



Two-level mortar domain decomposition preconditioners for heterogeneous elliptic problems

Todd Arbogast^{a,b,*}, Hailong Xiao^c

^a University of Texas at Austin, Institute for Computational Engineering and Sciences, 201 EAST 24th St., Stop C0200, Austin, TX 78712-1229, USA

^b Mathematics Department, RLM 8.100, 2515 Speedway, Stop C1200, Austin, TX 78712-1202, USA

^c Institute for Computational Engineering and Sciences, 201 EAST 24th St., Stop C0200, Austin, TX 78712-1229, USA

Available online 7 November 2014

Highlights

- We use nonoverlapping domain decomposition for mixed method approximations.
- We propose a two-level preconditioner based on the interfaces between subdomains.
- The coarse preconditioner uses the multiscale mortar domain decomposition method.
- Prolongation is defined uniquely to preserve projection onto normal velocities.
- Numerical tests of highly heterogeneous porous media show efficiency and robustness.

Abstract

We consider a second order elliptic problem with a heterogeneous coefficient, which models, for example, single phase flow through a porous medium. We write this problem in mixed form and approximate it for parallel computation using the multiscale mortar domain decomposition mixed finite element method, which gives rise to a saddle point linear system. We use a relatively fine mortar space, which allows us to enforce continuity of the normal velocity flux, or nearly so in the case of nonmatching meshes. To solve the Schur complement linear system for the mortar unknowns, we propose a two-level preconditioner based on the interfaces between subdomains. The coarse preconditioner also uses the multiscale mortar domain decomposition method, but with instead a very coarse mortar space. We show that the prolongation operator of the coarse mortar to the fine is defined uniquely by the condition that the L^2 -projection of a coarse mortar agrees with its projection onto the space of normal velocity fluxes, i.e., no energy is introduced when changing mortar scales. The local smoothing preconditioner is based on block Jacobi, using blocks defined by the interfaces. We use restrictive smoothing domains that are smaller normal to the interfaces, and overlapping in the directions tangential to the interfaces. In the simplest case, the condition number of the preconditioned interface operator is bounded by a multiple of $(\log(1 + H/h))^2$. We show several numerical examples involving strongly heterogeneous porous media to demonstrate the efficiency and robustness of the preconditioner. We see that it is often desirable, and

* Corresponding author at: University of Texas at Austin, Institute for Computational Engineering and Sciences, 201 EAST 24th St., Stop C0200, Austin, TX 78712-1229, USA.

E-mail addresses: arbogast@ices.utexas.edu (T. Arbogast), xiaohl@utexas.edu (H. Xiao).

sometimes necessary, to use a piecewise linear or higher order coarse mortar space to achieve good convergence for heterogeneous problems.

© 2014 Elsevier B.V. All rights reserved.

MSC: 65F08; 76S99

Keywords: Two-level preconditioner; Multiscale mortar mixed method; Heterogeneous porous media; Weak continuity; Homogenization; Two-scale

1. Introduction

On a domain $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3 , we consider the second order elliptic problem

$$\mathbf{u} = -a\nabla p \quad \text{in } \Omega, \quad (1)$$

$$\nabla \cdot \mathbf{u} = f \quad \text{in } \Omega, \quad (2)$$

$$\mathbf{u} \cdot \nu = 0 \quad \text{on } \partial\Omega, \quad (3)$$

wherein ν is the outer unit normal vector to the domain, which is written in mixed form, i.e., as a system of two first order equations plus the boundary condition. The equation arises from minimizing the functional $F(\mathbf{u}, p) = \frac{1}{2}E(\mathbf{u}) + \int_{\Omega} (\nabla \cdot \mathbf{u} - f) p \, dx$, where $E(\mathbf{u}) = \int_{\Omega} a^{-1} |\mathbf{u}|^2 \, dx$ is the energy of the system and p enforces the divergence constraint. We target applications involving flow in porous media [1,2], in which case p is the fluid pressure, \mathbf{u} is the (Darcy) velocity, and the coefficient a is the permeability. The permeability is often highly anisotropic and heterogeneous, varying by many orders of magnitude from point to point. In fact, often the permeability has narrow channels within which the flow is concentrated. These channels are high in permeability and correlated for great distances in some directions but not in others (see, e.g., [3]).

Understanding and predicting fluid flow processes is critical in many subsurface applications, such as CO₂ sequestration, nuclear waste storage, and oil and natural gas production. Furthermore, the flow problem is one of the most time-consuming parts of these simulations. With the development of reservoir characterization methods and geostatistical modeling techniques, the description of reservoir properties can be detailed at multiple scales, from core scales (centimeters) to geological scales (kilometers). A typical reservoir or aquifer is extremely large, and so the geocellular model may have billions of mesh elements. Subsurface processes often last hundreds of years, as in the case of CO₂ migration, or even millions of years for nuclear contaminants. Therefore, we can only simulate these processes using massively parallel supercomputers.

One way of tackling this problem is to reduce its size through upscaling or multiscale techniques [4]. However, the accuracy of the upscaled solution can deteriorate with increasing channel correlation length. Moreover, the flow solution is often coupled to a transport problem, which can magnify errors associated to the flow (see Section 2.2). Our goal is to solve the system on a fine-scale and use multiscale ideas to design effective and robust two-level preconditioners that are suitable for parallel computing when combined with a Krylov accelerator. It is not a new idea to use multiscale ideas to design preconditioners (see, e.g., [5–7]), multigrid methods (e.g., [8,9]), and other iterative procedures (e.g., [10]).

Our approach is to use domain decomposition methods to increase parallelism. In pioneering work, Glowinski and Wheeler [11] defined a nonoverlapping domain decomposition approach to solve the mixed system (1)–(3). We base our work on this method, as modified later to incorporate a general mortar space [12], which became the *multiscale mortar mixed method* [13,14]. This method fully resolves the problem within the subdomains and glues them together with a mortar finite element space. A multiscale solution is obtained when the mortar uses only a few degrees of freedom per interface between subdomains. We use a relatively fine scale mortar space to obtain a fine-scale solution, and a coarse mortar space to define the coarse level preconditioner. The key is to define the extension operator R_0^T from the coarse to the fine mortar space. In fact, we will show that it is uniquely defined by the energy minimizing condition (16) that the L^2 -projection of a coarse mortar agree with the projection of its extension, where the projection is onto the space of traces of the normal component of the velocity on the interfaces between subdomains.

The local, smoothing part of the two-level preconditioner is based on the block Jacobi method, where the block is determined by the unknowns on each interface between subdomains. We advocate using a nonsymmetric, overlapping smoother [15–22].

There is a vast literature on the subject of preconditioners for elliptic problems [22,23]. We mention only a few closely related works that apply to mixed or saddle point systems. The balancing domain decomposition (BDD) [24,25] and BDD with constrains (BDDC) [26] approaches are very similar. These two base their coarse and local preconditioners on single subdomain problems rather than on the interfaces between adjacent subdomains. The method in [27] shares many similarities in terms using domain decomposition and multiscale preconditioners. Solving the interface system using multigrid has been proposed in [28,29]. In the simplest case, our preconditioner has a bounded condition number; the proof (not given here) is related to well-developed theory for similar methods [30,18,24,31,19,22].

Numerical examples show that the convergence performance of our preconditioners is not very sensitive to the ratio of the highest to smallest permeability in a high contrast medium (at least when appearing in a checkerboard arrangement). Problems involving heterogeneous porous media, such as the permeability fields from the SPE10 benchmark problem [3], show that the preconditioned system has a low condition number and eigenvalues clustered around 1, and that our preconditioner is relatively efficient and robust for these types of problems.

In the sequel, Section 2 describes the general mortar domain decomposition methods used and shows why a fine-scale solution is desired for the coupled flow-transport system of two-phase flow in porous media. Section 3 introduces the fine and coarse multiscale mortar domain decomposition methods. The coarse and local preconditioners are defined in Sections 4–5, and these are combined into two-level preconditioners in Section 6. The bound on the condition number is given in Section 7. Numerical results are presented in Section 8, and a summary and conclusions are given in the last section.

2. Mortar domain decomposition

Let the domain Ω be decomposed into n nonoverlapping subdomains Ω_i , $i = 1, 2, \dots, n$. Define the interface $\Gamma_{ij} = \text{interior}(\bar{\Omega}_i \cap \bar{\Omega}_j)$, $1 \leq i, j \leq n$ and the union of the internal interfaces $\Gamma = \bigcup_{1 \leq i, j \leq n} \Gamma_{i,j}$.

Let $\mathcal{T}_{h,i}$ be a conforming, quasi-uniform, finite element partition of Ω_i , with h_i denoting the maximum element diameter of the partition $\mathcal{T}_{h,i}$ and $h = \max_i h_i$. Then $\mathcal{T}_h = \bigcup_{i=1}^n \mathcal{T}_{h,i}$ is the finite element partition over the entire domain Ω . Let $\mathbf{V}_{h,i} \times W_{h,i}$ denote any of the usual inf–sup stable mixed finite element spaces [32,33], e.g., the Raviart–Thomas spaces [34], that enforce the outer boundary condition (3), and let $\mathbf{V}_h = \bigoplus_{i=1}^n \mathbf{V}_{h,i}$ and $W_h = \bigoplus_{i=1}^n W_{h,i}/\mathbb{R}$.

Denote by $\mathcal{T}_{H,ij}$ a quasi-uniform finite element partition of Γ_{ij} , with maximal diameter H_{ij} and $H = \max_{1 \leq i, j \leq n} H_{ij}$. Let $M_{H,ij} \subset L^2(\Gamma_{ij})$ be the local mortar finite element space and $M_H = \bigoplus_{i \neq j} M_{H,ij}$ be the entire coarse mortar space. This space can be continuous or discontinuous polynomials [13,14] or a more general finite element space [35,36]. For the most part, $\mathcal{T}_{H,ij}$ will be either a single element or the coarser of the traces of the meshes $\mathcal{T}_{h,i}$ and $\mathcal{T}_{h,j}$ onto Γ_{ij} .

Denote the standard inner-product on $L^2(\omega)$ by $(\cdot, \cdot)_\omega$ when $\omega \subset \Omega$ and by $\langle \cdot, \cdot \rangle_\omega$ when $\omega \subset \partial\Omega \cup \Gamma$. The discrete variational form of (1)–(3) is formulated as [13,14]: Find $\mathbf{u}_h \in \mathbf{V}_h$, $p_h \in W_h$, and $\lambda_H \in M_H$ such that for $1 \leq i \leq n$,

$$(a^{-1}\mathbf{u}_h, \mathbf{v})_{\Omega_i} - (p_h, \nabla \cdot \mathbf{v})_{\Omega_i} + \langle \lambda_H, \mathbf{v} \cdot \nu_i \rangle_{\partial\Omega_i} = 0 \quad \forall \mathbf{v} \in \mathbf{V}_{h,i}, \tag{4}$$

$$(\nabla \cdot \mathbf{u}_h, w)_{\Omega_i} = (f, w)_{\Omega_i} \quad \forall w \in W_{h,i}, \tag{5}$$

$$\sum_{i=1}^n \langle \mathbf{u}_h \cdot \nu_i, \mu \rangle_{\partial\Omega_i} = 0 \quad \forall \mu \in M_H. \tag{6}$$

Eqs. (4)–(5) represent (1)–(2) locally for a consistent pressure λ_H on Γ , and (6) enforces (possibly only weakly) continuity of the normal flux $\mathbf{u}_h \cdot \nu$.

We remark that the finite element partitions on Ω_i and Ω_j are allowed to be non-matching across some Γ_{ij} (or to not match $\mathcal{T}_{H,ij}$). This is the *nonmatching mesh* case, and the method gives a nonconforming approximation of the true solution. A unique discrete solution exists provided only that a technical condition is met [14], which basically says that we do not over-resolve each interface Γ_{ij} . It is given in (10).

