

# Additive Schwarz with Variable Weights

Chen Greif<sup>1</sup>, Tyrone Rees<sup>2</sup>, and Daniel B. Szyld<sup>3</sup>

## 1 Introduction and Motivation

We consider the numerical solution of nonsymmetric linear systems of equations of the form

$$A\mathbf{u} = \mathbf{f}, \quad (1)$$

that arise from the discretization of partial differential equations (PDEs). In practical problems, the number of mesh points is very large, and thus also the number of unknowns in (1), and the resulting matrix is large and sparse. In these circumstances, iterative methods are often used, due to their ability to deal more effectively with a high degree of sparsity. A popular iterative method is the *Generalized Minimum Residual* iterative scheme, or GMRES [8, 9, 10]. This method is based on minimizing at the  $k$ th iterate the residual within the affine Krylov subspace  $\mathbf{u}_0 + \mathcal{K}^k(A, \mathbf{r}_0)$ , where  $\mathbf{u}_0$  is an initial vector,  $\mathbf{r}_0 = \mathbf{f} - A\mathbf{u}_0$  is the initial residual, and

$$\mathcal{K}^k(A, \mathbf{r}_0) = \text{span}(\mathbf{r}_0, A\mathbf{r}_0, \dots, A^{k-1}\mathbf{r}_0).$$

The performance of GMRES is often (though not exclusively) determined by the structure of the eigenvalues of the matrix  $A$ . Loosely speaking, if they are strongly clustered, then GMRES is expected to converge fast. To accomplish a clustering effect, a *preconditioner*  $M$  is typically used: instead of solving (1) we solve, say,

$$AM\tilde{\mathbf{u}} = \mathbf{f},$$

where  $M$  is constructed so that  $AM$  has a more favorable eigenstructure than  $A$ . Upon incorporating the preconditioner  $M$ , the Krylov subspace changes accordingly: the matrix associated with the subspace becomes  $AM$ , and the preconditioned residual is now minimized.

A common way of dealing with the large number of degrees of freedom in a fine mesh is to break the problem down into a number of more manageable sub-problems. This amounts to the technique of *domain decomposition*; see, e.g., [11]. We can then incorporate preconditioners that work on the subdomains into the general iterative framework.

The additive Schwarz preconditioner [11] and its restricted variant (RAS) [3], can be written in the form

---

<sup>1</sup> Department of Computer Science, University of British Columbia, Vancouver, B.C., Canada. e-mail: greif@cs.ubc.ca <sup>2</sup> Scientific Computing Department, STFC Rutherford Appleton Laboratory, Chilton, Didcot, UK. e-mail: tyrone.rees@stfc.ac.uk <sup>3</sup> Department of Mathematics, Temple University, Philadelphia, Pennsylvania, USA. e-mail: szyld@temple.edu

$$M = \sum_{i=1}^t \tilde{R}_i A_i^{-1} R_i^T,$$

where  $t$  is usually the number of subdomains,  $\tilde{R}_i$  is a restriction operator,  $R_i^T$  is a prolongation operator, and  $A_i = R_i^T A R_i$  is the restriction of  $A$  onto the  $i$ th subdomain.

A possible generalization would be to use a weighted additive or restricted additive Schwarz preconditioner, say of the form

$$M^{(k)} = \sum_{i=1}^t \alpha_i^{(k)} \tilde{R}_i A_i^{-1} R_i^T,$$

where the weights  $\alpha_i^{(k)}$  are chosen at the  $k$ th iteration of GMRES so as to minimize the preconditioned residual, cf. [1]<sup>1</sup>. What we propose in this paper is to go a step further, and implicitly find at each iteration both the current weights and all the weights at the previous iterations, so as to minimize the residual at the current step.

Incorporating weights which change from one iteration to the next is significant and we can no longer talk about a standard iterative method with a single preconditioner. Instead, the proposed strategy fits into the MPMGMRES paradigm the authors recently described in [5], where more than one preconditioner may be applied simultaneously.<sup>2</sup> Our main goal in this paper is to show that this methodology is particularly effective in the domain decomposition paradigm, since we can associate each subdomain with a specific, unique preconditioner.

An outline of the remainder of this paper follows. In Section 2 we briefly describe Additive and Restricted Additive Schwarz Preconditioning. In Section 3 we describe the MPMGMRES algorithm. We address the question of computational cost of the algorithm and characterize the generalized Krylov subspace and its unique features in domain decomposition setting. In Section 4 we provide some details on numerical experiments. Finally, in Section 5 we make some concluding remarks.

