U_{ik} = (\mathbf{A}_{\mathcal{H}})_{jk}$ to get L_{jk}.

2) Compute L_{jj} and U_{jj} by $\quad L_{jj}U_{jj} = A_{jj} - \sum_{i=1}^{j-1} L_{ji}U_{ij}$.

3) **for** $k = j+1, \ldots, N$

Solve the system $\sum_{i=1}^{j} L_{ji}\, U_{ik} = (\mathbf{A}_{\mathcal{H}})_{jk}$ to get U_{jk}.

3) **for** $i = 1, \ldots,$ MaxIt

Compute $r = b - \mathbf{A}\, x$ and $\texttt{err}=\|r\|_2 / \|b\|_2$

Compute $x = x + U^{-1}\left(L^{-1}\, r\right)$.

if $\texttt{err} < \texttt{TOL}$ **break**

4) Compute $\texttt{error} = \|\mathbf{A}\, x - b\|_2$.

4 Main algorithm and numerical experiments

In this section, we first present the main algorithm of this work, and then we study three numerical examples to solve the linear system arising from the Maxwell equations using an $\mathcal{H}$-matrix approximation as a preconditioner. For this, we employ the geometrically balanced cluster tree presented in [6], and we set the admissibility parameter $\eta = 2$. We use a truncated singular value decomposition (SVD) with different ranks r to compute $\mathbf{B}_{\mathcal{H}}$ as the inverse of $\mathbf{A}_{\mathcal{H}}$ obtained from the Schulz iteration. In other words, for admissible blocks (τ, σ), we have $\mathbf{A}_{\mathcal{H}}|_{\tau\times\sigma} := \mathbf{U}_r\mathbf{S}_r\mathbf{V}_r^T$ where $\mathbf{U}_r \in \mathbb{R}^{\tau\times r}$, $\mathbf{S}_r \in \mathbb{R}^{r\times r}$ and $\mathbf{V}_r \in \mathbb{R}^{\sigma\times r}$ are the first r columns of $\mathbf{U}$, $\mathbf{S}$ and $\mathbf{V}$, respectively. Then, we find the inverse $\mathbf{B}_{\mathcal{H}}$ for the matrix $\mathbf{A}_{\mathcal{H}}$.

4.1 Example 1: a unit box

The geometry is $\Omega = (0, 1)^3$ and $\boldsymbol{J}_S = [0, 0, 1]^\top$. The coarse mesh consists of 6 tetrahedrons. This mesh is uniformly refined k times, $k = 2, \ldots, 7$. All computations are performed in MATLAB with 125 cores. Table 1 displays the iteration numbers of the preconditioned GMRES method with the described $\mathcal{H}$-matrix preconditioner for different values of κ. The GMRES method is stopped if a relative accuracy of $TOL = 10^{-5}$ of the residual is reached. In all experiments, the parameter $\beta = 1$ is chosen. After 100 iterations, we restart GMRES. The results show that the $\mathcal{H}$-matrices can be used as an efficient preconditioner if $\kappa \leq 20^2$. For higher frequencies, the iteration numbers grow in some, but not all levels k. This is due to the computation of the LU decomposition of the $\mathcal{H}$-matrix. The approximation of the original matrix by the $\mathcal{H}$-matrix is still very good, also in the case of $k = 5$ and $\kappa = 900$.

Table 1 GMRES iterations numbers for Example 1.

κ	N_{dof}	25	100	225	400	625	900
$k = 2$	98	1	1	1	1	1	1
$k = 3$	604	2	2	2	2	2	2
$k = 4$	3 184	2	4	5	5	4	3
$k = 5$	41 024	3	4	8	9	11	> 3000
$k = 6$	238 688	3	6	20	4	48	25
$k = 7$	1 807 264	4	5	7	19	46	93

4.2 Example 2: two boxes

Here the geometry consists of two boxes, i.e., $\Omega := (-1, 1) \times (-1, 1) \times [-1, -2) \cup (-2, 2) \times [1, 2) \times (-1, 1) \cup (-2, 2) \times [-1, 1] \times [-1, 1] \cup (-1, 1) \times (-1, 1) \times [1, 2) \cup (-2, 2) \times [-1, 2) \times (-1, 1)$. We set the parameters $\kappa = 1$, $\boldsymbol{J}_S = [1, 1, 1]^\top$, and $\beta = 1$. The computational domain with and the inverse of $\mathcal{H}$-matrix approximation of the stiffness matrix is shown in Figure 1. We have 24 440 admissible blocks, 48 404 small

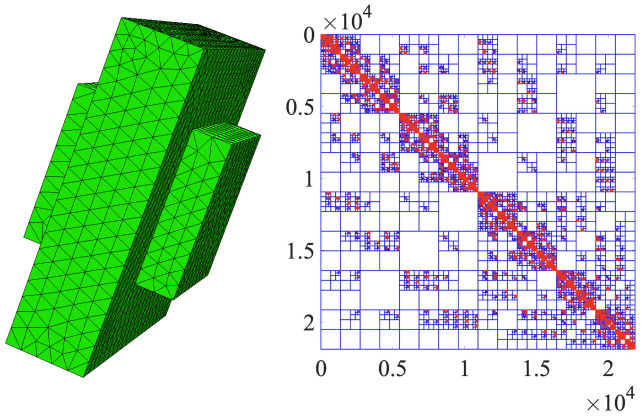


Fig. 1 Example 2. The computational geometry (left) and the inverse of $\mathcal{H}$-matrix approximation of the stiffness matrix (7) $\mathcal{H}$-matrix clustering (right) for $N_{\text{dof}} = 22\,001$.

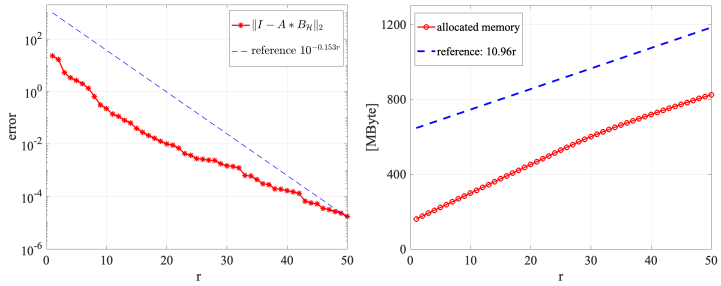


Fig. 2 Approximation error of $B_{\mathcal{H}}$ in Example 2 for a stiffness matrix with $N_{\text{dof}} = 22\,001$ (left) and the relative allocated memory (right).

blocks and the depth is 16. The decay of the approximation error versus r and the corresponding allocated memory are shown in Figure 2. As shown, using higher r gives rise to a reliable inverse of the $\mathcal{H}$-matrix approximation. The computed $\mathbf{B}_{\mathcal{H}}$ can be used as a preconditioner to solve the linear system (7) directly.

4.3 Example 3: a magnet surrounded by air

We consider a magnet surrounded by air where the box is $1 \times 1 \times 1$ and the magnet dimension is $0.5 \times 0.5 \times 0.75$. We set $\kappa = 10$, $\beta = 10$ and $\boldsymbol{J}_S = [10, 10, 10]^\top$. The geometry and the $\mathcal{H}$-approximation of $\mathbf{A}$ for $N_{\text{dof}} = 122\,202$ is shown in Figure 3. In this approximation, we have 215 964 admissible blocks, 402 451 small blocks, and the depth is 15. The $\mathcal{H}-$LU decomposition of $\mathbf{A}$ is given in Figure 4. We use Algorithm 1 for different N_{dof} to solve the linear system. Table 2 shows the results for different matrices

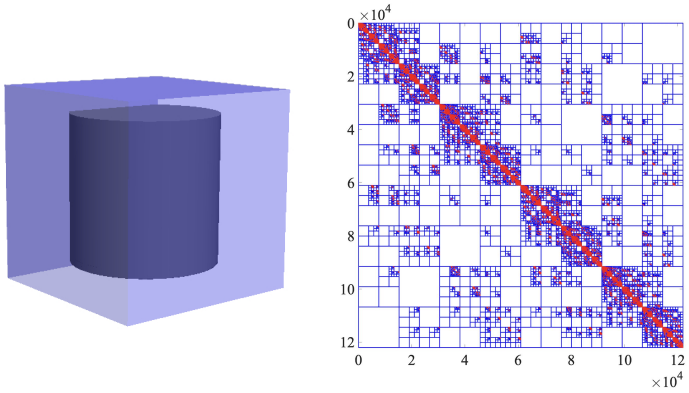


Fig. 3 Example 3. The computational geometry (left) and $\mathcal{H}$-matrix approximation of $\mathbf{A}$ for $N_{\text{dof}} = 122\,202$ (right).

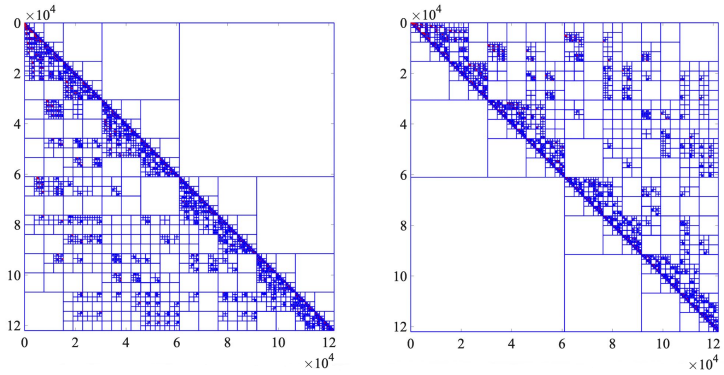


Fig. 4 The $\mathcal{H}$-LU decomposition for the stiffness matrix resulting from Example 3 corresponding to Figure 3.

Table 2 The iterative solver preconditioned by $\mathcal{H} - LU$ used to solve the Maxwell equations for different N_{dof}.

N_{dof}	5 492	8 050	13 602	33 933	48 811	70 133	78 603	96 846	129 200	304 309
`error`	7.22 e -9	1.07 e -9	5.80e-9	2.23e-8	5.11e-8	9.06e-8	5.98e-10	4.38e-10	5.04e-10	5.78e-9
time [s]	11.52	35.24	60.52	151.71	256.18	546.9	602.73	481	820.9	2335
iterations	3	3	3	3	4	4	5	5	6	7

using $TOL = 1 \times 10^{-8}$. For all cases, the negligible `error` indicates the accuracy the method, and the elapsed CPU time points out its efficiency. For the last two examples, we used *Netgen/ngsolve* [16] to produce the meshes (denoting different N_{dof}) and assembling the matrices of (7).

5 Conclusion

In this work, the $\mathcal{H}$-matrix approximation was used to solve the time-harmonic Maxwell equations. As a direct solver, the inverse of the hierarchical matrix approximation of the linear system was employed as a preconditioner, where an accurate approximation was achieved. Additionally, we then employed an $\mathcal{H} - LU$ factorization as a preconditioner.

In both cases, the use of $\mathcal{H}$ matrix approximations could reduce the computational cost and increase the accuracy of the solution. The $\mathcal{H}$ matrices can be coupled with the domain decomposition to take advantage of both approaches, i.e., to reduce the complexity and accelerate the convergence of the iterative solvers. This possibility will be addressed in the future papers.

Acknowledgements The authors acknowledge the Deutsche Forschungsgemeinschaft (DFG) under Germany Excellence Strategy within the Cluster of Excellence PhoenixD (EXC 2122, Project ID 390833453). Maryam Parvizi is funded by Alexander von Humboldt Foundation project named $\mathcal{H}$-matrix approximability of the inverses for FEM, BEM and FEM-BEM coupling of the electromagnetic problems. Finally, the authors thank Sebastian Kinnewig for fruitful discussions.

References

1. Bebendorf, M. *Hierarchical matrices*. Springer (2008).
2. Bebendorf, M. and Kramer, F. Hierarchical matrix preconditioning for low-frequency–full-maxwell simulations. *Proceedings of the IEEE* **101**(2), 423–433 (2012).
3. Bebendorf, M. and Ostrowski, J. Parallel hierarchical matrix preconditioners for the curl-curl operator. *Journal of Computational Mathematics* 624–641 (2009).
4. Ernst, O. G. and Gander, M. J. Why it is difficult to solve Helmholtz problems with classical iterative methods. 325–363. Springer (2012).
5. Faustmann, M., Melenk, J. M., and Parvizi, M. $\mathcal{H}$-matrix approximability of inverses of FEM matrices for the time-harmonic Maxwell equations. *Advances in Computational Mathematics* **48**(5) (2022).
6. Grasedyck, L., Hackbusch, W., and Borne, S. L. Adaptive geometrically balanced clustering of $\mathcal{H}$-matrices. *Computing* **73**(1), 1–23 (2004).
7. Hackbusch, W. *Hierarchical matrices: algorithms and analysis*, vol. 49. Springer (2015).
8. Hackbusch, W. and Khoromskij, B. N. A sparse $\mathcal{H}-$matrix arithmetic: general complexity estimates. *Journal of Computational and Applied Mathematics* **125**(1-2), 479–501 (2000).
9. Henneking, S. and Demkowicz, L. A numerical study of the pollution error and dpg adaptivity for long waveguide simulations. *Computers & Mathematics with Applications* **95**, 85–100 (2021).
10. Hiptmair, R. Finite elements in computational electromagnetism. *Acta Numerica* **11**, 237–339 (2002).
11. Liu, J. W. The multifrontal method for sparse matrix solution: Theory and practice. *SIAM Review* **34**(1), 82–109 (1992).
12. Melenk, J. M. and Sauter, S. Wavenumber explicit convergence analysis for galerkin discretizations of the helmholtz equation. *SIAM Journal on Numerical Analysis* **49**(3), 1210–1243 (2011).
13. Monk, P. et al. *Finite element methods for Maxwell's equations*. Oxford University Press (2003).
14. Ostrowski, J. M., Bebendorf, M., Hiptmair, R., and Krämer, F. $\mathcal{H}$-matrix-based operator preconditioning for full maxwell at low frequencies. *IEEE Transactions on Magnetics* **46**(8), 3193–3196 (2010).
15. Saad, Y. *Iterative methods for sparse linear systems*. SIAM (2003).
16. Schöberl, J. et al. Netgen/ngsolve. *https://ngsolve.org* (2017).

Convergence Bounds for Parareal with Spatial Coarsening

Aušra Pogoželskytė and Martin J. Gander

1 Introduction

In the times of exascale computing, very efficient algorithms for space parallelism exist and communication between processors has become a bottleneck; parallelism in the time dimension is necessary. The Parareal algorithm [10] allows us to do this in a non-intrusive fashion. However, it is limited by the sequential nature of its coarse operator. A solution would be to consider a multilevel version of Parareal such as Multigrid Reduction in Time (MGRIT) [3], but it also suffers from scalability issues due to coarsening only in the time dimension. There are some algorithms that are scalable such as the Space-Time Multigrid algorithm of Gander and Neumüller [8], but they are more intrusive.

Parareal with spatial coarsening offers a solution that would be both scalable and non-intrusive. It has been first introduced in [4] in order to preserve the stability of the scheme while solving the Navier-Stokes equations, but it has only been numerically studied in [11].

Many bounds for Parareal exist [5, 7, 9], but they do not provide useful information in the case where spatial coarsening is used, or have assumptions that cannot be satisfied such as simultaneous diagonalizability in [2, 12]. We present here two convergence bounds for Parareal with spatial coarsening, and illustrate our results using the heat equation.

Aušra Pogoželskytė, Martin J. Gander
University of Geneva, rue du Conseil-Général 7-9, Geneva, e-mail: ausra.pogozelskyte@unige.ch, martin.gander@unige.ch

2 Parareal with spatial coarsening

Let $A \in \mathbb{R}^{n_x \times n_x}$ be a matrix coming from a uniform n_x point spatial discretization of a partial differential equation and define $f\colon \mathbb{R} \to \mathbb{R}^{n_x}$ to be a source term. Consider the initial value problem,

$$y'(t) = A\,y(t) + f(t),\ t \in (0,T], \quad y(0) = y_0\ . \tag{1}$$

Note that the assumptions on the linearity of the problem and the time independence of A are only necessary to facilitate the analysis. The problem (1) can be then discretized in time on a uniform $n_t + 1$ point mesh using a one-step method. This results in the time-stepping iteration,

$$y_{n+1} = \Phi\, y_n + f_n, \quad n = 0, \ldots, n_t\ . \tag{2}$$

Now, coarsen the spatial and time mesh by a factor of m_x and m_t such that the fine mesh can be seen as a refinement of the coarse mesh. Define u and g to be the restrictions of y and f to the coarse grid. Proceeding in the same way, a time-stepping iteration can be defined on the coarse mesh,

$$u_{n+1} = \Psi\, u_n + g_n, \quad n = 0, \ldots, n_t/m_t\ , \tag{3}$$

where Ψ is an operator depending on the coarse time-step and the spatial discretization matrix of the coarse problem. Define also the operators $F := \Phi^{m_t}$ and $G := \Psi$.

Example: Discretization of the heat equation

Consider the one-dimensional heat equation,

$$\begin{cases} \partial_t y(x,t) = c^2 \partial_{xx} y(x,t) + f(x,t)\ , & (x,t) \in (0,L) \times (0,T], \\ y(x,0) = y_0(x)\ , & x \in (0,L), \\ y(0,t) = y(L,t) = 0\ , & t \in (0,T]\ . \end{cases} \tag{4}$$

Denote $\{x_i\}_{i=0}^{n_x}$ and $\{X_j\}_{j=0}^{N_x}$ to be fine and coarse discretizations of $[0,L]$ such that $x_i = i\Delta x$ and $X_j = j m_x \Delta x$ where $\Delta x = L/n_x$ and $n_x = m_x N_x$. This leads to a system of the form (1) where A and A_Δ are discrete Laplacians on the fine and coarse grids,

$$A := \frac{c^2}{\Delta x^2} \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{pmatrix}, \quad A_\Delta := \frac{c^2}{(m_x \Delta x)^2} \begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{pmatrix}.$$

Using a Runge-Kutta method whose stability function is given by R, one gets the operators $F = R(\Delta t A)^{m_t}$ and $G = R(m_t \Delta t A_\Delta)$.

The time stepping (2) can be parallelized in the time dimension using the Parareal algorithm with spatial coarsening. It was first introduced by Fischer, Hecht and Maday in [4] in order to preserve the stability of their numerical scheme when solving the Navier-Stokes equations using Parareal as well as reducing the serial overhead of the coarse operator. Define $I \in \mathbb{R}^{n_x \times N_x}$ and $R \in \mathbb{R}^{N_x \times n_x}$ to be interpolation and restriction operators. The algorithm is initialized as

$$Y_0^0 = y_0 \,, \quad Y_n^0 = (I\,G\,R) Y_{n-1}^0 \,, \qquad n = 1, \ldots, N_t \tag{5}$$

and its kth iteration is given by

$$Y_0^k = y_0 \,, \quad Y_n^k = F\,Y_{n-1}^{k-1} - (I\,G\,R) Y_{n-1}^{k-1} + (I\,G\,R) Y_{n-1}^k \,, \quad n = 1, \ldots, N_t \tag{6}$$

Remark 1 The iteration (6) is the same as traditional Parareal, except that the coarse operator $\widetilde{G} := IGR$ is used. This means, in particular, that properties such as finite time convergence [6, Theorem 5] are conserved. The finite time convergence property states that the algorithm will converge in the worst case at iteration $k = N_t$.

3 Analysis of parareal with spatial coarsening

To get insight on the convergence of the algorithm, we will investigate how it behaves for each spatial mode of the error. Therefore, the spatial dimension will be transformed in a Fourier basis, leaving the time dimension as is. For simplicity of the analysis, we will assume that the coarsening factor in space is $m_x = 2$.

3.1 Derivation of the error propagation operator

The first step of the analysis is to derive the error propagation operator. Let Y_n be the fine solution of (1) and $e_n^k = Y_n - Y_n^k$ denote the error of the Parareal algorithm. Then, the error verifies the recurrence relation

$$\begin{aligned} e_{n+1}^{k+1} &= F\,Y_n - F\,Y_n^k + (I\,G\,R) Y_n^k - (I\,G\,R) Y_n^{k+1} + (I\,G\,R) Y_n - (I\,G\,R) Y_n \\ &= (F - I\,G\,R)\, e_n^k + (I\,G\,R)\, e_n^{k+1} \,. \end{aligned} \tag{7}$$

Let $\boldsymbol{e}^k := [e_0^k, \ldots, e_{n_t}^k]^\top$ be the vector of the errors at iteration k for all time steps. Define Γ_{-i} to be a matrix with ones on its ith subdiagonal and zeros elsewhere. Then, the relation (7) can be written as the linear system

$$(I_t \otimes I_x - \Gamma_{-1} \otimes (I\,G\,R))\, \boldsymbol{e}^{k+1} = (\Gamma_{-1} \otimes (F - I\,G\,R))\, \boldsymbol{e}^k \,, \tag{8}$$

which can be solved to get the iteration $\boldsymbol{e}^{k+1} = E\,\boldsymbol{e}^{k}$. The matrix E is called the *error propagation operator* and describes the evolution of the error between two iterations for all time (and space) steps.

Lemma 1 *The error propagation operator E is given by*

$$E = \left(\sum_{i=0}^{n_t-1} \Gamma_{-i-1} \otimes (I\,G\,R)^i\right)(I_t \otimes (F - I\,G\,R))\ .$$

Proof The result is obtained by direct computation, see [1, Lemma 2.1] for similar computations. □

3.2 Analysis of the error propagation operator in Fourier space

Let $U \in \mathbb{R}^{n_x \times n_x}$ be a basis that diagonalizes the fine operator F. The columns of the matrix U will be referred to as *modes* and can be further split into *low* and *high* frequency modes $\check{u}_\kappa$ and $\hat{u}_\kappa$. Low modes are defined to be the modes that can be well represented on the considered mesh, and high modes those who can not.

When considering coarsening in space, the issue of *aliasing* arises, see Figure 1. That means that there are pairs of low and high frequency modes $\{\check{u}_\kappa, \hat{u}_\kappa\}$ that are mapped to the same coarse mode u_κ on the coarse grid. Those pairs are said to be *harmonic* of each other. We can define the *space of harmonics* as the space spanned by pairs of such modes (see [13] for a more detailed discussion).

Assume that the transfer operators R and I keep the spaces of harmonics invariant[1]. Then, there exist symbols $\check{\iota}_\kappa$, $\hat{\iota}_\kappa$, $\check{r}_\kappa$ and $\hat{r}_\kappa$ such that

$$R\check{u}_\kappa = \check{r}_\kappa u_\kappa\ , \quad R\hat{u}_\kappa = \hat{r}_\kappa u_\kappa\ , \quad Iu_\kappa = \check{\iota}\check{u}_\kappa + \hat{\iota}\hat{u}_\kappa\ .$$

Likewise, under our hypothesis on the coarse grid, the coarse modes u_κ will diagonalize the operator G,

$$Gu_\kappa = \mu_\kappa u_\kappa, \quad F\check{u}_\kappa = \check{\lambda}_\kappa \check{u}_\kappa, \quad F\hat{u}_\kappa = \hat{\lambda}_\kappa \hat{u}_\kappa\ .$$

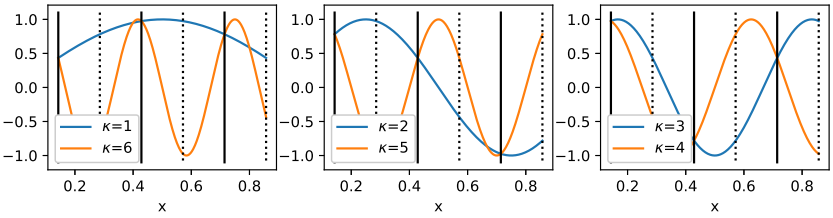


Fig. 1 Example of aliasing for the heat equation with $n_x = 7$. Points on the fine and coarse grid intersect the dotted and continuous lines. One can observe pairs of modes: one low (blue) and one high (orange), that get mapped to a single coarse mode on the coarse grid.

[1] This is true for common Multigrid transfer operators, see [13].

Example: Modes and Fourier symbols for the heat equation

For the heat equation (4), due to Dirichlet boundary conditions, the fine operator is diagonalized by a basis of sines, thus the columns of U are given by

$$u_\kappa(x) = \sin\left(\frac{\kappa\pi x}{L}\right) , \quad x \in \{x_j\}_{j=0}^{n_x}, \quad \kappa = 1, \dots, n_x .$$

The corresponding low (coarse) and high modes are for $\kappa = 1, \dots, N_x$

$$\check{u}_\kappa(x) = \sin\left(\frac{\kappa\pi x}{L}\right) \equiv u_\kappa(x) , \quad \hat{u}_\kappa(x) = \sin\left(\frac{(n_x-\kappa)\pi x}{L}\right) \equiv u_{n_x-\kappa}(x) .$$

Consider linear interpolation and full-weighting restriction, defined as

$$(R_{FW}u)_j = \frac{1}{4}(u_{2j-1} + 2u_{2j} + u_{2j+1}), \quad j = 0, \dots, n/2-1 ,$$
$$(I_{lin}v)_j = \begin{cases} v_{j/2} & \text{if } j \text{ is even,} \\ \frac{1}{2}(v_{(j-1)/2} + v_{(j+1)/2}) & \text{if } j \text{ is odd.} \end{cases}$$

Then the associated Fourier symbols are given by

$$\check{\iota}_\kappa = \check{r}_\kappa = \frac{1}{2}(1 + \cos(\pi x/L)) , \quad \hat{\iota}_\kappa = \hat{r}_\kappa = \frac{1}{2}(1 - \cos(\pi x/L)) .$$

Lemma 2 *The error propagation operator keeps the space of harmonics invariant.*

Proof By Lemma 1, the error propagation operator depends on the fine and coarse propagators F and G as well as on the transfer operators R and I, and powers thereof. Indeed,

$$IGR\check{u}_\kappa = IG\check{r}_\kappa u_\kappa = I\mu_\kappa\check{r}_\kappa u_\kappa = \check{\iota}_\kappa\mu_\kappa\check{r}_\kappa\check{u}_\kappa + \hat{\iota}_\kappa\mu_\kappa\check{r}_\kappa\hat{u}_\kappa,$$
$$IGR\hat{u}_\kappa = IG\hat{r}_\kappa u_\kappa = I\mu_\kappa\hat{r}_\kappa u_\kappa = \check{\iota}_\kappa\mu_\kappa\hat{r}_\kappa\check{u}_\kappa + \hat{\iota}_\kappa\mu_\kappa\hat{r}_\kappa\hat{u}_\kappa.$$

This can be summarized as

$$IGR[\check{u}_\kappa, \hat{u}_\kappa] = [\check{u}_\kappa, \hat{u}_\kappa]\mu_\kappa \begin{pmatrix} \check{\iota}_\kappa\check{r}_\kappa & \check{\iota}_\kappa\hat{r}_\kappa \\ \hat{\iota}_\kappa\check{r}_\kappa & \hat{\iota}_\kappa\hat{r}_\kappa \end{pmatrix} =: [\check{u}_\kappa, \hat{u}_\kappa]\mu_\kappa\Pi_\kappa . \tag{9}$$

Similarly, we can write an analog expression for the fine operator F,

$$F[\check{u}_\kappa, \hat{u}_\kappa] = [\check{u}_\kappa, \hat{u}_\kappa] \begin{pmatrix} \check{\lambda}_\kappa & 0 \\ 0 & \hat{\lambda}_\kappa \end{pmatrix} =: [\check{u}_\kappa, \hat{u}_\kappa]\Lambda_\kappa . \tag{10}$$

Thus, as the space of harmonics is invariant for the operators IGR and F, it is also invariant for the error propagation operator. □

Lemma 2 says that the error propagator E can be transformed in the Fourier space to an operator that is diagonal by blocks, where each block is given by

$$\widetilde{E}_\kappa = \left(\sum_{i=0}^{n_t-1} \Gamma_{-i-1} \otimes (\mu_\kappa \Pi_\kappa)^i \right) (I_t \otimes (\Lambda_\kappa - \mu_\kappa \Pi_\kappa)) \ . \tag{11}$$

Denote by $\tilde{e}_n^k(\kappa)$ the error at time-step n and iteration k of Parareal in Fourier space for the wavenumber κ and its corresponding high frequency mode. Using the relationship (11), we can get a relationship on the pairs of harmonics that can be bounded in the 2-norm as

$$\|\tilde{e}_{n+1}^{k+1}(\kappa)\|_2 = \|\Lambda_\kappa - \mu_\kappa \Pi_\kappa\|_2 \|\tilde{e}_n^k(\kappa)\|_2 + \|\mu_\kappa \Pi_\kappa\|_2 \, \|\tilde{e}_n^{k+1}(\kappa)\|_2 \ . \tag{12}$$

To shorten the notation, we will omit κ in $\tilde{e}_n^k(\kappa)$ and refer to it as $\tilde{e}_n^k$, and all the norms in what follows will be 2-norms, i.e. $\|\cdot\| := \|\cdot\|_2$.

Theorem 1 (Linear bound)

Let Λ_κ and $\mu_\kappa \Pi_\kappa$ be defined as in (9) *and* (10)*. Then, for a given wavenumber κ, the error of Parareal with spatial coarsening in Fourier space is bounded as*

$$\max_{1 \le n \le N_t} \|\tilde{e}_n^{k+1}\| \le \frac{1 - \|\mu_\kappa \Pi_\kappa\|^{N_t}}{1 - \|\mu_\kappa \Pi_\kappa\|} \|\Lambda_\kappa - \mu_\kappa \Pi_\kappa\| \max_{1 \le n \le N_t} \|\tilde{e}_n^k\| \ . \tag{13}$$

Proof Let $\alpha = \|\Lambda_\kappa - \mu_\kappa \Pi_\kappa\|_2$ and $\beta = \|\mu_\kappa \Pi_\kappa\|_2$. Using iteration (12) and iterating on the term $\|\tilde{e}_n^{k+1}\|$ yields

$$\|\tilde{e}_{n+1}^{k+1}\| \le \alpha \|\tilde{e}_n^k\| + \beta\big(\alpha \|\tilde{e}_{n-1}^k\| + \beta \|\tilde{e}_{n-1}^{k+1}\|\big) \le \ldots \le \sum_{i=0}^{n-1} \beta^i \alpha \|\tilde{e}_{n-i}^k\| + \beta^n \|\tilde{e}_0^{k+1}\| \ .$$

As the error at the initial time step is $e_0^{k+1} = 0$, we have $\tilde{e}_0^{k+1} = 0$. Taking the maximum over n and using the closed form of geometric series concludes the proof.

Remark 2 One can notice that (13) is very similar to the bound found in [2] where no spatial coarsening is considered. The only addition is this matrix Π_κ which accounts for the effect of transfer operators on the Fourier modes.

Remark 3 Note that in Theorem 1 the inequality step is due to the submultiplicative property of the matrix norm and the fact that

$$\|(\mu_\kappa \Pi_\kappa)^i\| \le \|\mu_\kappa \Pi_\kappa\|^i \ . \tag{14}$$

However, the inequality (14) is an equality, as Π_κ is a rank one matrix. Indeed, there exists a matrix Π_0 such that $\Pi^k = \mathrm{tr}(\Pi)^k \Pi_0$. In our case, $\Pi_0 := \frac{1}{\mathrm{tr}(\Pi)} \Pi$.

Lemma 3 (Theorem 2.10 in [7])

Let $\alpha, \beta \in \mathbb{R}$, then the recurrence relation $e_{n+1}^{k+1} \leq \alpha e_n^k + \beta e_n^{k+1}$ implies

$$e_{n+1}^k \leq \frac{\alpha^k}{(k-1)!} \sum_{i=0}^{n-k} \prod_{l=1}^{k-1} (i+l) \beta^i \max_n e_n^0 \,. \tag{15}$$

Theorem 2 (Superlinear bound)

Let Λ_κ and $\mu_\kappa \Pi_\kappa$ be defined as in (9) *and in* (10)*. Then, for a given wavenumber κ, the error of Parareal with spatial coarsening in Fourier space is bounded by*

$$\|\tilde{e}_{n+1}^k\| \leq \frac{\|\Lambda_\kappa - \mu_\kappa \Pi_\kappa\|^k}{(k-1)!} \sum_{i=0}^{N_t-k} \prod_{l=1}^{k-1} (i+l) \, \|\mu_\kappa \Pi_\kappa\|^i \max_n \|\tilde{e}_n^0\| \,.$$

Proof From (12) and setting $\alpha = \|\Lambda_\kappa - \mu_\kappa \Pi_\kappa\|$ and $\beta = \|\mu_\kappa \Pi_\kappa\|$, we obtain the relationship in Lemma 3, which concludes the proof. □

Remark 4 As we have a bound for the error in Fourier space, we can deduce a bound for the error in the real space e_n^k. Indeed, by the discrete equivalent of the Perseval theorem, we get

$$\|e_n^k\|_2^2 = \frac{1}{n_x} \sum_{\kappa=1}^{n_x} |\tilde{e}_n^k|^2 = \frac{1}{m_x N_x} \sum_{\kappa=1}^{N_x} (|\check{e}_n^k|^2 + |\hat{e}_n^k|^2) = \frac{1}{m_x N_x} \sum_{\kappa=1}^{N_x} \|\tilde{e}_n^k\|_2^2 \,,$$

where the second equality is obtained by splitting the error modes into high and low frequencies, and the third by grouping them into pairs. In particular,

$$\|e_n^k\|^2 \leq \frac{1}{m_x} \max_\kappa \|\tilde{e}_n^k\|^2 \,.$$

We show in Figure 2 a numerical illustration of the linear and superlinear bounds from Theorem 1 and Theorem 2. We see that our convergence bounds for Parareal with

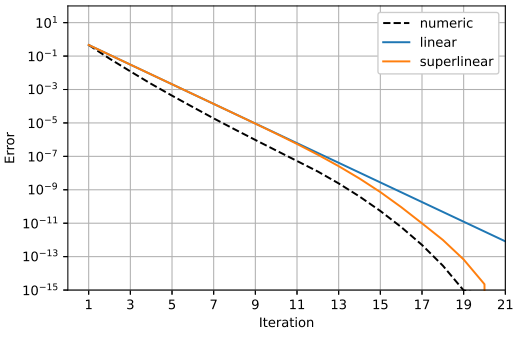


Fig. 2 Numerical error for Parareal with spatial coarsening, and linear (Theorem 1) and superlinear (Theorem 2) convergence bounds. The numerical errors are measured in the 2-norm in space and ∞-norm in time when solving the heat equation with $n_x = n_t = 200$, $m_x = 2$, $m_t = 10$ on $[0, 10^{-1}]$.

spatial coarsening are indeed capturing the convergence behavior of the algorithm. Our results are thus an important step for fully understanding Parareal with spatial coarsening. In particular, they emphasize the crucial role of transfer operators in the convergence of the algorithm, through the matrix Π.

Acknowledgements This project has received funding from the Federal Ministry of Education and Research and the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955701, Time-X. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Belgium, France, Germany, Switzerland.

References

1. Angel, J., Götschel, S., and Ruprecht, D. Impact of spatial coarsening on Parareal convergence (2021). URL http://arxiv.org/abs/2111.10228.
2. Dobrev, V. A., Kolev, T., Petersson, N. A., and Schroder, J. B. Two-Level Convergence Theory for Multigrid Reduction in Time (MGRIT). *SIAM J. Sci. Comput.* **39**(5), S501–S527 (2017).
3. Falgout, R. D., Friedhoff, S., Kolev, T. V., MacLachlan, S. P., and Schroder, J. B. Parallel Time Integration with Multigrid. *SIAM J. Sci. Comput.* **36**(6), C635–C661 (2014).
4. Fischer, P. F., Hecht, F., and Maday, Y. A Parareal in Time semi-implicit Approximation of the Navier-Stokes Equations. In: *Domain Decomposition Methods in Science and Engineering*, vol. 40, 433–440. Springer, Berlin (2005).
5. Gander, M. J. and Hairer, E. Nonlinear Convergence Analysis for the Parareal Algorithm. In: *Domain Decomposition Methods in Science and Engineering XVII*, vol. 60, 45–56. Springer, Berlin (2008).
6. Gander, M. J., Kwok, F., and Zhang, H. Multigrid interpretations of the parareal algorithm leading to an overlapping variant and MGRIT. *Comput. Visual Sci.* **19**(3), 59–74 (2018).
7. Gander, M. J., Lunet, T., Ruprecht, D., and Speck, R. A unified analysis framework for iterative parallel-in-time algorithms (2022). URL http://arxiv.org/abs/2203.16069.
8. Gander, M. J. and Neumüller, M. Analysis of a New Space-Time Parallel Multigrid Algorithm for Parabolic Problems. *SIAM J. Sci. Comput.* **38**(4), A2173–A2208 (2016).
9. Gander, M. J. and Vandewalle, S. Analysis of the Parareal Time-Parallel Time-Integration Method. *SIAM J. Sci. Comput.* **29**(2), 556–578 (2007).
10. Lions, J.-L., Maday, Y., and Turinici, G. Résolution d'EDP par un schéma en temps "pararéel". *C. R. Acad. Sci. Paris Sér. I Math.* **332**(7), 661–668 (2001).
11. Ruprecht, D. Convergence of Parareal with spatial coarsening. *PAMM* **14**(1), 1031–1034 (2014).
12. Southworth, B. S. Necessary Conditions and Tight Two-level Convergence Bounds for Parareal and Multigrid Reduction in Time. *SIAM J. Matrix Anal. Appl.* **40**(2), 564–608 (2019).
13. Trottenberg, U., Oosterlee, C. W., and Schüller, A. *Multigrid*. Academic Press, Inc., San Diego, CA (2001).

Three-Level NOSAS Preconditioners

Yi Yu and Marcus Sarkis

1 The standard NOSAS preconditioners

The nonoverlapping spectral additive Schwarz methods (NOSAS) were first introduced as two-level domain decomposition preconditioners [5, 6] designed to solve symmetric positive definite and sparse linear system $Ax = b$ arising from highly heterogeneous coefficients. NOSAS are of the nonoverlapping Schwarz type, and the subdomain interactions are via the coarse problem. NOSAS preconditioners have the following form

$$M_{\text{NOSAS}}^{-1} = \underbrace{R_0^T A_0^{-1} R_0}_{\text{Coarse Level}} + \underbrace{\sum_{i=1}^{N} R_i^T A_i^{-1} R_i}_{\text{First Level}}, \tag{1}$$

where R_i are restriction matrices from Ω to the nonoverlapping open subdomains Ω_i. We require each subdomain to be the union of elements with nodes on the boundaries of neighboring subdomains matching across the interface. $A_i = R_i A R_i^T$ are the local Dirichlet solvers on Ω_i, and A_0 is the coarse matrix corresponding to the global coarse bilinear form on the interface $\Gamma := \cup_{i=1}^{N} \Gamma_i = \cup_{i=1}^{N} (\partial\Omega_i \backslash \partial\Omega)$. A_0 can be constructed as "exact" with $A_0 = R_0 A R_0^T$ or as "inexact" with different choices of $B_{\Gamma\Gamma}^{(i)}$ to obtain better scalability property; see [6]. R_0^T is the global extension operator, which is the sum of a discrete a-harmonic extension of the low-frequency eigenfunctions and a low-cost extension for the high-frequency eigenfunctions inside each subdomain. The eigenfunctions are obtained locally and in parallel from the following generalized eigenvalue problem in each subdomain, (Cf. eq. (3.7) in [1] and eq. (7.3) in [2])

$$S^{(i)}\xi := (A_{\Gamma\Gamma}^{(i)} - A_{\Gamma I}^{(i)} (A_{II}^{(i)})^{-1} A_{I\Gamma}^{(i)})\xi = \lambda B_{\Gamma\Gamma}^{(i)} \xi, \tag{2}$$

Yi Yu
Guangxi University, Nanning, Guangxi, P. R. China, e-mail: yiyu@gxu.edu.cn

Marcus Sarkis
Worcester Polytechnic Institute, 100 Institute Rd, Worcester, USA, e-mail: msarkis@wpi.edu

where $S^{(i)}$ is the local Schur complement, $A^{(i)}_{\Gamma\Gamma}, A^{(i)}_{\Gamma I}, A^{(i)}_{I\Gamma}, A^{(i)}_{II}$ are obtained from the local Neumann matrices $A^{(i)}$, where the subscripts Γ and I denote the part associated with the interface and interior of the subdomain, respectively. The right hand side $B^{(i)}_{\Gamma\Gamma}$ is positive definite, and we have several choices [8, 7]. We can choose $B^{(i)}_{\Gamma\Gamma} = A^{(i)}_{\Gamma\Gamma}$, which is the energy of the zero extension, or $B^{(i)}_{\Gamma\Gamma} = \hat{A}^{(i)}_{\Gamma\Gamma}$ as the diagonal or block diagonal of $A^{(i)}_{\Gamma\Gamma}$. Next, a threshold η_1 is set up to decompose the space of degrees of freedom on Γ orthogonally with respect to $S^{(i)}$ and $B^{(i)}_{\Gamma\Gamma}$ into two subspaces, the low-frequency (eigenvalues smaller or equal to η_1) eigenfunctions and the high-frequency (eigenvalues larger than η_1) eigenfunctions. This decomposition defines naturally the extension R_0^T, which is the discrete a-harmonic extension for the low-frequency eigenfunctions and zero extension for the high-frequency eigenfunctions. Using $B^{(i)}_{\Gamma\Gamma}$ and low-frequency eigenfunctions we can construct a coarse bilinear form $a_0(\cdot,\cdot)$ and its corresponding matrix form A_0. $a_0(\cdot,\cdot)$ can also be interpreted as the sum of the energy of low-frequency eigenfunctions with respect to $S^{(i)}$ and the energy of high-frequency eigenfunctions with respect to $B^{(i)}_{\Gamma\Gamma}$. To obtain a better convergence rate and a smaller global problem, instead of using the zero extension for the high-frequency eigenfunctions in R_0^T, we can also use $\mathcal{H}^{(i)}_{\delta,D}$, which is the minimum a-energy extension inside a δ layer of Γ_i and zero Dirichlet condition elsewhere inside the subdomain. For this specific R_0^T, the right-hand side $B^{(i)}_{\Gamma\Gamma}$ can be chosen as $S^{(i)}_{\delta,D}$, which is the Schur complement corresponding to $\mathcal{H}^{(i)}_{\delta,D}$; or choose $B^{(i)}_{\Gamma\Gamma}$ as $S^{(i)}_F$, which is a block matrix constructed from the zero extension of vertices and Schur complement of each edge/face in a δ layer. The latter choice has excellent parallel scalability since, then, the assembling $B_{\Gamma\Gamma}$ are block diagonal with blocks related to the edges/faces only. In order to decrease the complexity of the generalized eigenvalue problems, there is also an economic version [8] by replacing the left-hand side $S^{(i)}$ with $S^{(i)}_{\delta,N}$, which is the Schur complement of the discrete a-harmonic extension in a δ layer and with zero Neumann condition inside.

For the "exact" A_0, i.e., choosing $B^{(i)}_{\Gamma\Gamma} = A^{(i)}_{\Gamma\Gamma}$ or $B^{(i)}_{\Gamma\Gamma} = S^{(i)}_{\delta,D}$, the size of the coarse problem is equal to the degrees of freedom (DOF) on the interface. However, for the "inexact" A_0, i.e., choosing $B^{(i)}_{\Gamma\Gamma} = \hat{A}^{(i)}_{\Gamma\Gamma}$ or $B^{(i)}_{\Gamma\Gamma} = S^{(i)}_F$ the coarse problem can be separated into local and global interactions by using the Sherman-Morrison-Woodbury formula [6]. The local part is based on the uncoupled $B^{(i)}_{\Gamma\Gamma}$, which corresponds to solving a small Dirichlet problem in a thin region near the edges/faces of the subdomains. The global part is based on coupled low-frequency modes across the subdomains, which are built from generalized eigenfunctions on the subdomains. The global part is designed to guarantee the robustness of the preconditioner to any ill-conditioned positive definite matrix A, and the size of the global problem is equal to the total number of selected eigenfunctions.

Now, let $V_h(\Omega)$ be any finite element space on a bounded polygonal (polyhedral) domain Ω, and the condition number of NOSAS preconditioners satisfies the following theorem. For a detailed proof, see [7, 8].

Theorem 1 *For any $u_h \in V_h(\Omega)$ the following holds:*

$$\frac{\eta_1}{C_1+1} a(u_h,u_h) \leq a(M_{NOSAS}^{-1} A u_h, u_h) \leq (C_1+1) a(u_h,u_h),$$

where C_1 is a constant based on different choices of $B_{\Gamma\Gamma}^{(i)}$.

For $B_{\Gamma\Gamma}^{(i)} = A_{\Gamma\Gamma}^{(i)}$ or $B_{\Gamma\Gamma}^{(i)} = S_{\delta,D}^{(i)}$, we have $C_1 = 1$. For $B_{\Gamma\Gamma}^{(i)} = \hat{A}_{\Gamma\Gamma}^{(i)}$, we have $A_{\Gamma\Gamma}^{(i)} \leq C_1 B_{\Gamma\Gamma}^{(i)}$ for $1 \leq i \leq N$ with $C_1 = 3$ in two dimensions and $C_1 = 4$ in three dimensions. For $B_{\Gamma\Gamma}^{(i)} = S_F^{(i)}$, we have $S_{\delta,D}^{(i)} \leq C_1 B_{\Gamma\Gamma}^{(i)}$ for $1 \leq i \leq N$ with $C_1 = 3$ in two dimensions and $C_1 = 5$ in three dimensions when $\delta < \frac{H}{2}$. Here we denote H as the size of the nonoverlapping subdomain and h as the size of the finite element. The threshold η_1 is usually chosen to be $O(\frac{h}{H})$ so that the preconditioned system has condition number $O(\frac{H}{h})$. Furthermore, the number of eigenfunctions we choose is only related to the geometry of the heterogeneous coefficients, and we give quantitative results in [6].

A unique feature of the NOSAS preconditioner is that no weighting is required to average the local solutions. This is different from methods like BDD, BDDC, FETI, and FETI-DP, which require expensive deluxe weighting for some highly heterogeneous problems. Moreover, the global matrix of the NOSAS has better sparsity than the coarse matrix of BDD or FETI, with zero blocks corresponding to the eigenfunctions in the subdomains that are not adjacent. NOSAS is constructed purely algebraically from unassembled Neumann matrices $A^{(i)}$, which facilitates the construction of the three-level NOSAS preconditioner, where we use the NOSAS idea recursively on the coarse level.

2 The three-level NOSAS preconditioners

We note that the size of the global problem for NOSAS is the total number of eigenfunctions we choose in all subdomains. Therefore, for a large number of subdomains, the coarse problem can become a bottleneck. The motivation of a three-level extension of the NOSAS methods is to approximate the coarse problem by replacing the direct solver with a new preconditioner; see the three-level BDDC method [4] and the three-level GDSW preconditioner [3]. We can also further recursively apply the preconditioners to new levels, which is algorithmically straightforward leading to multilevel extensions.

We first introduce some notations to define our three-level NOSAS. We decompose Ω into N_0 nonoverlapping open polygonal subregions $\Omega_{j,0}$ of size $O(H_0)$. We denote $W_{jk} = \overline{\Omega}_{j,0} \cap \overline{\Omega}_{k,0}$, which is the common vertex/edge/face of two adjacent subregions when not empty. We further decompose each subregion $\Omega_{j,0}$ into some subdomains Ω_i of size $O(H)$. We define $\Gamma_{j,0}$ as the interface of subregion $\Omega_{j,0}$, and define $\Gamma_{j,I}$ as the union of all subdomain interfaces Γ inside $\Omega_{j,0}$ without touching $\Gamma_{j,0}$. The global interface Γ_0 and the global interface interior Γ_I are the union of $\Gamma_{j,0}$ and $\Gamma_{j,I}$, respectively; see Figure 1 as an illustration. Therefore, we have that $\Gamma = \Gamma_0 \oplus \Gamma_I$. Unless otherwise specified, we use $V_h(D)$ to denote $\{v|_D : v \in V_h(\Omega)\}$, where D is a set in Ω.

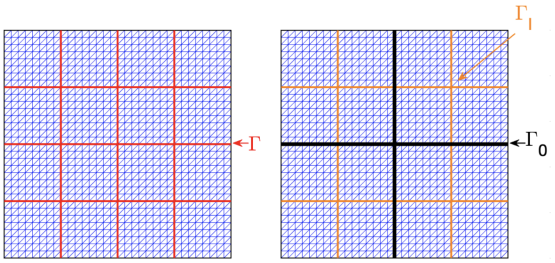


Fig. 1 Comparison of a two-level mesh (left) and a three-level mesh (right) with $h = 1/32$, $H = 1/4$, $H_0 = 1/2$.

We need to decompose the coarse space $V_0 := \{v|_\Gamma : v \in V_h(\Omega)\}$ into some new local spaces and a coarser space. Following the procedure of additive Schwarz methods, we define the local spaces $V_{j,0}$ $(1 \le j \le N_0)$ as the restriction of V_0 on $\Gamma_{j,I}$ and vanishing on $\Gamma_{j,0}$. The coarser space $V_{0,0}$ is the restriction of V_0 on Γ_0. We also define extrapolation operators $R_{j,0}^T : V_{j,0} \to V_0$ as the extension by zero outside of $\Omega_{j,0}$ for $1 \le j \le N_0$, and $R_{0,0}^T : V_{0,0} \to V_0$ will be defined later. Then V_0 admits the following direct sum decomposition

$$V_0 = R_{0,0}^T V_{0,0} \oplus R_{1,0}^T V_{1,0} \oplus \cdots \oplus R_{N_0,0}^T V_{N_0,0}.$$

Next, we follow the procedure of the NOSAS methods to define $R_{0,0}^T$. We will use only the Neumann matrices $A_0^{(i)}$ associated with the bilinear form $a_0^{(i)}(\cdot,\cdot)$. In the subregion $\Omega_{j,0}$ $(1 \le j \le N_0)$, we define the corresponding coarse bilinear form

$$a_{0,0}^{(j)}(\cdot,\cdot) = \sum_{i \in \mathcal{N}(j)} a_0^{(i)}(\cdot,\cdot),$$

where $\mathcal{N}(j)$ is the set of indices of subdomains Ω_i contained in the subregion $\Omega_{j,0}$.

