

Nonlinear Two-Level Schwarz Methods: A Parallel Implementation in FROSch

Alexander Heinlein^[0000-0003-1578-8104]

Kyrill Ho^[0009-0001-4522-5647]

Axel Klawonn^[0000-0003-4765-7387]

Martin Lanser^[0000-0002-4232-9395]

1 Introduction and Nonlinear Problems

There is a growing body of work on nonlinear domain decomposition methods (DDMs). The interest in this class of methods can be attributed to their favorable properties for solving highly nonlinear problems, together with their inherent parallelizability. One class of nonlinear DDMs is that of nonlinear Schwarz methods [11, 2, 7, 15, 16, 3] on which we focus here. Previous parallel implementations of nonlinear Schwarz methods have not taken advantage of the GDSW (Generalized Dryja-Smith-Widlund)-type coarse spaces [5], for example [13] and [17]. The two-level implementations that incorporate this family of coarse space have been serial prototype codes implemented in high-level scripting languages such as MATLAB. Following the favorable results seen in [11] for the two-level nonlinear Schwarz method with energy-minimizing coarse spaces, we have implemented a parallel two-level nonlinear Schwarz solver based on the FROSch (Fast and Robust Overlapping Schwarz) solver framework. FROSch is part of Trilinos and implements linear multilevel Schwarz methods with GDSW-type coarse spaces. It has been shown to scale efficiently for linear and nonlinear problems as a preconditioner for an iterative linear solver [10]. In this article, we introduce the new nonlinear Schwarz implementation and discuss the first scalability results for a scalar nonlinear diffusion problem and a 2D nonlinear Neo-Hooke elasticity problem.

Alexander Heinlein¹, Kyrill Ho³, Axel Klawonn^{2,3}, Martin Lanser^{2,3}

¹Delft University of Technology, Delft Institute of Applied Mathematics, Mekelweg 4, 2628 CD Delft, Netherlands, e-mail: a.heinlein@tudelft.nl

²Department 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, url: <https://www.numerik.uni-koeln.de>

³Center for Data and Simulation Science, University of Cologne, Germany, e-mail: kyrill.ho@uni-koeln.de, url: <https://www.cds.uni-koeln.de>

Nonlinear Problems

As stated above, the first of two nonlinear problems that we consider is a scalar nonlinear diffusion problem

$$\begin{aligned} -\nabla((u^2 + 1)\nabla u) &= 1 \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{1}$$

where Ω is the unit square. As a second model problem we consider a vector valued 2D nonlinear elasticity problem using a compressible plane-stress Neo-Hooke material model: find $u \in (H_0^1(\Omega))^2$, such that

$$\begin{aligned} -\operatorname{div}(P(F)) &= f_{vol} \text{ in } \Omega, \\ u &= g_D \text{ on } \partial\Omega_D, \\ n \cdot P(F) &= g_N \text{ on } \partial\Omega_N, \end{aligned} \tag{2}$$

where F is the deformation gradient, and Ω_D and Ω_N represent the Dirichlet and Neumann boundaries, respectively. The first Piola-Kirchhoff stress tensor is

$$P(F) = \frac{E}{(1 + \nu)}(F - F^{-T}) + \frac{E\nu}{(1 + \nu)(1 - 2\nu)}\ln(\det(F))F^{-T}.$$

We set the Poisson ratio to $\nu = 0.3$ and Young's modulus to $E = 210$ GPa. To model a simple beam, we choose the domain Ω to be a rectangle with *width* = 5m and *height* = 1m. We fix the beam on either end by setting $g_D = 0$ and $g_N = 0$. We apply a volume force in the vertical direction by setting $f_{vol} = (0, -f_y)$ for some scalar $f_y > 0$.