2.1. The linear system

Let bases be given for the finite element spaces,

$$\mathbf{V}_h = \text{span}\{\mathbf{v}_k\}, \quad W_h = \text{span}\{w_k\}, \quad M_H = \text{span}\{\mu_k\}_{k=1}^{n_f},$$

and define the matrices

$$A_{k\ell} = (a^{-1}\mathbf{v}_\ell, \mathbf{v}_k)_\Omega, \quad B_{k\ell} = -(w_\ell, \nabla \cdot \mathbf{v}_k)_\Omega, \quad L_{k\ell} = \sum_{i=1}^n \langle \mu_\ell, \mathbf{v}_k \cdot \mathbf{v}_i \rangle_{\partial\Omega_i}.$$

Then the linear system representing (4)–(6) is the saddle point system

$$\begin{bmatrix} A & B & L \\ B^T & 0 & 0 \\ L^T & 0 & 0 \end{bmatrix} \begin{pmatrix} \vec{\mathbf{u}} \\ \vec{p} \\ \vec{\lambda} \end{pmatrix} = \begin{pmatrix} 0 \\ -\vec{f} \\ 0 \end{pmatrix}, \quad (7)$$

where we use the convention that for each unknown, the same expression overset by an arrow represents the vector of weights in the basis expansion (so, e.g., $\lambda_H(x) = \sum_{k=1}^{n_f} \tilde{\lambda}_k \mu_k(x)$).

Removing the local degrees of freedom $\vec{\mathbf{u}}$ and \vec{p} , we have the Schur complement system

$$L^T C L \vec{\lambda} = S \vec{\lambda} = \vec{b}, \quad (8)$$

where $C = A^{-1}B(B^T A^{-1}B)^{-1}B^T A^{-1} + A^{-1}$ and $\vec{b} = L^T A^{-1}B(B^T A^{-1}B)^{-1}\vec{f}$. Application of S to $\vec{\lambda}$ is achieved simply by solving (4)–(5) for $\vec{\mathbf{u}}$ and \vec{p} given the boundary condition represented by $\vec{\lambda}$, and then computing the jump (difference) in the normal flux from (6), i.e., computing $\vec{b} = L^T \vec{\mathbf{u}}$. Generally we solve (8) using a Krylov method, such as the Preconditioned Conjugate Gradient (PCG) [37,38] or the Generalized Minimal Residual (GMRES) algorithm [39,38].

2.2. The need for flux continuity

If the mortar space is small, we can solve (8) relatively easily. Accuracy can be maintained by using multi-scale finite element techniques (see, e.g., [14,40,41,35,36]). These techniques require oversampling or special homogenization-based mortar spaces, which lead to discontinuities in the flux. That is, (6) imposes continuity of the flux only very weakly when the mortar space is small. Nevertheless, the method gives a quite accurate velocity in terms of the L^2 -norm.

Often the velocity field is used to transport some substance. Generally a transport problem is advection dominated, i.e., nearly hyperbolic. We give an example representing two-phase flow in a porous medium [1,2]. The equations consist of an elliptic flow equation of the form (1)–(3) and an advection–diffusion equation. The former equation is solved implicitly using the mortar domain decomposition scheme described here. The latter equation is solved using operator splitting of the advection and diffusion. The hyperbolic advection is further operator split into locally one-dimensional problems and solved explicitly using a formally fifth order accurate Eulerian–Lagrangian WENO scheme [42]. The parabolic diffusion is solved implicitly using cell-centered finite differences [43].

The permeability coefficient a was geostatistically generated on a uniform 50×50 grid. It is moderately heterogeneous, mildly correlated, locally isotropic, and varies on a log scale by about five orders of magnitude (see Fig. 1). The square domain was decomposed into a 5×5 array of subdomains, each with a 10×10 subgrid. An injection well is placed at the bottom left element and a production well is at the top right element.

As can be seen in Fig. 1 by comparing to the fine scale reference saturation, the reduced degree of freedom mortar methods are completely unsatisfactory. The error in the velocity itself is actually very small, especially for the homogenization-based mortar [35]. However, error accumulates over time in the transport problem due to slight discontinuities in the velocity field at the subdomain interfaces (where the velocity is only weakly continuous). This suggests that one would like to compute the fine scale solution to the flow equations (1)–(3), or at least a nearly continuous velocity field, when solving coupled, nonlinear, time dependent problems in a heterogeneous medium at least when using an IMPES solution strategy (i.e., implicit methods for flow and explicit methods for advection).

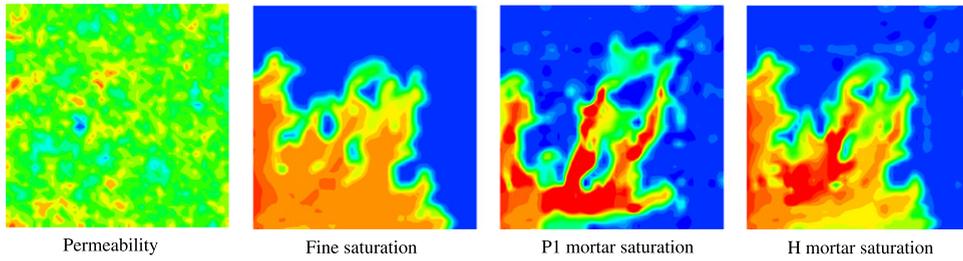


Fig. 1. The permeability (left) and three saturations at day 100, using a fine mortar, a discontinuous linear mortar along each subdomain interface, and a homogenization-based mortar, respectively, for the flow velocity.

A scheme was developed for the multiscale finite volume method [44–46] that included a post-processing technique to impose normal continuity of the flux. For multiscale mixed methods, as we use here, a general postprocessing technique was developed by Sun and Wheeler [47]. However, the algorithm they propose corrects the flux by solving an isotropic diffusion problem. This is unlikely to work well in the case of the highly heterogeneous and highly nonlinear, coupled problem considered here, since we would expect to generate unphysical flows for transport. We therefore concentrate on the case of a full (or nearly full) mortar space, so that a continuous (or nearly continuous) normal flux is generated directly. To solve this larger linear system (8), we develop a suitable two-level preconditioner.

3. Two-level domain decomposition preconditioners

A two-level preconditioner $M^{-1} \approx S^{-1}$ has two components, a global coarse-scale preconditioner M_0^{-1} and a local fine-scale smoothing preconditioner M_{loc}^{-1} , which will be combined later in Section 6. Effective coarse preconditioners, such as the ones in [48,49,40], are not trivial to construct. We propose to construct the coarse preconditioner simply by solving (8) using a coarse mortar space that has only a few degrees of freedom per interface Γ_{ij} . Nevertheless, the numerical results will show that this simple procedure works well for problems arising in porous medium applications.

Henceforth we assume that M_H is a relatively fine discrete mortar space. We assume only that it is not too fine so that (10) holds. Let $\tilde{\mathcal{P}} : (L^2(\Gamma))^2 \rightarrow \Phi_h$ be $(L^2)^2$ -projection into the space of normal fluxes

$$\Phi_h = \bigoplus_{i=1}^n \mathbf{V}_{h,i} \cdot \mathbf{v}_i,$$

which is double valued on each interface Γ_{ij} . Further, let $\mathcal{P} : L^2(\Gamma) \rightarrow \Phi_h$ be defined by $\mathcal{P}\lambda = \tilde{\mathcal{P}}(\lambda, \lambda)$, so that

$$\sum_{i=1}^n \langle \mathcal{P}\lambda, \mathbf{v} \cdot \mathbf{v}_i \rangle_{\partial\Omega_i} = \sum_{i=1}^n \langle \lambda, \mathbf{v} \cdot \mathbf{v}_i \rangle_{\partial\Omega_i} \quad \forall \mathbf{v} \in \mathbf{V}_h. \tag{9}$$

Our assumption is that

$$\mathcal{P}\lambda = 0 \implies \lambda = 0; \tag{10}$$

that is, the matrix L has full column rank.

3.1. Some examples of mortar spaces

The fine mortar space M_H may consist of piecewise constant or linear (or higher order polynomial) functions on a relatively fine mesh $\mathcal{T}_{H,ij}$ on each Γ_{ij} . The coarse mortar space $M_{H,0}$ may be defined similarly over a much coarser mesh.

However, there are other possibilities for the mortar spaces. For example, one might simply take a relatively coarse mesh, even a single element over each Γ_{ij} , and compensate by using higher order functions. The simplest example is to use polynomials, such as, for a one-dimensional interface,

$$M_H|_{\Gamma_{ij}} = \text{span}\{1, x, \dots, x^{p_f}\} \quad \text{and} \quad M_{H,0}|_{\Gamma_{ij}} = \text{span}\{1, x, \dots, x^{p_c}\}, \tag{11}$$

where $p_f > p_c \approx 1$. Another choice is to base an expansion of the mortar pressure on cosine series, leading to

$$M_H|_{\Gamma_{ij}} = \text{span}\{1, \cos x, \dots, \cos(p_f x)\} \quad \text{and} \quad M_{H,0}|_{\Gamma_{ij}} = \text{span}\{1, \cos x, \dots, \cos(p_c x)\}, \tag{12}$$

assuming the interface is scaled to $[0, \pi]$. Of course, we can mix and match the types of mortar spaces between fine and coarse as well.

3.2. The coarse mortar problem

Let $M_{H,0}$ denote a coarse-scale mortar space, containing at least piecewise constant functions over each interface Γ_{ij} and being much smaller than M_H . We assume that

$$\mathcal{P}M_{H,0} \subset \mathcal{P}M_H. \tag{13}$$

With this mortar space, we have a similar domain decomposition problem (4)–(6) for $\lambda_{H,0}$ using $M_{H,0}$ in place of M_H . Then the linear system is modified using a basis and matrix

$$M_{H,0} = \text{span}\{\mu_{0,k}\}_{k=1}^{n_c}, \quad L_{0,k\ell} = \sum_{i=1}^n \langle \mu_{0,\ell}, \mathbf{v}_k \cdot \mathbf{v} \rangle_{\partial\Omega_i}$$

to

$$\begin{bmatrix} A & B & L_0 \\ B^T & 0 & 0 \\ L_0^T & 0 & 0 \end{bmatrix} \begin{pmatrix} \vec{\mathbf{u}}_0 \\ \vec{p}_0 \\ \vec{\lambda}_0 \end{pmatrix} = \begin{pmatrix} 0 \\ -\vec{f} \\ 0 \end{pmatrix} \quad \text{and} \quad L_0^T C L_0 \vec{\lambda}_0 = S_0 \vec{\lambda}_0 = \vec{b}_0. \tag{14}$$

In order to design a successful coarse preconditioner, we need some properties. First, we need a small dimensional problem to achieve computational efficiency, since this is the only part in the two-level domain decomposition algorithm that is not naturally parallel. Second, we need a stable decomposition (see, e.g., [30,31,19,22]), which is usually achieved by requiring that the interpolation operator preserves constants [30] and [31, p. 132] and that the coarse space has low energy and provides good approximation properties [19,22]. Our mortar spaces include at least piecewise constants. The preconditioner will have the low energy property for two reasons. First, we solve the subdomain problems (4)–(5), minimizing the energy locally within each subdomain. Second, the conditions (13) and later (16) ensure that no energy is introduced on the interfaces Γ when changing scales from coarse to fine.

It is interesting that the error analysis of domain decomposition mixed methods requires that at least piecewise linear functions be used for the mortar spaces to achieve accuracy [13]. This is one of the conclusions of the numerical result below (Fig. 2), i.e., that such a coarse mortar provides superior results to simply using piecewise constants.

3.3. Some general remarks on two-level preconditioners

It is perhaps fairly well known that when using a two-level preconditioner, as long as coarse-scale information is effectively transmitted, the local preconditioner will smooth out the fine-scale error. That is, when solving problems with relatively homogeneous permeability, the convergence behavior using a more accurate coarse preconditioner is often only marginally better than using a more modest one [7].

The simplest coarse solver, however, is not likely to work for a highly heterogeneous problem. Consider the strongly heterogeneous problem in Fig. 2, which has a 3×11 domain decomposition and 20×20 subdomains. We depict the value or error in the magnitude of the velocity, i.e., the speed, for several cases. On the far left is the converged fine-scale reference solution, which clearly shows a complex channelized flow field. Using only a local block-Jacobi preconditioner (blocked by the subdomains), the error after ten iterations is large and not evenly distributed. It has features related to the channeling of the flow. If we add a second, coarse level preconditioning using a coarse mortar space consisting of a single constant over each interface Γ_{ij} , considerable error remains along the channels, although the maximum error goes down. However, when we use a linear instead of a constant coarse mortar space, the error reduces greatly and it is evenly distributed except at the corners of the subdomains. That is, channel-like flow cannot be resolved by a constant coarse mortar space.