Let Neumann matrices $A_{0,0}^{(j)}$ be associated with the bilinear form $a_{0,0}^{(j)}(\cdot,\cdot)$ defined above. Then, $A_{0,0}^{(j)}$ can be decomposed and written as the block matrix $\begin{bmatrix} A_{\Gamma_0\Gamma_0}^{(j)} & A_{\Gamma_0\Gamma_I}^{(j)} \\ A_{\Gamma_I\Gamma_0}^{(j)} & A_{\Gamma_I\Gamma_I}^{(j)} \end{bmatrix}$, where subscripts Γ_0, Γ_I denote the parts associated with Γ_0 and Γ_I, respectively. We consider the following local generalized eigenvalue problem in each subregion $(1 \le j \le N_0)$ separately

$$S_0^{(j)} \phi_k^{(j)} := (A_{\Gamma_0\Gamma_0}^{(j)} - A_{\Gamma_0\Gamma_I}^{(j)} (A_{\Gamma_I\Gamma_I}^{(j)})^{-1} A_{\Gamma_I\Gamma_0}^{(j)}) \phi_k^{(j)} = \mu_k^{(j)} B_{\Gamma_0\Gamma_0}^{(j)} \phi_k^{(j)}, \quad (k = 1, \cdots, n_{j,0}) \tag{3}$$

where $S_0^{(j)}$ is the Schur complement of $A_{0,0}^{(j)}$ and $n_{j,0}$ is the number of DOFs on $\Gamma_{j,0}$. Similar to the two-level NOSAS, we have the following choices for the right-hand side $B_{\Gamma_0\Gamma_0}^{(j)}$:

1. $B_{\Gamma_0\Gamma_0}^{(j)} = A_{\Gamma_0\Gamma_0}^{(j)}$;
2. $B_{\Gamma_0\Gamma_0}^{(j)} = \hat{A}_{\Gamma_0\Gamma_0}^{(j)}$, which is the diagonal or block diagonal version of $A_{\Gamma_0\Gamma_0}^{(j)}$;

3. $B^{(j)}_{\Gamma_0\Gamma_0} = S^{(j)}_{\delta_0,D}$, which is the Schur complement defined as follows
$$v_j^T S^{(j)}_{\delta_0,D} u_j = a^{(j)}_{0,0}(\mathcal{H}^{(j)}_{\delta_0,D} u_j, \mathcal{H}^{(j)}_{\delta_0,D} v_j), \quad \text{for all } u_j, v_j \in V_h(\Gamma_{j,0}).$$
Here, we set $\delta_0 = lH$ with some integer l, and $\mathcal{H}^{(j)}_{\delta_0,D}$ is defined as the minimum $a^{(j)}_{0,0}$-energy extension from $V_h(\Gamma_{j,0})$ to $V_h(\Gamma_{j,0} \bigcup \Gamma_{j,I})$ with a zero Dirichlet condition outside of a δ_0 layer from $\Gamma_{j,0}$;
4. $B^{(j)}_{\Gamma_0\Gamma_0} = S^{(j)}_W$, which is the block diagonal of $\{A^{(j)}_{W_{jk},W_{jk}}\}_{W_{jk}\in\Gamma_{j,0}}$ if W_{jk} is a vertex in 2D (vertex and edge in 3D) and $\{S^{(j)}_{W_{jk},\delta_0}\}_{W_{jk}\in\Gamma_{j,0}}$ if W_{jk} is an edge in 2D (face in 3D), where $A^{(j)}_{W_{jk},W_{jk}}$ is the submatrix of $A^{(j)}_{0,0}$ relative to W_{jk}, and
$$v_{jk}^T S^{(j)}_{W_{jk},\delta_0} u_{jk} = a^{(j)}_{0,0}(\mathcal{H}^{(j)}_{W_{jk},\delta_0} u_{jk}, \mathcal{H}^{(j)}_{W_{jk},\delta_0} v_{jk}), \quad \text{for all } u_{jk}, v_{jk} \in V_h(W_{jk}),$$
where $\mathcal{H}^{(j)}_{W_{jk},\delta_0}$ is defined as the minimum $a^{(j)}_{0,0}$-energy extension from $V_h(W_{jk})$ to $V_h(\Gamma_{j,0} \bigcup \Gamma_{j,I})$ with zero Dirichlet condition outside of a δ_0 layer from W_{jk}.

For a detailed comparison of the choices above; see [7]. Next, we solve the local generalized eigenvalue problem (3) and fix a threshold $\eta_0 < 1$. We pick the smallest k_j eigenvalues less than η_0 and their corresponding eigenvectors to construct the space $Q_0^{(j)}$ and the local orthogonal projection $\Pi^{(j)}_{0,S}\colon V_h(\Gamma_{j,0}) \to Q_0^{(j)}$ with respect to $B^{(j)}_{\Gamma_0\Gamma_0}$ norm as follows
$$Q_0^{(j)} = [\phi_1^{(j)}, \phi_2^{(j)}, \cdots, \phi_{k_j}^{(j)}], \quad \Pi^{(j)}_{0,S} = Q_0^{(j)}(Q_0^{(j)^T} B^{(j)}_{\Gamma_0\Gamma_0} Q_0^{(j)})^{-1} Q_0^{(j)^T} B^{(j)}_{\Gamma_0\Gamma_0}.$$
We also denote $\Pi^{(j)^\perp}_{0,S} = I_0^{(j)} - \Pi^{(j)}_{0,S}$, where $I_0^{(j)}\colon V_h(\Gamma_{j,0}) \to V_h(\Gamma_{j,0})$ is the identity mapping. Based on different choices of $B^{(j)}_{\Gamma_0\Gamma_0}$, we have the following choices for $R^{(j)^T}_{0,0}\colon V_h(\Gamma_{j,0}) \to V_h(\Gamma_{j,0} \bigcup \Gamma_{j,I})$:

i. $\mathcal{H}_0^{(j)}\Pi^{(j)}_{0,S} + \mathcal{E}_0^{(j)}\Pi^{(j)^\perp}_{0,S}$,
ii. $\mathcal{H}_0^{(i)}\Pi^{(j)}_{0,S} + \sum_{W_{jk}\in\Gamma_{j,0}} \mathcal{H}^{(j)}_{\delta_0,D}\Pi^{(j)^\perp}_{0,S}$,

where $\mathcal{H}_0^{(j)}$ and $\mathcal{E}_0^{(j)}$ are the minimum $a^{(j)}_{0,0}$-energy extension and zero extension from $V_h(\Gamma_{j,0})$ to $V_h(\Gamma_{j,0} \bigcup \Gamma_{j,I})$, respectively. For simplicity, we choose 1. and 2. for $B^{(j)}_{\Gamma_0\Gamma_0}$ and their corresponding option i. in $R^{(j)^T}_{0,0}$ for the rest of the paper. Therefore, $\forall u_{\Gamma_0} \in V_{0,0}$, we define $R^T_{0,0}\colon V_{0,0} \to V_0$ as
$$R^T_{0,0} u_{\Gamma_0} = \begin{bmatrix} u_{\Gamma_0} \\ \sum_{j=1}^{N_0} - R^T_{I_jI_0}(A^{(j)}_{\Gamma_I\Gamma_I})^{-1} A^{(j)}_{\Gamma_I\Gamma_0} \Pi^{(j)}_{0,S} R_{\Gamma_j\Gamma_0} u_{\Gamma_0} \end{bmatrix},$$
where $R_{\Gamma_j\Gamma_0}\colon V_{0,0} \to V_h(\Gamma_{j,0})$ and $R^T_{I_jI_0}\colon V_h(\Gamma_{j,I}) \to V_h(\Gamma_I)$ are the trivial restriction and extension operators, respectively.

Then, we define $\hat{A}_{0,0}$ corresponding to the following bilinear form

$$\hat{a}_{0,0}(u_{\Gamma_0}, v_{\Gamma_0}) = v_{\Gamma_0}^T \sum_{j=1}^{N_0} \Big((\Pi_{0,S}^{(j)} R_{\Gamma_j\Gamma_0})^T S_0^{(j)} (\Pi_{0,S}^{(j)} R_{\Gamma_j\Gamma_0}) + (\Pi_{0,S}^{(j)^\perp} R_{\Gamma_j\Gamma_0})^T B_{\Gamma_0\Gamma_0}^{(j)} (\Pi_{0,S}^{(j)^\perp} R_{\Gamma_j\Gamma_0}) \Big) u_{\Gamma_0}$$

$$= v_{\Gamma_0}^T \sum_{j=1}^{N_0} R_{\Gamma_j\Gamma_0}^T \big(B_{\Gamma_0\Gamma_0}^{(j)} - B_{\Gamma_0\Gamma_0}^{(j)} Q_0^{(j)} D_0^{(j)} (Q_0^{(j)^T} B_{\Gamma_0\Gamma_0}^{(j)} Q_0^{(j)})^{-1} Q_0^{(j)^T} B_{\Gamma_0\Gamma_0}^{(j)} \big) R_{\Gamma_j\Gamma_0} u_{\Gamma_0} \quad \forall u_{\Gamma_0}, v_{\Gamma_0} \in V_{0,0},$$

where $D_0^{(j)} = \text{diagonal}(1-\mu_1^{(j)}, 1-\mu_2^{(j)}, \cdots, 1-\mu_{k_j}^{(j)})$ and $\mu_k^{(j)}$ are the generalized eigenvalues corresponding to $\phi_k^{(j)}$.

Then, the three-level NOSAS preconditioners have the following form

$$M_{\text{3NOSAS}}^{-1} = \underbrace{R_0^T \big(\overbrace{R_{0,0}^T \hat{A}_{0,0}^{-1} R_{0,0}}^{\text{Third Level}} + \overbrace{\sum_{j=1}^{N_0} R_{j,0}^T \hat{A}_{j,0}^{-1} R_{j,0}}^{\text{Second Level}} \big) R_0}_{\text{Coarse Level}} + \underbrace{\sum_{i=1}^{N} R_i^T A_i^{-1} R_i}_{\text{First Level}}. \tag{4}$$

For the first level, $A_i = R_i A R_i^T, (1 \le i \le N)$ are the matrices corresponding to the exact local bilinear form of A, the same as in the two-level method. For the second level, $\hat{A}_{j,0} = R_{j,0} A_0 R_{j,0}^T, (1 \le j \le N_0)$ are the matrices form corresponding to the following exact local bilinear form of A_0

$$\hat{a}_{j,0}(u_{j,0}, v_{j,0}) = a_0(R_{j,0}^T u_{j,0}, R_{j,0}^T v_{j,0}), \quad \forall u_{j,0}, v_{j,0} \in V_{j,0}.$$

For the third level, $\hat{A}_{0,0}$ is the matrix form of $\hat{a}_{0,0}(\cdot,\cdot)$ defined above.

To show the condition number of three-level NOSAS preconditioners, we first focus on the preconditioner in the coarse level and define B_0^{-1} as

$$B_0^{-1} = R_{0,0}^T \hat{A}_{0,0}^{-1} R_{0,0} + \sum_{j=1}^{N_0} R_{j,0}^T \hat{A}_{j,0}^{-1} R_{j,0}.$$

We note that B_0 can be seen as an approximation of A_0, and B_0^{-1} is a two-level NOSAS preconditioner of A_0. Therefore, similar to the two-level methods, we should also consider the relation of $B_{\Gamma_0\Gamma_0}^{(j)}$ with $A_{\Gamma_0\Gamma_0}^{(j)}$. For different choices of $B_{\Gamma_0\Gamma_0}^{(j)}$, let C_0 be the constant such that $A_{\Gamma_0\Gamma_0}^{(j)} \le C_0 B_{\Gamma_0\Gamma_0}^{(j)}$ for $1 \le j \le N_0$. For $B_{\Gamma_0\Gamma_0}^{(j)} = A_{\Gamma_0\Gamma_0}^{(j)}$, we have $C_0 = 1$. For $B_{\Gamma_0\Gamma_0}^{(j)} = \hat{A}_{\Gamma_0\Gamma_0}^{(j)}$, we have $C_0 = 3$ in two dimensions and $C_0 = 4$ in three dimensions. Then, using the NOSAS methods property we have shown in Theorem 1, we have $\forall u_\Gamma \in V_0$,

$$\frac{\eta_0}{C_0+1} u_\Gamma^T A_0^{-1} u_\Gamma \le u_\Gamma^T B_0^{-1} u_\Gamma \le (C_0+1) u_\Gamma^T A_0^{-1} u_\Gamma.$$

Since A_0 and B_0 are symmetric and positive definite matrices, it is equivalent to

$$\frac{1}{C_0+1} u_\Gamma^T A_0 u_\Gamma \le u_\Gamma^T B_0 u_\Gamma \le \frac{C_0+1}{\eta_0} u_\Gamma^T A_0 u_\Gamma.$$

Using the above property of B_0 and combining it with the abstract theory of the additive Schwarz method, we can obtain the following condition number for the three-level NOSAS methods.

Theorem 2 *For any $u_h \in V_h(\Omega)$ the following holds:*

$$\Big(\frac{C_1}{\eta_1}+\frac{C_0+1}{\eta_1\eta_0}\Big)^{-1} a(u_h,u_h) \le a(M_{3NOSAS}^{-1}Au_h,u_h) \le (1+C_1+C_1C_0)a(u_h,u_h).$$

3 Numerical experiments

We present numerical results for the variational formulation of $\int_\Omega \rho(x)\nabla u\cdot\nabla v dx = \int_\Omega f v dx$ with $f=1$ and a heterogeneous coefficient function $\rho(x)$. We choose four specific $\rho(x)$ from the SPE10 model problems, Kxx_06 and Kxx_85, with the computational domain $\Omega=(0,22)\times(0,6)$. We decompose Ω into 33 congruent square subregions of size $H_0=2$, and 528 congruent square subdomains of size $H=1/2$. We further decompose each square subdomain into $(H/h)^2$ congruent small squares of size $h=0.1$. The shape-regular partition $\mathcal{T}_h$ is obtained by dividing each of these small squares into two right triangle elements. $V_h(\Omega)$ are the piecewise linear basis functions on the triangulation $\mathcal{T}_h$. We impose a zero Dirichlet boundary condition on $\partial\Omega$ and use the PCG method for the preconditioned system with the relative residual error 10^{-6} in the l^2 norm.

Table 1 The two-level NOSAS preconditioners applied to four SPE10 model problems with different η_1.

η_1	Iter.	Cond.	Size of global problem
0.025	46	34.45	431
0.05	39	24.47	443
0.1	32	15.52	486
0.2	23	6.82	735
0.4	16	4.05	1674

Kxx_06

η_1	Iter.	Cond.	Size of global problem
0.025	46	29.43	480
0.05	42	27.71	513
0.1	33	13.92	600
0.2	25	8.61	833
0.4	18	4.57	1535

Kxx_85

Table 2 The three-level NOSAS preconditioners applied to four SPE10 meshes with $\eta_0=0.25$ and different η_1.

η_1	Iter.	Cond.	Size of global problem
0.025	57	48.62	123
0.05	49	31.16	124
0.1	45	31.19	130
0.2	34	14.94	163
0.4	26	9.36	236

Kxx_06

η_1	Iter.	Cond.	Size of global problem
0.025	62	61.18	169
0.05	54	38.68	174
0.1	45	27.55	191
0.2	39	23.88	226
0.4	28	11.93	276

Kxx_85

The scalability of NOSAS methods with the "inexact" solver is shown in [5, 6]. The main focus of our numerical experiments is to compare two-level NOSAS preconditioners with three-level NOSAS preconditioners, and show that three-level NOSAS have a smaller size of the global problem while maintaining a good iteration and condition number. For the two-level NOSAS preconditioners, we choose $B_{\Gamma\Gamma}^{(i)} = S_F^{(i)}$ with $\delta = 2h$, and choose R_0^T the discrete a-harmonic extension for the low-frequency eigenfunctions and $\mathcal{H}_{\delta,D}^{(i)}$ for the high-frequency eigenfunctions in (2). For the three-level NOSAS preconditioners, $B_{\Gamma\Gamma}^{(i)}$, R_0^T are the same as the two-level preconditioners. Then we choose $B_{\Gamma_0\Gamma_0}^{(j)} = \hat{A}_{\Gamma_0\Gamma_0}^{(j)}$ as the diagonal of $A_{\Gamma_0\Gamma_0}^{(j)}$, and $R_{0,0}^T$ as the discrete a-harmonic extension for the low-frequency eigenfunctions and zero extension for the high-frequency eigenfunctions in (3). Note that $\frac{h}{H} = 0.2$ and Table 1 shows the performance of the two-level NOSAS with different thresholds η_1 for the SPE10 model problems. For the three-level NOSAS, we choose a fixed $\eta_0 = \frac{H}{H_0}$ and show the corresponding results for different thresholds η_1 in Table 2. All our test results support the theoretical condition number bound in Theorem 1 and Theorem 2. In addition, we observe a much smaller condition number numerically. The reason is that numerically, the constant C_0 is close to 1.8 and C_1 is close to 1.5 for the "inexact" solver.

References

1. Dolean, V., Nataf, F., Scheichl, R., and Spillane, N. Analysis of a two-level Schwarz method with coarse spaces based on local Dirichlet-to-Neumann maps. *Computational Methods in Applied Mathematics* **12**(4), 391–414 (2012).
2. Heinlein, A., Klawonn, A., Knepper, J., Rheinbach, O., and Widlund, O. B. Adaptive GDSW coarse spaces of reduced dimension for overlapping Schwarz methods. *SIAM Journal on Scientific Computing* **44**(3), A1176–A1204 (2022).
3. Heinlein, A., Klawonn, A., Rheinbach, O., and Röver, F. A three-level extension of the GDSW overlapping Schwarz preconditioner in two dimensions. In: *Advanced Finite Element Methods with Applications: Selected Papers from the 30th Chemnitz Finite Element Symposium 2017 30*, 187–204. Springer ((2019)).
4. Tu, X. Three-level BDDC in two dimensions. *International journal for numerical methods in engineering* **69**(1), 33–59 (2007).
5. Yu, Y., Dryja, M., and Sarkis, M. Nonoverlapping spectral additive Schwarz methods. In: *Domain Decomposition Methods in Science and Engineering XXV 25*, 375–382. Springer ((2020)).
6. Yu, Y., Dryja, M., and Sarkis, M. From additive average Schwarz methods to nonoverlapping spectral additive Schwarz methods. *SIAM Journal on Numerical Analysis* (2021).
7. Yu, Y., Dryja, M., and Sarkis, M. Nonoverlapping spectral additive Schwarz methods for hybrid discontinuous Galerkin discretizations (Accepted by IMA Journal of Numerical Analysis, 2022).
8. Yu, Y. and Sarkis, M. A family of nonoverlapping spectral additive Schwarz methods and their economic versions (submitted to Journal of Computational and Applied Mathematics, 2022).

Optimized Schwarz Method for Coupled Direct-Adjoint Problems Applied to Parameter Identification in Advection-Diffusion Equation

Alexandre Vieira and Pierre-Henri Cocquet

1 Introduction

Let Ω be a bounded open set with Lipschitz boundary. We want to solve the problem

$$\begin{aligned} &\min_k \frac{1}{2}\|T - T_{\text{target}}\|^2_{L^2(\Omega)}, \\ &\text{s.t. } \operatorname{div}(\mathbf{u}T) - \operatorname{div}(k\nabla T) = f \text{ in } \Omega,\ T|_{\partial\Omega} = T_0 \text{ and } k \in U_{ad}, \end{aligned} \tag{1}$$

with $f \in L^2(\Omega), \mathbf{u} \in H^1(\Omega)$ given s.t. div$\mathbf{u} = 0$. The set of admissible control U_{ad} contains all $k(x) \in [a, b]$ for a.e. x and $a > 0$ and is chosen such that any sequences $(k_n)_n \subset U_{ad}$ have a subsequence converging a.e. in Ω. Such $k \in U_{ad}$ must be more regular (e.g with bounded variation) and we refer to [9, p. 9, Assumption 1] for an example of such U_{ad}. With all these assumptions, (1) has at least one optimal solution (see e.g. [9, Theorem 3]).

We will focus on finding ways to compute a solution to (1) on several subdomains. In recent years, a lot of papers started to look at ways to decompose the resolution of optimal control problems. In [5], the authors split the optimization problem as two independent optimization problems, splitted by subdomains, with an augmented cost. The necessary (and sufficient) conditions of optimality let us see that it actually reduces to a classical Schwarz method applied to the direct and adjoint systems, where the control could be eliminated (see also [1, 2, 7]).

These papers rely on the huge literature analyzing the different flavors of the Schwarz iterative method: we only refer to [4] for an introduction and to [8] for a more in depth presentation of these methods (and other decomposition methods).

A. Vieira
Université de la Réunion, PIMENT, Sainte-Clotilde, France,
e-mail: alexandre.vieira@univ-reunion.fr

P.-H. Cocquet
Université de Pau et des Pays de l'Adour, SIAME, rue de l'Université, Pau, France,
e-mail: pierre-henri.cocquet@univ-pau.fr

In order to decompose the resolution of (1) across several subdomains, we will also adopt an *indirect approach*: we will decompose the computations of the gradient of the cost, which will be used afterward in a descent method. Since the control needs to be defined on the whole domain Ω, it seems hard to define a decomposition of (1) using an overlap. Therefore, we will focus on finding an optimized Schwarz iteration, without overlap, to compute the gradient of the cost.

2 Direct and adjoint equations

First of all, we express the gradient of the cost which can be given thanks to an adjoint equation. The next result can be proved using [6, Corollary 1.3].

Theorem 1 *Let* $J(k) = \frac{1}{2}\|T(k) - T_{target}\|^2_{L^2(\Omega)}$ *with* $T(k) \in H^1(\Omega)$ *be the solution to* $\mathrm{div}(\mathbf{u}T) - \mathrm{div}(k\nabla T) = f$ *in* Ω *and* $T|_{\partial\Omega} = T_0$. *Then:* $\partial_k J(k) = \nabla T \cdot \nabla\lambda$, *where* λ *solves*

$$\mathrm{div}(k\nabla\lambda - \mathbf{u}\lambda) = T - T_{target},\ \lambda|_{\partial\Omega} = 0.$$

Therefore, the gradient of J can be computed by solving for fixed k:

$$\begin{cases} -\mathrm{div}(k\nabla T - \mathbf{u}T) = f, & T|_{\partial\Omega} = T_0, \\ \mathrm{div}(k\nabla\lambda + \mathbf{u}\lambda) = T - T_{\mathrm{target}}, & \lambda|_{\partial\Omega} = 0. \end{cases} \tag{2}$$

We now expose our strategy in order to accelerate the resolution of (1): we would like to decompose the resolution of (2) across several subdomains in order to accelerate the computation of the gradient. However, since our optimization parameter k is defined on Ω, it seems hard to imagine a decomposition method using an overlap. If we were computing a solution of (2) on two subdomains Ω_1 and Ω_2 with an overlap, then we would end with two different gradients $\partial_k J(k)$ on $\Omega_1 \cap \Omega_2$, depending on which side the gradient is computed. Therefore, using a descent technique would produce two different controls k_1 on Ω_1 and k_2 on Ω_2, with possibly different values on $\Omega_1 \cap \Omega_2$. This could prevent the convergence to an optimal solution of (1).

To summarize, we are interested in non-overlapping Schwarz techniques. It should be noted that optimized Schwarz method for an advection-diffusion equation has been done in [3] but, to the best of our knowledge, never for (2).

3 Optimized Schwarz method for coupled direct-adjoint system

We assume there is open sets Ω_i such that $\overline{\Omega} = \overline{\Omega_1 \cup \Omega_2}$ with interface $\Gamma_\cap = \overline{\Omega_1} \cap \overline{\Omega_2}$. A non-overlapping Schwarz method for (2) can then be roughly defined as

1. Take an initial guess (T_i^0, λ_i^0) defined on Ω_i,
2. Until some stopping criteria are met: Compute $(T_i^{n+1}, \lambda_i^{n+1})$ satisfying

$$\begin{cases} -\mathrm{div}(k\nabla T_i^{n+1}) + \mathrm{div}(uT_i^{n+1}) = f, & \text{on } \Omega_i,\ T_i^{n+1}|_{\partial\Omega_i\setminus\Gamma_\cap} = T_0, \\ \mathrm{div}(k\nabla \lambda_i^{n+1}) + \mathrm{div}(u\lambda_i^{n+1}) = T^{n+1} - T_{\text{target}} & \text{on } \Omega_i,\ \lambda_i^{n+1}|_{\partial\Omega_i\setminus\Gamma_\cap} = 0, \end{cases} \tag{3}$$

and the following transmission conditions on the interface

$$\begin{aligned} k\partial_{\mathbf{n}}\begin{pmatrix} T_i^{n+1} \\ \lambda_i^{n+1}\end{pmatrix} - \frac{\mathbf{u}\cdot\mathbf{n}}{2}\begin{pmatrix} T_i^{n+1} \\ -\lambda_i^{n+1}\end{pmatrix} + (-1)^{i+1}\mathcal{S}_i\begin{pmatrix} T_i^{n+1}|_{\Gamma_\cap} \\ \lambda_i^{n+1}|_{\Gamma_\cap}\end{pmatrix} \\ = k\partial_{\mathbf{n}}\begin{pmatrix} T_{3-i}^{n} \\ \lambda_{3-i}^{n}\end{pmatrix} - \frac{\mathbf{u}\cdot\mathbf{n}}{2}\begin{pmatrix} T_{3-i}^{n} \\ -\lambda_{3-i}^{n}\end{pmatrix} + (-1)^{i+1}\mathcal{S}_i\begin{pmatrix} T_{3-i}^{n}|_{\Gamma_\cap} \\ \lambda_{3-i}^{n}|_{\Gamma_\cap}\end{pmatrix}, \end{aligned} \tag{4}$$

where $\mathbf{n}$ is the outer normal to $\partial\Omega_1$. In (3)–(4), $\mathcal{S}_i$ are linear operators acting on traces of (T_i, λ_i) (e.g. $\mathcal{S}_i = p_i\,\mathrm{id}$ where p_i are some constants and id is the identity operator, or some linear differential operator involving tangential derivatives).

To study the convergence of the non-overlapping Schwarz method as well as its convergence properties, we are going to restrict ourselves to the case k = constant, $\Omega = \mathbb{R}^2$, $\Omega_1 = (-\infty, 0)\times\mathbb{R}$ and $\Omega_2 = (0, +\infty)\times\mathbb{R}$. In such setting, we can rely on Fourier analysis [4] to study the convergence and also to design optimized transmission operators $\mathcal{S}_i$ that accelerate the convergence. Without loss of generality, we also suppose that $f = T_{\text{target}} = 0$ since we are interested in the error.

3.1 Computation of the optimal transmission operator

We start by applying Fourier transform to (2) along the y axis:

$$-k\partial_{xx}\begin{pmatrix} \hat{T}_i^n \\ \hat{\lambda}_i^n\end{pmatrix} + u_1\partial_x\begin{pmatrix} \hat{T}_i^n \\ -\hat{\lambda}_i^n\end{pmatrix} + \begin{pmatrix} k\omega^2 - iu_2\omega & 0 \\ 1 & k\omega^2 + iu_2\omega\end{pmatrix}\begin{pmatrix} \hat{T}_i^n \\ \hat{\lambda}_i^n\end{pmatrix} = 0, i \in \{1,2\}.$$

Along x, this is a second order ordinary differential equation which can be solved explicitly. Define :

$$r_\pm^T(\omega) = \frac{u_1 \pm \sqrt{u_1^2 + 4k^2\omega^2 - 4iku_2\omega}}{2k},\ r_\pm^\lambda(\omega) = \frac{-u_1 \pm \sqrt{u_1^2 + 4k^2\omega^2 + 4iku_2\omega}}{2k}.$$

Concerning $\hat{T}$, using the Dirichlet condition at infinity, there exist functions $A_T^n(\omega)$ and $B_T^n(\omega)$ such that:

$$\hat{T}_1^n(x,\omega) = A_T^n(\omega)e^{r_+^T(\omega)x},\ \hat{T}_2^n(x,\omega) = B_T^n(\omega)e^{r_-^T(\omega)x}.$$

These solutions are reintroduced into the equation in order to solve it for $\hat{\lambda}$. There, the equation is non-homogeneous, but the right hand-side is of the form $C(\omega)e^{D(\omega)x}$, for some functions C and D independent of x. An arbitrary solution is therefore easily found, and one proves that they take the form:

$$\hat{\lambda}_1^n(x,\omega) = A_\lambda^n(\omega)e^{r_+^\lambda(\omega)x} - A_{\lambda T}(\omega)\hat{T}_1^n(x,\omega),$$
$$\hat{\lambda}_2^n(x,\omega) = B_\lambda^n(\omega)e^{r_-^\lambda(\omega)x} - B_{\lambda T}(\omega)\hat{T}_2^n(x,\omega),$$

where $A_{\lambda T}(\omega) = (-kr_+^T(\omega)^2 - u_1 r_+^T(\omega) + k\omega^2 + iu_2\omega)^{-1}$ and $B_{\lambda T}(\omega) = (-kr_-^T(\omega)^2 - u_1 r_-^T(\omega) + k\omega^2 + iu_2\omega)^{-1}$.

We may now derive each solution with x:

$$\partial_x\hat{T}_1^n(x,\omega) = r_+^T(\omega)\hat{T}_1^n(x,\omega),\ \partial_x\hat{T}_2^n(x,\omega) = r_-^T(\omega)\hat{T}_2^n(x,\omega),$$
$$\partial_x\hat{\lambda}_1^n(x,\omega) = r_+^\lambda(\omega)A_\lambda^n(\omega)e^{r_+^\lambda(\omega)x} - A_{\lambda T}(\omega)r_+^T(\omega)\hat{T}_1^n(x,\omega),$$
$$\partial_x\hat{\lambda}_2^n(x,\omega) = r_-^\lambda(\omega)B_\lambda^n(\omega)e^{r_-^\lambda(\omega)x} - B_{\lambda T}(\omega)r_-^T(\omega)\hat{T}_2^n(x,\omega).$$

We now assume that $\mathcal{F}_y(\mathcal{S}_i(T,\lambda))(x,\omega) = \sigma_i(\omega)(\hat{T},\hat{\lambda})$, where σ_i is a 2×2 complex matrix. The transmission conditions then read:

$$k\partial_x\begin{pmatrix}\hat{T}_1^n\\ \hat{\lambda}_1^n\end{pmatrix}(x,\omega) - \frac{u_1}{2}\begin{pmatrix}\hat{T}_1^n\\ -\hat{\lambda}_1^n\end{pmatrix}(x,\omega) + \sigma_1(\omega)\begin{pmatrix}\hat{T}_1^n\\ \hat{\lambda}_1^n\end{pmatrix}(x,\omega)$$
$$= \left(M_r^+(x,\omega) + \sigma_1(\omega)M_0^+(x,\omega)\right)\begin{pmatrix}A_T^n(\omega)\\ A_\lambda^n(\omega)\end{pmatrix},$$
$$k\partial_x\begin{pmatrix}\hat{T}_2^n\\ \hat{\lambda}_2^n\end{pmatrix}(x,\omega) - \frac{u_1}{2}\begin{pmatrix}\hat{T}_2^n\\ -\hat{\lambda}_2^n\end{pmatrix}(x,\omega) + \sigma_2(\omega)\begin{pmatrix}\hat{T}_2^n\\ \hat{\lambda}_2^n\end{pmatrix}(x,\omega)$$
$$= \left(M_r^-(x,\omega) + \sigma_2(\omega)M_0^-(x,\omega)\right)\begin{pmatrix}B_T^n(\omega)\\ B_\lambda^n(\omega)\end{pmatrix},$$

$$M_r^+(x,\omega) = \begin{pmatrix}(kr_+^T(\omega) - \frac{u_1}{2})e^{r_+^T(\omega)x} & 0\\ -(kr_+^T(\omega) - \frac{u_1}{2})A_{\lambda T}(\omega)e^{r_+^T(\omega)x} & (kr_+^\lambda(\omega) - \frac{u_1}{2})e^{r_+^\lambda(\omega)x}\end{pmatrix}$$
$$M_0^+(x,\omega) = \begin{pmatrix}e^{r_+^T(\omega)x} & 0\\ A_{\lambda T}(\omega)e^{r_+^T(\omega)x} & e^{r_+^\lambda(\omega)x}\end{pmatrix},\ M_0^-(x,\omega) = \begin{pmatrix}e^{r_-^T(\omega)x} & 0\\ -B_{\lambda T}(\omega)e^{r_-^T(\omega)x} & e^{r_-^\lambda(\omega)x}\end{pmatrix}$$
$$M_r^-(x,\omega) = \begin{pmatrix}(kr_-^T(\omega) - \frac{u_1}{2})e^{r_-^T(\omega)x} & 0\\ -(kr_-^T(\omega) - \frac{u_1}{2})B_{\lambda T}(\omega)e^{r_-^T(\omega)x} & (kr_-^\lambda(\omega) - \frac{u_1}{2})e^{r_-^\lambda(\omega)x}\end{pmatrix}$$

Using the conditions at $x = 0$, we get the following recurrence:

$$\begin{pmatrix}A_T^n(\omega)\\ A_\lambda^n(\omega)\end{pmatrix} = \underbrace{\left[M_r^+(0,\omega) + \sigma_1(\omega)M_0^+(0,\omega)\right]^{-1}\left[M_r^-(0,\omega) + \sigma_1(\omega)M_0^-(0,\omega)\right]}_{M_1(\omega)}$$
$$\underbrace{\left[M_r^-(0,\omega) - \sigma_2(\omega)M_0^-(0,\omega)\right]^{-1}\left[M_r^+(0,\omega) - \sigma_2(\omega)M_0^+(0,\omega)\right]}_{M_2(\omega)}$$
$$\times\begin{pmatrix}A_T^{n-2}(\omega)\\ A_\lambda^{n-2}(\omega)\end{pmatrix}$$

$$\begin{pmatrix} B_T^n(\omega) \\ B_\lambda^n(\omega) \end{pmatrix} = \underbrace{\left[M_r^-(0,\omega) - \sigma_2(\omega)M_0^-(0,\omega)\right]^{-1}\left[M_r^+(0,\omega) - \sigma_2(\omega)M_0^+(0,\omega)\right]}_{M_2(\omega)}$$
$$\underbrace{\left[M_r^+(0,\omega) + \sigma_1(\omega)M_0^+(0,\omega)\right]^{-1}\left[M_r^-(0,\omega) + \sigma_1(\omega)M_0^-(0,\omega)\right]}_{M_1(\omega)}$$
$$\times \begin{pmatrix} B_T^{n-2}(\omega) \\ B_\lambda^{n-2}(\omega) \end{pmatrix}$$

Therefore, the optimal choice of σ_i cancels $M_1(\omega)M_2(\omega)$ and $M_2(\omega)M_1(\omega)$; this reads:

$$\begin{aligned}
\sigma_1^{\text{opt}}(\omega) &= -M_r^-(0,\omega)\left(M_0^-(0,\omega)\right)^{-1} \\
&= \begin{pmatrix} -kr_-^T(\omega) + \frac{u_1}{2} & 0 \\ -kB_{\lambda T}(\omega)[r_-^\lambda(\omega) - r_-^T(\omega)] & -kr_-^\lambda(\omega) - \frac{u_1}{2} \end{pmatrix}, \\
\sigma_2^{\text{opt}}(\omega) &= M_r^+(0,\omega)\left(M_0^+(0,\omega)\right)^{-1} \\
&= \begin{pmatrix} kr_+^T(\omega) - \frac{u_1}{2} & 0 \\ -kA_{\lambda T}(\omega)[r_+^T(\omega) - r_+^\lambda(\omega)] & kr_+^\lambda(\omega) + \frac{u_1}{2} \end{pmatrix}.
\end{aligned}$$

However, as it is usual concerning the optimal Schwarz operator, an inverse Fourier transform proves that $\mathcal{S}_i^{opt}$, the inverse Fourier transform of σ_i^{opt}, is a non-local operator [4]. This property can be difficult to handle in a numerical method. This is why we will restrict the set of admissible transmission operator $\mathcal{S}_i$ to local constant operators.

3.2 Computation of optimized transmission operator

Instead of using the optimal (non-local) operator, we will search for an optimal lower-triangular matrix P_i, which we will suppose to be constant in ω. All the calculations above can be done similarly with this new assumption, and we may write similarly the new matrices $M_1(\omega)$ and $M_2(\omega)$. Suppose $\sigma_1 = \begin{pmatrix} \sigma_{11} & 0 \\ \sigma_{13} & \sigma_{14} \end{pmatrix}$ and $\sigma_2 = \begin{pmatrix} \sigma_{21} & 0 \\ \sigma_{23} & \sigma_{24} \end{pmatrix}$. Then $M_l(\omega)M_m(\omega) = \begin{pmatrix} M_1^1(\omega) & 0 \\ M_3^{lm}(\omega) & M_4^1(\omega) \end{pmatrix}$, where

$$M_1^1(\omega) = \frac{(2kr_-^T(\omega) + 2\sigma_{11} - u_1)(-2kr_+^T(\omega) + 2\sigma_{21} + u_1)}{(2kr_+^T(\omega) + 2\sigma_{11} - u_1)(-2kr_-^T(\omega) + 2\sigma_{21} + u_1)},$$

$$M_4^1(\omega) = \frac{(2kr_-^\lambda(\omega) + 2\sigma_{14} + u_1)(2kr_+^\lambda(\omega) - 2\sigma_{24} + u_1)}{(2kr_+^\lambda(\omega) + 2\sigma_{14} + u_1)(2kr_-^\lambda(\omega) - 2\sigma_{24} + u_1)},$$

and $M_3^{lm}(\omega)$ for $l, m = 1, 2$ can be computed as above but are not given since their expressions are not needed in the subsequent analysis.

A way that seems natural is to optimize the transmission conditions consists in solving the min-max problem: $\min_{\sigma_{11},\sigma_{13},\sigma_{14}} \max_{\omega\in[\omega_1,\omega_2]} \|M_1(\omega)M_2(\omega)\|$ for some matrix norm $\|\cdot\|$ and some constants $\omega_1 < \omega_2$. Solving this min-max problem can be tricky: the result may depend on the chosen norm, and the complicated expression of the components of M_1M_2 makes the whole analysis inextricable. Furthermore, it is not entirely clear how one could use either the product M_1M_2 or M_2M_1 in this min max problem. This could change the nature of the result.

However, we remark that the spectral radius appears to be useful in this case, since $\rho(M_1(\omega)M_2(\omega)) = \rho(M_2(\omega)M_1(\omega))$ only depend on σ_{11}, σ_{21}, σ_{14} and σ_{24}. Furthermore, optimizing the spectral radius of the matrices may be done in two independent optimization problems:

$$\min_{\sigma_{11},\sigma_{21}} \max_{\omega\in[\omega_1,\omega_2]} \left| \frac{\left(-\sqrt{u_1^2+4k^2\omega^2-4iku_2\omega}+2\sigma_{11}\right)\left(-\sqrt{u_1^2+4k^2\omega^2-4iku_2\omega}+2\sigma_{21}\right)}{\left(\sqrt{u_1^2+4k^2\omega^2-4iku_2\omega}+2\sigma_{11}\right)\left(\sqrt{u_1^2+4k^2\omega^2-4iku_2\omega}+2\sigma_{21}\right)} \right|,$$

$$\min_{\sigma_{14},\sigma_{24}} \max_{\omega\in[\omega_1,\omega_2]} \left| \frac{\left(-\sqrt{u_1^2+4k^2\omega^2+4iku_2\omega}+2\sigma_{14}\right)\left(-\sqrt{u_1^2+4k^2\omega^2+4iku_2\omega}+2\sigma_{24}\right)}{\left(\sqrt{u_1^2+4k^2\omega^2+4iku_2\omega}+2\sigma_{14}\right)\left(\sqrt{u_1^2+4k^2\omega^2+4iku_2\omega}+2\sigma_{24}\right)} \right|.$$

This kind of min max problem can be solved. Suppose $\omega_1 = 0$, $\omega_2 = \pi/h$ and applying results from [3, p. 35, Eq. (2.11)] prove, assuming h is small enough, that the solution in this case is:

$$\sigma_{11} = \sigma_{14} = \left(\frac{k\pi|u_1|^3}{2h}\right)^{\frac{1}{4}},\ \sigma_{21} = \sigma_{24} = \left(\frac{2^5k^3\pi^3|u_1|}{h^3}\right)^{\frac{1}{4}}. \tag{5}$$

Other similar results can be found in [3]. However, this approach of optimizing the spectral radius let the parameters σ_{13} and σ_{23} free, and $M_3^{12}(\omega)$ and $M_3^{21}(\omega)$ both depend on σ_{13} and σ_{23}. We show below how these parameters can be chosen.

3.3 Random tests for the last parameters

In order to see the influence of the parameters σ_{i3}, we ran a batch of numerical tests with random values for σ_{i3}. We then solve (2) on $\Omega = (-1,1)\times(0,1)$ with $k = 1$, $f = T_{\text{target}} = 0$, $\mathbf{u} = (-2,0)$, $T_0|_{x=-1} = 0$, $T_0|_{x=1} = 2$ and $T_0|_{y\in\{0,1\}} = 1$. We first solve the equation on Ω, and compare it with the solution of (3)–(4), where $\Omega_1 = (-1,0)\times(0,1)$, $\Omega_2 = (0,1)\times(0,1)$, $\Gamma_\cap = \{0\}\times[0,1]$ and σ_{i1}, σ_{i4} are assigned using (5). σ_{13} and σ_{23} are assigned randomly between -150 and $+150$. We used second order centered finite differences, a ghost point for the Robin boundary conditions and a 20×20 uniform grid for each subdomain. We then run 5 and 10 iterations of (3)–(4), and compare the result with the solution on the whole

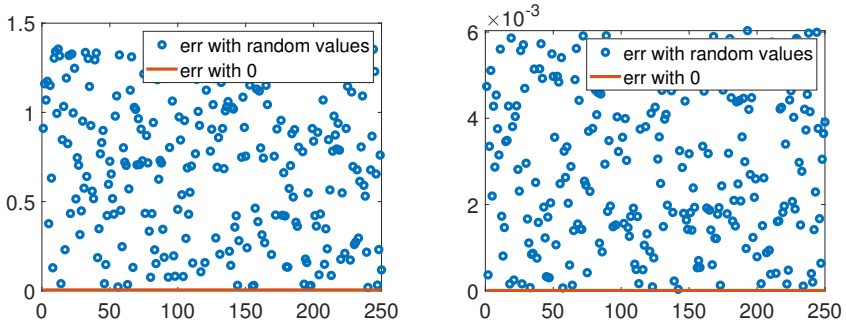


Fig. 1 Number random values between −150 and +150 generated for σ_{i3} VS Error (infinity norm) on λ at the 5th (Left) and 10th (Right) iteration using random antidiagonal elements (blue dots) or just 0 (red line).

domain, which let us compute an error at the end of the iterations. We did this experiment with 250 random couples for σ_{13} and σ_{23}, and plot the error. The results are given in Figure 1. From these results, we see that the choice $\sigma_{13} = \sigma_{23} = 0$ seems to give the lowest error (the red line in (1)). Indeed, after 5 (resp. 10) iterations, the lowest error is at 0.0189 (resp. 3.5919×10^{-5}) with the random values, while the error with $\sigma_{13} = \sigma_{23} = 0$ is at 0.0051 (resp. 8.4667×10^{-6}). This choice is special since it decouples T and λ at the interface $\Gamma_\cap$. It then suggests that the resolution of T first, and then λ, is more efficient, with respect to the number of Schwarz iterations.

3.4 Schwarz iteration as critical points of an optimization problem

We conclude this proceeding by showing that each iteration of the Schwarz method can be obtained by computing the critical points of some specific Lagrange functional. We restrict ourselves to transmission conditions (4), where $\mathcal{S}_i$ are lower-triangular matrices with constant coefficients and recall that $\mathbf{n}$ is the outer normal to $\partial\Omega_1$. We consider the next sub-domain problem

$$\begin{cases} -\mathrm{div}(k\nabla T_i - \mathbf{u}T_i) = f & \text{in } \Omega_i, \\ k\partial_{\mathbf{n}} T_i + a_i \mathbf{u}\cdot\mathbf{n} T_i + (-1)^{i+1} p_i T_i = (-1)^{i+1} g_i \text{ on } \Gamma_\cap,\ T_i = T_0 \text{ on } \partial\Omega_i \setminus \Gamma_\cap. \end{cases} \tag{6}$$

Its variational formulation is: *Find* $T \in H^1(\Omega_i)$ *such that* $T_i|_{\partial\Omega_i \setminus \Gamma_\cap} = T_0$ *and*

$$
\begin{aligned}
a_i(T_i,\lambda_i) :=& \int_{\Omega_i} k\nabla T_i\cdot\nabla\lambda_i - T_i\,\mathbf{u}\cdot\nabla\lambda_i\,dx \\
&- \int_{\Gamma_\cap}\left(p_i + (-1)^{i+1}(1+a_i)\mathbf{u}\cdot\mathbf{n}\right)T_i\lambda_i\,ds \\
=& \int_{\Omega_i} f\,\lambda_i\,dx + \int_{\Gamma_\cap} g_i\lambda_i\,ds,\ \forall\lambda_i\in V_i := \left\{\varphi\in H^1(\Omega_i)\mid \varphi|_{\partial\Omega_i\setminus\Gamma_\cap}=0\right\}.
\end{aligned}
$$

We then have the next result whose proof can formally be done by direct computation.

Theorem 2 *Given $\alpha_i\in L^\infty(\Gamma_\cap)$, $\beta_i\in L^2(\Gamma_\cap)$, we consider the Lagrangian*

$$
\begin{aligned}
\mathcal{L}_i(T_i,\lambda_i) = \frac{1}{2}\|T_i - T_{target}\|^2_{L^2(\Omega_i)} + a_i(T_i,\lambda_i) - \int_{\Omega_i} f\,\lambda_i\,dx - \int_{\Gamma_\cap} g_i\lambda_i\,ds \\
+ \int_{\Gamma_\cap}\left(\frac{\alpha_i}{2}T_i^2 + \beta_i T_i\right)ds,\ \forall\left(T_i - \widetilde{T_{0,i}}\right),\lambda_i\in V_i,
\end{aligned}
$$

where $\widetilde{T_{0,i}}\in V_i$ is an extension of T_0. Let (T_i,λ_i) satisfying $\partial_{T_i,\lambda_i}\mathcal{L}_i(T_i,\lambda_i)=0$. Then T_i is a weak solution to (6) *and $\lambda_i\in V_i$ is a weak solution to the (adjoint) problem*

$$
\begin{cases}
\mathrm{div}(k\nabla\lambda_i + \mathbf{u}\lambda_i) = T_i - T_{target}\ \textit{in}\ \Omega_i,\ \lambda_i = 0\ \textit{on}\ \partial\Omega_i\setminus\Gamma_\cap, \\
k\partial_{\mathbf{n}}\lambda_i + (1+a_i)\mathbf{u}\cdot\mathbf{n}\lambda_i + (-1)^{i+1}p_i\lambda_i = (-1)^{i+1}\left(\alpha_i T_i + \beta_i\right)\ \textit{on}\ \Gamma_\cap.
\end{cases}
\tag{7}
$$

From Theorem 2, we see that chosing $a_i=-\frac{1}{2}$, $p_i=\sigma_{i1}$, $\alpha_i=-\sigma_{i3}$,

$$
\begin{aligned}
g_i =& k\partial_{\mathbf{n}}T^n_{3-i} + a_i\mathbf{u}\cdot\mathbf{n}T^n_{3-i} + (-1)^{i+1}p_iT^n_{3-i}, \\
\beta_i =& \left(k\partial_{\mathbf{n}}\lambda^n_{3-i} + (1+a_i)\mathbf{u}\cdot\mathbf{n}\lambda^n_{3-i} + (-1)^{i+1}p_i\lambda^n_{3-i} + \alpha_iT^n_{3-i}\right),
\end{aligned}
$$

yields that $(T_i^{k+1} - \widetilde{T_{0,i}}, \lambda_i^{k+1})\in V_i\times V_i$ is a critical point of $\mathcal{L}_i$. Each iterate of the DDM can then be obtained by solving an optimization problem on each subdomain (see also e.g. [1, 5]).

4 Conclusion

Using a Schwarz method on (2) appears to be harder than expected. The transmission conditions found in [3] can be adapted to this case, but only gives a partial clue to define some optimized transmission operator. Furthermore, solving (2) only let us compute the gradient of the cost which only accelerates the computation of the gradient, but not necessarily the resolution of (1). Concerning (2), we still wonder if one can take advantage of the triangular structure of (1): is it better to solve first for T alone, and then for λ, or could we find efficient iterations to compute the couple (T,λ)? Additional works in this direction are on-going projects.

Acknowledgements All the authors are supported by the "Agence Nationale de la Recherche" (ANR), Project O-TO-TT-FU number ANR-19-CE40-0011.

References

1. Ciaramella, G., Halpern, L., and Mechelli, L. Convergence analysis and optimization of a Robin Schwarz waveform relaxation method for periodic parabolic optimal control problems. MOX-Report No. 62/2022 (2022).
2. Ciaramella, G., Kwok, F., and Müller, G. Nonlinear optimized Schwarz preconditioner for elliptic optimal control problems. In: *Domain Decomposition Methods in Science and Engineering XXVI*. Springer (2020).
3. Dubois, O. *Optimized Schwarz Methods for the Advection-Diffusion Equation and for Problems with Discontinuous Coefficients*. Ph.D. thesis, McGill University (2007).
4. Gander, M. J. Optimized Schwarz methods. *SIAM Journal on Numerical Analysis* **44**(2), 699–731 (2006).
5. Gong, W., Kwok, F., and Tan, Z. Convergence analysis of the Schwarz alternating method for unconstrained elliptic optimal control problems. *arXiv preprint arXiv:2201.00974* (2022).
6. Hinze, M., Pinnau, R., Ulbrich, M., and Ulbrich, S. *Optimization with PDE constraints*, vol. 23. Springer Science & Business Media (2008).
7. Lagnese, J. E., Leugering, G., and Leugering, G. *Domain in Decomposition Methods in Optimal Control of Partial Differential Equations*. 148. Springer Science & Business Media (2004).
8. Quarteroni, A. and Valli, A. *Domain decomposition methods for partial differential equations*. Oxford University Press (1999).
9. Vieira, A., Bastide, A., and Cocquet, P.-H. Topology optimization for steady-state anisothermal flow targeting solids with piecewise constant thermal diffusivity. *Applied Mathematics & Optimization* **85**(3), 1–32 (2022).

