Available online at www.sciencedirect.com

ScienceDirect

Comput. Methods Appl. Mech. Engrg. 313 (2017) 259–278

Computer methods in applied mechanics and engineering

www.elsevier.com/locate/cma

A multiscale flux basis for mortar mixed discretizations of Stokes–Darcy flows

Benjamin Ganis[a], Danail Vassilev[b], ChangQing Wang[c,*], Ivan Yotov[c]

[a] Center for Subsurface Modeling, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, TX, 78712, USA
[b] Cobham plc, UK
[c] Department of Mathematics, University of Pittsburgh, Pittsburgh, PA 15260, USA

Received 4 August 2016; accepted 28 September 2016
Available online 4 October 2016

Highlights

- A multiscale flux basis algorithm for the coupled Stokes–Darcy problem is developed.
- The algorithm is more efficient than the original MMMFEM implementation.
- Only fixed number of local fine grid solves are required.
- Gain in efficiency increases with problem complexity.
- The algorithm can improve the efficiency of the balancing preconditioner in Darcy.

Abstract

A multiscale flux basis algorithm is developed for the Stokes–Darcy flow problem. The method is based on a non-overlapping domain decomposition algorithm, where the global problem is reduced to a coarse scale mortar interface problem that is solved by an iterative solver. Subdomain solves are required at each interface iteration, so the cost for the method without a multiscale basis can be high when the number of subdomains or the condition number of the interface problem is large. The proposed algorithm involves precomputing a multiscale flux basis, which consists of the flux (or velocity trace) response from each mortar degree of freedom. It is computed by each subdomain independently before the interface iteration begins. The subdomain solves required at each iteration are substituted by a linear combination of the multiscale basis. This may lead to a significant reduction in computational cost since the number of subdomain solves is fixed, depending only on the number of mortar degrees of freedom associated with a subdomain. Several numerical examples are carried out to demonstrate the efficiency of the multiscale flux basis implementation for large-scale Stokes–Darcy problems.

Keywords: Multiscale flux basis; Mortar finite element; Mixed finite element; Stokes–Darcy flow; Non-overlapping domain decomposition; FETI method, balancing preconditioner

* Corresponding author.
E-mail addresses: bganis@ices.utexas.edu (B. Ganis), chw92@pitt.edu (C. Wang), yotov@math.pitt.edu (I. Yotov).

http://dx.doi.org/10.1016/j.cma.2016.09.037
0045-7825/© 2016 Elsevier B.V. All rights reserved.

1. Introduction

In this paper, a mortar mixed finite element method using multiscale flux basis is developed for the coupled Stokes–Darcy flow problem. The Stokes–Darcy problem has been studied intensively due to its various applications, such as coupled ground water and surface flows, flows in fractured porous media, arterial flows, fuel cells, and others. The Stokes and Darcy equations model the flow motion and infiltration process, respectively. The Beavers–Joseph–Saffman slip with friction condition [1,2] is applied on the Stokes–Darcy interface. The numerical approximation of the coupled problem has been studied extensively, see, e.g., [3–6]. In our approach, we utilize the multiscale mortar mixed finite element method (MMMFEM) with non-overlapping domain decomposition [7–9]. In this method, the computational domain is decomposed into several subdomains of either Stokes or Darcy type. Each subdomain is discretized on a local fine mesh, allowing for non-matching grids across the subdomain interfaces. This capability is helpful in practice, since it allows for describing complex geometries as a union of simpler locally discretized subdomains, as well as resolving internal features such as geological layers and faults. A mortar finite element space is introduced on the interfaces, which serves as a Lagrange multiplier to impose weakly appropriate continuity conditions: normal velocity and normal stress on Stokes–Darcy interfaces, normal velocity and pressure on Darcy–Darcy interfaces, and velocity vector and normal stress vector on Stokes–Stokes interfaces. The numerical analysis of the method is carried out in [7] on fairly general grid configurations, allowing for multiscale approximations with coarse scale H mortar spaces and fine scale h subdomain discretizations. The work in [7] extends earlier works that consider a single Stokes–Darcy interface and employ a mortar space defined as the normal trace of the Darcy velocity space [3,5,6,10]. A non-overlapping domain decomposition for the Stokes–Darcy problem with many subdomains is developed and analyzed in [9]. Earlier works in two-subdomain case include [11–15]. In [9], by eliminating the subdomain unknowns, the fine scale global problem is reduced to an interface problem that can be solved by an iterative method. The action of the interface operator requires solving Neumann problems in Stokes subdomains and Dirichlet problems in Darcy subdomains. The finite element tearing and interconnecting (FETI) method [16,17] is employed to deal with the possibly singular Stokes subdomain problems. This is done by projecting the iterates into a space orthogonal to the kernel of the subdomain operators via solving a coarse space problem. The main computational cost of the algorithm comes from the subdomain solves required in every interface iteration. Increasing the number of subdomains and refining the grids both lead to an increase in the number of iterations and the number of subdomain solves.

In this paper, we develop an alternative implementation of the MMMFEM for Stokes–Darcy flows that is based on precomputing a multiscale flux basis, which can reduce the computational cost significantly. Our approach extends the multiscale flux basis implementation of the MMMFEM for Darcy flow developed in [18]. The MMMFEM was first developed in [19] for Darcy problems. It is an alternative to other existing multiscale methods such as the variational multiscale method [20,21] and the multiscale finite element method [22,23]. Methods involving enriched multiscale basis for high-contrast problems using local spectral information have been developed in [24–26]. All three methods require solving relatively small fine scale subdomain problems that are only coupled on the coarse scale through a reduced number of degrees of freedom. The mortar multiscale approach provides the extra flexibility to adaptively refine the mortar grids based on a posteriori error estimation in order to improve the global accuracy [19]. The variational multiscale method and multiscale finite elements both compute a multiscale basis by solving a fixed number of local fine scale problems with boundary conditions or a source term corresponding to the coarse scale degrees of freedom. This basis is then used to solve the coarse scale problem. The multiscale flux basis implementation of the MMMFEM for Stokes–Darcy flows developed in this paper provides a similar computational structure. The method yields the same solution as the original MMMFEM implementation but can be much more computationally efficient. A multiscale flux basis consists of the flux (or velocity trace) response from each mortar degree of freedom, which is computed by each subdomain independently before the interface iteration begins. Then the subdomain solves during the interface iteration can be replaced by linear combinations of the multiscale flux basis. This implementation has a number of fine scale subdomain solves that is independent of the number of interface iterations. It reduces the computational cost if there are more iteration steps than number of mortar degree of freedoms per subdomain. In addition, when performing studies where the same input data is used repeatedly in different situations, the multiscale flux basis can be computed once and stored to disk in an offline step, so it can be reused across different simulations. A typical example is the stochastic Stokes–Darcy flow problem [27], where the permeability in the Darcy region is given as a stochastic parameter presented by a sum of the local Karhunen–Loève (KL) expansion [28].

The development of the multiscale flux basis implementation of the MMMFEM for Stokes–Darcy flows involves a number of major technical difficulties compared to the Darcy problem. One issue is the need to solve different types of local problems to compute the basis, Dirichlet in Darcy and Neumann in Stokes. Another difficulty is handling singular full Neumann Stokes subdomains. In particular, since the Neumann boundary condition provided by the mortar basis leads to a right hand side that is not orthogonal to the kernel of the subdomain matrix, computing the multiscale flux basis in Stokes involves solving incompatible Neumann problems. However, due to the application of the FETI coarse solve, the multiscale basis is used only to compute the action of the interface operator on compatible data, which is a well-posed Neumann solve. Another new development in this paper is combining the use of the multiscale flux basis with the balancing preconditioner in the Darcy region [29–31]. The use of the preconditioner is motivated by the fact that the number of interface iterations is not insignificant for the CPU time, with the cost for computing orthogonal projections and linear combinations, and the inter-processor communication time all playing a role. The balancing preconditioner involves solving Neumann subdomain problems and a coarse problem to exchange global information. It is very efficient and exhibits condition number that grows very mildly with respect to mesh and subdomain size. As a result, the number of interface iterations is reduced significantly, but at the cost of one additional Dirichlet and one Neumann solve per Darcy subdomain per iteration. While the multiscale flux basis can be utilized for the efficient computation of the extra Dirichlet solves, one needs to compute a new multiscale basis for the Neumann solves. This results in a preconditioned algorithm with the number of local solves independent of the number of interface iterations.

In the numerical examples we compare the computational cost of the implementations with and without multiscale flux basis. Our tests for a wide range of problems show that the multiscale flux basis can improve the efficiency for both unpreconditioned and preconditioned problems.

The rest of the paper is organized as follows. The Stokes–Darcy coupled problem, its domain decomposition formulation, the MMMFEM discretization, and the reduction to a mortar interface problem are presented in Section 2. The multiscale flux basis algorithm is developed in Section 3. Numerical examples illustrating the efficiency of the method are presented in Section 4. We give some conclusions in Section 5.

