

Algebraic Adaptive Multipreconditioning applied to Restricted Additive Schwarz

Nicole Spillane¹

In 2006 the Multipreconditioned Conjugate Gradient (MPCG) algorithm was introduced by Bridson and Greif [4]. It is an iterative linear solver, adapted from the Preconditioned Conjugate Gradient (PCG) algorithm [22], which can be used in cases where several preconditioners are available or the usual preconditioner is a sum of operators. In [4] it was already pointed out that Domain Decomposition algorithms are ideal candidates to benefit from MPCG. This was further studied in [13] which considers Additive Schwarz preconditioners in the Multipreconditioned GMRES (MPGMRES) [14] setting. In 1997, Rixen had proposed in his thesis [21] the Simultaneous FETI algorithm which turns out to be MPCG applied to FETI. The algorithm is more extensively studied in [12] where its interpretation as an MPCG solver is made explicit.

The idea behind MPCG is that if at a given iteration N preconditioners are applied to the residual, then the space spanned by all of these directions is a better minimization space than the one spanned by their sum. This can significantly reduce the number of iterations needed to achieve convergence, as we will observe in Section 3, but comes at the cost of losing the short recurrence property in CG. This means that at each iteration the new search directions must be orthogonalized against all previous ones. For this reason, in [25] it was proposed to make MPCG into an Adaptive MPCG (AMPCG) algorithm where, at a given iteration, only the contributions that will accelerate convergence are kept, and all others are added into a global contribution (as they would be in classical PCG). This works very well for FETI and BDD but the theory in that article does not apply to Additive Schwarz. Indeed, the assumption is made that the smallest eigenvalue of the (globally) preconditioned operator is known. The test (called the τ -test), which chooses at each iteration which contributions should be kept, heavily relies on it. More precisely, the quantity that is examined by the τ -test can be related

CMA, École Polytechnique, Route de Saclay, 91128 Palaiseau, France.
nicole.spillane@cmap.polytechnique.fr

to a Rayleigh quotient, and the vectors that are selected to form the next minimization space correspond to large frequencies of the (globally) preconditioned operator. These are exactly the ones that are known to slow down convergence of BDD and FETI. Moreover, they are generated by the first few iterations of PCG [30]. These two reasons make BDD and FETI ideal for the AMPCG framework.

The question posed by the present work is whether an AMPCG algorithm can be developed for Additive Schwarz type preconditioners. The goal is to design an adaptive algorithm that is robust at a minimal cost. One great feature of Additive Schwarz is that it is algebraic (all the components in the preconditioner can be computed from the knowledge of the matrix \mathbf{A}), and we will aim to preserve this property. The algorithms will be presented in an abstract framework. Since the short recurrence property is lost anyway in the MPCG setting, we will consider the more efficient [11] Restricted Additive Schwarz preconditioner (RAS) [6] in our numerical experiments, instead of its symmetric counterpart the Additive Schwarz preconditioner (see [29]). RAS is a non symmetric preconditioner but, provided that full recurrence is used, conjugate gradient based algorithms apply and still have nice properties (in particular the global minimization property). We will detail this in the next section where we briefly introduce the problem at hand, the Restricted Additive Schwarz preconditioner, and the MPCG solver. Then in Section 2, we propose two ways to make MPCG adaptive. Finally, Section 3 presents some numerical experiments on matrices arising from the finite element discretization of two dimensional elasticity problems. Three types of difficulties will be considered: heterogeneous coefficients, automatic (irregular) partitions into subdomains and almost incompressible behaviour.

These are sources of notoriously hard problems that have been, and are still, at the heart of much effort in the domain decomposition community, in particular by means of choosing an adequate coarse spaces (see [23, 20, 26, 10, 3, 24, 27, 15, 19, 5, 8, 18] and many more).

1 Preliminaries

Throughout this work, we consider the problem of solving the linear system

$$\mathbf{A}\mathbf{x}_* = \mathbf{b},$$

where $\mathbf{A} \in \mathbb{R}^{n \times n}$ is a sparse symmetric positive definite matrix, $\mathbf{b} \in \mathbb{R}^n$ is a given right hand side, and $\mathbf{x}_* \in \mathbb{R}^n$ is the unknown. We consider Conjugate Gradient type solvers preconditioned by the Restricted Additive Schwarz (RAS) preconditioner. To construct the RAS preconditioner, a non overlapping partition of the degrees of freedom into N subsets, or subdomains, must first be chosen and then overlap must be added to each subset to get an

overlapping partition. Denoting for each $s = 1, \dots, N$, by $\tilde{\mathbf{R}}^s$ and \mathbf{R}^s , the restriction operators from $\llbracket 1, n \rrbracket$ into the s -th non overlapping and overlapping subdomains, respectively, the preconditioner is defined as:

$$\mathbf{H} = \sum_{s=1}^N \mathbf{H}^s \text{ with } \mathbf{H}^s = \tilde{\mathbf{R}}^{s\top} \mathbf{A}^{s-1} \mathbf{R}^s \text{ and } \mathbf{A}^s = \mathbf{R}^s \mathbf{A} \mathbf{R}^{s\top}.$$