Three-Level BDDC for Virtual Elements

Axel Klawonn, Martin Lanser, and Adam Wasiak

1 Introduction

The Virtual Element Method (VEM) is a Galerkin-type method for the solution of partial differential equations which allows for the discretization with general polygonal/polyhedral meshes. Furthermore, the VEM framework allows for the relatively simple construction of trial and test spaces with desirable properties on these general meshes. In recent years, numerous variants of the VEM have been proposed and analyzed, which include nonconforming, high-regularity, high-order, and hourglass-stabilized variants [4, 5, 9, 11]. The different approaches have been applied to many different model problems. As a framework to make the VEM suitable for large scale problems, the FETI-DP (Finite Element Tearing and Interconnecting - Dual Primal) and BDDC (Balancing Domain Decomposition by Constraints) domain decomposition methods have been introduced for virtual element discretizations [6, 7], which allows for an efficient and parallel iterative solution on large-scale computers. Recently, the analysis has been extended to the Stokes problem in [8], and adaptive coarse spaces for virtual element discretizations have been considered in [10] for mixed form problems in three dimensions and in [13] for stationary diffusion and linear elasticity in two dimensions. The use of adaptive coarse spaces allows for the solution of highly heterogeneous problems, for example, stationary diffusion problems with jumps in the diffusion coefficient, since, in the case of both finite and virtual elements, the method is provably robust. In [14] a condition number bound of the preconditioned system which only depends on geometrical constants and a user defined tolerance was shown for finite element discretizations and in [13]

Axel Klawonn, Martin Lanser
Department of Mathematics and Computer Science, Division of Mathematics, University of Cologne, Weyertal 86-90, 50931 Cologne, Germany and Center for Data and Simulation Science, University of Cologne, Germany, e-mail: axel.klawonn@uni-koeln.de, martin.lanser@uni-koeln.de

Adam Wasiak
Department of Mathematics and Computer Science, Division of Mathematics, University of Cologne, Weyertal 86-90, 50931 Cologne, Germany, e-mail: adam.wasiak@uni-koeln.de

the same was shown for the virtual element case. Unfortunately, adaptive coarse spaces can be large, especially for decompositions with many subdomains and/or difficult coefficient distributions. Also, classical coarse spaces grow proportionally with the number of subdomains and, in a parallel context, with the number of parallel resources. These large global coarse problems are a typical parallel scalability bottleneck in BDDC and FETI-DP methods, since the exact solution using, for example, sparse direct solvers does not scale. To alleviate this difficulty in BDDC, numerous multilevel approaches have been proposed, where the solution of the coarse problem is approximated by applying BDDC recursively using a very coarse domain decomposition. This allows for a parallel solution of the coarse problem and thus improves scalability. Here, we consider the three-level BDDC preconditioner introduced in [16] and apply it to the BDDC method with virtual element discretizations for the first time.

2 Model problems and the virtual element method

The domain $\Omega \subset \mathbb{R}^2$ is assumed to be a polygon. Let $f \in L^2(\Omega)$. We consider the stationary diffusion equation with homogeneous Dirichlet boundary values

$$-\nabla \cdot (\rho \nabla u) = f \quad \text{in } \Omega, \qquad u = 0 \quad \text{on } \partial\Omega.$$

Here, we assume ρ to satisfy $0 < \rho_* \leq \rho(x) \leq \rho^*$ for two constants $\rho_*, \rho^* \in \mathbb{R}$. The corresponding weak formulation is given by

$$\begin{cases} \text{Find } u \in H^1_0(\Omega) \text{ such that} \\ a(u,v) = (f,v)_{L^2(\Omega)} \text{ for all } v \in H^1_0(\Omega), \end{cases} \tag{1}$$

where $a(v,w) := (\rho \nabla v, \nabla w)_{L^2(\Omega)}$ for $v, w \in H^1_0(\Omega)$. We briefly introduce the VEM as it is presented in [2, 1]. Let $\{\mathcal{T}_h\}_h$ be a sequence of quasi-uniform tessellations of Ω into a finite number of simple polygons K, where $h := \max_{K \in \mathcal{T}_h} h_K$ and $h_K := \operatorname{diam}(K)$. Each polygon has a finite number of vertices. Let $\mathbb{P}_k(K)$ denote the space of polynomials of at most degree k on K. The meshes are assumed to satisfy the following condition. There exists a $\gamma > 0$ such that for all h and for all $K \in \mathcal{T}_h$:

1. K is star-shaped with respect to a ball of radius $\geq \gamma h_K$.
2. The distance between any two vertices of K is $\geq \gamma h_K$.

Denoting the set of edges of K by $\mathcal{E}^K$ and defining $\mathbb{P}_{-1} = \{0\}$, a suitable local virtual element space for the target order of accuracy $k \in \mathbb{N}$ is given by

$$V^h(K) = \{v \in H^1(K) \colon v|_e \in \mathbb{P}_k(e)\, \forall e \subset \mathcal{E}^K,\ v|_{\partial K} \in C(\partial K),\ \Delta v \in \mathbb{P}_{k-2}(K)\},$$

where $C(\partial K)$ denotes the continuous functions on the boundary of K. Then the global virtual element space can be defined as $V^h = \{v \in H^1_0(\Omega) \colon v|_K \in V^h(K)\}$. We can choose the following degrees of freedom on V^h:

- The values of v_h on each polygon vertex.
- For $k \geq 2$, the $k-1$ values of v_h on each point of the Gauss-Lobatto quadrature rule on every edge of the tessellation.
- For $k \geq 2$ and all $K \in \mathcal{T}_h$, the volume moments up to order $k-2$ of v_h in K.

The term $a(u_h, v_h)$ cannot be computed for $v_h, w_h \in V^h$ from the given degrees of freedom. Therefore, we replace $a(\cdot,\cdot)$ with a suitable approximate bilinear form $a_h(\cdot,\cdot)$ obtaining the discrete variational problem: Find $u_h \in V^h$ such that $a_h(u_h, v_h) = f_h(v_h)\ \forall v_h \in V^h$. For more details on the construction and implementation of $a_h(\cdot,\cdot)$ and related theoretical estimates we refer to [1, 2, 3].

3 BDDC and three-level BDDC

Let us give a brief description of the BDDC method as it applies to virtual element discretizations. Let $\{\Omega_i\}_{i=1}^N$ be a nonoverlapping domain decomposition of Ω such that $\overline{\Omega} = \cup_{i=1}^N \overline{\Omega}_i$, equipped with sequences of quasi-uniform tessellations $\mathcal{T}_i^h$, $i = 1, ..., N$ that satisfy the VEM grid assumptions. For each subdomain Ω_i, we obtain local stiffness matrices $K^{(i)}$ and load vectors $f^{(i)}$ using the VEM.

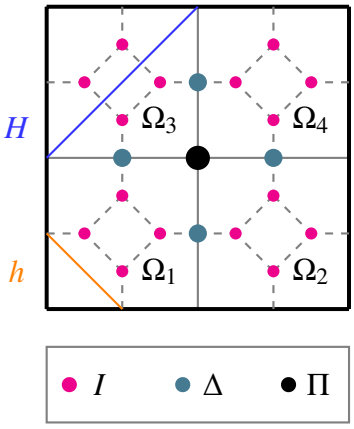


Fig. 1 Domain decomposition with polygonal meshes.

We denote by H_i the diameter of Ω_i and define $H := \max_i H_i$. Let $\Gamma := \cup_{i \neq j} \partial\Omega_i \cap \partial\Omega_j \backslash \partial\Omega_D$ be the interface, that is, the set of all points that belong to at least two subdomains. Further denoting by Γ_h the set of all degrees of freedom (d.o.f.) which lie on the interface, we split these into two distinct sets, the set of primal degrees of freedom (Π) and the set of dual degrees of freedom (Δ) obtaining $\Gamma_h = \Delta \cup \Pi$. In this article, the primal variables are chosen as the subdomain vertices. For degrees of freedom in the interior, we use the index I. A depiction can be found in Fig. 1. Finally, we require the decomposition to be conforming, that is, the virtual element nodes coincide on the interface. We denote the local discrete virtual element spaces $V^h(\Omega_i) := V^h \cap H^1(\Omega_i)$. We further define the local discrete trace spaces $W_i := V^h(\partial\Omega_i \cap \Gamma_h)$ and let $W := \prod_{i=1}^N W_i$.

3.1 Standard BDDC

The BDDC method is defined as follows. We assume the following local ordering of the degrees of freedom which yields the following representation of the decomposed stiffness matrices, solution vectors, and right-hand sides

$$K = \begin{bmatrix} K_{II} & K_{I\Gamma} \\ K_{\Gamma I} & K_{\Gamma\Gamma} \end{bmatrix}, \qquad u = \begin{bmatrix} u_I \\ u_\Gamma \end{bmatrix}, \qquad \text{and} \qquad f = \begin{bmatrix} f_I \\ f_\Gamma \end{bmatrix},$$

where $K_{II} := \operatorname{diag}_{i=1}^{N} K_{II}^{(i)}$ and $K_{I\Gamma} := \operatorname{diag}_{i=1}^{N} K_{I\Gamma}^{(i)}$. In the same way, we have $u_I^T = (u_I^{(1)T}, \dots, u_I^{(N)T})$, and similarly for u_Γ, f_I, and f_Γ. We define the unassembled Schur complement and the reduced right-hand side by

$$S := S_{\Gamma\Gamma} = K_{\Gamma\Gamma} - K_{\Gamma I} K_{II}^{-1} K_{I\Gamma} \qquad \text{and} \qquad \mathrm{g} := \mathrm{g}_\Gamma = \mathrm{f}_\Gamma - \mathrm{K}_{\Gamma\mathrm{I}} \mathrm{K}_{\mathrm{II}}^{-1} \mathrm{f}_\mathrm{I}.$$

We denote by $R_\Pi^T = (R_\Pi^{(1)T}, R_\Pi^{(2)T}, \dots, R_\Pi^{(N)T})$ and $R_\Delta^T = (R_\Delta^{(1)T}, R_\Delta^{(2)T}, \dots, R_\Delta^{(N)T})$ the partial finite element assembly operators with values in $\{0, 1\}$, which assemble the system in the primal variables. We further define $R_\Gamma = \operatorname{diag}(R_\Delta, I_\Pi)$.

By assembling S and g in the primal variables we obtain

$$\widetilde{S} = \begin{bmatrix} I_\Delta & \\ & R_\Pi^T \end{bmatrix} S \begin{bmatrix} I_\Delta & \\ & R_\Pi \end{bmatrix} =: \begin{bmatrix} S_{\Delta\Delta} & \widetilde{S}_{\Delta\Pi} \\ \widetilde{S}_{\Pi\Delta} & \widetilde{S}_{\Pi\Pi} \end{bmatrix} \qquad \text{and} \qquad \widetilde{g} = \begin{bmatrix} I_\Delta & \\ & R_\Pi^T \end{bmatrix} g.$$

By assembling these systems in the dual variables we obtain the standard BDDC system

$$R_\Gamma^T \widetilde{S} R_\Gamma u_g = R_\Gamma^T \widetilde{g} \quad \Longleftrightarrow: \quad S_g u_g = g_g.$$

Next, we introduce scaling matrices $D^{(i)}$ belonging to their subdomains Ω_i. Consider the domain Ω_i which shares the edges $\mathcal{E}^{ij_1}, \dots, \mathcal{E}^{ij_n}$ with the subdomains $\Omega_{j_1}, \dots, \Omega_{j_n}$, respectively. Ordering $D^{(i)}$ according to the edges, yields $D^{(i)} = \operatorname{diag}_{m=1}^{n} D_{\mathcal{E}^{ijm}}^{[i]}$. We further require that the two scaling matrices belonging to an interface edge $\mathcal{E}^{ij}$ satisfy $D_{\mathcal{E}^{ij}}^{[i]} + D_{\mathcal{E}^{ij}}^{[j]} = I$, where I denotes the identity matrix. Here, we consider ρ-scaling [15]. With these scaling matrices, the scaled versions of R_Δ and R_Γ are defined as $R_{D,\Delta}^T = (R_{D,\Delta}^{(1)T}, \dots, R_{D,\Delta}^{(N)T})$ and $R_{D,\Gamma} = \operatorname{diag}(R_{D,\Delta}, I_\Pi)$, where $R_{D,\Delta}^{(i)} = D^{(i)} R_\Delta^{(i)}$. Finally the preconditioned BDDC system is given by

$$M^{-1} S_g u_g = M^{-1} g_g, \qquad \text{where} \qquad M^{-1} := M_{\mathrm{BDDC}}^{-1} := R_{D,\Gamma}^T \widetilde{S}^{-1} R_{D,\Gamma}.$$

4 Three-level BDDC

The three-level BDDC method is now characterized by an approximate solution of the linear Schur complement system $\widetilde{S} z = r$ which occurs in the preconditioner and thus has to be solved in each iteration for an arbitrary residual vector r. The approximation is based on a recursive application of the two-level BDDC preconditioner to the coarse problem using a coarser third level decomposition for the set of primal degrees of freedom. More precisely, the exact inverse $\widetilde{S}_{\Pi\Pi}^{-1}$ is replaced by the BDDC preconditioner on the third level within each application of $\widetilde{S}^{-1}$. We define the

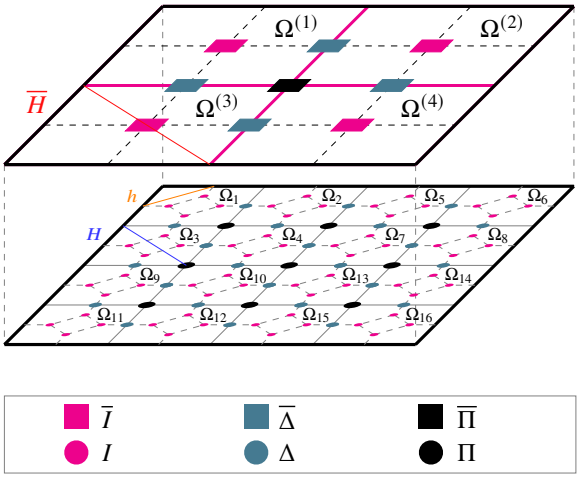


Fig. 2 Example of a three-level domain decomposition into 16 regular subdomains (bottom) and 4 regular subregions (top) with polygonal meshes on the subdomains. The interface $\overline{\Gamma}$ between subregions is marked in magenta.

operator $\widehat{S}$ such that the solution of $\hat{z} = \widehat{S}^{-1}r$ is the desired approximation of z and we will discuss its construction below. This allows us to define the three-level preconditioner

$$M_{3\mathrm{L}}^{-1} := R_{D,\Gamma}^T \widehat{S}^{-1} R_{D,\Gamma}.$$

We decompose Ω into N subregions $\Omega^{(j)}$ with diameters $\overline{H}$. Each subregion is the union of N_j subdomains, which we will denote by $\Omega_i^{(j)}$, $i = 1, \ldots, N_j$. To create a third level, we split the primal variables Π into different categories, just as in the two level case. Let $\overline{\Gamma} \subset \Pi$ be the interface between the subregions, that is, the primal variables belonging to two or more subregions. We further split the subregion interface into dual and primal variables obtaining $\overline{\Gamma} = \overline{\Delta} \cup \overline{\Pi}$. Here, the subregion primal variables are those that are connected to three or more subregions, that is, the vertices of the subregions. The remaining primal variables are denoted as $\overline{I}$. An example of a three-level decomposition is shown in Fig. 2. The operator $\widehat{S}$ is constructed by applying BDDC to the subregion decomposition. Instead of assembling the global Schur complement on all primal variables, the third-level decomposition is used to assemble a Schur complement on each subregion. For these subregion Schur complements, the BDDC preconditioner is built analogously to the second level and replaces the inverse action of $\widetilde{S}_{\Pi\Pi}^{-1}$ in each iteration of BDDC. In general (under certain assumptions on the coefficient distribution), the resulting system requires more PCG iterations to converge to the desired tolerance and shows higher condition numbers than the classical BDDC method but is more efficient due to being able to be computed in parallel. For more details we refer to [12, 16].

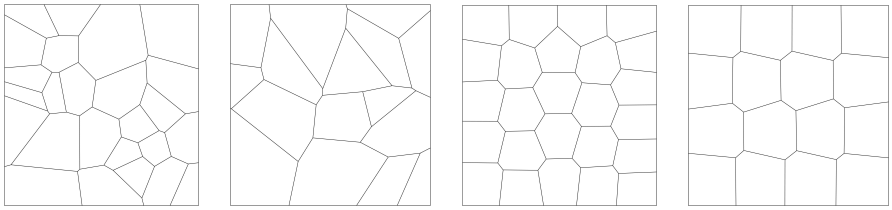
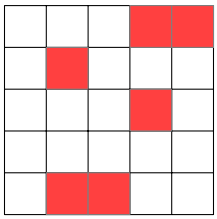


Fig. 3 Voronoi tessellations (two figures on the left) and Centroidal Voronoi Tessellations (CVT) (two figures on the right) with 25 and 16 elements respectively used for the numerical experiments.



BDDC			Three-level BDDC					
			$\overline{H}/H = 5,\ H/h \approx 5$			sr $= 5 \times 5,\ \overline{H}/H = 5$		
H/h	it	cond	sr	it	cond	$\overline{H}/H$	it	cond
≈ 25	15	2.38	5×5	21	4.09	5	21	4.09
≈ 50	16	2.40	10×10	25	4.73	10	24	5.12
≈ 75	16	2.40	15×15	26	4.90	15	28	6.43

Fig. 4 Condition numbers (cond) and iteration numbers (it) for BDDC and three-level BDDC with linear virtual element discretizations for a stationary diffusion problem. The coefficient distribution and the decomposition in subregions (sr) for the third level in the case of 5×5 subregions are shown on the left side. The Voronoi tessellation with 25 elements shown in Fig. 3 is used on each subdomain. The coefficient function is 10^6 on the red patches and 1 on the white ones.

5 Numerical results

For the numerical experiments, we consider $\Omega = [0,1]^2$ and regular domain decompositions into $m \times m$ quadratic subdomains and $M \times M$ quadratic subregions. To create a conforming decomposition, the meshes in Fig. 3 are mirrored across the subdomain interface. The PCG method is iterated until a relative reduction of the residual of 10^{-8} is reached. The results in Fig. 4 confirm the expected behavior of BDDC and three-level BDDC for the case of virtual elements, where using the three-level variant increases the iteration numbers and condition numbers slightly. Nonetheless, the method is fairly robust and scalable against increasing the number of subregions or their size. This is comparable to the finite element case. Similar results can be obtained for CVT meshes. Considering a subregion checkerboard pattern as the coefficient distribution, both meshtypes, and virtual elements of order $k = 1, 2$ in Table 1, we can observe a similar behavior.

To conclude, we have applied the three-level BDDC method to virtual element discretizations. The method shows a similar performance to its finite element counterpart. A proof of the three-level BDDC condition number bound to virtual element discretizations is in preperation.

Table 1 Condition numbers (cond) and iteration numbers (it) for three-level BDDC with virtual element discretizations with polynomial degree given by k for a coefficient distribution in a subregion (sr) checkerboard pattern with a contrast of 10^6. The subdomain meshes for $H/h \approx 5$ and $H/h \approx 4$ are shown in Fig. 3.

$\overline{H}/H = 5,\ H/h \approx 5$				
$k = 1$	Voronoi		CVT	
sr	it	cond	it	cond
5×5	14	2.33	13	2.33
10×10	14	2.33	13	2.33
15×15	14	2.35	13	2.35

sr $= 5 \times 5,\ H/h \approx 5$				
$k = 1$	Voronoi		CVT	
$\overline{H}/H$	it	cond	it	cond
5	14	2.33	13	2.33
10	15	2.38	14	2.38
15	15	2.39	14	2.39

sr $= 5 \times 5,\ \overline{H}/H = 5$				
$k = 1$	Voronoi		CVT	
H/h	it	cond	it	cond
≈ 5	14	2.33	14	2.33
≈ 10	17	3.19	17	3.22
≈ 15	19	3.65	19	3.76

$\overline{H}/H = 4,\ H/h \approx 4$				
$k = 2$	Voronoi		CVT	
sr	it	cond	it	cond
4×4	17	3.53	16	3.55
8×8	17	3.54	16	3.55
12×12	17	3.54	16	3.55

sr $= 4 \times 4,\ H/h \approx 4$				
$k = 2$	Voronoi		CVT	
$\overline{H}/H$	it	cond	it	cond
4	17	3.53	16	3.55
8	18	3.53	18	3.57
12	19	3.56	18	3.60

sr $= 4 \times 4,\ \overline{H}/H = 4$				
$k = 2$	Voronoi		CVT	
H/h	it	cond	it	cond
≈ 4	17	3.53	16	3.55
≈ 8	20	4.49	20	4.44
≈ 12	21	4.66	22	5.64

References

1. Ahmad, B., Alsaedi, A., Brezzi, F., Marini, L. D., and Russo, A. Equivalent projectors for virtual element methods. *Computers & Mathematics with Applications* **66**(3), 376 – 391 (2013).
2. Beirão da Veiga, L., Brezzi, F., Cangiani, A., Manzini, G., Marini, L. D., and Russo, A. Basic principles of virtual element methods. *Mathematical Models and Methods in Applied Sciences* **23** (2012).
3. Beirão da Veiga, L., Brezzi, F., Marini, L. D., and Russo, A. The hitchhiker's guide to the virtual element method. *Mathematical Models and Methods in Applied Sciences* **24**(08), 1541–1573 (2014).
4. Beirão da Veiga, L., Dassi, F., and Russo, A. High-order virtual element method on polyhedral meshes. *Computers & Mathematics with Applications* **74**(5), 1110–1122 (2017). SI: SDS2016 - Methods for PDEs.
5. Beirão da Veiga, L. and Manzini, G. A virtual element method with arbitrary regularity. *IMA Journal of Numerical Analysis* **34**(2), 759–781 (2013).
6. Bertoluzza, S., Pennacchio, M., and Prada, D. BDDC and FETI-DP for the virtual element method. *Calcolo* **54**, 1565–1593 (2017).
7. Bertoluzza, S., Pennacchio, M., and Prada, D. FETI-DP for the three dimensional virtual element method. *SIAM Journal on Numerical Analysis* **58**(3), 1556–1591 (2020).
8. Bevilacqua, T. and Scacchi, S. BDDC preconditioners for divergence free virtual element discretizations of the Stokes equations. *Journal of Scientific Computing* **92**(2) (2022).
9. Cangiani, A., Manzini, G., Russo, A., and Sukumar, N. Hourglass stabilization and the virtual element method. *International Journal for Numerical Methods in Engineering* **102**(3-4), 404–436 (2015).
10. Dassi, F., Zampini, S., and Scacchi, S. Robust and scalable adaptive BDDC preconditioners for virtual element discretizations of elliptic partial differential equations in mixed form. *Computer Methods in Applied Mechanics and Engineering* **391**, 114620 (2022).
11. de Dios, B. A., Lipnikov, K., and Manzini, G. The nonconforming virtual element method. *ESAIM: Mathematical Modelling and Numerical Analysis* **50**(3), 879–904 (2016).

12. Klawonn, A., Lanser, M., Rheinbach, O., and Weber, J. Preconditioning the coarse problem of BDDC methods - three-level, algebraic multigrid, and vertex-based preconditioners. *ETNA - Electronic Transactions on Numerical Analysis* **51**, 432–450 (2019).
13. Klawonn, A., Lanser, M., and Wasiak, A. Adaptive and frugal FETI-DP for virtual elements. *Vietnam Journal of Mathematics* (2022).
14. Klawonn, A., Radtke, P., and Rheinbach, O. A comparison of adaptive coarse spaces for iterative substructuring in two dimensions. *Electron. Trans. Numer. Anal.* **45**, 75–106 (2016).
15. Klawonn, A., Widlund, O. B., and Dryja, M. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. *SIAM Journal on Numerical Analysis* **40**(1), 159–179 (2002).
16. Tu, X. Three-level BDDC in two dimensions. *International Journal for Numerical Methods in Engineering* **69**(1), 33–59 (2007).

An Adaptive Overlapping Schwarz Algorithm for Isogeometric Analysis

Olof B. Widlund, Luca F. Pavarino, Simone Scacchi, and Stefano Zampini

1 Introduction

The algorithm considered in this paper is known as the RAGDSW – the Reduced Adaptive Generalized Dryja-Smith-Widlund – method. Following an idea of Clark Dohrmann, the GDSW algorithms were introduced in order to avoid the need for coarser meshes for overlapping Schwarz algorithms, see [4]; for a general introduction to Schwarz method see [9, Chap. 2–3]. This was accomplished by borrowing coarse spaces from another family of domain decomposition algorithms namely the iterative substructuring methods, see [9, Chap. 4–6]. These algorithms were later further refined decreasing the dimension of the coarse spaces, see [5, 6]. We note that the GDSW methods, without adaptation, can be used for problems for which only fully assembled stiffness matrices are available. Unfortunately, the adaptive variants require access to the stiffness matrices for individual subdomains, matrices that cannot be recovered from fully assembled matrices.

The purpose of our present work is to extend previous work on low order finite elements to isogeometric analysis (IgA) based on B-splines and NURBS (nonuniform rational B-splines) of arbitrary order p; for an introduction to IgA, see, e.g., [1]. Our elliptic problems are scalar elliptic problems and compressible linear elasticity in two

Olof B. Widlund
Courant Institute, 251 Mercer Street, New York, NY 10012, USA, e-mail: widlund@cims.nyu.edu

Luca F. Pavarino
Dipartimento di Matematica, Università degli Studi di Pavia, via Ferrata 5, 27100 Pavia, Italy, e-mail: luca.pavarino@unipv.it

Simone Scacchi
Dipartimento di Matematica, Università degli Studi di Milano, via Saldini 50, 20133 Milano, Italy, e-mail: simone.scacchi@unimi.it

Stefano Zampini
Extreme Computing Research Center, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia, e-mail: stefano.zampini@kaust.edu.sa

or three dimensions. We note that the NURBS are commonly used in computer aided design and we always assume that the domains of the elliptic problems considered can be represented exactly using this class of functions. With B-splines of order p and smoothness $k = p - 1$, there are p one-dimensional (1D) B-spline basis functions which differ from zero at any fixed internal knot. This is the origin of our *fat interfaces*; see further Section 3.

New coarse spaces for overlapping Schwarz methods are generated adaptively by solving generalized eigenvalue problems on subsets of the fat interface which subdivide the domain of the elliptic problem into subdomains. We call these subsets *eigensets*. The resulting eigenvectors with eigenvalues less than or equal to a tolerance are then extended by zero to the rest of the fat interface to provide Dirchlet data for the computation of basis functions, of minimum energy, for the coarse subspace of our Schwarz algorithms. We note that the AGDSW algorithms considered in [7] use eigensets of the interface based of equivalence classes directly related to subdomain vertices, edges, and faces while the RAGDSW algorithms of [8] use only one type of eigensets each associated with a vertex of the interface. The latter choice leads to considerably much smaller coarse subspaces. The two papers just cited, which have provided a foundation for our work, are for low order finite elements. These new algorithms improve the rate of convergence of the overlapping Schwarz algorithms, in particular, for cases when the material coefficients of our problems vary considerably; the work by Heinlein et al. have very strong results of that kind.

Our theoretical result provides an estimate of the condition number for our preconditioned conjugate gradient methods in terms of the tolerance used in the selection of eigenvectors of the generalized eigenvalue problems. Our experimental work with our algorithm is still in progress and will further be reported in a forthcoming paper, which will also provide complete proofs of our theoretical results.

2 The discrete problems

In this paper, the coarse space of the two-level additive Schwarz methods is given in terms of a coarse partition of the domain into non-overlapping subdomains $\{\Omega_k\}$. The union of the intersections of the boundaries of these subdomains form the interface Γ. In a reference domain, each non-overlapping subdomains, $\widehat{\Omega}_k$, is a preimage of Ω_k and is a square with a side length H each of which is partitioned into elements, with a side length of about h, by B-spline knots which we assume to form a quasi-uniform mesh. The coarse space of the pioneering paper [2] is associated with B-spline elements given in terms of the reference subdomains and of the same degree as those on the fine decomposition into small elements. In order to keep the dimension of the coarse space small, maximal smoothness of the B-splines, $k = p-1$, is chosen in that paper and in our work and this assumption also assures us that the coarse space is contained in the global space on the fine mesh which is also chosen to be of maximal smoothness. In the reference subdomains, tensor-product B-splines, $B_i^p(x)B_j^p(y)$ and $B_i^p(x)B_j^p(y)B_k^p(z)$, of order p in all coordinates in two and three dimensions

(2D and 3D), respectively, are used. The physical subdomains are the images of the reference subdomains under a mapping using NURBS. As already indicated the coarse spaces of our algorithm is chosen differently. Our local subproblems are given by Dirichlet problems on subdomains which share at least one layer of knots with all their neghbors. This overlap is measured by a parameter $r \geq 0$ where $r = 0$ represents minimal overlap and a value of $r > 0$ indicates that r layers of knots are shared between the neighboring local problems.

3 Equivalence classes, related subspaces, and preconditioners

For 2D, the subdomain vertices and edges of Γ are associated with equivalence classes of the knots of the fine mesh. The pairs of indices, (i, j), associated with the *fat interface*, Γ^{fat}, are determined by the set of 2D B-splines with values which differ from zero on part of the interface Γ. This fat interface, in turn, is divided into equivalence classes of *fat vertices* and *fat edges*.

Each interior subdomain vertex, and for $k = p - 1$, is associated with a fat vertex set of p^2 knots with p^2 B-spline basis functions that do not vanish at that vertex. Similarly, each interior subdomain edge is associated with a fat edge with basis functions which vanish at the two vertices at the end of the edge but differ from zero on part of a particular edge of Γ. Each fat edge can be viewed as being built from p thin edges of knots parallel to the subdomain edge in question in the reference domain. There is also an equivalence class of knots of the interior of any subdomain basis functions which vanish identically on the interface Γ. In 3D, there are also fat faces and the subspaces associated with fat vertices, fat edges, fat faces, and interiors of subdomains which are defined very similarly to the 2D case.

The eigensets of the RAGDSW algorithms are only of one type each associated with a subdomain vertex V; see Fig. 1. We denote such an eigenset by R and by Ω_R the interior of the union of the closure of four or eight subdomains which share the vertex at the center of this eigenset for 2D and 3D, respectively. In Fig. 1, the dots represent the locations of the maxima of the different B-spline basis functions associated with the fat interfaces. In the 2D case, the dots of R are those of the fat vertex and of the parts of the fat edges which are closer to V than to any other subdomain vertex. We note that for an even value of p, none of these dots would fall on the interface Γ. The figures would also look different if the parameter $k < p - 1$.

To decrease the cost of the computation of the elements of the matrices of the generalized eigenvalue problems (1), we can also use a subset of Ω_R making sure that all the basis functions associated with the set R are supported in the set that replaces Ω_R. We note that the theory, which we have developed, is equally valid in this case.

Considering the 3D case, we can now provide details on the construction of the set R. The B-spline knots and tensor-product B-splines associated with one of these eigensets are those of its fat vertex and the halves of the fat edges closest to the vertex and the nearest quarters of the adjacent fat faces; see Fig. 1. In case the

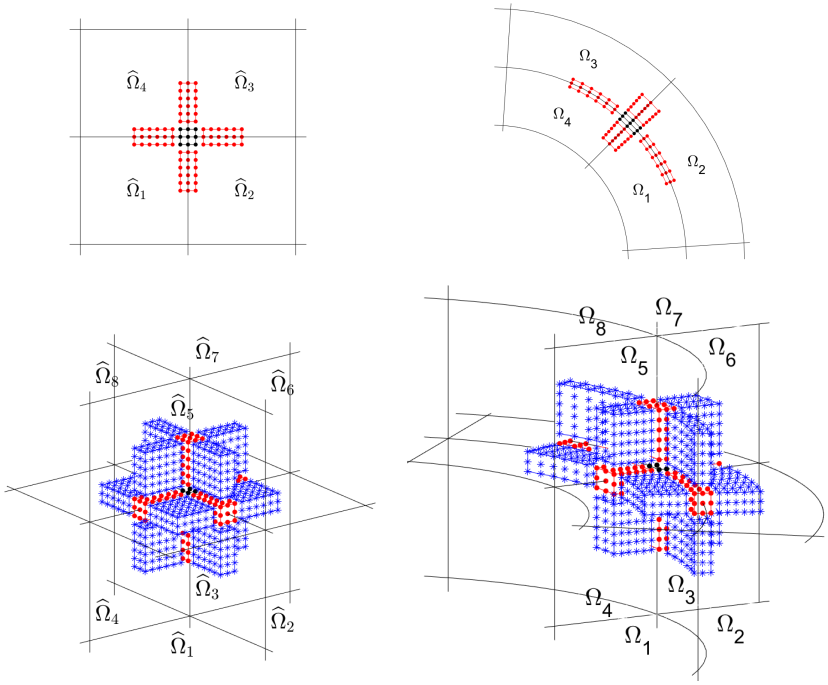


Fig. 1 RAGDSW eigensets for $p = 3, k = 2$. Top left: A 2D eigenset consisting of the union of a fat vertex (black dots) and adjcent halves of fat edges edges (red dots). Top right: the eigenset on a quarter-ring domain. Bottom left: A 3D eigenset consisting of the union of a fat vertex (black dots), halves of the adjacent fat edges (red dots), and quarters of adjacent fat faces (blue asterisks). Bottom right: The eigenset in a thick quarter-ring domain.

number of knots on a subdomain edge is not even, we allocate the knots furthest away from the relevant fat vertices to any of its closest eigensets. We also make similar minor modifications of the set of knots of the fat subdomain faces. The union of these eigensets, which do not overlap, covers the entire fat interface. We note that the knots originating from the fat vertex and subsets of fat edges and fat faces are displayed with different symbols and colors; we will need to partition this eigenset R, accordingly, when constructing the generalized eigenvalue problems.

The generalized eigenvalue problem associated with such an eigenset is of the form

$$S_{RR}^{\Omega_R} \tau_{\star,R} = \lambda_{\star,R} K_{RR}^{\Omega_R} \tau_{\star,R}, \tag{1}$$

defined by the Schur complement $S_{RR}^{\Omega_R}$ generated from the stiffness matrix K^{Ω_R} of the Neumann problem on Ω_R built from the four or eight subdomains sharing the eigenset or from a subset of Ω_R as indicated above. The Schur complement is generated by eliminating all degrees of freedom except those of R. The other matrix, $K_{RR}^{\Omega_R}$, is the principal minor, associated with R, of the same stiffness matrix

after eliminating the off-diagonal blocks that represent coupling between the sets from which the set R is constructed.

It is easy to see that the Schur complement is singular with one constant null vector for any scalar elliptic problem. For elasticity, there are three- and six-dimensional null spaces for 2D and 3D, respectively; they originate from the rigid body modes. This fact shows that the null space condition will always be satified. For elasticity and the case $p = 1$, the other matrix of the generalized eigenvalue problems can also be singular but this issue does not arise if $p \geq 2$.

Working with a tolerance $tol_R \geq 0$, we select all eigenvectors with $\lambda_{\star,R} \leq tol_R$, extend their values by zero to the rest of the fat interface and then compute their minimal energy extensions to find the coarse space elements $v_{\star,R}$ associated with R. For each eigenvector, this requires the solutions of a Dirichlet problem for each of the subdomains of Ω_R with a zero right hand side. Thus, any such basis function has values in the interior of the subdomains obtained by a minimal energy extension.

4 A theoretical result

Our theoretical result is an estimate of the condition number of the two-level additive Schwarz algorithm using the coarse space obtained from the coarse basis functions introduced above and one local subspace for each sudomain Ω_k. Any such local subspace is associated with all the knots of the subdomain and the part of the fat interface adjacent to the subdomain. Currently our proof does not work for smaller overlaps.

Theorem 1 *There are constants C_1 and C_2 such that the condition number of the two-level additive Schwarz operator P_{add} satifies*

$$\kappa(P_{add}) \leq C_1(1 + C_2/tol). \tag{2}$$

Here tol is the smallest tolerance tol_R used for the generalized eigenvalue problems and C_1 and C_2 are computable constants independent of the number of subdomains, the dimension of the subprobems, and the coefficients of our elliptic problems.

Our proof relies to a large extent on the work reported in the two papers by Heinlein et al.

5 Numerical results

In this section, we report on numerical experiments with the isogeometric RAGDSW preconditioner for the 2D Poisson equation on a quarter-ring domain, discretized by isogeometric NURBS spaces with mesh size h, polynomial degree p, regularity $k = p - 1$, and the overlap parameter r. The domain is decomposed into N nonoverlapping subdomains of characteristic size H. The linear systems of equations arising

from the discretizations are solved by the PCG algorithm accelerated by the isogeometric RAGSW preconditioner, with a zero initial guess and a stopping criterion of a 10^{-6} reduction of the Euclidean norm of the PCG residual. In the tests, we study how the convergence rate of the RAGDSW preconditioner depends on the parameters h, N, p, and r The numerical tests have been performed with a MATLAB code based on the GeoPDEs library [3]. We expect to be able to show results for linear elasticity and much larger 3D problems in a forthcoming paper.

Table 1 RAGDSW preconditioner in 2D quarter-ring domain: condition number κ_2, iteration count, it, and coarse problem size N_Π as a function of the number of subdomains N and mesh size h. Fixed spline parameters $p = 2$, $k = 1$, minimal overlap parameter $r = 0$, $tol = 0.1$.

RAGDSW preconditioner, quarter-ring domain $p = 2, k = 1, r = 0, tol = 0.1$															
	$1/h = 8$			$1/h = 16$			$1/h = 32$			$1/h = 64$			$1/h = 128$		
N	κ_2	it.	N_Π	κ_2	it.	N_Π	κ_2	it.	N_Π	κ_2	it.	N_Π	κ_2	it.	N_Π
2×2	3.60	10	1	6.69	14	2	9.23	17	3	16.10	22	4	30.60	29	4
4×4				7.63	16	9	8.77	18	18	16.16	24	18	16.31	23	36
8×8							10.85	20	49	10.38	19	98	18.90	25	98
16×16										13.73	22	225	11.40	19	450
32×32													16.23	24	961

Table 2 RAGDSW preconditioner in 2D quarter-ring domain: condition number κ_2, iteration counts it. and coarse problem size N_Π as a function of the number of subdomains N and mesh size h, Fixed spline parameters $p = 3$, $k = 2$, minimal overlap parameter $r = 0$, $tol = 0.1$.

RAGDSW preconditioner, quarter-ring domain $p = 3, k = 2, r = 0, tol = 0.1$															
	$1/h = 8$			$1/h = 16$			$1/h = 32$			$1/h = 64$			$1/h = 128$		
N	κ_2	it.	N_Π	κ_2	it.	N_Π	κ_2	it.	N_Π	κ_2	it.	N_Π	κ_2	it.	N_Π
2×2	7.15	14	1	9.72	15	2	16.87	18	3	20.10	20	4	26.63	22	8
4×4				11.79	22	9	14.41	23	18	17.34	24	27	22.40	24	36
8×8							17.16	27	49	14.53	23	98	17.55	23	147
16×16										22.15	29	225	14.91	23	450
32×32													26.42	30	961

5.1 Scalability in N and quasi-optimality in H/h

The condition number κ_2 of the RAGDSW preconditioned system and the conjugate gradient iteration count, it, are reported in Tables 1 and 2 as a function of the number of subdomains N and the mesh size h for $p = 2$ and $p = 3$, respectively. In both cases, we consider the maximal regularity $k = p - 1$. We set the adaptive tolerance to $tol = 0.1$. The results show that the proposed preconditioner is scalable, since, moving along the diagonals of each table, both the condition number and iteration count exhibit a moderate increase that seems to level off and approach a constant

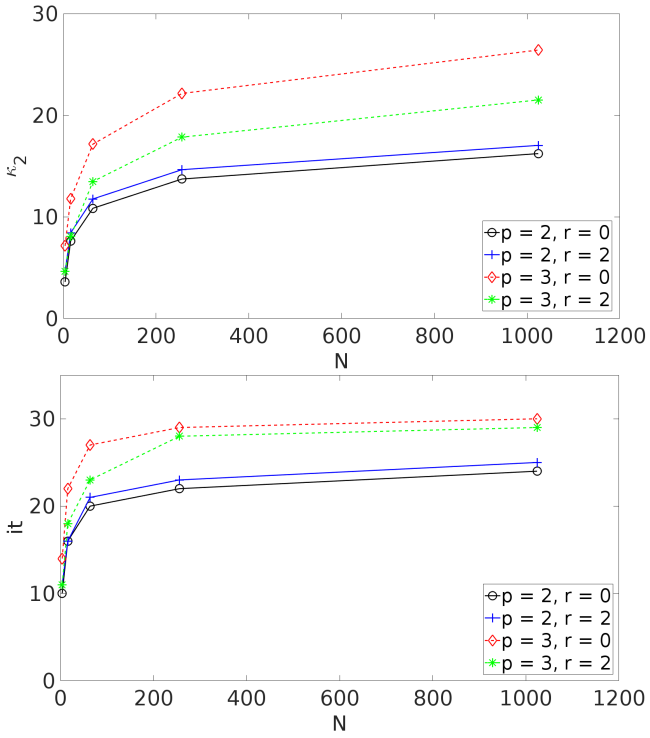


Fig. 2 Scalability of RAGDSW preconditioner in 2D quarter-ring domain: condition number κ_2 (top) and iteration counts it. (bottom) as a function of the number of subdomains N, fixed ratio $H/h = 4$, overlap parameter $r = 0$ and $r = 2$, $tol = 0.1$.

value; see also Fig. 2. We note that for the largest problems of these tables, the dimension of the coarse space appears to increase about four times when the number of subdomains increases by four.

5.2 Dependence on p

In this test, we study the robustness of the RAGDSW preconditioner with respect to the spline polynomial degree p. The quarter-ring domain is discretized with a mesh size $h = 1/64$ and $N = 4 \times 4$ subdomains, while the degree p varies from 2 to 8 and the regularity $k = p - 1$ is always maximal. The results reported in Table 3 show that the condition numbers and iteration counts exhibit a moderate increase up to $p = 5$. They then start to increase, more slowly when the adaptive tolerance tol is large and the coarse space sufficiently rich. We note that the condition numbers without preconditioning – not reported – grows very rapidly with p.

Table 3 RAGDSW preconditioner in 2D quarter-ring domain: condition number κ_2 and iteration counts it. as a function of the spline polynomial degree p and adaptive tolerance parameter tol, with maximal regularity $k = p - 1$, fixed number of subdomains $N = 4 \times 4$, $1/h = 64$, $r = 0$. N_Π denotes the dimension of the coarse space.

RAGDSW prec., quarter-ring domain $N = 4 \times 4$, $1/h = 64$, $r = 0$													
		$tol = 0.05$			$tol = 0.1$			$tol = 0.2$			$tol = 0.5$		
p	dofs	κ_2	it	N_Π	κ_2	it	N_Π	κ_2	it	N_Π	κ_2	it	N_Π
2	4356	16.16	24	18	16.16	24	18	10.76	18	36	10.76	18	36
3	4489	22.21	27	18	17.34	24	27	12.20	20	36	8.56	17	81
4	4624	17.56	25	18	14.98	21	27	10.60	19	63	9.24	17	144
5	4761	29.15	31	18	20.05	26	63	13.51	22	108	11.48	20	225
6	4900	52.77	40	54	31.18	32	99	26.33	29	288	24.09	27	324
7	5041	35.19	41	99	26.17	33	252	25.14	31	441	25.14	31	441
8	5184	135.56	73	243	89.68	57	513	82.49	55	576	82.49	55	576

References

1. Beirão da Veiga, L., Buffa, A., Sangalli, G., and Vázquez, R. Mathematical analysis of variational isogeometric methods. *Acta Numer.* **23**, 157–287 (2014).
2. Beirão da Veiga, L., Cho, D., Pavarino, L. F., and Scacchi, S. Overlapping Schwarz methods for isogeometric analysis. *SIAM J. Numer. Anal.* **50**(3), 1394–1416 (2012).
3. De Falco, C., Reali, A., and Vázquez, R. GeoPDEs: a research tool for isogeometric analysis of PDEs. *Adv. Eng. Softw.* **42**(12), 1020–1034 (2011).
4. Dohrmann, C. R. and Widlund, O. B. An overlapping Schwarz algorithm for almost incompressible elasticity. *SIAM J. Numer. Anal.* **47**(4), 2897–2923 (2009).
5. Dohrmann, C. R. and Widlund, O. B. Hybrid domain decomposition algorithms for compressible and almost incompressible elasticity. *Internat. J. Numer. Methods Engrg.* **82**(2), 157–183 (2010).
6. Dohrmann, C. R. and Widlund, O. B. On the design of small coarse spaces for domain decomposition algorithms. *SIAM J. Sci. Comput.* **39**(4), A1466–A1488 (2017).
7. Heinlein, A., Klawonn, A., Knepper, J., and Rheinbach, O. Adaptive GDSW coarse spaces for overlapping Schwarz methods in three dimensions. *SIAM J. Sci. Comput.* **41**(5), A3045–A3072 (2019).
8. Heinlein, A., Klawonn, A., Knepper, J., Rheinbach, O., and Widlund, O. B. Adaptive GDSW coarse spaces of reduced dimension for overlapping Schwarz methods. *SIAM J. Sci. Comput.* **44**(3), A1176–A1204 (2022).
9. Toselli, A. and Widlund, O. *Domain Decomposition Methods - Algorithms and Theory*, *Springer Series in Computational Mathematics*, vol. 34. Springer-Verlag, Berlin Heidelberg New York (2005).

Part III
Contributed Talks

Numerical Assessment of PML Transmission Conditions in a Domain Decomposition Method for the Helmholtz Equation

Niall Bootland, Sahar Borzooei, Victorita Dolean, and Pierre-Henri Tournier

1 Introduction

Finite element discretisations of large-scale time-harmonic wave problems typically lead to ill-conditioned linear systems with a large number of unknowns. A promising class of methods to solve such huge systems in parallel, both in terms of convergence and computing time, is offered by domain decomposition methods (DDMs). These approaches rely on a partition of the computational domain into smaller subdomains, leading to subproblems of smaller sizes which are manageable by direct solvers. A robust domain decomposition (DD) preconditioner for large-scale computations is given in [4]. However, improving the efficiency of such preconditioners continues to be a challenging issue. Recent work has shown that transmission operators based on perfectly matched layers (PMLs) are well-suited for two-dimensional configurations of the Helmholtz problem within non-overlapping DDMs [10]. Further, PMLs have been used successfully as transmission conditions in DDMs applied to geophysical applications modelled by the Helmholtz equation [11].

In this work, we present an efficient PML-based Schwarz-type preconditioner with overlapping subdomains to solve large-scale wave propagation problems. We then assess the performance of this one-level DD algorithm, where the transmission conditions at the boundaries between subdomains are PML conditions in order to

Niall Bootland
STFC Rutherford Appleton Laboratory, Harwell Campus, UK, e-mail: niall.bootland@stfc.ac.uk

Sahar Borzooei
University Côte d'Azur, CNRS, LJAD, France, e-mail: Sahar.Borzooei@univ-cotedazur.fr

Victorita Dolean
University Côte d'Azur, CNRS, LJAD, France, and University of Strathclyde, UK, e-mail: work@victoritadolean.com

Pierre-Henri Tournier
Sorbonne Université, CNRS, Université Paris Cité, Inria, Laboratoire Jacques-Louis Lions (LJLL), F-75005 Paris, France, e-mail: pierre-henri.tournier@sorbonne-universite.fr

provide a better approximation to the transparent boundary operator. Further, we will investigate the convergence properties and compare them with the use of more standard impedance transmission conditions.

2 Mathematical model

As an underlying model we consider the Helmholtz equation in free space, given by

$$-(\Delta + k^2(\mathbf{x}))u(\mathbf{x}) = g(\mathbf{x}), \quad \mathbf{x} \in \Omega \tag{1}$$

for $\Omega = \mathbb{R}^d$ in dimension $d = 2$ or 3, where $k(\mathbf{x}) = \frac{2\pi}{\lambda}$ is the wavenumber, with $\lambda = \frac{c}{f}$ being the wavelength, $c(\mathbf{x})$ the wave speed and f the frequency. Note that the angular frequency is then defined as $\omega = 2\pi f$. To close the problem we prescribe the physically relevant condition at infinity known as the far field Sommerfeld radiation condition

$$\lim_{|\mathbf{x}|\to\infty} |\mathbf{x}|^{\frac{d-1}{2}} \left(\frac{\partial u}{\partial |\mathbf{x}|} - iku \right) = 0. \tag{2}$$

Since we can not compute on the whole free space domain, we consider truncating to an appropriate finite domain. Let us suppose now that $\Omega \subset \mathbb{R}^d$ represents a finite computational domain capturing the physical area of interest. A typical approach, as in [7], is to replace the Sommerfeld condition (2) with the first-order approximation

$$\frac{\partial u}{\partial \mathbf{n}} + iku = 0, \quad \mathbf{x} \in \partial\Omega, \tag{3}$$

known as the impedance (or Robin) boundary condition (Imp BC), with $\mathbf{n}$ being the unit outward normal to the boundary $\partial\Omega$. This enables the appropriate description of wave behaviour in a bounded domain. The finite element discretisation of (1) can then be written as a linear system $A\mathbf{u} = \mathbf{b}$.

2.1 PML formulation

Perfectly matched layers (PMLs) were introduced as a better alternative to absorbing boundary conditions (ABCs) by Berenger [2] to achieve a higher accuracy in domain truncation by eliminating undesired numerical reflections from boundaries, leading to exponential convergence of the numerical solution to the exact solution [1]. PML implementation is done by stretching Cartesian coordinates such that the stretching is defined in a layer surrounding Ω, as in [6], giving a larger computational domain Ω_{PML}. In this regard, we assume the boundaries of the artificially truncated domain Ω are aligned with the coordinate axes.