These results motivate us to design higher dimensional coarse mortar spaces in the next section. Moreover, the error left at the corners motivates us to design restricted overlapping local Block-Jacobi preconditioners in Section 5.

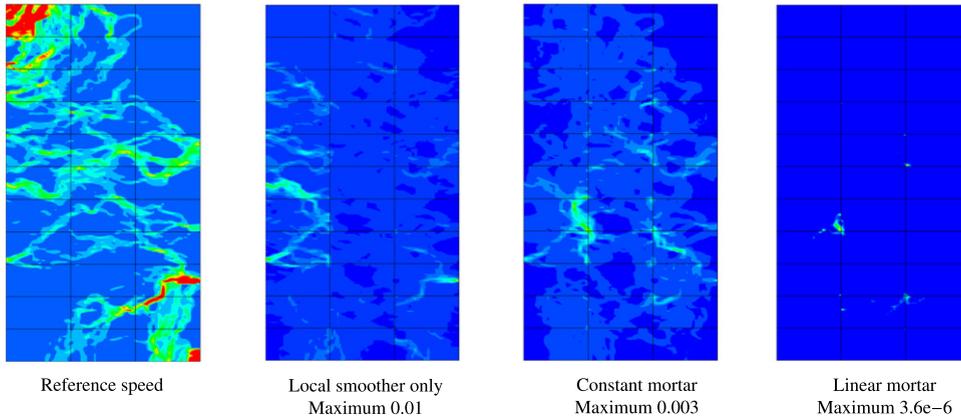


Fig. 2. For a heterogeneous problem, the reference speed and the error in the speed after 10 iterations preconditioned only with the local block Jacobi preconditioner, and with that and the constant and linear mortar space.

4. Global coarse preconditioners

The coarse preconditioner is defined to be

$$M_0^{-1} = R_0^T S_0^{-1} R_0 \tag{15}$$

in terms of an interpolation or extension matrix R_0^T that we need to define. It is the transpose of the restriction matrix $R_0 : \mathbb{R}^{n_f} \rightarrow \mathbb{R}^{n_c}$. Since the Schur complement represents the jump in the normal flux, we need to preserve the jump during interpolation to avoid introducing additional energy into the system. In terms of functions, R_0 is the operator $\mathcal{R}_0 : M_H \rightarrow M_{H,0}$, and because of (13), we can require that

$$\mathcal{P}\lambda_{H,0} = \mathcal{P}\mathcal{R}_0^T\lambda_{H,0} \quad \forall \lambda_{H,0} \in M_{H,0}. \tag{16}$$

It is easy to show that if

$$D_{k\ell} = \sum_{i=1}^n \langle \mathbf{v}_\ell \cdot \mathbf{v}_i, \mathbf{v}_k \cdot \mathbf{v}_i \rangle_{\partial\Omega_i},$$

then the matrices associated to \mathcal{P} restricted to M_H and $M_{H,0}$ are

$$P = D^{-1}L \quad \text{and} \quad P_0 = D^{-1}L_0,$$

respectively, and so the requirement (16) is $P_0\vec{\lambda}_0 = PR_0^T\vec{\lambda}_0$, or more simply

$$L_0 = LR_0^T \iff R_0L^T = L_0^T. \tag{17}$$

We claim that

$$R_0L^T = L_0^T \iff R_0L^TL = L_0^TL.$$

The forward implication is trivial. For the converse, let $n_\phi = \dim \Phi_h$. In terms of column and null spaces, we have that

$$\mathbb{R}^{n_\phi} = C(L) \oplus N(L^T) = C(L_0) \oplus N(L_0^T).$$

Assumption (13) says that $C(D^{-1}L_0) \subset C(D^{-1}L)$, and so also $C(L_0) \subset C(L)$, which leads us to conclude that

$$N(L^T) \subset N(L_0^T).$$

For $\vec{\phi} \in \mathbb{R}^{n_\phi}$, we decompose as $\vec{\phi} = L\vec{\alpha} + \vec{\beta}$, where $\vec{\beta} \in N(L^T) \subset N(L_0^T)$. Then

$$R_0L^T\vec{\phi} = R_0L^T(L\vec{\alpha} + \vec{\beta}) = R_0L^TL\vec{\alpha} = L_0^TL\vec{\alpha} = L_0^T\vec{\phi} - L_0^T\vec{\beta} = L_0^T\vec{\phi},$$

and the converse is proven. Finally, assumption (10) implies that $L^T L$ is invertible, and so the requirement (16) holds if, and only if, we make the definition

$$R_0 = L_0^T L (L^T L)^{-1}. \tag{18}$$

4.1. The matching mesh case

When the traces of the meshes on Ω_i and Ω_j agree on Γ_{ij} for all i and j , the double valued space Φ_h can be interpreted as single valued. In that case, let L^{ij} denote the matrix L as assembled from either side (both will agree). We then have a simpler expression for R_0 restricted to degrees of freedom associated to Γ_{ij} , to wit

$$R_0|_{\Gamma_{ij}} = L_0^{ij,T} L^{ij} (L^{ij,T} L^{ij})^{-1}. \tag{19}$$

In the matching mesh case, it is possible to take on each Γ_{ij} the full mortar space of Lagrange multipliers that appear in the hybrid form of the mixed method [50]. This was done by Glowinski and Wheeler [11] to obtain the full fine-scale solution of the discrete form of the problem (1)–(3), i.e., (4)–(6) using a single subdomain. If we take this space, which is

$$M_H|_{\Gamma_{ij}} = \mathbf{V}_{h,i} \cdot \mathbf{v}_i|_{\Gamma_{ij}} = \mathbf{V}_{h,j} \cdot \mathbf{v}_i|_{\Gamma_{ij}} = \Lambda_{h,ij}, \tag{20}$$

then L^{ij} is invertible (it is the identity if the correct basis is chosen for M_H) and

$$R_0|_{\Gamma_{ij}} = L_0^{ij,T} L^{ij,-T}. \tag{21}$$

We can now simplify the action of the matrix $M_0^{-1} = R_0^T S_0^{-1} R_0$ on a residual vector $\vec{\lambda}_{\text{old}}$. Let $\vec{\mathbf{u}}_{\text{old}} = L^{-T} \vec{\lambda}_{\text{old}}$, which corresponds to the function $\mathbf{u}_{h,\text{old}} \in \mathbf{V}_h$. Application of the coarse preconditioner M_0^{-1} to $\vec{\lambda}_{\text{old}}$ is equivalent to solving the variational problem: Given $\mathbf{u}_{h,\text{old}}$, find $\lambda_{H,0}$ (and \mathbf{u}_h and p_h) such that for $1 \leq i \leq n$,

$$(a^{-1} \mathbf{u}_h, \mathbf{v})_{\Omega_i} - (p_h, \nabla \cdot \mathbf{v})_{\Omega_i} + \langle \lambda_{H,0}, \mathbf{v} \cdot \mathbf{v}_i \rangle_{\partial \Omega_i} = 0 \quad \forall \mathbf{v} \in \mathbf{V}_{h,i}, \tag{22}$$

$$(\nabla \cdot \mathbf{u}_h, w)_{\Omega_i} = 0 \quad \forall w \in W_{h,i}, \tag{23}$$

$$\sum_{i=1}^n \langle \mathbf{u}_h \cdot \mathbf{v}_i, \mu_0 \rangle_{\partial \Omega_i} = \sum_{i=1}^n \langle \mathbf{u}_{h,\text{old}} \cdot \mathbf{v}_i, \mu_0 \rangle_{\partial \Omega_i} \quad \forall \mu_0 \in M_{H,0}, \tag{24}$$

and then setting $M_0^{-1} \vec{\lambda}_{\text{old}} = R_0^T \vec{\lambda}_0$ (where $\vec{\lambda}_0$ corresponds to $\lambda_{H,0}$). This observation allows us to save one application of L^{-T} and L_0^T compared to a direct calculation of $M_0^{-1} \vec{\lambda}_{\text{old}}$, when $\vec{\mathbf{u}}_{\text{old}}$ is stored and available for computation.

4.2. The nonmatching mesh case

In the nonmatching mesh case, we normally prefer to make a stronger assumption than (13), which is that

$$M_{H,0} \subset M_H. \tag{25}$$

In this case, it is trivial to realize that R_0^T is simply the change of basis inclusion matrix, since R_0 is uniquely defined by (16).

5. Local smoothing preconditioners

For the local preconditioner M_{loc}^{-1} , we use a type of block Jacobi (BJ) smoother with each block associated with an interface between two subdomains. Let $\mathcal{R}_{ij} : M_H \rightarrow M_H|_{\Gamma_{ij}}$ denote the restriction operator from the interface Γ to Γ_{ij} . The corresponding matrix is R_{ij} .

For each interface Γ_{ij} , we will smooth over a domain $\tilde{\Omega}_{ij} \supset \Gamma_{ij}$ which is a union of elements from the fine partition \mathcal{T}_h . Let $\mathcal{E}_{ij} : M_H \rightarrow M_H|_{\Gamma \cap \tilde{\Omega}_{ij}}$ denote the restriction operator from the interface Γ to $\tilde{\Omega}_{ij} \cap \Gamma$. To simplify

the notation and the ideas, we assume that degrees of freedom defining M_H over the mesh \mathcal{T}_H can be associated with nodal points contained within $\tilde{\Omega}_{ij}$. This may not hold in some cases when the meshes do not match, so we will need to choose carefully our domains and spaces. For example, with $\Omega_{ij} = \Omega_i \cup \Omega_j \cup \Gamma_{ij}$, we might choose $\tilde{\Omega}_{ij} = \Omega_{ij}$, and then there is no difficulty whatsoever. Under this assumption, we have the discrete matrix E_{ij} corresponding to the operator \mathcal{E}_{ij} . The local preconditioner is

$$M_{\text{loc}}^{-1} = \sum_{i < j} R_{ij}^T (E_{ij} S E_{ij}^T)^{-1} E_{ij}. \tag{26}$$

If $E_{ij} = R_{ij}$, then M_{loc}^{-1} is symmetric and PCG should be used as the outer accelerator (provided that a symmetric two-level preconditioner is chosen in the next section). Otherwise, M_{loc}^{-1} is nonsymmetric and an algorithm such as GMRES is required for the outer accelerator.

Application of this matrix is equivalent to computing contributions from each block. For block Γ_{ij} , it consists in solving a modification of the original variational problem (4)–(6), but posed on the smaller domain $\tilde{\Omega}_{ij}$. Let

$$\tilde{\mathbf{V}}_{h,ij} = \mathbf{V}_h|_{\tilde{\Omega}_{ij}} \cap \{\mathbf{v} : \mathbf{v} \cdot \nu = 0 \text{ on } \partial\tilde{\Omega}_{ij}\}, \quad \tilde{W}_{h,ij} = W_h|_{\tilde{\Omega}_{ij}}, \quad \tilde{M}_{H,ij} = M_H|_{\tilde{\Omega}_{ij} \cap \Gamma}.$$

To apply M_{loc}^{-1} to a mortar vector \vec{r} , we find $\mathbf{u}_h \in \tilde{\mathbf{V}}_{h,ij}$, $p_h \in \tilde{W}_{h,ij}$, and $\lambda_{H,ij} \in \tilde{M}_{H,ij}$ such that

$$(a^{-1} \mathbf{u}_h, \mathbf{v})_{\tilde{\Omega}_{ij}} - (p_h, \nabla \cdot \mathbf{v})_{\tilde{\Omega}_{ij}} + \sum_{k,\ell} \langle \lambda_{H,ij}, \mathbf{v} \cdot \nu_k \rangle_{\Gamma_{k\ell} \cap \tilde{\Omega}_{ij}} = 0 \quad \forall \mathbf{v} \in \tilde{\mathbf{V}}_{h,ij}, \tag{27}$$

$$(\nabla \cdot \mathbf{u}_h, w)_{\tilde{\Omega}_{ij}} = 0 \quad \forall w \in \tilde{W}_{h,ij}, \tag{28}$$

$$\sum_{k,\ell} \langle \mathbf{u}_h \cdot \nu_k, \mu_m \rangle_{\Gamma_{k\ell} \cap \tilde{\Omega}_{ij}} = r_m \quad \forall \mu_m \in \text{the basis for } \tilde{M}_{H,ij}. \tag{29}$$

In this problem, we have imposed zero Dirichlet boundary conditions on $\partial\tilde{\Omega}_{ij} \setminus \partial\Omega$ and zero boundary conditions of the same type as the original problem (3) on $\partial\Omega$. The result is

$$M_{\text{loc}}^{-1} \vec{r} = \vec{\lambda} = \sum_{i < j} R_{ij}^T \tilde{\lambda}_{ij},$$

where $\tilde{\lambda}_{ij}$ corresponds to $\lambda_{H,ij}$.

