Received: 2 April 2018 | Revised: 15 February 2019 | Accepted: 15 February 2019

DOI: 10.1002/nla.2236

RESEARCH ARTICLE

WILEY

A coarse–fine-mesh stabilization for an alternating Schwarz domain decomposition method

Owe Axelsson[1] | Ivar Gustafsson[2]

[1]Institute of Geonics, Czech Academy of Sciences, Ostrava, Czech Republic

[2]Department of Mathematical Sciences, Chalmers University of Technology/University of Gothenburg, Gothenburg, Sweden

Correspondence
Owe Axelsson, Institute of Geonics, Czech Academy of Sciences, Studentská 1768, 708 00 Ostrava, Czech Republic.
Email: Owe.Axelsson@it.uu.se

Summary

Domain decomposition methods can be solved in various ways. In this paper, domain decomposition in strips is used. It is demonstrated that a special version of the Schwarz alternating iteration method coupled with coarse–fine-mesh stabilization leads to a very efficient solver, which is easy to implement and has a behavior nearly independent of mesh and problem parameters. The novelty of the method is the use of alternating iterations between odd- and even-numbered subdomains and the replacement of the commonly used coarse-mesh stabilization method with coarse–fine-mesh stabilization.

KEYWORDS

alternating Schwarz, coarse–fine-mesh stabilization, domain decomposition, preconditioning

1 | INTRODUCTION

To achieve an acceptable computing time when solving large-scale partial differential equation (PDE) problems, that is, to enable readily parallelizable methods, one can use domain decomposition methods, that is, split the given domain (Ω) of definition in some subdomains ($\cup_i \Omega_i$). In practice, one must solve the global problem by some iterative method. The classical Schwarz method[1] was used to prove the existence of harmonic functions with prescribed Dirichlet data on the boundary of more complicated domains using the known existence on simpler domains. It is an alternating iteration method between two subdomains but can be readily extended in various ways to a set of several subdomains. Domain decomposition methods have been an important research topic since long. For early references, see the works of Chan and Mathew,[2] Smith et al.,[3] Quarteroni and Valli,[4] and Xu,[5] and for a more recent one, see the work of Toselli and Widlund[6] as well as the references therein.

In practical problems, often, subdomains with different materials in long layers arise, leading to PDEs with discontinuous coefficients. It is then natural to make use of a subdivision of the domain in strips corresponding to the different layers. When solving the arising system, it can then be efficient to combine pairs of neighboring subdomains into a larger subdomain and alternate the Schwarz iterations between an odd-numbered sequence and an even-numbered sequence of such extended subdomains. This approach is used in this paper. It increases the potential for parallel computations and avoids special treatment of interface boundary conditions.

However, clearly, such iteration methods can converge slowly. They mainly damp out the more local error components but not the globally oriented, that is, the slower-varying components. For instance, if there are $2m$ subdomains, it takes m iterations before information from an outer edge of the domain has reached the opposite edge. However, since we aim at solving elliptic problems, where Green's function is spread out globally, all information must be spread globally also

Dedication: This paper is dedicated to the memory of Gunhild Lindskog, who recently and unexpectedly passed away and who had some highly cited joint publications with us.

for the iterative solution method to make it efficient. To cope with that, one must couple the iteration method with a preconditioner that includes a global spread of information. A simple and commonly used way to do this is by the use of a coarse-mesh matrix, for instance, based only on coarse node points on the intersecting lines between the subdomains. This damps out the slower iteration error modes in the alternating iteration method and can, hence, increase its rate of convergence.

We show, however, that a substantially better improvement is achieved with the use of a combined coarse–fine-mesh matrix preconditioner, which can be based on long thin triangular elements, with edges H and h, where H, h are the coarse-mesh and fine-mesh sizes, respectively. Here, the long edges are orthogonal to the intersecting lines. This method needs few and a nearly fixed number of iterations, which holds uniformly with respect to mesh discretization and problem parameters, so that it can be considered as a very cheap solution method. The arising inner subdomain and coarse–fine matrix preconditioners can also be solved cheaply by iteration.

In the actual implementation of the methods used, several important issues arise, such as how accurate the inner systems, that is, for the subdomains and preconditioner, should be solved at each outer iteration to get the shortest solution time. Since the bandwidth of the local matrices is small for thin layers, the subdomain systems can often best be solved by a direct solution method. However, we aim here at having a more moderate number of subdomains, that is, not very thin subdomains, in which case, it can be more efficient to use an iterative solution method also for the subdomain systems. This will be a preconditioned conjugate gradient (PCG) method.

A conjugate gradient method involves the computation of several inner products, which need global spread of information of data, which can substantially slow down the execution time for parallel computations between many computer processors. In the method used, global communication is limited to communications within the subdomains only, that is, normally within each computer processor.

The coarse-mesh system is often best solved by a direct solution method. However, the coarse–fine stabilization matrix system is of larger size and should be solved by iteration. The choice of a preconditioner, either based on Chebyshev or conjugate gradient iterations, plays then an important role. This choice depends on the ratio between parallel speedup, communication time, and single-processor time.

In the paper, we present simple proofs of the properties of the methods that are based on methods related to two-level hierarchical basis function methods.[7–12] Furthermore, the theoretical properties are illustrated by extensive numerical tests. It is found that substantial reduction in the number of subdomain iterations can be done by the use of varying stopping criteria during the Schwarz iterations.

This paper is composed as follows. In Section 2, we describe the alternating Schwarz iteration method and analyze its rate of convergence but in a more introductory way. We also describe the global stabilization methods by the use of a coarse-mesh or a coarse–fine-mesh matrix preconditioner combined with the alternating Schwarz method. In Section 3, extensive numerical tests are presented. In Section 4, some implementational issues, such as the choice of stopping criteria related to the use of inner iterations, are discussed, both for continuous and discontinuous coefficient problems. This paper ends with some concluding remarks and an Appendix, devoted to the basic properties of hierarchical basis function methods used to construct efficient preconditioners and are presented to prove an upper bound on the convergence factor of the iteration method.

The following notations are used. $2m$ denotes the number of strips, $H = 1/m$, and h is the fine-mesh size parameter.

2 | A SCHWARZ ALTERNATING ITERATION METHOD AND ITS STABILIZATION

For presentations of Schwarz-type domain decomposition methods, see the works of Chan and Mathew,[2] Smith et al.,[3] and Toselli and Widlund.[6] For a nice shorter overview, see the work of Blaheta[13]; for a more recent presentation, see the work of Axelsson and Blaheta[14]; and for a particular application, see the work of Blaheta et al.[15] In this section, the presentation is devoted to the use of striped domain decomposition methods and odd–even-numbered sets of subdomains.

2.1 | A Schwarz alternating iteration method for a striped domain decomposition method with maximal overlap

To solve an elliptic-PDE problem $\mathcal{L}u = f$, where $\mathcal{L}u = \sum_{i,j=1}^{2} \frac{\partial}{\partial x_i}(a_{ij}\frac{\partial u}{\partial x_j})$ and a_{ij}, $i,j = 1,2$, forms a symmetric and positive-definite matrix, or a corresponding linear elasticity problem, on a domain Ω, one can use domain decomposition

methods. The domains are then split into subdomains of approximately equal size. Domain decomposition methods are particularly important for the implementation of algorithms to be used on a parallel computer platform. Here, one can balance the size and number of subdomains to cope with the number of processors and available memory storage in each processor. There are various forms of such methods.

The two major classes of algorithms are nonoverlapping and overlapping domain decomposition methods. The subdomains are then disjoint, or rather, their closure has just a common intersection, or they may be overlapping. In connection with the use of finite element discretization, in nonoverlapping methods, one splits the domain via element splittings, which means that node points on the interfaces are shared between two subdomains when a partitioning in strips has been done.

The normally imposed interface condition is the Dirichlet condition $u_i^- = u_i^+$ or the flux balance condition

$$\left.\frac{\partial u}{\partial n}\right|_\Gamma^- + \left.\frac{\partial u}{\partial n}\right|_\Gamma^+ = 0,$$

where n denotes the outward-pointing normal vector and where the superscript $\pm$ stands for values taken on the left and right sides of the interface.

By partitioning the node points in interior and interface nodes, the corresponding matrix system takes the form

$$A\begin{bmatrix} x_1 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} A_{11} & A_{1\Gamma} \\ A_{\Gamma 1} & A_{\Gamma\Gamma} \end{bmatrix}\begin{bmatrix} x_1 \\ x_\Gamma \end{bmatrix} = \begin{bmatrix} f_1 \\ f_\Gamma \end{bmatrix}, \tag{1}$$

where A_{11}, x_1 corresponds to interior nodes and $A_{\Gamma\Gamma}$, x_Γ corresponds to interface nodes. This corresponds to the use of Neumann boundary conditions, which means that the matrix $A_{\Gamma\Gamma}$ is singular, and one must split off the singular part. The matrix A_{11} corresponds to homogeneous Dirichlet boundary conditions and is hence nonsingular. With a proper node ordering, it is block diagonal, with each block corresponding to the interior node points in one of the subdomains. System (1) can be solved via the reduced, Schur complement system

$$Sx_\Gamma = \left(A_{\Gamma\Gamma} - A_{\Gamma 1}A_{11}^{-1}A_{1\Gamma}\right)x_\Gamma = f_\Gamma - A_{\Gamma 1}A_{11}^{-1}f_1.$$

Here, S, which is a full matrix, is not formed explicitly, but the equation is solved by an iterative method, whereby each iteration requires only the action of S on vectors, that is, it requires some matrix vector multiplication and a solution of a system with the interior-point matrix. Hence, it can be performed by a parallel solution of all subdomain problems.