For simplicity of exposition, we will focus on truncating the problem in the x direction. Let us suppose that the PML extends from the boundary of our domain of interest at $x = a$ to $x = a^*$ and Dirichlet conditions are imposed on $x = a^*$. The stretched coordinate mapping used is given by

$$\frac{\partial}{\partial x_{\text{pml}}} \mapsto \frac{1}{1 - \frac{i}{\omega}\sigma(x)} \frac{\partial}{\partial x}, \quad \text{where} \quad \begin{cases} \sigma(x) = 0 & \text{if } x < a, \\ \sigma(x) > 0 & \text{if } a < x < a^*. \end{cases} \tag{4}$$

In the PML region, where $\sigma(x) > 0$, oscillating solutions turn into exponentially decaying ones. In the original domain Ω, $\sigma(x) = 0$ so that the underlying equation is unchanged. In this work we will study three different stretching functions [3, 8], namely

$$\sigma_{-1}(x) = \frac{1}{a^* - x}, \quad \sigma_{-2}(x) = \frac{2}{(a^* - x)^2}, \quad \text{and} \quad \sigma_2(x) = \alpha(a^* - x)^2. \tag{5}$$

In $\sigma_2(x)$, α is experimentally chosen to take the value 30 in our simulations. To incorporate a PML into other coordinate directions, we simply apply equivalent one-dimensional transformations to obtain $\frac{\partial}{\partial y_{\text{pml}}}$ and $\frac{\partial}{\partial z_{\text{pml}}}$. At the corners of the extended computational domain Ω_{PML} we will have PML regions that stretch along two or three directions simultaneously; this will not generate any problems. Implementing this mapping in, for instance, a three-dimensional domain requires a slight change to the Helmholtz equation (1) over Ω_{PML}, resulting in the Laplace operator Δ being replaced by following operator which stretches in the PMLs

$$\Delta_{\text{pml}} = \frac{\partial^2}{\partial {x_{\text{pml}}}^2} + \frac{\partial^2}{\partial {y_{\text{pml}}}^2} + \frac{\partial^2}{\partial {z_{\text{pml}}}^2}. \tag{6}$$

2.1.1 Accuracy assessment for PMLs

In this section we will solve the Helmholtz equation when PMLs are applied as global boundary conditions for a 2D domain of length 10λ in each direction. We will compute the L^2 relative error with respect to the analytical exact solution and compare it with the situation where impedance boundary conditions are used instead. We consider a scattering problem of a plane wave by a circular obstacle, with a Dirichlet boundary condition on the boundary of the obstacle, shown in Figure 1. First, in Table 1, we compare different stretching functions σ with the utilization of higher order P3 Lagrange finite elements and discretization of $n_\lambda = 20$ points per wavelength. We find that the best accuracy is obtained with σ_{-1} and so we continue our tests with this function here. Within our tests we vary the number of points per wavelength n_λ and the PML length in order to investigate their relative effect on the resulting error. Results are detailed in Table 2. We see that, except for $n_\lambda = 5$, PMLs provide higher accuracy compared to impedance boundary conditions, even when the length of the PMLs incorporate only 0.1λ. Moreover, for a fixed PML length,

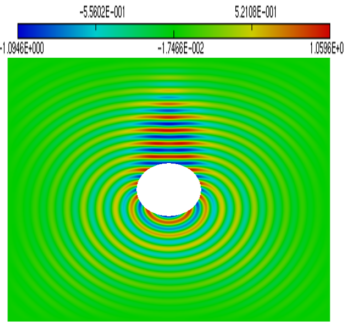


Fig. 1 Plane wave excitation solution when using PMLs as global boundary conditions in 2D.

Table 1 L^2 relative error for different stretching functions σ with PML length $L_{\text{pml}} = \lambda$. The radius of the circular obstacle is $R = \lambda$.

Stretching functions		
σ_{-1}	σ_{-2}	σ_2
0.00112	0.001517	0.075495

Table 2 L^2 relative error for different PML lengths with σ_{-1} or impedance boundary conditions (Imp BCs), $R = \lambda$.

	PML length										
n_λ	0.1λ	0.2λ	0.3λ	0.5λ	λ	2λ	3λ	4λ	5λ	10λ	Imp BCs
5	0.10408	0.02441	0.02684	0.02120	0.01685	0.01251	0.00927	0.00741	0.00605	0.00265	0.05118
10	0.01354	0.01011	0.00665	0.00534	0.00425	0.00311	0.00235	0.00184	0.00150	0.00067	0.04642
20	0.00893	0.00467	0.00268	0.00159	0.00112	0.00078	0.00059	0.00046	0.00038	0.00017	0.04620
30	0.00797	0.00320	0.00175	0.00083	0.00050	0.00035	0.00026	0.00021	0.00017	0.00007	0.04620
40	0.00617	0.00246	0.00121	0.00056	0.00029	0.00020	0.00015	0.00012	0.00009	0.00006	0.046212
50	0.00578	0.00192	0.00096	0.00041	0.00020	0.00013	0.00010	0.00008	0.00006	0.00003	0.046216

and again even for 0.1λ, the error still decreases when increasing n_λ, whereas for impedance boundary conditions the error is dominated by the domain truncation even for $n_\lambda = 5$. Of course, the error also decreases significantly with increasing PML length, all the way down to 3×10^{-5} for $n_\lambda = 50$ and 10λ.

2.2 Domain decomposition preconditioner

A preconditioner M^{-1} is a linear operator whose use aims to reduce ill-conditioning and allow faster convergence of an iterative solver. Usually (but not always) this approximates A^{-1} and has a matrix–vector product that is much cheaper to compute than solving the original linear system. To this end, we employ right preconditioning within GMRES to solve our discretised linear system, namely by solving

$$AM^{-1}\mathbf{y} = \mathbf{f}, \quad \text{where} \quad \mathbf{u} = M^{-1}\mathbf{y}. \tag{7}$$

Right preconditioning benefits from minimising a residual that is independent of the preconditioner, unlike left-preconditioned GMRES. Overlapping Schwarz methods come with the advantages of better convergence and easier implementation compared to substructuring methods. Furthermore, contrary to non-overlapping methods, corners do not need specific treatment. Overlapping methods are also a natural choice to consider when using PML transmission conditions, as the added PML can be naturally included in the overlap region. In this work we use the optimised restricted additive Schwarz (ORAS) domain decomposition preconditioner, given by

$$M_{\text{ORAS}}^{-1} = \sum_{s=1}^{N_{\text{sub}}} R_s^T D_s A_s^{-1} R_s, \tag{8}$$

where N_{sub} is the number of overlapping subdomains Ω_s into which the domain Ω is decomposed. To define the matrices present in (8), let $\mathcal{N}$ be an ordered set of the unknowns of the whole domain and let $\mathcal{N} = \bigcup_{s=1}^{N_{\text{sub}}} \mathcal{N}_s$ be its decomposition into the non-disjoint ordered subsets corresponding to the different overlapping subdomains Ω_s. Further, define $N = |\mathcal{N}|$ and $N_s = |\mathcal{N}_s|$. The $N_s \times N_s$ matrices A_s stem from the discretisation of local boundary value problems on Ω_s with transmission conditions chosen as either Robin or PML conditions to be implemented at the subdomain interfaces. The $N_s \times N$ matrix R_s is the Boolean restriction matrix from Ω to subdomain Ω_s while R_s^T is then the extension matrix from subdomain Ω_s to Ω. The $N_s \times N_s$ diagonal matrices D_s provide a discrete partition of unity, i.e., are such that $\sum_{s=1}^{N_{\text{sub}}} R_s^T D_s R_s = I$. See, e.g., [5, 9] for further details on such methods. PMLs are introduced as transmission conditions on the interface boundaries of the local subdomains in [10]. In this approach, the PML region is included strictly inside the overlap, the PML region being the outermost layers within each overlapping domain. This ensures that there is enough overlap for the approach to be efficient and sufficient length of the PML for a good approximation of the interface transmission condition.

3 Numerical results

3.1 PML as transmission conditions for a 2D domain

As a simple model, we consider excitation by a Gaussian point source, $S(x, y) = e^{-30k((x-5)^2+(y-5)^2)}$, in the center of a 2D domain of size $[0, 10] \times [0, 10]$, as shown in Figure 2 (left). The convergence rate is studied when either PML or impedance conditions are imposed as global boundary conditions (BCs) or interface conditions (ICs). This leads to four different configurations in total[1]. To discretise we employ P3 finite elements on regular grids with $n_\lambda = 15$.

[1] In 2D we present results only with PMLs for the global BCs; a comparison with impedance BCs will be given later for the full 3D problem in Table 5.

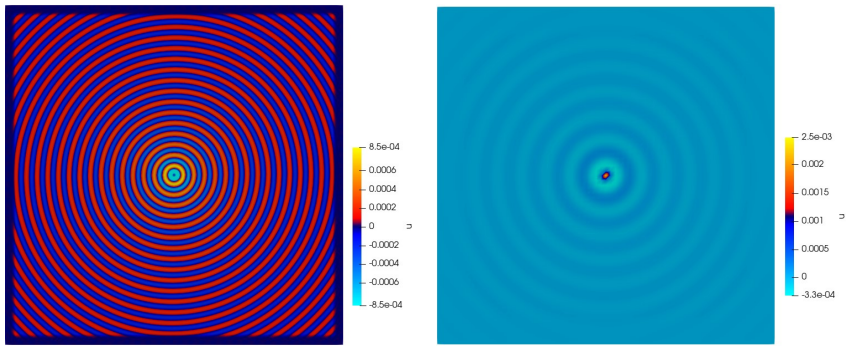


Fig. 2 The solution of the 2D, with $f = 3Hz$ (left), and 3D, with $f = 1Hz$ (right), point source excitation problem with PML boundary conditions.

In our tests, we set the wave speed to $c = 1$ and we vary the frequency f from 3Hz to 10Hz, which leads to a number of wavelengths in the domain ranging from 30 to 100. That also results in different values of #DoFs, which represents the number of degrees of freedom in the discrete problem. We decompose the global domain into either $N = 8 \times 8 = 64$ or $N = 10 \times 10 = 100$ square subdomains and use interface PML regions of length L_{pmli}.

In Table 3, simulations results are given using the σ_{-1} stretching function[2], where the interface PML length is $L_{\text{pmli}} = 1h$ and $h = \frac{\lambda}{n_\lambda}$ is the mesh size. The PML length on the global boundary is chosen to be $L_{\text{pml}} = 2\lambda$ and the number of overlapping layers of elements between subdomains is varied from 2 to 8 layers. We first observe that when interface PMLs are used we always require fewer iterations compared to using impedance ICs. Secondly, with the impedance condition the iteration counts increase with frequency f, but this is not the case when using interface PMLs where iteration counts remain insensitive to f. Finally, we note that an overlap of 4 layers is sufficient here for the PMLs with little benefit seen as we increase the overlap further while for the impedance condition a larger overlap is needed to continually reduce the iteration counts.

In Table 4, the simulations with $f = 3$ Hz are repeated but now with $L_{\text{pmli}} = 5h$. Note that for the appropriate transmission of data between subdomains, we should consider the length of the overlap to be larger than the length of the PML region. This can be seen in Table 4 where an overlap of more than 5 layers is required for good convergence. Comparing the number of iterations, when the overlap is sufficient, with those in Table 3, we can see a small improvement in the convergence when using a larger interface PML region. Note that the one-level preconditioner is by nature not robust, in the sense that the number of iterations usually depends on the number of subdomains. That is to say, the number of iterations does not depend only on the quality of the approximation of the absorbing interface conditions, which as we can see from Table 2 is already good when using PMLs of small length.

[2] A comparison with other choices of stretching function σ will be given later in Table 7.

Table 3 Iteration counts for varying frequency f and choices of ICs, discretised using P3 elements with $n_\lambda = 15$. Within the PMLs we use σ_{-1}, $L_{\text{pmli}} = 1h$ and $L_{\text{pml}} = \lambda$.

				Overlap							
				$N = 64$				$N = 100$			
BCs	ICs	f (Hz)	#DoFs	2	4	6	8	2	4	6	8
PML	Imp	3	2,076,481	52	48	44	41	65	58	53	51
PML	PML			39	33	33	32	49	42	40	42
PML	Imp	5	5,480,281	63	59	56	54	70	65	60	57
PML	PML			41	35	33	34	49	42	41	42
PML	Imp	10	21,077,281	69	67	63	61	77	71	67	68
PML	PML			40	34	33	34	48	41	41	41

Table 4 Iteration counts for $f = 3\,Hz$ and varying choices of ICs, discretised using P3 elements with $n_\lambda = 15$. Within the PMLs we use σ_{-1}, $L_{\text{pmli}} = 5h$ and $L_{\text{pml}} = \lambda$.

				Overlap							
				$N = 64$				$N = 100$			
BCs	ICs	f (Hz)	#DoFs	2	4	6	8	2	4	6	8
PML	Imp	3	2,076,481	52	48	44	41	65	58	53	51
PML	PML			110	64	31	31	140	80	39	39

3.2 PML as transmission conditions for a 3D domain

In this section, we consider a similar Gaussian point source excitation in the center of the 3D domain, $S(x, y, z) = e^{-30k((x-5)^2+(y-5)^2+(z-5)^2)}$; see Figure 2 (right). For this problem we discretise with P2 finite elements and use $n_\lambda = 5$ and $L_{\text{pml}} = 2\lambda$. When recording iteration counts in this section, the use of – means the simulation failed due to memory limitations while • indicates a lack of convergence in 2000 iterations. In Table 5 we use σ_{-1} and compare all four combinations of BCs and ICs when $L_{\text{pmli}} = 1h$ and $f = 1\,\text{Hz}$, this results in #DoFs = 2,803,221. We observe that using PML rather than impedance conditions reduces iteration counts both when used as BCs or ICs, in particular, when swapping from impedance for both BCs and ICs to PMLs we see at least a $2/3$ reduction in iterations. Furthermore, using PML BCs again provides a somewhat more accurate solution when comparing L^2 relative error with respect to the analytical exact solution. Here, we consider $N = 6 \times 6 \times 5 = 180$ and $N = 7 \times 7 \times 6 = 294$ subdomains.

In Table 6, simulations for the full PML case are repeated for different lengths of L_{pmli}. Again we see the overlap should be larger than the interface PML length and, when so, iteration counts slowly decrease as L_{pmli} increases.

Finally, we compare different stretching functions σ for the case of $L_{\text{pmli}} = 4h$. The results in Table 7 show that the best convergence is provided when we choose σ_{-1}. While the iteration counts when using σ_{-2} have only a small increase, it is always more effective to use σ_{-1}. The convergence observed for σ_2 is much poorer, demon-

Table 5 Iteration counts and L^2 error for $f = 1$ Hz and varying choices of BCs and ICs, discretised using P2 elements with $n_\lambda = 5$. Within the PMLs we use σ_{-1}, $L_{\text{pmli}} = 1h$ and $L_{\text{pml}} = 2\lambda$.

		Overlap								
		$N = 180$				$N = 294$				
BCs	ICs	2	4	6	8	2	4	6	8	L^2 relative error
Imp	Imp	40	31	27	–	45	35	32	34	0.211853
Imp	PML	36	29	26	–	42	35	29	31	
PML	Imp	30	22	20	–	33	25	23	21	0.0709828
PML	PML	24	20	18	–	27	23	20	19	

Table 6 Iteration counts for $f = 1$ Hz with PML BCs and ICs varying the PML interface length L_{pmli}, discretised using P2 elements with $n_\lambda = 5$. Within the PMLs we use σ_{-1} and $L_{\text{pml}} = 2\lambda$.

			Overlap							
			$N = 180$				$N = 294$			
BCs	ICs	L_{pmli}	2	4	6	8	2	4	6	8
PML	PML	$1h$	24	20	18	–	27	23	20	19
PML	PML	$2h$	30	19	17	–	34	22	19	18
PML	PML	$4h$	37	21	15	–	42	24	17	15
PML	PML	$6h$	38	21	18	–	43	24	21	15

Table 7 Iteration counts for $f = 1$ Hz and varying choice of ICs and PML stretching function σ, discretised using P2 elements with $n_\lambda = 5$. Within the PMLs we use $L_{\text{pmli}} = 4h$ and $L_{\text{pml}} = 2\lambda$.

			Overlap							
		Stretching	$N = 180$				$N = 294$			
BCs	ICs	function	2	4	6	8	2	4	6	8
PML	Imp	σ_{-1}	30	22	20	–	33	25	23	21
PML	PML		37	21	15	–	42	24	17	15
PML	Imp	σ_{-2}	33	28	24	–	38	30	27	26
PML	PML		•	33	19	–	•	38	23	19
PML	Imp	σ_2	•	•	973	–	•	•	1984	1201
PML	PML		•	•	•	–	•	•	•	•

strating the importance of choosing a suitable stretching function in order to be advantageous in the domain decomposition preconditioner. In our tests σ_{-1} provided the best choice and justifies its use in our previous simulations.

4 Conclusion

In this work, we have introduced the use of PMLs as interface conditions within an overlapping domain decomposition solver for Helmholtz equations. With the choice of PMLs as interface conditions, better convergence is achieved compared to using impedance conditions. Results on 2D and 3D model problems show the utility of the approach with a suitable choice of stretching function.

References

1. Bao, G. and Wu, H. Convergence analysis of the perfectly matched layer problems for time-harmonic maxwell's equations. *SIAM journal on numerical analysis* **43**(5), 2121–2143 (2005).
2. Berenger, J.-P. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics* **114**(2), 185–200 (1994).
3. Bermudez, A., Hervella-Nieto, L., Prieto, A., and Rodríguez, R. Perfectly matched layers for time-harmonic second order elliptic problems. *Archives of Computational Methods in Engineering* **17**, 77–107 (2010).
4. Bonazzoli, M., Dolean, V., Graham, I., Spence, E., and Tournier, P.-H. Domain decomposition preconditioning for the high-frequency time-harmonic maxwell equations with absorption. *Mathematics of Computation* **88**(320), 2559–2604 (2019).
5. Bonazzoli, M., Dolean, V., Hecht, F., and Rapetti, F. Overlapping schwarz preconditioning for high order edge finite elements: Application to the time-harmonic maxwell's equations (2016).
6. Borzooei, S., Dolean, V., Migliaccio, C., Tournier, P.-H., and Pichot, C. A fast and precise parallel numerical model using pml for maxwell's equations. In: *2022 IEEE International Symposium on Antennas and Propagation and USNC-URSI Radio Science Meeting (AP-S/URSI)*, 153–154. IEEE (2022).
7. Borzooei, S., Dolean, V., Tournier, P.-H., and Migliaccio, C. Solution of time-harmonic maxwell's equations by a domain decomposition method based on pml transmission conditions. *arXiv preprint arXiv:2211.04912* (2022).
8. Marburg, S. and Nolte, B. *Computational acoustics of noise propagation in fluids: finite and boundary element methods*, vol. 578. Springer (2008).
9. Nataf, F. Interface connections in domain decomposition methods. *Modern methods in scientific computing and applications* 323–364 (2002).
10. Royer, A., Geuzaine, C., Béchet, E., and Modave, A. A non-overlapping domain decomposition method with perfectly matched layer transmission conditions for the helmholtz equation. *Computer Methods in Applied Mechanics and Engineering* **395**, 115006 (2022).
11. Tournier, P.-H., Jolivet, P., Dolean, V., Aghamiry, H. S., Operto, S., and Riffo, S. 3d finite-difference and finite-element frequency-domain wave simulation with multilevel optimized additive schwarz domain-decomposition preconditioner: A tool for full-waveform inversion of sparse node data sets. *Geophysics* **87**(5), T381–T402 (2022).

Unmapped Tent Pitching Schemes by Waveform Relaxation

Gabriele Ciaramella, Martin J. Gander, and Ilario Mazzieri

1 Introduction

The *mapped tent pitching* algorithm (MTP) is a very advanced domain decomposition strategy for the parallel solution of hyperbolic problems. MTP was introduced in [4] and computes the solution by iteratively constructing new polygonal space-time subdomains, called *tents*, in a way that the hyperbolic problem can be solved exactly within them. Due to the polygonal space-time structure of the subdomains, the numerical solution is obtained by a process that maps the tents into space-time cylinders (rectangles for 1D spatial problems), computes the solution in the transformed subdomains, and maps it back into the original tents. Due to the tent mapping leading to singularities, special time integrators are needed to mitigate order reduction.

To avoid this, we introduce a new, *unmapped tent pitching* algorithm (UTP), based on a conceptual idea from Nievergelt in 1964 [5]: "In numerical analysis, one has always tried to speed up computation by reducing the amount of work to be done, not by performing redundant computations." Introducing redundant computations, we eliminate the mapping process from the MTP with a Schwarz waveform relaxation method (SWR). We present our new UTP for the model problem

$$\begin{aligned} \partial_{tt} u(x,t) &= c^2 \partial_{xx} u(x,t) \quad \text{for } (x,t) \in \Omega \times (0,T), \\ u(x,0) &= g_0(x) \text{ and } \partial_t u(x,0) = g_1(x) \quad \text{for } x \in \Omega, \\ u(0,t) &= u(1,t) = 0 \quad \text{for } t \in [0,T], \end{aligned} \tag{1}$$

where $\Omega = (0,1)$, $T > 0$, and g_0 and g_1 are sufficiently regular functions. We first explain in Section 2 the classical MTP process for the solution of (1) and characterize

G. Ciaramella, I. Mazzieri
MOX, Politecnico di Milano, Italy, e-mail: gabriele.ciaramella@polimi.it, ilario.mazzieri@polimi.it

M.J. Gander
Section de Mathématiques, Université de Génève, Switzerland, e-mail: martin.gander@unige.ch

the corresponding advancing front in case of a uniform space decomposition. Then, in Section 3, we introduce a red-black Schwarz waveform relaxation method (RBSWR) and prove a particular relation between RBSWR and MTP. This relation leads us very naturally to introduce our UTP in Section 4.

2 The mapped tent-pitching algorithm (MTP)

To describe the MTP algorithm introduced in [4] for the solution of (1), consider a set $\Omega_0 = \{x_j\}_{j=0}^N \subset \Omega$ of nodes $0 = x_0 < x_1 < \cdots < x_N = 1$. The core of MTP is the strategy used to *pitch tents* at the nodes and define the *advancing front* of the computed exact solution. In our one-dimensional setting, a tent is a hat-function ϕ_j with value 1 at the node x_j and zero at the remaining nodes of Ω_0. The advancing front is the curve representing a portion of the boundary of the space-time subdomain of $\Omega \times [0,T]$ in which the exact solution has been computed by MPT at a certain iteration. More precisely, the advancing front (at iteration $k \in \mathbb{N}$) is a continuous functions $\tau_k^{\mathrm{MTP}} \colon \Omega \to \mathbb{R}$, which is linear in the subintervals (x_j, x_{j+1}). The MTP iteration is initialized with $\tau_0^{\mathrm{MTP}} \equiv 0$ and at the k-th iteration a new advancing front τ_k^{MTP} is computed from $\tau_{k-1}^{\mathrm{MTP}}$ with the property that $\tau_k^{\mathrm{MTP}}(x) \geq \tau_{k-1}^{\mathrm{MTP}}(x)$ for all $x \in \Omega$. The process terminates when an iteration $k = K > 0$ is reached with $\tau_K^{\mathrm{MTP}} \equiv T$. To obtain τ_k^{MTP} one needs to *pitch a new tent* on the front $\tau_{k-1}^{\mathrm{MTP}}$, that means to select an appropriate node x_j in Ω_0 and a $v_j^k > 0$, and update the advancing front as

$$\tau_k^{\mathrm{MTP}}(x) := \tau_{k-1}^{\mathrm{MTP}}(x) + v_j^k \phi_j(x). \tag{2}$$

The node x_j and the value v_j^k are computed to ensure that $|(\tau_k^{\mathrm{MTP}})'(x)| \leq \frac{1}{c}$ for all $x \in \Omega \setminus \Omega_0$. This is a CFL condition [2] and since τ_k^{MTP} is piecewise linear, it is equivalent to[1]

$$\frac{|\tau_k^{\mathrm{MTP}}(x_\ell) - \tau_k^{\mathrm{MTP}}(x_{\widetilde{\ell}})|}{|x_\ell - x_{\widetilde{\ell}}|} \leq \frac{1}{c} \quad \text{for all } \ell = 0, \ldots, N \text{ and } \widetilde{\ell} \in \mathcal{N}_\ell, \tag{3}$$

where $\mathcal{N}_\ell$ denotes the set of indices of the neighboring nodes to x_ℓ. Now, since ϕ_j is zero on $\Omega_0 \setminus \{x_j\}$, one has that $\tau_k^{\mathrm{MTP}}(x_\ell) = \tau_{k-1}^{\mathrm{MTP}}(x_\ell)$ for all $x_\ell \in \Omega_0 \setminus \{x_j\}$. Thus, given a $\tau_{k-1}^{\mathrm{MTP}}$ satisfying (3), the new tent must be pitched in a way that τ_k^{MTP} satisfies (3) as well, that is

$$\frac{|\tau_{k-1}^{\mathrm{MTP}}(x_j) + v_j^k \phi_j(x_j) - \tau_{k-1}^{\mathrm{MTP}}(x_{\widetilde{\ell}})|}{|x_j - x_{\widetilde{\ell}}|} \leq \frac{1}{c} \quad \text{for all } \widetilde{\ell} \in \mathcal{N}_j. \tag{4}$$

[1] In [4], condition (3) appears with an additional constant depending on the shape regularity of the decomposition. This constant is 1 in our one-dimensional framework.

Algorithm 1 Mapped Tent Pitching (sequential)

Require: A decomposition Ω_0.
1: Set $k = 0$ and initialize $\tau_k^{\mathrm{MTP}} \equiv 0$.
2: **while** $\tau_k^{\mathrm{MTP}} \not\equiv T$ **do**
3: Compute the set J_k.
4: Select an index $j \in J_k$, the corresponding node x_j and set $v_j^k = w_j^k$.
5: Update the advancing front: $\tau_k^{\mathrm{MTP}}(x) := \tau_{k-1}^{\mathrm{MTP}}(x) + v_j^k \phi_j(x)$.
6: Solve (1) in the domain between τ_k^{MTP} and $\tau_{k-1}^{\mathrm{MTP}}$ (below the new tent).
7: Update $k = k + 1$.
8: **end while**

Since $v_j^k \phi_j(x) \geq 0$, $\tau_{k-1}^{\mathrm{MTP}}$ satisfies (3), and $\phi_j(x_j) = 1$, (4) becomes $v_j^k \leq \min_{\widetilde{\ell} \in \mathcal{N}_j} \left(\frac{|x_j - x_{\widetilde{\ell}}|}{c} + \tau_{k-1}^{\mathrm{MTP}}(x_{\widetilde{\ell}}) - \tau_{k-1}^{\mathrm{MTP}}(x_j) \right)$. To satisfy this condition and maximize the advancement of the front, we define

$$w_\ell^k := \min\left(T - \tau_{k-1}^{\mathrm{MTP}}(x_\ell), \min_{\widetilde{\ell} \in \mathcal{N}_\ell} \left(\frac{|x_\ell - x_{\widetilde{\ell}}|}{c} + \tau_{k-1}^{\mathrm{MTP}}(x_{\widetilde{\ell}}) - \tau_{k-1}^{\mathrm{MTP}}(x_\ell) \right) \right) \tag{5}$$

for $\ell = 1, \ldots, N$, and the set of admissible values v_j^k as $J_k := \{\ell \in \{1, \ldots, N\} : w_\ell^k > 0\}$. Thus, at the k-th iteration MTP selects any node x_j with $j \in J_k$ and pitches a tent of height $v_j^k = w_j^k$. Once a new tent is pitched, MTP solves the problem within this new tent by a mapping process that transforms the tent into a cylinder (a rectangle in this one-dimensional setting). The overall MTP procedure is given in Algorithm 1. This is the sequential version of MTP. A parallel version can be easily obtained by pitching multiple tents at each iteration, namely by modifying Step 4 and Step 5:

4: Select a set $\mathcal{S}_k \subset J_k$ of all indices $j \in \mathcal{S}_k$ such that the corresponding nodes are not neighbors. Pick all nodes x_j with $j \in \mathcal{S}_k$ and set $v_j^k = w_j^k$.
5: Update the advancing front: $\tau_k^{\mathrm{MTP}}(x) := \tau_{k-1}^{\mathrm{MTP}}(x) + \sum_{j \in \mathcal{S}_k} v_j^k \phi_j(x)$.

We illustrate the parallel MTP procedure with an example using a space decomposition of 7 points $(x_j, j = 0, \ldots, 6)$, see Fig. 1, top left. The MTP is initialized with $\tau_0^{\mathrm{MTP}} \equiv 0$. For $k = 1$ all nodes can be potentially selected, that is $J_1 = \{0, \ldots, 6\}$, but not all of them can be simultaneously selected. Thus, we assume that the nodes x_1, x_3 and x_5 are selected and three tents are pitched on τ_0^{MTP}. The new resulting front is τ_1^{MTP}, which is represented by the red line in Fig. 1, top left. Notice that the slopes of τ_1^{MTP} are lower or equal to the slopes of the characteristic curves, because of condition (5) and the fact that the decomposition considered is nonuniform[2]. Once τ_1^{MTP} is obtained, the set of admissible nodes is $J_2 = \{0, 2, 4, 6\}$. These can be all selected and give rise to the hat-functions (multiplied by the corresponding values v_j^k) represented by the blue dashed lines in Fig. 1, top right. The new front τ_2^{MTP} (blue line in Fig. 1, top right) is then obtained by summing all these functions to τ_1^{MTP}. Repeating this process at iterations 3 and 4 leads to the fronts τ_3^{MTP} (magenta line in

[2] For uniform decompositions, tents are always pitched along characteristic lines.

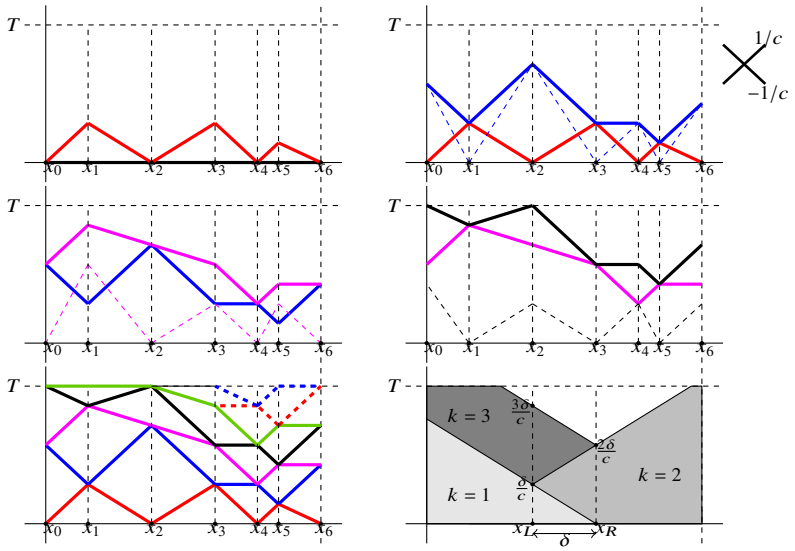


Fig. 1 **Top row, left**: MTP iteration 1: τ_1^{MTP} (red) and τ_0^{MTP} (black). The tents $v_j^1\phi_j$ coincide with the red lines. The cross on the top right gives the slopes of the characteristic lines. **Top row, right**: MTP iteration 2: τ_2^{MTP} (blue), τ_1^{MTP} (red), new tents $v_j^2\phi_j$ (blue dashed). **Middle row, left**: MTP iteration 3: τ_3^{MTP} (magenta), τ_2^{MTP} (blue), new tents $v_j^3\phi_j$ (magenta dashed). **Middle row, right**: MTP iteration 4: τ_4^{MTP} (black), τ_3^{MTP} (magenta), new tents $v_j^4\phi_j$ (black dashed). **Bottom row, left**: Full decomposition constructed by MTP. **Bottom row, right**: First three iterations of SWR. The gray areas are the regions where the exact solution is computed.

Fig. 1, middle left) and τ_4^{MTP} (black line in Fig. 1, middle right). At convergence, we obtain the decomposition shown in Fig. 1, bottom left, which is not uniform since the initial space decomposition Ω_0 is not uniform. It is finer (in time) where the space decomposition is finer, and the front advances more slowly there. Note also that at each iteration the MTP process solves the problem below characteristics, and the conditions used to pitch new tents are satisfied when exact data is available on the lower boundary of the new tent and can be propagated into it. We now characterize the behavior of the advancing front for a uniform decomposition.

Lemma 1 (MTP advancing front for uniform decompositions)

Let the decomposition Ω_0 be uniform with $h := x_j - x_{j-1}$ for $j = 1, \dots, N$. Consider any interior subinterval $\mathcal{I} = [x_L, x_R]$, with $R \in \mathbb{N}$ even and $L = R - 1$. Assume that the (parallel) MTP selects alternatingly odd and even nodes of Ω_0 at odd and even iterates, respectively. Then, starting from $\tau_0^{\text{MTP}} \equiv 0$, we have that $\tau_1^{\text{MTP}}(x_L) = \frac{h}{c}$ and $\tau_1^{\text{MTP}}(x_R) = 0$, and for any $n > 0$ that

$$\tau_{2n}^{\text{MTP}}(x_L) = (2n-1)h/c \qquad \tau_{2n}^{\text{MTP}}(x_R) = 2nh/c, \tag{6a}$$

$$\tau_{2n+1}^{\text{MTP}}(x_L) = (2n+1)h/c, \qquad \tau_{2n+1}^{\text{MTP}}(x_R) = 2nh/c. \tag{6b}$$

Proof Denote by $\mathcal{N}_\ell$ the set of neighboring nodes of x_ℓ. The proof works by induction and uses (2) and (5). We begin with the base case $n = 0$. Since $\tau_0^{\mathrm{MTP}} \equiv 0$, using (5) we compute $v_L^1 = \frac{h}{c}$. Thus, (2) leads to $\tau_1^{\mathrm{MTP}}(x_L) = \frac{h}{c}$ and $\tau_1^{\mathrm{MTP}}(x_\ell) = 0$ for all $\ell \in \mathcal{N}_L$, and then $\tau_1^{\mathrm{MTP}}(x_R) = 0$. Now, we consider the induction step. Thus, assuming that (6a) and (6b) hold, we use (2) to write $\tau_{2n+2}^{\mathrm{MTP}}(x_\ell) = \tau_{2n+1}^{\mathrm{MTP}}(x_\ell) + v_R^{2n+2}\phi_R(x_\ell)$ for $\ell \in \{R, L\}$. Using (5) with the fact that the decomposition is uniform, we obtain for $\ell \in \mathcal{N}_R$ that

$$v_R^{2n+2} = \frac{h}{c} + \tau_{2n+1}^{\mathrm{MTP}}(x_\ell) - \tau_{2n+1}^{\mathrm{MTP}}(x_R) = \frac{h}{c} + (2n+1)\frac{h}{c} - 2n\frac{h}{c} = 2\frac{h}{c},$$

and thus $\tau_{2n+2}^{\mathrm{MTP}}(x_L) = (2n+1)\frac{h}{c}$ and $\tau_{2n+2}^{\mathrm{MTP}}(x_R) = (2n+2)\frac{h}{c}$. Now, (2) implies that $\tau_{2n+3}^{\mathrm{MTP}}(x_\ell) = \tau_{2n+2}^{\mathrm{MTP}}(x_\ell) + v_L^{2n+2}\phi_L(x_\ell)$ for $\ell \in \{R, L\}$, and (5) allows us to compute $v_L^{2n+2} = 2\frac{h}{c}$. Hence, we get that $\tau_{2n+3}^{\mathrm{MTP}}(x_L) = (2n+3)\frac{h}{c}$ and $\tau_{2n+3}^{\mathrm{MTP}}(x_R) = (2n+2)\frac{h}{c}$, and the claim follows. □

3 Red-black Schwarz waveform relaxation (RBSWR)

Consider a decomposition of Ω into $N-1$ subdomains $I_j = (x_j, x_{j+2})$, $j = 0, \ldots, N-2$, where x_j are the nodes in Ω_0. This is a decomposition with generous overlap. Let $\mathcal{R} = \{0, 2, 4, \ldots\}$ and $\mathcal{B} = \{1, 3, 5, \ldots\}$ be two subsets of $\{0, 1, \ldots, N-2\}$. RBSWR is defined by solving in parallel the subproblems

$$\partial_{tt} u_j^k(x,t) = c^2 \partial_{xx} u_j^k(x,t) \qquad \text{in } I_j \times (0,T), \tag{7}$$

$$u_j^k(x,0) = g_0(x) \text{ and } \partial_t u_j^k(x,0) = g_1(x) \qquad \text{for } x \in I_j, \tag{8}$$

$$u_j^k(x_j,t) = u_{j-1}^{k-1}(x_j,t) \qquad \text{for } t \in [0,T], \tag{9}$$

$$u_j^k(x_{j+2},t) = u_{j+1}^{k-1}(x_{j+2},t) \qquad \text{for } t \in [0,T], \tag{10}$$

where k is the iteration index, and $j \in \mathcal{R}$ for k odd and $j \in \mathcal{B}$ for k even. Moreover, the exterior boundary conditions have to be appropriately replaced for $j = 0$ at x_0 and for $j = N-2$ at x_{N-1}. Now, we assume that the decomposition Ω_0 is uniform and denote the overlap by $\delta = x_j - x_{j-1}$. Convergence of (7) was proved in [3, Theorem 1], where it is shown that the exact solution is obtained for $k \geq \frac{Tc}{\delta}$. The convergence behavior depends on the propagation of the exact solution in the overlap; see [3, Figure 1] and Fig. 1, bottom right. In particular, it is possible to show that at odd iterations $k = 2n+1$, $n = 0, 1, 2, \ldots$, the exact solution is computed in the overlap below the characteristic curve intersecting the interface $\{x_L\} \times (0,T)$ at $(2n+1)\frac{\delta}{c}$, cf. Fig. 1, bottom right. Similarly, at even iterations $k = 2n$, $n = 1, 2, \ldots$, the exact solution is computed in the overlap below the characteristic curve intersecting the interface $\{x_R\} \times (0,T)$ at $2n\frac{\delta}{c}$, cf. Fig. 1, bottom right. Thus, we can define a *RBSWR advancing front*, denoted by $\tau_k^{\mathrm{RBSWR}}(x)$, as the function lying on the characteristic curves and such that below its graph the method has already computed the exact solution, independently of the initial guess u^0. An example of the first 4 iterations of

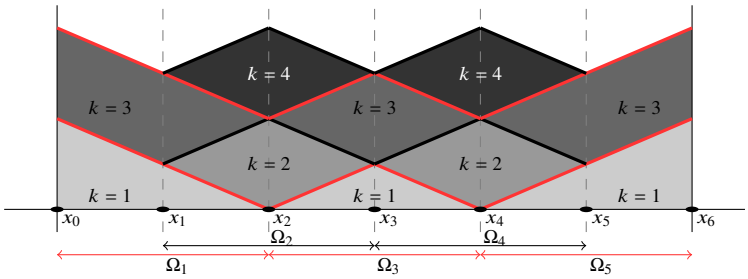


Fig. 2 First four iterations of the red-black SWR for a 5-subdomain case. The gray areas are the regions where the exact solution is computed.

RBSWR is given in Fig. 2. The fronts $\tau_k^{\mathrm{RBSWR}}(x)$ are red and black lines delimiting the gray regions where the exact solution has been computed.

The RBSWR advancing front is characterized in the next lemma, whose proof can be deduced from Fig. 1, bottom right, and Fig. 2.

Lemma 2 (RBSWR advancing front)

Assume that the decomposition Ω_0 is uniform with $h = x_j - x_{j-1}$ for $j = 1, \dots, N$. Consider any interior subinterval $\mathcal{I} = [x_L, x_R]$, with $R \in \mathbb{N}$ even and $L = R - 1$. Consider the RBSWR with overlap $\delta = x_R - x_L$ and initialized with any (sufficiently regular) function u^0 such that $\tau_0^{\mathrm{RBSWR}} \equiv 0$. The advancing front τ_k^{RBSWR} satisfies $\tau_1^{\mathrm{RBSWR}}(x_L) = \frac{\delta}{c}$, $\tau_1^{\mathrm{RBSWR}}(x_R) = 0$, and, for any $n = 1, 2, \dots$, the relations

$$\tau_{2n}^{\mathrm{RBSWR}}(x_L) = (2n-1)\delta/c, \qquad \tau_{2n}^{\mathrm{RBSWR}}(x_R) = 2n\delta/c, \tag{11a}$$

$$\tau_{2n+1}^{\mathrm{RBSWR}}(x_L) = (2n+1)\delta/c, \qquad \tau_{2n+1}^{\mathrm{RBSWR}}(x_R) = 2n\delta/c. \tag{11b}$$

The relation between MTP and RBSWR arises immediately by comparing Lemma 1 and Lemma 2 and it is stated in the following theorem.

Theorem 1 (RBSWR and MPT for uniform decompositions)

Consider a uniform decomposition Ω_0 with $h = x_j - x_{j-1}$ for $j = 1, \dots, N$. Assume that the (parallel) MTP selects alternately odd and even nodes of Ω_0 at odd and even iterates, respectively. Further, notice that the overlap is $\delta = h$. Then, for any initial guess u^0 such that $\tau_0^{\mathrm{RBSWR}} \equiv 0$, the fronts τ_k^{MTP} and τ_k^{RBSWR} coincide in all interior nodes of Ω_0, thus in all interior subintervals.

4 Unmapped tent-pitching

Theorem 1 suggests that the mapping process is not necessary to obtain the exact solution below the tents. This process can be avoided by using SWR on appropriately defined space-time subdomains, even though few redundant computations need to be performed. The key idea is to consider rectangular space-time subdomains having

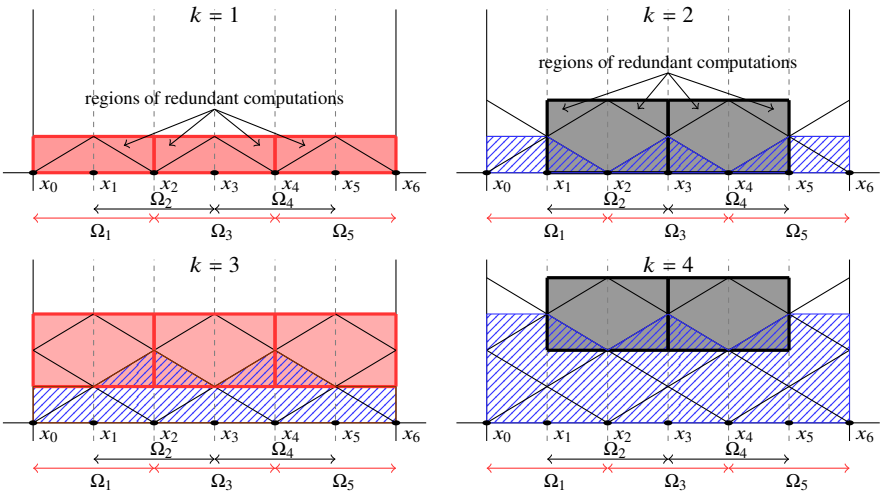


Fig. 3 First four iterations of UTP on a 5 subdomain decomposition. Red and black boxes are the space-time subdomains constructed by UTP at odd and even iterations. The black lines correspond to the tents that MTP constructs. The blue hatched regions are the portions of the domain where UTP computes the exact solution.

the same height of the tents pitched on the space subdomains and width equal to the length of the space subdomains themselves. The space-time subdomains can be considered as rectangular tents, in which the solution can be computed directly, using, e.g., a time-stepping method, without the need of mapping the tent into a rectangular box (the subdomain is already a rectangular tent!). We call this approach the *unmapped tent pitching* (UTP) algorithm, and describe it in detail for a uniform space decomposition Ω_0 and for a parallel MTP selecting alternatingly odd and even nodes. Extensions to nonuniform decompositions and higher dimensions are possible, but beyond the scope of this short manuscript. They will be presented in the future work [1]. The UTP process begins by selecting the odd nodes of Ω_0 and computing the heights v_j^0 of the tents that the MTP would pitch. Instead, rectangular space-time subdomains $\mathcal{T}_j$ are pitched, and one RBSWR iteration is performed restricted on them. This step is shown in Fig. 3, top left, where the three (red) subdomains are represented together with the tents that the parallel MTP would pitch. RBSWR computes the exact solution below the tents, as represented by the blue hatched regions in Fig. 3, top right. However, wrong approximations are computed in the areas above the tents, which correspond to the regions where redundant computations are performed. The second iteration of the UTP is shown in Fig. 3, top right. Here, the new pitched rectangular subdomains are depicted in black. Within them one RBSWR iteration is performed. The exact solution is obtained below the classical MTP tents, while redundant computations are performed above them. As a result, the exact solution is computed in the blue hatched area depicted in Fig. 3, bottom left. By repeating this process iteratively one obtains the subdomains and

Algorithm 2 Unmapped Tent Pitching by RBSWR

Require: A decomposition Ω_0 of N nodes and an initial guess function u^0.

1: Set $k = 1$ and $v_j^0 = 0$ for all $j = 1, \dots, N$.
2: **while** $\exists j \in \{0, 1, \dots, N-1\} \ : \ v_j^{k-1} \neq T$ **do**
3: Set $J_k = \{1, 3, \dots\}$ if k is odd and $J_k = \{2, 4, \dots\}$ if k is even.
4: Use (5) to compute the heights $v_j^k = w_j^k + v_j^{k-1}$ for all $j \in J_k$.
5: For each $j \in J_k$ pitch a rectangular subdomain $\mathcal{T}_j := [x_{j-1}, x_{j+1}] \times [v_j^{k-1}, v_j^k]$.
6: Solve (7) to get u_j^{k+1} in $\mathcal{T}_j$ for all $j \in J_k$, and extend them by u^0 above $\mathcal{T}_j$.
7: Update $k = k + 1$.
8: **end while**

the exact solution areas shown in Fig. 3 for $k = 3$ and $k = 4$. The overall UTP Algorithm 2 terminates when the exact solution is computed in the entire space-time domain.

To conclude, our new unmapped tent pitching algorithm computes to the mapped tent pitching algorithm equivalent approximations, using redundant computations. It is however cheaper, since it does not have to compute the tent mappings, and the volume of the redundant computations is also present in the tents after the mapping. Its implementation is also straightforward, and one can use standard time integrators, since there is no danger of order reduction without the tent mapping.

References

1. Ciaramella, G., Gander, M., and Mazzieri, I. Space-time RAS methods for wave-propagation problems. *in preparation* (2023).
2. Courant, R., Friedrichs, K., and Lewy, H. On the partial difference equations of mathematical physics. *IBM J. Res. Dev.* **11**(2), 215–234 (1967).
3. Gander, M. J., Halpern, L., and Nataf, F. Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation. *Proceedings in Domain Decomposition Methods in Science and Engineering XIX* (1999).
4. Gopalakrishnan, J., Schöberl, J., and Wintersteiger, C. Mapped tent pitching schemes for hyperbolic systems. *SIAM J. Sci. Comput.* **39**(6), B1043–B1063 (2017).
5. Nievergelt, J. Parallel methods for integrating ordinary differential equations. *Commun. ACM* **7**(12), 731–733 (1964).

A 2-Level Domain Decomposition Preconditioner for KKT Systems with Heat-Equation Constraints

Eric C. Cyr

1 Introduction

This paper develops a new domain-decomposition method for solving the KKT system with heat-equation constraints. This effort is driven by the quadratic optimization problem of the form

$$\begin{aligned} \min_{z} \quad & \frac{1}{2}\int_0^T \|u-\tilde{u}\|^2_{L^2(\Omega)}\,dt + \frac{\omega}{2}\int_0^T \|z\|^2_{L^2(\Omega)}\,dt \\ \text{s.t.} \quad & \partial_t u - \nu\nabla\cdot\nabla u = z, && x\in\Omega\subset\mathbb{R}^2, t\in[0,T] \\ & u(x,t)=0, \quad x\in\partial\Omega, t\in[0,T], && u(x,0)=u_0(x), \quad x\in\Omega \end{aligned} \tag{1}$$

This quadratic PDE-constrained optimization problem finds a control z such that the solution u to the heat equation matches the target $\tilde{u}$. The spatial domain is Ω, the time interval is $[0,T]$, and the heat conductivity is ν. Uniform homogenous boundary conditions are assumed for all time, and the initial condition is prescribed by u_0.

Many nonlinear methods use a series of quadratic approximations of the form represented by Eq. 1 to solve PDE-constrained optimization problems (see for instance sequential quadratic programming methods [8, 15, 27]). There have been several studies focused on developing scalable preconditioners for the saddle-point system that arises from the first-order necessary conditions. Often preconditioners for saddle-point systems take the form of approximate factorization block preconditioners [3]. These were explored for KKT systems in [4, 5]. Our work relies heavily on the block preconditioners from the Wathen group [23, 24, 25].

This effort focuses on transient PDE constraints where the size of the system scales with the number of spatial unknowns times the number of time steps, resulting in substantial computational effort. To alleviate this, a number of efforts have proposed accelerating the time solve using adaptive space-time discretiza-

Eric C. Cyr
Sandia National Laboratories, Albquerque, NM, 87123, e-mail: eccyr@sandia.gov

tions [16, 17], parareal [12, 20, 27], multigrid approaches [6, 7, 10, 13, 19], block preconditioning [23, 24], and domain decomposition methods [14].

Our approach is also built on block preconditioning ideas. A difference is that our technique exploits an observation that the Schur-complement of the KKT system is elliptic in time (see [11, 18]). This allows us to leverage existing two level domain decomposition approaches for elliptic systems to improve the parallel scalability of the block preconditioner. Good performance is achieved by algorithmic choices that ensure the forward and backward in time integrators can be applied on the fine level.

2 Discrete system and block preconditioner

In this article the PDE in Eq. 1 will be discretized on a 2D Cartesian grid using first order backward Euler in time, and a second order finite difference stencil in space. A row of the discrete space-time system for the heat equation satisfies:

$$u_{i,j}^{n+1} - u_{i,j}^{n} + \Delta t \nu \left(\frac{-u_{(i+1)j}^{n+1} + 2u_{ij}^{n+1} - u_{(i-1)j}^{n+1}}{\Delta x^2} + \frac{-u_{i(j+1)}^{n+1} + 2u_{ij}^{n+1} - u_{i(j-1)}^{n+1}}{\Delta y^2} \right) = \Delta t z_{ij}^{n+1}. \tag{2}$$

Here i, j are the interior space indices defined over $1 \ldots n_x - 1$ and $1 \ldots n_y - 1$. The exterior indices are eliminated using the homogenous boundary conditions. The superscript time index n runs from $0 \ldots N_t$. Each n is referred to below as a *time-node*. The control variable z's index matches the implicit index on u, therefore z^{n+1} is associated with the nth time interval. For a single time interval, Eq. 2 rewritten in matrix form is

$$J_{(n+1)(n+1)} u^{n+1} + J_{(n+1)n} u_n + L_{(n+1)(n+1)} z^{n+1} = 0, \tag{3}$$

and the global space-time system is

$$Ju + Lz = f. \tag{4}$$

The right hand side f includes contributions from the initial conditions. The matrix J is block lower triangular and the matrix L is block diagonal.

The linear system whose solution solves the quadratic optimization problem from Eq. 1 is the celebrated KKT system $K\mathbf{u} = \mathbf{f}$ where

$$K = \begin{bmatrix} M_u & & J^T \\ & \omega M_z & L^T \\ J & L & \end{bmatrix}, \ \mathbf{u} = \begin{bmatrix} u \\ z \\ w \end{bmatrix}, \ \mathbf{f} = \begin{bmatrix} f_u \\ f_z \\ f \end{bmatrix}. \tag{5}$$

The final row is the discrete form of the PDE constraint, enforced by the Lagrange multiplier w. We will also refer to w as the adjoint solution. M_u and M_z are identity matrices scaled by $\Delta t \Delta x \Delta y$. The matrix K is a saddle point matrix, whose structure is frequently observed in numerical optimization. Many effective block preconditioners

have been developed for this class of matrix [2, 3, 4]. We focus on the block preconditioning approach developed by Wathen and collaborators for solving linearized PDE-constrained optimization problems [23, 24, 25].