2. Model problem and MMMFEM

2.1. Stokes–Darcy coupled problem

Assuming $d = 2, 3$, let $\Omega_S \subset \mathbb{R}^d$ be the fluid region governed by the Stokes equations, with outside boundary Γ_S and outward unit normal $\mathbf{n}_S$. Let $\Omega_D \subset \mathbb{R}^d$ be the porous media region governed by Darcy's law, with outside boundary Γ_D and outward unit normal $\mathbf{n}_D$. Each region is a union of possibly disjoint subregions. Let $\Omega = \Omega_S \cup \Omega_D$ be the entire simulation domain and let $\Gamma_{SD} = \partial\Omega_S \cap \partial\Omega_D$ be the Stokes–Darcy interface. The velocity and pressure in Ω_S and Ω_D are denoted by $\mathbf{u}_S$, p_S and $\mathbf{u}_D$, p_D, respectively. Let μ_S and μ_D be the viscosity coefficients in the Stokes and Darcy regions, and let $\mathbf{K}$ be the permeability tensor, assumed to be symmetric and uniformly positive definite. The deformation rate tensor $\mathbf{D}$ and the stress tensor $\mathbf{T}$ of the Stokes flow are denoted by

$$\mathbf{D}(\mathbf{u}_S) := \frac{1}{2}(\nabla\mathbf{u}_S + (\nabla\mathbf{u}_S)^T), \quad \mathbf{T}(\mathbf{u}_S, p_S) := -p_S\mathbf{I} + 2\mu_S\mathbf{D}(\mathbf{u}_S).$$

The Stokes flow model with no-slip boundary condition and body force $\mathbf{f}_S$ is:

$$-\nabla\cdot\mathbf{T}(\mathbf{u}_S, p_S) \equiv -2\mu_S\nabla\cdot\mathbf{D}(\mathbf{u}_S) + \nabla p_S = \mathbf{f}_S \quad \text{in } \Omega_S, \tag{1}$$

$$\nabla\cdot\mathbf{u}_S = 0 \quad \text{in } \Omega_S, \tag{2}$$

$$\mathbf{u}_S = \mathbf{0} \quad \text{on } \Gamma_S. \tag{3}$$

The Darcy flow model with no-flow boundary condition, gravity force $\mathbf{f}_D$ and external source q_D is:

$$\mu_D\mathbf{K}^{-1}\mathbf{u}_D + \nabla p_D = \mathbf{f}_D \quad \text{in } \Omega_D, \tag{4}$$

$$\nabla\cdot\mathbf{u}_D = q_D \quad \text{in } \Omega_D, \tag{5}$$

$$\mathbf{u}_D\cdot\mathbf{n}_D = 0 \quad \text{on } \Gamma_D. \tag{6}$$

We assume that the source q_D satisfies the solvability condition $\int_{\Omega_D} q_D = 0$. The interface conditions on Γ_{SD} are:

$$\mathbf{u}_S\cdot\mathbf{n}_S + \mathbf{u}_D\cdot\mathbf{n}_D = 0 \quad \text{on } \Gamma_{SD}, \tag{7}$$

$$-(\mathbf{T}(\mathbf{u}_S, p_S)\mathbf{n}_S)\cdot\mathbf{n}_S \equiv p_S - 2\mu_S(\mathbf{D}(\mathbf{u}_S)\mathbf{n}_S)\cdot\mathbf{n}_S = p_D \quad \text{on } \Gamma_{SD}, \tag{8}$$

$$-(\mathbf{T}(\mathbf{u}_S, p_S)\mathbf{n}_S)\cdot\boldsymbol{\tau}^l_{SD} \equiv -2\mu_S(\mathbf{D}(\mathbf{u}_S)\mathbf{n}_S)\cdot\boldsymbol{\tau}^l_{SD} = \frac{\mu_S\alpha}{\sqrt{K_l}}\mathbf{u}_S\cdot\boldsymbol{\tau}^l_{SD}, \quad 1 \le l \le d-1 \quad \text{on } \Gamma_{SD}. \tag{9}$$

Conditions (7) and (8) incorporate continuity of flux and normal stress, respectively. Condition (9) represents the Beaver–Joseph–Saffman slip with friction condition [1,2], where $K_l = (\mathbf{K}\boldsymbol{\tau}^l_{SD})\cdot\boldsymbol{\tau}^l_{SD}$, $\{\boldsymbol{\tau}^l_{SD}\}_{l=1}^{d-1}$ is an orthogonal system of unit tangent vectors on Γ_{SD}, and the constant $\alpha \ge 0$ is determined experimentally.

The L^2-inner product and norm of scalar and vector valued functions in domain $G \subset \mathbb{R}^d$ are denoted by $(\cdot,\cdot)_G$ and $\|\cdot\|_G$, respectively. We omit the subscript G if $G = \Omega$. For a section of the interface or the domain boundary $S \subset \mathbb{R}^{d-1}$, the L^2-inner product (or duality pairing) and norm are denoted by $\langle\cdot,\cdot\rangle_S$ and $\|\cdot\|_S$, respectively.

2.2. Domain decomposition and variational formulation

The domain Ω is decomposed into N non-overlapping subdomains Ω_i, $i = 1,\ldots,N$, where

$$\Omega_S = \cup_{i=1}^{N_S}\Omega_i, \quad \Omega_D = \cup_{i=N_S+1}^{N}\Omega_i, \quad N = N_S + N_D.$$

For $1 \le i < j \le N$, we define $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$ as the interface between any two subdomains, which can be of zero measure if they are not adjacent. Define $\Gamma_i = \cup_j \Gamma_{ij}$ as the union of interfaces of subdomain Ω_i and $\Gamma = \cup_{i,j}\Gamma_{i,j}$ as the union of all interfaces. Let Γ_{SS} be the set of Stokes–Stokes interfaces and let Γ_{DD} be the set of Darcy–Darcy interfaces. The following interface conditions are imposed:

$$[\mathbf{u}_D\cdot\mathbf{n}] = 0,\ [p_D] = 0 \text{ on } \Gamma_{DD}, \quad [\mathbf{u}_S] = \mathbf{0},\ [\mathbf{T}(\mathbf{u}_S, p_S)\mathbf{n}] = \mathbf{0} \text{ on } \Gamma_{SS}, \tag{10}$$

where $[\ \cdot\ ]$ denotes the jump on the interface. In particular, on Γ_{ij}, $[p] = (p_i - p_j)|_{\Gamma_{ij}}$, $[\mathbf{u}\cdot\mathbf{n}] = \mathbf{u}_i\cdot\mathbf{n}_i + \mathbf{u}_j\cdot\mathbf{n}_j$, with the notation $\mathbf{u}_i := \mathbf{u}|_{\Omega_i}$, $p_i := p|_{\Omega_i}$, $\mathbf{n}_i$ being the outward unit normal vector to $\partial\Omega_i$. Also, denote $\mathbf{f}_i = \mathbf{f}_S|_{\Omega_i}$ for $1 \le i \le N_S$ and $\mathbf{f}_i = \mathbf{f}_D|_{\Omega_i}$ for $N_S + 1 \le i \le N$.

Following the variational formulation derived in [7], we define the velocity and pressure spaces

$$V^S = \{\mathbf{v}_S \in (L^2(\Omega_S))^d : \mathbf{v}_S|_{\Omega_i} \in (H^1(\Omega_i))^d, 1 \le i \le N_S, \mathbf{v}_S = 0 \text{ on } \Gamma_S\},\quad W^S = L^2(\Omega_S)$$

in the Stokes region Ω_S, and

$$V^D = \{\mathbf{v}_D \in (L^2(\Omega_D))^d : \mathbf{v}_D|_{\Omega_i} \in H(\text{div};\Omega_i), N_s + 1 \le i \le N, \mathbf{v}_D\cdot\mathbf{n}_D = 0 \text{ on } \Gamma_D\},$$
$$W^D = L^2(\Omega_D)$$

in the Darcy region Ω_D, where

$$H(\text{div};\Omega_i) = \{\mathbf{v} \in (L^2(\Omega_i))^d : \nabla\cdot\mathbf{v} \in L^2(\Omega_i)\}.$$

The velocity and pressure spaces on the whole domain are given by $V = V^S \times V^D$ and

$$W = \left\{ w = (w_S, w_D) \in W^S \times W^D : \int_\Omega w = 0 \right\}.$$

To impose the continuity conditions on the interfaces we define the Lagrange multiplier space

$$\Lambda = \Lambda^{SD}\times\Lambda^{DD}\times\Lambda^{SS},\ \Lambda^{DD} = (V^D\cdot\mathbf{n}|_{\Gamma_{DD}})',\ \Lambda^{SD} = (V^D\cdot\mathbf{n}|_{\Gamma_{SD}})',\ \Lambda^{SS} = (V^S|_{\Gamma_{SS}})'.$$

The variational formulation of the coupled problems (1) – (9) is: find $(\mathbf{u}, p, \lambda) \in V\times W\times\Lambda$ such that

$$a(\mathbf{u},\mathbf{v}) + b(\mathbf{v},p) + b_I(\mathbf{v},\lambda) = (\mathbf{f},\mathbf{v}), \quad \forall\,\mathbf{v}\in V, \tag{11}$$

$$b(\mathbf{u},w) = -(q_D, w)_{\Omega_D}, \quad \forall\,w\in W, \tag{12}$$

$$b_I(\mathbf{u},\mu) = 0, \quad \forall\,\mu\in\Lambda, \tag{13}$$

where

$$a_i(\mathbf{u},\mathbf{v}) = 2\,\mu_S(\mathbf{D}(\mathbf{u}_i),\mathbf{D}(\mathbf{v}_i))_{\Omega_i} + \sum_{l=1}^{d-1}\left\langle\frac{\mu_S\alpha}{\sqrt{K_l}}(\mathbf{u}_i\cdot\boldsymbol{\tau}_l)(\mathbf{v}_i\cdot\boldsymbol{\tau}_l)\right\rangle_{\partial\Omega_i\cap\Gamma_{SD}},$$

$$
\begin{aligned}
&\quad 1 \le i \le N_S, \quad \forall (\mathbf{u}, \mathbf{v}) \in V^S \times V^S, \\
&a_i(\mathbf{u}, \mathbf{v}) = \mu_D (\mathbf{K}^{-1}\mathbf{u}_i, \mathbf{v}_i)_{\Omega_i}, \quad N_S + 1 \le i \le N, \quad \forall (\mathbf{u}, \mathbf{v}) \in V^D \times V^D, \\
&b_i(\mathbf{v}, w) = -(\nabla \cdot \mathbf{v}_i, w_i)_{\Omega_i}, \quad 1 \le i \le N, \quad \forall \mathbf{v} \in V, \quad \forall w \in W, \\
&a(\mathbf{u}, \mathbf{v}) = \sum_{i=1}^{N} a_i(\mathbf{u}, \mathbf{v}), \quad b(\mathbf{v}, w) = \sum_{i=1}^{N} b_i(\mathbf{v}, w), \\
&b_I(\mathbf{v}, \mu) = \langle [\mathbf{v}], \mu \rangle_{\Gamma_{SS}} + \langle [\mathbf{v} \cdot \mathbf{n}], \mu \rangle_{\Gamma_{DD}} + \langle [\mathbf{v} \cdot \mathbf{n}], \mu \rangle_{\Gamma_{SD}}, \quad \forall (\mathbf{v}, \mu) \in V \times \Lambda.
\end{aligned}
$$

The Lagrange multiplier λ has the physical meaning of normal stress vector on Γ_{SS} and pressure on $\Gamma_{DD} \cup \Gamma_{SD}$. Eq. (13) is needed to weakly enforce the continuity conditions (7) and (10) on the different types of interfaces. The reader is referred to [7,3] for proof of the existence and uniqueness of a solution to the variational formulation (11)–(13).