As is well known (see, e.g., the works of Axelsson and Barker,[8] Axelsson,[9] and Axelsson and Polman[16]), for second-order elliptic problems, the global matrix has a large spectral condition number, $O(h^{-2})$, whereas S has a reduced condition number, $O(h^{-1})$, $h \to 0$. This is still large. Therefore, although each action of S can be performed in parallel, the solution of (1) can be expensive, and one must precondition S. This can be done in various ways. For instance, the coarse–fine matrix to be used in this paper is applicable. There are also other methods available using different combinations of interface boundary conditions.[2–6] In general, these methods are quite complicated. Another problem with the nonoverlapping domain decomposition method (1) is that if only continuity of the solution but not continuity of the solution fluxes is imposed, then, particularly for problems with discontinuous coefficients, it can lead to much less accurate solutions. Alternatively, it can be imposed as a constraint condition and use of dual variables, but this leads to a more involved, saddle-point structure of the global matrix problem to be solved. This is one reason why we here instead choose a form of the overlapping domain decomposition method. Here, the matching of the solution and that of its normal derivative along the interfaces are imposed explicitly.

One normally uses some form of an alternating iteration method between subdomains. The classical Schwarz method was used for two overlapping subdomains. Here, we shall use an arbitrary number of subdomains that do not overlap in the common way but that, by the use of the alternating iteration method, can be considered as a two-domain method with strong, maximal overlap. Due to the recursive use of sets of even- and odd-numbered subdomains, the resulting overlap avoids any form of complications of the abovementioned matching of the solution and of its normal derivative along the interfaces.

For simplicity, we use only domain decomposition in strips. This has the advantage that, for a proper ordering of node points, the local subdomain problems can be solved readily using a direct solution method for a matrix with a fairly small bandwidth, $O(\frac{1}{mh})$, where $2m$ is the number of subdomains and h is a mesh size parameter. Alternatively, when the bandwidth is not small, one can use an iterative solution method. This can be a Cholesky incomplete factorization method, a modified incomplete factorization method such as MIC(2) (see the works of Axelsson[9] and Gustafsson[17]), or a multilevel iteration method, where we use a combination of a coarse-mesh basis function matrix in such subdomains and

the inner-node-point matrix (see, e.g., the work of Axelsson and Gustafsson[7] for a two-level version and those of Axelsson and Vassilevski[10,11] for a multilevel version; see also the work of Vassilevski[12]). The latter method has an optimal order of rate of convergence, that is, the number of iterations is bounded with respect to mesh size h. Although the MIC(2) method does not have an optimal order of convergence, from the following analysis and numerical tests, we can see that it performs very well. As described in the works of Axelsson[9] and Gustafsson,[17] in the MIC(2) method, one uses an incomplete factorization, allowing a double amount of sparsity pattern in the factorization as compared to that in the given matrix.

For simplicity, we assume a rectangular domain of unit width. We order the interface edges consecutively. Furthermore, we assume Dirichlet boundary conditions on the outer vertical boundaries, as indicated in Figure 1. The boundary condition on the horizontal boundaries can be Dirichlet, Neumann, or mixed.

The alternating iteration method takes the following form: Given initial values, for instance, homogeneous Dirichlet boundary conditions, on the even-numbered edges ($\Gamma_i, i = 2, 4, \ldots, 2m-2$), then

(i) compute the solution on the odd-numbered edges by solving the separate subdomain problems for the overlapping domains $\Omega_i, i = 1, 3, 5, \ldots$ (see Figure 1);
(ii) use the so-computed values at edges Γ_i, $i \in 1, 3, \ldots, 2m-1$ as Dirichlet boundary conditions, to compute the solution at the edges $(2, 4, \ldots, 2m-2)$, now by solving the local subdomain problems in the overlapping subdomains $\Omega_i, i = 2, 4, \ldots, 2m-2$ (see Figure 2). (Here, we do not have to solve the subdomain problems at the outer boundaries to the left and right.)

This gives updated values at the even-numbered edges $2, 4, \ldots, 2m-2$. Then, restart the alternating iteration method and continue until convergence, $\|\mathcal{L}u - f\| \leq \varepsilon \|f\|$, for some chosen relative accuracy ε, $0 < \varepsilon \ll 1$.

The convergence of the method can be studied on the continuous operator level. The only mechanism for sharing information is local, either through the interface or through the overlapping regions. However, since we are solving elliptic-PDE problems, the corresponding Green's function has global support. To enable information to travel from one end of the domain to the opposite part of the domain, one needs m steps of the method. This indicates that at least $O(m)$ iteration steps are needed. The convergence can also be studied algebraically. The global matrix restricted to a subdomain corresponds to a block-diagonal matrix block. Hence, one step of the method corresponds to a block-diagonal preconditioner. As is well known (see, e.g., the works of Axelsson and Gustafsson,[7] Axelsson and Barker,[8] and Axelsson and Vassilevski[10]), the condition number of such a block-diagonal preconditioned system is $O(H^{-2})$. Hence, the number of iterations of the method grows rather as $O(m^2)$ than as the lower bound, $O(m)$ indicates. This will be stated more clearly in Proposition 1 to follow.

The alternating iteration method can be written as follows. Let R_i, $i = 1, 2$, be the Boolean restriction matrices that restrict the mesh components by deleting even- and odd-numbered interfaces, respectively. Correspondingly, R_i^T

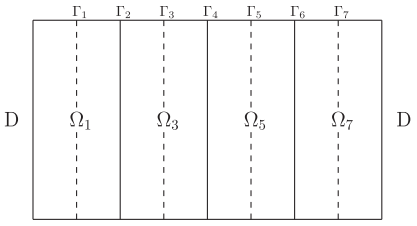

FIGURE 1 Odd-numbered overlapping subdomains Ω_i for the first step (eight strips)

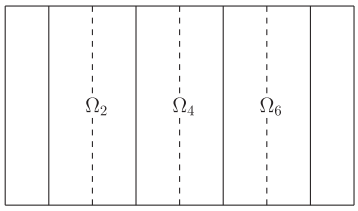

FIGURE 2 Even-numbered overlapping subdomains Ω_i used for the second step (eight strips)

interpolates back to the whole set. The entries of R_i are just ones and zeros; hence, the matrices $A_i = R_i A R_i^T$ we work with are principle submatrices of A.

The iteration method can be written as follows: Given u^0, for $k = 0, 1, \ldots$ until convergence, compute

$$u^{(k+1/2)} = u^{(k)} + R_1^T A_1^{-1} R_1 (f - A u^{(k)})$$
$$u^{(k+1)} = u^{(k+1/2)} + R_2^T A_2^{-1} R_2 (f - A u^{(k+1/2)}),$$

which corresponds to the iteration matrix

$$\left(I - R_2^{-T} A_2^{-1} R_2 A\right)\left(I - R_1^T A_1^{-1} R_1 A\right),$$

that is, a multiplicative Schwarz method. Let V contain all finite element basis functions and V_i the basis functions for the interior node points in a subdomain Ω_i, $i = 1, \ldots, 2m$, and let $P_i = R_i^T A_i^{-1} R_i A$. Then, $P_i^2 = P_i$, $P_i V = V_i$, that is, P_i is a Ritz projection matrix onto V_i.

Given the variational formulation of $\mathcal{L}u = f$, it holds that

$$a(P_i u, v) = a(u, v) \qquad \forall u \in V, \quad \forall v \in V_i.$$

For the error $z^{(k)} = u - u^{(k)}$, it holds that

$$z^{(k+1)} = (I - P_2)(I - P_1) z^{(k)}, \qquad k = 0, 1, \ldots .$$

Such a multiplicative Schwarz iteration method is a form of a block Gauss–Seidel iteration method. Alternatively, one can consider an additive form of the Schwarz iteration method, where

$$z^{(k+1)} = (I - P_2 - P_1) z^{(k)}.$$

This corresponds to a block Jacobi iteration method. However, in this method, the derivative interface conditions are not as accurately approximated as in the multiplicative method, and the rate of convergence is slower. The convergence factor is the product of the two individual factors, that is, still $O(H^{-2})$.

In the next proposition, we show the rate of convergence for the unstabilized Schwarz method and for solving the arising inner subdomain systems by iteration. For a thorough presentation of the Schwarz alternating iteration method, see the earlier given references, particularly that of Quarteroni and Valli.[4] For readers less familiar with domain decomposition methods, we do this here in a more elementary way.

Proposition 1. *Consider the discretized elliptic problem $\mathcal{L}u = f$ on a unit square domain, regularly partitioned in strips with width $H = 1/m$. Assume that the diffusion coefficients are smooth within each subdomain and that H/h is constant. Then,*

(i) the number of unstabilized alternating Schwarz iterations increases as $O(m^2)$, $m \to \infty$, independently on h;
(ii) the number of iterations for solving the subdomain equations using an MIC PCG method increases as $O((H/h)^{1/2})$.

Proof.

(i) For regularly partitioned domains with long, thin subdomains, it is readily seen that, in an approximate sense, it suffices for the analysis to reduce the analyses of the convergence rate of the Schwarz alternating iteration method for the layered domains to a one-dimensional problem, $-\frac{d}{dx}\left(a \frac{dy}{dx}\right) = f$. In our case, the method alternates between the intervals for the odd-numbered points followed by the intervals bounded by the even-numbered points for which the solutions at the odd-numbered points are used to repeat the process until convergence. We assume that the coefficients a and the right-hand side function f are constant on each interval. The ordering used follows that in Figures 1 and 2.

On the iteration level $l + 1$, we solve then the subproblems

$$-\frac{d}{dx}\left(a \frac{dy}{dx}\right) = f, \qquad x_{2i-2} < x < x_{2i}, \qquad i = 1, 2, \ldots, m,$$

that is, in particular, we compute

$$y^{(l+1/2)}(x_{2i-1}) = \frac{1}{2}\left(y_{2i-2}^{(l)} + y_{2i}^{(l)}\right) + \frac{1}{2}H^2\left(\frac{1}{a_{2i-2}} f_{2i-2} + \frac{1}{a_{2i}} f_{2i}\right).$$

What then follows is the solution of

$$-\frac{d}{dx}\left(a\frac{dy}{dx}\right) = f, \qquad x_{2i-1} < x < x_{2i+1}, \qquad i = 1, 2, \ldots, m-1,$$

that is, in particular, it holds that

$$y^{(l+1)}(x_{2i}) = \frac{1}{2}\left(y_{2i-1}^{(l+1/2)} + y_{2i+1}^{(l+1/2)}\right) + \frac{1}{2}H^2\left(\frac{1}{a_{2i-1}}f_{2i-1} + \frac{1}{a_{2i+1}}f_{2i+1}\right) =$$
$$= \frac{1}{4}\left(y_{2i-2}^{(l)} + 2y_{2i}^{(l)} + y_{2i+2}^{(l)}\right) + \cdots.$$

(The not given part of the expression for the right-hand side function is not important for the analyses of the rate of convergence.)