We write a block LDU factorization of the matrix K

$$K = \begin{bmatrix} I & & \\ & I & \\ JM_u^{-1} & \omega^{-1}LM_z^{-1} & I \end{bmatrix} \begin{bmatrix} M_u & & \\ & \omega M_z & \\ & & -S \end{bmatrix} \begin{bmatrix} I & & M_u^{-1}J^T \\ & I & \omega^{-1}M_z^{-1}L^T \\ & & I \end{bmatrix} \tag{6}$$

where the Schur-complement is $S = JM_u^{-1}J^T + \frac{1}{\omega}LM_z^{-1}L^T$. Following [23], K is preconditioned using the block diagonal operator

$$P = \begin{bmatrix} M_u & & \\ & \omega M_z & \\ & & \hat{S} \end{bmatrix}, \text{ where } \hat{S} = \hat{J}M_u^{-1}\hat{J}^T, \hat{J} = J + \omega^{-1/2}L. \tag{7}$$

This preconditioner leverages the result in [21], and approximately inverts the block diagonal in the LDU factorization. The matrix $\hat{J}$ used in the approximate Schur complement $\hat{S}$ is block lower triangular (similar to J), a fact that we will exploit below. The choice of $\hat{S}$ integrates the state Jacobian and the effects of the regularization parameter. In [23, 24] and [26], this approximation is developed and shown to lead to robust performance with respect to ω.

3 Two-level domain decomposition Schur-complement

We propose a new domain decomposition approach for approximately inverting $\hat{S}$. This is motivated by the observation that the operator S is elliptic in time (see [18] and [11]). For simplicity, we show this discretely using only the term $JM_u^{-1}J^T$. Consider the ODE $\partial_t y = -y$ discretized over three time steps with forward Euler: $y^{n+1} - y^n + \Delta t y^n = 0$. With $M_u = I$, the Schur-complement $\hat{S}$ is

$$\begin{bmatrix} 1 & & \\ -1+\Delta t & 1 & \\ & -1+\Delta t & 1 \end{bmatrix} \begin{bmatrix} 1 & -1+\Delta t & \\ & 1 & -1+\Delta t \\ & & 1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & -(1-\Delta t) & \\ -(1-\Delta t) & 2(1-\Delta t)+\Delta t^2 & -(1-\Delta t) \\ & -(1-\Delta t) & 2(1-\Delta t)+\Delta t^2 \end{bmatrix}. \tag{8}$$

Examining the second row it is clear the operator has a 1D Laplacian stencil in time, with a positive perturbation on the diagonal. To take advantage of this ellipticity, we will apply existing domain decomposition approaches to the $\hat{S}$ operator. This *ellipticity principle* enables scalable performance of a preconditioned Krylov method. We also impose an *efficiency constraint* that the computational kernels in our preconditioner use the time integration method for the state and adjoint unknowns.

The *ellipticity principle* is realized by considering a restricted additive Schwarz (RAS) method with N_D subdomains (see [9]). Each subdomain contains the spatial unknowns associated with a subset of time steps. For instance, if there are $N_t = 9$ time steps, then for $N_D = 3$ the subdomains contain time-nodes $\{1, 2, 3\}$, $\{3, 4, 5, 6\}$, and $\{6, 7, 8, 9\}$ (the 0^{th} time-node is the excluded initial condition). Notice that the time-nodes are overlapped but the time steps are not. With these subsets, boolean operators R_s are defined that restrict a space-time vector to the time-nodes in a subdomain, giving the RAS preconditioner

$$\hat{S}^{-1}_{\text{RAS}} = \sum_{s=1}^{N_D} R_s^T D_s \left(R_s \hat{J} M_u^{-1} \hat{J}^T R_s^T \right)^{-1} R_s \tag{9}$$

where D_s is the (boolean) partition of unity matrix (Defn. 1.11 of [9]). RAS is known to lead to effective preconditioners for elliptic problems and can be extended to a multi-level schemes. While the *ellipticity principle* is exploited in $\hat{S}^{-1}_{\text{RAS}}$, the explicit formation of the product $\hat{J} M_u^{-1} \hat{J}^T$ does not satisfy the *efficiency constraint*.

To satisfy the *efficiency constraint* note that the range of $\hat{J}^T R_s^T$ is nonzero on time-nodes in the s^{th} subdomain and one time-node earlier. For instance, if the subdomain contains nodes $\{3, 4, 5, 6\}$ then the range is nonzero on $\{2, 3, 4, 5, 6\}$. Let Q_s be a new extended restriction operator whose action produces a space-time vector for time-nodes that are nonzero in the range of $\hat{J}^T R_s^T$. Further, choose Q_s so that

$$Q_s = \begin{bmatrix} W_s \\ R_s \end{bmatrix} \text{ and } Q_s Q_s^T = I. \tag{10}$$

The operator W_s restricts the space-time vector to the time-nodes contained in the earlier time step relative to the s^{th} subdomain. Because Q_s is a restriction operator that represents the nonzero range of $\hat{J}^T R_s^T$, we have that $Q_s^T Q_s \hat{J}^T R_s^T = \hat{J}^T R_s^T$. Recalling that M_u is diagonal, we can rewrite the subdomain solve in $\hat{S}^{-1}_{RAS}$ as

$$R_s \hat{J} M_u^{-1} \hat{J}^T R_s^T = R_s \hat{J} Q_s^T (Q_s M_u^{-1} Q_s^T) Q_s \hat{J}^T R_s^T \tag{11}$$

Using the constraint in Eq. 10, we have the additional identities $R_s = R_s Q_s^T Q_s$ and $R_s^T = Q_s^T Q_s R_s^T$. This permits a final rewrite of the operator in Eq. 11

$$R_s \hat{J} M_u^{-1} \hat{J}^T R_s^T = R_s Q_s^T \hat{J}_s M_s^{-1} \hat{J}_s^T Q_s R_s^T \tag{12}$$

where $\hat{J}_s = Q_s \hat{J} Q_s^T$ and $M_s^{-1} = Q_s M_u^{-1} Q_s^T$. The inverse action of $\hat{J}_s M_s^{-1} \hat{J}_s^T$ is easily computed in a matrix free way on the time-nodes in the extended subdomain. Motivated by this equivalence, we define a new one-level preconditioner

$$\hat{S}^{-1}_{\text{RASQ}} = \sum_{s=1}^{N_D} R_s^T D_s \left(R_s Q_s^T \hat{J}_s^{-T} M_s \hat{J}_s^{-1} Q_s R_s^T \right) R_s. \tag{13}$$

The term in parentheses is different from the term inverted in Eq. 9. The difference is that the inverse computed in Eq. 9 is constrained to have a zero initial condition

outside of the subdomain. This revised operator satisfies our *efficiency constraint* as computing $\hat{J}_s^{-1}$ and $\hat{J}_s^{-T}$ is done using the time integration method.

To obtain scalability with respect to the number of subdomains a coarse grid correction is required. Again leveraging the elliptic nature of the Schur-complement we consider the Nicolaides coarse space developed for solving the Poisson problem [22]. Following the presentation in [9], the Nicolades coarse space is defined as

$$Z = \begin{bmatrix} D_1 R_1 \Phi_1 & & \dots & \\ & D_2 R_2 \Phi_2 & \ddots & \\ \vdots & \ddots & \ddots & \\ & \dots & & D_{N_D} R_{N_D} \Phi_{N_D} \end{bmatrix}, \text{ where } \Phi_s = \begin{bmatrix} 1 & -1 \\ 1 & -1 + \frac{2}{N_s} \\ \vdots & \vdots \\ 1 & -1 + 2\frac{N_s}{N_s} \end{bmatrix} \tag{14}$$

The columns of Φ_s form a constant and linear basis over the N_s subdomain time-nodes. The coarse restriction $R_0 = Z^T$ is used in the definition of the coarse operator $\hat{S}_0 = R_0 \hat{J} M_u^{-1} \hat{J}^T R_0^T$. The coarse solve is applied in a multiplicative way

$$\hat{S}_{2-\text{level}}^{-1} = \hat{S}_{\text{RASQ}}^{-1} R_0^T \hat{S}_0^{-1} R_0. \tag{15}$$

Due to the structure of R_0 the coarse operator $\hat{S}_0$ can be constructed in parallel. This does represent a violation of the *efficiency constraint* to be addressed by future work.

4 Numerical experiments

To demonstrate this approach we discretize the quadratic optimization problem from Eq. 1 as described in Sec. 2. The 2D spatial domain is $\Omega = (0, 1) \times (0, 2) \subset \mathbb{R}^2$, and the time domain is $[0, 1]$. The initial conditions and target solutions are

$$u_0(x, y) = -xy(x-1)(y-2), \quad \tilde{u}(x, y, t) = \sin(2.0\pi t) \sin(2.0\pi x) \sin(2.0\pi y). \tag{16}$$

The regularization parameter ω varies over five orders of magnitude. Experiments were run with 9×9, 17×17, and 33×33 mesh points. Qualitatively, variation with number of spatial points was not a factor in the convergence. This is not surprising as the implicit operator in space is inverted with a direct solve. As a result, the computations below are all for the case of 17×17 mesh points. Recall that homogeneous boundary conditions are removed, giving 15×15 unknowns in each time step. The linear system $K\mathbf{u} = \mathbf{f}$ (Eq. 5) is solved using right preconditioned GMRES from PyAMG [1] iterated until a relative residual tolerance of 10^{-6} is achieved.

Figure 1 presents three weak-scaling studies ranging from 100 to 3200 time steps. The number of time steps per subdomain is fixed at 80 in the left plot, 20 in the center plot, and 5 in the right plot. For the case of 80 steps, the fewest number of time steps is 200 (the minimum number of time steps evenly divisible by 80 in the chosen sequence). The plots, show the number of iterations as a function of time

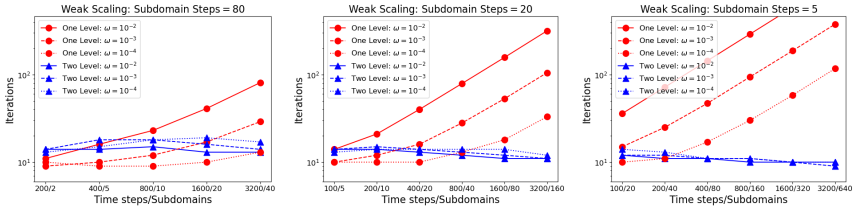


Fig. 1 Three weak scaling studies for different numbers of time-steps per subdomain. The two level scheme (triangles) has flat iteration counts for regardless of the number of time steps, the subdomain size, and the regularization parameter. Asymptotically the one level method shows a strong dependence with respect to the number of subdomains and time steps.

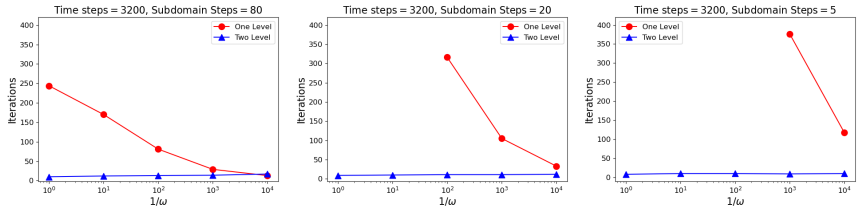


Fig. 2 This plot demonstrates the robustness of the two level scheme (triangle markers) with respect to the regularization parameter ω. Note that for many cases the one level scheme (circle markers) did not converge in the 420 iterations (the maximum allowed), thus those values are omitted.

step count for GMRES preconditioned using P from Eqn 7 with Schur complement approximations $\hat{S}^{-1}_{\text{RASQ}}$ for the one level case (circle markers), and $\hat{S}^{-1}_{2-\text{level}}$ for the two level case (triangle markers). Different values for the regularization parameter ω are indicated using solid (10^{-2}), dashed (10^{-3}) or dotted (10^{-4}) lines. These plots demonstrate that the performance of the two level method is independent of both the number of subdomains, and the number of time steps. Further, independence holds regardless of the value of the regularization parameter. As anticipated the one level method has substantial growth with the number of time steps, and variability with the regularization parameter. However, it is worth noting that dependent on the number of subdomains and the size of the regularization parameter the one level method may be faster despite its lack of scalability. For instance, when using 40 subdomains and a regularization parameter of 10^{-4} the one-level method takes the same number of iterations but lacks the synchronization and added cost of the two level method.

The scaling with respect to the regularization parameter is investigated in Figure 2. In these plots the preconditioned iteration counts are plotted as a function of the inverse regularization parameter. Data points are excluded when the number of iterations exceeded the maximum iteration count for GMRES (in this case 420). Here again the two level method scales well, yielding essentially flat iteration counts as a function of the regularization parameter. The one level method shows strong dependence on ω, though it improves dramatically for smaller values.

5 Conclusion

In this paper, motivated by results in block preconditioning and the elliptic-in-time nature of the KKT system, we develop a two level domain decomposition preconditioner that facilitates a parallel-in-time solver for the discrete optimality system constrained by the heat equation. While limited in their breadth, initial results for this approach show excellent scalability with respect to the number of time steps, subdomains, and the regularization parameter. Future work will focus on achieving improved scaling by including more levels in the hierarchy, and applying this technique to a broader class of problems and discretizations.

Acknowledgements The author is indebted anonymous reviewers whose comments resulted in a significant strengthening of the paper. The author also acknowledges support from the SEA-CROGS project and the Early Career program, both funded by the DOE Office of Science. This article has been authored by an employee of National Technology & Engineering Solutions of Sandia, LLC under Contract No. DE-NA0003525 with the U.S. Department of Energy (DOE). The employee owns all right, title and interest in and to the article and is solely responsible for its contents. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this article or allow others to do so, for United States Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan https://www.energy.gov/downloads/doe-public-access-plan. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

References

1. Bell, N., Olson, L. N., and Schroder, J. PyAMG: Algebraic multigrid solvers in python. *Journal of Open Source Software* **7**(72), 4142 (2022).
2. Benzi, M. and Golub, G. H. A preconditioner for generalized saddle point problems. *SIAM Journal on Matrix Analysis and Applications* **26**(1), 20–41 (2004).
3. Benzi, M., Golub, G. H., and Liesen, J. Numerical solution of saddle point problems. *Acta numerica* **14**, 1–137 (2005).
4. Benzi, M., Haber, E., and Taralli, L. A preconditioning technique for a class of PDE-constrained optimization problems. *Advances in Computational Mathematics* **35**(2), 149–173 (2011).
5. Biros, G. and Ghattas, O. Parallel Lagrange–Newton–Krylov–Schur methods for PDE-constrained optimization. Part I: The Krylov–Schur solver. *SIAM Journal on Scientific Computing* **27**(2), 687–713 (2005).
6. Borzı, A. Multigrid methods for parabolic distributed optimal control problems. *Journal of Computational and Applied Mathematics* **157**(2), 365–382 (2003).
7. Borzi, A. and Schulz, V. Multigrid methods for PDE optimization. *SIAM review* **51**(2), 361–395 (2009).
8. Byrd, R. H., Curtis, F. E., and Nocedal, J. An inexact SQP method for equality constrained optimization. *SIAM Journal on Optimization* **19**(1), 351–369 (2008).
9. Dolean, V., Jolivet, P., and Nataf, F. *An Introduction to Domain Decomposition Methods*. Society for Industrial and Applied Mathematics, Philadelphia, PA (2015). https://epubs.siam.org/doi/pdf/10.1137/1.9781611974065.

10. Falgout, R. D., Maldague, J.-M., Meyers, I., Munar, J., Neville, E., and Overman, T. TriMGRIT: An Extension of Multigrid Reduction in Time for Constrained Optimization. In: *PinT 2020 – 9th Parallel-in-Time Workshop* (2020).
11. Gander, M. J. and Kwok, F. Schwarz methods for the time-parallel solution of parabolic control problems. In: *Domain decomposition methods in science and engineering XXII*, 207–216. Springer (2016).
12. Gander, M. J., Kwok, F., and Salomon, J. PARAOPT: A parareal algorithm for optimality systems. *SIAM Journal on Scientific Computing* **42**(5), A2773–A2802 (2020).
13. Günther, S., Gauger, N. R., and Schroder, J. B. A non-intrusive parallel-in-time approach for simultaneous optimization with unsteady PDEs. *Optimization Methods and Software* **34**(6), 1306–1321 (2019).
14. Heinkenschloss, M. A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems. *Journal of Computational and Applied Mathematics* **173**(1), 169–198 (2005).
15. Heinkenschloss, M. and Ridzal, D. A matrix-free trust-region SQP method for equality constrained optimization. *SIAM Journal on Optimization* **24**(3), 1507–1541 (2014).
16. Langer, U., Steinbach, O., Tröltzsch, F., and Yang, H. Space-time finite element discretization of parabolic optimal control problems with energy regularization. *SIAM Journal on Numerical Analysis* **59**(2), 675–695 (2021).
17. Langer, U., Steinbach, O., Tröltzsch, F., and Yang, H. Unstructured space-time finite element methods for optimal control of parabolic equations. *SIAM Journal on Scientific Computing* **43**(2), A744–A771 (2021).
18. Lewis, R. M. and Nash, S. G. Model problems for the multigrid optimization of systems governed by differential equations. *SIAM Journal on Scientific Computing* **26**(6), 1811–1837 (2005).
19. Lin, S. *Multilevel-in-Time Methods for Optimal Control of PDEs and Training of Recurrent Neural Networks*. Ph.D. thesis, Rice University (2022).
20. Maday, Y. and Turinici, G. A parareal in time procedure for the control of partial differential equations. *Comptes Rendus Mathematique* **335**(4), 387–392 (2002).
21. Murphy, M. F., Golub, G. H., and Wathen, A. J. A note on preconditioning for indefinite linear systems. *SIAM Journal on Scientific Computing* **21**(6), 1969–1972 (2000).
22. Nicolaides, R. A. Deflation of Conjugate Gradients with Applications to Boundary Value Problems. *SIAM Journal on Numerical Analysis* **24**(2), 355–365 (1987).
23. Pearson, J. W., Stoll, M., and Wathen, A. J. Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems. *SIAM Journal on Matrix Analysis and Applications* **33**(4), 1126–1152 (2012).
24. Pearson, J. W., Stoll, M., and Wathen, A. J. Preconditioners for state-constrained optimal control problems with moreau–yosida penalty function. *Numerical Linear Algebra with Applications* **21**(1), 81–97 (2014).
25. Rees, T., Dollar, H. S., and Wathen, A. J. Optimal solvers for PDE-constrained optimization. *SIAM Journal on Scientific Computing* **32**(1), 271–298 (2010).
26. Schiela, A. and Ulbrich, S. Operator preconditioning for a class of inequality constrained optimal control problems. *SIAM Journal on Optimization* **24**(1), 435–466 (2014).
27. Ulbrich, S. Generalized SQP methods with "parareal" time-domain decomposition for time-dependent PDE-constrained optimization. In: *Real-time PDE-constrained optimization*, 145–168. SIAM (2007).

Auxiliary Space Preconditioning with a Symmetric Gauss-Seidel Smoothing Scheme for IsoGeometric Discretization of $\mathbf{H_0(curl)}$-elliptic Problem

Abdeladim El Akri, Khalide Jbilou, Nouredine Ouhaddou, and Ahmed Ratnani

1 Introduction

The IsoGeometric Analysis (IgA), introduced by Hughes et al. in [4], is a computational method that provides a general framework for the design and analysis of numerical approximation of partial differential equations (PDEs). The IgA is based on the Galerkin formulation followed by the construction of a finite-dimensional subspace, which approximates the solution space, determined by a finite set of basis functions. These functions are adopted from the geometry description of the PDE domain which usually employs B-spline functions, as done by computer-aided design algorithms [6]. As a consequence, the geometry is maintained exactly and the use of high-regularity functions is settled by simply increasing or decreasing the multiplicities of knots.

The discrete problems produced by isogeometric methods are usually very hard to solve by the standard methods; they are ill-conditioned and the development of a preconditioning strategy is not straightforward, specially in the case of problems characterized by the presence of a large kernel of the PDE operator (like e.g the model problem considered in the present paper). In this case a natural way of constructing the preconditioner is the Auxiliary Space Preconditioning (ASP) method introduced by Xu in [9], see also [3]. The latter is a preconditioning technique based on a simple smoothing scheme (e.g Jacobi or Gauss-Seidel method) and an auxiliary space. The method has the main advantage of linking the solution space directly with functions in the potential space, which makes it possible to control the drawback of the presence of a large null space.

Abdeladim El Akri, Nouredine Ouhaddou, Ahmed Ratnani
Lab. MSDA, Mohammed VI Polytechnic University, Green City, Morocco e-mail: abdeladim.elakri@um6p.ma, nouredine.ouhaddou@um6p.ma, ahmed.ratnani@um6p.ma

Khalide Jbilou
Lab. MSDA, Mohammed VI Polytechnic University, Green City, Morocco and Lab. LMPA, University of Littoral Côte d'Opale, Calais cedex, France e-mail: khalide.jbilou@univ-littoral.fr

Due to page limitation, in the present work we consider only one model problem (even if the results are valid for a variety of $\boldsymbol{H}(\mathbf{curl})$ and $\boldsymbol{H}(\mathrm{div})$ problems)

$$\mathbf{curl}\,\mathbf{curl}\,\boldsymbol{u} + \tau\boldsymbol{u} = \boldsymbol{f} \text{ in } \Omega, \quad \boldsymbol{u} \times \boldsymbol{n} = 0 \text{ on } \partial\Omega, \tag{1}$$

where the vector function $\boldsymbol{f} \in (L^2(\Omega))^3$, τ is a positive constant and $\Omega = (0,1)^3$. We develop a fast preconditioned iterative linear solver for (1). The resulting algorithm relies on a symmetric Gauss-Seidel smoothing scheme, Poisson problem solvers, and a GLT-based smoother to remove the dependence on the degree p . For the former we provide a new algorithm which exploits the block representation of the matrix of the resulting discrete system through sum of Kronecker products. For the isogeometric discretization of the Poisson problems, we adopt the fast diagonalization method developed in [8]. The GLT smoother is taken from [5].

The rest of the paper is organized as follows. Section 2 presents the IgA finite element discretization of the model (1). In Section 3, we propose a new algorithm for the symmetric Gauss-Seidel method that utilizes the block structure of the matrix-based discretization of (1). Next, in Section 4, we introduce the auxiliary space preconditioner, and in Section 5, we combine it with a GLT-based smoother to control the p-dependency of the solver. Finally, in Section 6, we illustrate the performance of our preconditioner with several numerical tests.

2 Isogeometric discretization

For the sake of simplicity, we shall consider only *non-periodic* and *uniform* knot vectors of the form

$$T = (\underbrace{0,\ldots,0}_{p+1}, t_{p+2} < t_{p+3} < \ldots t_{n-1} < t_n, \underbrace{1,\ldots,1}_{p+1}),$$

where t_i is the i-th knot, n is the number of basis functions and p is the polynomial order. B-spline basis functions are defined recursively and they begin with order $p = 0$ such as

$$B_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \le t < t_{i+1}, \\ 0 & \text{otherwise} \end{cases}$$

and for higher order $p \ge 1$ as follows

$$B_{i,p}(t) = \frac{t - t_i}{t_{i+p} - t_i} B_{i,p-1}(t) + \frac{t_{i+p+1} - t}{t_{i+p+1} - t_{i+1}} B_{i+1,p-1}(t),$$

in which a fraction with zero denominator is assumed to be zero. We let

$$\mathcal{S}^p = \operatorname{span}\left\{B_{i,p} \,:\, i = 1,\ldots,n\right\}, \quad \mathcal{S}_0^p = \operatorname{span}\left\{B_{i,p} \,:\, i = 2,\ldots,n-1\right\},$$

the *uni-variate spline* spaces spanned by the B-spline functions. For three-dimensional vector field structures, we specify a tridirectional knot vector $\boldsymbol{T} = T_1 \times T_2 \times T_3$, where each T_i is an open and uniform univariate knot vector related to a B-spline degree p_i. We let then

$$\boldsymbol{V}_{h,0}(\mathbf{curl}) = \left(\mathcal{S}^{p_1-1} \otimes \mathcal{S}_0^{p_2} \otimes \mathcal{S}_0^{p_3}\right) \times \left(\mathcal{S}_0^{p_1} \otimes \mathcal{S}^{p_2-1} \otimes \mathcal{S}_0^{p_3}\right) \times \left(\mathcal{S}_0^{p_1} \otimes \mathcal{S}_0^{p_2} \otimes \mathcal{S}^{p_3-1}\right),$$

the three-dimensional isogeometric approximation of $\boldsymbol{H}_0(\mathbf{curl})$ (see [1]). However, we shall need also the following discrete counterpart of space $H_0^1(\Omega)$

$$V_{h,0}(\mathbf{grad}) = \mathcal{S}_0^{p_1} \otimes \mathcal{S}_0^{p_2} \otimes \mathcal{S}_0^{p_3}.$$

Among the important properties, spaces $V_{h,0}(\mathbf{grad})$ and $\boldsymbol{V}_{h,0}(\mathbf{curl})$ feature quasi interpolation operators $\Pi_{h,0}^{\mathbf{grad}}$ and $\Pi_{h,0}^{\mathbf{curl}}$ (see [1], for instance) that make the (DeRham) diagram

$$\begin{array}{ccc} H_0^1 & \xrightarrow{\mathbf{grad}} & \boldsymbol{H}_0(\mathbf{curl}) \\ \Pi_{h,0}^{\mathbf{grad}} \downarrow & & \Pi_{h,0}^{\mathbf{curl}} \downarrow \\ \boldsymbol{V}_{h,0}(\mathbf{grad}) & \xrightarrow{\mathbf{grad}} & \boldsymbol{V}_{h,0}(\mathbf{curl}) \end{array}$$

commutes and exact.

Our discrete solution $\boldsymbol{u}_h \in \boldsymbol{V}_{h,0}(\mathbf{curl})$ satisfies the weak formulation

$$(\mathbf{curl}\, \boldsymbol{u}_h, \mathbf{curl}\, \boldsymbol{v}_h) + \tau(\boldsymbol{u}_h, \boldsymbol{v}_h) = (\boldsymbol{f}, \boldsymbol{v}_h), \quad \forall \boldsymbol{v}_h \in \boldsymbol{V}_{h,0}(\mathbf{curl}), \tag{2}$$

where $(\cdot,\cdot)$ refers to the $(L^2(\Omega))^3$ inner-product. With the standard basis for $\boldsymbol{V}_{h,0}(\mathbf{curl})$ (see [7]), we can write (2) as a linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, where $\boldsymbol{A}$ is a (symmetric) 3×3 block matrix of the form

$$\boldsymbol{A} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}, \tag{3}$$

where each diagonal block matrix A_{ii} is a sum of Kronecker products of 3 matrices while the non-diagonal matrices A_{ij} $(i \neq j)$ are Kronecker products of 3 matrices. Algorithms presented in the next section exploit this (tensor-product) structure.

3 Block fast Gauss-Seidel method for sum of Kronecker products

In this section we present an efficient implementation of the block Gauss-Seidel method that is specifically designed for solving systems of equations involving a sum of Kronecker product matrices. This implementation is a key contribution of our paper and is used in the Gauss-Seidel smoothing step of the optimal ASP-based algorithm presented in Section 5.

To begin, we recall the symmetric Gauss-Seidel method in Algorithm 1. Our implementation uses the `spsolve` driver, which has different implementations depending on the type of the matrix $\boldsymbol{A}$ (lower or upper triangular matrix). These implementations are given in algorithms 2–3.

Algorithm 1: `Symmetric Gauss Seidel solver`

Input : $\boldsymbol{A}$: A given matrix, $\boldsymbol{b}$: A given vector, $\boldsymbol{x}$: A starting point, ν_1: The number of iterations
Output $\boldsymbol{x}$: The approximate solution of $\boldsymbol{Ax} = \boldsymbol{b}$
:

for $i \leftarrow 1$ *to* ν_1 **do**
  $\boldsymbol{x} \leftarrow \boldsymbol{x} + \texttt{spsolve}(\boldsymbol{A}, \boldsymbol{b} - \boldsymbol{Ax}, \texttt{lower} = \texttt{True})$
end
for $i \leftarrow 1$ *to* ν_1 **do**
  $\boldsymbol{x} \leftarrow \boldsymbol{x} + \texttt{spsolve}(\boldsymbol{A}, \boldsymbol{b} - \boldsymbol{Ax}, \texttt{lower} = \texttt{False})$
end

Algorithm 2: `spsolve`: Lower triangular solver for 3×3 block matrix

Input : $\boldsymbol{A}$: Lower triangular matrix, $\boldsymbol{b}$: A given vector
Output $\boldsymbol{x}$: Solution of $\boldsymbol{Ax} = \boldsymbol{b}$
:

$b_1, b_2, b_3 \leftarrow \texttt{unfold}(\boldsymbol{b})$
$x_1 \leftarrow \texttt{spsolve}(A_{11}, b_1, \texttt{lower} = \texttt{True})$
$\tilde{b}_2 \leftarrow b_2 - A_{21}x_1$
$x_2 \leftarrow \texttt{spsolve}(A_{22}, \tilde{b}_2, \texttt{lower} = \texttt{True})$
$\tilde{b}_3 \leftarrow b_3 - A_{31}x_1 - A_{32}x_2$
$x_3 \leftarrow \texttt{spsolve}(A_{33}, \tilde{b}_3, \texttt{lower} = \texttt{True})$
$\boldsymbol{x} \leftarrow \texttt{fold}(x_1, x_2, x_3)$

Algorithm 3: `spsolve`: Upper triangular solver for 3×3 block matrix

Input : $\boldsymbol{A}$: Upper triangular matrix, $\boldsymbol{b}$: A given vector
Output $\boldsymbol{x}$: Solution of $\boldsymbol{Ax} = \boldsymbol{b}$
:

$b_1, b_2, b_3 \leftarrow \texttt{unfold}(\boldsymbol{b})$
$x_3 \leftarrow \texttt{spsolve}(A_{33}, b_3, \texttt{lower} = \texttt{False})$
$\tilde{b}_2 \leftarrow b_2 - A_{23}x_3$
$x_2 \leftarrow \texttt{spsolve}(A_{22}, \tilde{b}_2, \texttt{lower} = \texttt{False})$
$\tilde{b}_1 \leftarrow b_1 - A_{12}x_2 - A_{13}x_3$
$x_1 \leftarrow \texttt{spsolve}(A_{11}, \tilde{b}_1, \texttt{lower} = \texttt{False})$
$\boldsymbol{x} \leftarrow \texttt{fold}(x_1, x_2, x_3)$

Next, we provide a new implementation for the lower triangular solver that is used in Algorithm 2 (the upper solver used in Algorithm 3 follows the same rationals). We refer to our driver as spsolve. Since the diagonal block matrices in (3) are sums of Kronecker products of 3 matrices, we can derive efficient matrix-free implementation as described in Algorithm 4 (in the case of a sparse matrix (CSR)).

4 Auxiliary space preconditioner

In this section, we present the auxiliary space preconditioning strategy for system (2). To keep the presentation focused, we only introduce the ASP preconditioner (we refer to [2] for more detailed analysis and further discussion of the preconditioner). For this purpose, we introduce the following matrices

- $\boldsymbol{H}$ defines the matrix related to the restriction of $(H_0^1(\Omega))^3$ inner product to $(V_{h,0}(\mathbf{grad}))^3$, and $\boldsymbol{M}$ is the matrix representation related to the restriction of the $(L^2(\Omega))^3$ inner product to $(V_{h,0}(\mathbf{grad}, \Omega))^3$.

Algorithm 4: `spsolve`: Lower triangular solver for sum of Kronecker product [CSR] matrices.

Input : $\boldsymbol{A}$: Lower triangular matrix of the form $\alpha A_1 \otimes A_2 \otimes A_3 + \beta B_1 \otimes B_2 \otimes B_3 + \gamma C_1 \otimes C_2 \otimes C_3$, $\boldsymbol{b}$: A given vector
Output $\boldsymbol{x}$: Solution of $\boldsymbol{Ax} = \boldsymbol{b}$
:

```
// n_l is the number of rows of matrices A_l, B_l, C_l, l = 1, 2, 3.
for i_1 ← 1 to n_1 do
    for i_2 ← 1 to n_2 do
        for i_3 ← 1 to n_3 do
            i ← multi_index(i_1, i_2, i_3)
            y_i ← 0
            a_d ← 1
            for k_1 ← A_1.indptr[i_1] to A_1.indptr[i_1 + 1] − 1 do
                j_1 ← A_1.indices[k_1]
                a_1 ← A_1.data[k_1]
                for k_2 ← A_2.indptr[i_2] to A_2.indptr[i_2 + 1] − 1 do
                    j_2 ← A_2.indices[k_2]
                    a_2 ← A_2.data[k_2]
                    for k_3 ← A_3.indptr[i_3] to A_3.indptr[i_3 + 1] − 1 do
                        j_3 ← A_3.indices[k_3]
                        a_3 ← A_3.data[k_3]
                        j ← multi_index(j_1, j_2, j_3)
                        if i < j then
                            y_i ← y_i + a_1 a_2 a_3 x[j]
                        else
                            a_d ← a_1 a_2 a_3
                        end
                    end
                end
            end
            z_i ← 0
            b_d ← 1
            for k_1 ← B_1.indptr[i_1] to B_1.indptr[i_1 + 1] − 1 do
                j_1 ← B_1.indices[k_1]
                a_1 ← B_1.data[k_1]
                for k_2 ← B_2.indptr[i_2] to B_2.indptr[i_2 + 1] − 1 do
                    j_2 ← B_2.indices[k_2]
                    a_2 ← B_2.data[k_2]
                    for k_3 ← B_3.indptr[i_3] to B_3.indptr[i_3 + 1] − 1 do
                        j_3 ← B_3.indices[k_3]
                        a_3 ← B_3.data[k_3]
                        j ← multi_index(j_1, j_2, j_3)
                        if i < j then
                            z_i ← z_i + a_1 a_2 a_3 x[j]
                        else
                            b_d ← a_1 a_2 a_3
                        end
                    end
                end
            end
            w_i ← 0
            c_d ← 1
            for k_1 ← C_1.indptr[i_1] to C_1.indptr[i_1 + 1] − 1 do
                j_1 ← C_1.indices[k_1]
                a_1 ← C_1.data[k_1]
                for k_2 ← C_2.indptr[i_2] to C_2.indptr[i_2 + 1] − 1 do
                    j_2 ← C_2.indices[k_2]
                    a_2 ← C_2.data[k_2]
                    for k_3 ← C_3.indptr[i_3] to C_3.indptr[i_3 + 1] − 1 do
                        j_3 ← C_3.indices[k_3]
                        a_3 ← C_3.data[k_3]
                        j ← multi_index(j_1, j_2, j_3)
                        if i < j then
                            w_i ← w_i + a_1 a_2 a_3 x[j]
                        else
                            c_d ← a_1 a_2 a_3
                        end
                    end
                end
            end
            x[i] ← 1/(α a_d + β b_d + γ c_d) (b[i] − α y_i − β z_i − γ w_i)
        end
    end
end
```

- We write $\boldsymbol{P}$ and $\boldsymbol{G}$ for matrices related to the transform operators $\Pi^{\mathbf{curl}}_{h,0}|_{(V_{h,0}(\mathbf{grad}))^3}$ and $\mathbf{grad}|_{V_{h,0}(\mathbf{grad})}$, respectively.
- Let $\boldsymbol{L}$ be the matrix related to the mapping

$$(\boldsymbol{\phi}_h, \widetilde{\boldsymbol{\phi}_h}) \in V_{h,0}(\mathbf{grad}) \times V_{h,0}(\mathbf{grad}) \longmapsto \left(\mathbf{grad}\, \boldsymbol{\phi}_h, \mathbf{grad}\, \widetilde{\boldsymbol{\phi}_h}\right).$$

- $\boldsymbol{S}$ stands for the matrix related to the smoother.

With these notations, ASP preconditioner for problem (2) is given by

$$\boldsymbol{B} = \boldsymbol{S} + \boldsymbol{K}, \quad \boldsymbol{K} := \boldsymbol{P}\left(\boldsymbol{H} + \tau \boldsymbol{M}\right)^{-1} \boldsymbol{P}^T + \tau^{-1} \boldsymbol{G} \boldsymbol{L}^{-1} \boldsymbol{G}^T. \tag{4}$$

The smoother $\boldsymbol{S}$ can be chosen by a simple relaxation scheme such as the Jacobi and symmetric Gauss-Seidel (GS) method. In this case, it has been proved in [2] that the spectral condition number $\kappa(\boldsymbol{BA})$ is bounded, with respect to discretization parameter h. However, the numerical tests developed in the aforementioned paper show that the overall performance obtained with Gauss-Seidel smoother is better than that obtained with Jacobi. That's why in the present paper we focus on the symmetric Gauss-Seidel method.

5 hp-Robust preconditioning algorithm

In this section, we introduce the ASP-GS-GLT algorithm, which is based on the ASP method and addresses the problem related to the B-Spline degree. Indeed, the ASP approach can be extended to construct a p-stable preconditioner by incorporating an extra smoother that controls the p-dependency of the preconditioner. To derive the smoother, we use the theory of Generalized Locally Toeplitz (GLT) sequences (see [5]).

The ASP-GS-GLT algorithm is formulated using the decomposition (4) as follows:

Algorithm 5: ASP-GS-GLT preconditioning for $\boldsymbol{V}_{h,0}(\mathbf{curl})$

Input : $\boldsymbol{A}$: The matrix given in (3), $\boldsymbol{b}$: A given vector, $\boldsymbol{x}$: A starting point, ν_1: The number of GS iterations , ν_2: The number of GLT iterations , ν_{ASP}: The number of ASP iterations
Output $\boldsymbol{x}$: The approximate solution of $\boldsymbol{Ax} = \boldsymbol{b}$
:
$k \leftarrow 0$
while $k \leq \nu_{ASP}$ ***and*** *not convergence* **do**
  $\boldsymbol{x} \leftarrow \mathtt{smoother}_1(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}, \nu_1)$ // Apply Symmetric GS smoother
  $\boldsymbol{x} \leftarrow \mathtt{smoother}_2(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}, \nu_2)$ // Apply GLT smoother
  $\boldsymbol{d} \leftarrow \boldsymbol{b} - \boldsymbol{Ax}$ // Compute the defect
  $\boldsymbol{x}_c \leftarrow \boldsymbol{K}\,\boldsymbol{d}$ // ASP correction
  $\boldsymbol{x} \leftarrow \boldsymbol{x} + \boldsymbol{x}_c$ // Update the solution
  $k \leftarrow k + 1$
end

Algorithm 5 is built upon three building blocks: a symmetric Gauss-Seidel smoothing, a GLT-based smoother, and an ASP correction. To implement the Gauss-Seidel smoothing, we employ the block fast Gauss-Seidel method described in Section 3.

Our GLT-smoothing strategy is adapted from the work of [5]. Additionally, the ASP correction utilizes solvers for Poisson problems to compute solutions for systems with matrices $\boldsymbol{H} + \tau \boldsymbol{M}$ and $\boldsymbol{L}$. For this purpose, we rely on the fast diagonalization method introduced in [8].

6 Numerical results

In this section, we present some numerical experiments to test the strategy proposed in this paper in view of further applications. In all these tests, we consider the model problem (1) in the computational domain $\Omega = (0, 1)^3$ subdivided into $2^k \times 2^k \times 2^k$ sub-domains ($k \geq 1$). As a right-hand side function we chose $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = (\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$. The IgA discrete system (2) is solved by the Conjugate Gradient (CG) method in the case of the un-preconditioned and preconditioned systems. The stopping criteria is $\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|/\|\boldsymbol{b}\| \leq 10^{-6}$ and the initial guess is chosen to be the zero vector.

Table 1 Un-preconditioned (NP) and ASP preconditioner (ASP): CG iterations counts for different values of $h = 1/2^k$ and p. '−' means that CG reaches the maximum number of iterations (set to 3000) without convergence. Parameter values $\tau = 10^{-4}$, $\nu_1 = 1$, $\nu_2 = p + 1$ and $\nu_{asp} = 3$.

	$h = 1/8$		$h = 1/16$		$h = 1/32$		$h = 1/64$			$h = 1/8$		$h = 1/16$		$h = 1/32$		$h = 1/64$	
p	NP	ASP	NP	ASP	NP	ASP	NP	ASP	p	NP	ASP	NP	ASP	NP	ASP	NP	ASP
1	151	3	328	4	511	6	879	6	6	–	4	–	4	–	4	–	4
2	520	2	975	4	1313	5	1962	6	7	–	4	–	4	–	4	–	4
3	–	2	–	3	–	4	–	6	8	–	4	–	4	–	4	–	4
4	–	3	–	3	–	4	–	5	9	–	5	–	4	–	4	–	5
5	–	3	–	3	–	4	–	5	10	–	5	–	5	–	5	–	5

Table 2 ASP preconditioner: CG iterations counts for different values of τ and p. '−' means that CG reaches the maximum number of iterations (set to 3000) without convergence. Parameter values $h = 1/64$, $\nu_1 = 1$, $\nu_2 = p + 1$ and $\nu_{asp} = 3$.

	$p = 1$		$p = 3$		$p = 8$			$p = 1$		$p = 3$		$p = 8$	
τ	NP	ASP	NP	ASP	NP	ASP	τ	NP	ASP	NP	ASP	NP	ASP
10^{-4}	879	6	–	6	–	4	10	244	6	1180	5	–	4
10^{-3}	755	6	–	6	–	4	10^2	131	6	361	4	2227	3
10^{-2}	610	7	–	6	–	4	10^3	41	4	101	2	687	4
10^{-1}	486	7	–	5	–	4	10^4	10	1	39	2	320	3
1	295	7	2278	5	–	4	10^5	9	1	39	1	318	2

In the first test, we keep following the number of the CG iterations for convergence for different values of k and p. The results are shown in Table 1. As we can observe from the table, this example indicates that our ASP preconditioner is robust in the sense that the number of iterations necessary to achieve the convergence is sufficiently small and is hardly dependent on the mesh parameter h and the B-spline degree p.

In the second test, we study the dependence of the ASP preconditioner on the parameter τ. For this objective, in Table 2 we provide GC iteration counts for different values of τ and p. The table shows a strong dependence of the un-preconditioned problem on τ, In contrast, however, the number of CG iterations, in the case of ASP preconditioner, is independent of τ. This shows that the ASP method is perfectly able to handle small values of τ.

Acknowledgements This action benefited from the support of the Chair "Multiphysics and HPC" led by Mohammed VI Polytechnic University, sponsored by OCP.

References

1. Da Veiga, L. B., Buffa, A., Sangalli, G., and Vázquez, R. Mathematical analysis of variational isogeometric methods. *Acta Numerica* **23**, 157–287 (2014).
2. El Akri, A., Jbilou, K., and Ratnani, A. Auxiliary splines space preconditioning for b-spline finite elements: the case of $\mathbf{H}(\mathbf{curl}, \omega)$ and $\mathbf{H}(\mathrm{div}, \omega)$ elliptic probelems. *arXiv preprint arXiv:2303.08375* (2023).
3. Hiptmair, R. and Xu, J. Nodal auxiliary space preconditioning in $\mathbf{H}(\mathbf{curl}, \omega)$ and $\mathbf{H}(\mathrm{div}, \omega)$ spaces. *SIAM Journal on Numerical Analysis* **45**(6), 2483–2509 (2007).
4. Hughes, T., Cottrell, J., and Bazilevs, Y. Isogeometric analysis: Cad, finite elements, nurbs, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering* **194**(39-41), 4135–4195 (2005).
5. Mazza, M., Manni, C., Ratnani, A., Serra-Capizzano, S., and Speleers, H. Isogeometric analysis for 2d and 3d $\mathbf{curl}-\mathrm{div}$ problems: Spectral symbols and fast iterative solvers. *Computer Methods in Applied Mechanics and Engineering* **344**, 970–997 (2019).
6. Piegl, L. and Tiller, W. *The NURBS book*. Springer Science & Business Media (1996).
7. Ratnani, A. and Sonnendrücker, E. An arbitrary high-order spline finite element solver for the time domain maxwell equations. *Journal of Scientific Computing* **51**(1), 87–106 (2012).
8. Sangalli, G. and Tani, M. Isogeometric preconditioners based on fast solvers for the sylvester equation. *SIAM Journal on Scientific Computing* **38**(6), A3644–A3671 (2016).
9. Xu, J. The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids. *Computing* **56**(3), 215–235 (1996).

Composing Two Different Nonlinear FETI–DP Methods

Stephan Köhler and Oliver Rheinbach

1 Nonlinear FETI–DP

Nonlinear FETI–DP methods [3] are nonlinear generalizations of linear FETI–DP domain decomposition methods [10]. Using a divide-and-conquer approach, the unconstrained minimization of some objective J is transformed into a constrained optimization problem over many subdomains,

$$\min_{\tilde{u}} \widetilde{J}(\tilde{u}) \quad \text{subject to (s.t.)} \quad B\tilde{u} = 0, \tag{1}$$

where the constraint $B\tilde{u} = 0$ enforces continuity across subdomain boundaries; here, $\tilde{u} := [u_{BB}^{(1)}, \ldots, u_{BB}^{(N)}, \tilde{u}_\Pi]^T$, where the subscript B refers to the union of the inner and dual variables, $\widetilde{J}(\tilde{u}) := \sum_{i=1}^{N} J^{(i)}(u_{BB}^{(i)}, R_\Pi^{(i)} \tilde{u}_\Pi)$, $J^{(i)}$ is the local objective of the i-th subdomain, $R_\Pi^{(i)}$ is the assembly operator of the primal variables as in linear FETI–DP methods [10]. The Lagrange function for (1) is $\mathcal{L}(\tilde{u}, \lambda) = \widetilde{J}(\tilde{u}) + \lambda^T B\tilde{u}$. The saddle point problem of the first-order necessary optimality condition

$$\begin{aligned} \nabla_{\tilde{u}} \mathcal{L}(\tilde{u}, \lambda) &= \nabla \widetilde{J}(\tilde{u}) + B^T \lambda = \tilde{f}, \\ \nabla_{\lambda} \mathcal{L}(\tilde{u}, \lambda) &= B\tilde{u} = 0, \end{aligned} \tag{2}$$

corresponds directly to the linear FETI–DP saddle point problem [10]. The nonlinear operator $\nabla \widetilde{J}(\tilde{u}) := R_\Pi^T \nabla J(R_\Pi \tilde{u})$ is obtained from finite element subassembly of the blocks $\nabla J^{(i)}(u_{BB}^{(i)}, R_\Pi^{(i)} \tilde{u}_\Pi)$ in the primal variables using the operator R_Π^T as in linear FETI–DP methods [10]. This coupling provides a nonlinear coarse problem for the method. Thus, $\nabla \widetilde{J}$ represents a nonlinear coarse approximation of the original problem.

Stephan Köhler, Oliver Rheinbach
Institut für Numerische Mathematik und Optimierung, Fakultät für Mathematik und Informatik, Technische Universität Bergakademie Freiberg, 09596 Freiberg,
e-mail: oliver.rheinbach@math.tu-freiberg.de, stephan.koehler@math.tu-freiberg.de

Next, we perform the nonlinear elimination: we split the first row in (2) according to disjoint index sets E, L (eliminate or linearize) and solve in a first step

$$\nabla_{\tilde{u}_E} \mathcal{L}(\tilde{u}_E, \tilde{u}_L, \lambda) = \nabla_{\tilde{u}_E} \widetilde{J}(\tilde{u}_E, \tilde{u}_L) + B_E^T \lambda = 0, \tag{3}$$

for $\tilde{u}_E$, given $\tilde{u}_L$ and λ. Then, we can insert $\tilde{u}_E$ into the remaining equations and solve by linearization in $\tilde{u}_L$ and λ, and using the implicit function theorem. In [3], four different elimination sets are considered: Nonlinear FETI–DP-1 (*NL*-1), where $E = \emptyset$, nonlinear FETI–DP-2 (*NL*-2), where E contains all variables and $L = \emptyset$, nonlinear FETI–DP-3 (*NL*-3), where E contains the inner and the dual variables, and Nonlinear FETI–DP-4 (*NL*-4), where E contains only the inner variables. Here, we focus on the two elimination sets of *NL*-2 and *NL*-4.

2 Composing two different nonlinear FETI–DP methods

We combine the two different nonlinear FETI–DP methods *NL*-2, where all variables are eliminated, and *NL*-4 where only the inner variables are eliminated. The idea is based on [9], where a similar approach is successfully applied for nonlinear FETI–1. In a first step, for given multipliers $\lambda^{(k)}$, the implicit function $g_1(\lambda^{(k)})$, such that

$$\nabla_{\tilde{u}} \mathcal{L}\big|_{(g_1(\lambda^{(k)}), \lambda^{(k)})} = 0, \tag{4}$$

where we denote the evaluation of $\nabla_{\tilde{u}} \mathcal{L}$ at the point $(g_1(\lambda^{(k)}), \lambda^{(k)})$ by $\nabla_{\tilde{u}} \mathcal{L}\big|_{(g_1(\lambda^{(k)}), \lambda^{(k)})}$, is computed. The function g_1 corresponds to *NL*-2. Afterwards, we compute a weighted average over the interface by

$$g_2(\lambda^{(k)}) := (I - {B^*}^T B) g_1(\lambda^{(k)}), \tag{5}$$

where B^* is a pseudo-inverse for B such that $B\, {B^*}^T B = B$ and $B^* B^T B^* = B^*$. A cost-saving variant for B^* is $B^* = B_{D,\Gamma}$, where $B_{D,\Gamma}$ corresponds to the Dirichlet preconditioner of the standard FETI–DP method; see, e.g., [8]. Due to (5), it follows immediately that $B g_2(\lambda^{(k)}) = 0$. However, there is an unnatural tension between the interface variables of g_2 and the variables adjacent to them. To resolve this tension, in a third step, we compute the implicit function corresponding to *NL*-4,

$$g_3(\lambda^{(k)}) := \left(h(g_{2,\Delta}(\lambda^{(k)}), g_{2,\Pi}(\lambda^{(k)}), \lambda^{(k)})^T,\, g_{2,\Delta}(\lambda^{(k)})^T,\, g_{2,\Pi}(\lambda^{(k)})^T\right)^T, \tag{6}$$

where $g_{2,\Delta}$, $g_{2,\Pi}$ are the dual, primal variables of g_2, respectively, and the implicit function $h(g_{2,\Delta}(\lambda^{(k)}), g_{2,\Pi}(\lambda^{(k)}), \lambda^{(k)})$ solves

$$\nabla_I \mathcal{L}\big|_{\left(h\left(g_{2,\Delta}(\lambda^{(k)}), g_{2,\Pi}(\lambda^{(k)}), \lambda^{(k)}\right),\, g_{2,\Delta}(\lambda^{(k)}),\, g_{2,\Pi}(\lambda^{(k)}),\, \lambda^{(k)}\right)} = 0, \tag{7}$$

where we denote the gradient with respect to the inner variables of $\tilde{u}$ by ∇_I.