2.3. Discretization

For simplicity, we denote $V_i = V|_{\Omega_i}$ for $1 \le i \le N$. In the MMMFEM [7,9], each subdomain Ω_i is discretized with a d-dimensional shape regular finite element partition $\mathcal{T}_{h_i}$, where h_i is the maximal element diameter. For any adjacent subdomains Ω_i and Ω_j, the partitions $\mathcal{T}_{h_i}$ and $\mathcal{T}_{h_j}$ need not match on Γ_{ij}. In addition, a coarse $(d-1)$-dimensional quasi-uniform affine mesh $\mathcal{T}_{H_{ij}}$ is defined on the interface Γ_{ij} with maximal element size H_{ij}. Let $h = \max_{i=1}^{N} h_i$ and $H = \max_{i,j} H_{ij}$. In any Stokes subdomain Ω_i, $1 \le i \le N_S$, let $V_{h,i} \times W_{h,i} \subset V_i \times W_i$ be a pair of finite element spaces satisfying the following discrete inf–sup condition for some constant $\beta_S > 0$:

$$
\inf_{0 \ne w_{h,i} \in W_{h,i}} \sup_{0 \ne \mathbf{v}_{h,i} \in V_{h,i}} \frac{(w_{h,i}, \nabla \cdot \mathbf{v}_{h,i})_{\Omega_i}}{\|\mathbf{v}_{h,i}\|_{H^1(\Omega_i)} \|w_{h,i}\|_{L^2(\Omega_i)}} \ge \beta_S > 0, \quad 1 \le i \le N_S. \tag{14}
$$

Some well-known examples of pairs satisfying (14) are the Taylor–Hood element [32], the MINI element [33], and the Bernardi–Raugel element [34]. In any Darcy subdomain Ω_i, $N_S + 1 \le i \le N$, let $V_{h,i} \times W_{h,i} \subset V_i \times W_i$ be a pair of mixed finite element spaces satisfying $\nabla \cdot V_{h,i} \subset W_{h,i}$ and the discrete inf–sup condition for some constant $\beta_D > 0$:

$$
\inf_{0 \ne w_{h,i} \in W_{h,i}} \sup_{0 \ne \mathbf{v}_{h,i} \in V_{h,i}} \frac{(w_{h,i}, \nabla \cdot \mathbf{v}_{h,i})_{\Omega_i}}{\|\mathbf{v}_{h,i}\|_{H(\mathrm{div};\Omega_i)} \|w_{h,i}\|_{L^2(\Omega_i)}} \ge \beta_D > 0, \quad N_S + 1 \le i \le N. \tag{15}
$$

Well-known pairs that satisfy these conditions include the Raviart–Thomas spaces [35], the Brezzi–Douglas–Marini (BDM) spaces [36], and the Brezzi–Douglas–Duran–Fortin (BDDF) spaces [37]. On each interface Γ_{ij}, a mortar space $\Lambda_{H,ij} \subset L^2(\Gamma_{ij})$ if $\Gamma_{ij} \subset \Gamma_{SD} \cup \Gamma_{DD}$ or $\Lambda_{H,ij} \subset (L^2(\Gamma_{ij}))^d$ if $\Gamma_{ij} \subset \Gamma_{SS}$ is defined to weakly impose the continuity conditions for the discrete velocity across the non-matching grids. These mortar spaces consist of continuous or discontinuous piecewise polynomials of degree that may vary on the different types of interfaces. Globally, the finite element spaces are defined as

$$
V_h = \bigoplus_{i=1}^{N} V_{h,i}, \quad W_h = \bigoplus_{i=1}^{N} W_{h,i}, \quad \Lambda_H = \bigoplus_{i,j} \Lambda_{H,ij}.
$$

In the multiscale mortar mixed finite element discretization of (11)–(13) we seek $(\mathbf{u}_h, p_h, \lambda_H) \in V_h \times W_h \times \Lambda_H$, such that

$$
a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) + b_I(\mathbf{v}_h, \lambda_H) = (\mathbf{f}, \mathbf{v}_h), \quad \forall \mathbf{v}_h \in V_h, \tag{16}
$$

$$
b(\mathbf{u}_h, w_h) = -(q_D, w_h)_{\Omega_D}, \quad \forall w_h \in W_h, \tag{17}
$$

$$
b_I(\mathbf{u}_h, \mu_H) = 0, \quad \forall \mu_H \in \Lambda_H. \tag{18}
$$

The following convergence result for (16)–(18) has been shown in [7]:

Theorem 2.1. *Assume that the solution to (11)–(13) is sufficiently smooth, and let r_S and r_D be the polynomial degrees of the velocity spaces in Stokes and Darcy respectively, and let m_S, m_D, and m_{SD} be the polynomial degrees of the*

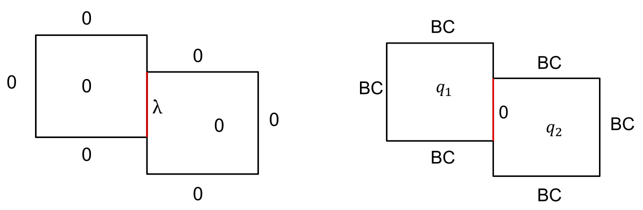

Fig. 1. Local problems for $(\mathbf{u}^*, p^*)$ (left) and $(\overline{\mathbf{u}}, \overline{p})$ (right).

mortar spaces on Γ_{SS}, Γ_{DD}, and Γ_{SD}, respectively. Then there exists a positive constant C independent of h and H such that

$$\|\mathbf{u} - \mathbf{u}_h\|_V + \|p - p_h\|_W \le C(h^{r_S} + h^{r_D+1} + H^{m_S+1/2} + H^{m_D+1/2} + H^{m_{SD}+1/2}).$$

2.4. Reduction to an interface problem

Following the algorithm in [9], we reduce the discretized problem (16)–(18) to an interface problem, which can be solved using a Krylov iterative solver. The original problem is split into two families of local problems on each Ω_i, one with zero source, zero outside boundary conditions, and specified interface value; the other with zero interface value, specified source, and specified outside boundary conditions. Correspondingly, the solution to (16)–(18) is decomposed into $\mathbf{u}_h = \mathbf{u}_h^* + \overline{\mathbf{u}}_h$, $p_h = p_h^* + \overline{p}_h$, see Fig. 1. On each Stokes subdomain, let $\lambda = (\lambda_n, \lambda_\tau)$, where λ_n and $\lambda_\tau = (\lambda_\tau^1, \dots, \lambda_\tau^{d-1})$ represent the normal stress and tangential stress on Γ_{SS}, respectively, Consider the set of Stokes subdomain problems with specified normal and tangential stress on the interfaces: find $(\mathbf{u}_{h,i}^*(\lambda), p_{h,i}^*(\lambda)) \in V_{h,i}/\ker a_i \times W_{h,i}$, $1 \le i \le N_S$, such that

$$a_i(\mathbf{u}_{h,i}^*(\lambda), \mathbf{v}_{h,i}) + b_i(\mathbf{v}_{h,i}, p_{h,i}^*(\lambda)) = -\langle \lambda_n, \mathbf{v}_{h,i} \cdot \mathbf{n}_i \rangle_{\partial\Omega_i \setminus \partial\Omega} - \sum_{l=1}^{d-1} \langle \lambda_\tau^l, \mathbf{v}_{h,i} \cdot \boldsymbol{\tau}_i^l \rangle_{\partial\Omega_i \cap \Gamma_{SS}},$$
$$\forall \mathbf{v}_i \in V_{h,i}/\ker a_i, \tag{19}$$
$$b_i(\mathbf{u}_{h,i}^*(\lambda), w_{h,i}) = 0, \quad \forall\, w_{h,i} \in W_{h,i}, \tag{20}$$

where $\{\boldsymbol{\tau}_i^l\}_{l=1}^{d-1}$ is an orthogonal set of unit vectors tangential to $\partial\Omega_i$ and the kernel space $\ker a_i := \{\mathbf{v} \in V_i : a_i(\mathbf{v}, \mathbf{v}) = 0\}$ consists of a subset of all rigid body motions depending on the types of boundary conditions on Ω_i. Some discussion on handling singular Stokes subdomain problems is given in the next subsection. The complementary set of local problems is to find $(\overline{\mathbf{u}}_{h,i}, \overline{p}_{h,i}) \in V_{h,i} \times W_{h,i}$ such that

$$a_i(\overline{\mathbf{u}}_{h,i}, \mathbf{v}_{h,i}) + b_i(\mathbf{v}_{h,i}, \overline{p}_{h,i}) = (\mathbf{f}_i, \mathbf{v}_{h,i})_{\Omega_i}, \quad \forall \mathbf{v}_{h,i} \in V_{h,i}/\ker a_i, \tag{21}$$
$$b_i(\overline{\mathbf{u}}_{h,i}, w_{h,i}) = 0, \quad \forall w_{h,i} \in W_{h,i}. \tag{22}$$