2 A Parallel implementation of Nonlinear Schwarz Methods

In this section, we provide an overview of the expressions that describe the one-level nonlinear Schwarz method that was first introduced in [2] and its extension to two-level methods described in [11] and [8]. In addition, we introduce our initial parallel implementation. We use the same definition for the subdomains Ω_i , their corresponding function spaces V_i , the prolongation $P_i : V_i \mapsto V$ and restriction $R_i : V \mapsto V_i$ operators, the coarse space V_0 and associated operators $P_0 : V_0 \mapsto V$ and $R_0 : V \mapsto V_0$ as used in [11]. In our tests we use the RGDSW (Reduced GDSW) coarse space with the option 2.2 as introduced in [6] but with a slight modification which is described in detail in [12].

To describe the one-level nonlinear Schwarz method we begin with an abstract discretized nonlinear problem

$$F(u) = 0 \tag{3}$$

and define local nonlinear correction terms $T_i(u) : V \mapsto V_i$ as the solution of the local nonlinear problems

$$R_i F(u - P_i T_i(u)) = 0, \quad i = 1, \dots, N. \quad (4)$$

Combining these nonlinear correction terms results in the alternative problem

$$\mathcal{F}_1(u) = \sum_{i=1}^N P_i T_i(u) = 0. \quad (5)$$

The one-level nonlinear Schwarz method results by solving Equation (5) using Newton's method. It is denoted ASPIN (Additive Schwarz Preconditioned Inexact Newton) in [2] because the tangent $D\mathcal{F}_1(u)$ is evaluated approximately. Here we always evaluate the tangent exactly yielding the ASPEN (Additive Schwarz Preconditioned Exact Newton) method. We thus have two nested Newton's methods. In the *outer loop* the alternative problem $\mathcal{F}_1(u) = 0$ is solved. In each iteration of the outer loop, the local nonlinear corrections $T_i(u^{(k)})$ must be computed to evaluate $\mathcal{F}_1(u^{(k)})$. This is done in *local inner loops* using Newton's method on each overlapping subdomain Ω_i to solve the local nonlinear problems (4). The outer loop requires the evaluation of the tangent $D\mathcal{F}_1(u)$ in each iteration. Taking the derivative of Equation (4) results in the following expression for the tangent

$$D\mathcal{F}_1(u) = \sum_{i=1}^N P_i DT_i(u) = \sum_{i=1}^N P_i (R_i DF(u_i) P_i)^{-1} R_i DF(u_i) \quad (6)$$

with $u_i := u - P_i T_i(u)$. To simplify the expression for the tangent we introduce

$$Q_i(u) := P_i (R_i DF(u) P_i)^{-1} R_i DF(u) \quad (7)$$

and write $D\mathcal{F}_1(u) = \sum_{i=1}^N Q_i(u_i)$. In this article, we prefer to work with a restricted Schwarz approach called RASPEN (Restricted ASPEN) which was described in [4]. This method introduces restricted prolongation operators $\tilde{P}_i : V_i \mapsto V$, $i = 1, \dots, N$, that are chosen to fulfill the partition of unity property

$$\sum_{i=1}^N \tilde{P}_i R_i = I. \quad (8)$$

To extend the one-level method, we define a coarse nonlinear correction $T_0(u) : V \mapsto V_0$ as the solution of the coarse nonlinear problem

$$R_0 F(u - P_0 T_0(u)) = 0. \quad (9)$$

Coupling of the nonlinear coarse correction with the local nonlinear corrections can be done in a variety of ways. Here we only describe the two possibilities for which we later show results. For further possibilities, see, e.g., [11]. The first way of coupling

is a simple additive approach in which the alternative problem is defined as

$$\mathcal{F}_a(u) = \sum_{i=1}^N P_i T_i(u) + P_0 T_0(u) = 0. \quad (10)$$

The second approach is a multiplicative hybrid approach, denoted $h1$ in [11] but which we denote here simply as h with

$$\mathcal{F}_h(u) = \sum_{i=1}^N P_i T_i(u - P_0 T_0(u)) + P_0 T_0(u) = 0. \quad (11)$$