We obtain the new multipliers by $\lambda^{(k+1)} = -B^T B^* \nabla \widetilde{J}\big|_{g_3(\lambda^{(k)})}$, for details, see [9]. If $(\tilde{u}^*, \lambda^*)$ is a KKT point of (1) and if $\nabla_{\tilde{u}\tilde{u}} \mathcal{L}\big|_{(\tilde{u}^*,\lambda^*)}$ is invertible, it follows from (4) by the implicit function theorem that $g_1(\lambda^*) = \tilde{u}^*$. Furthermore, it follows that $B\tilde{u}^* = 0$ and therefore, from (4) and (5), we have $g_1(\lambda^*) = g_2(\lambda^*)$. From (4), it follows that $g_1(\lambda^*) = g_3(\lambda^*)$. By the first part of the first-order necessary optimality condition (2), we have $\nabla \widetilde{J}(\tilde{u}^*) = -B^T \lambda^*$. Therefore, the nonlinear root-finding problem is

$$r(\lambda) := -B^T B^* \nabla \widetilde{J}\big|_{g_3(\lambda^{(k)})} - B^T \lambda = -B^T B^* \nabla_{\tilde{u}} \mathcal{L}\big|_{g_3(\lambda^{(k)})}. \tag{8}$$

Since we assume that B^T has full rank, which can always guaranteed by the use of nonredundant multipliers, we can rewrite (8) as $r(\lambda) := B^* \nabla_{\tilde{u}} \mathcal{L}\big|_{g_3(\lambda^{(k)})}$. We apply Newton's method to $r(\lambda)$. By similar arguments as outlined in [9], we have

$$Dr(\lambda) \approx \left(B^*_{\widetilde{\Gamma}} S_{\widetilde{\Gamma}\widetilde{\Gamma}}\big|_{(g_3(\lambda),\lambda)} B^{*T}_{\widetilde{\Gamma}} B_{\widetilde{\Gamma}} \; S_{\widetilde{\Gamma}\widetilde{\Gamma}}\big|_{(g_3(\lambda),\lambda)}{}^{-1} \; B^T_{\widetilde{\Gamma}} \right), \tag{9}$$

where

$$S_{\widetilde{\Gamma}\widetilde{\Gamma}}\big|_{(g_3(\lambda),\lambda)} := \left(\begin{matrix} \nabla^2_{\Delta\Delta}\mathcal{L} - \nabla^2_{\Delta I}\mathcal{L}\, \nabla^2_{II}\mathcal{L}^{-1}\, \nabla^2_{I\Delta}\mathcal{L} & \nabla^2_{\Delta\Pi}\mathcal{L} - \nabla^2_{\Delta I}\mathcal{L}\, \nabla^2_{II}\mathcal{L}^{-1}\, \nabla^2_{I\Pi}\mathcal{L} \\ \nabla^2_{\Pi\Delta}\mathcal{L} - \nabla^2_{\Pi I}\mathcal{L}\, \nabla^2_{II}\mathcal{L}^{-1}\, \nabla^2_{I\Delta}\mathcal{L} & \nabla^2_{\Pi\Pi}\mathcal{L} - \nabla^2_{\Pi I}\mathcal{L}\, \nabla^2_{II}\mathcal{L}^{-1}\, \nabla^2_{I\Pi}\mathcal{L} \end{matrix} \right)\Bigg|_{(g_3(\lambda),\lambda)}$$

and $B^*_{\widetilde{\Gamma}}$, $B_{\widetilde{\Gamma}}$ correspond to the interface part of B^*, B, respectively. Let us remark that the approximation (9) uses $g_1(\lambda) \approx g_3(\lambda)$.

The operator in (9) consists of two parts: the FETI–DP system matrix $B_{\widetilde{\Gamma}} \; S_{\widetilde{\Gamma}\widetilde{\Gamma}}\big|_{(g_3(\lambda),\lambda)}{}^{-1} \; B^T_{\widetilde{\Gamma}}$ and the Dirichlet preconditioner $B^*_{\widetilde{\Gamma}} S_{\widetilde{\Gamma}\widetilde{\Gamma}}\big|_{(g_3(\lambda),\lambda)} B^{*T}_{\widetilde{\Gamma}}$; for the linear case, see, e.g., [8, 10] and for the nonlinear case, see, e.g.,[3, 7]. The Newton equation for $r(\lambda)$ is given by

$$\left(B^*_{\widetilde{\Gamma}} S_{\widetilde{\Gamma}\widetilde{\Gamma}} B^{*T}_{\widetilde{\Gamma}} B_{\widetilde{\Gamma}} \; S_{\widetilde{\Gamma}\widetilde{\Gamma}}{}^{-1} \; B^T_{\widetilde{\Gamma}} \right)\Big|_{(g_3(\lambda),\lambda)} \delta\hat{\lambda} = -B^* \nabla_{\tilde{u}} \mathcal{L}\big|_{(g_3(\lambda),\lambda)}. \tag{10}$$

The system matrix in (10) corresponds to the system matrix of a standard preconditioned FETI-DP system, however, the preconditioner $B^*_{\widetilde{\Gamma}} S_{\widetilde{\Gamma}\widetilde{\Gamma}} B^{*T}_{\widetilde{\Gamma}}$ is not applied to the right hand side, which is an important difference.

For completeness, we show the preconditioned Newton equation for *NL*-2:

$$\left(B^*_{\widetilde{\Gamma}} S_{\widetilde{\Gamma}\widetilde{\Gamma}} B^{*T}_{\widetilde{\Gamma}}\, B_{\widetilde{\Gamma}}\, S_{\widetilde{\Gamma}\widetilde{\Gamma}}{}^{-1}\, B^T_{\widetilde{\Gamma}}\right)\Big|_{(g_1(\lambda),\lambda)} \delta\lambda = -B^*_{\widetilde{\Gamma}} S_{\widetilde{\Gamma}\widetilde{\Gamma}}\Big|_{(g_1(\lambda),\lambda)} B^{*T}_{\widetilde{\Gamma}}\, B g_1(\lambda).$$

Note, the difference between the two equations is the evaluation point of the operator and the right hand side.

3 Globalization of nonlinear FETI–DP

For the globalization of the method outlined in Section 2, we use the exact differentiable penalty function

$$P(\tilde{u}, \lambda; M, \mu) = \mathcal{L}(\tilde{u}, \lambda) + \tfrac{\mu}{2}\|c(\tilde{u})\|^2 + \tfrac{1}{2}\|M\nabla_{\tilde{u}}\mathcal{L}(\tilde{u}, \lambda)\|^2 \tag{11}$$

introduced in [2]. For a detailed analysis of P, we refer to [1]. For nonlinear FETI–DP, we have $c(\tilde{u}) = B\tilde{u}$ and $M = \eta\, B$, where B is the FETI–DP jump operator. First results for globalization of nonlinear FETI–DP by P were presented in [6] and for a detailed analysis we refer to [5]. The methods presented in [6] make explicit use of the nonlinear elimination. Indeed, the nonlinear elimination needs to be computed in every step of the backtracking. Such an approach for the function g_3 from Section 2 is computationally expensive also when computing the exact Jacobian Dg_3. Let us keep in mind that (9) uses the approximation $g_3(\lambda) \approx g_1(\lambda)$.

Simplified backtracking Hence, we need to modify the globalization approach. The main idea is that for given point $(\tilde{u}^{(k)}, \lambda^{(k)})$ we compute a new trial point $(g_3(\hat{\lambda}^{(k)}), \hat{\lambda}^{(k)})$, where $\hat{\lambda}^{(k)} = \lambda^{(k)} + \delta\hat{\lambda}^{(k)}$ and $\delta\hat{\lambda}^{(k)}$ is the solution of (10) at $(g_3(\lambda^{(k)}), \lambda^{(k)})$. Afterwards, we compute our search direction by

$$d_1^{(k)} = \left((g_3(\hat{\lambda}^k) - \tilde{u}^{(k)})^T,\, \delta\hat{\lambda}^{(k)\,T}\right)^T. \tag{12}$$

Similarly as for a Newton-direction, it is unclear if $d_1^{(k)}$ is a descent direction. Therefore, we must ensure that a generalized angle condition holds if we use $d_1^{(k)}$. If $d_1^{(k)}$ does not fulfill a generalized angle condition, i.e.,

$$\nabla\, P^{(k)\,T}\, d_1^k \geq -\min\{\eta_1, \eta_2\, \|\nabla P^{(k)}\|_\infty^p\}\, \|d_1^{(k)}\|_2\, \|\nabla P^{(k)}\|_\infty, \tag{13}$$

where $\nabla P^{(k)} := \nabla P\big|_{(\tilde{u}^{(k)}, \lambda^{(k)}; M, \mu_k)}$ or if

$$\|d_1^{(k)}\|_2 < \eta_3 \left(-\nabla\, P^{(k)\,T}\, d_1^{(k)}\right) / \|d_1^{(k)}\|_2, \tag{14}$$

we compute a new direction $d_2^{(k)}$ by the solution of the standard Lagrange-Newton equation at the point $(\tilde{u}^{(k)}, \lambda^{(k)})$. Let us remark that the solution of the Lagrange-Newton equation correspond to a Newton-like search direction for P; see, e.g., [1].

Init: $(\tilde{u}^{(0)}, \lambda^{(0)})$, $\eta_1, \rho \in (0,1)$, $\varepsilon_{\text{update}} > 1$, $\varepsilon_{\text{tol}}, \mu_0, \eta_2, \eta_3, p > 0$.

for $k = 0, 1, \ldots$ until convergence **do**

1. If $\|\nabla \mathcal{L}^{(k)}\|_\infty \leq \varepsilon_{\text{tol}}$, *STOP*.
2. (a) Compute $g_3(\lambda^{(k)})$ // Includes computation of NL-2 and NL-4.
   (b) Solve $B^*_{\widetilde{\Gamma}} S^{(k)}_{\widetilde{\Gamma}\widetilde{\Gamma}} B^{*T}_{\widetilde{\Gamma}} B_{\widetilde{\Gamma}} \, S^{(k)\,-1}_{\widetilde{\Gamma}\widetilde{\Gamma}} \, B^T_{\widetilde{\Gamma}} \, \delta\hat{\lambda}^{(k)} = -B^* \nabla_{\tilde{u}} \mathcal{L}^{(k)}$.
   (c) Compute $g_3(\lambda^{(k)} + \delta\hat{\lambda}^{(k)})$ // Includes computation of NL-2 and NL-4.
   (d) Set $d^{(k)} = \begin{pmatrix} g_3(\lambda^{(k)} + \delta\hat{\lambda}^{(k)}) - \tilde{u}^{(k)} \\ \delta\hat{\lambda}^{(k)} \end{pmatrix}$.

   if (13) **or** (14) **then**

   Set $d^{(k)} = -\nabla^2 \mathcal{L}^{-1}\big|_{(\tilde{u}^{(k)}, \lambda^{(k)})} \nabla \mathcal{L}\big|_{(\tilde{u}^{(k)}, \lambda^{(k)})}$

   if (13) **then**

   Set $d^{(k)} = -\nabla^2 P\big|_{(\tilde{u}^{(k)}, \lambda^{(k)})}$

   end

   end
3. Compute the step length α_k based on the Armijo rule.
4. Set $\tilde{u}^{(k+1)} = \tilde{u}^{(k)} + \alpha_k d^{(k)}_{\tilde{u}}$ and $\lambda^{(k+1)} = \lambda^{(k)} + \alpha_k d^{(k)}_{\lambda}$, where $d^{(k)} = (d^{(k)\,T}_{\tilde{u}}, d^{(k)\,T}_{\lambda})^T$.
5. **if** $\| B\tilde{u}^{(k+1)} \| \geq \rho \| B\tilde{u}^{(k)} \|$ **then**

   Set $\mu_{k+1} = \varepsilon_{\text{update}} \mu_k$.

   else

   Set $\mu_{k+1} = \mu_k$.

   end

end

Fig. 1: Minimization algorithm for P.

If $d_2^{(k)}$ does also not fulfill the generalized angle condition (13), we use $-\nabla P^{(k)}$ as the search direction. Afterwards, we compute the step length by the Armijo rule. In contrast to the algorithm outlined in [6], we do not compute any nonlinear elimination in the backtracking of the Armijo rule. This is important to reduce the runtime, since the computation of $g_3(\lambda^{(k)})$ takes some effort. We refer to this as *simplified backtracking*.

Minimization algorithm We outline our minimization algorithm in Fig. 1. Let us explain some details: Our framework is based on a general line search algorithm and a globalized Newton line search algorithm; for details, see, e.g., [11]. We use the simplified backtracking to save some runtime. Furthermore, we do not rely on the exact computation of $g_3(\lambda^{(k)})$, since we are only interested in a descent for P. Therefore, we can abort the computation of the $g_1(\lambda^{(k)})$, which corresponds to the nonlinear elimination of *NL*-2, after a few iterations. This holds also for the nonlinear elimination corresponding to the *NL*-4 step. By such an inexact nonlinear elimination, we try to avoid over solving of the nonlinear elimination problems and the idea is based on the approach in [4]. The inexact nonlinear elimination is an important difference to Newton's method applied to $r(\lambda)$, which needs, formally, the exact computation of $g_3(\lambda^{(k)})$. The disadvantage in Fig. 1 is that if Fig. 1 2. (b) does not provide a descent direction, we have to compute and factorize $\nabla^2 \mathcal{L}$ again at the old point $(\tilde{u}^{(k)}, \lambda^{(k)})$, which takes some additional runtime.

Let us remark that the globalization strategy in Fig. 1 combined with an inexact computation of g_3 is based on the ideas in [5].

4 Numerical results

We consider red a two-dimensional beam bending benchmark problem with a Neo-Hookean constitutive law using no or almost incompressible inclusions embedded in each subdomain. The strain energy density function for the compressible matrix material part is given by $J(x) = \frac{\mu}{2}(\mathrm{tr}(F(x)^T F(x)) - 2) - \mu \log(\psi(x)) + \frac{\lambda}{2}(\log(\psi(x)))^2$, where $\psi(x) = \det(F(x))$, $F(x) = \nabla\varphi(x)$, $\varphi(x) = x + u(x)$, $u(x)$ denotes the displacement and μ and λ are the Lamé constants. The nearly incompressible part is given by $J(x) = \frac{\mu}{2}(\mathrm{tr}(\frac{1}{\psi(x)} F(x)^T F(x)) - 2) + \frac{\kappa}{2}(\psi(x) - 1)^2$, where $\kappa = \frac{\lambda(1+\mu)}{3\mu}$; see, e.g. [12, 5] and references therein. As material parameters, we use $E = 210$ and $\nu = 0.3$ for the matrix material and $E = 210$ and $\nu = 0.499$ for the (mildly) almost incompressible inclusions. For the discretization, we use $P2$ elements, which are not stable for the incompressible case.

For the computation of $g_1(\lambda^{(k)})$ in Fig. 1 (a), we solve the minimization problem $\min_{\tilde{u}} \mathcal{L}(\tilde{u}, \lambda^{(k)})$. We solve this problem inexactly, in the sense that we abort the computation if $\frac{|\mathcal{L}(\tilde{u}_{\ell+1}^{(k)}, \lambda^{(k)}) - \mathcal{L}(\tilde{u}_{\ell}^{(k)}, \lambda^{(k)})|}{|\mathcal{L}(\tilde{u}_{\ell}^{(k)}, \lambda^{(k)})|} < \gamma_1$ or if $(1 - \gamma_2) < \frac{\|\nabla_{\tilde{u}} \mathcal{L}(\tilde{u}_{\ell+1}^{(k)}, \lambda^{(k)})\|_\infty}{\|\nabla_{\tilde{u}} \mathcal{L}(\tilde{u}_{\ell}^{(k)}, \lambda^{(k)})\|_\infty}$, where $\tilde{u}_{\ell+1}^{(k)}$ is the current iterate in the computation of $g_1(\lambda^{(k)})$ and $\tilde{u}_{\ell}^{(k)}$ is the previous one. In a similar way, we compute also the NL-4 part of $g_3(\lambda^{(k)})$ inexactly. Let us remark that we use a globalized Newton method with a computation of the Newton step using a direct sparse solver for the Newton equation. This can be afforded since this is an operation local to the subdomains and for g_1 this involves also the (small) coarse space.

As Krylov methods in Fig. 1, we use GMRES. In Table 1, we show the number of (outer) iterations for the $2D$ Neo-Hookean beam bending benchmark problem with a homogeneous material model, see upper part of Table 1, and (mildly) almost incompressible inclusions, see lower part of Table 1. We report the iterations for the standard nonlinear FETI–DP methods NL-1, NL-2, and NL-2 with simplified backtracking, which includes also the inexact nonlinear elimination and the new approach outlined in section 2, NL-2/4, with simplified backtracking and inexact nonlinear elimination. In brackets, we show the cumulative iterations for the nonlinear elimination corresponding to the NL-2 elimination set, we refer to this as inner iterations. This does not include the iterations for the nonlinear elimination corresponding to the NL-4 part in NL-2/4.

In the upper part of Table 1 shows that there is a small increase in the number of outer iterations for NL-2 simpl. compared to the standard NL-2 method; the number of inner iterations, however, decreases significantly. For NL-2/4 simpl., the increase of the number of outer iterations is more significant, but the number of inner iterations stays the same or decreases slightly, again, compared with the NL-2 method. Let us remark that for 4 000 subdomains and NL-2, we need 6 outer iterations instead of the previous 2, since we need to make 4 gradient steps, which are not as effective as Newton steps.

Table 1: Nonlinear FETI–DP-1, 2 (NL-1, 2), nonlinear FETI–DP-2 with simplified backtracking (NL-2 simpl.), and nonlinear FETI–DP-2/4 with simplified backtracking (NL-2/4 simpl.); $H/h \approx 21$; beam bending problem in $2D$; coarse space: vertices, edge, and rotational averages; globalization based on P and using Fig. 1 for NL-2 linear. and NL-2/4 linear. for NL-1, 2 see [5, 6]; start penalty parameter $\mu_0 = 100$; exact diff penalty method; number of iteration is shown, in brackets the cumulative number of nonlinear elimination steps for the NL-2 part; stopping criterion: $\|\nabla \mathcal{L}^{(k)}\|_\infty < 10^{-6}$.

using globalization: NL-1,2, see [6, 5]; NL-2 linear., NL-2-4 linear. see Fig. 1					
homogeneous Neo-Hooke					
		body force $f = (0, -1.0)^T$			
		Standard Methods		New Methods	
#d.o.f.	#Sub.	NL-1	NL-2	NL-2 simpl.	NL-2/4 simpl.
963 202	250	16	2 [16]	3 [11]	5 [16]
3 844 392	1 000	15	1 [15]	3 [11]	5 [16]
15 360 772	4 000	15	1 [15]	3 [11]	5 [15]
		body force $f = (0, -2.0)^T$			
		Standard Methods		New Methods	
#d.o.f.	#Sub.	NL-1	NL-2	NL-2 simpl.	NL-2/4 simpl.
963 202	250	17	2 [19]	4 [14]	6 [17]
3 844 392	1 000	17	2 [19]	4 [14]	6 [17]
15 360 772	4 000	17	6 [28]	3 [13]	6 [18]
incomp. inclusions ($\nu = 0.499$)					
		body force $f = (0, -1.0)^T$			
		Standard Methods		New Methods	
#d.o.f.	#Sub.	NL-1	NL-2	NL-2 simpl.	NL-2/4 simpl.
963 202	250	36	2 [36]	9 [26]	13 [33]
3 844 392	1 000	36	2 [38]	9 [25]	13 [34]
15 360 772	4 000	35	2 [38]	8 [24]	13 [35]
		body force $f = (0, -2.0)^T$			
		Standard Methods		New Methods	
#d.o.f.	#Sub.	NL-1	NL-2	NL-2 simpl.	NL-2/4 simpl.
963 202	250	43	2 [46]	12 [30]	16 [41]
3 844 392	1 000	44	2 [46]	11 [32]	15 [42]
15 360 772	4 000	44	2 [48]	11 [30]	16 [41]

In the lower part of Table 1, we observe a significant increase of the outer iterations for NL-2 simpl. and NL-2/4 simpl. compared to NL-2, but the number of inner iterations descreases significantly for NL-2 simpl. and there is a small improvement for NL-2/4 simpl.

In our experiments, NL-2 simpl. seems to be a give better results than NL-2/4, which was not expected; we suspect that this is due to our large coarse space which includes edge averages and rotations. Hence, the jump at the interface is not large

and therefore the correction part on the interface, which includes NL-4, does not lead to a significant improvement. We guess that this will change if the jump of the interface is larger. Note, however, that using a smaller coarse space resulted in very ill-conditioned tangent systems, i.e., the number Krylov iterations was high.

References

1. Bertsekas, D. P. *Constrained optimization and Lagrange multiplier methods*. Computer Science and Applied Mathematics. Academic Press Inc. [Harcourt Brace Jovanovich, Publishers] New York-London (1982).
2. Di Pillo, G. and Grippo, L. A new class of augmented Lagrangians in nonlinear programming. *SIAM J. Control Optim.* **17**(5), 618–628 (1979).
3. Klawonn, A., Lanser, M., Rheinbach, O., and Uran, M. Nonlinear FETI–DP and BDDC methods: a unified framework and parallel results. *SIAM J. Sci. Comput.* **39**(6), C417–C451 (2017).
4. Klawonn, A., Lanser, M., Rheinbach, O., and Uran, M. On the accuracy of the inner newton iteration in nonlinear domain decomposition. In: *International Conference on Domain Decomposition Methods*, 435–443. Springer (2017).
5. Köhler, S. *Globalization of Nonlinear FETI–DP Methods*. Ph.D. thesis, Technische Universität Bergakademie Freiberg (submitted).
6. Köhler, S. and Rheinbach, O. Globalization of Nonlinear FETI–DP Methods. In: *Domain Decomposition Methods in Science and Engineering XXVI*, *Lect. Notes Comput. Sci. Eng.*, vol. 145, 327–334. Springer Cham (2022).
7. Lanser, M. H. *Nonlinear FETI-DP and BDDC Methods*. Ph.D. thesis, Universität zu Köln (2015).
8. Li, J. and Widlund, O. B. FETI-DP, BDDC, and block cholesky methods. *Internat. J. Numer. Methods Engrg.* **66**(2), 250–271 (2006).
9. Negrello, C., Gosselet, P., and Rey, C. Nonlinearly preconditioned FETI solver for substructured formulations of nonlinear problems. *Mathematics* **9**(24), 3165 (2021).
10. Toselli, A. and Widlund, O. *Domain decomposition methods—algorithms and theory*, *Springer Series in Computational Mathematics*, vol. 34. Springer, Berlin (2005).
11. Ulbrich, M. and Ulbrich, S. *Nichtlineare Optimierung*. Birkhäuser Basel (2012).
12. Uran, M. *High-Performance Computing Two-Scale Finite Element Simulations of a Contact Problem Using Computational Homogenization*. Ph.D. thesis, Universität zu Köln (2020).

Biot Model with Generalized Eigenvalue Problems for Scalability and Robustness to Parameters

Pilhwa Lee

1 Introduction

Poroelasticity, *i.e.*, elasticity of porous media with permeated Darcy flow, is pioneered by Biot [2, 3]. In this paper, we propose a numerical scheme for solving the Biot model with three-fields linear poroelasticity. We consider a discontinuous Galerkin discretization, *i.e.*, the displacement and Darcy flow flux discretized as piecewise continuous in P_1 elements, and the pore pressure as piecewise constant in the P_0 element with a stabilizing term. The emerging formulation is a saddle-point problem, and more specifically, a twofold saddle-point problem.

This indefinite system is computational challenging with slow convergence in iterative methods. It is necessary to incorporate relevant preconditioners for saddle-point problems. There have been decomposition methods through overlapping Schwarz methods [6, 9, 11, 13]. We use GMRES as the outer iterative solver accelerated by a parallelized block-triangular preconditioner with overlapping additive Schwarz method (OAS) for displacement, and Darcy flow flux and Schur complement for pressure by Cholesky factorizations. In order to make the scheme scalable and robust to broad ranges of parameters and their potential heterogeneous distributions, the coarse grid should be well constructed. In the paper, we take the approach of constructing coarse spaces with eigenfunctions based on generalized eigenvalue problems [4, 8, 14]. Specifically, we devise a parallel preconditioner of two-level OAS with coarse grid construction by Generalized Eigenvalue problems in the Overlaps (GenEO) [15].

Pilhwa Lee
Department of Mathematics, Morgan State University, 1700 E. Cold Spring Lane, Baltimore, MD, USA, e-mail: Pilhwa.Lee@morgan.edu

2 Linear poroelastic model

Poroelastic models describe the interaction of fluid flows and deformable elastic porous media saturated in the fluid. Let $\boldsymbol{u}$ be the elastic displacement, p be the pore pressure. We assume that the permeability is homogeneous: $\mathbf{K} = \kappa\mathbf{I}$. Denote $\boldsymbol{z}$ as the Darcy volumetric fluid flux. The quasi-static Biot model reads as:

$$-\nabla \cdot (\sigma(\boldsymbol{u}) - \alpha p\mathbf{I}) = \boldsymbol{f}, \tag{1}$$

$$\mathbf{K}^{-1}\boldsymbol{z} + \nabla p = \boldsymbol{b}, \tag{2}$$

$$\frac{\partial}{\partial t}(\alpha \nabla \cdot \boldsymbol{u} + c_0 p) + \nabla \cdot \boldsymbol{z} = g, \tag{3}$$

where $\sigma(\boldsymbol{u})$ is the deviatoric stress, $\boldsymbol{f}$ is the body force on the solid, $\boldsymbol{b}$ is the body force on the fluid, g is a source or sink term, $c_0 > 0$ is the constrained specific storage coefficient, α is the Biot-Willis constant which is close to 1. For the ease of presentation, we consider mixed partial Neumann and partial Dirichlet boundary conditions. Specifically, the boundary $\partial\Omega$ is divided into the following:

$$\partial\Omega = \Gamma_{\mathrm{d}} \cup \Gamma_{\mathrm{t}} \quad \text{and} \quad \partial\Omega = \Gamma_{\mathrm{p}} \cup \Gamma_{\mathrm{f}},$$

where Γ_{d} and Γ_{t} are for displacement and stress boundary conditions; Γ_{p} and Γ_{f} are for pressure and flux boundary conditions. Accordingly, the boundary conditions are the following:

$$\boldsymbol{u} = \mathbf{0} \quad \text{on } \Gamma_{\mathrm{d}}, \qquad (\sigma(\boldsymbol{u}) - \alpha p\mathbf{I}) \cdot \boldsymbol{n} = \mathbf{t} \quad \text{on } \Gamma_{\mathrm{t}}, \tag{4}$$

$$p = 0 \quad \text{on } \Gamma_{\mathrm{p}}, \qquad \boldsymbol{z} \cdot \boldsymbol{n} = g_2 \quad \text{on } \Gamma_{\mathrm{f}}. \tag{5}$$

For simplicity, the Dirichlet conditions are assumed to be homogeneous.

3 Saddle-point problem: discretization of $\mathbf{P_1} - \mathbf{P_1} - P_0$

We apply the finite element method where domains are normally shaped as triangles in $\mathbb{R}^2$. Let $\mathcal{T}_h$ be a partition of Ω into non-overlapping elements K. We denote by h the size of the largest element in $\mathcal{T}_h$. On the given partition $\mathcal{T}_h$ we apply the following finite element spaces [1].

$$\boldsymbol{V}_h := \{\boldsymbol{u}_h \in (C^0(\Omega))^d : \mathbf{u}_h|K \in \mathbf{P}_1(K)\ \forall K \in \mathcal{T}^h, \boldsymbol{u}_h = 0 \text{ on } \Gamma_{\mathrm{d}}\} \tag{6}$$

$$\boldsymbol{W}_h := \{z_h \in (C^0(\Omega))^d : \mathbf{z}_h|K \in \mathbf{P}_1(K)\ \forall K \in \mathcal{T}^h, z_h \cdot \boldsymbol{n} = 0 \text{ on } \Gamma_{\mathrm{f}}\} \tag{7}$$

$$Q_h := \{p_h : p_h|K \in \mathbf{P}_0(K)\ \forall K \in \mathcal{T}^h\} \tag{8}$$

The problem is to find $(\boldsymbol{u}_h^n, \boldsymbol{z}_h^n, p_h^n) \in \boldsymbol{V}_h \times \boldsymbol{W}_h \times Q_h$ at the time step n such that

$$\begin{cases} a(\boldsymbol{u}_h^n, \boldsymbol{v}_h) - (p_h^n, \nabla \cdot \boldsymbol{v}_h) = (\boldsymbol{f}^n, \boldsymbol{v}_h) + (\boldsymbol{t}^n, \boldsymbol{v}_h)_{\Gamma_t}, \ \forall \boldsymbol{v}_h \in \boldsymbol{V}_h \\ (K^{-1}\boldsymbol{z}_h^n, \boldsymbol{w}_h) - (p_h^n, \nabla \cdot \boldsymbol{w}_h) = (\boldsymbol{b}^n, \boldsymbol{w}_h), \ \forall \boldsymbol{w}_h \in \boldsymbol{W}_h \\ (\nabla \cdot \boldsymbol{u}_{\Delta t,h}^n, q_h) + (\nabla \cdot \boldsymbol{z}_h^n, q_h) + \frac{c_0}{\alpha}(p_h^n, q_h) + J(p_{\Delta t,h}^n, q_h) = \frac{1}{\alpha}(g^n, q_h), \ \forall q_h \in Q_h \end{cases} \tag{9}$$

where

$$J(p, q) = \delta_{\text{STAB}} \sum_K \int_{\partial K \setminus \partial\Omega} h_{\partial K}[p][q] ds$$

is a stabilizing term [5], and $p_{\Delta t,h}^n = (p_h^n - p_h^{n-1})/\Delta t$. The finite element discretization will lead to a twofold saddle-point problem of the following form:

$$\begin{bmatrix} A_{\boldsymbol{u}} & 0 & B_1^T \\ 0 & A_{\boldsymbol{z}} & B_2^T \\ B_1 & B_2 & -A_p \end{bmatrix} \begin{bmatrix} \boldsymbol{u}_h \\ \boldsymbol{z}_h \\ p_h \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_1 \\ \boldsymbol{f}_2 \\ \boldsymbol{f}_3 \end{bmatrix}. \tag{10}$$

Denote the block matrices of A, B, and the Schur complement S in the following:

$$A = \begin{bmatrix} A_{\boldsymbol{u}} & 0 \\ 0 & A_{\boldsymbol{z}} \end{bmatrix}, \quad B = \begin{bmatrix} B_1 & B_2 \end{bmatrix}, \quad S = -(BA^{-1}B^T + A_p). \tag{11}$$

Usually, for saddle-point problem of the form (10), one takes the preconditioner as a block lower-triangular,

$$T = \begin{bmatrix} A & 0 \\ B & S \end{bmatrix}. \tag{12}$$

3.1 Two-level additive Schwarz algorithm (OAS-2) for $A_{\boldsymbol{u}}$

We now introduce the decomposition into local and coarse spaces. The local problems are defined on the extended subdomains Ω_i'. To each of the Ω_i', we associate a local space

$$\boldsymbol{V}_i = \boldsymbol{V}^h(\Omega_i') \cap \boldsymbol{H}_0^1(\Omega_i'), \tag{13}$$

and a bilinear form $a_i'(\boldsymbol{u}_i, \boldsymbol{v}_i) := a(R_i^T \boldsymbol{u}_i, R_i^T \boldsymbol{v}_i)$, where $R_i^T : \boldsymbol{V}_i \to \boldsymbol{V}^h$, simply extends any element of $\boldsymbol{V}_i$ by zero outside Ω_i'. Then, as we will only consider algorithms for which the local problems are solved exactly, we find that the local operators are

$$A_i' = R_i A R_i^T, \quad i = 1, \ldots, N. \tag{14}$$

Given the local and coarse embedding operators $R_i^T : \boldsymbol{V}_i \to \boldsymbol{V}^h$, $i = 1, \ldots, N$, and $R_0^T : \boldsymbol{V}_0 \to \boldsymbol{V}^h$, the discrete space $\boldsymbol{V}^h$ can be decomposed into coarse and local spaces as

$$\boldsymbol{V}^h = R_0^T \boldsymbol{V}_0 + \sum_i R_i^T \boldsymbol{V}_i. \tag{15}$$

The coarse space on the coarse subdomain mesh τ_H is denoted by

$$\boldsymbol{V}_0 = \boldsymbol{V}^H := \{\boldsymbol{v} \in \boldsymbol{V} : \boldsymbol{v}|_{\Omega_i} \in (\mathbf{P}_1(\Omega_i))^d \ \ \forall \Omega_i \in \tau_H\}. \tag{16}$$

3.2 Construction of coarse spaces by GenEO for A_u

For all subdomains $1 \leq i \leq N$, the generalized eigenvalue problem is to find $(V_{ik}, \lambda_{ik}) \in \text{range}(A_i') \times \mathbb{R}$ such that

$$A_i' V_{ik} = \lambda_{ik} D_i A_i' D_i V_{ik}. \tag{17}$$

where $\{D_i\}_{i=1}^N$ defines a partition of unity, $\sum_{i=1}^N R_i^T D_i R_i = I$. The GenEO coarse space V_0 is based on the following local contributions:

$$Z_{i\tau} = \text{span}\{V_{ik} | \lambda_{ik} < \tau\}, \tag{18}$$

which are weighted with the partition of unity as follows:

$$V_0 = \oplus_{i=1}^N R_i^T D_i Z_{i\tau}. \tag{19}$$

When Z_0 be a column matrix so that V_0 is spanned by its columns, $R_0 = Z_0^T$. Two-level overlapping Schwarz method by GenEO is summarized in Algorithm 1.

Algorithm 1 Two-level overlapping Schwarz method by GenEO [10]

1. Solve the local generalized eigenvalue problem:
   $A_i' y_i = \lambda_i \tilde{R}_i R_i^T A_i' R_i \tilde{R}_i^T y_i$.
   $\{\tilde{R}_i\}_{i=1}^N$ are the same operators as $\{R_i\}_{i=1}^N$ except that entries on the overlap are set to 0 [7].
2. Collect the ν_i smallest eigenpairs $\{y_{i_j}, \lambda_{i_j}\}_{j=1}^{\nu_i}$.
3. Assemble a local deflation dense matrix:
   $W_i = [D_i y_{i_1} \cdots D_i y_{i_{\nu_i}}]$.
   $\{D_i = \tilde{R}_i R_i^T\}_{i=1}^N$ defines the partition of unity.
4. Define a global deflation matrix:
   $P = [R_1^T W_1 \cdots R_N^T W_N]$.
5. Define a two-level preconditioner using the Galerkin product of A and P:
   $M^{-1} = \sum_{i=1}^N \tilde{R}_i^T (R_i A R_i^T)^{-1} R_i$,
   $Q = P(P^T A P)^{-1} P^T$,
   $M_{\text{additive}}^{-1} = Q + M^{-1}$ or $Q + M^{-1}(I - AQ)$.

4 Numerical experiments

A test problem is formulated with $\alpha = 1$, $c_0 = 0$, $\Omega = [0, 1]^2$ and $t \in [0, 0.25]$:

$$\begin{aligned} -(\lambda + \mu)\nabla(\nabla \cdot \boldsymbol{u}) - \mu\nabla^2\boldsymbol{u} + \nabla p &= 0, \\ \mathbf{K}^{-1}\boldsymbol{z} + \nabla p &= 0, \\ \nabla \cdot (\boldsymbol{u}_t + \boldsymbol{z}) &= g_1. \end{aligned} \tag{20}$$

The involving initial and boundary conditions are the following:

$$\begin{cases} \boldsymbol{u} = \mathbf{0} \quad \text{on } \partial\Omega = \Gamma_{\mathrm{d}}, \\ \boldsymbol{z} \cdot \boldsymbol{n} = g_2 \quad \text{on } \partial\Omega = \Gamma_{\mathrm{f}}, \\ \boldsymbol{u}(\boldsymbol{x}, 0) = 0, \boldsymbol{x} \in \Omega, \\ p(\boldsymbol{x}, 0) = 0, \boldsymbol{x} \in \Omega. \end{cases} \tag{21}$$

We consider the following analytic solution

$$\begin{aligned} \boldsymbol{u} &= \frac{-1}{4\pi(\lambda + 2\mu)} \begin{bmatrix} \cos(2\pi x)\sin(2\pi y)\sin(2\pi t) \\ \sin(2\pi x)\cos(2\pi y)\sin(2\pi t) \end{bmatrix}, \\ \boldsymbol{z} &= -2\pi k \begin{bmatrix} \cos(2\pi x)\sin(2\pi y)\sin(2\pi t) \\ \sin(2\pi x)\cos(2\pi y)\sin(2\pi t) \end{bmatrix}, \\ p &= \sin(2\pi x)\sin(2\pi y)\sin(2\pi t), \end{aligned} \tag{22}$$

and derive the compatible source term of g_1.

4.1 Numerical implementation

As the focus of this paper is to justify the effectiveness and the efficiency of the algorithm, we mainly study the performance of the parallel preconditioner discussed as above. In our implementation, we use a fiinite element library, libMesh. We apply triangular element with 3 nodes. The GMRES method and overlapping Schwarz preconditioners are based on PETSc. The initial guess is zero and the stopping criterion is set as a 10^{-8}, reduction of the residual norm. In each test, we count the iterations. For unstructured domain partition, we apply ParMETIS, and DMPlex for overlapping subdomains [12]. In our implementation, $A_{\boldsymbol{u}}$ is approximated by using the two-level additive Schwarz preconditioner from PETSc with PCHPDDM [10] and SLEPc for GenEO.

4.2 Dependency on the subdomain number N

Scalability of GMRES-block triangular preconditioner is tested increasing the number of subdomains N. The subdomain size is set with $H/h = 8$, and the overlapping width is set with $\delta/h = 1$. Two cases are considered, 1) compressible and strongly permeable, 2) almost incompressible and weakly permeable. In the compressible and weakly permeable case, there comes a moderate scalable trend through $N = 49$ and $N = 64$ (Table 1, Left).

Table 1 Left: Scalability of GMRES-block triangular preconditioner. Iteration counts for increasing number of subdomains N. Fixed $H/h = 8$, $\delta/h = 1$, deflation $N = 15$, $r_{\text{tol}} = 10^{-8}$. **Right**: H/δ-dependency of GMRES-block triangular preconditioner. Iteration counts for increased overlapping. Fixed $N = 16$ and $H/h = 64$, $r_{\text{tol}} = 10^{-8}$.

N	$\nu = 0.3$, $\kappa = 10^{-2}$ iteration	$\nu = 0.4999$, $\kappa = 10^{-9}$ iteration
4	4	6
9	5	9
16	7	12
25	7	14
36	9	17
49	11	19
64	11	20

H/δ	$\nu = 0.3$, $\kappa = 10^{-2}$ iteration	$\nu = 0.4999$, $\kappa = 10^{-9}$ iteration
8	52	56
16	59	52
32	60	48
64	78	50

4.3 Dependency on H/δ

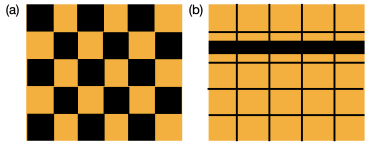
H/δ dependency of GMRES-block triangular preconditioner is tested with the overlap changed from 1 to 8 with the domain size in each dimension $H/h = 64$ and the number of subdomains $N = 16$. The increment of overlap does reduce the GMRES iterations for the compressible and strongly permeable case. However, for the almost incompressible and weakly permeable case, the GMRES iterations increase with the change of H/δ from 32 to 16 and 8 (Table 1, Right).

4.4 Dependency on permeability k

The robustness to κ of the GMRES-block triangular preconditioner is tested with κ changed from 1 to 10^{-9}. Whether compressible or almost incompressible, GMRES iterations show biphasic pattern of decrease from κ=1 to 10^{-3} and increase when κ changes towards $\kappa = 10^{-9}$ (Table 2, Upper). Overall, the numerical scheme of GMRES-block triangular preconditioner show evident robustness in broad ranges of permeability both in compressible and almost incompressible regimes.

4.5 Dependency on heterogeneous material properties of ν and κ

To test the block preconditioner and GenEO two-level OAS solver for the displacement, the primary parameters of Poisson ratio ν and permeability κ are treated as nonuniform. Figure 1(a) is in the pattern of checkboard and jump across subdomains and Figure 1(b) is in the pattern of jump along subdomains. Compressible and strongly permeable poroelasticity (ν=0.3 and $\kappa = 10^{-2}$) is prescribed to yellow regions, and almost incompressible and weakly permeable poroelasticity ($\nu = 0.4999$ and $\kappa = 10^{-9}$) is prescribed to black regions. The proposed GMRES-block triangular preconditioner solver shows robustness to material heterogeneity with finite iterations for both patterns of non-uniformity (Table 2, Lower).



k	$\nu = 0.3$ iteration	$\nu = 0.4999$ iteration
1	51	43
10^{-1}	19	17
10^{-3}	7	9
10^{-5}	14	9
10^{-7}	16	14
10^{-9}	16	47
	($\nu = 0.3$ and $\kappa = 10^{-2}$) vs. ($\nu = 0.4999$ and $\kappa = 10^{-9}$)	
heter. – subdomain	175	
heter. – along	87	

Fig. 1 (a) Jump across subdomains. (b) Jump along subdomains. **Table 2**: Robustness to k of GMRES-block triangular preconditioner. Iteration counts for decreasing permeability. Fixed $N = 16$ and $H/h = 8$, deflation $N = 15$, $r_{\text{tol}} = 10^{-10}$.

5 Conclusion

We proposed domain decomposition preconditioners for the saddle point problem of the three-field Biot model and performed numerical experiments for scalability and robustness in parameters. GMRES with block triangular preconditioner with two-level OAS with coarse space by GenEO for $\boldsymbol{u}$ and LU for $\boldsymbol{z}$ and p is scalable and robust in broad ranges of parameters (ν, κ) and their heterogeneity. Future works are 1) some theoretical analysis on condition number bounds of proposed preconditioned systems, e.g. field-of-value analysis, 2) domain decomposition preconditioners for 3D poroelastic large deformation.

Acknowledgements The author gives thanks to Dr. Mingchao Cai for the introduction of Biot models and invaluable discussions. The author was partly supported by NSF DMS-1831950, and the virtual attendance to DD27 conference by Penn State University NSF travel fund.

References

1. Berger, L., Bordas, R., Kay, D., and Tavener, S. Stabilized lowest-order finite element approximation for linear three-field poroelasticity. *SIAM J Sci Comput* **37**, A2222–A2245 (2015).
2. Biot, M. A. General theory of three-dimensional consolidation. *J Appl Phys* **12**, 155–164 (1941).
3. Biot, M. A. Theory of elasticity and consolidation for a porous anisotropic solid. *J Appl Phys* **26**, 182–185 (1955).
4. Bjorstad, P. and Krzyzanowski, P. A flexible 2-level neumann-neumann method for structural analysis problems. In: Wyrzykowski, R., Dongarra, J., Paprzycki, M., and Waśniewski, J. (eds.), *International Conference on Parallel Processing and Applied Mathematics*, *Lecture Notes in Computer Science*, vol. 2328, 387–394. Parallel Processing and Applied Mathematics, Springer, Berlin, Heidelberg (2001).
5. Burman, E. and Hansbo, P. A unified stabilized method for Stokes' and Darcy's equations. *J Comput Appl Math* **198**, 35–51 (2007).
6. Cai, M., Pavarino, L., and Widlund, O. Overlapping Schwarz methods with a standard coarse spaces for almost incompressible linear elasticity. *SIAM J Sci Comput* **37**(2), A811–A831 (2015).
7. Cai, X.-C. and Sarkis, M. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J Sci Comput* **21**, 792–797 (1999).
8. Galvis, J. and Efendiev, Y. Domain decomposition preconditioners for multiscale flows in high-contrast media. *Multiscale Model Simul* **8**, 1461–1483 (2010).
9. Heinlein, A., Klawonn, A., and Rheinbach, O. A parallel implementation of a two-level overlapping Schwarz method with energy-minimizing coarse space based on Trilinos. *SIAM J Sci Comput* **38**, C713–C747 (2016).
10. Jolivet, P., Roman, J., and Zampini, S. KSPHPDDM and PCHPDDM: extending PETSc with advanced Krylov methods and robust multilevel overlapping Schwarz preconditioners. *Comput Math with Appl* **84**, 277–295 (2021).
11. Klawonn, A. and Pavarino, L. F. Overlapping Schwarz methods for mixed linear elasticity and Stokes problems. *Comput Methods Appl Mech Engrg* **165**(1), 233–245 (1998).
12. Knepley, M. and Karpeev, D. Mesh algorithms for PDE with Sieve I: mesh distribution. *Scientific Programming* **17**, 215–230 (2009).
13. Pavarino, L. F. Indefinite overlapping Schwarz methods for time-dependent Stokes problems. *Comput Methods Appl Mech Engrg* **187**, 35–51 (2000).
14. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., and Scheichl, R. A robust two-level domain decomposition preconditioner for systems of PDEs. *C R Acad Sci Paris, Ser. I* **349**, 1255–1259 (2011).
15. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., and Scheichl, R. Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. *Numer Math* **126**, 741–770 (2014).

Adaptive Schwarz Method for a Non-Conforming Crouzeix-Raviart Discretization of a Multiscale Elliptic Problem

Leszek Marcinkowski and Talal Rahman

1 Introduction

In many physical or engineering practical applications, we see a heterogeneity of coefficients; e.g., in ground flow problems in heterogeneous media. We also see that many models of those phenomena are differential ones, i.e. the physical phenomenon is modeled by partial differential equations. Then, if those PDEs models are discretized by a finite element method, one gets the discrete system which is quite often very hard to solve by standard iterative methods without a proper precondition; see, e.g., [21].

The Domain Decomposition Methods (DDMs) approach, in particular, Schwarz methods; see, e.g., [24], allow us to construct a large class of parallel and effective preconditioners. A very important role in such construction is taken by a carefully defined coarse space. The classical DDMs constructed in the 1990s and 2000s are well suited only for problems with coefficients that are constant or slightly varying in subdomains. However, those 'classical' methods are not effective when the coefficients may be highly varying and discontinuous almost everywhere. Since the classical coarse spaces of Schwarz methods do not give us efficient and robust solvers for multiscale problems with heterogeneous coefficients we will propose a way of enrichment of the coarse spaces which made DDMs effective for heterogeneous problems. That gives us new adaptive coarse spaces which are independent or robust for the jumps of the coefficients, i.e., the convergence of the constructed DDM is independent of the distribution and the magnitude of the coefficients of the original

Leszek Marcinkowski
Faculty of Mathematics, University of Warsaw, Banacha 2, 02-097 Warszawa, Poland e-mail: Leszek.Marcinkowski@mimuw.edu.pl

Talal Rahman
Faculty of Engineering and Science, Western Norway University of Applied Sciences, Inndalsveien 28, 5063 Bergen, Norway e-mail: Talal.Rahman@hvl.no

problem. We refer to [9], [23] and the references therein for similar earlier works on domain decomposition methods used adaptively in the construction of coarse spaces.

In recent years there appeared many new research results on this topic; see, e.g., [3, 4, 5, 6, 7, 8, 10, 12, 13, 14, 15, 16, 17, 18, 19, 20] and many others.

In our paper, we consider a minimal overlap Schwarz method for the nonconforming Crouzeix-Raviart (CR) element discretization, also called the nonconforming P_1 element discretization; see, e.g., [1]. We extend the results from [10] where the conforming P_1 element is considered to the case of the CR non-conforming discretization applied to highly heterogeneous coefficients.

The remainder of the paper is organized as follows: in Section 2 we introduce our differential problem and its CR discretization. In Section 3 a classical overlapping Additive Schwarz method is presented and the theoretical bound for the condition number of the resulting system is given.

2 Discrete problem

Our model differential problem is the following elliptic second order boundary value problem: Find $u^* \in H_0^1(\Omega)$ such that

$$\int_\Omega \alpha(x)\nabla u^* \nabla v \, dx = \int_\Omega f v \, dx, \qquad \forall v \in H_0^1(\Omega),$$

where Ω is a polygon in $\mathbb{R}^2$, $0 < \alpha_0 \leq \alpha(x) \leq \alpha_1$ is a coefficient, α_0, α_1 are positive constant, and $f \in L^2(\Omega)$.

We need a quasi-uniform triangulation $\mathcal{T}_h = \{K\}$ of Ω consisting of open triangles such that $\overline{\Omega} = \bigcup_{K \in \mathcal{T}_h} \overline{K}$. Let further, h_K be the diameter of $K \in \mathcal{T}_h$, and we define $h = \max_{K \in \mathcal{T}_h} h_K$ as the triangulation diameter.

We also introduce a coarse non-overlapping partitioning of Ω (see, Fig. 1) into open, connected Lipschitz polygonal subdomains (substructures) Ω_i such that

$$\overline{\Omega} = \bigcup_{i=1}^{N} \overline{\Omega}_i,$$

which are aligned to the fine triangulation, i.e. we have that any fine triangle $K \in \mathcal{T}_h$ is contained in a coarse substructure Ω_k. Thus each substructure Ω_j has its local triangulation $\mathcal{T}_h(\Omega_j)$ of triangles from $\mathcal{T}_h$ which are contained in $\overline{\Omega}_j$. For the simplicity of presentation, we further assume that these substructures form a coarse triangulation of the domain which is shape-regular in the sense of [2] and let $H = \max_j \operatorname{diam}(\Omega_j)$ be its coarse parameter. Let Γ_{ij} denote the open edge common to subdomains Ω_i and Ω_j not in $\partial\Omega$ and let Γ be the union of all $\partial\Omega_k \setminus \partial\Omega$.

However, it is good to note that the theory of this paper holds also for the case when the coarse partition is obtained by a mesh partitioner. Then naturally an edge

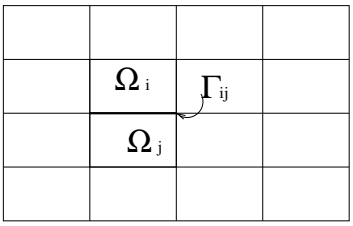


Fig. 1 An example of a coarse partition of Ω, where Γ_{ij} is an edge on the interface.

(interface) Γ_{ij} is not a straight segment but a 1D curve made of respective edges of some fine triangles.

Let Ω_h^{CR}, $\partial\Omega_h^{CR}$, $\Omega_{i,h}^{CR}$, $\partial\Omega_{i,h}^{CR}$, and $\Gamma_{ij,h}^{CR}$ be defined as the sets of midpoints of fine edges of the elements of $\mathcal{T}_h$, contained in Ω, $\partial\Omega$, Ω_i, $\partial\Omega_i$, and Γ_{ij}, respectively. We call those sets the CR nodal points of the respective sets.