In Algorithm 1 the MPCG iterations are defined. Each contribution \mathbf{H}^s to \mathbf{H} is treated separately. This corresponds to the non adaptive algorithm, *i.e.*, the condition in line 8 is not satisfied and N search directions are added to the minimization space at each iteration (namely the columns in \mathbf{Z}_{i+1}). We have denoted by Δ_i^\dagger the pseudo inverse of Δ_i to account for the fact that some search directions may be linearly dependent (see [12, 25]).

Although RAS is a non symmetric preconditioner the following properties hold:

- $\|\mathbf{x}_* - \mathbf{x}_i\|_{\mathbf{A}} = \min \left\{ \|\mathbf{x}_* - \mathbf{x}\|_{\mathbf{A}}; \mathbf{x} \in \mathbf{x}_0 + \sum_{j=0}^{i-1} \text{range}(\mathbf{P}_j) \right\},$
- $\mathbf{P}_j^\top \mathbf{A} \mathbf{P}_i = \mathbf{0}$ ($i \neq j$), $\mathbf{r}_i^\top \mathbf{P}_j = \mathbf{0}$ ($i > j$), and $\mathbf{r}_i^\top \mathbf{H} \mathbf{r}_j = 0$ ($i > j$).

This can be proved easily following similar proofs in [25] and the textbook [22]. The difference from the symmetric case is that the two last properties only hold for $i > j$, and not for every pair $i \neq j$.

Algorithm 1: Adaptive Multipreconditioned Conjugate Gradient Algorithm for $\mathbf{A}\mathbf{x}_* = \mathbf{b}$. Preconditioners: $\{\mathbf{H}^s\}_{s=1,\dots,N}$. Initial guess: \mathbf{x}_0 .

```

1  $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ ;  $\mathbf{Z}_0 = [\mathbf{H}^1 \mathbf{r}_0 | \dots | \mathbf{H}^N \mathbf{r}_0]$ ;  $\mathbf{P}_0 = \mathbf{Z}_0$ ;
2 for  $i = 0, 1, \dots$ , convergence do
3    $\mathbf{Q}_i = \mathbf{A} \mathbf{P}_i$ ;
4    $\Delta_i = \mathbf{Q}_i^\top \mathbf{P}_i$ ;  $\gamma_i = \mathbf{P}_i^\top \mathbf{r}_i$ ;  $\alpha_i = \Delta_i^\dagger \gamma_i$ ;
5    $\mathbf{x}_{i+1} = \mathbf{x}_i + \mathbf{P}_i \alpha_i$ ;
6    $\mathbf{r}_{i+1} = \mathbf{r}_i - \mathbf{Q}_i \alpha_i$ ;
7    $\mathbf{Z}_{i+1} = [\mathbf{H}^1 \mathbf{r}_{i+1} | \dots | \mathbf{H}^N \mathbf{r}_{i+1}]$ ; // Generate  $N$  search directions.
8   if Adaptive Algorithm then
9     | Reduce number of columns in  $\mathbf{Z}_{i+1}$  (see Section 2);
10  end
11   $\Phi_{i,j} = \mathbf{Q}_j^\top \mathbf{Z}_{i+1}$ ;  $\beta_{i,j} = \Delta_j^\dagger \Phi_{i,j}$  for each  $j = 0, \dots, i$ ;
12   $\mathbf{P}_{i+1} = \mathbf{Z}_{i+1} - \sum_{j=0}^i \mathbf{P}_j \beta_{i,j}$ ;
13 end
14 Return  $\mathbf{x}_{i+1}$ ;
```

Multipreconditioning significantly improves convergence as has already been observed [4, 13, 12, 25] and as will be illustrated in the numerical result section. The drawback is that a dense matrix $\Delta_i \in \mathbb{R}^{N \times N}$ must be factorized at each iteration and that N search directions per iteration need to be stored.

In the next section, we will try to remove these limitations by reducing the number of search directions at every iteration. We aim to do this without having too strong a negative impact on the convergence.