Similarly, on each Darcy domain Ω_i, $N_S + 1 \le i \le N$, the first set of local problems is to find $(\mathbf{u}_{h,i}^*(\lambda), p_{h,i}^*(\lambda)) \in V_{h,i} \times W_{h,i}$ with specified interface pressure λ such that

$$a_i(\mathbf{u}_{h,i}^*(\lambda), \mathbf{v}_{h,i}) + b_i(\mathbf{v}_{h,i}, p_{h,i}^*(\lambda)) = -\langle \lambda, \mathbf{v}_{h,i} \cdot \mathbf{n}_i \rangle_{\partial\Omega_i \setminus \partial\Omega}, \quad \forall \mathbf{v}_{h,i} \in V_{h,i}, \tag{23}$$
$$b_i(\mathbf{u}_{h,i}^*(\lambda), w_{h,i}) = 0, \quad \forall\, w_{h,i} \in W_{h,i}. \tag{24}$$

The corresponding complementary problem is to find $(\overline{\mathbf{u}}_{h,i}, \overline{p}_{h,i}) \in V_{h,i} \times W_{h,i}$ such that

$$a_i(\overline{\mathbf{u}}_{h,i}, \mathbf{v}_{h,i}) + b_i(\mathbf{v}_{h,i}, \overline{p}_{h,i}) = (\mathbf{f}_i, \mathbf{v}_{h,i})_{\Omega_i}, \quad \forall \mathbf{v}_{h,i} \in V_{h,i}, \tag{25}$$
$$b_i(\overline{\mathbf{u}}_{h,i}, w_{h,i}) = -(q_D, w_{h,i})_{\Omega_i}, \quad \forall w_{h,i} \in W_{h,i}. \tag{26}$$

Note that for the local Stokes problems (19)–(20), the boundary conditions on the interfaces Γ_{SS} are of Neumann type:

$$-(\mathbf{T}\mathbf{n}_i) \cdot \mathbf{n}_i = \lambda_n, \quad -(\mathbf{T}\mathbf{n}_i) \cdot \boldsymbol{\tau}_i^l = \lambda_\tau^l, \quad 1 \le l \le d-1, \quad 1 \le i \le N_S,$$

and on the interfaces Γ_{SD} are of Robin type:

$$-(\mathbf{T}\mathbf{n}_i) \cdot \mathbf{n}_i = \lambda_n, \quad -(\mathbf{T}\mathbf{n}_i) \cdot \boldsymbol{\tau}_i^l - \frac{\mu_S \alpha}{\sqrt{K_l}} \mathbf{u}_i \cdot \boldsymbol{\tau}_i^l = 0, \quad 1 \le l \le d-1, \ 1 \le i \le N_S.$$

For the local Darcy problems (23)–(24), the boundary conditions on the interfaces $\Gamma_{DD} \cup \Gamma_{SD}$ are of Dirichlet type:

$$p_i = \lambda, \quad N_S + 1 \le i \le N.$$

It is easy to verify that problem (16)–(18) is equivalent to the interface problem for $\lambda_H \in \Lambda_H$:

$$s_H(\lambda_H, \mu_H) := -b_I(\mathbf{u}_h^*(\lambda_H), \mu_H) = b_I(\overline{\mathbf{u}}_h, \mu_H), \quad \forall \mu_H \in \Lambda_H, \tag{27}$$

where the above equation follows from the interface condition (18) and the global solution can be recovered by

$$\mathbf{u}_h = \mathbf{u}_h^*(\lambda_H) + \overline{\mathbf{u}}_h, \quad p_h = p_h^*(\lambda_H) + \overline{p}_h.$$

Later it will be convenient to write

$$b_I(\mathbf{v}, \mu) = \sum_{i=1}^{N} b_I^i(\mathbf{v}_i, \mu),$$

where

$$b_I^i(\mathbf{v}_i, \mu) = \begin{cases} \langle \mu_n, \mathbf{v}_i \cdot \mathbf{n}_i \rangle_{\partial\Omega_i \setminus \partial\Omega} + \sum_{l=1}^{d-1} \langle \mu_\tau^l, \mathbf{v}_i \cdot \boldsymbol{\tau}_i^l \rangle_{\partial\Omega_i \cap \Gamma_{SS}}, & 1 \le i \le N_S, \\ \langle \mu_n, \mathbf{v}_i \cdot \mathbf{n}_i \rangle_{\partial\Omega_i \setminus \partial\Omega}, & N_s + 1 \le i \le N. \end{cases} \tag{28}$$

Analysis of the condition number of the reduced problem (27) is performed theoretically and numerically in [9].

Theorem 2.2. *Assuming $H = O(h)$, there exist positive constants C_1, C_2 independent of h and H, such that for all $\lambda \in \Lambda_H$,*

$$C_1 \frac{K_{min}^2}{K_{max}} (h \|\lambda\|_{\Gamma_{SS}}^2 + \|\lambda\|_{\Gamma_{DD} \cup \Gamma_{SD}}^2) \le s_H(\lambda, \lambda) \le C_2 \left(\|\lambda\|_{\Gamma_{SS}}^2 + \frac{K_{max}}{h} \|\lambda\|_{\Gamma_{DD} \cup \Gamma_{SD}}^2 \right), \tag{29}$$

where K_{min} and K_{max} are the minimal and maximal eigenvalues of the permeability $\mathbf{K}$, *respectively.*

2.5. Algebraic interpretation and FETI method

Another way to interpret the derivation of the interface problem (27) is from the algebraic form of the discretized problem (16)–(18),

$$\begin{pmatrix} A & B^T & C^T \\ B & 0 & 0 \\ C & 0 & 0 \end{pmatrix} \begin{pmatrix} u \\ p \\ \lambda \end{pmatrix} = \begin{pmatrix} f \\ q \\ 0 \end{pmatrix} \quad \Leftrightarrow \quad \begin{pmatrix} M & L^T \\ L & 0 \end{pmatrix} \begin{pmatrix} \xi \\ \lambda \end{pmatrix} = \begin{pmatrix} r \\ 0 \end{pmatrix}, \tag{30}$$

where $\xi = (u, p)^T$ is the vector of subdomain unknowns, $r = (f, q)^T$ is the vector of discrete right hand side functions in the coupled system,

$$M = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}, \text{ and } \quad L = \begin{pmatrix} C & 0 \end{pmatrix}.$$

Then by forming the Schur complement of (30), we obtain the matrix form of the interface problem (27),

$$LM^{-1}L^T\lambda = LM^{-1}r. \tag{31}$$

Theorem 2.2 implies that the matrix on the left in (31) is symmetric and positive definite, and therefore the problem can be solved using a Krylov iterative method such as the conjugate gradient (CG) method. At each iteration we need to evaluate the action of M^{-1}. Since M^{-1} is block-diagonal,

$$M^{-1} = \begin{pmatrix} M_1^{-1} & & \\ & \ddots & \\ & & M_N^{-1} \end{pmatrix},$$

this action requires solving N local subdomain problems, which is done in parallel. One issue that arises in these solves is the occurrence of floating subdomains, that is, if a Stokes subdomain Ω_i is surrounded entirely by other Stokes domains, local full Neumann problems occur and the local matrix M_i is singular. In other words, the kernel space $\ker a_i$ in (19) and (21) becomes non-trivial. To handle this issue, we follow the algorithm introduced in [9] which is based on the one-level FETI method proposed by Farhat and Roux [16]. Let $\overline{r}_i$ be orthogonal projection of r onto $\ker a_i$. The algebraic system (30) can be written as

$$M\xi + L^T\lambda = r,$$

subject to the constraint

$$L\xi = 0.$$

Let R be a matrix whose columns form or contain a basis for $\ker(M)$ and define an operator

$$G = LR.$$

We can split $\lambda = \lambda_0 + \lambda_1$, where

$$\lambda_0 = G(G^TG)^{-1}R^T\overline{r}_S, \quad \text{and} \quad \lambda_1 \in ker(G^T).$$

Computing $(G^TG)^{-1}$ requires solving a coarse problem, which can be reduced to solving a local problem of size $k_i \times k_i$ in each subdomain Ω_i, with $k_i = \dim(\ker(M_i))$. Let the operator P be the orthogonal projector onto $\ker(G^T)$:

$$P = I - G(G^TG)^{-1}G^T.$$

Applying P^T on both sides of Eq. (31) and letting $\lambda_1 = P\nu$, we derive the projected interface problem

$$P^TLM^{-1}L^TP\nu = P^TL(M^{-1}(\overline{r} - L^T\lambda_0) + M^{-1}(r - \overline{r})). \tag{32}$$

Note that in (32) there are three actions of M^{-1}, i.e., three local subdomain solves. The two solves on the right hand side both satisfy $\overline{r} - L^T\lambda_0 \perp \ker(M)$ and $r - \overline{r} \perp \ker(M)$, so in the Stokes region they are compatible Neumann solves. On the other hand, for any $P\nu \in \ker(G^T)$, one can verify that $L^TP\nu \perp \ker(M)$, so the local solve on the left hand side is also compatible. Moreover, the matrix $P^TLM^{-1}L^TP$ is symmetric and positive definite in the space $\ker(G^T)$ and therefore (32) can be solved with the CG method.