The tangents of both methods are given by

$$D\mathcal{F}_a(u) = \sum_{i=0}^N Q_i(u_i) \quad \text{and} \quad D\mathcal{F}_h(u) = \sum_{i=1}^N Q_i(v_i)(I - Q_0(u_0)) + Q_0(u_0) \quad (12)$$

with the previous definition of u_i expanded to include $i = 0$ and $v_i := u_0 - P_i T_i(u_0)$ for $i = 1, \dots, N$. Our current parallel implementation is based on extending the functionality of the linear Schwarz operators of the Trilinos package FROSch. We refer to [9] and [10] for an overview of the various FROSch operators and an explanation of how GDSW coarse basis functions are built. Constructing RGDSW coarse basis functions uses the same mechanisms. Here, we give a brief overview of the current state of our implementation.

The nonlinear Schwarz solver requires finite element assembly routines to re-assemble the tangent and residual periodically. Currently, we use the C++ package FEDDLib (Finite Element and Domain Decomposition Library) [19] for this and other preprocessing tasks such as mesh generation and distribution. The outer loop of the nonlinear Schwarz solver is simply an implementation of Newton's method in which the tangent system

$$D\mathcal{F}_x(u^{(k)})\delta u = \mathcal{F}_x(u^{(k)}) \quad (13)$$

is solved in each iteration with $x \in \{1, a, h\}$. This requires evaluation of the nonlinear operator $\mathcal{F}_x(u)$ and formation of the tangent $D\mathcal{F}_x(u)$ at an arbitrary linearization point u .

In our implementation, we have separated the evaluation of the residual into various operators that all inherit from existing FROSch Schwarz operators. The one-level residual \mathcal{F}_1 is evaluated by the class `NonLinearSchwarzOperator` which inherits from `SchwarzOperator`. The coarse component $P_0 T_0(u)$ of the two-level residual is evaluated by the `CoarseNonLinearSchwarzOperator` class. The tangent is similarly compartmentalized into the `OverlappingOperator` class which is an original FROSch class, and the `SimpleCoarseOperator` class which inherits from `CoarseOperator`.

To construct arbitrary two-level nonlinear Schwarz methods we have implemented a pair of abstract operators `NonLinearCombineOperator` and `CombineOperator`

that define the residual $\mathcal{F}_x(u)$ and the tangent $D\mathcal{F}_x(u)$, respectively. Specializations of these classes define the order in which local and coarse operators are combined. Employing this interface, we have implemented the three methods (one-level, two-level additive, and two-level hybrid) and can easily implement further operators for other multiplicative and hybrid two-level methods.

Using our current nonlinear Schwarz implementation is very similar to using a classical Newton-Krylov-Schwarz solver. The user must provide the assembly routines for the residual and tangent of the problem they wish to solve, configure the solver parameters, and start the solver. For now, the assembly routines must be implemented within a `FEDDLib NonLinearProblem` class object. This is because the assembly routines for the residual and tangent are currently expected to use the interface prescribed therein. We plan to make our implementation more accessible in the future by providing the same assembly routine interface as the Trilinos nonlinear solver package `NOX`.

The key difference to using a Newton-Krylov-Schwarz solver is that an overlapping mesh partitioning is required. The overlapping partition must be element based rather than node based, that is, use the dual graph of the mesh. This facilitates assembly on the overlapping subdomains since finite element assembly routines generally perform quadrature on a per-element basis. Since the linear Schwarz preconditioner works algebraically, it only requires a node-based overlapping partitioning. We have implemented routines in the `FEDDLib` to build an overlapping distributed dual graph using `METIS/ParMETIS` [14] and distribute the mesh correspondingly.

3 Parallel Numerical Results

The parallel results shown in this section were generated using the NHR supercomputer Fritz; see the Acknowledgements for further details.