2 An adaptive algorithm

There is definitely a balance to be found between the number of iterations, the cost of each iteration, and the memory required for storage. Here, we do not claim that we have achieved a perfect balance, but we introduce some ways to influence it. More precisely, we propose two methods of reducing the number of columns in \mathbf{Z}_{i+1} (or in other words how to fill in line 9 in Algorithm 1). In subsection 2.1, we propose a τ -test that measures the relevance of every *candidate* $\mathbf{H}^s \mathbf{r}_{i+1}$ and only keeps the most relevant contributions. In Subsection 2.2, we propose to form m coarser subdomains (which are *agglomerates* of the initial N subdomains) and *aggregate* the N *candidates* into only m search directions. Note that there is a definite connection with multigrid studies from where we have borrowed some vocabulary (see [31, 7, 2] and many references therein).

2.1 Select search directions with a τ -test

The τ -test in the original AMPCG publication [25] is based on the assumption that the smallest eigenvalue for the globally preconditioned operator $\mathbf{H}\mathbf{A}$ is known [29]. This allows for an error estimate inspired by those in [1], and the choice of the τ -test is a direct consequence of it. Here, the largest eigenvalue is known and it is the presence of small eigenvalues that is responsible for slow convergence. Unfortunately, we have failed to produce an estimate similar to that in [25] in this case. Note that there is no such estimate in [1] either, and we believe that this is inherent to the properties of the conjugate gradient algorithm.

The approach that we propose here to select local contributions is different. It is well known by now (see, *e.g.*, [22]) that, at each iteration, the approximate solution returned by the conjugate gradient algorithm is the \mathbf{A} -orthogonal projection of the exact solution \mathbf{x}_* onto the minimization space. Here, the property satisfied by the update between in iteration $i + 1$ is

$$\|\mathbf{x}_* - \mathbf{x}_{i+1}\|_{\mathbf{A}} = \min \{ \|\mathbf{x}_* - \mathbf{x}\|_{\mathbf{A}}; \mathbf{x} \in \mathbf{x}_i + \text{range}(\mathbf{P}_i) \},$$

where \mathbf{P}_i forms a basis of $\text{range}(\mathbf{Z}_i)$ after orthogonalization against previous search spaces (line 12 in Algorithm 1).

For this reason, the τ -test that we propose aims at evaluating, for each $s = 1, \dots, N$, the ratio between the norm of the error projected onto the global vector $\mathbf{H}\mathbf{r}_{i+1}$ and the norm of the error projected onto the local candidate $\mathbf{H}^s\mathbf{r}_{i+1}$. More precisely, we compute (with $\langle \cdot, \cdot \rangle$ denoting the ℓ_2 inner product)

$$t_i^s = \frac{\langle \mathbf{r}_{i+1}, \mathbf{H}\mathbf{r}_{i+1} \rangle^2}{\langle \mathbf{H}\mathbf{r}_{i+1}, \mathbf{A}\mathbf{H}\mathbf{r}_{i+1} \rangle} \times \frac{\langle \mathbf{H}^s\mathbf{r}_{i+1}, \mathbf{A}\mathbf{H}^s\mathbf{r}_{i+1} \rangle}{\langle \mathbf{r}_{i+1}, \mathbf{H}^s\mathbf{r}_{i+1} \rangle^2}. \quad (1)$$

This is indeed the announced quantity, since the square of the \mathbf{A} -norm of the \mathbf{A} orthogonal projection of $\mathbf{x}_* - \mathbf{x}_i$ onto any vector \mathbf{v} is

$$\|\mathbf{v}(\mathbf{v}^\top \mathbf{A} \mathbf{v})^{-1} \mathbf{v}^\top \underbrace{\mathbf{A}(\mathbf{x}_* - \mathbf{x}_i)}_{=\mathbf{r}_i}\|_{\mathbf{A}}^2 = \frac{\langle \mathbf{r}_i, \mathbf{v} \rangle^2}{\langle \mathbf{v}, \mathbf{A} \mathbf{v} \rangle}.$$

Then, given a threshold τ , the number of columns in \mathbf{Z}_{i+1} is reduced by eliminating all those for which $t_i^s > \tau$. In order for the global preconditioned residual $\sum_{s=1}^N \mathbf{H}^s\mathbf{r}_{i+1}$ to be included in the search space (as is always the case in PCG), we add it to \mathbf{Z}_{i+1} in a separate column. This way we obtain a minimization space range(\mathbf{P}_i) of dimension anywhere between 1 and N .

An important question is of course how to choose τ . Considering that t_i^s measures the (inverse of the) impact of one of N contributions compared to the impact of the sum of the N contributions, it is quite natural to choose $\tau \approx N$. In the next section, we illustrate the behaviour of the adaptive algorithm with the τ -test for values of τ ranging between $N/10$ and $10N$ with satisfactory results.