## 2 Additive Schwarz Preconditioning

Suppose we divide the domain  $\Omega$  containing  $n$  nodes into  $t$  subdomains  $\Omega_1, \dots, \Omega_t$ , which overlap by bands of width  $\delta$  nodes. Suppose each subdomain consists of  $m_i \ll n$  nodes, which we denote as the entries of the set  $I_i$ . We can define a prolongation matrix  $R_{i,\delta}^T \in \mathbb{R}^{n \times m_i}$  which extends vectors  $\mathbf{u}^{(i)} \in \mathbb{R}^{m_i}$  to  $\mathbb{R}^n$  by

$$(R_{i,\delta}^T \mathbf{u}^{(i)})_k = \begin{cases} (\mathbf{u}^{(i)})_k & \text{if } k \in I_i \\ 0 & \text{otherwise.} \end{cases}$$

<sup>1</sup> We point out that this is completely different than the approach in [4], where the weights are zeros and ones, and the emphasis is on asynchronous iterations.

<sup>2</sup> This algorithm extends previous work on using a combination of preconditioners – e.g., flexible GMRES [7] with alternating preconditioners, as described by Rui *et al.* [6] in the method they call multipreconditioned GMRES – by making an ‘optimal’ choice of weights. See [5] for a discussion.

The transpose of this matrix defines a restriction operator  $R_i$  which restricts vectors in  $\mathbb{R}^n$  to the subdomain  $\Omega_i$ . The restriction of the discretized PDE,  $A$ , to the  $i$ th subdomain is given by  $A_i = R_{i,\delta} A R_{i,\delta}^T$ .

We can now define the *additive Schwarz* preconditioner as

$$M := \sum_{i=1}^t R_{i,\delta}^T A_i^{-1} R_{i,\delta} = \sum_{i=1}^t M_i, \quad (2)$$

where  $M_i := R_{i,\delta}^T (R_{i,\delta} A R_{i,\delta}^T)^{-1} R_{i,\delta}$ . Note that, by the definition of  $R_{i,\delta}^T$ , there exists some permutation  $\Pi_i$  such that, for all  $\mathbf{x}$ ,

$$\Pi_i M_i \mathbf{x} = (\times \cdots \times 0 \cdots \cdots 0)^T,$$

i.e., the vector resulting from multiplication by the  $M_i$  (regardless of the permutation) will be sparse.

We can also define a *restricted additive Schwarz* (RAS) preconditioner [5] by considering the prolongation  $R_{i,0}^T$  instead of  $R_{i,\delta}^T$  in (2).

### 3 The MPGMRES Algorithm for Domain Decomposition Problems

MPGMRES [5] is a minimal residual algorithm for solving a linear system of equations which allows the user to apply more than one preconditioner simultaneously (see also [2] for a multipreconditioned version of the conjugate gradient method). At each step, new search directions are added to the search space, corresponding to  $A M_i \mathbf{v}$  for each  $i = 1, \dots, t$ , and for each basis vector  $\mathbf{v}$  of the current search space. The multipreconditioned search directions are all combined into a generalized Krylov subspace, and the minimization procedure requires solving a linear least-squares problem. As opposed to standard GMRES, here the subspace grows quickly due to the presence of multiple search directions and the projection can be expressed in terms of a block upper Hessenberg matrix; see Figure 1. It has been shown in [5] that a so-called *selective MPGMRES* (sMPGMRES) algorithm – which chooses a subset of  $t$  search directions and hence keeps the size of the search space growing only linearly – can be an effective method. MPGMRES (in both complete and selective forms) is given as Algorithm 1.

#### 3.1 Computational Work

In the selective algorithm we need  $t$  matrix-vector products and  $t$  preconditioner solves per iteration, as opposed to one for both in the standard preconditioned GMRES algorithm. The main other source for work is the inner products. Note