The discrete solution space is the Crouzeix-Raviart finite element space, (see, e.g., [1]), or nonconforming P_1 element space defined as:

$$V_h(\Omega) = V_h = \{v \in L^2(\Omega) : v_{|K} \in P_1(K),\ v \text{ continuous at } \Omega_h^{CR},\\ v(m) = 0 \quad m \in \partial\Omega_h^{CR}\},$$

where $P_1(K)$ is the space of linear polynomials defined on K.

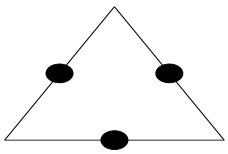


Fig. 2 The CR nodal points, i.e., the degrees of freedom of the Crouzeix-Raviart finite element space on a fine triangle.

The degrees of freedom of a CR finite element function u on a triangle K with the three edges e_k $k = 1, 2, 3$, are: $\{u(m_{e_k})\}_{k=1,2,3}$, where m_{e_k} is the midpoint of the fine edge e_k; see, Fig. 2. Note that a function in V_h is multivalued on boundaries of all fine triangles of $\mathcal{T}_h$ except the midpoints of the edges (CR nodal points). Thus $V_h \not\subset H_0^1(\Omega)$ is a space of discontinuous functions. V_h is only a subspace of $L^2(\Omega)$, and in this lies the non-conformity of this discretization.

We introduce the following Crouzeix-Raviart discrete problems: find $u_h^* \in V_h$ such that :

$$a_h(u_h^*.v) = f(v) \qquad \forall v \in V_h, \tag{1}$$

where the broken bilinear form $a_h : (V_h \cup H_0^1(\Omega)) \times (V_h \cup H_0^1(\Omega)) \to \mathbb{R}$ is defined as $a_h(u, v) = \sum_{K\in\mathcal{T}_h} \int_K \alpha_{|K}(x)\nabla u \nabla v\, dx$. It is easy to see that the broken form is V_h elliptic; see, e.g., [1], and we see that our discrete problem has a unique solution.

We see that ∇u_h for $u_h \in V_h$ is constant over any fine triangle $K \in \mathcal{T}_h$, thus

$$\int_K \alpha \nabla u \nabla v \, dx = (\nabla u)_{|K} (\nabla v)_{|K} \int_K \alpha(x) \, dx.$$

Hence, we can further assume that α is piecewise constant function over the elements of $\mathcal{T}_h$.

3 Additive Schwarz method (ASM)

In this section, we present our Schwarz method for solving (1) which is based on the abstract Additive Schwarz Method framework; see, e.g., [24]. Our method is of minimal overlap, however, the same estimates hold if we introduce a more generous overlap. In the abstract scheme of ASM one has to introduce a decomposition of the discrete space into subspaces, usually, a coarse space and local subspaces. We also need local bilinear forms defined on those subspaces respectively. In our case for simplicity of presentation, all bilinear forms are taken as equal to the original broken-form $a_h(u, v)$.

The local spaces are defined as:

$$V_i = \{v \in V_h : v(m) = 0 \quad m \notin \overline{\Omega}_{i,h}^{CR}\},$$

i.e. V_i is formed by all discrete CR FEM functions which are zero at all CR nodes not in $\overline{\Omega}_i$. Thus, it is a minimal overlap subspace since a function $u \in V_i$ can be nonzero on $\overline{\Omega}_i$ and the fine triangles which have an edge on the boundary of Ω_i. We see that $V_h = \sum_{i=1}^N V_i$.

In our case, the coarse space will be a harmonically enriched CR version of the multiscale coarse space introduced in [11] for standard conforming linear finite element space. Let $\mathcal{T}_h(\Omega_k)$ be a local triangulation of Ω_k inherited from $\mathcal{T}_h$. We now

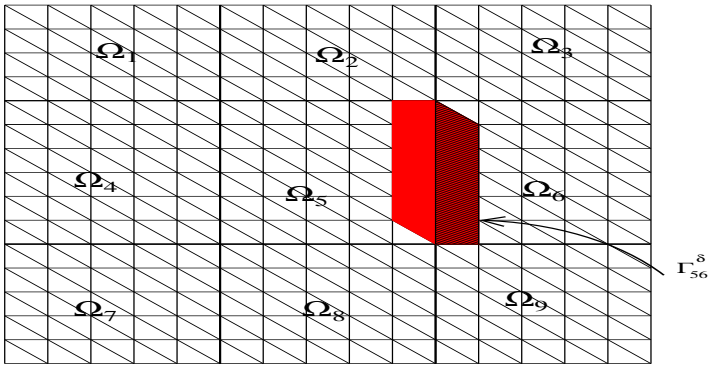


Fig. 3 An edge patch.

introduce a patch around a coarse interface Γ_{kl} the common edge of Ω_k, Ω_l. We define $\overline{\Gamma}^{\delta}_{kl}$ as the closure of the boundary patch around Γ_{kl} the union of all closed fine triangles, such that each fine triangle of the patch has a vertex on Γ_{kl}. The open patch Γ^{δ}_{kl} is then defined as the interior of $\overline{\Gamma}^{\delta}_{kl}$; see, Fig. 3.

For simplicity of presentation, we assume that if two edges Γ_{kl}, Γ_{kj} which have a common vertex (crosspoint - a common vertex of $\Omega_k, \Omega_j, \Omega_l$) then the patches $\Gamma^{\delta}_{kl}, \Gamma^{\delta}_{kj}$ are disjoint. Each patch Γ^{δ}_{kl} can be split into two subpatches – the respective subsets contained in one of two subdomains:

$$\Gamma^{\delta,i}_{kl} = \Gamma^{\delta}_{kl} \cap \Omega_i, \quad i = k, l.$$

Naturally, we have that $\overline{\Gamma^{\delta}_{kl}} = \overline{\Gamma^{\delta,l}_{kl} \cup \Gamma^{\delta,k}_{kl}}$. We next introduce the interior boundary layer of Ω_k:

$$\Omega^{in,\delta}_k = \bigcup_{\Gamma_{kl} \subset \partial\Omega_k \cap \Gamma} \Gamma^{\delta,k}_{kl}.$$

We also define the local subspaces: let $V_{h,k}$ be formed by the restrictions to $\overline{\Omega}_k$ of the functions from V_h, i.e.

$$V_{h,k} = \{v \in L^2(\Omega_k) : v_{|K} \in P_1(K), K \in \mathcal{T}_h(\Omega_k), \\ v - \text{continuous at CR nodes}, v_{|\partial\Omega^{CR}_h} = 0\}$$

Let $V^0_{h,k} \subset V_{h,k}$ be space of functions that are zero at $\partial\Omega^{CR}_{k,h}$ and at the CR nodes in the interior $\Omega^{in,\delta}_k$. Any $u \in V^0_{h,k}$ can be extended by zero to the whole Ω and we will further identify $V^0_{h,k}$ with the subspace of V_h formed by such zero extensions of functions in this local space.

Let $\mathcal{P}_k : V^h \to V^0_{h,k}$ be the orthogonal projection:

$$a_{k,h}(\mathcal{P}_k u, v) = a_h(\mathcal{P}_k u, v) = a_h(u, v) \quad \forall v \in V^0_{h,k}, \tag{2}$$

where $a_{k,h}(u, v)$ is the local bilinear form defined as the restriction of the broken form to Ω_k. Let $\mathcal{P}u = \sum_{k=1}^{N} \mathcal{P}_k u$, ($\mathcal{P}_k u$ extended by zero to $\overline{\Omega}$). Then the discrete harmonic operator is set as $\mathcal{H} = I - \mathcal{P}$ and we say that $u \in V_h$ is discrete harmonic if:

$$u = \mathcal{H}u. \tag{3}$$

Next, we need to set a local edge related space $V_{kl} \subset V_h$:

$$V_{kl} = \{v \in V_h : v(m) = 0 \quad m \notin \overline{\Gamma}^{\delta,CR}_{kl,h}\}.$$

The support of any function $u \in V_{kl}$ is not contained in the patch Γ^{δ}_{kl}.

We also need a subspace of V^v_{kl} defined as:

$$V^v_{kl} = \{v \in V_{kl} : v(m) = 0 \quad m \in \mathcal{V}(\Gamma_{kl})\},$$

where $\mathcal{V}(\Gamma_{kl}) \subset \Gamma_{kl,h}^{CR}$ comprise the two CR nodes of the edge which are next to the ends of this edge.

Let $V_0^{msc} \subset V_h$ be the multiscale part of the coarse space (analogous to the one in [11]), i.e., the space of discrete harmonic functions; see, (3), which satisfy

$$a_{kl,h}(u, v) = 0 \quad \forall v \in V_{kl}^v, \tag{4}$$

where $a_{kl,h}(u, v) = \sum_{K \subset \Gamma_{kl}^\delta} \int_K \alpha_{|K} \nabla u \nabla v \, dx$ for any edge $\Gamma_{kl} \subset \Gamma$.

Let us introduce the local generalized eigenvalue problem, which is to find all eigenpairs: $(\lambda_i^{kl}, \psi_i^{kl}) \in \mathbb{R}_+ \times V_{kl}^v$ such that

$$a_{kl,h}(\psi_i^{kl}, v) = \lambda_i^{kl} b_{kl}(\psi_i^{kl}, v), \qquad \forall v \in V_{kl}^v, \tag{5}$$

where $b_{kl}(u, v) = h^{-2} \int_{\Gamma_{kl}^\delta} \alpha u v \, dx$.

Any eigenfunction ψ_j^{kl} can be extended further onto other patches as zero and then, further to the interiors of all subdomains as a discrete harmonic function. Then, we will further denote it by Ψ_j^{kl}. We can number the eigenvalues in increasing order: $0 < \lambda_1^{kl} \leq \lambda_2^{kl} \leq \ldots \leq \lambda_{M_{kl}}^{kl}$ for $M_{kl} = \dim(V_{kl}^v)$. Next, we introduce the local spectral component of the coarse space for all Ω_j:

$$V_{kl}^{eig} = \text{Span}(\Psi_i^{kl})_{i=1}^{n_{kl}}, \tag{6}$$

where $0 \leq n_{kl} \leq M_{kl}$ can be pre-selected by the user. It can be decided using the experience or by some rule; e.g., one can include all eigenfunctions for which related eigenvalues are below a certain threshold. The coarse space V_0 is introduced as:

$$V_0 = V_0^{msc} + \sum_{\Gamma_{kl} \subset \Gamma}^{N} V_{kl}^{eig}.$$

Next, we define the projection operators $T_i : V_h \to V_i$ for $i = 0, \ldots, N$ as

$$a_h(T_i u, v) = a_h(u, v), \quad \forall v \in V_i;$$

see, e.g., [22]. Note that to compute the $T_i u$, $i = 1, \ldots, N$ we have to solve N independent local problems.

Let $T := \sum_{i=0}^N T_i$, be the additive Schwarz operator; see, e.g., [22]. We further replace (1) by the following equivalent problem: Find $u_h^* \in V_h$ such that

$$T u_h^* = g,$$

where $g = \sum_{i=0}^N g_i$ and $g_i = T_i u_h^*$. The functions g_i may be computed without knowing the solution u_h^* of (1); see, e.g., [22].

The following theoretical estimate of the condition number can be obtained:

Theorem 1 *For all $u \in V_h$, the following holds,*

$$c\left(1+\max_{\Gamma_{kl}\subset\Gamma}\left(\lambda^{kl}_{n_{kl}+1}\right)^{-1}\right)^{-1} a_h(u,u)\le a_h(Tu,u)\le C\, a_h(u,u),$$

where C and c are positive constants independent of the coefficient α, the mesh parameter h and the subdomain size H, and $\lambda^{kl}_{n_{kl}+1}$ is defined in (5) for both types of the coarse space.

Below we give a very brief sketch of the proof, which is based on the standard abstract ASM Method framework; see, [24]. We have to prove three key assumptions, the most technical is the stable splitting ass., namely, we show that for any $u \in V_h$ there exists: $u_j \in V_j$ $j = 0,\dots,N$ such that $\sum_{j=0}^{N} a_h(u_j,u_j) \le c^{-1}\left(1+\max_{\Gamma_{kl}}\left(\lambda^{kl}_{n_{kl}+1}\right)^{-1}\right) a(u,u)$. The two others assumptions are easy to verify.

References

1. Brenner, S. C. and Scott, L. R. *The mathematical theory of finite element methods*, *Texts in Applied Mathematics*, vol. 15. Springer, New York, third ed. (2008).
2. Brenner, S. C. and Sung, L.-Y. Balancing domain decomposition for nonconforming plate elements. *Numer. Math.* **83**(1), 25–52 (1999).
3. Calvo, J. G. and Widlund, O. B. An adaptive choice of primal constraints for BDDC domain decomposition algorithms. *Electron. Trans. Numer. Anal.* **45**, 524–544 (2016).
4. Chartier, T., Falgout, R. D., Henson, V. E., Jones, J., Manteuffel, T., McCormick, S., Ruge, J., and Vassilevski, P. S. Spectral AMGe (ρAMGe). *SIAM J. Sci. Comput.* **25**(1), 1–26 (2003).
5. Efendiev, Y., Galvis, J., Lazarov, R., Margenov, S., and Ren, J. Robust two-level domain decomposition preconditioners for high-contrast anisotropic flows in multiscale media. *Comput. Methods Appl. Math.* **12**(4), 415–436 (2012).
6. Efendiev, Y., Galvis, J., Lazarov, R., and Willems, J. Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms. *ESAIM Math. Mod. Num. Anal.* **46**, 1175–1199 (2012).
7. Eikeland, E., Marcinkowski, L., and Rahman, T. Overlapping Schwarz methods with adaptive coarse spaces for multiscale problems in 3D. *Numer. Math.* **142**(1), 103–128 (2019).
8. Eikeland, E., Marcinkowski, L., and Rahman, T. An adaptively enriched coarse space for Schwarz preconditioners for P_1 discontinuous Galerkin multiscale finite element problems. *IMA J. Numer. Anal.* **41**(4), 2873–2895 (2021).
9. Galvis, J. and Efendiev, Y. Domain decomposition preconditioners for multiscale flows in high-contrast media. *Multiscale Model. Simul.* **8**(4), 1461–1483 (2010).
10. Gander, M. J., Loneland, A., and Rahman, T. Analysis of a new harmonically enriched multiscale coarse space for domain decompostion methods. Eprint arXiv:1512.05285 (2015).
11. Graham, I. G., Lechner, P. O., and Scheichl, R. Domain decomposition for multiscale PDEs. *Numer. Math.* **106**(4), 589–626 (2007).
12. Heinlein, A., Klawonn, A., Knepper, J., and Rheinbach, O. Multiscale coarse spaces for overlapping Schwarz methods based on the ACMS space in 2D. *Electron. Trans. Numer. Anal.* **48**, 156–182 (2018).
13. Heinlein, A., Klawonn, A., Knepper, J., and Rheinbach, O. Adaptive GDSW coarse spaces for overlapping Schwarz methods in three dimensions. *SIAM J. Sci. Comput.* **41**(5), A3045–A3072 (2019).

14. Heinlein, A., Klawonn, A., Knepper, J., Rheinbach, O., and Widlund, O. B. Adaptive GDSW coarse spaces of reduced dimension for overlapping Schwarz methods. *SIAM J. Sci. Comput.* **44**(3), A1176–A1204 (2022).
15. Kim, H. H., Chung, E., and Wang, J. BDDC and FETI-DP preconditioners with adaptive coarse spaces for three-dimensional elliptic problems with oscillatory and high contrast coefficients. *J. Comput. Phys.* **349**, 191–214 (2017).
16. Klawonn, A., Radtke, P., and Rheinbach, O. FETI-DP methods with an adaptive coarse space. *SIAM J. Numer. Anal.* **53**(1), 297–320 (2015).
17. Klawonn, A., Radtke, P., and Rheinbach, O. A comparison of adaptive coarse spaces for iterative substructuring in two dimensions. *Electronic Transactions on Numerical Analysis* **45**, 75–106 (2016).
18. Mandel, J. and Sousedík, B. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Comput. Methods Appl. Mech. Engrg.* **196**(8), 1389–1399 (2007).
19. Nataf, F., Xiang, H., and Dolean, V. A two level domain decomposition preconditioner based on local Dirichlet-to-Neumann maps. *C. R. Math. Acad. Sci. Paris* **348**(21-22), 1163–1167 (2010).
20. Nataf, F., Xiang, H., Dolean, V., and Spillane, N. A coarse space construction based on local Dirichlet-to-Neumann maps. *SIAM J. Sci. Comput.* **33**(4), 1623–1642 (2011).
21. Saad, Y. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second ed. (2003).
22. Smith, B. F., Bjørstad, P. E., and Gropp, W. D. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, Cambridge (1996).
23. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., and Scheichl, R. Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. *Numer. Math.* **126**, 741–770 (2014).
24. Toselli, A. and Widlund, O. *Domain decomposition methods—algorithms and theory*, *Springer Series in Computational Mathematics*, vol. 34. Springer-Verlag, Berlin (2005).

A Variational-Based Multirate Time-Integrator for FETI and Structural Dynamics: Lagrange-Multiplier with Micro-Discretization

Andreas S. Seibold and Daniel J. Rixen

1 Introduction

The FETI-method is well known for its scalability and applicability to nonlinear structural dynamics [3, 4]. In case of models with different fast dynamics, the classical FETI-method with common time-discretizations can become inefficient, as the subdomain with slow dynamics has to be solved more often, than necessary. The PH-method [8] and BGC-macro-method [1] enable subcycling of a macro-time-discretization, but suffer from spurious oscillations and are not variational methods. In literature, a variational framework for multiple time-discretizations has been introduced [7]. In this work, we further extend this approach to a micro-discretization. In section 2, the FETI-method and nonlinear BGC-macro method are introduced. In section 3, the variational-based multirate method is derived with its modifications and in section 4 both methods are compared in numerical experiments.

2 Nonlinear BGC-macro method for the FETI-method

2.1 Model problem and FETI-method

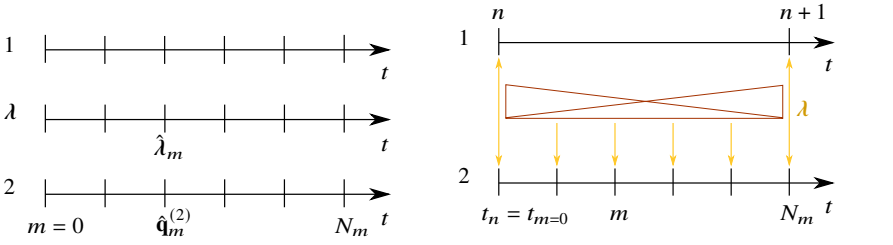
The dynamic behavior over time of a solid elastic body with nonlinear material can be modeled by a nonlinear hyperbolic partial differential equation (PDE). For the solution of such a hyperbolic PDE, consider a geometrical discretization with the Finite Element method and the Finite Element Tearing and Interconnecting (FETI) for the spacial non-overlapping domain decomposition into N_s subdomains. Hence,

Andreas S. Seibold, Daniel J. Rixen
Chair of Applied Mechanics, TUM School of Engineering and Design, Technical University of Munich, Boltzmannstr. 15 D-85748 Garching, e-mail: andreas.seibold@tum.de, rixen@tum.de

the spacially discretized, time-continuous differential equation of motion of a subdomain s and the compatibility condition for velocities are

$$\mathbf{M}^{(s)}\ddot{\mathbf{q}}^{(s)} + \mathbf{f}_{int}(\mathbf{q}^{(s)}) + \mathbf{B}^{(s)^T}\boldsymbol{\lambda} - \mathbf{f}_{ext}^{(s)}(t) = \mathbf{0}, \quad \sum_{s=1}^{N_s} \mathbf{B}^{(s)}\dot{\mathbf{q}}^{(s)} = \mathbf{0}.$$

Here, $\mathbf{q}^{(s)}$ describes the nodal displacements and its time-derivatives $\dot{\mathbf{q}}^{(s)}$ and $\ddot{\mathbf{q}}^{(s)}$ are the velocities and accelerations. $\mathbf{M}^{(s)}$ is the mass-matrix, $\mathbf{f}_{int}$ are the nonlinear internal forces and $\mathbf{f}_{ext}^{(s)}$ are the external forces of the subdomain. The dual quantity or interface-force is described by $\boldsymbol{\lambda}$ and $\mathbf{B}^{(s)}$ is a signed boolean matrix mapping the subdomain's geometrical degrees of freedom (dof) to interface-dofs. The unknowns $\mathbf{q}^{(s)}$, its derivatives and $\boldsymbol{\lambda}$ are discretized in time, with a common time-step-size Δt and to time-nodal values $\hat{\mathbf{q}}_m^{(s)}$ and $\hat{\boldsymbol{\lambda}}_m$ at a time-step m, as depicted in Fig. 1a. For the time-stepping from time-step m to $m+1$, a time-integration scheme is applied, such as the Newmark-β scheme.



(a) Time-discretization in nodal values at time-steps m.

(b) Subcycling of the time-discretization of two subdomains.

Fig. 1 Time-discretizations for two exemplary subdomains 1 and 2 and the Lagrange-multipliers.

2.2 Multirate with nonlinear BGC-macro method

Having different time-step-sizes in the FETI-method can be achieved by the BGC-macro method [1], later adapted for nonlinear problems and FETI [9]. The time-discretization on the subdomain with the smaller time-step-size, also referred to as the micro-discretization, subcycles subdomains with a larger time-step-size, the macro-discretization, as depicted in Fig. 1b. The Lagrange-multipliers are discretized with the macro-discretization and interpolated linearly on the micro-discretized subdomain. Hence, the dynamic equation of motion and the compatibility condition, which is enforced at the macro-discretization, follow as

$$\mathbf{M}^{(s)}\ddot{\hat{\mathbf{q}}}_m^{(s)} + \mathbf{f}_{int}(\hat{\mathbf{q}}_m^{(s)}) + \mathbf{B}^{(s)^T}\boldsymbol{\lambda}_m - \mathbf{f}_{ext}^{(s)}(t_m) = \mathbf{0} \qquad \sum_{s=1}^{N_s} \mathbf{B}\dot{\hat{\mathbf{q}}}_n^s = \mathbf{0}$$

with the interpolated Lagrange-multiplier

$$\boldsymbol{\lambda}_m = \frac{t_{n+1} - t_m}{t_{n+1} - t_n}\boldsymbol{\lambda}_n + \frac{t_m - t_n}{t_{n+1} - t_n}\boldsymbol{\lambda}_{n+1}.$$

3 Variational multirate method with micro discretization of the dual field

The equation of motion (1) and the well-known Newmark-β time-integration scheme can also be derived from the variational principle for $\gamma = 0.5$ and $\beta = 0.25$, as shown by Kane e.a. [5]. We define the time-continuous kinetic energy of a subdomain as $\mathcal{T} = \frac{1}{2}\dot{\mathbf{q}}^{(s)^T}\mathbf{M}\dot{\mathbf{q}}^{(s)}$, the nonlinear potential energy $\mathcal{V}(\mathbf{q}^{(s)})$ and the interface-energy $\mathcal{G} = \mathbf{g}(\mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(N_s)})^T\boldsymbol{\lambda}$ with the gap on interface $\mathbf{g}$, corresponding to the Lagrange-multipliers $\boldsymbol{\lambda}$. In case of the FETI-method this gap is $\mathbf{g}(\mathbf{q}^{(1)}, \ldots, \mathbf{q}^{(N_s)}) = \sum_{s=1}^{N_s}\mathbf{B}^{(s)}\mathbf{q}^{(s)}$, which has not been explicitly specified in literature [7]. The Lagrangian then follows as

$$\mathcal{L}(\dot{\mathbf{q}}^{(1)}, \mathbf{q}^{(1)}, \ldots, \dot{\mathbf{q}}^{(N_s)}, \mathbf{q}^{(N_s)}, \boldsymbol{\lambda}) = \sum_{s=1}^{N_s}\left(\mathcal{T}(\dot{\mathbf{q}}^{(s)}) - \mathcal{V}(\mathbf{q}^{(s)})\right) + \mathcal{G}. \tag{1}$$

According to Hamilton's principle, the mechanical system will move such that the action integral of this Lagrangian is stationary. Hence, we first discretize the Lagrangian in time with time-shape-functions $\Phi^{(s)}(t)$ and $\Theta(t)$, that fulfill partition of unity, we can approximate displacements, velocities and Lagrange-multipliers as

$$\mathbf{q}^{(s)}(t) \approx \sum_{m=0}^{N_m^{(s)}}\Phi_m^{(s)}(t)\hat{\mathbf{q}}_m^{(s)}, \quad \dot{\mathbf{q}}^{(s)}(t) \approx \sum_{m=0}^{N_m^{(s)}}\frac{d\Phi_m^{(s)}(t)}{dt}\hat{\mathbf{q}}_m^{(s)}, \quad \boldsymbol{\lambda}(t) \approx \sum_{j=0}^{N_j}\Theta_j\hat{\boldsymbol{\lambda}}_j.$$

Throughout this paper, we assume linear time-shape-functions. This results in the discrete Lagrangian

$$\mathcal{L}_d(\hat{\mathbf{q}}_0^{(s)}, \ldots, \hat{\mathbf{q}}_{N_m}^{(s)}, \hat{\boldsymbol{\lambda}}_0, \ldots, \hat{\boldsymbol{\lambda}}_{N_j}, t) = \sum_{s=1}^{N_s}\left(\mathcal{T}(\hat{\mathbf{q}}^{(s)}, t) - \mathcal{V}(\hat{\mathbf{q}}^{(s)}, t)\right) + \mathcal{G}(t) \tag{2}$$

which is then integrated with a numerical quadrature rule, such as the generalized midpoint-rule, to the discrete action integral

$$\mathcal{S}_d = \sum_{k=0}^{N_k}\Delta t_k\,\mathcal{L}_d(t_{k+\alpha}), \tag{3}$$

where we have N_k common integration-segments and $\mathcal{L}_d$ is evaluated at a generalized mid-point of these segments $t_{k+\alpha}$. This discrete action integral has to remain

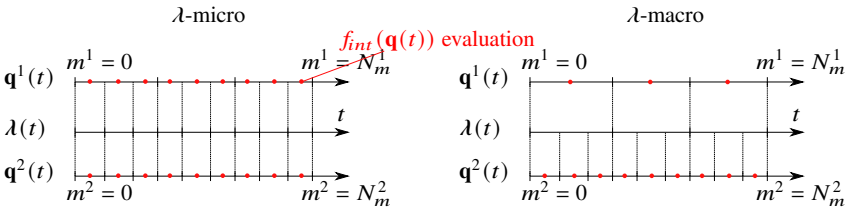
stationary $\sum_{s=1}^{N_s}\sum_{m=0}^{N_m^{(s)}}\frac{\partial \mathcal{S}_d}{\partial \hat{\mathbf{q}}_m^{(s)}}\delta\hat{\mathbf{q}}_m^{(s)}+\sum_{j=0}^{N_j}\frac{\partial \mathcal{S}_d}{\partial \hat{\boldsymbol{\lambda}}_j}\delta\hat{\boldsymbol{\lambda}}_j=\mathbf{0}$ for arbitrary variations of time-nodal quantities, while the endpoints $\delta\hat{\mathbf{q}}_0^{(s)}$ and $\delta\hat{\mathbf{q}}_{N_m}^{(s)}$ remain fixed. This way, we also obtain a local variational integration scheme, such as the non-dissipative Newmark-β method and a variational coupling condition. A variational method comes with some beneficial properties by design, such as symplecticity, conservation of momentum and energy-oscillations remain bounded [6]. We could now solve this problem with a Newton-Raphson scheme and solve the Lagrange-multipliers at each Newton-iteration with a FETI-solver. However, in general, all equations have to be solved at once and a more memory-efficient time-stepping can only be applied on the subdomain-level [9]. The constraint equation for Lagrange-multiplier j

$$\frac{\partial \mathcal{S}_d}{\partial \hat{\boldsymbol{\lambda}}_j}=\sum_{k=0}^{N_k}\Delta t_k\Theta_j(t_{k+\alpha})\sum_{s=1}^{N_s}\mathbf{B}^{(s)}\sum_{m=0}^{N_m^{(s)}}\Phi_m^{(s)}(t_{k+\alpha})\hat{\mathbf{q}}_m^{(s)}=\mathbf{0},$$

is a constraint for several time-nodal displacements $\hat{\mathbf{q}}_m^{(s)}$. Hence, In the following sections 3.1 and 3.2, we introduce special cases and some modifications to this variational method, to still enable time-stepping.

3.1 Downsampling of Lagrange-multipliers

The quadrature (3) suggests the evaluation of the discrete Lagrangian is performed at each time-step $t_{k+\alpha}$ regardless of each subdomain's time-discretization and therefore the evaluation of the nonlinear potential energy derivative $\frac{\partial \mathcal{V}}{\partial \mathbf{q}}$. Hence, in terms of computational efficiency, one could as well choose a micro-discretization in all subdomains. In the following, we consider a subcycled time-discretization on all subdomains and Lagrange-multipliers. If the macro-discretization is chosen for the Lagrange-multiplier, one can just evaluate at the local time-step's midpoint, as depicted in Fig. 2b to properly integrate the Lagrangian. This high number of evaluations is especially needed if the time-discretization of the Lagrange-multiplier is chosen as a micro-discretization, as shown in Fig. 2a. For such cases, we introduce a downsampling of the Lagrange-multiplier by inserting an additional local Lagrange-multiplier field $\bar{\boldsymbol{\lambda}}^{(s)}$, as depicted in Fig. 3. With an artificial



(a) Evaluation-points for micro-discretization of the Lagrange-multiplier.

(b) Evaluation-points for macro-discretization of the Lagrange-multiplier.

Fig. 2 Subcycling time-discretization of two subdomains and evaluation-points for quadrature.

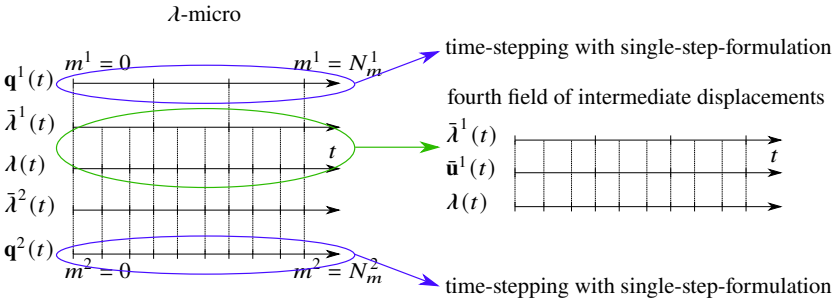


Fig. 3 Additional local Lagrange-multiplier-field and artificial displacement-field for local downsampling.

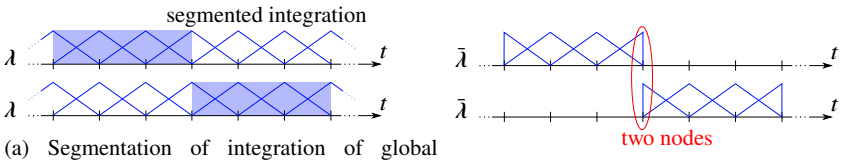
displacement-field $\bar{\mathbf{u}}^{(s)}$, connecting both Lagrange-multiplier fields, we can reformulate the constraint-energy $\mathcal{G} = \left(\sum_{s=1}^{N_s} \mathbf{B}^{(s)} \sum_{m=0}^{N_m^{(s)}} \Phi_m^{(s)}(t)\bar{\mathbf{u}}_m^{(s)}\right)^T \sum_{j=0}^{N_j} \Theta_j(t)\hat{\boldsymbol{\lambda}}_j + \sum_{s=1}^{N_s}\left(\left(\sum_{m=0}^{N_m^{(s)}} \Phi_m^{(s)}(t)\hat{\mathbf{q}}_m^{(s)} - \sum_{m=0}^{N_m^{(s)}} \Phi_m^{(s)}(t)\bar{\mathbf{u}}_m^{(s)}\right)^T \sum_{m=0}^{N_m^{(s)}} \Phi_m^{(s)}(t)\bar{\boldsymbol{\lambda}}_m^{(s)}\right)$ and apply the generalized midpoint-rule and variational calculus to obtain constraints $\frac{\partial \mathcal{G}}{\partial \hat{\boldsymbol{\lambda}}_j} = \mathbf{0}$ and $\frac{\partial \mathcal{G}}{\partial \bar{\boldsymbol{\lambda}}_m^{(s)}} = \mathbf{0}$, that are fulfilled in a weak sense. With the variation for $\bar{\mathbf{u}}_m^{(s)}$ follows the downsampling-equation

$$\sum_{k=0}^{N_k} \Delta t_k \left(\mathbf{B}^{(s)^T} \Phi_m^{(s)}(t_{k+\alpha}) \sum_{j=0}^{N_j} \Theta_j(t_{k+\alpha})\hat{\boldsymbol{\lambda}}_j - \Phi_m^{(s)}(t_{k+\alpha}) \sum_{m=0}^{N_m^{(s)}} \Phi_m^{(s)}(t_{k+\alpha})\bar{\boldsymbol{\lambda}}_m^{(s)} \right) = \mathbf{0}$$

and from $\sum_{m=0}^{N_m^{(s)}} \Delta t_m \Phi_m^{(s)}(t_{m+\alpha}) \sum_{m=0}^{N_m^{(s)}} \Phi_m^{(s)}(t_{m+\alpha})\bar{\boldsymbol{\lambda}}_m^{(s)}$ follows together with the kinetic and potential part of the discrete Lagrangian the local equation of motion and the time-stepping-scheme. All these equations can now be solved by a Newton-Raphson scheme. Due to $\hat{\boldsymbol{\lambda}}$ at the macro-discretization influencing both sides, the left and the right, the interface-problem still has to be solved all at once.

3.2 Reduce to time-stepping

To enable at least a time-stepping on the interface-problem from one macro-time-step to the next and only solve the subcycled Lagrange-multipliers between two macro-time-steps at once, we have to reduce the global integration and introduce some errors that way. The integration of the previously introduced equations is no longer performed from 0 to N_k, but only from one macro-time-step to the next one, which is visualized in Fig. 4a. While the global Lagrange-multiplier-field itself stays continuous, this requires the local Lagrange-multiplier to become discontinuous at the macro-time-steps, as can be seen in Fig. 4b. Finally, we have to apply some numerical dissipation or formulate the constraints for velocities, instead of displacements.



(a) Segmentation of integration of global Lagrange-multipliers.

(b) Discontinuous local Lagrange-multipliers.

Fig. 4 Segmentation of integration of Lagrange-multipliers according to macro-discretization.

Otherwise, high-frequency instabilities might prevent the solver from converging as pointed out by Farhat e.a. [2]. Hence, we replace the nodal displacements in the constraints with nodal velocities. Of course, with these modifications, our framework is no longer a variational method, but as shown in section 4, some beneficial properties of variational methods are still preserved, which is why we call it a variational-based framework instead.

4 Numerical experiments

In this section, we compare the accuracy of the BGC-macro method with our variational-based multirate method. To this end, we apply both methods to a nonlinear split Duffing-oscillator, as proposed by Prakash e.a. [8] and depicted in Fig. 5 and solve the interface-problem with a GMRes-method. The velocities from the BGC-macro in Fig. 6a exhibit the well-known spurious oscillations [8, 9], leading to rather large incompatibilities in the displacements. These spurious oscillations are reduced by the micro-discretization of the variational based method in Fig. 6b, which improves the compatibility of displacements. The solution from the BGC-macro method shows slightly less phase-error, as the displacement-curve is closer to the fine-solution, compared to the variational-based method, but the solution still remains in the margin between the fine and the coarse singlerate Newmark solution. The energy-behavior of the variational-based method in Fig. 7b is also still better compared to the BGC-macro method in Fig. 7a, despite the modifications made. The total energy's oscillations remain bounded, while we can observe a slight decline in the BGC-macro's total energy. Also the amplitude of the interface-energy's oscillations is smaller for the variational-based method. However, all this comes at the cost of a larger interface-problem.

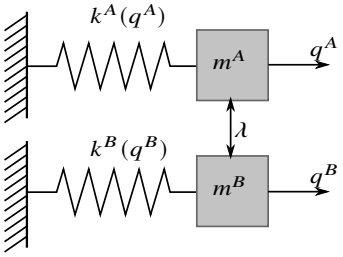
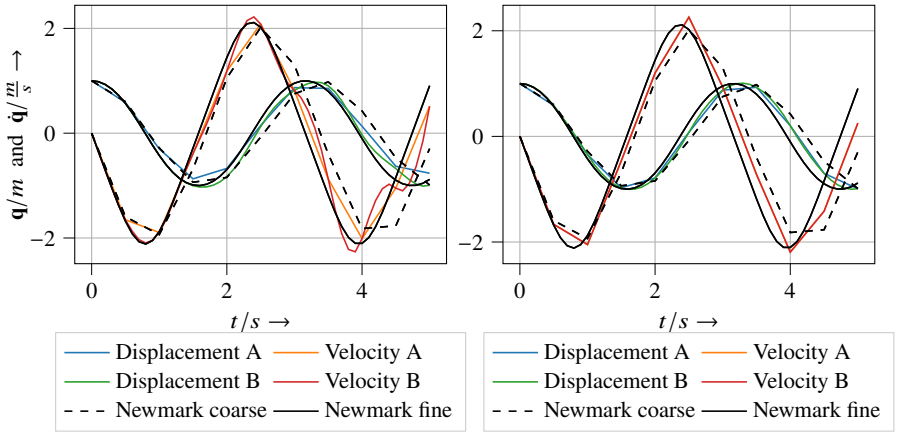


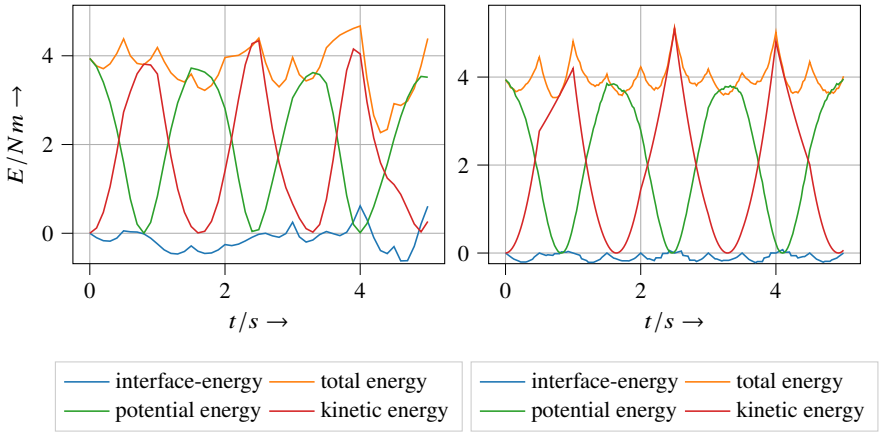
Fig. 5 Split Duffing-oscillator with stiffnesses $k^{(1)}(q^{(1)}) = 1\frac{N}{m} \cdot q^{(1)} + 1\frac{N}{m} \cdot q^{(1)^3}$, $k^{(2)}(q^{(2)}) = 10\frac{N}{m} \cdot q^{(2)} - 5\frac{N}{m} \cdot q^{(2)^3}$, masses $m^{(1)} = 1kg$, $m^{(2)} = 1kg$, time-step-sizes $\Delta t^{(1)} = 0.5s$, $\Delta t^{(2)} = 0.1s$.



(a) BGC-macro.

(b) Variational-based multirate integrator.

Fig. 6 Displacements and Velocities of the split Duffing-oscillator.



(a) BGC-macro.

(b) Variational-based multirate integrator.

Fig. 7 Energies of the split Duffing-oscillator.

5 Conclusions

The derived variational-based multirate method and its interface-problem is solved by a FETI-solver. The method enables a macro-time-stepping and still exhibits a better accuracy than the BGC-macro method. This comes at the cost of a larger interface-problem. A suitable preconditioner remains to be constructed.

Acknowledgements We thank the Deutsche Forschungsgemeinschaft (DFG) for the funding of project 357361040, in which context this work has been done.

References

1. Brun, M., Gravouil, A., Combescure, A., and Limam, A. Two FETI-based heterogeneous time step coupling methods for Newmark and alpha-schemes derived from the energy method. *Computer Methods in Applied Mechanics and Engineering* **283**, 130–176 (2015).
2. Farhat, C., Crivelli, L., and Géradin, M. Implicit time integration of a class of constrained hybrid formulations—Part I: Spectral stability theory. *Computer Methods in Applied Mechanics and Engineering* **125**, 71–107 (1995).
3. Farhat, C., Crivelli, L., and Roux, F.-X. A transient FETI methodology for large-scale parallel implicit computations in structural mechanics. *International Journal for Numerical Methods in Engineering* **37**, 1945–1975 (1994).
4. Farhat, C., Pierson, K., and Lesoinne, M. The second generation FETI methods and their application to the parallel solution of large-scale linear and geometrically non-linear structural analysis problems. *Computer Methods in Applied Mechanics and Engineering* **184**, 333–374 (2000).
5. Kane, C., Marsden, J. E., Ortiz, M., and West, M. Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems. *International Journal for Numerical Methods in Engineering* **49**, 51–63 (2000).
6. Leyendecker, S., Marsden, J. E., and Ortiz, M. Variational integrators for constrained dynamical systems. *ZAMM Zeitschrift für Angewandte Mathematik und Mechanik* **88**, 677–708 (2008).
7. Leyendecker, S. and Ober-Blöbaum, S. A Variational Approach to Multirate Integration for Constrained Systems. *Paul Fisette and Jean-Claude Samin, editors, ECCOMAS Thematic Conference: Multibody Dynamics: Computational Methods and Applications* **28**, 677–708 (2011).
8. Prakash, A., Taciroglu, E., and Hjelmstad, K. D. Computationally efficient multi-time-step method for partitioned time integration of highly nonlinear structural dynamics. *Computers & Structures* **133**, 51–63 (2014).
9. Seibold, A. S. and Rixen, D. J. A Variational Approach to Asynchronous Time-Integration of Structural Dynamics Problems in the Context of FETI and Spurious Oscillations on the Interfaces. *EURODYN 2020, XI International Conference on Structural Dynamics* 26–43 (2020).

Accelerated Convergence of the Pipelined Dynamic Iteration Method for RLC Circuits

Hélèna Shourick, Damien Tromeur-Dervout, and Laurent Chédot

1 Introduction

Since the pioneering work of Lelarasme et al. [9] that analyze in time domain large-scale problems arising from the modeling of integrated circuits, waveform relaxation methods (WR) [10] also known as dynamic iteration methods, a term first introduced by Miekkala and Nevanlinna [13, Eq. (2.2)], arouses more and more interest with the development of parallel computers. Let us recall the method of dynamic iteration as it was described by Miekkala[12] for ODE systems and adapted by Jiang & Wing for DAE systems [7]: Let $M \in \mathbb{C}^{n_1 \times n_1}$, $A \in \mathbb{C}^{n_1 \times n_1}$, $B \in \mathbb{C}^{n_1 \times n_a}$, $C \in \mathbb{C}^{n_a \times n_1}$, $D \in \mathbb{C}^{n_a \times n_a}$ matrices and $f_1 \colon]0,T] \to \mathbb{C}^{n_1}$, $f_2 \colon]0,T] \to \mathbb{C}^{n_a}$ functions, $x_0 \in \mathbb{C}^{n_1}$ initial state value. We define the DAE system in its state-space form:

$$\begin{cases} M\dot{x}(t) + Ax(t) + By(t) = f_1(t),\ t \in [0,T], \\ Cx(t) + Dy(t) = f_2(t),\ t \in [0,T], \\ x(0) = x_0, \end{cases} \tag{1}$$

where $x \colon]0,T] \to \mathbb{C}^{n_1}$ are the n_1 searched state solutions and $y \colon]0,T] \to \mathbb{C}^{n_a}$ are the n_a searched algebrical solutions.

Definition 1 (Dynamic Iteration for linear DAE) The Dynamic Iteration scheme for (1) considers the splitting of matrices M, A, B, C, D as $M = M_1 - M_2$, $A = A_1 - A_2$,

Hélèna Shourick
SuperGrid-Institute, 23 rue Cyprian, 69100 Villeurbanne, University of Lyon, UMR5208 U.Lyon1-CNRS, Institut Camille Jordan, e-mail: helena.shourick@supergrid-institute.com

Damien Tromeur-Dervout
University of Lyon, UMR5208 U.Lyon1-CNRS, Institut Camille Jordan,
e-mail: damien.tromeur-dervout@univ-lyon1.fr

Laurent Chédot
SuperGrid-Institute, 23 rue Cyprian, 69100 Villeurbanne,
e-mail: laurent.chedot@supergrid-institute.com

$B = B_1 - B_2$, $C = C_1 - C_2$, $D = D_1 - D_2$, where matrices M_1 and D_1 are assumed non-singular (which implies that the DAE system has index one)

$$\begin{cases} M_1\dot{x}^{(k)}(t) + A_1x^{(k)}(t) + B_1y^{(k)}(t) = M_2\dot{x}^{(k-1)}(t) + A_2x^{(k-1)}(t) + B_2y^{(k-1)}(t) \\ \qquad\qquad\qquad\qquad\qquad\qquad +f_1(t),\ t \in [0,T] \\ C_1x^{(k)}(t) + D_1y^{(k)}(t) = C_2x^{(k-1)}(t) + D_2y^{(k-1)}(t) \\ \qquad\qquad\qquad\qquad\qquad\qquad +f_2(t),\ t \in [0,T], \\ x^{(k)}(0) = x_0, \end{cases} \tag{2}$$

This fixed-point process must be contracting to converge. We propose to combine the DI method with the Restricted Additive Schwarz splitting and the Aitken's acceleration of the convergence technique to obtain a DI method less sensitive to the contracting property even with applying it on a pipeline of several time step or on a nonlinear problem. Related works on improvement of DI are that follow. Arnold & Gunther [1] proposed several techniques for preconditioning the fixed-point problem. Some waveform successive overrelaxation (SOR) techniques have been proposed by Janssen and Vandewalle [6] to accelerate the standard waveform method. Leimkuhler proposed to accelerate the WR by solving the defect equations with a larger timestep, or by using a recursive procedure based on a succession of increasing timesteps [8]. Lumdaisne & Wu proposed to accelerate the WR by Krylov subspace techniques (WGMRES) [11] to solve time-dependent problems. Gausling & al [5] analyzed the contraction and the rate of convergence of the co-simulation process for a test circuit subjected to uncertainties on the parameters of its components. In section 2, we consider the Restrictive Additive Schwarz [3] for the Eq. (1) with $M = I$ and we show that is a DI scheme with a specific splitting. Then we can apply the Aitken's acceleration of the convergence technique to obtain the true solution whether the DI is contracting or not (it must not stagnate). Section 3 considers advantages and drawbacks of the sequential (time step after time step) and the pipelined (several time steps at once) implementations of DI and their acceleration of convergence. Section 4 gives some numerical results of the pipelined DI accelerated by the Aitken's technique while section 5 concludes.

2 DI with RAS splitting

Let us consider Eq. (1) resulting from the modeling of an electrical network where we choose $M = I$, this choice corresponds to a change of variables on the voltage terms in the Kirchhoff's law. According to the RAS method notation of Cai & Sarkis [3] applied to the graph of the operator $\begin{pmatrix} I & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} \epsilon A & B \\ C & D \end{pmatrix}$, where the ϵ is chosen in order to keep all data dependencies, we define the associated restriction operators R_i^p and $\tilde{R}_i^0$. Then, the k^{th} RAS iteration can be written as:

$$\begin{cases} \dot{x}_i^{(k)}(t) + A_i x_i^{(k)}(t) + B_i y_i^{(k)}(t) = b_i^b(t) - E_{i,d}^d x_{ie}^{(k-1)}(t) - E_{i,d}^a y_{ie}^{(k-1)}(t), \\ C_i x_i^{(k)}(t) + D_i y_i^{(k)}(t) = b_i^a(t) - E_{i,a}^d x_{ie}^{(k-1)}(t) - E_{i,a}^a y_{ie}^{(k-1)}(t), \\ x_i^{(k)}(0) = R_i^{p,d} x_0, \quad t \in [0,T]. \end{cases} \tag{3}$$

With $A_i = R_i^{p,d} A (R_i^{p,d})^T$, $B_i = R_i^{p,d} B (R_i^{p,a})^T$, $C_i = R_i^{p,a} C (R_i^{p,d})^T$, $D_i = R_i^{p,a} D (R_i^{p,a})^T$, $E_{i,d}^d = R_i^{p,d} A (R_{i,e}^{p,d})^T$, $E_{i,d}^a = R_i^{p,d} B (R_{i,e}^{p,a})^T$, $E_{i,a}^d = R_i^{p,a} C (R_{i,e}^{p,d})^T$ and $E_{i,a}^a = R_i^{p,a} D (R_{i,e}^{p,a})^T$. The operator $R_i^{p,d}$ (respectively $R_i^{p,a}$) is the restriction to the differential variables (respectively algebraical variables) of R_i^p. We also define $\tilde{R}_i^{0,d}$ and $\tilde{R}_i^{0,a}$ such that $\tilde{R}_i^0 = \begin{pmatrix} \tilde{R}_i^{0,d} & 0_{n_{i_1} \times n_2} \\ 0_{n_{i_2} \times n_1} & \tilde{R}_i^{0,a} \end{pmatrix}$ as we have chosen to separate the differential and algebraic parts.

