A Comparison of Additive Schwarz Preconditioners for Parallel Adaptive Finite Elements

Sébastien Loisel[1] and Hieu Nguyen[1]

1 Introduction

We consider a second order elliptic boundary value problem in the variational form: find $u^* \in H_0^1(\Omega)$, for a given polygonal (polyhedral) domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ and a source term $f \in L^2(\Omega)$, such that

$$\underbrace{\int_\Omega \nabla u^*(x) \cdot \nabla v(x)\, dx}_{\equiv a(u^*,v)} = \underbrace{\int_\Omega f(x)v(x)\, dx}_{\equiv (f,v)}, \quad \text{for all } v \in H_0^1(\Omega). \tag{1}$$

The Bank-Holst parallel adaptive meshing paradigm [2, 3, 1] is utilised to solve (1) in a combination of domain decomposition and adaptivity. It can be summarised as follows:

Step I - Mesh Partition: Starting with a coarse mesh $\mathcal{T}_H$, the domain is partitioned into non-overlapping subdomains: $\Omega = \cup_{i=1}^p \Omega_i$.

Step II - Adaptive Meshing: Each processor i is provided with $\mathcal{T}_H$ and instructed to sequentially solve the *entire* problem, with the stipulation that its adaptive enrichment should be limited largely to Ω_i. At the end of this step, the local mesh $\mathcal{T}_i$ on processor i are regularised such that the global fine mesh described in Step III is conforming.

Step III - Global Solve: A final finite element solution is computed on the mesh $\mathcal{T}_h = \cup_{i=1}^p \mathcal{T}_i|_{\Omega_i}$, which is the union of the refined submeshes.

An example of meshes in different steps of the Bank-Holst paradigm is illustrated in Figure 1.

Discretizing (1) using linear finite elements on the global mesh $\mathcal{T}_h$, we arrive at the following system of linear equations:

[1] Department of Mathematics, Heriot-Watt University, Riccarton, Edinburgh, EH14 4additive Schwarz, United Kingdom, e-mail: {S.Loisel}{H.Nguyen}@hw.ac.uk

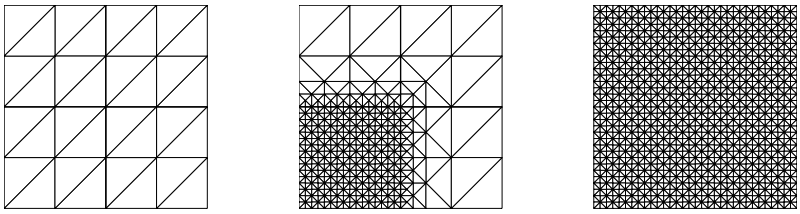

Fig. 1 A coarse mesh with partition (left), a local mesh on a processor (middle) and the global mesh (right).

$$Au = f, \quad A \in \mathbb{R}^{n\times n},\ u,\ f \in \mathbb{R}^n. \tag{2}$$

The purpose of this paper is to formulate and compare three additive Schwarz preconditioners that can be used to accelerate Krylov methods in solving (2). The improved convergence analysis will be reported somewhere else. The considered preconditioners are: the two-level additive Schwarz preconditioner with small overlap [5, 8], two-level additive Schwarz preconditioner with weakly overlapping [4] and optimal one-level additive Schwarz preconditioner based on full-domain decomposition [6].

2 Preconditioners Formulation

As all of the considered preconditioners are additive Schwarz preconditioners, they can be formulated and analyzed using the abstract theory of Schwarz methods (cf. [8]) which is summarized as follows.

Assume the global finite element space V_h associated with $\mathcal{T}_h$ admits the decomposition

$$V_h = \sum_{i=i_0}^{p} V_i, \tag{3}$$

where V_i are subspaces of V_h and $i_0 = 0$ or 1. The subspace V_0 is usually related to a coarse problem, built on a coarse mesh ($\mathcal{T}_H$ in the Bank-Holst paradigm). The subspaces V_i, on the other hand, are often related to a partition in subdomains and are associated with local submeshes. But, this is not the case for the third considered preconditioner, which is proposed in [6].