In order to determine whether or not $t_i^s \leq \tau$ (i.e., perform the τ -test) it is necessary to compute t_i^s . Here, we will not discuss how to do this at the smallest cost but it is of course an important consideration (that was discussed for the AMPCG algorithm applied to BDD in [25]). One noteworthy observation is that if \mathbf{H} were either the Additive Schwarz (AS), or the Additive Schwarz with Harmonic overlap (ASH [6]) preconditioner (i.e., $\mathbf{H} = \sum_{s=1}^N \mathbf{R}^s \mathbf{A}^{s-1} \mathbf{R}^s$ or $\mathbf{H} = \sum_{s=1}^N \mathbf{R}^s \mathbf{A}^{s-1} \tilde{\mathbf{R}}^s$) then all terms involving $\mathbf{H}^s \mathbf{A} \mathbf{H}^s$ would simplify since, obviously, $\mathbf{A}^{s-1} \mathbf{R}^s \mathbf{A} \mathbf{R}^s \mathbf{A}^{s-1} = \mathbf{A}^{s-1}$.

Another option is to prescribe a number m of vectors to be selected at each iteration instead of a threshold τ , and keep the m vectors with smallest values of t_i^s . Then, only the second factor in (1) would be required. We leave a more in depth study of these questions for future work.

2.2 Aggregate search directions

Here, we propose a completely different, and much simpler, way of reducing the number of vectors in \mathbf{Z}_{i+1} . This is to choose a prescribed number m ,

with $m \leq N$, of search directions per iteration, and a partition of $\llbracket 1, N \rrbracket$ into m subsets. Then, the columns of \mathbf{Z}_{i+1} that correspond to the same subset are simply replaced by their sum, leaving m vectors. We refer to this as aggregation as it is the same as assembling coarse domains from the original subdomains and computing coarse search directions as sums of the \mathbf{H}_{i+1}^s . The question of how to choose m is of course important. It can be a fraction of N or the maximal size of the dense matrix that the user is prepared to factorize. In the next section, we consider values ranging from $N/20$ to N .

3 Numerical Results with FreeFem++ [16] and GNU Octave [9]

In this section, we consider the linear elasticity equations posed in $\Omega = [0, 1]^2$ with mixed boundary conditions. We search for $\mathbf{u} = (u_1, u_2)^\top \in H^1(\Omega)^2$ such that

$$\begin{cases} -\operatorname{div}(\sigma(\mathbf{u})) = (0, 0)^\top, & \text{in } \Omega, \\ \mathbf{u} = (1/2(y(1-y)), 0)^\top, & \text{on } \{(x, y) \in \partial\Omega : x = 0\}, \\ \mathbf{u} = (-1/2(y(1-y)), 0)^\top, & \text{on } \{(x, y) \in \partial\Omega : x = 1\}, \\ \sigma(\mathbf{u}) \cdot \mathbf{n} = 0, & \text{on the rest of } \partial\Omega \text{ (}\mathbf{n}\text{: outward normal)}. \end{cases}$$

The stress tensor $\sigma(\mathbf{u})$ is defined by $\sigma_{ij}(\mathbf{u}) = 2\mu\varepsilon_{ij}(\mathbf{u}) + \lambda\delta_{ij}\operatorname{div}(\mathbf{u})$ for $i, j = 1, 2$ where $\varepsilon_{ij}(\mathbf{u}) = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$, δ_{ij} is the Kronecker symbol and the Lamé coefficients are functions of Young's modulus E and Poisson's ratio ν : $\mu = \frac{E}{2(1+\nu)}$, $\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}$. In all test cases, ν is uniform and equal either to 0.4 (compressible test case) or 0.49999 (almost incompressible test case) while E varies between 10^6 and 10^{12} in a pattern presented in Figure 1–left. The geometries of the solutions are also presented in this figure.

The computational domain is discretized into a uniform mesh with mesh size: $h = 1/60$, and partitioned into $N = 100$ subdomains by the automatic graph partitioner METIS [17]. One layer of overlap is added to each subdomain. In the compressible case, the system is discretized by piecewise second order polynomial (\mathbb{P}_2) Lagrange finite elements. In the almost incompressible setting it is known that the locking phenomenon occurs rendering the solution unreliable. To remedy this, the problem is rewritten in a mixed formulation with an additional unknown $p = \operatorname{div}(u)$, and then discretized. Although the $\mathbb{P}_2\text{--}\mathbb{P}_0$ mixed finite element does not satisfy the discrete inf-sup condition it is often used in practice, and we choose it here. Finally, the pressure unknowns are eliminated by static condensation.