By summing up the contribution of each RAS partition, we can show that the k^{th} RAS iteration for solving (1) (with $M = I$) is a DI as defined in (2) associated to the following splitting of the operators $A = A_1^d - A_2^d$, $B = B_1^d - B_2^d$, $C = C_1^a - C_2^a$, $D = D_1^a - D_2^a$:

$$\begin{cases} \dot{x}^{(k)}(t) + A_1^d x^{(k)}(t) + B_1^d y^{(k)}(t) = b^d(t) + A_2^d x^{(k-1)}(t) + B_2^d y^{(k-1)}(t), \\ C_1^a x^{(k)}(t) + D_1^a y^{(k)}(t) = b^a(t) + C_2^a x^{(k-1)}(t) + D_2^a y^{(k-1)}(t), \\ x^{(k)}(0) = x_0, \quad t \in [0,T]. \end{cases} \tag{4}$$

with

$$A_1^d = \sum_{i=0}^{N-1} \tilde{R}_i^{0,d} A_i R_i^{p,d},\ A_2^d = -\sum_{i=0}^{N-1} \tilde{R}_i^{0,d} E_{i,d}^d R_{ie}^{p,d},\ b^d(t) = \sum_{i=0}^{N-1} \tilde{R}_i^{0,d} R_i^{p,d} b^d(t),$$

$$B_1^d = \sum_{i=0}^{N-1} \tilde{R}_i^{0,d} B_i R_i^{p,a},\ B_2^d = -\sum_{i=0}^{N-1} \tilde{R}_i^{0,d} E_{i,d}^a R_{ie}^{p,a},$$

$$C_1^a = \sum_{i=0}^{N-1} \tilde{R}_i^{0,a} C_i R_i^{p,d},\ C_2^a = -\sum_{i=0}^{N-1} \tilde{R}_i^{0,a} E_{i,a}^d R_{ie}^{p,d},$$

$$D_1^a = \sum_{i=0}^{N-1} \tilde{R}_i^{0,a} D_i R_i^{p,a},\ D_2^a = -\sum_{i=0}^{N-1} \tilde{R}_i^{0,a} E_{i,a}^a R_{ie}^{p,a},\ b^a(t) = \sum_{i=0}^{N-1} \tilde{R}_i^{0,d} R_i^{p,a} b^a(t).$$

Thus, the RAS method applied to DAE system belongs to the DI methods with a specific splitting of the operators. Then we can reduce this specific DI method to an interface problem and we can accelerate its convergence to the true solution with the Aitken's acceleration of the convergence technique as in [4]. Denoting $W_{i,e}^{p,d}$ and $W_{i,e}^{p,a}$ the differential and algebraical components of $W_{i,e}^p$, we define $\Gamma^d = \{W_{0,e}^{p,d}, \ldots, W_{N-1,e}^{p,d}\}$, $\Gamma^a = \{W_{0,e}^{p,a}, \ldots, W_{N-1,e}^{p,a}\}$ and $\Gamma = \{\Gamma^d, \Gamma^a\}$ and R_Γ the restriction to the global interface $R_\Gamma = \begin{pmatrix} R_\Gamma^d & 0 \\ 0 & R_\Gamma^a \end{pmatrix}$ with $R_\Gamma^d = (R_{0,ie}^{p,d}, \ldots, R_{N-1,ie}^{p,d})^T$, $R_\Gamma^a = (R_{0,ie}^{p,a}, \ldots, R_{N-1,ie}^{p,a})^T$ and finally $z^{(k)} = (x^{(k)T}, y^{(k)T})^T$ and $z_\Gamma^{(k)} = R_\Gamma z^{(k)}$.

The DI with RAS splitting defined by Eq. (4) applied to a linear DAE system with D_1^a invertible has an error operator $P_{t,\Gamma}, t \in]0,T]$ for the problem interface that does not depend on the iteration number, such that the restriction of the iteration to the global interface satisfies: $z_\Gamma^{(k)} = P_{t,\Gamma} z_\Gamma^{(k-1)} + c$. Therefore, the convergence of the DI to the true solution $z^{(\infty)}$ can be performed using the Aitken's technique for accelerating the convergence, if 1 does not belong to the spectrum of $P_{t,\Gamma}$, as follows:

$$z_\Gamma^{(\infty)} = (I - P_{t,\Gamma})^{-1}(z_\Gamma^{(1)} + P_{t,\Gamma} z_\Gamma^{(0)}) \tag{5}$$

Numerically, the time derivative in Eq. (4) must be discretized using backward Euler scheme with a regular time step Δt for example. We write the DI with RAS splitting on the discretized system as:

$$\begin{cases} \tilde{A}_1^d x^{n+1,(k+1)} + \tilde{B}_1^d y^{n+1,(k+1)} = \tilde{b}^{n+1,d} + \tilde{A}_2^d x^{n+1,(k)} + \tilde{B}_2^d y^{n+1,(k)}, \\ C_1^a x^{n+1,(k+1)} + D_1^a y^{n+1,(k+1)} = b^{n+1,a} + C_2^a x^{n+1,(k)} + D_2^a y^{n+1,(k)}, \\ x^{0,(k+1)} = x_0. \end{cases} \tag{6}$$

with $\tilde{A}_1^d = I_{n,1}^d + \Delta t A_1^d, \tilde{B}_1^d = \Delta t B_1^d, \tilde{A}_2^d = \Delta t A_2^d, \tilde{B}_2^d = \Delta t B_2^d, \tilde{b}^{n+1,d} = x^{n,*} + \Delta t b^{n+1,d}$ where $x^{n,*} = x^{n,(k+1)}$ or $x^{n,*} = x^{n,(\infty)}$ leading to the sequential DI or pipelined DI strategies. Locally, it is written, with $x_i^{0,(k+1)} = R_i^{p,d} x_0$:

$$\underbrace{\begin{pmatrix} x_i^{n+1,(k+1)} \\ y_i^{n+1,(k+1)} \end{pmatrix}}_{z_i^{n+1,(k+1)}} = \underbrace{\begin{pmatrix} \tilde{A}_i & \tilde{B}_i \\ C_i & D_i \end{pmatrix}^{-1}}_{\tilde{\mathbb{A}}_i^{-1}} \left(\underbrace{\begin{pmatrix} \tilde{b}_{i,d}^{n+1} \\ b_{i,a}^{n+1} \end{pmatrix}}_{\tilde{b}_i^{n+1}} - \underbrace{\begin{pmatrix} \tilde{E}_{i,d}^d & \tilde{E}_{i,d}^a \\ E_{i,a}^d & E_{i,a}^a \end{pmatrix}}_{\tilde{\mathbb{E}}_i} \underbrace{\begin{pmatrix} x_{i,e}^{n+1,(k)} \\ y_{i,e}^{n+1,(k)} \end{pmatrix}}_{z_{i,e}^{n+1,(k)}} \right) \tag{7}$$

The choice for the term $x^{n,*}$ has an impact on the implementation and the Aitken's acceleration of convergence technique as described in the next section.

3 Pipelined time stepping strategy for DI

Let's consider the DI applied over a time interval $[t^0, t^F]$ with a constant time step Δt satisfying $t^F - t^0 = \Xi \Delta t$ with $\Xi \in \mathbb{N}^*$.

The sequential DI strategy consists in iterating the DI method until convergence on one time step before applying it to the next time step (Algorithm 1, $x^{n,*} = x^{n,(\infty)}$ in (6)). In the pipelined DI strategy, each DI iteration is performed over several time steps, these iterations are repeated until convergence (Algorithm 2, $x^{n,*} = x^{n,(k)}$ in (6)). The two algorithms differ by the choice of $x^{n,*} = x^{n,(k)}$ but also by the inversion of the order of loops 1 and 2.

In the following, we adapt the Aitken's acceleration of the convergence technique to accelerate the pipelined DI with RAS splitting.

Algorithm 1 Sequential DI strategy

1: **for** $n = 0 \dots \Xi - 1$ **do**
2: **for** $k = 1 \dots$ until convergence **do**
3: Solve $z_i^{n+1,(k+1)} = \tilde{\mathbb{A}}_i^{-1}\tilde{b}_i^{n+1} - \tilde{\mathbb{B}}_i z_{i,e}^{n+1,(k)}$.
4: **end for**
5: **end for**

Algorithm 2 Pipelined DI strategy

1: **for** $k = 1 \dots$ until convergence **do**
2: **for** $n = 0 \dots \Xi - 1$ **do**
3: Solve $z_i^{n+1,(k+1)} = \tilde{\mathbb{A}}_i^{-1}\tilde{b}_i^{n+1} - \tilde{\mathbb{B}}_i z_{i,e}^{n+1,(k)}$.
4: **end for**
5: **end for**

Definition 2 We note $Z_i^{(k)} \in \mathbb{C}^{\Xi n}$ the $(k)^{th}$ DI iteration corresponding to the concatenation over the Ξ time steps of the i^{th} partition W_i^p of the $(k+1)^{th}$ pipelined DI iteration: $Z_i^{(k)} = ((z_i^{1,(k)})^T, \dots, (z_i^{\Xi,(k)})^T)^T$, and the dependencies as $Z_{i,e}^{(k)} = ((z_{i,e}^{1,(k)})^T, \dots, (z_{i,e}^{\Xi,(k)})^T)^T$.

We define the operator $\mathbb{I}_{d,i}$ such that: $\mathbb{I}_{d,i} z_i^{n,(k)} = \begin{pmatrix} \Delta t\, x_i^{n,(k)} \\ 0_{n_{i,a}} \end{pmatrix}$. We also define $Z_\Gamma^{(k)} \in \mathbb{C}^{\Xi n_\Gamma}$ denote the k^{th} pipelined DI iterations of the global interface values of the Ξ time steps: $Z_\Gamma^{(k)} = ((z_\Gamma^{1,(k)})^T, \dots, (z_\Gamma^{\Xi,(k)})^T)^T$ and let $\mathbb{I}_d$ be the operator that follows: $\mathbb{I}_d = (\mathbb{I}_{d,1}^T, \dots, \mathbb{I}_{d,\Xi}^T)^T$.

Proposition 1 *The k^{th} pipelined DI iteration applied on to Ξ time steps Δt is written locally on the partition W_i^p :*

$$\begin{pmatrix} \tilde{\mathbb{A}}_i & & & \\ -\mathbb{I}_{d,i}\ \tilde{\mathbb{A}}_i & & & \\ & \ddots & \ddots & \\ & & -\mathbb{I}_{d,i}\ \tilde{\mathbb{A}}_i \end{pmatrix} Z_i^{(k)} = \begin{pmatrix} b_i^1 + \mathbb{I}_{d,i} z_i^0 \\ b_i^2 \\ \vdots \\ b_i^\Xi \end{pmatrix} - \begin{pmatrix} \tilde{\mathbb{E}}_i & & \\ & \tilde{\mathbb{E}}_i & \\ & & \ddots \\ & & \tilde{\mathbb{E}}_i \end{pmatrix} Z_{i,e}^{(k-1)} \tag{8}$$

Equation (8) has the same form (and the same properties) than the sequential DI. We can then apply the same methodology as in the sequential case. That is to say: as the convergence is purely linear one can thus build an operator of error $\mathbb{P}_\Gamma$ and use the Aitken acceleration of the convergence method. $\mathbb{P}_\Gamma$ can be computed algebraically or numerically. It will be of size $\Xi n_\Gamma \times \Xi n_\Gamma$ and we will need $\Xi n_\Gamma + 1$ RAS iterations to calculate it numerically. Nevertheless, we can take advantage of the structure of $\mathbb{P}_\Gamma$ for linear DAE with regular time stepping as the $\mathbb{P}_\Gamma$ operator and the P_Γ operator are linked as shown below.

By noting $M_{n,RAS}^{-1}$ the RAS operator (defined as in [3]) and $P_{n,\Gamma}$ the error operator associated to the n^{th} time step. Then, similarly to the sequential DI, we can restrict the pipelined DI iteration to the global interface of (5) of all the Ξ time steps:

Proposition 2 *The k^{th} iteration of the pipelined DI can be written on the global interface of the Ξ time steps:*

$$Z_\Gamma^{(k)} = \begin{pmatrix} I & & & \\ M_2^{-1} & I & & \\ & \ddots & \ddots & \\ & & M_\Xi^{-1} & I \end{pmatrix}^{-1} \begin{pmatrix} P_{1,\Gamma} & & & \\ & P_{2,\Gamma} & & \\ & & \ddots & \\ & & & P_{\Xi,\Gamma} \end{pmatrix} Z_\Gamma^{(k-1)} + \begin{pmatrix} R_\Gamma M_{1,RAS}^{-1} \mathbb{I}_d z^0 + c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix} \quad (9)$$

where $M_i^{-1} = R_\Gamma M_{i,RAS}^{-1} \mathbb{I}_d R_\Gamma^T$, $i = 2 \ldots \Xi$

The error operator can be calculated in two ways algebraically or numerically. We recall that the global interface Γ is defined as the concatenation of $W_{i,e}^p$, that is $\Gamma = \left\{ W_{0,e}^p, \ldots, W_{N-1,e}^p \right\}$ of size $n_\Gamma = \sum_{i=0}^{N-1} n_{i,e}$. It is pointed out that to numerically calculate the error operator, it is necessary to perform one more iteration than the size of the vector to be accelerated.

In the sequential DI strategy, we can apply the Aitken's technique for accelerating convergence, after $n_\Gamma + 1$ DI iterations for the first regular time step, in order to numerically build the P_Γ operator. Then, if we use the same time step size for the following time steps, and if there is no non-linearity and no change in the topology, we can perform the Aitken's convergence acceleration technique after one DI iteration. However, it is necessary to recalculate the error operator P_Γ at each change of topology or at each change of time step. Indeed the matrices $\mathbb{A}$ and $\mathbb{E}$ which have an impact in the P_Γ are modified by the changes in topology and changes in time step size (because of the discretization).

Moreover, in the pipelined DI strategy, the interface to be accelerated is the concatenation of the interfaces over the entire period of interest ($\Xi \Delta t$ here). This interface is of size $\Xi \times n_\Gamma$, it will therefore take $\Xi \times n_\Gamma + 1$ DI iterations in order to compute the error operator $\mathbb{P}_\Gamma$, whether or not there is a change in topology or time step. On the other hand, changes in the size of the time steps must be planned before launching the simulation.

Table 1 summarizes the number of iterations needed to simulate the problem on the period $\Xi \Delta t$, depending on the strategy chosen.

Table 1 Number of iterations needed to performed the simulation using the DI method accelerated with the Aitken acceleration technique

DI Strategy	Sequential	First time step sequential, computation of the error operator then pipelined	Pipelined
no non-linearity& fixed and equidistant time steps	$\Xi + n_\Gamma$	$2(\Xi - 1) + n_\Gamma + 1$	$\Xi \times n_\Gamma + 1$
fixed variable time step distribution with j changes in time step length	$\Xi - j + j \times (n_\Gamma + 1)$	$2(\Xi - j) + j \times (n_\Gamma + 1)$	$\Xi \times n_\Gamma + 1$
non-fixed variable time step distribution with j changes in time step length	$\Xi - j + j \times (n_\Gamma + 1)$	Strategy non valid	Strategy non valid
presence of non-linear components	$\Xi \times (n_\Gamma + 1)$	Strategy non valid	$\Xi \times n_\Gamma + 1$
j non-linearity events	$\Xi - j + j \times (n_\Gamma + 1)$	$2(\Xi - j) + j \times (n_\Gamma + 1)$	$\Xi \times n_\Gamma + 1$

4 Numerical results

The numerical results are performed on the test circuit of Pade and Tischendorf [14], with the sequential strategy we know from the spectrum of the error operator that the convergence of the method depends on the value of the components but especially on the size of the time step Δt. The size of the problem is 9 and the global interface Γ size is 3. We refer to [2] for parallel results approximating P_Γ by using singular values decomposition of the RAS interface solution iterates and the Aitken's acceleration for solving large-scale elliptic Darcy flows 3D problems. Figure 1 (left) shows the evolution of the spectral radius of $\mathbb{P}_\Gamma$ computed numerically with respect to the number of regular time steps in the pipeline. It shows that the convergence of the pipelined DI with RAS splitting deteriorates with increasing number of pipelined time steps. This result is corroborated by Figure 1 (middle) which shows the error of the pipelined DI with RAS splitting between two consecutive iterations, with respect to the RAS iterations, for each time step in the pipeline. Although the first few time steps in the pipeline the DI converge, this is not the case for the last few time steps. This shows the limitation of the pipeline size. Nevertheless, with Aitken's convergence acceleration, we can still accelerate the DI for all time steps. We should note that we also have a limitation on the pipeline size due to numerical problems in the numerical computation of the error operator if the DI diverges too strongly for some time steps. Nevertheless, in the pipelined strategy, the numerically calculated error operator can take into account some changes in the behavior of the electrical components, which allows us to apply Aitken's convergence acceleration technique even in the presence of nonlinear components, as shown in the figure (right) with a nonlinear resistor in the test circuit [14].

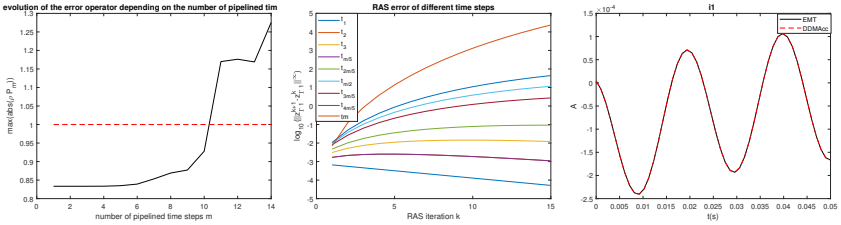


Fig. 1 (left) Evolution of the spectral radius of the error operator depending on the number of pipelined regular time steps of size $\Delta t = 1.1\ 10^{-3}$ (left, $\Xi = 14$), (middle) DI with the RAS splitting convergence behavior ($\log_{10}(||z^{(2k)} - z_{\text{ref}}||_\infty)$) on each of the pipelined time steps with respect to the iterations, (right) Comparison between the DI with the RAS splitting with the Aitken's technique for accelerating convergence and the DAE monolithic reference with a non-linear component, with $\Delta t = 1, 1.10^{-4}$, $\Xi = 100$.

5 Conclusion

Starting from the RAS method applied to a state-space DAE system, we show that this method is a dynamic iteration method with a specific operator splitting.

Then, the DI with RAS splitting inherits the property of reducing the size of the error operator to the size of a global interface problem. It also inherits its pure linear convergence/divergence when applied to a linear problem. We are then able to accelerate the convergence by Aitken acceleration by working on the global interface, thus, we get rid of the contracting constraint of the error operator. Writing RAS as a DI with RAS splitting also makes us consider the implementation of the pipelined strategy performing iteration over several time steps. We have shown the link between the error operators of the sequential DI and pipelined DI strategies, which allows us to apply Aitken's convergence acceleration technique on the pipelined DI. The optimal use cases for these strategies were also discussed. Numerical results show that pipelined DI with RAS splitting successfully applies Aitken acceleration to both contracting and non-contracting DI. It also opens up the Aitken acceleration of DI convergence for nonlinear problems (using a different linearization of the nonlinear problem on the time steps in the pipeline) and for pipelined steps with different sizes.

References

1. Arnold, M. and Gunther, M. Preconditioned dynamic iteration for coupled differential-algebraic systems. *BIT* **41**(1), 1–25 (2001).
2. Berenguer, L. and Tromeur-Dervout, D. Sparse Aitken–Schwarz domain decomposition with application to Darcy flow. *Computers & Fluids* **249**, 105687 (2022).
3. Cai, X.-C. and Sarkis, M. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.* **21**(2), 792–797 (1999).
4. Garbey, M. and Tromeur-Dervout, D. On some Aitken-like acceleration of the Schwarz method. *Int. J. Numer. Meth. Fluids* **40**, 1493 – 1513 (2002).
5. Gausling, K. and Bartel, A. Density Estimation Techniques in Cosimulation Using Spectral- and Kernel Methods. In: Langer, U and Amrhein, W and Zulehner, W (ed.), *Scientific Computing in Engineering, SCEE 2016*, *Mathematics in Industry-Cham*, vol. 28, 81–89 (2018).
6. Janssen, J. and Vandewalle, S. On SOR waveform relaxation methods. *SIAM Journal on Numerical Analysis* **34**(6), 2456–2481 (1997).
7. Jiang, Y. and Wing, O. A note on the spectra and pseudospectra of waveform relaxation operators for linear differential-algebraic equations. *SIAM Journal on Numerical Analysis* **38**(1), 186–201 (2000).
8. Leimkuhler, B. Timestep acceleration of waveform relaxation. *SIAM Journal on Numerical Analysis* **35**(1), 31–50 (1998).
9. Lelarasmee, E., Ruehli, A., and Vincentelli, A. The Waveform Relaxation Method for Time-Domain Analysis of Large Scale Integrated Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **1**, 131–145 (1982).
10. Lumsdaine, A. and White, J. Accelerating Wave-Form relaxation methods with application to parallel semiconductor-device simulation. *Numer. Funct. Anal. Optim.* **16**(3-4), 395–414 (1995).
11. Lumsdaine, A. and Wu, D. Krylov subspace acceleration of waveform relaxation. *SIAM Journal on Numerical Analysis* **41**(1), 90–111 (2003).
12. Miekkala, U. Dynamic Iteration methods applied to linear DAE systems. *J. Comput. Appl. Math.* **25**(2), 133–151 (1989).
13. Miekkala, U. and Nevanlinna, O. Convergence of Dynamic Iteration methods for initial-value problems. *SIAM Journal on Scientific and Statistical Computing* **8**(4), 459–482 (1987).
14. Pade, J. and Tischendorf, C. Waveform relaxation: a convergence criterion for differential-algebraic equations. *Numer. Algorithms* **81**(4, SI), 1327–1342 (2019).

GPU Optimizations for the Hierarchical Poincaré-Steklov Scheme

Anna Yesypenko and Per-Gunnar Martinsson

1 Introduction

We describe methods for solving boundary value problems of the form

$$\begin{cases} \mathcal{A}u(x) = f(x), & x \in \Omega, \\ u(x) = g(x), & x \in \partial\Omega, \end{cases} \tag{1}$$

where $\mathcal{A}$ is a second order elliptic differential operator, and Ω is domain in two dimensions with boundary $\partial\Omega$. For the sake of concreteness, we will focus on the case where $\mathcal{A}$ is a variable coefficient Helmholtz operator

$$\mathcal{A}u(x) = -\Delta u(x) - \kappa^2 b(x)u(x), \tag{2}$$

where κ is a reference wavenumber, and where $b(x)$ is a smooth non-negative function that typically satisfies $0 \leq b(x) \leq 1$. Upon discretizing (1), one obtains a linear system

$$\mathbf{Au} = \mathbf{f} \tag{3}$$

involving a sparse coefficient matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$. The focus of this work is on efficiently solving the sparse system (3) for the Hierarchical Poincaré-Steklov (HPS) discretization. HPS is a multi-domain spectral collocation scheme that allows for relatively high choices of p, while interfacing well with sparse direct solvers. For (1) discretized with HPS with local polynomial order p, the cost of factorizing $\mathbf{A}$ directly is

$$T_{\text{build}} = O\left(\underbrace{p^4 N}_{\text{leaf operations}} + \underbrace{N^{3/2}}_{\text{direct solver}}\right). \tag{4}$$

Anna Yesypenko, Per-Gunnar Martinsson
Oden Institute, e-mail: annayesy@utexas.edu, pgm@oden.utexas.edu

After the leaf operations are complete, the cost to factorize the system directly has no pre-factor dependence on p. The pre-factor cost of the leaf operations, however, has long been viewed as prohibitively expensive. This manuscript describes simple GPU optimizations using batched linear algebra that substantially accelerate the leaf operations and shows compelling results for p up to 42. We also demonstrate that the choice of p does not have substantial effects on the build time for the direct factorization stage, allowing p to be chosen based on physical considerations instead of practical concerns.

High order discretization is crucial in resolving variable-coefficient scattering phenomena due to the well known "pollution effect" that generally requires the number of points per wavelength to increase, the larger the computational domain is. The pollution effect is very strong for low order discretizations, but quickly gets less problematic as the discretization order increases [2, 7]. HPS is less sensitive to pollution because the scheme allows for high choices of local polynomial order p [11, 16]. Combining HPS discretization with efficient sparse direct solvers provides a powerful tool for resolving challenging scattering phenomena to high accuracy, especially for situations where no efficient preconditioners are known to exist (e.g. trapped rays, multiple reflections, backscattering) [9].

2 HPS Discretization and interfacing with sparse direct solvers

We next discuss the HPS discretization and efficient methods to interface the resulting sparse linear system with direct solvers. We introduce the HPS briefly for the simple model problem (1), and refer the reader to [1, 17, 13] for details and extensions. An important limitation of the discretization is that we assume the solution is smooth and that the coefficients in the operator $\mathcal{A}$ of (1) are smooth as well.

The domain Ω is partitioned into non-overlapping subdomains. The discretization is described by two parameters, a and p, which are the element size and local polynomial order, respectively. On each subdomain, we place a $p \times p$ tensor product mesh of Chebyshev points. Internal to each subdomain, the PDE is enforced locally via spectral differentiation and direct collocation. On element boundaries, we enforce that the flux between adjacent boundaries is continuous. On each subdomain of p^2 nodes, the spectral differentiation operators lead to a dense matrix of interactions of size $p^2 \times p^2$. To improve efficiency of sparse direct solvers for HPS discretizations, we "eliminate" the dense interactions of nodes interior to each subdomain. This process is referred to as "static condensation" [4, 12]. The remaining active nodes are on the boundaries between subdomains. As a result of the leaf elimination, we produce a smaller system

$$\tilde{\mathbf{A}}\tilde{\mathbf{u}} = \tilde{\mathbf{f}}. \tag{5}$$

of size $\approx N/p$ with equivalent body load $\tilde{\mathbf{f}}$ on the active nodes located on the boundaries between subdomains, as shown in Figure 2.

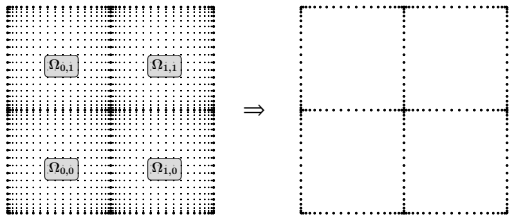


Fig. 1 Prior to interfacing with sparse direct solvers, we do static condensation to produce an equivalent system (5) to solve on the remaining active nodes. The original grid has N points, and remaining grid has $\approx N/p$ points.

Due to the domain decomposition used in HPS, the leaf operations required to produce the equivalent system (5) can be done embarrassingly in parallel. The leaf operations require independent dense linear algebraic operations (e.g., LU factorization, matrix-matrix multiply) on N/p^2 systems, each of size $p^2 \times p^2$, resulting in an overall cost of $O(p^4 N)$. For p up to about 42, these operations can be efficiently parallelized with batched linear algebra (BLAS). However, for larger p, methods that produce a sparser equivalent system may be more appropriate [3, 10].

Overhead costs can make achieving high arithmetic intensity for many small parallel tasks a challenge. However, batched BLAS offers a solution. It is highly optimized software for parallel operations on matrices that are small enough to fit in the top levels of the memory hierarchy (i.e., smaller than the L2 cache size) [8]. The framework groups small inputs into larger "batches" to automatically achieve good parallel performance on high-throughput architectures such as GPUs.

The technique we present is most readily applicable to the case where the same discretization order p is used on every discretization patch. However, it would not be too difficult to allow p to be chosen from a fixed set of values (say $p \in \{6, 10, 18, 36\}$ or something similar). This would enable many of the advantages of hp-adaptivity, while still enabling batching to accelerate computations.

Remark 1 Since the leaf computations are *very* efficient, we saved memory and reduced communication by not explicitly storing the factorizations of the local spectral differentiation matrices. Instead, these are reformed and refactored after each solve involving the reduced system (5).

We combined the fast leaf factorization procedure with two methods for solving the reduced system (5). The first option for solving (5) uses a black-box sparse direct solver with the nested dissection (ND) ordering. ND is a based on a multi-level graph partitioning of nodes and produces a sparse factorization with minimal fill-in [5, 6]. In 2D, sparse factorization using the ND ordering requires $O\left(N^{3/2}\right)$ time to build and $O\left(N \log N\right)$ time to apply.

As a second option for solving (5), we used a scheme we refer to as SlabLU, which is a simplified two-level scheme (as opposed to standard hierarchical schemes) that is designed for ease of parallelization [18]. To be precise, SlabLU uses a decomposition of the domain into elongated “slab” subdomains, as shown in Figure 2. With this decomposition, the linear system (5) has the block form

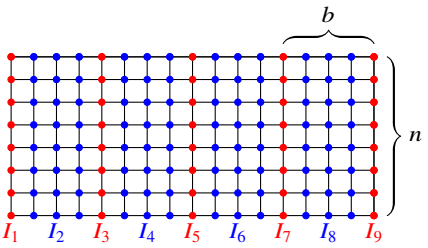


Fig. 2 Domain decomposition used in SlabLU. The even-numbered nodes correspond to the nodes interior to each subdomain. The odd-numbered nodes correspond to interfaces between slabs. The slab partitioning is chosen so that interactions between slab interiors are zero. The slabs have width of b points.

$$\begin{bmatrix} \tilde{\mathbf{A}}_{11} & \tilde{\mathbf{A}}_{12} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \dots \\ \tilde{\mathbf{A}}_{21} & \tilde{\mathbf{A}}_{22} & \tilde{\mathbf{A}}_{23} & \mathbf{0} & \mathbf{0} & \dots \\ \mathbf{0} & \tilde{\mathbf{A}}_{32} & \tilde{\mathbf{A}}_{33} & \tilde{\mathbf{A}}_{34} & \mathbf{0} & \dots \\ \mathbf{0} & \mathbf{0} & \tilde{\mathbf{A}}_{43} & \tilde{\mathbf{A}}_{44} & \tilde{\mathbf{A}}_{45} & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}}_1 \\ \tilde{\mathbf{u}}_2 \\ \tilde{\mathbf{u}}_3 \\ \tilde{\mathbf{u}}_4 \\ \vdots \end{bmatrix} = \begin{bmatrix} \tilde{f}_1 \\ \tilde{\mathbf{f}}_2 \\ \tilde{f}_3 \\ \tilde{f}_4 \\ \vdots \end{bmatrix}. \tag{6}$$

The nodes internal to each slab are eliminated by computing sparse factorizations of the diagonal blocks $\tilde{\mathbf{A}}_{22}, \tilde{\mathbf{A}}_{44}, \dots$ in parallel. This results in another block tridiagonal coefficient matrix $\mathbf{T}$ that has much smaller blocks than $\tilde{\mathbf{A}}$ (and half as many). The blocks of $\mathbf{T}$ are dense, but can be represented efficiently using data sparse formats such as the $\mathcal{H}$-matrix format of Hackbusch. The corresponding $\mathcal{H}$-matrices have significantly low ranks, due to the thinness of the slabs. The blocks of $\mathbf{T}$ can be rapidly constructed in $\mathcal{H}$-matrix format using the black-box randomized compression techniques described in [14].

The reduced linear system with blocks having $\mathcal{H}$-matrix structure can in principle be solved efficiently using rank-structured linear algebra. However, we found that for 2D problems, it is most efficient to relinquish the rank structure and simply convert all blocks to a dense format before factorizing the block tridiagonal system. (In 3D, this simplistic approach is possible only for small problems.) With a choice of slab width b that grows slowly with the problem size as $b \sim n^{2/3}$, the resulting two-level scheme has complexity $O(N^{5/3})$ to factorize $\tilde{\mathbf{A}}$ directly and $O\left(N^{7/6}\right)$ complexity to apply the computed factors to solve (5). SlabLU is simple scheme that leverages high concurrency and batched BLAS to achieve high performance on modern hybrid architectures. Despite the asymptotically higher costs, SlabLU performs favorably compared to multi-level nested dissection schemes in its build time and memory footprint, as we demonstrate in Section 3. [18] provides details on SlabLU.

3 Numerical experiments

We demonstrate the effectiveness of the HPS discretization combined with sparse direct solvers in solving high-frequency Helmholtz equations. The experiments were conducted on a desktop computer equipped with a 16-core Intel i9-12900k CPU and 128GB of memory, and a NVIDIA RTX 3090 GPU with 24GB of memory.

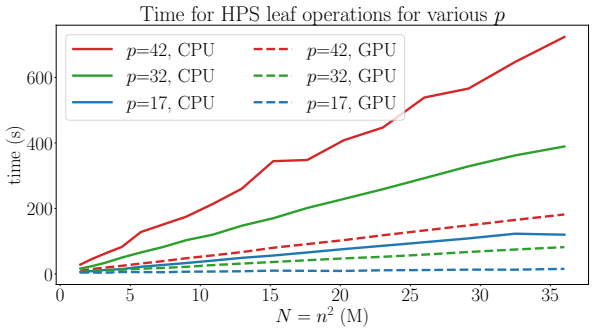


Fig. 3 Leaf operations for HPS require $O(p^4N)$ operations, though the practical scaling for parallel operations has a small constant prefactor for p up to 42. Parallel HPS leaf operations are further accelerated on GPUs, with a speed-up of at least 4x.

We show that GPU optimizations enable efficient leaf operations for various local polynomial orders, cf. Figure 3. After the leaf operations, we directly factorize the reduced system (5) using efficient sparse direct solvers. We demonstrate that the choice of p does not significantly affect the time to factorize $\tilde{\mathbf{A}}$. Having the freedom to choose p allows the user to resolve highly oscillatory PDEs to high-order accuracy without worrying about how the choice may affect the cost of solving (3) directly.

To demonstrate the effectiveness of the HPS discretization resolving oscillatory solutions to high accuracy, we report results for a PDE with a known analytic solution

$$\begin{cases} -\Delta u(x) - \kappa^2 u(x) = 0, & x \in \Omega = [0,1]^2, \\ u(x) = u_{\text{true}}(x), & x \in \partial\Omega, \end{cases} \tag{7}$$

The true solution u_{true} is given by $u_{\text{true}} = J_0\left(\kappa|x - (-0.1, 0.5)|\right)$, where $x \mapsto J_0(\kappa|x|)$ is the free-space fundamental solution to the Helmholtz equation. We discretize (7) using HPS for various choices of p and set the wavenumber κ to increase with N to maintain 10 points per wavelength with increasing problem size. After applying a direct solver to solve (5) on the reduced HPS grid, we re-factorize the linear systems on interior leaf nodes to calculate the solution $\mathbf{u}_{\text{calc}}$ on the full HPS grid. The leaf solve requires time $O(p^4N)$ but is particularly efficient using the GPU optimizations described. The reported build times and solve times include the leaf operations. We report the relative error with respect to the residual of the discretized system (3). When a true solution is known, we also report the relative error with respect to the true solution $\mathbf{u}_{\text{true}}$ evaluated on the collocation points of the full HPS grid

$$\text{relerr}_{\text{res}} = \frac{\|\mathbf{A}\mathbf{u}_{\text{calc}} - \mathbf{f}\|_2}{\|\mathbf{f}\|_2}, \qquad \text{relerr}_{\text{true}} = \frac{\|\mathbf{u}_{\text{calc}} - \mathbf{u}_{\text{true}}\|_2}{\|\mathbf{u}_{\text{true}}\|_2}. \tag{8}$$

3.1 Comparison of sparse direct solvers

The system (5) is solved using two different sparse direct solvers, SuperLU and SlabLU. SuperLU is a black-box solver that finds an appropriate ordering of the system to minimize fill-in while increasing concurrency by grouping nodes into

super-nodes [15]. We accessed SuperLU through the Scipy interface (version 1.8.1) and called it with the COLAMD ordering. This version of Scipy uses the CPU only. Not many GPU-aware sparse direct solvers are widely available, though this is an active area of research. SuperLU uses a pivoting scheme that can exchange rows between super-nodes to attain almost machine precision accuracy in the residual of the computed solutions.

SlabLU, on the other hand, uses an ordering based on a decomposition of the domain into slabs that has a limited pivoting scheme. Despite this limitation, SlabLU can achieve 10 digits of accuracy in the residual, which also gives high-order true relative accuracy, depending on the choice of p. SlabLU is a simple two-level framework that achieves large speedups over SuperLU by leveraging batched BLAS and GPU optimizations. Figure 4 provides a comparison between SuperLU and SlabLU in factorizing $\tilde{\mathbf{A}}$ to solve (5). The figure shows that SlabLU is faster to build and requires a smaller memory footprint than SuperLU, where the memory footprint refers to how much main memory is required to store the sparse factorization of $\tilde{\mathbf{A}}$. Figure 5 presents a comparison of the accuracy of computed solutions when using SlabLU and SuperLU.

Next we demonstrate the ability of HPS, combined with SlabLU, for various p to solve Helmholtz problems of size up to $900\lambda \times 900\lambda$ (for which N=81M) to high-order accuracy. Figure 6 reports build and solve times for various choices of p, and Figure 7 reports the accuracy of the calculated solutions.

3.2 Convergence for scattering problems for various p

We will now demonstrate the ability of HPS, combined with SlabLU as a sparse direct solver, to solve complex scattering phenomena on various 2D domains. For the presented PDEs, we will show how the accuracy of the calculated solution converges to a reference solution depending on the choice of p in the discretization. Specifically, we will solve the BVP (1) with the variable-coefficient Helmholtz operator (2) for various Dirichlet data on smooth and rectangular domains.

We fix the PDE and refining the mesh to compare calculated solutions to a reference solution obtained on a fine mesh with high p, as the exact solution is unknown. The relative error is calculated by comparing $\mathbf{u}_{\text{calc}}$ to the reference solution $\mathbf{u}_{\text{ref}}$ at a small number of collocation points $\{x_j\}_{j=1}^{M}$ using the l_2 norm

$$\text{relerr}_{\text{approx}} = \frac{\|\mathbf{u}_{\text{calc}} - \mathbf{u}_{\text{ref}}\|_2}{\|\mathbf{u}_{\text{ref}}\|_2}. \tag{9}$$

We demonstrate the convergence on a unit square domain $\Omega = [0, 1]^2$ with a variable coefficient field b_{crystal} corresponding to a photonic crystal, shown in Figure 8. The convergence plot is presented in Figure 9.

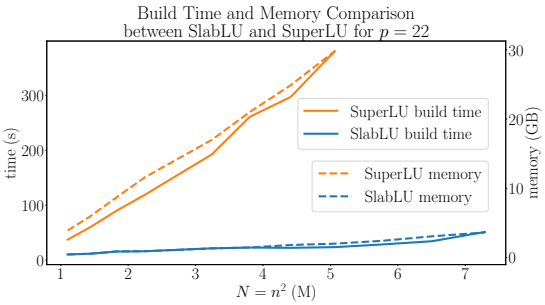


Fig. 4 Build time and memory footprint comparison between SuperLU and SlabLU for (7) discretized with HPS (p=22), where κ is increased to maintain 10 points per wavelength. For N=5.06M, the SlabLU build time is faster by a factor of 16x and the memory footprint is less by a factor of 14x.

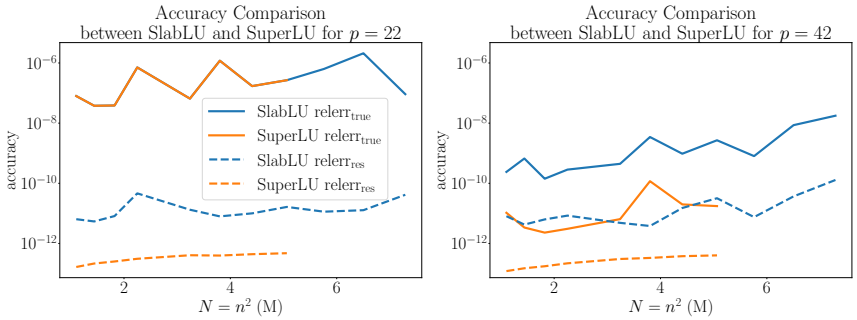


Fig. 5 Accuracy comparison between SuperLU and SlabLU for (7) discretized with HPS (p=22,42), where κ is increased to maintain 10 points per wavelength. SuperLU uses a more sophisticated pivoting scheme to achieve high accuracy in the residual error. For p=22, SlabLU and SuperLU both resolve the solution to 6 digits in the true relative accuracy. For p=42, SuperLU is able to achieve 10 digits in the true relative accuracy, while SlabLU achieves 8 digits.

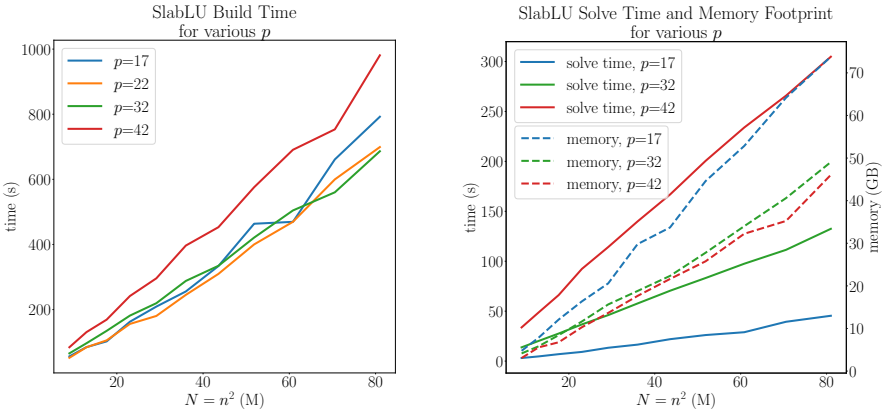


Fig. 6 Build time and solve time for HPS with various p for (7) where κ is increased with N to maintain 10 points per wavelength. The choice of p does not substantially affect the time needed to factorize the sparse linear system with SlabLU. As p increases, the memory footprint required to store the factorization decreases and the solve time increases.

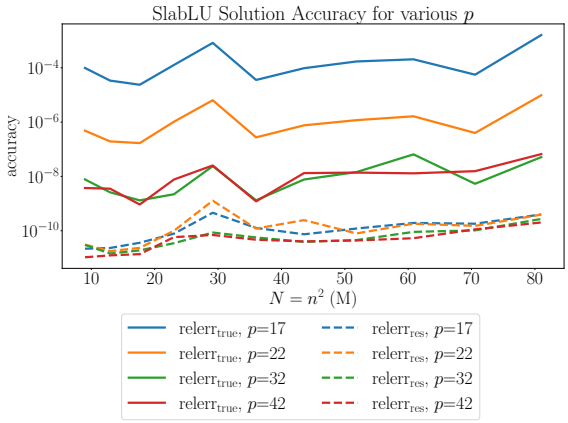
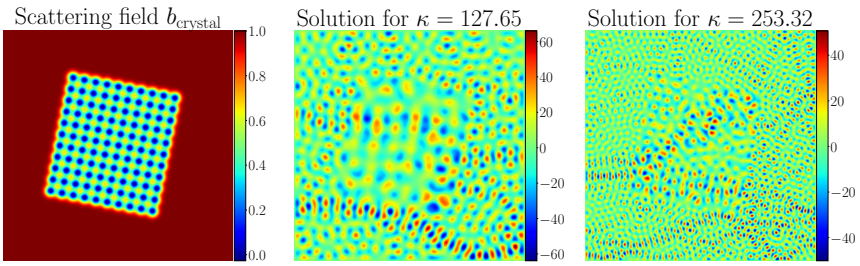


Fig. 7 Solution accuracy for HPS with various p for (7) where κ is increased with N to maintain 10 points per wavelength. Regardless of the choice of p, SlabLU resolves the solution to at least 10 digits of relative accuracy in the residual. With increasing p, one can calculate solutions with higher relative accuracy, compared to the true solution of the PDE.



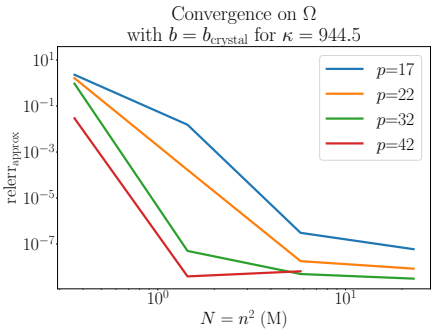


Fig. 8 Solutions of variable-coefficient Helmholtz problem on square domain Ω with Dirichlet data given by $u \equiv 1$ on $\partial\Omega$ for various wavenumbers κ. The scattering field is b_{crystal}, which is a photonic crystal that blocks the wave from propagating.

Fig. 9 Convergence on square domain Ω for reference solution $\mathbf{u}_{\text{ref}}$ on HPS discretization for N=36M with $p = 42$.

Next, we show the convergence on a curved domain Ψ with a constant-coefficient field $b \equiv 1$, where Ψ is given by an analytic parameterization over a reference square $\Omega = [0,1]^2$. The domain Ψ is parametrized as

$$\Psi = \left\{ \left(x_1, \frac{x_2}{\psi(x_1)} \right) \text{ for } (x_1, x_2) \in \Omega = [0,1]^2 \right\}, \text{ where } \psi(z) = 1 - \frac{1}{4}\sin(z). \quad (10)$$

Using the chain rule, (2) on Ψ takes the following form on Ω

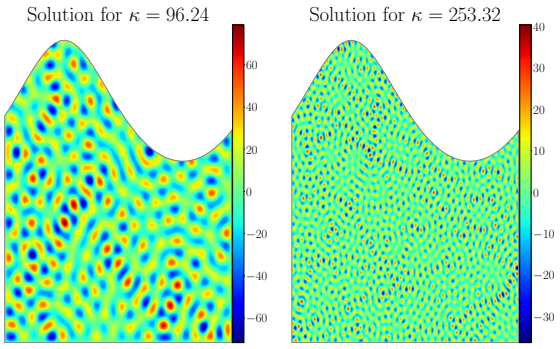


Fig. 10 Solutions of constant-coefficient Helmholtz problem on curved domain Ψ with Dirichlet data given by $u \equiv 1$ on $\partial\Psi$ for various wavenumbers κ. The solutions are calculated parametrizing Ψ in terms of a reference square domain Ω as (10) and solving (11) on Ω.

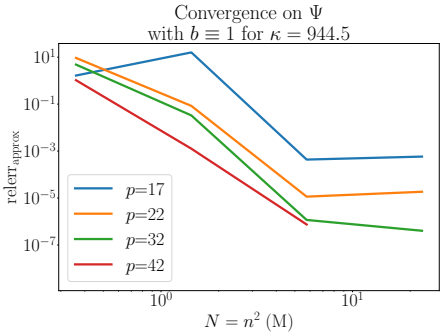


Fig. 11 Convergence on curved domain Ψ for reference solution $\mathbf{u}_{\text{ref}}$ on HPS discretization for N=36M with $p = 42$. The solutions exhibit mildly singular behavior near the corners, and choosing high orders of p aids in the convergence.

$$-\frac{\partial^2 u}{\partial x_1^2} - 2\frac{\psi'(x_1)x_2}{\psi(x_1)}\frac{\partial^2 u}{\partial x_1 \partial x_2} - \left(\left(\frac{\psi'(x_1)x_2}{\psi(x_1)}\right)^2 + \psi(x_1)^2\right)\frac{\partial^2 u}{\partial x_2^2} - \frac{\psi''(x_1)x_2}{\psi(x_1)}\frac{\partial u}{\partial x_2} - \kappa^2 u = 0, \qquad (x_1, x_2) \in \Omega. \tag{11}$$

The solutions on Ψ are shown in Figure 10, and the convergence plot is presented in Figure 11.

4 Conclusions

HPS is a high-order convergent discretization scheme that interfaces well with sparse direct solvers. In this manuscript, we describe GPU optimizations of the scheme that enable rapid and memory-efficient direct solutions of (3) for resulting linear systems. First, we perform the leaf operations in parallel using batched BLAS. Then, we factorize a smaller system (5) of size $\approx N/p$ using sparse direct solvers, where p denotes the local order of convergence, which we show can be chosen as high as 42. The numerical results feature comparisons between sparse direct solvers and demonstrate that SlabLU can factorize systems corresponding to domains of size up to $900\lambda \times 900\lambda$ (for which N=81M) in less than 20 minutes. The approach is effective in resolving challenging scattering problems on various domains to high accuracy.

The techniques described are currently being implemented for three dimensional problems. The parallelizations described are immediately applicable. The scaling with p deteriorates from $O(p^4N)$ to $O(p^6N)$, which limits how large p can be chosen. However, initial numerical experiments demonstrate that $p = 15$ remains viable on current hardware, which is high enough for most applications.

Acknowledgements The work reported was supported by the Office of Naval Research (N00014-18-1-2354), by the National Science Foundation (DMS-1952735 and DMS-2012606), and by the Department of Energy ASCR (DE-SC0022251).

References

1. Babb, T., Gillman, A., Hao, S., and Martinsson, P.-G. An Accelerated Poisson Solver based on Multidomain Spectral Discretization. *BIT Numerical Mathematics* **58**, 851–879 (2018).
2. Bériot, H., Prinn, A., and Gabard, G. Efficient Implementation of High-Order Finite Elements for Helmholtz Problems. *International Journal for Numerical Methods in Engineering* **106**(3), 213–240 (2016).
3. Brubeck, P. D. and Farrell, P. E. A Scalable and Robust Vertex-Star Relaxation for High-Order FEM. *SIAM Journal on Scientific Computing* **44**(5), A2991–A3017 (2022).
4. Cockburn, B. Static Condensation, Hybridization, and the Devising of the HDG Methods. In: *Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations*, 129–177. Springer (2016).
5. Davis, T. A. *Direct Methods for Sparse Linear Systems*, vol. 2. SIAM (2006).
6. Davis, T. A., Rajamanickam, S., and Sid-Lakhdar, W. M. A Survey of Direct Methods for Sparse Linear Systems. *Acta Numerica* **25**, 383 – 566 (2016).
7. Deraemaeker, A., Babuška, I., and Bouillard, P. Dispersion and Pollution of the FEM Solution for the Helmholtz Equation in One, Two and Three Dimensions. *International Journal for Numerical Methods in Engineering* **46**(4), 471–499 (1999).
8. Dongarra, J., Hammarling, S., Higham, N. J., Relton, S. D., Valero-Lara, P., and Zounon, M. The Design and Performance of Batched BLAS on Modern High-Performance Computing Systems. *Procedia Computer Science* **108**, 495–504 (2017).
9. Ernst, O. G. and Gander, M. J. Why It Is Difficult to Solve Helmholtz Problems with Classical Iterative Methods. *Numerical Analysis of Multiscale Problems* 325–363 (2012).
10. Fortunato, D., Hale, N., and Townsend, A. The Ultraspherical Spectral Element Method. *Journal of Computational Physics* **436**, 110087 (2021).
11. Gillman, A. and Martinsson, P.-G. A Direct Solver with O(N) Complexity for Variable Coefficient Elliptic PDEs Discretized via a High-Order Composite Spectral Collocation Method. *SIAM Journal on Scientific Computing* **36**(4), A2023–A2046 (2014).
12. Guyan, R. J. Reduction of Stiffness and Mass Matrices. *AIAA journal* **3**(2), 380–380 (1965).
13. Hao, S. and Martinsson, P.-G. A Direct Solver for Elliptic PDEs in Three Dimensions based on Hierarchical Merging of Poincaré–Steklov Operators. *Journal of Computational and Applied Mathematics* **308**, 419–434 (2016).
14. Levitt, J. and Martinsson, P.-G. Linear-Complexity Black-Box Randomized Compression of Hierarchically Block Separable Matrices. *arXiv preprint arXiv:2205.02990* (2022).
15. Li, X. S. and Shao, M. A Supernodal Approach to Incomplete LU Factorization with Partial Pivoting. *ACM Transactions on Mathematical Software (TOMS)* **37**(4), 1–20 (2011).
16. Martinsson, P.-G. A direct solver for variable coefficient elliptic PDEs discretized via a composite spectral collocation method. *Journal of Computational Physics* **242**, 460–479 (2013).
17. Martinsson, P.-G. *Fast Direct Solvers for Elliptic PDEs*, *CBMS-NSF Conference Series*, vol. CB96. SIAM (2019).
18. Yesypenko, A. and Martinsson, P.-G. SlabLU: A Two-Level Sparse Direct Solver for Elliptic PDEs. *arXiv preprint arXiv:2211.07572* (2022).