**Fig. 1** Schematic of Arnoldi decompositions in complete and selective MPGMRES

---

**Algorithm 1** MPGMRES

---

```

Choose  $\mathbf{u}_0, \mathbf{r}_0 = \mathbf{f} - \mathcal{A}\mathbf{u}_0$ 
 $\beta = \|\mathbf{r}_0\|, \mathbf{v}_1 = \mathbf{r}_0/\beta$ 
 $Z_1 = [\mathcal{M}_1\mathbf{v}_1 \cdots \mathcal{M}_t\mathbf{v}_1]$ 
for  $k = 1, \dots$ , until convergence do
     $W = \mathcal{A}Z_k$ 
    for  $j = 1, \dots, k$  do
         $H_{j,k} = (V_j)^T W$ 
         $W = W - V_j H_{j,k}$ 
    end for
     $W = V_{k+1} H_{k+1,k}$  (skinny QR factorization)
     $\mathbf{y}_k = \operatorname{argmin} \|\beta \mathbf{e}_1 - \tilde{H}_k \mathbf{y}\|_2$ 
     $\mathbf{u}_k = \mathbf{u}_0 + [Z_1 \cdots Z_k] \mathbf{y}_k$ 
     $Z_{k+1} = \begin{cases} [\mathcal{M}_1 \mathbf{v}_{k+1} \cdots \mathcal{M}_t \mathbf{v}_{k+1}] & \text{for complete MPGMRES} \\ [\mathcal{M}_1 \mathbf{v}_{k+1} \mathbf{1} \cdots \mathcal{M}_t \mathbf{v}_{k+1} \mathbf{1}] & \text{for selective MPGMRES} \end{cases}$ 
end for

```

---

that every entry in the Hessenberg matrix  $H_k$  is the result of an inner product, and these are the only inner products in the algorithm. MPGMRES therefore needs  $(2k-1)\frac{t^2}{2} + \frac{3}{2}t$  inner products at the  $k$ th step [5, Table 4.1].

Significantly, in the domain decomposition setting, due to the nature of the standard Additive Schwarz preconditioner, the preconditioning step is *exactly* the same cost when using both selective MPGMRES and standard preconditioned GMRES. Moreover, since the vectors we obtain by applying the preconditioners are sparse, the cost of the matrix-vector products will also be of the same order as in the standard GMRES algorithm – the only extra expense coming from the overlapping nodes. Indeed, if we use RAS, then the cost of a matrix-vector product would be identical here too. While we studied RAS in the context of MPGMRES in [5], in the rest of this paper we restrict our comments and experiments to additive Schwarz. The extra cost in the MPGMRES approach therefore lies completely with the inner products. The vectors here are, in general, dense, as we lose sparsity of  $W$  in the modified Gram-Schmidt step (in the inner loop of Algorithm 1).

### 3.2 The subspace in complete MPMGRES

Recall that (complete) MPMGRES minimizes over the multi-Krylov subspace

$$\mathcal{K}_{M_1, \dots, M_t}^k(A, \mathbf{r}_0),$$

where

$$\begin{aligned} \mathcal{K}_{M_1, \dots, M_t}^1(A, \mathbf{r}_0) &= \text{span}\{M_1 \mathbf{A} \mathbf{r}_0, \dots, M_t \mathbf{A} \mathbf{r}_0\}, \\ \mathcal{K}_{M_1, \dots, M_t}^2(A, \mathbf{r}_0) &= \text{span}\{M_1 \mathbf{A} \mathbf{r}_0, \dots, M_t \mathbf{A} \mathbf{r}_0, M_1 \mathbf{A} M_1 \mathbf{r}_0, \dots, M_1 \mathbf{A} M_t \mathbf{r}_0, \dots \\ &\quad \dots, M_t \mathbf{A} M_1 \mathbf{r}_0, \dots, M_t \mathbf{A} M_t \mathbf{r}_0\}, \end{aligned}$$

etc. Usually the size of this space grows exponentially with each iteration. However, in an additive Schwarz context the situation is not quite so dire, as we see below.

First, note that each preconditioned matrix is a projection, since

$$M_i \mathbf{A} M_i = R_{i,\delta}^T (R_{i,\delta} \mathbf{A} R_{i,\delta}^T)^{-1} R_{i,\delta} \mathbf{A} R_{i,\delta}^T (R_{i,\delta} \mathbf{A} R_{i,\delta}^T)^{-1} R_{i,\delta} = M_i.$$

Hence applying  $M_i$  to  $\mathbf{A} M_i$  does nothing to enrich the space.

Next, note that

$$M_i \mathbf{A} M_j = R_{i,\delta}^T (R_{i,\delta} \mathbf{A} R_{i,\delta}^T)^{-1} R_{i,\delta} \mathbf{A} R_{j,\delta}^T (R_{j,\delta} \mathbf{A} R_{j,\delta}^T)^{-1} R_{j,\delta}.$$

In the middle of this expression is the cross-term  $R_{i,\delta} \mathbf{A} R_{j,\delta}^T$ . Now note that  $R_{i,\delta} \mathbf{A} R_{j,\delta}^T = 0$  whenever  $I_i \cap I_j = \emptyset$ . Provided the overlap  $\delta$  is not large enough to touch two subdomains, this implies that only the contributions from sub-domains that touch each other add anything to the multi-Krylov subspace. This is the number of edges + corners in 2D (a maximum of 8 for a tensor product-based grid), and these plus the number of faces in 3D (a max of 26 for a tensor product-based grid). Altogether, this means that

$$\dim(\mathcal{K}_{M_1, \dots, M_t}^k(A, \mathbf{r}_0)) = (kc + 1)t,$$

where  $c$  is a constant independent of  $k, t$ . Therefore, even in the complete MPMGRES case, we only have *linear* growth in the search space.