Hence, we have

$$\mathbf{y}^{(l+1)} = Q\,\mathbf{y}^{(l)}, \qquad l = 0, 1, \ldots,$$

where

$$Q = \frac{1}{4}\begin{bmatrix} 2 & 1 & & & 0 \\ 1 & 2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ 0 & & & 1 & 2 \end{bmatrix} = I - \frac{1}{4}\begin{bmatrix} 2 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ 0 & & & -1 & 2 \end{bmatrix}.$$

It is readily seen that the same recursion holds for the iteration errors $[e_i^{(l)}]$, where $e_0 = e_{2m} = 0$ (see, e.g., the work of Axelsson and Barker[8]).

It follows from the well-known trigonometric expression for the eigenvalues of the elementary difference matrix Q that the spatial radius $\varrho(Q) = 1 - \frac{1}{4}\lambda_{\min}(Q_0) = 1 - O(H^2)$, from which follows that the number of iterations increases as $O(H^{-2}) = O(m^2)$.

(ii) Let $A^{(i)}$ be the restriction of A to subdomain Ω_i, and let $C^{(i)}$ be the corresponding MIC preconditioner. Since the bandwidth of the block triangular matrix $A^{(i)}$ is $O(H/h)$, it follows from, for instance, the works of Axelsson[9] and Gustafsson[17] that $\kappa({C^{(i)}}^{-1}A^{(i)}) \leq O(H/h)$. A conjugate gradient method requires then $O(H/h)^{1/2}$ iterations for each subdomain.

□

That the number of iterations in the alternating Schwarz method grows as $O(H^{-2})$ is also clearly seen in the numerical tests in Section 3. This holds both for continuous and discontinuous coefficient problems, at least if the coefficients are smooth within each subdomain.

For the practical implementations of the method, it is important to mention that we do not need to solve the subdomain systems very accurately. The numerical tests indicate that it suffices with relative accuracy of 10^{-4} for each subdomain system to get nearly the shortest solution time. For further discussion on this, see Section 4.

Clearly, during each step of the alternating Schwarz method, we can solve all subdomain systems in parallel. Furthermore, it does not involve computations of inner products as is required in Krylov subspace methods, which saves much global communication times between the parallel processors. If we order the node points within each subdomain along lines parallel to its edges (i.e., vertical in Figures 1 and 2), then the bandwidth of the local matrix becomes $O(H/h)$ and the computational cost of a direct solution method becomes $O(h^{-1}(H/h)^3)$. To get a lower cost, we must use an iterative solution method, such as MIC(2), or even a multilevel iteration method to get a nearly scalable method, that is, with a cost nearly proportional to the degrees of freedom, $O((hH)^{-1})$, in each subdomain. It follows from Proposition 1 that, assuming sufficiently smooth coefficients in each subdomain, the number of iterations for the MIC method grows as $O((H/h)^{1/2})$. Depending on how many coarse-mesh points are used in a two-level or a multilevel method, the solution cost can be lower. If we use an MIC method, then the solution cost for each subdomain is $O(h^{-1}H^{-1}(H/h)^{1/2})$. Compared to a direct solver, we can then gain a factor $O(H^{-1}(H/h)^{2.5})$ less computer time, normally a substantial factor. As we have already remarked, there is no need to solve the subdomain iteration problems to very high accuracy, and this shows further that iterative solution methods are most efficient.

However, we have seen that the Schwarz alternating iteration method needs $O(H^{-2})$ iterations. This means that even if we use parallel computation, the computer time will grow as $O(H^{-1})$, for H^{-1} processors. Hence, the pure alternating Schwarz method is less suitable for fast parallel computations. Therefore, it must be stabilized.

Remark 1. If we use a three-step version of the alternating Schwarz method, it corresponds to a multiplicative method with the iteration matrix

$$\left(I - AR_1^T A_1^{-1} R_1\right)\left(I - R_2^{-T} A_2^{-1} R_2 A\right)\left(I - R_1^T A_1^{-1} R_1 A\right).$$

Then, the matrix is symmetric with respect to the A inner product, and therefore, we can use a standard conjugate gradient acceleration method to reduce the number of iterations to $O(H^{-1})$. More generally, one can use a multilevel, multiplicative method with q projection matrices taken in reverse order, resulting in a symmetrized error projection operator, that is,

$$(I - \mathcal{P}_1)(I - \mathcal{P}_2)\cdots(I - \mathcal{P}_q)(I - \mathcal{P}_q)\cdots(I - \mathcal{P}_1).$$

Here, however, we must use globally computed inner products that, as has already been commented on, due to overhead for very large-scale problems, can increase the elapsed computer time much.

2.2 | Stabilized alternating Schwarz methods

For earlier presentations of coarse-mesh stabilization, see the works of Chan and Mathew,[2] Smith et al.,[3] Axelsson and Gustafsson,[7] and Blaheta.[13] In the following section, we give a more special but simplified presentation.

2.2.1 | Coarse-mesh stabilization

As an introduction to the stabilized versions of the alternating Schwarz method, consider first a global matrix in the form $\mathcal{A} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_c \end{bmatrix}$. Here, A_{11} corresponds to the fine-mesh finite element basis functions, A_c corresponds to the coarse-mesh basis functions, and A_{12} and A_{21} correspond to the coupling between coarse- and fine-mesh nodes. For a general presentation of the use of coarse-mesh spaces in domain decomposition methods, see, for example, the work of Widlund.[18] In our application, all coarse-mesh nodes are chosen to be located on the interfaces between the subdomains. If we use the block-diagonal part of $\mathcal{A}$ as a preconditioner, that is, let $\mathcal{B} = \begin{bmatrix} A_{11} & 0 \\ 0 & A_c \end{bmatrix}$, then it has been shown, for example, in the works of Axelsson and Gustafsson,[7] Axelsson,[9] and Axelsson and Blaheta[14,19] that the condition number is equal to $\kappa(\mathcal{B}^{-1}\mathcal{A}) = \frac{1+\gamma}{1-\gamma}$, where $\gamma \le \sqrt{\frac{m_0^2-1}{m_0^2}} = \sqrt{1-(\frac{h}{H})^2}$, $m_0 = H/h$, and γ is the constant in the strengthened Cauchy–Bunyakowski–Schwarz (CBS) inequality corresponding to the bilinear product

$$a(u,v) = \int_\Omega \mathcal{L}uv\, dx = \int \rho \nabla u \cdot \nabla v\, dx,$$

where

$$\mathcal{L}u = -\nabla \cdot (\rho \nabla u),$$

and the solution to the differential equation satisfies homogeneous boundary conditions. Here, $u \in \{\varphi_i\}$, the set of interior-node basis functions, and $v \in \{\Psi_i\}$, the set of boundary-node basis functions. For an improvement of this bound, see the Appendix. Since the number of iterations in a PCG method grows as $O(\kappa(\mathcal{B}^{-1}\mathcal{A})^{1/2})$ (see, e.g., the work of Axelsson and Barker[8]), it follows that the number of iterations grows as $O(m_0) = O(H/h)$. Hence, the method is not of optimal order.

However, the method can be modified to make fuller use of the coarse-mesh solution that must be computed. Assuming sufficient regularity of the solution to $\mathcal{L}u = f$, it holds that the discretization error of the coarse-mesh solution is $O(H^{1+\alpha})$, for some α, $0 < \alpha \le 1$, where $\alpha = 1$ if the solution u belongs to $H^2(\Omega)$, the second-order Sobolev function space. Hence, if we use quadratic interpolation, the interpolated solution on the interfaces has also such a discretization error, $O(H^{1+\alpha})$. This can be utilized if we first solve the coarse-mesh problem, interpolate the solution, and use these more accurate values of the approximate solution as boundary values to start the alternating Schwarz iteration method with. After one such iteration with solution u^1, we can compute the corresponding residual

$$\delta f^1 = \mathcal{L}u^1 - f$$

and then repeat the method, that is, solve the coarse-mesh problem for the defect-correction equation

$$\mathcal{L}(\delta u) = -\delta f^1,$$

interpolate, use an alternating Schwarz iteration, and repeat until full convergence. Here, the errors from the coarse-mesh solution dominate the inner solution errors $O(h^{1+\alpha})$ until a sufficient number of iterations have been done.

This is only an indication of the effect of the procedure. In practice, we steadily approximate lower-order error terms; hence, the residuals will contain an increasing number of higher-order oscillating error components that are more difficult to approximate on the coarse mesh. Therefore, the process slows down. However, the numerical tests show that the number of required iterations decreases for smaller values of the ratio H/h, which indicates that we can anyhow achieve a good parallelization if we use at least H^{-1} computer processors. In the alternating iteration method, we alternate in this way between even- and odd-numbered subdomains.

The method, where no coarse mesh is used, can be described as a *one-level* method and, with the use of a coarse mesh, as a *two-level* method. The two-level method takes the following simple form.

Coarse-mesh stabilization

Construct a coarse mesh with only vertex nodes on the even-numbered interfaces. Given an initial approximation $u^{(0)}$, for $k = 0, 1, \ldots$ until convergence.

1. **Coarse-mesh solution**
   Compute the residual $\delta f^{(k)} = f - Au^{(k)}$ and solve the coarse-mesh problem $A_c \delta u_c^{(k)} = \delta f^{(k)}$. Update the solution, $u_c^{(k+1/2)} = u_c^{(k)} + \delta u_c^{(k)}$. Interpolate the solution to the fine-mesh node points on the even-numbered interface edges (preferably using a higher-order interpolation method).
2. **Fine-mesh alternating Schwarz**
   Apply once, or more times, the two steps with the alternating Schwarz method to find the solution $u^{(k+1)}$.

Repeat.