Now let $\{\psi_1^{(i)}(x), \ldots, \psi_{n_i}^{(i)}(x)\}$ be a basis of V_i and let $x_1, \ldots, x_n$ be the nodal points of the global mesh $\mathcal{T}_h$. We define

$$R_i = \begin{bmatrix} \psi_1^{(i)}(x_1) & \cdots & \psi_1^{(i)}(x_n) \\ \vdots & \ldots & \vdots \\ \psi_{n_i}^{(i)}(x_1) & \cdots & \psi_{n_i}^{(i)}(x_n) \end{bmatrix}. \tag{4}$$

It can be noted that R_i is the matrix representation of the restriction operator from V_i to V. Using this operator, the local stiffness matrix associated with subspace V_i is defined by

$$A_i = R_i A R_i^T. \tag{5}$$

Then the additive Schwarz preconditioner associated with the decomposition (3) is

$$P = \sum_{i=i_0}^{p} R_i^T A_i^{-1} R_i \tag{6}$$

The preconditioner P is said to be two-level when i_0 is 0 (coarse level: V_0, fine level: $\{V_i\}_{i=1}^p$) or one-level when i_0 is 1.

Next we will formulate three different additive Schwarz preconditioners for the Bank-Holst paradigm using different decomposition (3) with different choices of V_i. For clarity, we will use different variations of the notations V_i, R_i and A_i to denote the subspace, its corresponding restriction matrix and local stiffness matrix.

Two-level additive Schwarz preconditioner with small overlap: This is the standard and most popular version of additive Schwarz preconditioner. It is introduced in a general context without adaptivity. However, it can be used for the Bank-Holst paradigm and we present it here for comparison. For this preconditioner, each subdomain Ω_i is extended to a larger region $\hat{\Omega}_i$ by adding a small number of layers of elements **in the global (fine) mesh** $\mathcal{T}_h$ (see Fig. 2, left). The subspaces $\hat{V}_i$ are then defined as

$$\hat{V}_i = \{v(x) \in H_0^1(\hat{\Omega}_i) | \quad v(x)|_T \in \mathbb{P}_1(T),\ \forall T \in \mathcal{T}_h\}. \tag{7}$$

The two-level additive Schwarz preconditioner with small overlap is simply

$$P_{SO} = R_0^T A_0^{-1} R_0 + \sum_{i=1}^{p} \hat{R}_i^T \hat{A}_i^{-1} \hat{R}_i. \tag{8}$$

The condition number of the preconditioned system associated with P_{SO} is bounded from above by $C(1 + (H/\delta))$, where C is a constant independent of the mesh sizes, H is the coarse mesh size and δ is the width of the overlap (cf. [5, 8] and references therein). If δ is of size $\mathrm{O}(h)$, the usual case in practice, the condition number of the preconditioned system in the Bank-Holst paradigm will increase linearly as the level of refinement increases. In case the overlap is "generous", δ is of size $\mathrm{O}(H)$, the condition number is bounded by a constant, i.e. $\mathrm{O}(1)$, independent of the mesh sizes H, h and the number of subdomains p. But, there is an important practical concern that the cost of using generous overlap is too expensive as the number of vertices in the overlapping region would be $\mathrm{O}(h^{-2})$ in 2D and $\mathrm{O}(h^{-3})$ in 3D.

Weakly overlapping two-level additive Schwarz preconditioner: The formulation of this preconditioner is very much similar to that of P_{SO}. The only difference is that each subdomain Ω_i is extended to a larger region

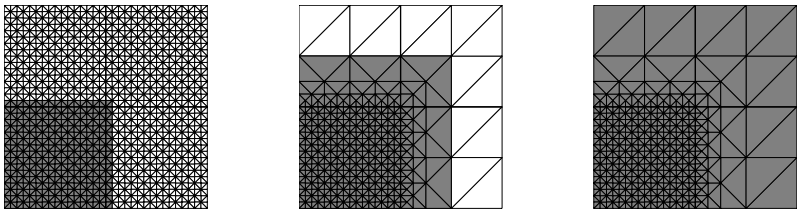

Fig. 2 Extension regions (shaded areas) and their associated meshes in cases of small overlap (left), weak overlap (center), full domain overlap (right).