Since M_i may be singular, one must use a pseudoinverse M_i^+. The traditional FETI method uses the Moore–Penrose pseudoinverse, which could be computationally expensive. Since our method involves solving only compatible problems, we can avoid this and in practice we use $M_i^+ := (M_i + \sqrt{\varepsilon}D_i)^{-1}$ to replace the evaluation of M_i^{-1}, where ε is the machine precision and D_i is the velocity mass matrix.

2.6. *Iterative solution of the interface problem*

For simplicity, we introduce a Steklov–Poincaré type operator $S_H : \Lambda_H \to \Lambda_H$ such that

$$\langle S_H\lambda_H, \mu_H\rangle_\Gamma = s_H(\lambda_H, \mu_H), \quad \forall \lambda_H, \mu_H \in \Lambda_H.$$

We note that $S_H\lambda_H = \sum_{i=1}^N S_{H,i}\lambda_{H,i}$, where $S_{H,i} : \Lambda_{H,i} \to \Lambda_{H,i}$ is defined by

$$\langle S_{H,i}\lambda_{H,i}, \mu_{H,i}\rangle_{\Gamma_i} = -b_I^i(\mathbf{u}_{h,i}^*(\lambda_{H,i}), \mu_{H,i}), \quad \forall \lambda_{H,i}, \mu_{H,i} \in \Lambda_{H,i}.$$

Let $\mathcal{L}_{h,i} : \Lambda_{H,i} \to V_{h,i}|_{\Gamma_i}$ on $\Gamma_i \cap \Gamma_{SS}$ or $\mathcal{L}_{h,i} : \Lambda_{H,i} \to V_{h,i} \cdot \mathbf{n}_i|_{\Gamma_i}$ on $\Gamma_i \cap (\Gamma_{SD} \cup \Gamma_{DD})$ be the L^2-orthogonal projection operator from the mortar space onto the (normal) trace of the velocity space in Ω_i. Correspondingly, let $\mathcal{L}_{h,i}^T : V_{h,i}|_{\Gamma_i} \to \Lambda_{H,i}$ or $\mathcal{L}_{h,i}^T : V_{h,i} \cdot \mathbf{n}_i|_{\Gamma_i} \to \Lambda_{H,i}$ be the L^2-orthogonal projection operator from the (normal) trace of the velocity space onto the mortar space. Using the definition (28) of $b_I^i(\cdot,\cdot)$, we have that

$$S_{H,i}\lambda_{H,i} = -\mathcal{L}_{h,i}^T \begin{pmatrix} \mathbf{u}_{h,i}^*(\lambda_{H,i}) \cdot \mathbf{n}_i \\ \mathbf{u}_{h,i}^*(\lambda_{H,i}) \cdot \boldsymbol{\tau}_i^l \end{pmatrix} \quad \text{on } \Gamma_i \cap \Gamma_{SS},$$

$$S_{H,i}\lambda_{H,i} = -\mathcal{L}_{h,i}^T\mathbf{u}_{h,i}^*(\lambda_{H,i}) \cdot \mathbf{n}_i \quad \text{on } \Gamma_i \cap (\Gamma_{SD} \cup \Gamma_{DD}).$$

Problem (27) can be rewritten as

$$S_H \lambda_H = g_H, \tag{33}$$

where $g_H \in \Lambda_H$ is defined by $\langle g_H, \mu_H \rangle_\Gamma = b_I(\overline{\mathbf{u}}_h, \mu_H), \quad \forall \mu_H \in \Lambda_H$. In the general case where floating Stokes subdomains are allowed, we need to solve the corresponding projected interface problem:

$$P^T S_H P \lambda_H = \tilde{g}_H, \tag{34}$$

where $\tilde{g}_H$ denotes the right hand side of Eq. (32).

The CG method is applied to solve (34), where on each iteration an operator action $P^T S_H P$ on data $\lambda_H \in \Lambda_H$ is computed as follows:

Algorithm 1. original CG implementation.

1. Project $\lambda_{H,i}$ onto $\ker(G^T)$: $\lambda_{H,i} \to P\lambda_{H,i}$.
2. Project $P\lambda_{H,i}$ onto the local subdomain boundary space: $\gamma_{h,i} = \mathcal{L}_{h,i} P\lambda_{H,i}$.
3. In the Stokes region, solve subdomain problems (19)–(20) with Neumann boundary data $\gamma_{h,i}$. In the Darcy region, solve subdomain problems (23)–(24) with Dirichlet boundary data $\gamma_{h,i}$. Denote the solutions by $(\mathbf{u}^*_{h,i}(\gamma_{h,i}), p^*_{h,i}(\gamma_{h,i}))$.
4. Project the resulting velocity in Stokes or resulting flux in Darcy back to the mortar space, i.e.,

$$S_{H,i} P\lambda_{H,i} = -\mathcal{L}^T_{h,i} \begin{pmatrix} \mathbf{u}^*_{h,i}(\gamma_{h,i}) \cdot \mathbf{n}_i \\ \mathbf{u}^*_{h,i}(\gamma_{h,i}) \cdot \boldsymbol{\tau}^l_i \end{pmatrix} \quad \text{on } \Gamma_i \cap \Gamma_{SS},$$

or

$$S_{H,i} P\lambda_{H,i} = -\mathcal{L}^T_{h,i} \mathbf{u}^*_{h,i}(\gamma_{h,i}) \cdot \mathbf{n}_i \quad \text{on } \Gamma_i \cap (\Gamma_{SD} \cup \Gamma_{DD}),$$

and compute the jump across all subdomain interfaces:

$$S_H P\lambda_H = \sum_{i=1}^{N} S_{H,i} P\lambda_{H,i}.$$

5. Apply $P\ (=P^T)$ to project the jump onto $\ker(G^T)$: $S_H P\lambda_H \to P^T S_H P\lambda_H$.

3. Multiscale flux basis implementation

Notice that the dominant computational costs in the above algorithm comes from the subdomain solves in step 3. Thus, for the original implementation of the MMMFEM, the total number of solves in each subdomain is approximately equal to the number of CG iterations. Even though all subdomain solves are computed in parallel, this can still be very costly when the condition number of the interface problem is large due to a highly refined mesh.

In this section we introduce the notion of a **multiscale flux basis**, following the idea from [18]. Our primary motivation is to improve the efficiency of the solution of the interface problem (34). This approach aims to eliminate the dependency between the total number of solves and the number of CG iterations. In order to achieve this, we precompute and store the flux or velocity subdomain responses, called multiscale flux basis, associated with each coarse scale mortar degree of freedom on every Darcy or Stokes subdomain. This requires solving a fixed number of subdomain solves. Then, the solution of subdomain problems on each CG iteration is replaced by linear combinations of the multiscale flux basis functions. As a result, the total number of solves per subdomain is independent of the number of CG iterations and thus insensitive to refining the subdomain grids.

In subdomain Ω_i, $i = 1, 2, \ldots, N$, let $\mathcal{N}_i$ be the number of degrees of freedom in the mortar space $\Lambda_{H,i}$ on Γ_i, and let $\left\{\xi^{\{k\}}_{H,i}\right\}_{k=1}^{\mathcal{N}_i}$ be a basis of $\Lambda_{H,i}$. Any $\lambda_{H,i} \in \Lambda_{H,i}$ can be expressed as $\lambda_{H,i} = \sum_{k=1}^{\mathcal{N}_i} \alpha^{\{k\}}_i \xi^{\{k\}}_{H,i}$. Define the multiscale basis $\left\{\phi^{\{k\}}_{H,i}\right\}_{k=1}^{\mathcal{N}_i} \subset \Lambda_{H,i}$ as

$$\phi^{\{k\}}_{H,i} = S_{H,i} \xi^{\{k\}}_{H,i}, \quad k = 1, \ldots, \mathcal{N}_i.$$

The action of the interface operator on any mortar function can then be computed as

$$S_{H,i}\lambda_{H,i} = S_{H,i}\left(\sum_{k=1}^{\mathcal{N}_i}\alpha_i^{\{k\}}\xi_{H,i}^{\{k\}}\right) = \sum_{k=1}^{\mathcal{N}_i}\alpha_i^{\{k\}}S_{H,i}\xi_{H,i}^{\{k\}} = \sum_{k=1}^{\mathcal{N}_i}\alpha_i^{\{k\}}\phi_{H,i}^{\{k\}},$$

which is simply a linear combination of the multiscale basis.

The algorithm for computing the multiscale basis $\left\{\phi_{H,i}^{\{k\}}\right\}_{k=1}^{\mathcal{N}_i} \subset \Lambda_{H,i}$ on a subdomain Ω_i is as follows:

Algorithm 2. computation of multiscale flux basis.