The method can be seen as a variant of a two-level multigrid method. It can be combined with a conjugate gradient method used on the coarse-mesh level. Thereby, the fine-mesh solution method can be seen as an inner solution method to compute the action of the fine-to-coarse-mesh Schur complement.

Note that the coarse-mesh solution is particularly cheap since it is based on node points only on even-numbered edges.

This coarse-mesh-stabilized Schwarz method can be seen as a multiplicative Schwarz method with the iteration matrix

$$\prod_{i=0}^{2} \left(I - R_i^T A_i^{-1} R_i A \right),$$

where $A_0 = R_0 A R_0^T$ and R_0 is the restriction matrix corresponding to the set of coarse-mesh points. The iteration matrix for the corresponding additive version is

$$I - \sum_{i=0}^{2} P_i,$$

where $P_i = R_i^T A_i^{-1} R_i A$ is the Ritz projection. It has been shown, for example, in the works of Chan and Mathew[2] and Toselli and Widlund[6] that these methods lead to condition number bounds that hold independently of h and H. However, the bounds may depend on the variation of coefficients in the elliptic operator. This follows since the spectral equivalence between the bilinear form $a(\cdot, \cdot)$ and the H^1-norm has been used in the proof. For other modifications of stabilized Schwarz methods, including use of aggregations and special treatment of interface regions when the coarse grid is extended not only to interface points but involves more points between the subdomains, see the work of Blaheta.[13] In the next subsection, the proposed interpolation for some intermediate points on the interfaces can be said to replace the local refinement method along interfaces used in the work of Blaheta.[13] Clearly, the interpolation method is simpler.

2.2.2 | Coarse–fine-mesh stabilization

There is a further improved version of the two-level method. In this method, we replace the coarse-mesh matrix with the coarse–fine matrix, based on narrow triangular mesh elements, as illustrated in Figure 3. In this case, there is no need to use extended interfaces as in the work of Blaheta.[13]

Hence, the matrix corresponds to the coarse-mesh points and all interior mesh points on the interfaces. Because we use an overlapping Schwarz method, here, it suffices to use points only on the set of even-numbered interfaces. In this method, there is no need to use interpolation to find the values at the inner node points on the interfaces, since they are already given from the solution of the fine–coarse-mesh system. However, to reduce the solution cost, one can use a method with fewer mesh points, for example, only each second or each fourth node point on the intersecting lines, solve

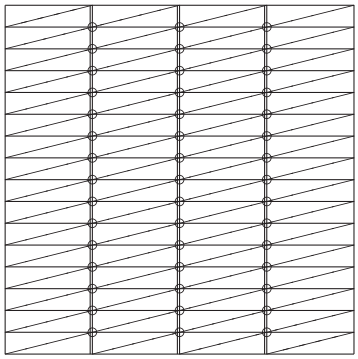

FIGURE 3 Coarse–fine-mesh nodes (four strips)

the corresponding smaller-sized system, and, then, interpolate the solution to the mesh points not used. The extreme choices of the coarse grid are where (a) only vertices of odd strip subdomains and (b) all nodes on the odd strip interfaces are used, respectively. By the use of interpolation, we can handle intermediate choices. This is another novel approach in the paper.

The method takes the following form.

Coarse–fine-mesh stabilization; defect-correction version

Choose a coarse–fine mesh including all coarse points on the even-numbered interfaces and all, each second, or each fourth node points on these edges and form the corresponding matrix $A_{c\Gamma}$.

Given an initial approximation $u^{(0)}$, for $k = 0, 1, \ldots$ until convergence.

1. Compute the residual $\delta f^{(k)} = f - Au^{(k)}$. Solve the coarse–fine-mesh system

$$A_{c\Gamma} \ \delta u^{(k)} = \delta f^{(k)}.$$

   If applicable, interpolate $\delta u^{(k)}$ to the remaining node points on the even-numbered interfaces.
2. Fine-mesh alternating Schwarz

   Apply once, or more times, the alternating Schwarz method to solve $A\delta u^{(k)} = \delta f^{(k)}$. Update the solution, $u^{(k+1)} = u^{(k)} + \delta u^{(k)}$.

Repeat.

In this defect-correction version of the method, we make full use of all components of the solution as computed in the alternating Schwarz method.

In these methods, the local residual errors in the direction of the interface lines are small, $O(h^{1+\alpha})$, and there are no or only correspondingly small interpolation errors. There will, in general, be error components of size $O(H^{1+\alpha})$ along the orthogonal direction, but they are quickly damped out by the use of the two steps of the alternating Schwarz method.

Here, numerical results indicate a very fast and optimal rate of convergence, independent of both H and h, and show a dramatic improvement compared to the two-level method with just a coarse-grid stabilization. In addition, the discontinuities in the normal derivatives of the solution, which are due to discontinuous coefficients in the differential operator, have no influence on the rate of convergence. The number of Schwarz iterations is very small and nearly constant and independent of mesh and problem parameters; hence, the method behaves closely as a direct, but very low cost, solution method. Since we use a nearly full (100%) overlapping method by the use of combined odd- and even-numbered sets both covering the whole domain, an upper bound of the iterative convergence factor can be derived by computing the CBS constant for just the combination of the odd-numbered set of interior-node subdomains and the coarse–fine interface set of node points. This is done in the Appendix.

Clearly, the solution of the fine–coarse stabilization matrix is more expensive than the solution of the coarse-mesh matrix systems. The latter can normally be done best by a direct solution method. For the coarse–fine matrix system, an iterative solution method should be used, however. Further tests indicate that it needs much fewer subdomain solves if the MIC(2) method is used for both the subdomain solves and the coarse–fine-mesh systems.

The MIC preconditioning method does not result in a condition number bounded uniformly with respect to h. Rather, the condition number increases as $O(h^{-1})$ (see, e.g., the works of Axelsson[9] and Gustafsson[17]), and the number of PCG iterations increases as $O(h^{-1/2})$. For discussions regarding the rate of convergence of the PCG method, see, for example,

the works of Axelsson and Barker,[8] Axelsson,[9] and Axelsson and Lindskog[20] as well as the references therein. This gives anyhow a small increase in the number of iterations as $h \to 0$. The numerical tests indicate indeed an increase in computer time with a factor smaller than the theoretical estimate $4\sqrt{2}$, that is, a factor-4 increase in the number of node points and a factor-2 increase in the condition number, which holds when h^{-1} is doubled. Other possible methods could be the use of a coarse–fine two-level or multilevel method, which leads to an optimal-order method. Since already the simpler approach using the MIC methods works very well and since the main purpose of this paper is to show the strong improvement in using a combination of alternating Schwarz and coarse–fine-mesh stabilization, for simplicity, we have not considered such methods here.

As remarked earlier, in common domain decomposition methods, one uses combined Neumann–Dirichlet boundary conditions at interfaces (see, e.g., the works of Chan and Mathew[2] and Toselli and Widlund[6]) implemented as constraints. This causes complications in solving the arising systems that are then of saddle-point type. In our method, we have avoided this simply by the use of interpolation of the coarse-mesh solution or by the use of a coarse–fine-mesh stabilization. Furthermore, by the use of the overlapping domain decomposition method, the flux interface balance condition is explicitly satisfied through the use of the finite element method applied globally.

3 | NUMERICAL TESTS

We consider the solution of an elliptic problem $\mathcal{L}u = f, f \equiv 1$ in Ω and $u = 0$ on $\partial\Omega$, with, in general, variable coefficients ϱ, where $\mathcal{L} = -\nabla \cdot (\varrho\nabla)$. For $\varrho = 1$, the exact solution is $u(x, y) = \frac{1}{4}(x(1-x) + y(1-y))$. For simplicity, here, ϱ is taken as a constant in each subdomain. The domain is a unit square, and a standard obtuse triangulation with piecewise-linear finite element basis functions is used.

All numerical tests are performed on a computer with two Intel Xeon CPU (3.40 GHz) and 4-GB RAM. In all results, the stopping criterion for the Schwarz iteration is $\|r^{(k)}\| \leq 10^{-6}\|r^{(0)}\|$. The coarse–fine-mesh systems are solved by the PCG method with MIC(2) as the preconditioner and the stopping tolerance $\|r^{(k)}\| \leq 10^{-2}\|r^{(0)}\|$.

We test first the number of iterations and computer execution time for (a) the pure version of the alternating Schwarz method and for two stabilized versions: (b) coarse-mesh stabilization and (c) coarse–fine-mesh stabilization. The tests are done for a fixed value of h but with varying H, that is, for a variable number of subdomains (strips).

TABLE 1 Total number of Schwarz iterations and computer times (in brackets) for $h = 1/128$ and a variable number of strips. Continuous coefficient

Number of strips (in bracket: value of H/h)	Relative stopping criteria (ε) on Ω_i	Method and stabilization (a)	(b)		Method (c)	
			Each iter.	Each 2nd iter.	Each iter.	Each 2nd iter.
8 (16)	10^{-2}	50 (6.3 s)	34 (4.9 s)	37 (5.0 s)	8 (1.6 s)	10 (2.0 s)
	10^{-4}	50 (7.5 s)	32 (6.9 s)	35 (6.9 s)	4 (1.5 s)	7 (2.4 s)
	10^{-6}	50 (8.5 s)	32 (9.3 s)	35 (8.4 s)	4 (2.2 s)	7 (3.4 s)
16 (8)	10^{-2}	195 (13.8 s)	58 (5.4 s)	74 (6.3 s)	6 (1.1 s)	10 (1.4 s)
	10^{-4}	195 (16.8 s)	49 (6.6 s)	71 (8.3 s)	4 (1.2 s)	6 (1.6 s)
	10^{-6}	195 (18.1 s)	49 (9.1 s)	71 (10.6 s)	4 (1.7 s)	5 (2.3 s)
32 (4)	10^{-2}	761 (43.3 s)	57 (4.4 s)	56 (4.1 s)	6 (1.9 s)	8 (1.8 s)
	10^{-4}	761 (46.2 s)	27 (3.6 s)	52 (5.8 s)	4 (1.8 s)	5 (1.8 s)
	10^{-6}	761 (49.8 s)	30 (5.7 s)	51 (8.3 s)	4 (2.2 s)	5 (2.2 s)

Table 1 shows the results for continuous coefficients for methods (a), (b), and (c), and Tables 2, 3, and 4 show the corresponding results for various degrees of discontinuities, varying alternatively as $1, \varrho, 1, \varrho, \ldots, 1, \varrho$ in the consecutive subdomains.