We can reformulate the local preconditioner as

$$M_{\text{loc}}^{-1} = \sum_{ij} R_{ij}^T \tilde{S}_{ij}^{-1} E_{ij}, \tag{30}$$

where \tilde{S}_{ij}^{-1} is the Schur complement of the local problem (27)–(29); that is, of

$$\begin{bmatrix} \tilde{A} & \tilde{B} & \tilde{L} \\ \tilde{B}^T & 0 & 0 \\ \tilde{L}^T & 0 & 0 \end{bmatrix} \begin{pmatrix} \tilde{\mathbf{u}} \\ \tilde{p} \\ \tilde{\lambda}_{ij} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ E_{ij} \vec{r} \end{pmatrix}, \tag{31}$$

which involves the restriction of A , B , and L to \tilde{A} , \tilde{B} , and \tilde{L} involving only degrees of freedom associated to $\tilde{\Omega}_{ij}$.

The local preconditioner is defined here in terms of each edge, not each subdomain. This is in contrast to the local preconditioners defined in balancing domain decomposition (BDD) [24,25] and BDD with constrains (BDDC) [26]. The one defined here may be more appropriate for strongly heterogeneous problems, since the definition of the local solver based on the subdomains in BDD and BDDC depends on a scaling matrix. For highly heterogeneous problems, the effectiveness of the local preconditioner is affected by the selection of the scaling matrix [51]. Our local solver (30) does not depend on such a scaling operator.

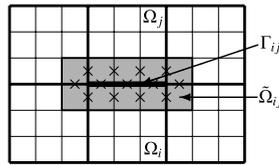


Fig. 3. An example of a domain $\tilde{\Omega}_{ij}$ containing the interface Γ_{ij} between Ω_i and Ω_j for matching 3×3 subdomain meshes. The restrictive overlapping local preconditioner depicted here involves the mortar degrees of freedom, denoted by crosses (\times), that are on Γ_{ij} and immediately adjacent to it.

5.1. The full BJ local preconditioner

The simplest choice is to take $\tilde{\Omega}_{ij} = \Omega_{ij} = \Omega_i \cup \Omega_j \cup \Gamma_{ij}$, and thus $E_{ij} = R_{ij}$. In this case, we have the usual, full BJ preconditioner associated to the interface blocks Γ_{ij} . The preconditioner is

$$M_{\text{loc,BJ}}^{-1} = \sum_{i < j} R_{ij}^T (R_{ij} S R_{ij}^T)^{-1} R_{ij} = \sum_{i < j} R_{ij}^T S_{ij}^{-1} R_{ij}, \quad (32)$$

which is symmetric, where $S_{ij} = \tilde{S}_{ij}$ is the local Schur matrix for degrees of freedom on Ω_{ij} .

5.2. A restrictive local preconditioner

In the application of S_{ij}^{-1} in the BJ preconditioner (32), we need to solve the problem (27)–(29) on Ω_{ij} , which is essentially twice as large as the subdomain Ω_i itself. In $d = 3$ dimensions, if we solve the local smoothing problems by a sparse direct solver, the application of $M_{\text{loc,BJ}}^{-1}$ is approximately four times more expensive than the application of a local preconditioner based on subdomains (as in BDD and BDDC). Furthermore, the number of interfaces are asymptotically $d = 2$ or 3 times more than the number of the subdomains for a rectangular coarse mesh. This is a computational bottleneck in applying the local block Jacobi preconditioner $M_{\text{loc,BJ}}^{-1}$.

In practice, we can greatly improve the computational efficiency by solving the local problems (27)–(29) on a much smaller region $\tilde{\Omega}_{ij} \subset \Omega_{ij}$ containing the edge Γ_{ij} . For example, $\tilde{\Omega}_{ij}$ may consist of Γ_{ij} and one or two layers of elements normal to it on both sides. In this case, again $E_{ij} = R_{ij}$, since we restrict to Γ_{ij} , and the preconditioner, called a *restrictive local preconditioner*, remains symmetric.

Based on the observation that the Green's function for an elliptic problem decays very fast, and that we only use M_{loc}^{-1} as a smoothing preconditioner for the outer accelerator (PCG or GMRES), we should retain efficient convergence behavior. We will show some three dimensional, highly heterogeneous examples in Section 8 to demonstrate that this is indeed the case.

5.3. A restrictive overlapping local preconditioner

Notice that the local smoother (32) or its restricted approximation uses the mortar \vec{r} only on Γ_{ij} and ignores it on $\partial\Omega_{ij}$, using instead a zero Dirichlet boundary condition there. This over-constrains the resulting flux near the corners, and accounts for why the errors in Fig. 2 are large at the corners. We need to account for \vec{r} on interfaces close to Γ_{ij} , as in the example depicted in Fig. 3.

We solve the local problem (27)–(29) on $\tilde{\Omega}_{ij}$ chosen to be an extension outside of Ω_{ij} in the direction tangential to Γ_{ij} and, most likely, restricted in the normal directions as explained in the previous subsection. To do this consistently, we may need to restrict the meshes \mathcal{T}_h and \mathcal{T}_H . For example, we can use a matching mesh \mathcal{T}_h and let \mathcal{T}_H be the trace of \mathcal{T}_h on Γ , as shown in Fig. 3. The resulting preconditioner $M_{\text{loc}}^{-1} = \sum_{i,j} R_{ij}^T \tilde{S}_{ij}^{-1} E_{ij}$ is called a *restrictive overlapping local preconditioner*. It is nonsymmetric, even for a symmetric problem, and we need to use something like GMRES as the outer accelerator.

The idea of using overlapping local preconditioners for nonoverlapping domain decomposition was used in the *Vertex Space Method* [16,17,19] (or Copper Mountain algorithm [15]) for Galerkin approximations. Cowsar [18, Section 2.4] extended the theory to hybrid mixed finite element methods. Nonsymmetric local preconditioners were also developed for overlapping domain decomposition as the *restricted additive Schwarz* (RAS) preconditioner [20,22,21]. The discovery of the RAS method by Cai and Sarkis in [20] is quite interesting. They

found it accidentally by removing the communication routine in the overlapping additive Schwarz/GMRES algorithm. But here we see clearly the need for a nonsymmetric local preconditioner. They showed that the nonsymmetric local preconditioner is more efficient than the symmetric one for several numerical examples. Efsthathiou and Gander [21] proved that, when using an overlapping local preconditioner as a solver, often the nonsymmetric solver will converge but not the symmetric one.

6. Two-level preconditioners

Now that we have defined the local M_{loc}^{-1} and coarse M_0^{-1} preconditioners, we can combine them in different ways to define several symmetric and nonsymmetric two-level preconditioners, to be used within the PCG or GMRES algorithms. The first and simplest is the *additive* preconditioner

$$M_{add}^{-1} = M_0^{-1} + M_{loc}^{-1}, \tag{33}$$

which is symmetric if M_{loc}^{-1} is symmetric. The second is the *hybrid* preconditioner

$$M_{hyb}^{-1} = M_0^{-1} + (I - P_0)M_{loc}^{-1}(I - P_0^T), \tag{34}$$

where $P_0 = M_0^{-1}S$ is the Schwarz projection operator. This preconditioner is symmetric if M_{loc}^{-1} is symmetric, and it is due to Mandel [52]. In practice, we do not need to apply $I - P_0^T$ in each PCG iteration, see [22, Lemma 2.11]. The third and last two-level preconditioner that we consider is the nonsymmetric *multiplicative* preconditioner

$$M_{mul}^{-1} = M_0^{-1} + M_{loc}^{-1} - M_{loc}^{-1}SM_0^{-1}. \tag{35}$$

Since it is nonsymmetric, we cannot use the PCG algorithm as the outside accelerator, so instead we use the right preconditioned GMRES algorithm. This two-level preconditioner is especially useful when we use the efficient nonsymmetric restricted overlapping local preconditioners.

We can modify the definition of M_{mul}^{-1} as

$$\tilde{M}_{mul}^{-1} = M_0^{-1} + M_{loc}^{-1} - M_0^{-1}SM_{loc}^{-1} = (I - P_0)M_{loc}^{-1} + M_0^{-1}. \tag{36}$$

By [53, Theorem 3.1], we have $\sigma(M_{mul}^{-1}S) = \sigma(\tilde{M}_{mul}^{-1}S) = \sigma(M_{hyb}^{-1}S)$, where $\sigma(\cdot)$ is the spectrum of the matrix. If we drop the term $(I - P_0^T)$ in the definition (34) of M_{hyb}^{-1} , we have $\tilde{M}_{mul}^{-1} = M_{hyb}^{-1}$.

Nabben and Vuik [54] defined an effective condition number which depends on the initial guess. They showed that for hybrid preconditioned CG, the S -norm of the error is controlled by the effective condition number of $M_{hyb}^{-1}S$, which is less than the condition number of $M_{add}^{-1}S$, if we use $\tilde{\lambda}^0 = M_0^{-1}\tilde{b}$ as the initial guess. Simply speaking, the hybrid preconditioner generally converges faster than the additive.

If we use nested mortar spaces $M_{H,0} \subset M_H$, as in (11) or (12), then M_H can be viewed as having two scales, a coarse and fine scale. In this case the hybrid two-scale preconditioner with the full BJ smoother has a nice interpretation. When we use the special starting value $M_0^{-1}\tilde{b}$ for the PCG iteration, it is equivalent to using the coarse solver to project out (i.e., exactly solve) the coarse components. PCG then works with the BJ smoother only to solve the Schur complement system representing the fine components of the mortar. For more details and a proof of this statement, see [55, Theorem 4.6.1].

7. A bound on the condition number in a special case

In this section we consider the simplest preconditioner that we can define. We consider only a problem defined using matching grids, and using the lowest order Raviart–Thomas or Brezzi–Douglas–Marini [56] mixed finite elements (or more general elements, see [18, Chapter 4]). The fine mortar space is the normal trace of the velocity space $M_H = \Lambda_h$, and the coarse mortar space is the piecewise constants, where the coarse mortar grid has a single element on each interface Γ_{ij} . We use the additive two-level preconditioner (33) combined with the simple full BJ local smoother $M_{loc,BJ}^{-1}$. We assume that exact coarse and local solvers are used, and that the coarse mesh is shape-regular. The permeability a is bounded and uniformly positive.

Theorem 7.1. *The condition number of the additive two-level preconditioned Schur matrix, i.e., $M_{\text{add}}^{-1}S$, is bounded by a multiple of $(\log(1 + H/h))^2$.*

For the proof, see [55, Chapter 4, Section 5]. The proof uses the abstract theory of Schwarz methods, technical tools for conforming finite element discretizations, and interpolation operators defined from the nonconforming mortar space to the conforming linear finite element space. It is based mainly on the work of Toselli and Widlund [22], Casarin [57], Cowsar, Mandel, and Wheeler [24], and Cowsar [18]. Our condition number bound is the same as for the classic BDD [18,24] and BDDC [26] two-level preconditioners.

8. Numerical examples

All our numerical examples are posed over a rectangular domain in $d = 2$ or $d = 3$ dimensions, and each uses a rectangular array of n subdomains, each of which has a rectangular fine mesh so that the meshes match across the subdomain boundaries. The subdomain problems are approximated using the lowest order Raviart–Thomas spaces RT0 [34] for $\mathbf{V}_{h,i} \times W_{h,i}$, which approximate to first order in h . Unless otherwise noted, the fine mortar space is the trace of the normal velocities $M_H = \Lambda_H$ (see (20)). The coarse mortar grid $\mathcal{T}_{H,0} = \cup_{i < j} \Gamma_{ij}$ uses a single element on each interface.