For $k = 1, \ldots, \mathcal{N}_i$

1. Project the mortar basis function onto the local subdomain boundary space:

$$\eta_{h,i}^{\{k\}} = \mathcal{L}_{h,i}\xi_{H,i}^{\{k\}}.$$

2. If Ω_i is a Stokes subdomain, solve the subdomain problem (19)–(20) with Neumann boundary data $\eta_{h,i}^{\{k\}}$. If it is Darcy, solve the subdomain problem (23)–(24) with Dirichlet boundary data $\eta_{h,i}^{\{k\}}$. Denote the solutions by $(\mathbf{u}_{h,i}^*(\eta_{h,i}^{\{k\}}), p_{h,i}^*(\eta_{h,i}^{\{k\}}))$.
3. Project the resulting velocity in Stokes or resulting flux in Darcy back to mortar space, which gives the multiscale flux basis:

$$\phi_{H,i}^{\{k\}} = -\mathcal{L}_{h,i}^T\begin{pmatrix}\mathbf{u}_{h,i}^*(\eta_{h,i}^{\{k\}})\cdot\mathbf{n}_i\\ \mathbf{u}_{h,i}^*(\eta_{h,i}^{\{k\}})\cdot\boldsymbol{\tau}_i^l\end{pmatrix} \quad \text{on } \Gamma_i\cap\Gamma_{SS},$$

or

$$\phi_{H,i}^{\{k\}} = -\mathcal{L}_{h,i}^T\mathbf{u}_{h,i}^*(\eta_{h,i}^{\{k\}})\cdot\mathbf{n}_i \quad \text{on } \Gamma_i\cap(\Gamma_{SD}\cup\Gamma_{DD}).$$

The multiscale flux basis can be used in the conjugate gradient method for solving (34). In every iteration, the operator action P^TS_HP on any $\lambda_H \in \Lambda_H$ is computed with the following steps:

Algorithm 3. CG with multiscale basis.

1. Project $\lambda_{H,i}$ onto $\ker(G^T)$: $\lambda_{H,i} \to P\lambda_{H,i}$.
2. Denote by $\left\{c_i^{\{k\}}\right\}_{k=1}^{\mathcal{N}_i}$ the expansion coefficients of $P\lambda_{H,i}$ in the mortar basis:

$$P\lambda_{H,i} = \sum_{k=1}^{\mathcal{N}_i}c_i^{\{k\}}\xi_{H,i}^{\{k\}}.$$

3. Use a linear combination of the multiscale flux basis to compute the resulting velocity (if Ω_i is Stokes) or flux (if Ω_i is Darcy):

$$S_{H,i}P\lambda_{H,i} = S_{H,i}\left(\sum_{k=1}^{\mathcal{N}_i}c_i^{\{k\}}\xi_{H,i}^{\{k\}}\right) = \sum_{k=1}^{\mathcal{N}_i}c_i^{\{k\}}S_{H,i}\xi_{H,i}^{\{k\}} = \sum_{k=1}^{\mathcal{N}_i}c_i^{\{k\}}\phi_{H,i}^{\{k\}}$$

and compute the jump across all subdomain interfaces:

$$S_HP\lambda_H = \sum_{i=1}^{N}S_{H,i}P\lambda_{H,i}.$$

4. Apply P $(=P^T)$ to project the jump onto $\ker(G^T)$: $S_HP\lambda_H \to P^TS_HP\lambda_H$.

Note that in the computation of the multiscale flux basis in Step 1 of Algorithm 2, the right hand side for the local Neumann solve in Stokes has not been projected to $\ker(G^T)$. However, in Algorithm 3, the multiscale basis is used only for computing the action of $S_{H,i}$ on $P\lambda_{H,i}$, which is a well-posed local Neumann solve. As a result, the computation of $S_{H,i}P\lambda_{H,i}$ via the linear combination of the multiscale flux basis in Algorithm 3 gives an equivalent

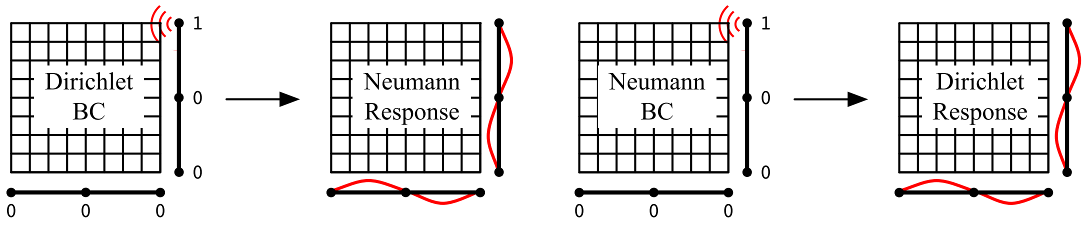

Fig. 2. Computation of the multiscale flux basis in Darcy (left) and Stokes (right).

result to computing $S_{H,i}P\lambda_{H,i}$ directly by solving a well posed local Neumann problem with data $P\lambda_{H,i}$. In addition, since P is linear, applying it on the computed jump in the last step is equivalent to projecting the multiscale flux basis to $\ker(G^T)$.

Fig. 2(left) illustrates one pressure mortar basis function in a Darcy domain and the computed flux response, which is the corresponding multiscale flux basis function. Similarly, Fig. 2(right) shows one normal stress mortar basis function in a Stokes domain and the computed velocity response. Comparing the new algorithm with the original MMMFEM, we notice that there are no subdomain solves in the CG iterations. The dominant cost now shifts to the computation of a multiscale flux basis, which depends on the number of mortar degrees of freedom. Since the mortar space is on the coarse scale, this cost is relatively small and independent of the fine grid. Furthermore, unlike other multiscale methods such as the variational multiscale method or the multiscale finite element method, where the fine scale solution on the entire coarse element needs to be stored, our method requires storing only coarse scale interface data—the flux or velocity response. Therefore the extra storage cost is significantly smaller compared to existing methods.

Even though the multiscale flux basis algorithm does not require subdomain solves in the CG iterations, the number of interface iterations is not insignificant. Some of the cost is due to the time needed to compute the orthogonal projections and linear combinations, but the more significant cost comes from inter-processor communication. It is therefore possible to reduce the overall computational cost by applying a preconditioner for the solution of the interface problem (34) in order to decrease its condition number, which results in decreasing the number of interface iterations. For the performance comparison in the numerical examples in the next section, we employ in the Darcy region the **balancing preconditioner** introduced in [29–31]. This preconditioner involves solving Neumann subdomain problems and a coarse problem to exchange global information. It is very efficient and exhibits condition number that grows very mildly with respect to h and H. We do not apply a preconditioner in the Stokes region, although due the coarse solver, the condition number in the Stokes region is insensitive to the subdomain size [9].

Let M_D be the Darcy component of M, which is the block-diagonal matrix with blocks M_i, $i = N_S + 1, \ldots, N$, and let Λ_H^D be the restriction of Λ_H to $\Gamma_{DD} \cup \Gamma_{SD}$. In algebraic form, the balancing preconditioner can be expressed as

$$B_{bal}^{-1} = \sum_{i=N_S+1}^{N} LM_i^{+}L^T,$$

where M_i^+ is the Moore–Penrose pseudo-inverse of M_i. The detailed algorithm is shown below. Define a partition of unity D_i such that

$$\sum_{i=N_S+1}^{N} D_i\lambda = \lambda, \quad \forall\lambda \in \Lambda_H^D,$$

and define a coarse space

$$Z = \left\{\lambda \in \Lambda_H^D : \lambda = \sum_{i=N_S+1}^{N} D_i\psi_i,\ \psi_i \in Z_i\right\},$$

where Z_i are spaces of constant vectors such that $\ker(M_i) \subseteq Z_i$, $i = N_S + 1, \ldots, N$. In every CG iteration, given residual $r \in \Lambda_H^D$, compute $B_{bal}^{-1}r$ as follows:

Algorithm 4. balancing preconditioner.

1. Solve a coarse problem: find $\tilde{r} \in Z$ such that
$$a_i(\tilde{r}, \mu) = \langle r, \mu \rangle, \quad \forall \mu \in Z,$$
and balance the residual:
$$r^{bal} = r - M_D \tilde{r}. \tag{35}$$
2. Distribute r^{bal} to subdomains:
$$r_i = D_i^T r^{bal}.$$
3. Solve local Neumann problem for $\lambda_i \in \Lambda_{H,i}$:
$$M_i \lambda_i = r_i \tag{36}$$
4. Average the local solutions:
$$\lambda = \sum_{i=N_S+1}^{N} D_i \lambda_i.$$
5. Solve a coarse problem for $\tilde{\lambda} \in Z$:
$$a_i(\tilde{\lambda}, \mu) = \langle r^{bal}, \mu \rangle - a_i(\lambda, \mu), \quad \forall \mu \in Z,$$
and update the local solutions:
$$B_{bal}^{-1} r = \lambda + \tilde{\lambda} + \tilde{r}.$$

In the next section we study numerically the efficiency of the multiscale flux basis implementation by comparing four different methods based on the above algorithms:

- Method 1: the original MMMFEM with no preconditioner, Algorithm 1,
- Method 2: the original MMMFEM with balancing preconditioner, Algorithms 1 and 4,
- Method 3: multiscale flux basis implementation of the MMMFEM with no preconditioner, Algorithms 2 and 3,
- Method 4: multiscale flux basis implementation of the MMMFEM with balancing preconditioner, Algorithms 2, 3, and 4.