TABLE 2 Number of iterations and computer times for different degrees of discontinuities. Number of strips is eight; (1)—number of alternating Schwarz iterations; (2)—interval for the number of Ω_i subdomain iterations

h	Rel. stop. ε on Ω_i	Discontinuity (ϱ)	Method (a)			Method (c) (Coarse–fine mesh) Each iteration			Method (c) (Coarse–fine mesh) Each 2nd iteration		
			(1) Iter.	(2)	Time	(1) Iter.	(2)	Time	(1) Iter.	(2)	Time
$\frac{1}{32}$	10^{-2}	1	48	6–12	0.40 s	5	8–13	0.09 s	7	7–12	0.11 s
		2	51	–	0.44 s	5	–	0.09 s	7	–	0.11 s
		5	63	6–12	0.54 s	5	8–13	0.09 s	7	7–13	0.11 s
		10	74	–	0.61 s	6	–	0.10 s	7	–	0.11 s
		100	94	6–14	0.80 s	7	5–13	0.11 s	8	5–13	0.12 s
	10^{-4}	1	48	6–18	0.45 s	4	14–21	0.11 s	5	12–21	0.12 s
		2	51	–	0.47 s	4	–	0.11 s	5	–	0.13 s
		5	63	6–20	0.57 s	4	14–21	0.11 s	5	12–21	0.12 s
		10	74	–	0.65 s	3	–	0.10 s	5	–	0.13 s
		100	94	6–21	0.82 s	4	14–23	0.12 s	5	12–22	0.14 s
$\frac{1}{128}$	10^{-2}	1	50	20–50	6.3 s	8	21–45	1.7 s	10	22–42	2.0 s
		2	54	–	7.5 s	8	–	1.77 s	10	–	2.0 s
		5	66	20–50	8.8 s	14	17–33	3.05 s	10	18–44	2.1 s
		10	77	–	9.5 s	12	–	2.39 s	10	–	2.0 s
		100	99	20–50	11.9 s	13	13–45	2.71 s	11	18–44	2.3 s
	10^{-4}	1	50	20–70	8.17 s	4	50–80	1.6 s	7	35–78	2.3 s
		2	54	–	8.35 s	4	–	1.57 s	7	–	2.3 s
		5	66	20–80	10.6 s	4	50–80	1.57 s	7	34–82	2.3 s
		10	77	–	11.9 s	4	–	1.57 s	7	–	2.4 s
		100	99	20–80	16.12 s	4	50–80	1.73 s	5	46–78	1.9 s

TABLE 3 Number of alternating Schwarz iterations and computer times for different fine-mesh sizes (h), different numbers of strips (H^{-1}), and different levels of discontinuities (ϱ). The tolerance for the Ω_i subdomain iterations in method (a) is $\|r^{(k)}\| \leq 10^{-2}\|r^0\|$. In method (c), the tolerance for the Ω_i subdomain iterations in the first Schwarz iteration is $\|r^{(k)}\| \leq 10^{-5}$, and in further Schwarz iterations, it is $\|r^{(k)}\| \leq 10^{-9}$. The Ω subdomain problems have been solved with MIC(2) as the preconditioner. The results are the same for column-wise and rowwise orderings

h	H^{-1}	Discontinuity (ϱ)	Method (a) Iter.	Method (a) Time	Method (c) Iter.	Method (c) Time
$\frac{1}{128}$	8	1	50	2.8 s	4	0.36 s
		10	77	3.9 s	4	0.38 s
		100	99	5.5 s	4	0.37 s
	16	1	195	6.8 s	4	0.7 s
		10	295	10.8 s	4	0.9 s
		100	377	13.0 s	4	0.9 s
$\frac{1}{256}$	8	1	51	15.6 s	5	2.1 s
		10	79	27.6 s	4	2.2 s
		100	101	34.8 s	3	2.1 s
	16	1	200	46.0 *vs*	4	2.4 s
		10	301	70.1 s	4	3.0 s
		100	386	87.6 s	5	3.1 s

TABLE 4 Number of Schwarz iterations and computer times for method (a) for different problem sizes. Number of strips is eight, and the stopping criterion for the Ω_i subdomain iterations is $\|r^{(k)}\| \leq 0.01\|r^{(0)}\|$

Discontinuity	$h = 1/64$		$h = 1/128$		$h = 1/256$	
	Number of Schwarz iterations	Time	Number of Schwarz iterations	Time	Number of Schwarz iterations	Time
1	49	1.14 s	50	6.3 s	51	52.5 s
10	76	1.76 s	77	9.5 s	79	81.8 s
100	96	2.15 s	99	11.9 s	101	103.6 s

In Tables 1, 2, and 4, we have used eight strips, and in Table 3, we present the results for both 8 and 16 strips. Here, only methods (a) and (c) have been tested. In Tables 1, 2, and 4, the subdomain equations are solved with Cholesky incomplete factorization, whereas in Table 3, MIC(2) incomplete factorization has been used.

Table 4 shows the number of alternating Schwarz iterations and computer times for different values of the fine-mesh parameter h and the relative stopping criterion $\varepsilon = 10^{-2}$ in Ω_i.

In accordance with the theory, Table 4 shows that the number of alternating Schwarz iterations depends only on the number of subdomains, that is, on H but not on h, for a fixed value of H. Since the number of iterations is nearly independent of the relative stopping criterion used, Table 1 indicates also that there is no need to solve the subdomain equations to a very high degree of relative accuracy. It was observed that for method (a), already $\varepsilon = 10^{-2}$ gives the same number of Schwarz iterations as if the subdomain problems were solved exactly.

In accordance with the theory, the number of iterations of the coarse-mesh-stabilized method is bounded for all values of H. The theory shows that they are also bounded for all values of h.

Table 1 indicates even that the number of iterations for the coarse-mesh-stabilized method can decrease for small values of the ratio H/h. Moreover, here, it is seen that it suffices with a not very small relative tolerance in solving the subdomain systems.

Table 3 shows again that the number of iterations in method (a) depends essentially only on the number of subdomains, that is, increases linearly with H^{-2}.

Furthermore, it shows that method (c) needs about the same computer time for 16 strips as for eight strips. This is very important since it shows that the gain in computer time using parallel computer with 16 processors as compared to eight can be close to the optimal speedup factor 2. Moreover, this holds even stronger for the fine mesh ($h = 2^{-8}$ as compared with $h = 2^{-7}$). This indicates that even if we solve the problem on a much finer mesh than done in this paper, the favorable results will be even further strengthened. It is also seen that the increase in computer time when doubling h^{-1} for $H^{-1} = 16$ subdomains is only about 3.4, that is, much smaller than the upper bound $4\sqrt{2}$, mentioned in the previous section.

Tables 1, 2, and 3 show that the coarse–fine stabilization method decreases the number of iterations significantly and to a small, fixed number if the relative stopping criteria for the inner subdomain solves are sufficiently small. In Table 2, the interval in which the total number of subdomain iterations varied is also shown.

4 | IMPLEMENTATIONAL ISSUES

While the previous section mainly considered how the rate of convergence of the methods depends on the parameters h and H, in this section, we take up some implementational issues. As has already been commented on, to lower the number of iterations and computer time, it is important to use efficient inner stopping criteria.

We comment first on the effects of using coupled outer–inner iterations. Such methods can be seen as nonlinear iteration methods, and the convergence can be sensitive to the inner-iteration accuracy. This holds if the outer-iteration condition number is large. For instance, even if the iteration matrix is symmetric, one gets normally complex-valued resulting eigenvalues. However, as will be seen in the Appendix, in the stabilized methods, the condition number is not large, and we hardly observe any effect at all of the use of inner iterations, if we use a reasonable stopping tolerance.

When solving the linear systems on the subdomains (Ω_i) in method (c) by the use of a relative stopping criterion $\|r^{(k)}\| \leq \varepsilon\|r^{(0)}\|$, the optimal choice of ε turns out to be about 10^{-4} in general in our tests. However, by the use of this relative stopping criterion, a number of unnecessary iterations are performed. In the first Schwarz iteration, this is so since

the solutions of the subdomain systems are rather far from the solution of the global system. In the following Schwarz iterations, the reason is the fact that the starting residuals $\|r^{(0)}\|$ in the solution of the subdomain systems are small, particularly in the final Schwarz iterations, where the values of $\|r^{(0)}\|$ are close to the stopping value for the Schwarz iterations. Therefore, by the use of an absolute stopping criterion based on the known stopping value for the Schwarz iterations, the total number of subdomain iterations can be much decreased. Compared to the tolerance for the Schwarz iterations, we use a weaker absolute tolerance for the subdomain iterations in the first Schwarz iteration and a somewhat stronger absolute tolerance for the subdomain iterations in the following Schwarz iterations. This choice of the absolute stopping criterion for the Ω_i subdomain iterations is used in method (c) in Tables 3 and 5–8. Besides the stopping values for the Ω_i subdomain iterations, we also give the stopping values for the Schwarz iterations in Tables 6 and 7.

For instance, for the case $h = 1/128$ (and $H = 1/8$) and the discontinuity $\rho = 100$ (see Table 2), the stopping criterion for the Schwarz iterations is $\|r^{(k)}\| \leq 10^{-6}\|r^{(0)}\| \approx 7.710^{-9}$. By the use of the relative stopping criterion $\|r^{(k)}\| \leq 10^{-4}\|r^{(0)}\|$ for the subdomain iterations, we get, in total, 1,801 iterations. However, use of the absolute criterion $\|r^{(k)}\| \leq 10^{-5}$ for the subdomain systems in the first Schwarz iteration and the absolute criterion $\|r^{(k)}\| \leq 4.410^{-9}$ for the subdomain systems in the following Schwarz iterations will decrease the total number of subdomain iterations to 1,223, that is, a significant reduction. See also Table 5. Note the large improvement compared to method (a) in Table 4.