We construct the coarse Schur matrix S_0 by coloring the subdomain interfaces and applying the operator to each color. Since the matrix S_0 is banded and sparse, to apply its inverse we use a direct solver, either LAPACK [58] in serial or MUMPS [59–61] in parallel.

To apply the local preconditioner M_{loc}^{-1} , we solve the local system (31) so that we need not explicitly calculate the entries of \tilde{S}_{ij} . Again we use a direct solver, either LAPACK, MUMPS, or PARDISO [62–65].

To apply the operators L and L^T in parallel, we use the message passing interface (MPI) [66] with nonblocking send and receive. In our implementations of the mortar mixed finite element method, we do not require that the partition of the coarse grid and the processor grid be the same, i.e., one subdomain per computer core. Instead, we allow the problem to be divided so that multiple subdomains may be solved by a single core. This gives us greater flexibility to adjust the size of the subdomains. As a consequence, the condition number bound and the size of the coarse matrix are independent of the number of cores.

The outer accelerator is PCG if the two-level preconditioner is symmetric, and GMRES otherwise.

We run all our three dimensional numerical examples on supercomputer *Stampede* supported by the Texas Advanced Computing Center (TACC) of the University of Texas at Austin. A Compute node consists of two Xeon Intel 8-Core 64-bit E5-processors (16 cores in all) on a single board, as a symmetric multiprocessing shared memory unit. The core frequency is 2.7 GHz and supports 8 floating-point operations per clock period with a peak performance of 21.6 GFLOPS per core or 346 GFLOPS per node. Each node contains 32 GB of memory (2 GB per core). The memory subsystem has 4 channels, each rated at 1600 MT/s (51.2 GB/s for all four channels in a socket). The processor interconnect runs at 8.0 GT/s between sockets.

8.1. Three-dimensional Poisson tests

The first test is Poisson’s equation on the unit cube $\Omega = [0, 1]^3$ with the exact solution $p(x, y, z) = \sin(\pi x) \sin(\pi y) \sin(\pi z)$. This is a good test example for demonstrating the consistency of the performance of two-level preconditioners with the theory of Section 7. The two-level preconditioner is the hybrid one (34) with the coarse level preconditioner constructed from the piecewise constant coarse mortar space. The full BJ local smoother is used.

For a fixed ratio of H/h , the number of iterations does not increase as H or h changes, as shown in Table 1. As we increase H/h by changing H or h , the number of iterations increases slowly. These numerical results are consistent with our Theorem 7.1, which says that the condition number is bounded by $(\log(1 + H/h))^2$. Actually, the results suggest that the bound might be $\log(1 + H/h)$.

8.2. High contrast checker board

In the next numerical example, we study the sensitivity of the multiplicative two-level preconditioner to a high contrast, isotropic medium. We define our permeability tensor a on a unit cubic domain $\Omega = [0, 1]^3$ with an

Table 1

The number of iterations for H versus H/h in the Poisson three dimensional test, which does not change for fixed ratio H/h and grows slowly as the ratio increases.

		H			
		1/2	1/4	1/8	1/16
H/h	4	7	7	7	7
	8	10	10	10	10
	16	12	13	13	13

alternating, checker board pattern. In the black regions $a = 10^\alpha$, while in the white regions $a = 1$. The source term $f = 1$, and the Dirichlet boundary condition $p = 0$ is imposed on $\partial\Omega$. We decompose the domain Ω into $8 \times 8 \times 8$ subdomains and each subdomain has an $8 \times 8 \times 8$ subgrid. We use a zero initial guess, and the stopping criterion that the residual is reduced by 6 orders of magnitude in the discrete ℓ_2 -norm. The local smoother is the full BJ preconditioner (32).

When the coarse preconditioner M_0^{-1} uses a single constant on each interface Γ_{ij} , the number of iterations is 20 for $\alpha = 0$ (no contrast), and 21 for $\alpha = 6$ and $\alpha = 12$. When the coarse preconditioner uses a single linear on each interface, the number of iterations is 15 for $\alpha = 0$, $\alpha = 6$, and $\alpha = 12$. The number of iterations is almost independent of the contrast 10^α . This is because the transmissibility coefficients defined on the faces of the fine grid elements are almost homogeneous except on the boundary of the domain Ω . That is, the flow pattern does not change much with the contrast α .

8.3. Two-dimensional examples from the SPE10 dataset

In this section, we present some numerical examples to study the performance of the different two-level preconditioners M_{add}^{-1} , M_{mul}^{-1} , and M_{hyb}^{-1} with different coarse preconditioners (constant, linear, cosine, homogenization based, etc. mortar) for homogeneous and strongly heterogeneous permeability coefficients.

In our two-dimensional examples, the fine scale mortar grid depends on which fine scale mortar space we use. When the fine scale mortar space is the trace of the velocity flux space $M_H = \Lambda_h = \mathbf{V}_h \cdot \nu$, the fine scale mortar grid is the trace of the subdomain grid. When the fine scale mortar space is spanned by discontinuous piecewise polynomials or cosine series, the fine scale mortar grid is the coarse scale mortar grid.

The permeability fields are taken from the Tenth Society of Petroleum Engineers Comparative Solution Project (SPE10) [3] benchmark problem model 2. We take the 36th and 85th layers, as shown in Fig. 4. The domain is 1200×2200 [ft²]. The fine scale grid has 60×220 elements. The test is an example of a quarter five-spot pattern of wells, with an injection well in the lower left corner element and a production well in the upper right corner element. We assume no-flow boundary conditions. These permeability fields give rise to strong, long-range channels and produces extremely complex velocity fields.

8.3.1. The spectrum of the preconditioned matrix

We decompose the 60×220 fine grid into a 3×11 coarse grid with a 20×20 subgrid. In this example, on each interface we use the first, the first two, or the first three functions of the cosine series on a reference element $[0, \pi]$ to define the coarse mortar space, and all 20 modes to define the fine mortar space (see (12)). The hybrid two-level preconditioner M_{hyb}^{-1} (34) is used with the full BJ local smoother. To reduce the size of the fine scale matrix, on each interface we divided the fine mortar space M_H into a coarse space of three cosine modes $M_{H,0}$ plus the other 17 modes. We then found the Schur complement for the finer modes.

In Fig. 5, we show on the left the histogram from 0 to 60 of the eigenvalues of the unpreconditioned matrix on a log scale from -17 to -8 . It has a condition number of $k = 1.0 \times 10^6$. The other three histograms, from 0 to about 800, show the eigenvalues of the preconditioned system (not on a log scale) from about 0 to 1.8. The condition numbers are $k = 33.5, 9.8,$ and 7.4 for $M_{H,0}$ consisting of one, two, and three cosine modes, respectively. Furthermore, the spectrum of the preconditioned matrices are clustered about 1, which is advantageous for Krylov algorithms. Similar results are obtained for the 36th layer (not shown), which has a simpler permeability field than the 85th layer. The

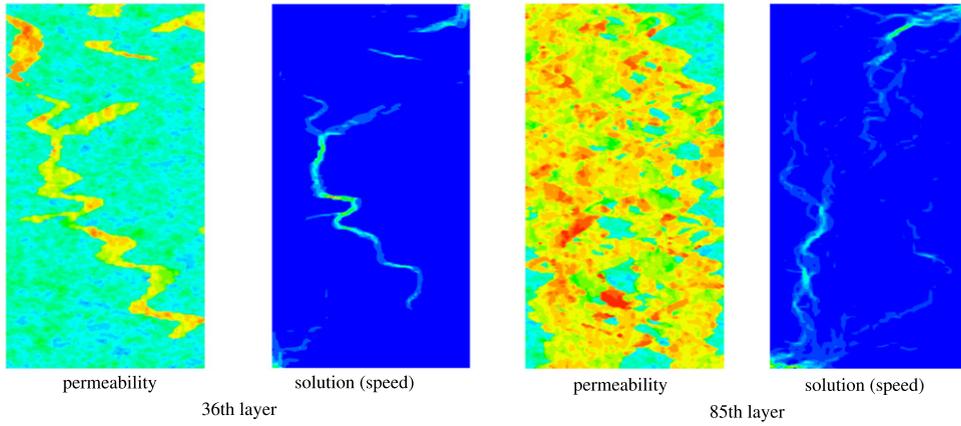


Fig. 4. The logarithm of the permeability field and the magnitude of the velocity solution (speed) for the 36th and 85th layers of the SPE10 benchmark problem.

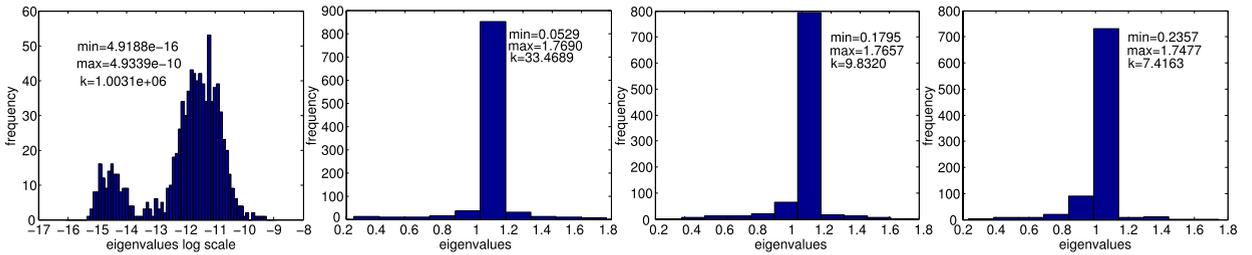


Fig. 5. Histogram of eigenvalues of the SPE10 85th layer, showing the unpreconditioned matrix on a log scale using only 3 degrees of freedom per edge (left) and the preconditioned matrix using a coarse space of 1, 2, and 3 degrees of freedom per edge.

Table 2

Number of iterations of the multiplicative two-level preconditioner using the full BJ local smoother for two SPE10 layers. Results are for 6×22 and 3×11 subdomains (with 10×10 and 20×20 subgrids) using no coarse preconditioner, and coarse mortar spaces of polynomials of degree 0 (const), 1 (lin), and 2 (quad), cosine series of 2 and 3 modes (cos), and the homogenization-based mortar space without (hom) and with oversampling (hom-os). The number of degrees of freedom (DOF) per edge is noted. Shown also are results for the BDD preconditioner.

M_0 coarse space	no	const	lin	quad	cos	cos	hom	hom-os	BDD
# DOF per edge	0	1	2	3	2	3	3	3	–
Layer 36									
6×22 subdomains	>100	33	20	15	19	15	12	11	
3×11 subdomains	80	24	18	15	18	16	13	12	27
Layer 85									
6×22 subdomains	>100	45	22	17	21	17	13	12	
3×11 subdomains	88	34	19	14	20	15	14	12	41

condition number reduces from $k = 3.9 \times 10^4$ to 11.9, 8.5, and 6.4, for $M_{H,0}$ consisting of one, two, and three cosine modes, respectively.

8.3.2. Comparison of different coarse preconditioners

We now compare various choices for the coarse preconditioner. In each test case, we use the multiplicative preconditioner M_{mul}^{-1} (35) and the full local BJ preconditioner. We set the stopping criterion as residual reduction to 10^{-6} of the initial residual.

In Table 2, we compare the number of iterations to reach the convergence criterion using various coarse mortar spaces. We compare results using no coarse preconditioner (i.e., BJ alone), and coarse mortar spaces of polynomials

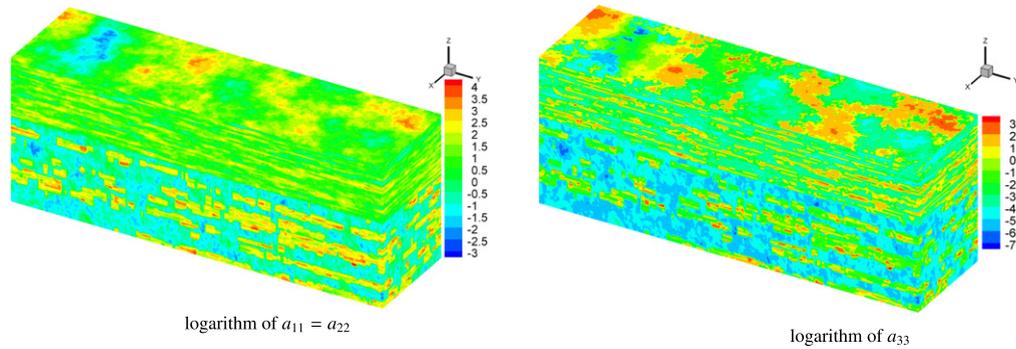


Fig. 6. Logarithm of the diagonal permeability tensor from SPE10 model 2.