In both cases the problem has 28798 degrees of freedom (once degrees of freedom corresponding to Dirichlet boundary conditions have been eliminated). As an initial guess, we first compute a random vector \mathbf{v} and then scale

it to form $\mathbf{x}_0 = \frac{\mathbf{b}^\top \mathbf{v}}{\|\mathbf{v}\|_A^2}$, according to what is proposed in [28]. This guarantees that $\|\mathbf{x}_* - \mathbf{x}_0\|_A \leq \|\mathbf{x}_*\|_A$: the initial error is at most as large as it would be with a zero initial guess.

In table 1, we report on the number of iterations needed to reduce the initial error $\|\mathbf{x}_* - \mathbf{x}_0\|_A$ by a factor 10^{-7} and on the size of the minimization space that was constructed to do this, which is $\sum_i \text{rank}(\mathbf{P}_i)$. Note that, although they are presented in the same table, we cannot compare the compressible and incompressible test cases as they are simply not the same problem. Figures 2, 3, 4 and 5 show in more detail the convergence behaviour of each method.

The first point to be made is that the MPCG algorithm does an excellent job at reducing the number of iterations. This can be observed by looking at the data for $m = 100 = N$ directions per iteration in Figures 4 and 5. The iteration counts are reduced from 889 to 60 and from over 999 to 56 compared to the classical PCG iterations ($m = 1$ direction per iteration). Secondly the adaptation steps that we introduced seem to do their job since they ensure fast convergence with smaller minimization spaces. In particular, all of these adaptive methods converged in less than 512 iterations even for the incompressible case (for which the usual PCG still has a relative error of $8 \cdot 10^{-4}$ after 999 iterations).

With the τ -test, the number of iterations is always reduced by a factor at least 8 compared to PCG even with the smallest threshold $\tau = 10 = N/10$. With $\tau = 10N$ the number of iterations is almost the same as with the full MPCG. For these test cases the choice $\tau = N$ advocated in Section 2 seems to be a good compromise.

With the aggregation procedure, convergence is achieved even when the coarsening is quite aggressive (5 vectors per iteration means that 20 local contributions have been added together to form the search direction). As expected, keeping more vectors per iteration yields significantly better results in terms of iteration count.

Based on these results, it is not possible to compare the two approaches and future work will definitely be focused on an optimized implementation and on decreasing the CPU time.

4 Conclusions and Future Work

In this work, we have implemented the MPCG [4, 13] algorithm for Restricted Additive Schwarz. We have observed very good convergence on test cases with known difficulties (heterogeneities and almost incompressible behaviour). This is a confirmation that multipreconditioning is a valuable tool to improve robustness. The main focus of this article has been to propose an adaptive version of the algorithm so that, when possible, the cost of each

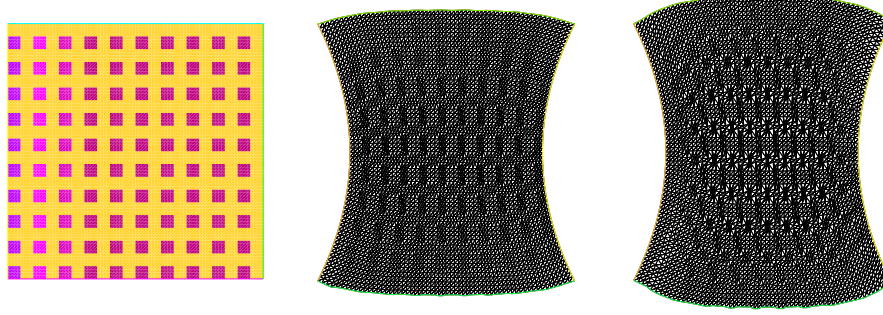


Fig. 1 Test case setup (all three configurations are drawn to scale). Left: Young's modulus – $E = 10^6$ with square inclusions of larger E , up to 10^{12} . Middle: Solution for $\nu = 0.4$. Right: Solution for $\nu = 0.49999$.