In Tables 6 and 7, the total number of subdomain iterations, that is, summed up over all Schwarz iterations, is given for Cholesky incomplete factorization (Table 6) and MIC(2) incomplete factorization (Table 7), when solving the subdomain systems. The actual value of the residual at stopping is also given. One can notice a very substantial difference in the total number of iterations. Note the doubling of iterations for the Cholesky factorization method when halving the value of h, but for the MIC(2) method, the increase in the total number of subdomain iterations is only about 50%.

In the previous tests, a zero vector has been used as the initial vector. We test now with a random initial vector $u^{(0)}$, with components $0 \leq u_i^{(0)} \leq 0.01$, and with zero initial vector but a random right-hand side vector (see Table 8). We have also tested random discontinuous coefficients on the strips in the case with $h = 1/128, H = 1/16, u^{(0)} = 0, f = 1$, and

TABLE 5 Total number of Schwarz iterations and number of Ω_i iterations for different choices of stopping tolerances for the Ω_i iterations in method (c). $h = 1/128, H = 1/8$

Discontinuity (ρ)	Stopping tolerances for the Ω_i iterations $\|r^{(k)}\| \leq 10^{-4}\|r^{(0)}\|$			$\|r^{(k)}\| \leq 10^{-5}$ in Schwarz iter.1 $\|r^{(k)}\| \leq 4 \times 10^{-9}$ in Schwarz iter.2,3,4		
	Number of Schwarz iterations	Number of Ω_i iter.	Time	Number of Schwarz iterations	Number of Ω_i iter.	Time
1	4	1,794	1.60 s	4	1,370	1.20 s
10	4	1,750	1.57 s	4	1,293	1.19 s
100	4	1,801	1.75 s	4	1,223	1.13 s

TABLE 6 Number of Schwarz iterations and total number of subdomain iterations for method (c) for different problem sizes and degrees of discontinuities. $H = 1/8$. Preconditioning for subdomain iterations: incomplete Cholesky

	$h = 1/128$			$h = 1/256$		
Discontinuity (ρ)	Number of Schwarz iter. with $\|r^{(k)}\| \leq 10^{-6}\|r^{(0)}\| = 7.7 \times 10^{-9}$	Total number of Ω_i iter. with stopping crit. Iter.1: $\|r^{(k)}\| \leq 10^{-5}$ Iter. 2,3,4: $\|r^{(k)}\| \leq 4 \times 10^{-9}$	Time	Number of Schwarz iter. with $\|r^{(k)}\| \leq 10^{-6}\|r^{(0)}\| = 3.8 \times 10^{-9}$	Total number of Ω_i iter. with stopping crit. Iter.1: $\|r^{(k)}\| \leq 10^{-5}$ Iter. 2,3,4: $\|r^{(k)}\| \leq 10^{-9}$	Time
1	4	1,370	1.20 s	5	3,281	12.0 s
2	4	1,310	1.18 s	4	2,966	11.0 s
5	4	1,435	1.23 s	4	2,893	10.5 s
10	4	1,293	1.19 s	4	2,848	10.7 s
100	4	1,223	1.13 s	3	2,495	9.3 s

TABLE 7 Number of Schwarz iterations and total number of subdomain iterations for method (c) for different problem sizes and degrees of discontinuities. $H = 1/8$. Preconditioner for subdomain iterations: MIC(2)

	$h = 1/128$			$h = 1/256$		
Discontinuity (ρ)	**Number of Schwarz iter. with** $\|\|r^{(k)}\| \le 10^{-6}\|\|r^{(0)}\| = 7.7 \times 10^{-9}$	**Total number of Ω_i iter. with stopping crit. Iter.1:** $\|\|r^{(k)}\| \le 10^{-5}$ **Iter. 2,3,4:** $\|\|r^{(k)}\| \le 5 \times 10^{-9}$	**Time**	**Number of Schwarz iter. with** $\|\|r^{(k)}\| \le 10^{-6}\|\|r^{(0)}\| = 3.8 \times 10^{-9}$	**Total number of Ω_i iter. with stopping crit. Iter.1:** $\|\|r^{(k)}\| \le 10^{-6}$ **Iter. 2,3,4:** $\|\|r^{(k)}\| \le 10^{-9}$	**Time**
1	4	212	0.34 s	4	339	2.11 s
2	4	209	0.33 s	4	338	2.27 s
5	4	205	0.35 s	4	328	2.30 s
10	4	204	0.35 s	4	323	2.28 s
100	4	192	0.34 s	4	290	1.99 s

TABLE 8 Number of Schwarz iterations and computer times for $h = 1/128$ and $H = 1/8$ for different levels of discontinuities (ρ). The tolerance for the Ω_i iterations in method (a) is $\|r^{(k)}\| \le 10^{-2}\|r^{(0)}\|$. In method (c), with $u^{(0)}$ being random and $f = 1$, the tolerance is $\|r^{(k)}\| \le 10^{-3}$ for the Ω_i iterations in the first Schwarz iteration, and for the Ω_i iterations in further Schwarz iterations, it is $\|r^{(k)}\| \le 10^{-6}$ in the case with $\rho = 1$, $\|r^{(k)}\| \le 0.5 \times 10^{-5}$ in the case with $\rho = 10$, and $\|r^{(k)}\| \le 0.5 \times 10^{-9}$ in the case with $\rho = 100$. In the case with $u^{(0)} = 0$ and f being random, the tolerance is $\|r^{(k)}\| \le 10^{-4}$ for the Ω_i iterations in the first Schwarz iteration and $\|r^{(k)}\| \le 2 \times 10^{-9}$ for the Ω_i iterations in further Schwarz iterations. The Ω_i systems are solved with incomplete Cholesky factorization as the preconditioner

$u^{(0)}, f$	**Discontinuity** (ρ)	**Method (a)**		**Method (c)**	
		Iter.	**Time**	**Iter.**	**Time**
$u^{(0)}$ random	1	33	4.3 s	5	0.70 s
$f = 1$	10	39	5.2 s	3	0.69 s
	100	40	5.6 s	3	0.54 s
$u^{(0)} = 0$	1	50	6.2 s	5	1.6 s
f random	10	77	9.7 s	5	1.6 s
	100	98	12.4 s	5	1.7 s

tolerance $\|r^{(k)}\| \le 10^{-2}\|r^{(0)}\|$ for the Ω_i iterations. For method (a), we get 193 Schwarz iterations in 6.6 s, and for method (c), we get only seven Schwarz iterations in 1.9 s. It is seen that the methods perform in a similar way as for the zero initial vector and the constant right-hand side vector, that is, it is fairly robust also with respect to variations of the initial vector and the right-hand side vector. Moreover, although one can observe some nonmonotone behavior with respect to the initial stopping criteria, this indicates that the variations are small.

We comment now shortly on the possibility to parallelize the methods. Clearly, by the use of subdomains, both the odd- and even-numbered groups, one can use a coarse-grain parallelization when solving the subdomain problems. For this, the number of parallel processors should ideally be equal to $1/H$. Hence, since H will not be extremely small, the method is more suitable for a more modest number of parallel computers. If each processor consists of a number of parallel cores, they can be utilized by the use of an internal subdomain decomposition of each subdomain and the use of the Schwarz alternating method with a coarse or coarse–fine stabilization for each. Hence, we can use the parallelizable methods on two levels, global and local. In particular, when solving three-dimensional problems, this can be an efficient way to get acceptable solution times.

The coarse-mesh-stabilized system can be solved by the use of a red–black (odd–even) or multicoloring method to utilize some available parallelism. The coarse–fine-stabilized system can be solved using various methods, such as mesh

node splittings in coarse and edge nodes, which can enable a parallel solution of approximate block-diagonal systems in the preconditioner. A novel approach is to use additive low-rank approximations of the inverse of the Schur complement matrix arising in a multiplicative, that is, approximated, inverse preconditioner (see the work of Axelsson and Blaheta[21]).

5 | CONCLUDING REMARKS

We have shown that there is a simple and efficient way to implement the Schwarz alternating iteration method for a domain decomposition method in strips. It can be stabilized by a coarse–fine-mesh preconditioner to yield a method that performs nearly as an optimal order of computational complexity method in the respect that the computational work is nearly proportional to the number of unknowns, and this holds uniformly with respect to both mesh discretization and problem parameters. In addition, it offers the possibility to parallelize the computations. The number of Schwarz alternating iterations is very small, nearly fixed with a good initial coarse mesh, and essentially independent of coefficient jumps, which makes the method highly competitive with many other existing domain decomposition methods.

As an illustration of the impressive reductions in the number of alternating Schwarz iterations, see, for example, the results in Table 1.

If $m = H/h$ is constant, the method has an optimal order of rate of convergence, independent of h, that is, the order of the global matrix. The total cost varies with H/h, at most, as $O\left((H/h)^{1/2}\right)$ times the total number of unknowns. A typical value of H/h could be 8 or 16, resulting in a factor increase of $2\sqrt{2}$ and 4, respectively. If there are $\frac{1}{2m} = h/2H$ parallel processors available, the solution cost will be nearly as small as for the solution of a single subdomain equation. If each processor contains a number of parallel working cores, the solution cost can be further diminished by applying a similar domain decomposition method for each subdomain problem, that is, in a two-level version of domain decomposition. The parallel computations benefit further from the absence of global inner product, that is, less need for synchronization and startup times.

In the method, the interface conditions are implemented explicitly by a standard finite element method, in this way avoiding the use of additional flux interface conditions that are frequently used in domain decomposition methods.

The method has been implemented for a two-dimensional (2D) elliptic problem with discontinuous diffusion coefficients. For three-dimensional (3D) problems, one can use the same type of method for the coarse–fine interface equation, which is then a 2D problem. Furthermore, for the 3D layers, one can use domain decomposition in boxes or inner subdomain strips to enable fast inner solutions also for 3D subdomain equations. One can then use a two-level strategy, that is, the inner subdomain solutions involve the 2D domain decomposition framework. However, a full presentation of 3D domain decomposition methods must be considered as a topic for a separate paper. It can also be interesting to consider the use of alternating coarsening directions, such as in the balanced algebraic multilevel iteration method (see the work of Lymbery and Margenov[22]). Another topic that can be considered is a method based on red–black ordering of node points.