To solve the global tangent system (13) we use `GMRES` provided by the Trilinos package `Belos` [1]. We set the relative `GMRES` tolerance to 10^{-6} , the maximum number of iterations to 1000, and the maximum number of restarts to 20. In the inner Newton methods, we use Intel `MKL Pardiso` as a sparse direct solver [18].

As a proof of concept to ensure that our implementation of the two-level nonlinear Schwarz variants is working as expected, we first solve the simple nonlinear diffusion problem introduced in Section 1. We do not compare the nonlinear Schwarz variants with a classical Newton-Krylov-Schwarz approach for this problem as it is too simple and we do not expect performance improvements from nonlinear Schwarz methods. We test weak parallel scalability of the one-level, the two-level additive, and the two-level hybrid nonlinear Schwarz methods. We use subdomains with 300 elements in each dimension on a regular square mesh, set the relative tolerance of the outer Newton method to 10^{-4} , the absolute and relative tolerances of the inner Newton methods to 10^{-11} and 10^{-5} respectively, and use an overlap of one. In the two-level variants, we use the `RGDSW` coarse space [6]. As an initial value, we set the solution to zero everywhere.

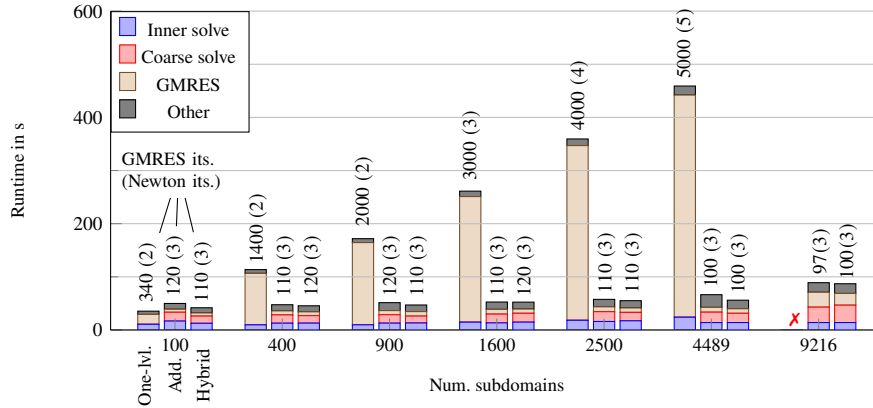


Fig. 1 Time to solution of the one-level and two-level additive and hybrid nonlinear Schwarz variants for the nonlinear diffusion problem (1). The outer Newton iterations and total number of GMRES iterations summed over all outer iterations is shown above each column.

Figure 1 demonstrates the necessity of a coarse level even for a simple nonlinear problem. The number of outer nonlinear iterations is not affected by whether a coarse space is used or not and remains fairly constant around three iterations. However, the condition number of the tangent $D\mathcal{F}(u)$ deteriorates with more subdomains in the one-level case and results in an increasing number of GMRES iterations. The increase in the time to solution of the one-level method clearly results directly from the increase in the number of GMRES iterations. The perfect weak scalability of the two-level variants is probably due to the fact that the nonlinear diffusion problem is simple enough that the coarse correction generates a very good approximation of the solution. We expect to see an increase in the time to solution of the two-level variants when the coarse problem grows large enough.

As a second problem that demonstrates the potential utility of nonlinear Schwarz variants we consider the beam test based on the two dimensional plane-stress Neo-Hooke model (2) introduced in Section 1. Again, we use the Fritz cluster and 576 CPU cores for a METIS decomposition into 576 subdomains. We make the problem incrementally more difficult for the nonlinear solver by increasing the volume force. We use a mesh consisting of 2.6×10^6 nodes and solve for the resulting deformation. We set the maximum number of GMRES iterations to 100, the overlap to 10, and otherwise use the same settings for the nonlinear Schwarz variants as for the previous problem. For the Newton-Krylov-Schwarz solver we also set the relative tolerance to 10^{-4} and use the same settings for the GMRES solver. However, in this case we employ a (linear) two-level Schwarz preconditioner using the RGDSW coarse space and an overlap of 5. The overlaps are chosen differently because nonlinear Schwarz uses the dual graph to determine the overlap, while linear Schwarz uses the nodal graph. Thus, the physical overlap is approximately equal. See [12] for details.