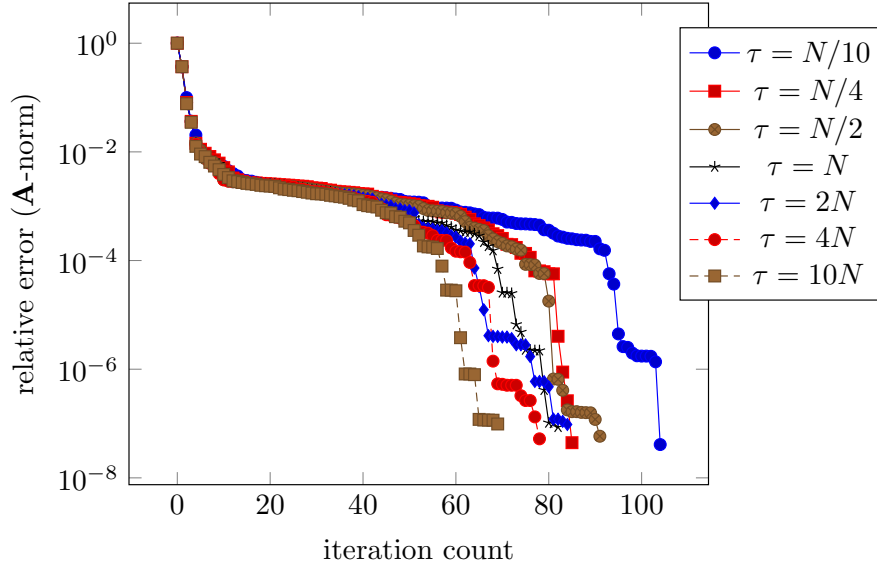


Fig. 2 Compressible test case – reducing the number of directions with the τ -test – error norm versus iteration count for different values of τ

iteration and the cost of storage can be reduced while maintaining fast convergence. To this end, we have introduced two methods to reduce the number of search directions at each iteration: one is based on the so called τ -test, and the other on adding some local components together. Numerical results have confirmed that both these approaches behave as expected.

One important feature of the algorithms proposed is that they are completely algebraic in that they can be applied to any symmetric, positive definite matrix \mathbf{A} without any extra knowledge.

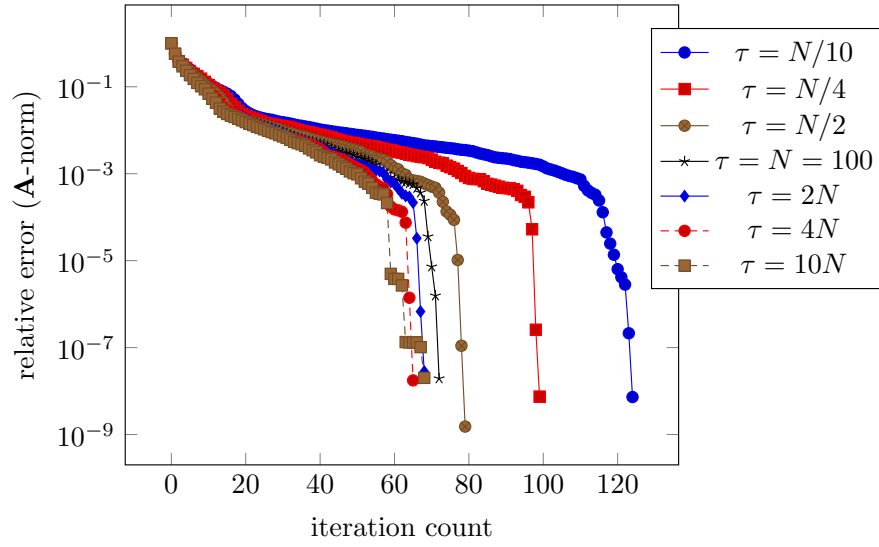


Fig. 3 Incompressible test case – reducing the number of directions with the τ -test – error norm versus iteration count for different values of τ

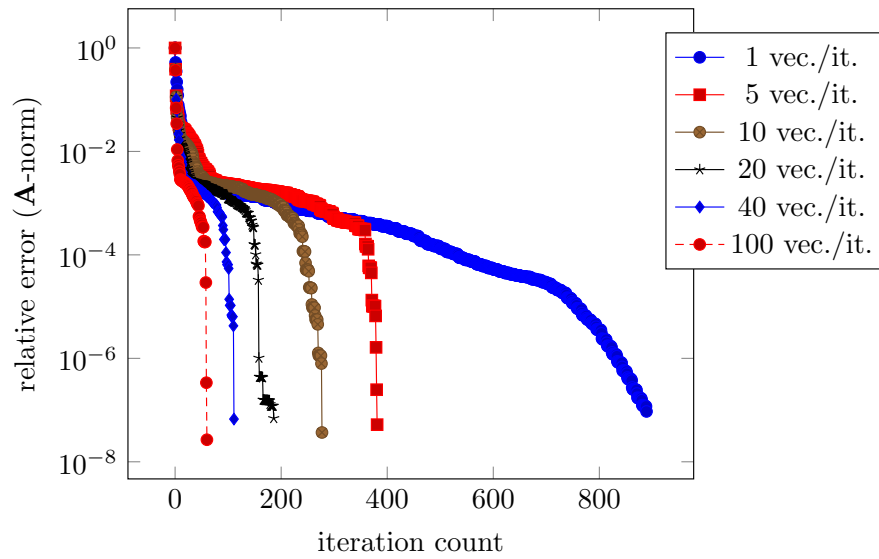


Fig. 4 Compressible test case – reducing the number of directions by aggregating them into m vectors – error norm versus iteration count for different values of m