ACKNOWLEDGEMENTS

The major numerical results in this paper were derived by Gunhild Lindskog. To our great regret, Gunhild passed away before we could submit our paper. We acknowledge her substantial contribution and suggestions for several improvements in the implementation of the numerical results.

This work was supported by the Ministry of Education, Youth and Sports through the National Programme of Sustainability (NPU II) Project "IT4Innovations excellence in science—LQ1602."

The authors declare that they have no conflict of interest.

ORCID

Owe Axelsson https://orcid.org/0000-0001-8753-2429

REFERENCES

1. Schwarz HA. Gesammelte mathematische abhandlungen. Vol. 2. Berlin, Germany: Springer-Verlag, 1890; p. 133–143. First published in 1870.
2. Chan TF, Mathew TP. Domain decomposition algorithms. Acta Numer. 1994;3:61–143.

3. Smith BF, Bjørstad PE, Gropp WD. Domain decomposition: Parallel multilevel methods for elliptic partial differential equations. New York, NY: Cambridge University Press; 1996. ISBN 0-521-49589-X.
4. Quarteroni A, Valli A. Domain decomposition methods for partial differential equations. Oxford, UK: Oxford University Press; 1999. ISBN 978-0-198-50178-7.
5. Xu J. Iterative methods by space decomposition and subspace correction. SIAM Rev. 1992;34(4):581–613.
6. Toselli A, Widlund O. Domain decomposition methods — algorithms and theory. Berlin, Germany: Springer Science & Business Media; 2004. ISBN 978-3-540-20696-5.
7. Axelsson O, Gustafsson I. Preconditioning and two-level multigrid methods of arbitrary degree of approximation. Math Comp. 1983;40(161):219–242.
8. Axelsson O, Barker VA. Finite element solution of boundary value problems: Theory and computation. Philadelphia, PA: SIAM; 2001. ISBN 978-0-89871-499-9.
9. Axelsson O. Iterative solution methods. Cambridge, UK: Cambridge University Press; 1996. ISBN 978-0-521-55569-2.
10. Axelsson O, Vassilevski PS. Algebraic multilevel preconditioning methods, I. Numer Math. 1989;56(2):157–177.
11. Axelsson O, Vassilevski PS. Algebraic multilevel preconditioning methods, II. SIAM J Numer Anal. 1990;27(6):1569–1590.
12. Vassilevski PS. Multilevel block factorization preconditioners. New York, NY: Springer; 2008. ISBN 978-0-387-71563-6.
13. Blaheta R. Space decomposition preconditioners and parallel solvers. In: Feistauer M, Dolejší V, Knobloch P, Najzar K, editors. Numerical mathematics and advanced applications (Enumath 2003). Berlin, Germany: Springer, 2004; p. 20–38.
14. Axelsson O, Blaheta R. Preconditioning of matrices partitioned in 2× 2 block form: eigenvalue estimates and Schwarz DD for mixed FEM. Numer Linear Algebra Appl. 2010;17(5):787–810.
15. Blaheta R, Kohut R, Neytcheva M, Stary J. Schwarz methods for discrete elliptic and parabolic problems with an application to nuclear waste repository modelling. Math Comput Simul. 2007;76(1–3):18–27.
16. Axelsson O, Polman B. On approximate factorization methods for block matrices suitable for vector and parallel processors. Linear Algebra Appl. 1986;77:3–26.
17. Gustafsson I. A class of first order factorization methods. BIT Numer Math. 1978;18:142–156.
18. Widlund O. The development of coarse spaces for domain decomposition algorithms. In: Bercovier M, Gander M, Kornhuber R, Widlund O, editors. Domain decomposition methods in science and engineering XVIII, Lecture notes in computational science and engineering. Vol. 70. Berlin, Germany: Springer-Verlag, 2009; p. 241–248. ISBN 978-3-642-02676-8.
19. Axelsson O, Blaheta R. Two simple derivations of universal bounds for the C.B.S. inequality constant. Appl Math. 2004;49(1):57–72.
20. Axelsson O, Lindskog G. On the rate of convergence of the preconditioned conjugate gradient method. Numer Math. 1986;48(5):499–523.
21. Axelsson O, Blaheta R. Low-rank improvements of two-level grid preconditioned matrices. J Comput Appl Math. 2018;340:432–442.
22. Lymbery M, Margenov S. Robust balanced semi-coarsening AMLI preconditioning of biquadratic FEM systems. AIP Conf Proc. 2011;1404(1):438–447.
23. Blaheta R. Nested tetrahedral grids and strengthened C.B.S. inequality. Numer Linear Algebra Appl. 2003;10(7):619–637.
24. Maitre JF, Musy F. The contraction number of a class of two-level methods; an exact evaluation for some finite element subspaces and model problems. In: Hackbusch W, Trottenberg U, editors. Multigrid methods, Lecture Notes in Mathematics. Vol. 960. Berlin, Germany: Springer, 1982; p. 535–544. ISBN 978-3-540-39544-7.
25. Margenov SD. Upper bound of the constant in the strengthened C.B.S. inequality for FEM 2D elasticity equations. Numer Linear Algebra Appl. 1994;1(1):65–74.
26. Achchab B, Axelsson O, Laayouni L, Souissi A. Strengthened Cauchy–Bunyakowski–Schwarz inequality for a three-dimensional elasticity system. Numer Linear Algebra Appl. 2001;8(3):191–205.
27. Axelsson O. On multigrid methods of the two-level type. In: Hackbusch W, Trottenberg U, editors. Multigrid methods, Lecture Notes in Mathematics. Vol. 960. Berlin, Germany: Springer, 1982; p. 352–367. ISBN 978-3-540-39544-7.
28. Napov A, Notay Y. An algebraic multigrid method with guaranteed convergence rate. SIAM J Sci Comput. 2012;34(2):A1079–A1109.

How to cite this article: Axelsson O, Gustafsson I. A coarse–fine-mesh stabilization for an alternating Schwarz domain decomposition method. *Numer Linear Algebra Appl.* 2019;e2236. https://doi.org/10.1002/nla.2236

APPENDIX

TWO-LEVEL AND HIERARCHICAL BASIS FUNCTION PRECONDITIONERS

A lower bound of the rate of convergence of the coarse–fine-stabilized Schwarz method can be estimated by computing the CBS hierarchical two-level constant for the set of combination of interior subdomain node points with the set of interface node points.

As has been shown in several publications,[7,9,10,12,23–27] two-level hierarchical basis function methods are useful in constructing preconditioners that lead to a number of iterations uniformly bounded with respect to the mesh size parameter h. Although there are already many such publications, for the benefit of possible new readers, we make here a more elementary presentation of the topic. We consider then an elliptic problem for which the bilinear form is

$$a(u,v) = \int_\Omega \sum_{i,j=1}^{2} a_{ij} \frac{\partial u}{\partial x_j} \frac{\partial v}{\partial x_i} dx,$$

where $u, v \in H^1(\Omega)$ and $[a_{ij}]$ is a matrix of problem coefficients, assumed to be symmetric and positive definite. For simplicity, we assume further that they are constant on each macroelement. Instead of piecewise-linear basis functions for both the coarse and the fine mesh, one can use hierarchical basis functions, that is, normally piecewise linear for the coarse, vertex mesh points and higher order for the refined mesh nodes. This extends standard versions of domain decomposition methods.

For illustration, consider a triangular mesh with edge lengths of order H, where each triangle is uniformly divided into conforming triangles by the successive halving of the edges. Let U_H be the set of piecewise-linear basis functions associated with the vertex point of the original triangles, and let V_h be the set of basis functions associated with interior and edge node points. They can be piecewise linear or piecewise polynomials of degree $p \geq 2$. We choose $p = 2, 4$, or 6, etc.

Then, the hierarchical basis function matrix takes the form $\hat{A} = \begin{bmatrix} \hat{A}_{11} & \hat{A}_{12} \\ \hat{A}_{21} & \hat{A}_{22} \end{bmatrix}$, where $\hat{A}_{22}$ corresponds to the vertex node points. We are interested in preconditioning $\hat{A}$ with its block-diagonal matrix $\begin{bmatrix} \hat{A}_{11} & 0 \\ 0 & \hat{A}_{22} \end{bmatrix}$ or in preconditioning the Schur complement matrix $\hat{S} = \hat{A}_{22} - \hat{A}_{21}\hat{A}_{11}^{-1}\hat{A}_{12}$ with $\hat{A}_{22}$. The off-diagonal matrices $\hat{A}_{ij}$, $i = j$, $i,j = 1, 2$, are less sparse; hence, one uses normally a replacement of them with an interpolation matrix J. However, it can be seen that the Schur complement matrix $S = A_{22} - A_{21}A_{11}^{-1}A_{12}$ for the standard basis function matrix, that is, where we use piecewise-linear or p-order basis functions for corresponding to the subdivided interior triangles also for the vertex points, is equal to $\hat{S}$. Therefore, we consider now only methods based on the Schur complements.