$\tilde{\Omega}_i$ by adding layers of elements **in the adaptive mesh $\mathcal{T}_i$ so that the overlap is of size** $\mathrm{O}(H)$ (see Fig. 2, center). Then the subspace $\tilde{V}_i$ is defined by

$$\tilde{V}_i = \{v(x) \in H_0^1(\tilde{\Omega}_i) | \quad v(x)|_T \in \mathbb{P}_1(T),\ \forall T \in \mathcal{T}_i\}, \tag{9}$$

and the weakly overlapping two-level additive Schwarz preconditioner is defined by

$$P_{WO} = R_0^T A_0^{-1} R_0 + \sum_{i=1}^{p} \tilde{R}_i^T \tilde{A}_i^{-1} \tilde{R}_i. \tag{10}$$

By using adaptive mesh $\mathcal{T}_i$ instead of $\mathcal{T}_h$, the number of vertices in the overlapping region is reduced to $\mathrm{O}(h^{-1})$ in 2D and $\mathrm{O}(h^{-2})$ in 3D. In addition, the condition number of the preconditioned system associated with P_{WO} can be bounded independently of the mesh sizes $H,\ h$ and the number of subdomains p, i.e. is $\mathrm{O}(1)$ (see [4]).

Optimal one-level additive Schwarz preconditioner: In order to take full advantage the Bank-Holst paradigm, Loisel and Nguyen [6] formulate an additive Schwarz preconditioner that utilises the subspaces associated with the local adaptive meshes in the paradigm. These are meshes of the whole domain Ω residing locally on each processor. They form a "full domain overlap" partition of the domain (see Fig. 2, right). In this case, the local subspaces are:

$$V_i = \{v(x) \in H_0^1(\Omega) | \quad v(x)|_T \in \mathbb{P}_1(T),\ \forall T \in \mathcal{T}_i\}. \tag{11}$$

And the optimal one-level additive Schwarz preconditioner is

$$P_{O_1} = \sum_{i=1}^{p} R_i^T A_i^{-1} R_i. \tag{12}$$

Here, we should emphasize that explicit coarse component (in two-level formulation) is not needed in this case because the coarse space V_0 is contained in each and every subspace V_i.

It is shown in [6] that the condition number of the preconditioned system associated with P_{O_1} can also be bounded independently of the mesh sizes H, h and the number of subdomains p, i.e. is $\mathrm{O}(1)$.

3 Remarks on the Implementation

In order to compute the restriction matrices as defined in (4), one usually uses the nodal basis functions corresponding to the submeshes/meshes associated with V_i for $\{\psi_1^{(i)}(x), \ldots, \psi_{n_i}^{(i)}(x)\}$. In cases of P_{SO}, the nodal points in the submesh associated with $\hat{V}_i$ form a subset of the fine nodal points $\{x_1, \ldots, x_n\}$. Consequently, $\hat{R}_i$, $i > 0$, are rectangular matrices of zeros and ones, which extracts the nodal points that lie in the extension region $\hat{\Omega}_i$. In case of P_{WO} and P_{O_1}, the nodal points associated with $\tilde{V}_i$ and V_i that lie outside Ω_i does not belong to the fine mesh $\mathcal{T}_h$. Therefore, the corresponding rows of $\tilde{R}_i$ and R_i can have values in $(0, 1)$. For simplex elements, one can compute these rows using the fact that $\psi_j^{(i)}(x_k)$ equals either zero or the barycentric coordinate of x_k with respect to the coarse element containing x_k and having $x_j^{(i)}$ as one of its vertices. Here $x_j^{(i)}$ is the nodal point in $\mathcal{T}_i$ associated with $\psi_j^{(i)}$. The same technique can be used to compute the restriction matrix R_0.

For P_{O_1}, if minimal refinement is allowed outside the local subdomain in each local adaptive mesh, the rows of R_i associated with nodal points far away from Ω_i are the same with the corresponding rows of R_0. Computing these rows requires only the knowledge of the coarse mesh $\mathcal{T}_H$ and the local submesh of $\mathcal{T}_h$ which is available locally on each processor. Therefore, each processor can compute parts of R_0 locally and exchange the information with others to construct the full R_i.