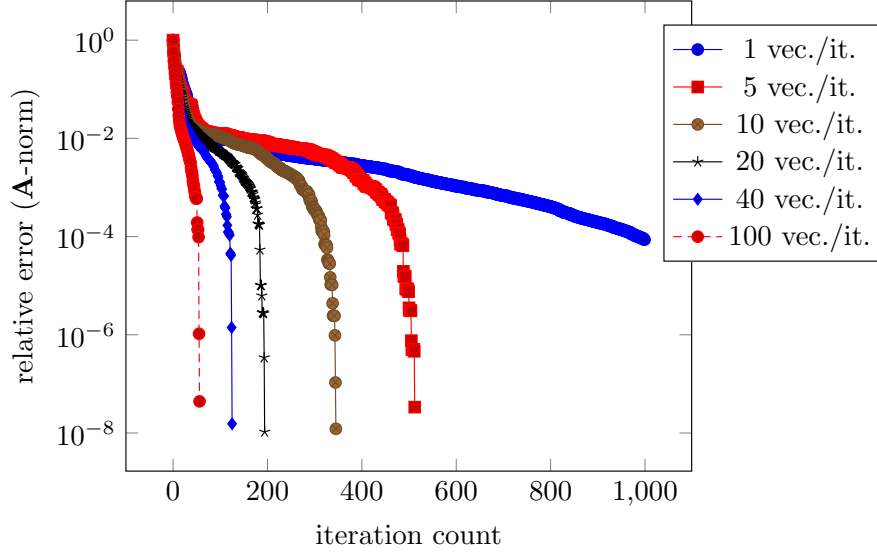


Fig. 5 Incompressible test case – reducing the number of directions by aggregating them into m vectors – error norm versus iteration count for different values of m

Compressible						Incompressible					
τ -test (see Fig. 2)			Aggregates (see Fig. 4)			τ -test (see Fig. 3)			Aggregates (see Fig. 5)		
τ	iter.	# vec.	m	iter.	# vec.	τ	iter.	# vec.	m	iter.	# vec.
10	104	6059	1	889	890	10	124	4865	1	> 999	>1000
25	85	5769	5	381	1910	25	99	4889	5	512	2565
50	91	6625	10	277	2780	50	79	4621	10	345	3460
100	82	6339	20	186	3740	100	72	4521	20	194	3900
200	84	6876	40	111	4480	200	68	4593	40	125	5040
400	78	6817	100	60	6100	400	65	4552	100	56	5700
1000	69	6153				1000	68	5156			

Table 1 Summary of all numerical results presented. *iter.*: number of iterations needed to reduce the initial error by a factor 10^{-7} . *# vec.*: size of the minimization space. There are two test cases: Compressible and Incompressible, and for each there are two ways of reducing the number of search directions at each iteration: with the τ -test (as proposed in Subsection 2.1) or by aggregating into m directions (as proposed in Subsection 2.2).

An optimized parallel implementation is the subject of ongoing work in order to compare MPCG and the two AMPCG algorithms in terms of CPU time. Scalability must also be measured. The author is quite confident that the *best* AMPCG algorithm should be a combination of the two adaptive approaches. Additionally there is no reason why the components that are added together in the aggregation procedure should not first be weighted by some optimized coefficients, turning the algorithm into a multilevel one.