As shown in previous publications, such as those by Axelsson and Gustafsson[7] and Axelsson,[9] the condition number can be easily computed as follows. Let

$$\gamma \begin{bmatrix} \hat{A}_{11} & 0 \\ 0 & \hat{A}_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \hat{\mathcal{A}} \begin{bmatrix} x \\ y \end{bmatrix}, \qquad \|x\| + \|y\| \neq 0,$$

that is,

$$\gamma \begin{bmatrix} \hat{A}_{11} & 0 \\ 0 & \hat{A}_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \hat{A}_{12} y \\ \hat{A}_{21} x \end{bmatrix}.$$

Here, we have

$$\max |\gamma - 1| = \varrho\left(\hat{A}_{11}^{-1/2}\hat{A}_{12}\hat{A}_{22}^{-1/2}\right);$$

hence, it remains to estimate the spectral radius of the normalized off-diagonal blocks. To do that, consider the bilinear form $a(u,v)$. It holds that

$$|a(u,v)| \leq \gamma (a(u,u)a(v,v))^{1/2}, \qquad \forall u \in U, v \in V,$$

where the constant in the strengthened CBS inequality $\gamma < 1$ since $U \cap V = \{0\}$. Note that γ equals the cosine for the angles between the spaces U and V measured in a metric (inner product) defined by $a(u,v)$. We scale the variables so that $a(u,u) = a(v,v)$. Then, we have

$$|a(u,v)| \leq \frac{1}{2}\gamma \left(a(u,u) + a(v,v)\right).$$

We can compute the constant γ locally on each macroelement E, that is,

$$a_E(u,v) \leq \frac{1}{2}\gamma_E \left(a(u,u) + a(v,v)\right) \quad u \in U_E, v \in V_E.$$

Then, by assembly, we obtain

$$a(u,v) \leq \frac{1}{2}\gamma \left(a(u,u) + a(v,v)\right) \quad u \in U_E, v \in V_E,$$

where $\gamma = \max_E \gamma_E$. This implies that it suffices to compute the CBS constant locally, which does not require much

computation. Several different methods based on local forms of the finite element basis functions have been used to estimate γ. Here, we use a new simple approach based on Richardson extrapolation, which is possible since we normally use regularly refined meshes. From the above follows

$$(1-\gamma)\left(x^T\hat{A}_{11}x+y^T\hat{A}_{22}y\right)\le(x,y)^T\mathcal{A}\begin{bmatrix}x\\y\end{bmatrix}\le(1+\gamma)\left(x^T\hat{A}_{11}x+y^T\hat{A}_{22}y\right),\ \forall x,y;$$

thus, the condition number of the block-diagonally preconditioned hierarchical basis function matrix is equal to $(1+\gamma)/(1-\gamma)$.

As remarked above, it can be more efficient to consider preconditioning of the Schur complement S, where we recall that $S=\hat{S}$. Therefore, we have

$$(1-\gamma^2)x^T\hat{A}_{22}x\le x^TSx\qquad\forall x;$$

hence, the condition number of the preconditioned Schur complement matrix is equal to $1/(1-\gamma^2)$.

As an example, consider triangles divided into four congruent triangles. Let the triangular angles be φ_i, $i=1,2,3$, and let $a=\cot\varphi_1, b=\cot\varphi_2, c=\cot\varphi_3$, and $d=a+b+c$. Then, an elementary computation shows that the refined mesh with a piecewise-linear basis function is equal to $A^{(1)}_{H/2}=\begin{bmatrix}A^{(1)}_{11}&A^{(1)}_{12}\\A^{(1)}_{21}&A^{(1)}_{12}\end{bmatrix}$, where

$$A^{(1)}_{11}=\begin{bmatrix}d&-c&-b\\-c&d&-a\\-b&-a&d\end{bmatrix},\qquad A^{(1)}_{12}=\left[A^{(1)}_{21}\right]^T=\begin{bmatrix}0&-a&-a\\-b&0&-b\\-c&-c&0\end{bmatrix},$$

and

$$A^{(1)}_{22}=\begin{bmatrix}b+c&0&0\\0&a+c&0\\0&0&a+b\end{bmatrix}.$$

Furthermore, the quadratic basis function matrix is equal to $A^{(2)}_{ij}=[A^{(2)}_{ij}]^T$, where $A^{(2)}_{ij}=\frac{4}{3}A^{(1)}_{ij}$, $i,j=1,1$ and $i,j=1,2$, and

$$A^{(2)}_{22}=\frac{1}{6}\begin{bmatrix}3(b+c)&c&b\\c&3(a+c)&a\\b&a&3(a+b)\end{bmatrix}.$$

The matrix corresponding to piecewise-linear basis functions on the whole triangle is equal to

$$A^{(1)}_{ij}=\frac{1}{2}\begin{bmatrix}(b+c)&-c&-b\\-c&(a+c)&-a\\-b&-a&(a+b)\end{bmatrix}.$$

It is seen that

$$A^{(2)}_{H}=\frac{4}{3}A^{(1)}_{H/2}-\frac{1}{3}\begin{bmatrix}0&0\\0&A^{(1)}_{H}\end{bmatrix}.\tag{A1}$$

This is a familiar relation; see, for example, the work of Axelsson and Blaheta.[19] We show now that such a relation can be derived more generally based on Richardson H-extrapolation, to derive locally higher-order accurate finite element approximations and corresponding matrices. Consider the Taylor expansion along any direction, that is,

$$\Delta_h:=\frac{1}{h^2}\left(u(x+h)+u(x-h)-2u(x)\right)=u^{(2)}(x)+h^2\frac{u^{(4)}(x)}{12}+h^4\frac{u^{(6)}(x)}{360}+\cdots,$$

and the refined difference, that is,

$$\Delta_{h/2}:=\frac{4}{h^2}\left(u\left(x+\frac{h}{2}\right)+u\left(x-\frac{h}{2}\right)-2u(x)\right)=u^{(2)}(x)+\frac{h^2}{4}\frac{u^{(4)}(x)}{12}+\frac{h^4}{16}\frac{u^{(6)}(x)}{360}+\cdots.$$

Then, we have

$$\frac{4}{3}\Delta_{h/2}-\frac{1}{3}\Delta_h=u^{(2)}(x)-\frac{h^4}{4}\frac{u^{(6)}(x)}{360}+\cdots,$$

and for the corresponding matrices, we get (A1). For regularly refined meshes, it follows that the corresponding recursive matrix relation (A1) holds. This can be repeated by the successive halving of the edges in the triangles and Richardson extrapolation to eliminate the next dominating error terms to get

$$A^{(2k)}_{H/k}=\frac{(2k)^2}{(2k)^2-1}A^{(2k-1)}_{H/2k}-\frac{1}{(2k)^2-1}\begin{bmatrix}0&0\\0&A^{(2k-1)}_{H/(2k-2)}\end{bmatrix}\qquad k=1,2,\ldots,\tag{A2}$$

whereby the successive remainder terms in the Taylor expansion along all directions are eliminated so that the remaining error term becomes $O(h^{2(k+1)})$.

It remains to estimate the CBS constants. We denote them with $\gamma_{H/2k}^{(2k)}$, that is, the constant for piecewise polynomials of degree $2k$ and a macroelement of size H divided into conforming elements of size $h = H/2k$. For piecewise-linear elements, we write $\gamma_H^{(1)}$.

Take then the Schur complements in (A2), which leads to

$$S_h^{(2k)} = \frac{(2k)^2}{(2k)^2-1} S_h^{(2k-1)} - \frac{1}{(2k)^2-1} A_{2h}^{(2k-1)}.$$

Hence, we have

$$\frac{y^T S_h^{(2k)} y}{y^T A_{2h}^{(2k-1)} y} = \frac{(2k)^2}{(2k)^2-1} \frac{y^T S_h^{(2k-1)} y}{y^T A_{2h}^{(2k-1)} y} - \frac{1}{(2k)^2-1}$$

and

$$1-(\gamma_{2k,h})^2 = \min_y \frac{y^T S_h^{(2k)} y}{y^T A_{2h}^{(2k-1)} y} = \frac{(2k)^2}{(2k)^2-1} \min_y \frac{y^T S_h^{(2k-1)} y}{y^T A_{2h}^{(2k-1)} y} - \frac{1}{(2k)^2-1} =$$
$$= \frac{(2k)^2}{(2k)^2-1}\left(1-\left(\gamma_{2k-1,h}\right)^2\right) - \frac{1}{(2k)^2-1} = 1 - \frac{(2k)^2}{(2k)^2-1}\left(\gamma_{2k-1,2h}\right)^2.$$

Therefore,

$$\left(\gamma_{2k-1,2h}\right)^2 = \left(1-\frac{1}{(2k)^2}\right)(\gamma_{2k,h})^2$$

and

$$(\gamma_{1,H})^2 = \prod_{l=1}^{k}\left(1-\frac{1}{(2l)^2}\right)(\gamma_{2k,H})^2 \le \prod_{l=1}^{k}\left(1-\frac{1}{(2l)^2}\right),$$

since $\gamma_{2k,H} < 1$. For instance, for $k = 1$, then we can bound $(\gamma_{1,H})^2 \le \frac{3}{4}$, and for $k = 2$, then $(\gamma_{2,H})^2 \le \frac{45}{64}$, etc.

This is a slight improvement for $k = 1, 2, 3, \ldots$ of the estimate $\gamma_H^{(1)} \le 1-\frac{1}{(2k)^2}$, $H = 2kh$, derived in the work of Axelsson and Blaheta.[19] For instance, if $k = 2$, then our estimate $\gamma_{1,H/2}^2 \le \frac{45}{64}$, whereas the previous estimate is

$$\gamma_{1,H/2}^2 \le 1-\frac{1}{16} = \frac{15}{16}.$$

These bounds hold uniformly with respect to the mesh parameter and coefficients in the differential equation operator.

From the above, it is seen that the coarse-mesh stabilization gives a condition number $1/\prod_{l=1}^{k}(1-\frac{1}{(2l)^2})$, where $k = H/h$.

On the other hand, the coarse–fine-mesh stabilization, where all interface node points are included in the stabilization matrix, corresponds to $k = 1$, that is, $(1+\gamma_h)/(1-\gamma_h) \equiv \frac{1+1/4}{1-1/4} = 5/3$, a substantial improvement. This holds uniformly with respect to coefficient jumps and explains the significant improvements in the number of outer iterations, as we have seen in the numerical tests. The use of a coarse–fine matrix, with only some of the intermediate interface node points between the vertex points, will give a condition number in between the above numbers.

For the alternating Schwarz method without any stabilization, it is readily seen that the condition number corresponds to the condition number for a one-dimensional difference matrix with step size H, that is, the condition number becomes $O(H^{-2})$.

A problem with two-level methods, which are based on a coarse-mesh preconditioner and a large number of interior mesh points, is that the matrix A_{11} can become ill-conditioned for irregular elements; hence, the solution of the corresponding linear systems needs many operations. Therefore, it can be more efficient to use an MIC method instead. An alternative method could be an algebraic multigrid method, which can even be combined with the MIC approach. For references to the algebraic multigrid method, see, for example, the work of Vassilevski[12] or Napov and Notay.[28]