(see (11)) of degree 0 (const), 1 (lin), and 2 (quad), cosine series (see (12)) of 2 and 3 modes (cos), and the special homogenization-based mortar space [35,36] using formal linear approximations homogenized over Ω_{ij} (hom) and on an oversampled domain (hom-os). Each space has the given number of degrees of freedom (DOF) per interface edge.

Of course, using only the block Jacobi local preconditioner without the coarse component requires many more iterations to converge, or possibly even diverges. However, a reasonable number of iterations is obtained by adding a piecewise constant coarse preconditioner. Moreover, we see a significant reduction using linears instead of constants, cutting the number of iterations by about a third and a half for the 36th and 85th layers, respectively. After that, adding more degrees of freedom shows a more marginal improvement, even though the coarse preconditioner is becoming more expensive (i.e., the coarse problem becomes larger).

The cosine series mortar spaces perform about as well as the polynomial ones for the same number of degrees of freedom. Generally, the coarse preconditioner defined by the homogenization-based mortar space, with or without the oversampling technique, performs the best. But it also requires the most work to set up the coarse level preconditioner matrix. We note that for the multigrid method, a coarse preconditioner based on homogenization theory has been defined already in [5,6] for Galerkin approximations to elliptic problems with highly oscillatory coefficients.

The more complicated 85th layer takes much more effort to converge with the piecewise constant coarse preconditioner than the 36th layer for the same subdomain partition. But this is not true for the other coarse preconditioners. This suggests that the linear, quadratic, etc., coarse preconditioners are much less sensitive to the heterogeneity of the problem.

There is an interesting and unusual phenomenon, that for a given coarse preconditioner, the number of iterations for the 3×11 ($H/h = 20$) coarse grid partition is less than the 6×22 ($H/h = 10$) partition for both the 36th and 85th layers. There are large errors inside the subdomains for the 6×22 subdomain partition, possibly due to the effect of heterogeneity and the way of setting up the right hand side for applying the local BJ preconditioner.

We also compare the number of iterations with the balancing domain decomposition method (BDD) [18,24,25] using 3×11 subdomains in Table 2. The fine scale mortar space is spanned by piecewise constant functions. The number of iterations of the BDD preconditioner is a little bit more than the multiplicative two-level one with a piecewise constant coarse preconditioner (27 to 24 for the 36th layer and 41 to 34 for the 85th layer). However, the BDD preconditioner is quite a bit cheaper to compute. The coarse space of the BDD method is constructed based on subdomains, not on interfaces, and thus the size of the coarse matrix is about 2 times smaller. Notice also that the local preconditioner in BDD is the Neumann to Dirichlet map, so the local preconditioner $M_{loc,BJ}^{-1}$ in M_{mul}^{-1} is also about twice as large as the local preconditioner in BDD.

8.4. Three-dimensional examples from the SPE10 dataset

In this section, the permeability field is again taken from the SPE10 benchmark problem [3], but we use the full three-dimensional dataset. The fine scale model has $60 \times 220 \times 85$ elements. The dimension of the model is $1200 \times 2200 \times 170$ [ft³]. Fig. 6 shows the logarithm of the x -direction (same for y -direction) and z -direction permeability fields. The top 35 layers form a Tarbert formation and represent a prograding near-shore environment. The bottom 50 layers represent an Upper Ness fluvial formation, with the channels clearly visible in the 85th layer shown in Fig. 4.

Table 3

Number of iterations and CPU time for the 3-D uniform SPE10 isotropic x -permeability using the full block Jacobi local smoother $M_{\text{loc,BJ}}^{-1}$. Shown are results using coarse mortar spaces of polynomials of degree 0 (const), 1 (lin), and 2 (quad) using the initial guess $M_0^{-1}\vec{b}$.

M_0 coarse space	const		lin		quad	
	iter	CPU (s)	iter	CPU (s)	iter	CPU (s)
$12 \times 44 \times 17$ subdomains						
Hybrid preconditioner	45	56.8	15	41.8	8	49.3
$6 \times 22 \times 17$ subdomains						
Additive preconditioner	48	50.5	23	45.8	14	48.0
Hybrid preconditioner	32	50.1	14	45.1	9	47.5
Multiplicative preconditioner	30	50.7	13	45.3	9	47.9
$6 \times 22 \times 5$ subdomains						
Hybrid preconditioner	73	81.6	44	72.2	22	66.6

The largest and smallest value of the permeability is 2×10^5 [md] and 6.65×10^{-8} [md]. This permeability field gives rise to strong, long-range correlated channels and produces an extremely complex velocity field.

In all our three dimensional examples, the additive and multiplicative two-level preconditioners use zero as the initial guess, and the hybrid uses $M_0^{-1}\vec{b}$ as the initial guess, except for the results of Table 3.

8.4.1. Isotropic permeability and uniform grids

For the following examples, we modify the SPE10 dataset to use an isotropic permeability field, i.e., $a_{11} = a_{22} = a_{33}$, where a_{11} is the x -direction permeability. We also set $\Omega = 60 \times 220 \times 85$ such that the fine grid is uniform. There is no source nor sink ($f = 0$). For the boundary conditions, the pressure on the left and right faces in the pictures of Fig. 6 is normalized to 1 and 0, respectively; the rest of the faces have the no-flow condition.

In the first numerical example, we study the effect of the subdomain partition and the choice of two-level preconditioner. We use the full BJ local smoothing preconditioner and a coarse preconditioner with a single polynomial on each interface of degree 0 (const), 1 (lin), or 2 (quad). We decompose the domain Ω into a different number of subdomains, $12 \times 44 \times 17$ with a $5 \times 5 \times 5$ subgrid, $6 \times 22 \times 17$ subdomains with a $10 \times 10 \times 5$ subgrid, and $6 \times 22 \times 5$ subdomains with a $10 \times 10 \times 17$ subgrid. The stopping criteria is that the residual is reduced by five orders of magnitude in the discrete ℓ_2 -norm. For this test only, we use the initial guess $M_0^{-1}\vec{b}$ for all test cases so that we can more fairly compare the two-level preconditioners.

As seen in Table 3, this strongly heterogeneous problem behaves differently from the homogeneous Poisson's problem as we change the size of the subdomains (H/h). Because of the discontinuity of the coefficients along the interfaces and the long range correlated channels, the performance of the two-level preconditioners becomes much less predictable. For the hybrid preconditioner, the best grid partition in terms of number of iterations and CPU time is $6 \times 22 \times 17$ subdomains with subgrid $10 \times 10 \times 5$. We use this partition for the other tests.

Table 3 shows that the additive two-level preconditioner M_{add}^{-1} takes more iterations, but not necessarily more CPU time, since each iteration applies the Schur complement S one time less than the hybrid and multiplicative preconditioners. The number of iterations and CPU time for the preconditioner M_{hyb}^{-1} and M_{mul}^{-1} are almost the same.

8.4.2. Nonsymmetric restricted overlapping local preconditioners

We now apply the restricted (RBJ), nonsymmetric overlapping (OBJ), and nonsymmetric restricted overlapping (ROBJ) local preconditioners discussed in Section 5.2 with the multiplicative two-level preconditioner and zero initial guess. We use a 3×3 matrix $(d_{k\ell})_{3 \times 3}$ to represent the dimensions of the local extended smoothing domains $\tilde{\Omega}_{ij}$ for the local preconditioners. A two dimensional example is shown in Fig. 7. Entry $d_{k\ell}$ gives the extension distance in the ℓ th direction from Γ_{ij} for an interface with normal in the k th direction. If $d_{k\ell} = 0$ for $k \neq \ell$, we have a nonoverlapping local preconditioner. If $d_{kk} < H_k/h_k$, we have a restrictive preconditioner in the k -direction.

For a given coarse preconditioner M_0^{-1} , there is little difference in terms of the number of iterations using the local BJ preconditioners configured with $d_{kk} = 1$ and $d_{k\ell} = 2$ for $k \neq \ell$. Thus, in the following numerical examples, we

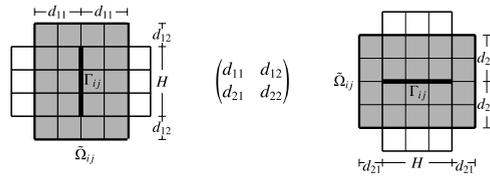


Fig. 7. Two dimensional matrix representation describing the domain $\tilde{\Omega}_{ij}$.

Table 4

Number of iterations and CPU time for SPE10 with isotropic permeability and uniform grid using the multiplicative two-level preconditioner and various coarse and local preconditioners.

M_0 coarse space	const	const	lin	lin	lin
$\tilde{\Omega}_{ij}$	$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 5 \end{pmatrix}$	$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{pmatrix}$
	RBJ	ROBJ	BJ	RBJ	ROBJ
iter	34	25	16	17	12
CPU (s)	24.5	25.7	42.0	18.9	20.7

will set $d_{k\ell} = 0$ or 1 for $k \neq \ell$. The least number of iterations should use the values

$$\begin{pmatrix} H_1/h_1 & 1 & 1 \\ 1 & H_2/h_2 & 1 \\ 1 & 1 & H_3/h_3 \end{pmatrix}.$$

In the following examples, we will give the smallest values of $(d_{k\ell})_{3 \times 3}$ needed to match this case.

The results shown in Table 4 use the isotropic physical parameters, uniform grid, and mesh settings as in Section 8.4.1, using $6 \times 22 \times 17$ subdomains ($10 \times 10 \times 5$ subgrid). Both piecewise constant and linear coarse preconditioners show fewer iterations for the overlapping ROBJ versus RBJ, but not improved CPU time for this problem. The restrictive preconditioners are much better than the full BJ in CPU time, but only marginally worse in the number of iterations (compare also to Table 3, which has a slightly different convergence criterion).

The ROBJ smoother is most effective for the multiplicative two-level preconditioner with a piecewise constant coarse mortar space. It reduces the number of iterations from 34 to 25. It is effective, but less so, for the piecewise linear coarse mortars, because these already capture errors along the vertices and edges of the interfaces much more effectively than the piecewise constant coarse mortars.

In [27], Zhou and Tchelepi had a similar number of iterations counts for this problem. But their CPU time seems faster, since the size of their local preconditioner is much smaller than the ones here. We also used PETSc [67–69] to solve some simpler problems. We notice that usually the PETSc default multigrid solver takes more iterations. However, each iteration is much faster than our implementation, and so the total solver time is 2 to 3 times faster, if the linear solver converged (it does not always converge).

We next use the unmodified, anisotropic permeability field and grid from the SPE10 dataset, i.e., $a_{11} = a_{22} \neq a_{33}$ and $h_1 = 2h_2 = 10h_3 = 20$. Boundary conditions and source terms are the same as before. We use a piecewise linear coarse mortar for these results. The results are given in Table 5. Due to the anisotropic permeability and the non-uniform grid, comparing with the results in Table 4, the number of iterations increases for this problem.

Compared to the full BJ preconditioner (21 iterations, 46.4 s), the RBJ (25 iterations, 23.2 s) and ROBJ (22 iterations, 23.0 s) are much faster and take only a few more iterations. The fewest number of iterations is given by the OBJ smoother, but it is of course the slowest (12 iterations, 65.5 s). The final ROBJ' matches this number of iterations and is faster (36.5 s), but not as fast as RBJ and ROBJ.

The number of iterations, shown in Column 5 and 6 of Table 5 for OBJ and ROBJ', are the same when we set $d_{13} = 0$. This is due to the fact that $a_{33} \ll a_{11} = a_{22}$. Thus, the amount of flux that goes through the z -direction interfaces is much smaller than x -direction interfaces. Therefore, we can ignore the flux jumps on the z -direction interfaces when we smooth the new mortar defined on the x -direction interfaces (i.e., we can set $d_{13} = 0$).