References

- [1] O. Axelsson and I. Kaporin. Error norm estimation and stopping criteria in preconditioned conjugate gradient iterations. *Numer. Linear Algebra Appl.*, 8(4):265–286, 2001.
- [2] A. Brandt, J. Brannick, K. Kahl, and I. Livshits. Bootstrap AMG. *SIAM J. Sci. Comput.*, 33(2):612–632, 2011.
- [3] M. Brezina, C. Heberton, J. Mandel, and P. Vaněk. An iterative method with convergence rate chosen a priori. Technical Report 140, University of Colorado Denver, April 1999.
- [4] R. Bridson and C. Greif. A multipreconditioned conjugate gradient algorithm. *SIAM J. Matrix Anal. Appl.*, 27(4):1056–1068 (electronic), 2006.
- [5] M. Cai, L. F. Pavarino, and O. B. Widlund. Overlapping Schwarz methods with a standard coarse space for almost incompressible linear elasticity. *SIAM Journal on Scientific Computing*, 37(2):A811–A830, 2015.
- [6] X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21(2):792–797 (electronic), 1999.
- [7] T. Chartier, R. D. Falgout, V. E. Henson, J. Jones, T. Manteuffel, S. McCormick, J. Ruge, and P. S. Vassilevski. Spectral AMGe (ρ AMGe). *SIAM J. Sci. Comput.*, 25(1):1–26, 2003.
- [8] C. R. Dohrmann and O. B. Widlund. Hybrid domain decomposition algorithms for compressible and almost incompressible elasticity. *Internat. J. Numer. Methods Engrg.*, 82(2):157–183, 2010.
- [9] J. W. Eaton, D. Bateman, and S. Hauberg. *GNU Octave version 3.0.1 manual: a high-level interactive language for numerical computations*. CreateSpace Independent Publishing Platform, 2009. ISBN 1441413006.
- [10] Y. Efendiev, J. Galvis, R. Lazarov, and J. Willems. Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms. *ESAIM Math. Model. Numer. Anal.*, 46(5):1175–1199, 2012.
- [11] E. Efsthathiou and M. J. Gander. Why restricted additive Schwarz converges faster than additive Schwarz. *BIT*, 43(suppl.):945–959, 2003.
- [12] P. Gosselet, D. Rixen, F.-X. Roux, and N. Spillane. Simultaneous FETI and block FETI: Robust domain decomposition with multiple search directions. *Internat. J. Numer. Methods Engrg.*, 104(10):905–927, 2015.
- [13] C. Greif, T. Rees, and D. Szyld. Additive Schwarz with variable weights. In *Domain Decomposition Methods in Science and Engineering XXI*. Springer, 2014.
- [14] C. Greif, T. Rees, and D. B. Szyld. MPGMRES: a generalized minimum residual method with multiple preconditioners. Technical Report 11-12-23, Temple University, 2014.
- [15] R. Haferssas, P. Jolivet, and F. Nataf. A robust coarse space for optimized Schwarz methods: SORAS-GenEO-2. *C. R. Math. Acad. Sci. Paris*, 353(10):959–963, 2015.

- [16] F. Hecht. *FreeFem++*. Numerical Mathematics and Scientific Computation. Laboratoire J.L. Lions, Université Pierre et Marie Curie, <http://www.freefem.org/ff++/>, 3.23 edition, 2013.
- [17] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM J. Sci. Comput.*, 20(1):359–392 (electronic), 1998.
- [18] A. Klawonn, M. Kühn, and O. Rheinbach. Adaptive coarse spaces for FETI-DP in three dimensions. Technical report, Submitted, 2016.
- [19] A. Klawonn, P. Radtke, and O. Rheinbach. FETI-DP methods with an adaptive coarse space. *SIAM J. Numer. Anal.*, 53(1):297–320, 2015.
- [20] F. Nataf, H. Xiang, V. Dolean, and N. Spillane. A coarse space construction based on local Dirichlet-to-Neumann maps. *SIAM J. Sci. Comput.*, 33(4):1623–1642, 2011.
- [21] D. Rixen. *Substructuring and Dual Methods in Structural Analysis*. PhD thesis, Université de Liège, Collection des Publications de la Faculté des Sciences appliquées, n.175, 1997.
- [22] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2003.
- [23] M. Sarkis. Partition of unity coarse spaces. In *Fluid flow and transport in porous media: mathematical and numerical treatment (South Hadley, MA, 2001)*, volume 295 of *Contemp. Math.*, pages 445–456. Amer. Math. Soc., Providence, RI, 2002.
- [24] B. Sousedík, J. Šístek, and J. Mandel. Adaptive-Multilevel BDDC and its parallel implementation. *Computing*, 95(12):1087–1119, 2013.
- [25] N. Spillane. An Adaptive Multipreconditioned Conjugate Gradient Algorithm. *SIAM J. Sci. Comput.*, 38(3):A1896–A1918, 2016.
- [26] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, and R. Scheichl. Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. *Numer. Math.*, 126(4):741–770, 2014.
- [27] N. Spillane and D. J. Rixen. Automatic spectral coarse spaces for robust FETI and BDD algorithms. *Int. J. Numer. Meth. Engng.*, 95(11):953–990, 2013.
- [28] Z. Strakoš and P. Tichý. Error estimation in preconditioned conjugate gradients. *BIT*, 45(4):789–817, 2005.
- [29] A. Toselli and O. Widlund. *Domain decomposition methods—algorithms and theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2005.
- [30] A. van der Sluis and H. A. van der Vorst. The rate of convergence of conjugate gradients. *Numer. Math.*, 48(5):543–560, 1986.
- [31] P. S. Vassilevski. *Multilevel block factorization preconditioners*. Springer, New York, 2008. Matrix-based analysis and algorithms for solving finite element equations.