In case of P_{SO} and P_{WO}, the only way of obtaining the local stiffness matrices A_i is via (5), which has the computational cost of $O(N_i^2)$. Here N_i is the number of degrees of freedom in Ω_i. If the global matrix A is assembled but distributed, there will also be communication cost that can be expensive. For P_{O1}, one is able to assemble A_i with the computational cost of roughly $O(N_i)$. The assembling requires no communication as $\mathcal{T}_i$ are available locally and are meshes of the whole domain Ω. In addition, the communication cost can be reduced further as A is not needed to be assembled.

4 Numerical Experiments

In this section, we present numerical experiments for the following problem

$$\begin{aligned} -\Delta u &= 1 \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{13}$$

where Ω is a L-shaped domain (the unit square missing the lower right quarter). The solution of this problem is shown in Figure 3 (left).

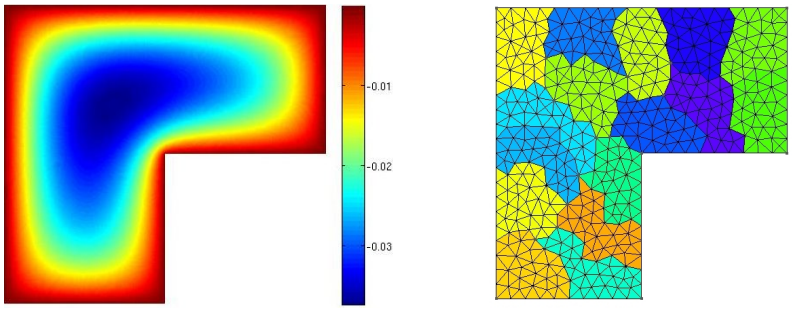

Fig. 3 Solution (left) and a coarse mesh with a partition of 16 subdomains (right).

We start with an unstructured triangular (coarse) mesh of 436 vertices and 1026 elements. Then, we partition it into p subdomains, $p = 16, 32, 64, 128$. Each processor gets exactly the same copy of this mesh. The coarse mesh with a partition of 16 subdomains are shown in Figure 3. In Step 2, local adaptive meshes are obtained by refining elements inside and surrounding local subdomains. In this experiment, we limit outside refinement by refining only ones which share at least one point with the local subdomain. Hanging nodes are allowed even though they are not considered as real nodal points. When an element is refined, it is split into four similar elements having half of its size. We use l levels of refinement for each local mesh, $l = 4, 5, 6$. The preconditioners P_{SO}, P_{WO} and P_{O_1} are implemented with the first two having the overlap of size h (one layer of fine elements) and H (equivalent to one layer of coarse elements) respectively.

Since all of the three preconditioners are symmetric positive definite, they are suitable to use with the CG method. However, it is well-known [7] that the convergence of CG in finite precision departs significantly from the theoretical convergence of CG in exact arithmetic. Therefore, we also use the GMRES method, which is slightly more numerically robust, in our experiments.

Table 1 reports the number of CG and GMRES iterations to bring the relative residual below 10^{-6}. The number of degrees of freedom and the average of elapsed time required to apply the preconditioners on a vector are also provided for comparison.

It can be seen that P_{SO} requires the most iterations for both CG and GMRES to converge. The iteration counts are clearly increasing as h becomes smaller (higher level of refinement). For GMRES, P_{O_1} is the best performer. It requires just half the number of iterations needed in case of P_{WO}. The numbers of GMRES iterations for these two preconditioners appear to be bounded by a constant, as predicted by theory. For CG, the number of iterations increases when l increases in case of P_{WO}, and when p increases in case of P_{O_1}. Between the two preconditioners, P_{O_1} has more wining cases.

Table 1 Number of CG and GMRES iterations to bring the relative residual below 10^{-6} and average of elapsed time (in seconds) to apply the preconditioners on a vector - Minimal outside refinement

	l=4, $N = 101761$				l=5, $N = 405761$				l =6, $N = 1620481$			
p=	16	32	64	128	16	32	64	128	16	32	64	128
$P = P_{SO}$												
CG no. it.	23	24	24	26	31	32	33	36	43	44	45	49
GMRES no. it.	13	14	15	15	16	17	18	18	20	20	21	22
time p. mult.	1.3	1.1	0.4	0.5	6.1	5.9	6.0	5.3	27.3	25.9	25.8	26.9
$P = P_{WO}$												
CG no. it.	18	18	18	20	19	20	21	22	25	26	26	28
GMRES no. it.	12	12	12	13	12	12	12	13	13	14	15	15
time. p. mult.	1.5	1.6	1.6	1.7	7.4	7.3	7.3	7.4	32.3	31.0	31.3	31.8
$P = P_{O_1}$												
CG no. it.	15	17	20	23	15	17	21	23	15	17	21	23
GMRES no. it.	6	7	8	8	6	7	8	8	6	7	8	8
time. p. mult.	1.7	1.9	2.3	2.7	7.2	7.5	8.3	9.4	32.7	32.3	33.2	34.4