## 4 Numerical Experiments

If we split the domain into a small number of subdomains, i.e., we have a high proportion of subdomains lying on an edge, then there may not be much difference between the spaces minimized over by the selective algorithm and the complete algorithm.

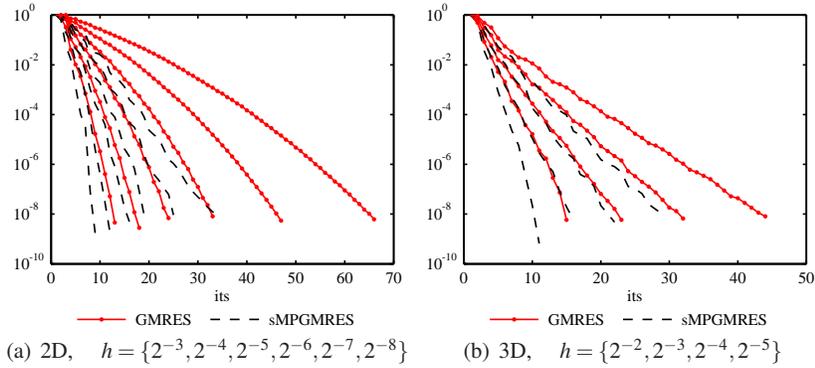
For example, consider the special case where we split the domain  $\Omega$  into two subdomains,  $\Omega_1$  and  $\Omega_2$  such that  $\Omega_1 \cup \Omega_2 = \Omega$ . Then it can be shown [5, Section 5.2.1] that, provided the subdomain solves are exact, the space over which we minimize in both selective and complete MPMGRES are identical.

Figure 2 shows the convergence curves for solving the advection-diffusion equation

$$-\nabla^2 u + \omega \cdot \nabla u = f \quad \text{in } \Omega \quad (3)$$

$$u = 0 \quad \text{on } \partial\Omega, \quad (4)$$

where  $\Omega$  denotes the unit square and  $\omega = 10 \left( \cos(\frac{\pi}{3}), \sin(\frac{\pi}{3}) \right)^T$ . This is discretized using finite differences with a uniform mesh size  $h$ , and the right hand side is taken to be the vector of ones. Thus, in 2D,  $n = 1/h^2$  and in 3D,  $n = 1/h^3$ .



**Fig. 2** Convergence curves for solving the advection-diffusion equation (3-4) with two subdomains in 2D and 3D. The iteration number is plotted along the x-axis, and  $\|r_k\|_2$  is plotted along the y-axis.

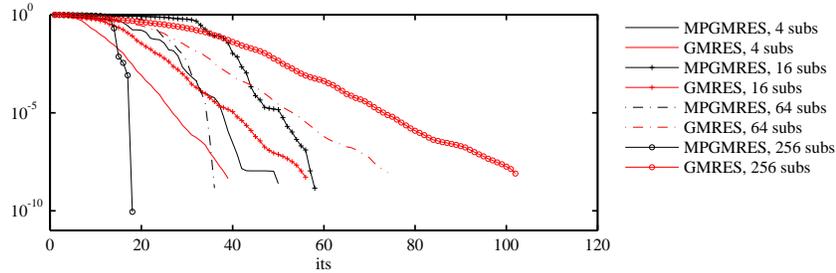
As we see in Figure 2, the iteration counts are significantly better using a multipreconditioned approach. Despite only having a serial MATLAB code, this also corresponds to significantly better timings, as is seen in Table 1: it is anticipated that the difference between the two approaches would be even more striking in a parallel implementation.

For a large numbers of subdomains, the work involved in the inner products and vector updates becomes significant, even though the work in actually applying the preconditioners is essentially the same as for the usual AS method. Convergence curves for the problem (3)-(4) are given in Figure 3.