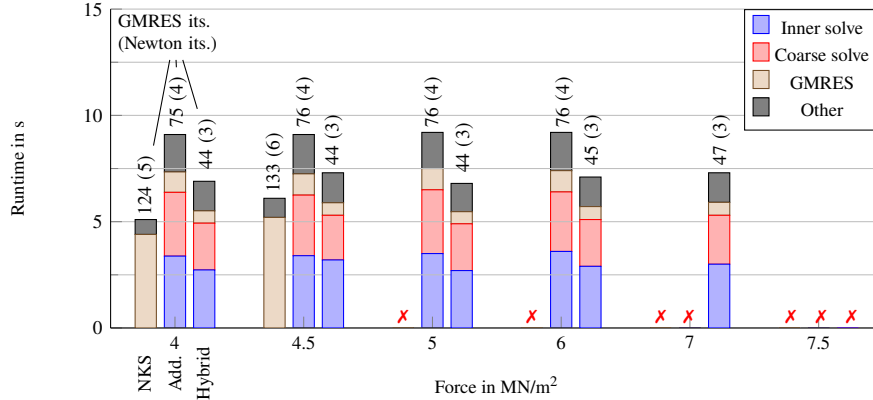


Fig. 2 Time to solution of a classical Newton-Krylov-Schwarz (NKS) solver, two-level additive and two-level hybrid nonlinear Schwarz variants for the nonlinear elasticity problem (2) solved using 576 subdomains. The outer Newton iterations and total number of GMRES iterations summed over all outer iterations is shown above each column. A red cross indicates that the solver failed to converge.

Figure 2 shows that the two-level nonlinear Schwarz variants are more robust as they are able to solve the problem for larger forces. The hybrid method is more robust than the additive method which coincides with the observations made previously in our MATLAB testing environment.

To summarize, we have shown some initial results from our parallel implementation of the nonlinear Schwarz variants demonstrating that it is working as expected. The weak scalability of the two-level methods is superior to the classical ASPEN approach and for nonlinear elasticity problems with large deformations they are more robust and have a similar time-to-solution as the Newton-Krylov-Schwarz solver for small test problems. Based on this implementation we plan to investigate the weak scalability of the nonlinear Schwarz variants compared to that of the classical Newton-Krylov-Schwarz method in future work.

Acknowledgements The authors would like to gratefully thank Lea Saßmannshausen (University of Cologne) for her help using the FEDDLib software library and Sharan Nurani Ramesh (Ruhr University Bochum) for his help with the hyperelasticity problem. The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under the NHR project k107ce. NHR funding is provided by federal and Bavarian state authorities. NHR@FAU hardware is partially funded by the German Research Foundation (DFG) - 440719683. Finally the authors acknowledge the financial support by the German Federal Ministry of Education and Research (BMBF) in the program SCALEXA.