In term of elapsed time, P_{O_1} and P_{WO} are roughly the same. Even though they are more expensive to apply, they are more efficient than P_{SO} because they require fewer number of iterations.

In the second experiment, we study whether it is beneficial to refine local meshes in the region outside local subdomains. Now instead of using minimal outside refinement, we perform at least one level of refinement for elements that do not belong to the local subdomain. It should be noted that the global mesh $\mathcal{T}_h$ and the global stiffness matrix A are the same with those in the previous experiments. The restriction matrices and local stiffness matrix, however, are changed.

We do not see any improvement in term of iterations count for P_{WO}. Perhaps, this is due to the fact that a coarse space is already incorporated in this preconditioner. We do see clear improvement for P_{O_1} with significant reduction in iteration counts and slight increase of time. However, care must be taken when using generous refinement outside subdomains as this would require more memory and time to calculate restriction matrices.

Acknowledgements This work was supported by the Numerical Algorithms and Intelligent Software Centre funded by the UK EPSRC grant EP/G036136 and the Scottish Funding Council.

References

[1] Randolph E. Bank. Some variants of the Bank-Holst parallel adaptive meshing paradigm. *Comput. Vis. Sci.*, 9(3):133–144, 2006.

Table 2 Number of CG and GMRES iterations to bring the relative residual below 10^{-6} and average of elapsed time (in seconds) to apply the preconditioners on a vector - Extra outside refinement.

	l=4, $N = 101761$				l=5, $N = 405761$				l =6, $N = 1620481$			
p=	16	32	64	128	16	32	64	128	16	32	64	128
$P = P_{WO}$												
CG no. it.	18	18	18	20	19	20	21	22	25	26	26	28
GMRES no. it.	12	12	12	13	12	12	12	13	13	14	15	15
time. p. mult.	1.7	1.7	1.8	1.8	7.7	7.6	7.6	7.4	35.6	32.6	33.0	32.5
$P = P_{O_1}$												
CG no. it.	13	14	15	18	13	14	15	18	13	15	16	19
GMRES no. it.	5	5	5	5	5	5	5	5	5	5	5	5
time. p. mult.	2.3	2.6	3.8	5.5	8.2	9.1	10.2	13.0	34.2	34.6	38.4	42.6

[2] Randolph E. Bank and Michael Holst. A new paradigm for parallel adaptive meshing algorithms. *SIAM J. Sci. Comput.*, 22(4):1411–1443 (electronic), 2000.

[3] Randolph E. Bank and Michael Holst. A new paradigm for parallel adaptive meshing algorithms. *SIAM Rev.*, 45(2):291–323 (electronic), 2003. Reprinted from SIAM J. Sci. Comput. **22** (2000), no. 4, 1411–1443 [MR1797889].

[4] Randolph E. Bank, Peter K. Jimack, Sarfraz A. Nadeem, and Sergei V. Nepomnyaschikh. A weakly overlapping domain decomposition preconditioner for the finite element solution of elliptic partial differential equations. *SIAM J. Sci. Comput.*, 23(6):1817–1841 (electronic), 2002.

[5] Maksymilian Dryja and Olof B. Widlund. Domain decomposition algorithms with small overlap. *SIAM J. Sci. Comput.*, 15(3):604–620, 1994. Iterative methods in numerical linear algebra (Copper Mountain Resort, CO, 1992).

[6] Sébastien Loisel and Hieu Nguyen. An optimal schwarz preconditioner for parallel adaptive finite elements. submitted.

[7] Gérard Meurant. *The Lanczos and conjugate gradient algorithms*, volume 19 of *Software, Environments, and Tools*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006. From theory to finite precision computations.

[8] Andrea Toselli and Olof Widlund. *Domain decomposition methods—algorithms and theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2005.