Although the iteration counts are impressive for a large number of subdomains (with, e.g., 101 iterations for GMRES with an additive Schwarz preconditioner being reduced to 17 iterations with selective MPMGRES for 256 subdomains), the timings in this case are not yet competitive – e.g., for the case with 256 subdomains GMRES converges in 2.5s whereas sMPGMRES takes 9s. This is due to the fact

**Table 1** Timings for sMPGMRES and GMRES with two subdomains in 2D (left) and 3D (right)

$h$	sMPGMRES	GMRES	$h$	sMPGMRES	GMRES
$2^{-3}$	0.008	0.007	$2^{-2}$	0.010	0.011
$2^{-4}$	0.015	0.023	$2^{-3}$	0.059	0.058
$2^{-5}$	0.13	0.087	$2^{-4}$	1.03	1.49
$2^{-6}$	0.32	0.55	$2^{-5}$	25.6	39.7
$2^{-7}$	2.1	3.7			
$2^{-8}$	15.3	28.6			

**Fig. 3** Convergence curves for multiple subdomains in 2D ( $h = 2^{-6}$ ). The iteration number is plotted along the x-axis, and  $\|\mathbf{r}_k\|_2$  is plotted along the y-axis.

that we are using a proof-of-concept (serial) MATLAB code. Recall that the only extra work between the methods is in calculating the inner products and the subsequent vector update in the Gram-Schmidt process. Due to the block nature of the proposed method much of this extra work could be distributed across any available processors. We envisage that a state-of-the-art implementation would yield great computational savings, which would be manifested in a significantly reduced running time. This would be especially true for very large scale problems, where the cost of the subdomain solves would dominate the cost of each iteration. A Fortran 95 implementation of MPGMRES – HSL\_MI29 – will be included in the 2013 release of the HSL subroutine library.

Recall from Algorithm 1 that in the implementation of sMPGMRES reported here we apply each preconditioner to the sum of the columns of  $V_{k+1}$ . This choice is by no means unique, and there are many other possible selection strategies [5, Section 2.3]. The approach employed here seems to perform well on a wide range of problems, but it is a somewhat arbitrary choice. There may be situations where another selection strategy would be superior; this is one avenue for future research.

## 5 Conclusions

We have presented an algorithm that applies Additive Schwarz with Variable Weights. The approach is incorporated as a set of multiple preconditioners into MPGMRES. Domain decomposition has a few unique features that make our approach particularly attractive. First, the preconditioning step entails the same cost when using both selective MPGMRES and standard preconditioned GMRES, and the cost of the matrix-vector products is also of the same order as in the standard GMRES algorithm. Secondly, because there is a very low degree of overlap between nodes in the different subdomains, the growth in the search space for complete MPGMRES is only linear, i.e., very modest. This is in contrast to other situations, where the search space for complete MPGMRES grows exponentially and we settle for a selective algorithm. For these reasons we believe that the combination of domain decomposition preconditioners and the MPGMRES framework is an effective method for the numerical solution of linear systems arising from PDEs.

**Acknowledgements** The work of the first author was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), and that of the third author in part by the U.S. National Science Foundation under grant DMS-1115520.

## References

1. Ayuso de Dios, B., Baker, A.T., Vassilevski, P.S.: A combined preconditioning strategy for nonsymmetric systems (2012). Arxiv:1208.4544v1
2. Bridson, R., Greif, C.: A multipreconditioned conjugate gradient algorithm. *SIAM Journal on Matrix Analysis and Applications* **27**(4), 1056–1068 (2006)
3. Cai, X.C., Sarkis, M.: A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing* **21**, 239–247 (1999)
4. Frommer, A., Schwandt, H., Szyld, D.B.: Asynchronous weighted additive Schwarz methods. *Electronic Transactions on Numerical Analysis* **5**, 48–61 (1997)
5. Greif, C., Rees, T., Szyld, D.B.: MPGMRES: a generalized minimum residual method with multiple preconditioners. Tech. Rep. 11-12-23, Department of Mathematics, Temple University (2011). Revised September 2012. Also available as Technical Report TR-2011-12, Department of Computer Science, University of British Columbia
6. Rui, P.L., Yong, H., Chen, R.S.: Multipreconditioned GMRES method for electromagnetic wave scattering problems. *Microwave and Optical Technology Letters* **50**(1), 150–152 (2008)
7. Saad, Y.: A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing* **14**, 461–469 (1993)
8. Saad, Y.: *Iterative methods for sparse linear systems*, second edn. SIAM, Philadelphia (2003)
9. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing* **7**, 856–869 (1986)
10. Simoncini, V., Szyld, D.B.: Recent computational developments in Krylov subspace methods for linear systems. *Numerical Linear Algebra with Applications* **14**, 1–59 (2007)
11. Toselli, A., Widlund, O.B.: *Domain Decomposition Methods - Algorithms and Theory*, *Springer Series in Computational Mathematics*, vol. 34. Springer, Berlin and Heidelberg (2005)