The maximal number of local solves per subdomain for each method is given in Table 1, where N_{iter} is the total number of CG iterations. The number of solves in Method 1 in each subdomain is $N_{iter} + 3$, since there is one solve per CG iteration and 3 extra solves for setting up the right hand side of (32) in Stokes and the recovery of the solution $(\mathbf{u}, p)$. In Method 2, there are two extra subdomain solves in each CG iteration in Darcy for the balancing preconditioner, one Dirichlet solve in (35) and one Neumann solve in (36), and at most 10 extra solves for the setup of the balancing preconditioner [31]. In Method 3, the maximal number of solves is given by $\max\{\mathcal{N}_i\}_{i=1}^{N} + 3$, which includes one solve in the computation of each multiscale flux basis, plus 3 extra solves for setting up the right hand side of (32) in Stokes and the recovery of the solution $(\mathbf{u}, p)$. In Method 4, the maximal number of solves in Stokes subdomains is the same as in Method 3, which is $\max\{\mathcal{N}_i\}_{i=1}^{N_S} + 3$. In Darcy subdomains, two different sets of multiscale basis are computed for both the Dirichlet solves in each CG iteration and the Neumann solves (36) in the balancing preconditioner, and these solves are replaced by a linear combination of the corresponding multiscale basis. Therefore, the maximal number of solves in the Darcy region is $2\max\{\mathcal{N}_i\}_{i=N_S+1}^{N} + 8$. We note that in the original implementation, Method 1 and Method 2, the number of subdomain solves is proportional to the number of CG iterations, while in the multiscale flux basis Method 3 and Method 4, the number of solves depends only on the number of local mortar degrees of freedom. Furthermore, the balancing preconditioner significantly reduces the number of CG iterations, which results in reduced computational time. In the examples below we compare all four methods and identify the most efficient method in terms of maximal number of solves per subdomain and computational time. We emphasize the comparison between the unpreconditioned Methods 1 and 3, as well as the preconditioned Methods 2 and 4 for better demonstration of the multiscale flux implementation.

Table 1
Maximal number of solves per subdomain for each method.

Method 1	$N_{iter} + 3$
Method 2	$3N_{iter} + 10$
Method 3	$\max\{\mathcal{N}_i\}_{i=1}^{N} + 3$
Method 4	$\max\{\max\{\mathcal{N}_i\}_{i=1}^{N_S} + 3,\ 2\max\{\mathcal{N}_i\}_{i=N_S+1}^{N} + 8\}$

Remark 3.1. In the numerical experiments we report both the number of subdomain solves and the CPU times. While the former is accepted as an objective measure of the computational efficiency of domain decomposition methods, we include the latter in order to provide a more complete picture of the total cost. We note that CPU runtime is highly machine dependent. In particular, interprocessor communication cost may play a significant role. As a result, the gains we observe in CPU time are smaller relative to the gains in number of solves. Communication cost can be reduced by exploring redundancy and shared multi-core memory architectures, which is beyond the scope of the paper.

4. Numerical experiments

In this section we present three numerical tests to illustrate the efficiency of the multiscale flux basis implementation by comparing the maximal number of subdomain solves and total runtime for Methods 1–4. In all examples, the lowest order Taylor–Hood triangular finite elements are used in Stokes and the lowest order Raviart–Thomas rectangular finite elements are used in Darcy. Discontinuous piecewise linear mortar finite elements are used for all mortar spaces on subdomain interfaces. We take $\mathbf{T}(\mathbf{u}_S, p_s) = -p_s\mathbf{I}+2\mu_S\nabla\mathbf{u}_S$, $\mu_S = \mu_D = \mu$ and $\mathbf{K} = K\mathbf{I}$, where K is a uniformly positive scalar function. All four methods produce the same solution, within the relative convergence tolerance 10^{-6}. We have previously performed extensive verification studies of the code, including testing the convergence of the numerical solution to the true solution as the grids are refined, see [9]. The results indicate convergence of order predicted by Theorem 2.1. One of the cases tested was the smooth solution in Example 1 below. The numerical tests presented in this paper are run on a parallel cluster of Intel Xeon CPU E5-2650 v3 @ 2.30 GHz processors with 192 GB RAM. The problems are solved in parallel such that each subdomain is assigned to one core.

4.1. Example 1: Regular shape domain with smooth solution

In this example, the domain is the unit square with $\Omega_S = (0, 1) \times (0.5, 1)$ and $\Omega_D = (0, 1) \times (0, 0.5)$. The problem has a given true solution such that

$$\mathbf{u}_S = \begin{bmatrix} (2-x)(1.5-y)(y-\xi) \\ -\dfrac{y^3}{3} + \dfrac{y^2}{2}(\xi+1.5) - 1.5\xi y - 0.5 + \sin(\omega x) \end{bmatrix},$$

$$\mathbf{u}_D = \begin{bmatrix} \omega\cos(\omega x)y \\ \chi(y+0.5) + \sin(\omega x) \end{bmatrix},$$

$$p_S = -\frac{\sin(\omega x)+\chi}{2K} + \mu(0.5-\xi) + \cos(\pi y),$$

$$p_D = -\frac{\chi}{K}\frac{(y+0.5)^2}{2} - \frac{\sin(\omega x)y}{K},$$

where

$$\mu = 0.1,\ K = 1,\ \alpha = 0.5,\ G = \frac{\sqrt{\mu K}}{\alpha},\ \xi = \frac{1-G}{2(1+G)},\ \chi = \frac{-30\xi - 17}{48},\ \text{and } \omega = 6.0.$$

The right hand sides $\mathbf{f}_S$, $\mathbf{f}_D$ and q_D in the Stokes–Darcy problem are computed using the given exact solution. The problem is solved by the four methods using four different levels of domain decomposition: 2×2, 4×4, 6×6, 8×8 and 10×10 subdomains. In any level, each subdomain is discretized by a 10×10 or 4×4 local mesh in a "checkerboard" manner, i.e., no neighboring subdomains have the same mesh. The mortar mesh on each subdomain interface is 2×1. The outside boundary conditions are given as follows: for Darcy, the left boundary is of Neumann type and the bottom

Table 2
Example 1 test results.

Subdomains	N_{iter}	Solves	Time (s)	N_{iter}	Solves	Time (s)
	Method 1			Method 2		
$2 \times 2 = 4$	38	41	0.587	22	74	0.607
$4 \times 4 = 16$	121	124	1.000	53	169	0.876
$6 \times 6 = 36$	182	185	1.566	78	244	1.341
$8 \times 8 = 64$	230	233	3.261	94	292	2.523
$10 \times 10 = 100$	280	283	7.352	98	304	4.764
	Method 3			Method 4		
$2 \times 2 = 4$	38	**27**	**0.579**	22	38	0.583
$4 \times 4 = 16$	121	**59**	0.808	53	72	**0.695**
$6 \times 6 = 36$	182	**67**	1.245	78	72	**1.151**
$8 \times 8 = 64$	230	**67**	2.328	94	72	**2.102**
$10 \times 10 = 100$	280	**67**	4.170	98	72	**3.392**

Bold denotes the smallest number of subdomain solves and the fastest run.

Table 3
Example 1 condition numbers.

Subdomains	Method 1, 3			Method 2, 4		
	eig.min.	eig.max.	cond.num.	eig.min.	eig.max.	cond.num.
$2 \times 2 = 4$	0.349	36.9	105.7	0.409	10.730	26.2
$4 \times 4 = 16$	0.134	90.0	670.8	0.134	30.409	226.3
$6 \times 6 = 36$	0.082	134.7	1635.4	0.082	32.135	393.1
$8 \times 8 = 64$	0.062	179.6	2881.6	0.061	32.108	527.1
$10 \times 10 = 100$	0.050	224.5	4502.6	0.050	31.973	644.3

and right boundaries are of Dirichlet type; for Stokes, the left boundary is of Dirichlet type and the top and right boundaries are Neumann type. Under this setting, all Stokes subdomains except the leftmost column are floating, with $\ker M_i = \text{span}\{(0, 1)\}$ if the domain has Stokes–Darcy interface on the bottom and $\ker M_i = \text{sp}\{(1, 0), (0, 1)\}$ otherwise. The test results for Example 1 are shown in Tables 2 and 3. In particular, number of CG iterations, maximal number of subdomain solves, and maximal CPU time per subdomain are reported in Table 2, indicating the highest workload of all CPUs in each parallel run. We note that the numbers match Table 1. The minimal and maximal eigenvalues and condition number are reported in Table 3. The computed solution with $4 \times 4 = 16$ subdomains is shown in Fig. 3.

Since the mesh size h decreases as we increase the number of subdomains, the interface condition number increases. This is evident from Table 3. Methods 2 and 4, which employ the balancing preconditioner in the Darcy region, control the maximal eigenvalue and result in a more modest increase in the condition number, compared to the unpreconditioned Methods 1 and 3. This has an effect on the number of CG iterations reported in Table 2, with a much sharper increase in N_{iter} for Methods 1 and 3. Regarding the number of subdomain solves, we observe that the multiscale flux basis Methods 3 and 4 have reduced number of solves compared to Methods 1 and 2. Method 3 has the smallest number of subdomain solves in all cases (marked in bold). Also Method 3 and 4 only involve a fixed number of solves for precomputing the multiscale flux basis, which depend on the local number degrees of freedom of the mortar space. Note that the number of solves for both methods stay fixed from the 36 domains case to the 100 domains case, and will not change if more subdomains are added, since there are no more new mortar degrees of freedom created per subdomain. In Method 4, the balancing preconditioner costs one extra local Neumann solve per degree of freedom in the precomputation step for each Darcy subdomain. However, the significantly reduced number of iterations results in reduced communication cost and savings in the overall computational time. We mark in bold the fastest run for each case in Table 2, which is achieved by either Method 3 or 4 in all four levels. It is not easy to make a fair comparison between Method 3 and 4, since the former reduces the number of solves, while the latter reduces the number of iterations at the cost of extra solve per iteration. Nevertheless, both methods are more efficient than Methods 1 and 2.

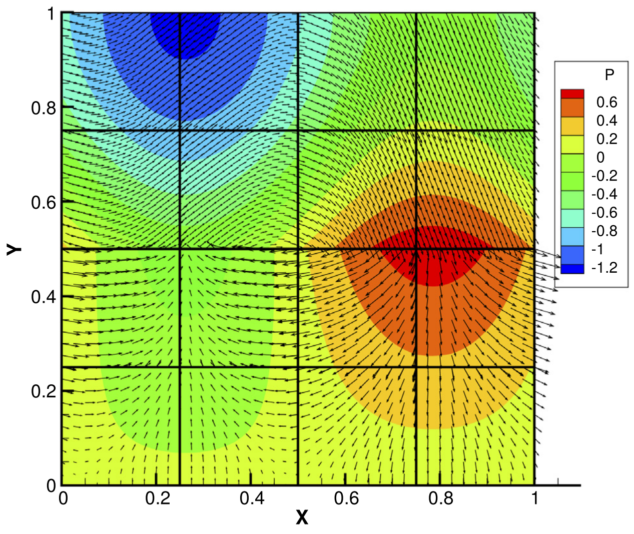

Fig. 3. Example 1, 16 subdomains, solution.