Table 5
Number of iterations and CPU time for the original SPE10 dataset using the multiplicative two-level preconditioner, linear coarse and various local preconditioners.

M_0 coarse space	lin	lin	lin	lin	lin
$\tilde{\Omega}_{ij}$	$\begin{pmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 5 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 3 & 0 \\ 0 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 10 & 1 & 1 \\ 1 & 10 & 1 \\ 1 & 1 & 5 \end{pmatrix}$	$\begin{pmatrix} 4 & 1 & 0 \\ 1 & 8 & 1 \\ 1 & 1 & 2 \end{pmatrix}$
	BJ	RBJ	ROBJ	OBJ	ROBJ'
iter	21	25	22	12	12
CPU (s)	46.4	23.2	23.0	65.5	36.5

Table 6
Number of iterations and CPU time for the original SPE10 dataset with the flow driven by a quarter five-spot pattern of wells. We use the multiplicative two-level preconditioner and various coarse and local preconditioners.

M_0 coarse space	lin	lin	quad	quad	quad
$\tilde{\Omega}_{ij}$	$\begin{pmatrix} 4 & 1 & 0 \\ 1 & 7 & 1 \\ 1 & 1 & 2 \end{pmatrix}$	$\begin{pmatrix} 2 & 0 & 0 \\ 1 & 4 & 0 \\ 0 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 2 & 1 & 0 \\ 1 & 7 & 1 \\ 1 & 1 & 2 \end{pmatrix}$	$\begin{pmatrix} 2 & 0 & 0 \\ 1 & 4 & 0 \\ 0 & 0 & 2 \end{pmatrix}$	$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 2 \end{pmatrix}$
	ROBJ	ROBJ'	ROBJ	ROBJ'	RBJ
iter	31	44	18	27	36
CPU (s)	52.9	39.6	41.7	35.6	40.2

Table 7
Residual for the original SPE10 dataset with the flow driven by a quarter five-spot pattern of wells as solved using the PETSc MG as a preconditioner with an ILU(2) smoother.

	Iteration #						
	0	1	10	20	30	100	200
Residual	1.95e+2	4.20e+1	1.67e+0	2.52e−1	4.34e−2	2.33e−2	2.02e−2

Finally, we show results for a quarter five-spot pattern of wells, with an injection well at the origin and an production well at the furthest corner element. The boundary conditions are no-flow on the outer boundaries of Ω . Other physical parameters are the same as the above example, i.e., we use the original anisotropic SPE10 dataset. Results appear in Table 6, using piecewise linear and quadratic mortars. When we use a one-element overlapping smoothing domain, the ROBJ smoothers given have the fewest number of iterations (31 iterations, 52.9 s for linears and 18 iterations, 41.7 s for quadratics), while the ROBJ' local smoothers give the fastest times (44 iterations, 39.6 s for linears and 27 iterations, 35.6 s for quadratics). The nonoverlapping RBJ is almost as effective for quadratics (36 iterations, 40.2 s).

The sparse direct solver MUMPS took 303.5 s to solve the problem, which is almost 9 times more than our iterative solver. The PETSc multigrid (MG) and algebraic multigrid (AMG) were also tested as preconditioners for the accelerator GMRES. We set the local smoother to be the incomplete LU factorization with 2-level fill-in (ILU(2)). The tolerance is 10^{-5} . Other options are taken as the defaults. In Table 7, we show the convergence history of the multigrid preconditioner. At the first 30 iterations, the residual is reduced quickly. However, it stagnates and does not converge. This convergence behavior is very similar to the one level preconditioner without a coarse component for a homogeneous problem. This indicates that we need a better coarse preconditioner to achieve convergence for this extremely heterogeneous problem. The AMG preconditioner behaves the same as the MG preconditioner.

8.5. Parallel strong scalability study for a problem of size 16M

As a final example, we test a large problem using an isotropic, heterogeneous permeability generated by a geological software package on a uniform fine grid, which varies over about 6 orders of magnitude. We impose Dirichlet boundary conditions on the external faces that are perpendicular to the x -axis with normalized pressure 1 and 0, and a no-flow Neumann condition on the other faces.

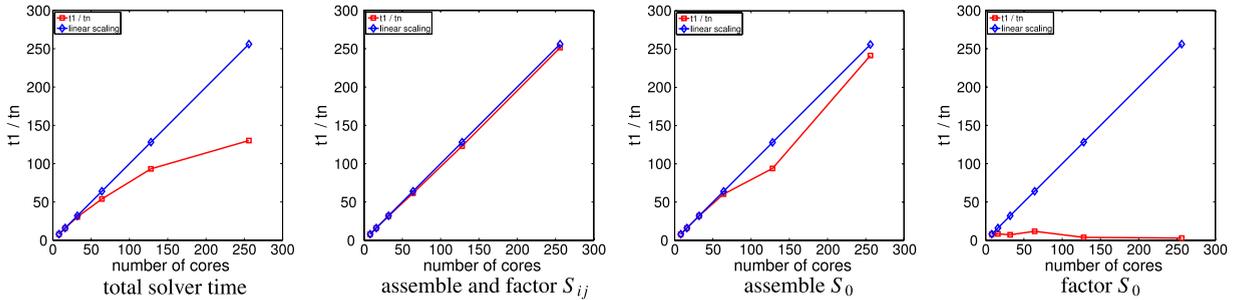


Fig. 8. Strong scalability plots (number of cores from 8 to 256 versus the ratio of the base time to the time required) for of the hybrid two-level preconditioner, including the total solver time, the time to assemble and factor the local matrices in the BJ smoother for applying S_{ij}^{-1} , the time to assemble the coarse interface matrix S_0 , and the time to factor the matrix S_0 using the sparse direct solver MUMPS. Ideal linear scaling plots are also shown.

Table 8

CPU times for factoring the coarse interface matrix S_0 with MUMPS. It does not scale well, since the size of the matrix is small ($34,560 \times 34,560$).

	Number of cores					
	8	16	32	64	128	256
Time (s)	0.49	0.47	0.55	0.33	1.04	1.35

We decompose the $256 \times 256 \times 256$ uniform fine grid into a $16 \times 16 \times 16$ coarse grid of subdomains, each with a $16 \times 16 \times 16$ subgrid. The number of fine grid elements is about 16.7 million, and the coarse mesh has 11,520 interfaces. We use piecewise linear functions to construct the coarse preconditioner M_0^{-1} . The size of the coarse mortar interface matrix S_0 is $34,560 \times 34,560$, but it is block sparse. The local preconditioner is the full nonoverlapping BJ smoother $M_{loc,BJ}^{-1}$ (32). The stopping criteria is that the residual is reduced by six orders of magnitude in the discrete ℓ_2 -norm. The hybrid two-level preconditioner takes 20 iterations to converge on the TACC computer *Stampede*, which was described above.

Strong scalability plots are shown in Fig. 8. Due to memory limitations, we use 8 cores to determine the base timings, and these are 180.6 s for the total solver time, 138.2 s to assemble and factor S_{ij} , 7.39 s to assemble S_0 , and 0.49 s to factor S_0 . We note that assembling and factoring the local preconditioner takes most of the CPU time. The only part of the system that does not scale well is factoring the coarse mortar interface matrix S_0 . The other routines in our implementation scale almost linearly.

In Table 8, we see that the time for factoring the matrix S_0 (of size $34,560 \times 34,560$) increases as the number of cores increases. This suggests that the communication time is significant compared with the CPU time of floating point operations; that is the size of the matrix S_0 is too small to scale well. Clearly, a more scalable coarse solver needs to be developed for our coarse solver, such as perhaps using multigrid to solve the interface system for the coarse mortar unknowns [28,29].

9. Summary and conclusions

We defined domain decomposition, two-level, additive, multiplicative, and hybrid iterative preconditioners (33)–(34) for elliptic problems in mixed form, most likely to be used within a Krylov accelerator such as PCG or GMRES. These two-level preconditioners incorporate a coarse preconditioner based on subdomain interfaces using the multiscale mortar method [14,35,36] and a local smoothing preconditioner based on block Jacobi (BJ), blocked by subdomain interfaces.

Given mixed finite element and fine scale mortar spaces for the discretization of the elliptic problem (4)–(6), we provided a framework for defining the coarse mortar space and thereby the coarse preconditioners. We need the technical assumption (10) for unique solvability, and the assumption (13), that the projection \mathcal{P} onto the space of normal fluxes of the coarse mortar space is contained in the projection of the fine mortar space. In terms of the coarse Schur complement matrix S_0 , the coarse preconditioner $M_0^{-1} = R_0^T S_0^{-1} R_0$ (i.e., (15)) is defined once the prolongation

R_0^T is defined. It is defined uniquely as $R_0 = L_0^T L (L^T L)^{-1}$ (i.e., (18)) by the condition that the projection \mathcal{P} of a coarse mortar agree with the projection of the extension (i.e., (16)). Application of R_0 can be simplified in the matching and nonmatching mesh cases.

We use local smoothers M_{loc}^{-1} based on BJ that arise from solving the local problem (27)–(29) on a domain $\tilde{\Omega}_{ij} \supset \Gamma_{ij}$. We use one of the full smoothing domain $\tilde{\Omega}_{ij} = \Omega_{ij}$ (BJ), a restrictive domain $\tilde{\Omega}_{ij} \subsetneq \Omega_{ij}$ (RBJ), an overlapping domain $\tilde{\Omega}_{ij} \supsetneq \Omega_{ij}$ (OBJ), or a domain restricted normal to Γ_{ij} and overlapping in the tangential directions (ROBJ). Generally, RBJ or ROBJ are the most efficient. The overlapping smoothers lead to an nonsymmetric local, and therefore also two-level, preconditioner.

The simplest additive two-level preconditioner that uses piecewise constant coarse mortars on matching grids has a preconditioned matrix with a condition number bounded by a multiple of $(\log(H/h))^2$ (Theorem 7.1), which is the same as in the classic BDD [18,24] and BDDC [26] two-level preconditioners, which are based on subdomains rather than interfaces.

Finally, we showed some numerical examples that demonstrate the convergence performance of the preconditioners. We observed that the two-level preconditioners are not very sensitive to the ratio of the highest to smallest permeability in a high contrast medium (at least when appearing in a checkerboard arrangement). We considered problems involving heterogeneous porous media, such as the permeability fields from the SPE10 benchmark problem. We observed that the preconditioned system has a low condition number and eigenvalues clustered around 1. We also found that it is often desirable, and even necessary, to use at least piecewise linear instead of piecewise constant coarse mortar spaces to achieve convergence and efficiency. Moreover, the use of restrictive local smoothers improved efficiency significantly, without increasing much the number of iterations to converge. Errors often accumulate around the corners of the subdomains, so the use of the nonsymmetric restrictive overlapping preconditioner (ROBJ) could result in a faster and more robust algorithm.

Acknowledgments

The authors were supported in part as part of the Center for Frontiers of Subsurface Energy Security, an Energy Frontier Research Center funded by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences under Award Number DE-SC0001114, and supported in part by the U.S. National Science Foundation grant DMS-0835745.