References

1. Bavier, E., Hoemmen, M., Rajamanickam, S., Thornquist, H.: Amesos2 and Belos: Direct and iterative solvers for large sparse linear systems. *Scientific Programming* **20**, 241–255 (2012)
2. Cai, X.C., Keyes, D.E.: Nonlinearly preconditioned inexact Newton algorithms. *SIAM J. Sci. Comput.* **24**(1), 183–200 (2002). URL <https://doi.org/10.1137/S106482750037620X>
3. Cai, X.C., Keyes, D.E., Marcinkowski, L.: Non-linear additive Schwarz preconditioners and application in computational fluid dynamics. pp. 1463–1470 (2002). URL <https://doi.org/10.1002/fld.404>. LMS Workshop on Domain Decomposition Methods in Fluid Mechanics (London, 2001)
4. Cai, X.C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing* **21**(2), 792–797 (1999). URL <https://doi.org/10.1137/S106482759732678X>
5. Dohrmann, C.R., Klawonn, A., Widlund, O.B.: Domain decomposition for less regular subdomains: Overlapping Schwarz in two dimensions. *SIAM Journal on Numerical Analysis* **46**(4), 2153–2168 (2008). URL <https://doi.org/10.1137/070685841>
6. Dohrmann, C.R., Widlund, O.B.: On the design of small coarse spaces for domain decomposition algorithms. *SIAM Journal on Scientific Computing* **39**(4), A1466–A1488 (2017). URL <https://doi.org/10.1137/17M114272>
7. Dolean, V., Gander, M.J., Kheriji, W., Kwok, F., Masson, R.: Nonlinear preconditioning: how to use a nonlinear Schwarz method to precondition Newton’s method. *SIAM J. Sci. Comput.* **38**(6), A3357–A3380 (2016). URL <https://doi.org/10.1137/15M102887X>
8. Heinlein, A., Klawonn, A., Lanser, M.: Adaptive nonlinear domain decomposition methods with an application to the p -Laplacian. *SIAM J. Sci. Comput.* **45**(3), S152–S172 (2023). URL <https://doi.org/10.1137/21M1433605>
9. Heinlein, A., Klawonn, A., Rajamanickam, S., Rheinbach, O.: FROSch: A Fast And Robust Overlapping Schwarz Domain Decomposition Preconditioner Based on Xpetra in Trilinos, pp. 176–184. Springer International Publishing (2020)
10. Heinlein, A., Klawonn, A., 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**(6), C713–C747 (2016)
11. Heinlein, A., Lanser, M.: Additive and hybrid nonlinear two-level Schwarz methods and energy minimizing coarse spaces for unstructured grids. *SIAM J. Sci. Comput.* **42**(4), A2461–A2488 (2020). URL <https://doi.org/10.1137/19M1276972>
12. Ho, K., Klawonn, A., Lanser, M.: A Parallel Implementation of Two-level Nonlinear Schwarz Methods - An Application to Hyperelasticity and Incompressible Flow Problems. In preparation.
13. Kamthorncharoen, C.: Two-level restricted additive Schwarz preconditioned exact newton with applications. Ph.d. thesis, Ruprecht-Karls-University, Heidelberg, Germany (2022). URL <https://archiv.ub.uni-heidelberg.de/volltextserver/32296/>
14. Karypis, G., Schloegel, K., Kumar, V.: Parmetis: Parallel graph partitioning and sparse matrix ordering library (1997)
15. Liu, L., Gao, W., Yu, H., Keyes, D.E.: Overlapping multiplicative Schwarz preconditioning for linear and nonlinear systems. *J. Comput. Phys.* **496**, Paper No. 112548, 22 (2024). URL <https://doi.org/10.1016/j.jcp.2023.112548>
16. Liu, L., Keyes, D.E.: Field-split preconditioned inexact Newton algorithms. *SIAM J. Sci. Comput.* **37**(3), A1388–A1409 (2015). URL <https://doi.org/10.1137/140970379>
17. Marcinkowski, L., Cai, X.C.: Parallel performance of some two-level aspin algorithms. pp. 639–646 (2005)
18. Schenk, O., Gärtner, K.: Solving unsymmetric sparse systems of linear equations with PARDISO. *Future Generation Computer Systems* **20**(3), 475–487 (2004). URL <https://www.sciencedirect.com/science/article/pii/S0167739X03001882>. Selected numerical algorithms
19. The FEDDLib Team: The open-source FEDDLib library. GitHub repository (2024). URL <https://github.com/FEDDLib/FEDDLib>. Homepage: <https://feddlib.github.io/>