References

- [1] D.W. Peaceman, *Fundamentals of Numerical Reservoir Simulation*, Elsevier, Amsterdam, 1977.
- [2] Z. Chen, G. Huan, Y. Ma, *Computational Methods for Multiphase Flows in Porous Media*, in: *Computational Science and Engineering Series*, vol. 2, SIAM, Philadelphia, 2006.
- [3] M.A. Christie, M.J. Blunt, Tenth SPE comparative solution project: a comparison of upscaling techniques, *SPE Reserv. Eval. Eng.* 4 (2001) 308–317.
- [4] Y. Efendiev, T.Y. Hou, *Multiscale Finite Elements Methods*, in: *Surveys and Tutorials in the Applied Mathematical Sciences*, vol. 4, Springer, New York, 2009.
- [5] E. Luo, *Multigrid method for elliptic equation with oscillatory coefficients* (Ph.D. thesis), Department of Mathematics, UCLA, 1993.
- [6] B. Engquist, E. Luo, Convergence of a multigrid method for elliptic equations with highly oscillatory coefficients, *SIAM J. Numer. Anal.* 37 (6) (1997) 2254–2273.
- [7] J. Aarnes, T.Y. Hou, Multiscale domain decomposition methods for elliptic problems with high aspect ratios, *Acta Math. Appl. Sin. Engl. Ser.* 18 (1) (2002) 63–76.
- [8] W. E, *Principle of Multiscale Modelling*, Cambridge University Press, 2010.
- [9] A. Brandt, *Multiscale scientific computation: review 2001*, in: *Multiscale and Multiresolution Methods*, Springer Verlag, 2001, pp. 1–96.
- [10] J. Geiser, *Coupled Systems: Theory, Models, and Applications in Engineering*, in: *Numerical Analysis and Scientific Computing Series*, Chapman & Hall/CRC, 2014.
- [11] R. Glowinski, M.F. Wheeler, Domain decomposition and mixed finite element methods for elliptic problems, in: R. Glowinski, et al. (Eds.), *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1988, pp. 144–172.
- [12] C. Bernardi, Y. Maday, A.T. Patera, A new nonconforming approach to domain decomposition: the mortar element method, in: H. Brezis, J.L. Lions (Eds.), *Nonlinear Partial Differential Equations and their Applications*, Longman Scientific & Technical, UK, 1994, pp. 13–51.
- [13] T. Arbogast, L.C. Cowsar, M.F. Wheeler, I. Yotov, Mixed finite element methods on non-matching multiblock grids, *SIAM J. Numer. Anal.* 37 (2000) 1295–1315.
- [14] T. Arbogast, G. Pencheva, M.F. Wheeler, I. Yotov, A multiscale mortar mixed finite element method, *Multiscale Model. Simul.* 6 (1) (2007) 319–346.

- [15] S.V. Nepomnyaschikh, Domain decomposition and Schwarz methods in a subspace for the approximate solution of elliptic boundary value problems (Ph.D. thesis), Computing Center of the Siberian Branch of the USSR Academy of Sciences, Novosibirsk, USSR, 1986.
- [16] B. Smith, An optimal domain decomposition preconditioner for the finite element solution of linear elasticity problems, *SIAM J. Sci. Stat. Comput.* 13 (1) (1992) 364–378.
- [17] M. Dryja, O. Widlund, Domain decomposition algorithms with small overlap, *SIAM J. Sci. Comput.* 15 (3) (1994) 604–620.
- [18] L.C. Cowsar, Some domain decomposition and multigrid preconditioners for hybrid mixed finite elements (Ph.D. thesis), Rice University, 1994.
- [19] J. Xu, J. Zou, Some non-overlapping domain decomposition methods, *SIAM Rev.* 40 (1998) 857–914.
- [20] X. Cai, M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, *SIAM J. Sci. Comput.* 21 (2) (1999) 792–797.
- [21] E. Efstathiou, M.J. Gander, Why restricted additive Schwarz converges faster than additive Schwarz, *BIT* 43 (2003) 945–959.
- [22] A. Toselli, O. Widlund, *Domain Decomposition Methods—Algorithms and Theory*, Springer-Verlag, Berlin, 2005.
- [23] T.P.A. Mathew, *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations*, in: *Lecture Notes in Computational Science and Engineering*, vol. 61, Springer, Berlin, 2008.
- [24] L.C. Cowsar, J. Mandel, M.F. Wheeler, Balancing domain decomposition for mixed finite elements, *Math. Comp.* 64 (211) (1995) 989–1015.
- [25] G. Pencheva, I. Yotov, Balancing domain decomposition for mortar mixed finite element methods, *Numer. Linear Algebra Appl.* 10 (2003) 159–180.
- [26] X. Tu, A BDDC algorithm for flow in porous media with a hybrid finite element discretization, *Electron. Trans. Numer. Anal.* 26 (2007) 146–160.
- [27] H. Zhou, H. Tchelepi, Two-stage algebraic multiscale linear solver for highly heterogeneous reservoir models, *SPE J.* 17 (2) (2012) 523–539. 141473-PA.
- [28] L.C. Cowsar, A. Weiser, M.F. Wheeler, Parallel multigrid and domain decomposition algorithms for elliptic equations, in: D. Keyes, et al. (Eds.), *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 1992, pp. 376–385.
- [29] M.F. Wheeler, I. Yotov, Multigrid on the interface for mortar mixed finite element methods for elliptic problems, *Comput. Methods Appl. Mech. Engrg.* 184 (2000) 287–302.
- [30] J. Mandel, Iterative solvers by substructuring for the p-version finite element method, *Comput. Methods Appl. Mech. Engrg.* 58 (1990) 79–93.
- [31] B. Smith, P.E. Bjørstad, W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, New York, 1996.
- [32] F. Brezzi, M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.
- [33] J.E. Roberts, J.-M. Thomas, Mixed and hybrid methods, in: P.G. Ciarlet, J.L. Lions (Eds.), *Handbook of Numerical Analysis*. Vol. 2, Elsevier Science Publishers B.V., North-Holland, Amsterdam, 1991, pp. 523–639. (Chapter) *Finite Element Methods (Part 1)*.
- [34] R.A. Raviart, J.M. Thomas, A mixed finite element method for 2nd order elliptic problems, in: I. Galligani, E. Magenes (Eds.), *Mathematical Aspects of Finite Element Methods*, in: *Lecture Notes in Math.*, vol. 606, Springer-Verlag, New York, 1977, pp. 292–315.
- [35] T. Arbogast, H. Xiao, A multiscale mortar mixed space based on homogenization for heterogeneous elliptic problems, *SIAM J. Numer. Anal.* 51 (1) (2013) 377–399.
- [36] T. Arbogast, Z. Tao, H. Xiao, Multiscale mortar mixed methods for heterogeneous elliptic problems, in: J. Li, et al. (Eds.), *Recent Advances in Scientific Computing and Applications*, in: *Contemporary Mathematics*, vol. 586, Amer. Math. Soc., Providence, Rhode Island, 2013, pp. 9–21.
- [37] M.R. Hestenes, E.L. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Res. Natl. Bur. Stand.* 49 (6) (1952) 409–436.
- [38] Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, 2003.
- [39] Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM J. Sci. Stat. Comput.* 7 (3) (1986) 856–869.
- [40] J. Galvis, Y. Efendiev, Domain decomposition preconditioners for multiscale flows in high-contrast media, *Multiscale Model. Simul.* 8 (4) (2010) 1461–1483.
- [41] Y. Efendiev, J. Galvis, X.-H. Wu, Multiscale finite element methods for high-contrast problems using local spectral basis functions, *J. Comput. Phys.* 230 (4) (2011) 937–955.
- [42] T. Arbogast, C.-S. Huang, An Eulerian–Lagrangian WENO scheme for nonlinear conservation laws, in preparation.
- [43] T. Arbogast, M.F. Wheeler, I. Yotov, Mixed finite elements for elliptic problems with tensor coefficients as cell-centered finite differences, *SIAM J. Numer. Anal.* 34 (1997) 828–852.
- [44] P. Jenny, Numerical subgrid upscaling of flow in porous media: multi-scale flux continuous finite-difference method for the elliptic problem, Internal Technical Report, Intersect Alliance Technology, Chevron Petroleum Technology Co., San Ramon, CA, USA, 2001.
- [45] P. Jenny, S.H. Lee, H.A. Tchelepi, Multi-scale finite-volume method for elliptic problems in subsurface flow simulation, *J. Comput. Phys.* 187 (2003) 47–67.
- [46] P. Jenny, S.H. Lee, H.A. Tchelepi, Adaptive multiscale finite-volume method for multiphase flow and transport in porous media, *Multiscale Model. Simul.* 3 (1) (2005) 50–64. <http://dx.doi.org/10.1137/030600795>.
- [47] S. Sun, M.F. Wheeler, Projections of velocity data for the compatibility with transport, *Comput. Methods Appl. Mech. Engrg.* 195 (2006) 653–673.
- [48] W.L. Wan, T.F. Chan, B. Smith, An energy-minimizing interpolation for robust multigrid methods, *SIAM J. Sci. Comput.* 21 (4) (2000) 1632–1649.
- [49] F. Nataf, H. Xiang, V. Dolean, N. Spillane, A coarse space construction based on local Dirichlet-to-Neumann maps, *SIAM J. Sci. Comput.* 33 (4) (2011) 1623–1642.
- [50] D.N. Arnold, F. Brezzi, Mixed and nonconforming finite element methods: implementation, postprocessing and error estimates, *RAIRO Modél. Math. Anal. Numér.* 19 (1985) 7–32.

- [51] M. Čížková, J. Šístek, P. Burda, On selection of interface weights in domain decomposition methods, in: *Programs and Algorithms of Numerical Mathematics*, vol. 16, Institute of Mathematics AS CR Prague, 2012, pp. 35–44.
- [52] J. Mandel, Hybrid domain decomposition with unstructured subdomains, in: J. Mandel, et al. (Eds.), *Sixth International Conference on Domain Decomposition Methods, Contemporary Mathematics*, American Mathematical Society, 1992, pp. 103–112.
- [53] J.M. Tang, R. Nabben, C. Vuik, Y.A. Erlangga, Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods, *J. Sci. Comput.* 39 (2009) 340–370.
- [54] R. Nabben, C. Vuik, A comparison of abstract versions of deflation, balancing and additive coarse grid correction preconditioners, *Numer. Linear Algebra Appl.* 15 (2008) 355–372.
- [55] H. Xiao, *Multiscale mortar mixed finite element methods for flow problems in highly heterogeneous porous media* (Ph.D. thesis), University of Texas at Austin, 2013.
- [56] F. Brezzi, J. Douglas Jr., L.D. Marini, Two families of mixed elements for second order elliptic problems, *Numer. Math.* 47 (1985) 217–235.
- [57] M.A. Casarin, *Schwarz preconditioners for spectral and mortar finite element methods with applications to incompressible fluids* (Ph.D. thesis), Courant Institute of Mathematical Sciences, New York University, 1996.
- [58] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J.D. Croz, A. Greenbaum, S. Hammarling, A. McKenney, D. Sorensen, *LAPACK Users' Guide*, third ed., Society for Industrial and Applied Mathematics, 1999.
- [59] P.R. Amestoy, I.S. Duff, J.-Y. L'Excellent, J. Koster, A fully asynchronous multifrontal solver using distributed dynamic scheduling, *SIAM J. Matrix Anal. Appl.* 23 (1) (2001) 15–41.
- [60] P.R. Amestoy, A. Guermouche, J.-Y. L'Excellent, S. Pralet, Hybrid scheduling for the parallel solution of linear systems, *Parallel Comput.* 32 (2006) 136–156.
- [61] I.S. Duff, A.M. Erisman, J.K. Reid, *Direct Methods for Sparse Matrices*, in: *Monographs on Numerical Analysis*, Oxford University Press, USA, 1989.
- [62] O. Schenk, M. Bollhoefer, R. Roemer, On large-scale diagonalization techniques for the Anderson model of localization, *SIAM Rev.* 50 (2008) 91–112.
- [63] O. Schenk, A. Waechter, M. Hagemann, Matching-based preprocessing algorithms to the solution of saddle-point problems in large-scale nonconvex interior-point optimization, *J. Optim. Theory Appl.* 36 (2–3) (2007) 321–341.
- [64] O. Schenk, K. Gärtner, Solving unsymmetric sparse systems of linear equations with PARDISO, *J. Future Gener. Comput. Syst.* 20 (3) (2004) 475–487.
- [65] O. Schenk, K. Gärtner, On fast factorization pivoting methods for symmetric indefinite systems, *Electron. Trans. Numer. Anal.* 23 (2006) 158–179.
- [66] W. Gropp, E.L. Lusk, A. Skjellum, *Using MPI: Portable Parallel Programming with the Message Passing Interface*, The MIT Press, Cambridge, 1999.
- [67] S. Balay, J. Brown, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Web page, 2013. <http://www.mcs.anl.gov/petsc>.
- [68] S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, *PETSc users manual*, Tech. Rep. ANL-95/11—Revision 3.4, Argonne National Laboratory, 2013.
- [69] S. Balay, W.D. Gropp, L.C. McInnes, B.F. Smith, Efficient management of parallelism in object oriented numerical software libraries, in: E. Arge, A.M. Bruaset, H.P. Langtangen (Eds.), *Modern Software Tools in Scientific Computing*, Birkhäuser Press, 1997, pp. 163–202.