4.2. *Example 2: Irregular shape domain with heterogeneous permeability*

The second example is a Stokes–Darcy problem with heterogeneous permeability field on an irregularly shaped domain. The domain is roughly contained in the $[0, 2] \times [0, 1]$ rectangular region, see Fig. 5, with the Stokes region in the top half and the Darcy region at the bottom. Following [38,8], we handle the irregular geometry by the multipoint flux mixed finite element method for the pressure in Darcy and standard conforming elements in Stokes. We also impose the mortar conditions on curved interfaces by mapping the physical grids to reference grids with flat interfaces, see [8].

The heterogeneous permeability is given by a single realization of a stochastic permeability field K, which can be generated by a sum of the local Karhunen–Loève (KL) expansion [28]. Let $Y = \ln(K)$ and let Y' be defined by

$$Y'(\mathbf{x}, \omega) := Y(\mathbf{x}, \omega) - E[Y](\mathbf{x}),$$

where $E[Y](\mathbf{x})$ stands for the expectation function. Denote the series expansion of the covariance function of Y as

$$C_Y(\mathbf{x}, \bar{\mathbf{x}}) = \sum_{j=1}^{\infty} \lambda_j f_j(\mathbf{x}) f_j(\bar{\mathbf{x}}).$$

Then the KL-expansion of Y' with N_{term} terms is given as

$$Y'(\mathbf{x}, \omega) \approx \sum_{j=1}^{N_{term}} \xi_j(\omega)\sqrt{\lambda_j} f_j(\mathbf{x}),$$

where ξ_j are normal random variables with zero mean and unit variance. In this example, the covariance function is specified as

$$C_Y(\mathbf{x}, \bar{\mathbf{x}}) = \sigma_Y^2 \exp\left[\frac{-|x_1 - \bar{x}_1|}{\eta_1} - \frac{|x_2 - \bar{x}_2|}{\eta_2}\right]$$

with $\sigma_Y = 2.1$, $\eta_1 = 0.1$, $\eta_2 = 0.05$ and $N_{term} = 400$. For the mean value we set $E[Y](\mathbf{x}) = 1.0$. A plot for the permeability realization is shown in Fig. 4.

All methods are tested on four different levels of domain decompositions: 4×2, 8×4, 12×6 and 16×8. The local meshes are 18×15, 15×12 "checkerboard" in Darcy; and 12×15, 9×12 "checkerboard" in Stokes. The mortar mesh is 4×1 on each subdomain interface. The outside boundary conditions are given as follows. In Darcy, no flow condition is specified on the left and right boundaries, with Dirichlet condition at the bottom. In Stokes, we specify inflow condition on the left boundary, zero flow condition on the right, and zero stress condition on the top.

Table 4
Example 2 test results.

Subdomains	N_{iter}	Solves	Time (s)	N_{iter}	Solves	Time (s)
	Method 1			Method 2		
$4 \times 2 = 8$	220	223	1.165	66	207	0.979
$8 \times 4 = 32$	1069	1072	6.312	194	592	2.303
$12 \times 6 = 72$	1525	1528	25.791	239	727	7.002
$16 \times 8 = 128$	2375	2378	90.088	301	913	16.261
	Method 3			Method 4		
$4 \times 2 = 8$	220	**83**	**0.850**	67	103	0.901
$8 \times 4 = 32$	1069	**115**	2.260	194	136	**1.733**
$12 \times 6 = 72$	1525	**131**	6.771	239	136	**3.754**
$16 \times 8 = 128$	2375	**131**	24.407	301	136	**9.828**

Bold—denotes the smallest number of subdomain solves and the fastest run.

Table 5
Example 2 condition numbers.

Subdomains	Method 1, 3			Method 2, 4		
	eig.min.	eig.max.	cond.num.	eig.min.	eig.max.	cond.num.
$4 \times 2 = 8$	0.114	416.2	3637.2	0.112	26.368	236.3
$8 \times 4 = 32$	0.025	2228.8	88015.3	0.023	44.044	1904.4
$12 \times 6 = 72$	0.017	2371.0	143620.6	0.016	38.682	2391.8
$16 \times 8 = 128$	0.013	4444.5	351727.1	0.012	39.937	3325.1

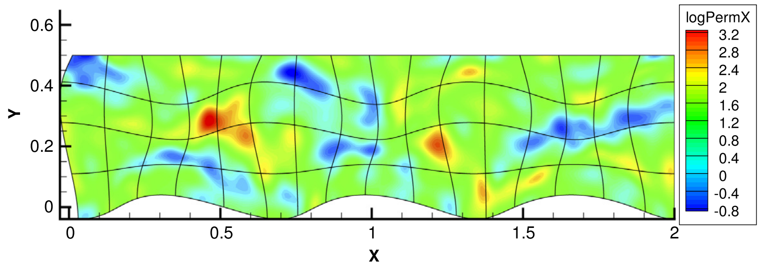

Fig. 4. Example 2, 128 subdomains, permeability.

Table 4 shows the results using Methods 1–4, and Table 5 shows the corresponding eigenvalues and condition numbers. Due to the size of K_{max} and a small h as the number of subdomain grows, the problem has a huge condition number and requires thousands of iterations for CG to converge in original MMMFEM implementation. With the multiscale flux basis, the number of solves in Method 3 and Method 4 is an order of magnitude smaller than Method 1 and Method 2, respectively, since it is independent of N_{iter}. We also observe that the communication cost plays a role in the runtime. As a result, even though Method 3 has the smallest number of solves, it is slower than Method 4 in the last three cases. We note that the balancing preconditioner is very effective in this situation. The maximal eigenvalue is controlled effectively and so is the number of CG iterations, as seen in Table 5. The gain in runtime due to the balancing preconditioner becomes more significant when increasing the number of subdomain.

4.3. Example 3: Adaptive mesh in Darcy

The third test case illustrates how the multiscale flux basis method reduces the computational cost when adaptive mesh refinement is used in the Darcy region. In this case, the permeability field K in the Darcy region is also generated from a single realization of a stochastic field using the KL expansion and it is highly heterogeneous. The KL parameters are correlation lengths $\eta_1 = 0.25$ and $\eta_2 = 0.125$, mean value $E[Y](\mathbf{x}) = 2.0$, variance $\sigma_Y = 2.1$ and $N_{term} = 400$. The generated permeability field is shown in Fig. 6. The domain is the unit square with the Stokes and Darcy regions in the top and bottom half, respectively. The boundary conditions for Darcy are no flow condition on

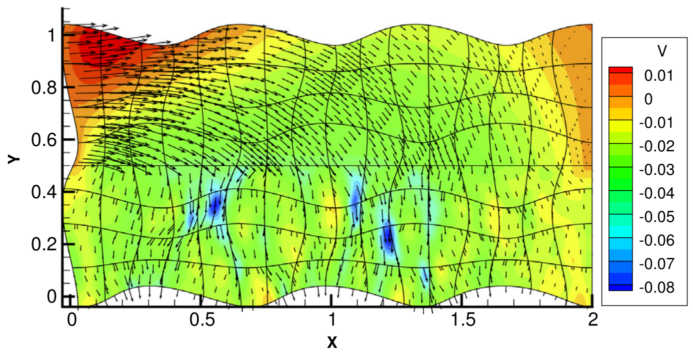

Fig. 5. Example 2, 128 subdomains, velocity solution.

Table 6
Example 3 with adaptive mesh refinement, computational results.

N_{iter}	Max solves	Time (s)	N_{iter}	Max solves	Time (s)
Method 1			Method 2		
9715	9733	11.10	823	2481	5.57
Method 3			Method 4		
9715	**488**	7.85	823	988	**4.66**

Bold—denotes the smallest number of subdomain solves and the fastest run.

the left and right with Dirichlet on the bottom. The boundary conditions for Stokes are inflow condition on the left and zero stress on the right, with specified horizontal velocity and zero normal stress on the top. The problem is solved on a 6×6 domain decomposition.

In this test, an adaptive mesh refinement algorithm [19,39] computes *a posteriori* error indicators in the Darcy region and refines the local and mortar grids for subdomains with large error. The meshes in the Stokes region and on the Stokes–Darcy interface are not affected. We set a starting local mesh of 4×4 in all subdomains and a uniform mortar mesh of 2×1 on every edge. We again run and compare all four methods as in the previous two examples. Table 6 reports the sum of numbers of CG iterations in all levels as well as the maximal number of solves and runtime.

The velocity computed by Method 3 on the last mesh refinement level is presented in Fig. 7. It is easy to observe that regions with higher heterogeneity are refined more times. The total number of solves after the last refinement on each subdomain for Method 1 and 3 are shown in Fig. 8 and Fig. 9, respectively. Without multiscale flux basis, the number of solves is the same in all Darcy subdomains, regardless of the difference of their meshes. Stokes domains report the same numbers with just four extra solves for implementing the FETI method. With multiscale flux basis, the number of solves depends on the mortar degrees of freedom on the surrounding interfaces. Hence, with different levels of mesh refinement, the Darcy subdomains are reporting different number of solves. Clearly the number of solves is reduced significantly in Method 3 compared to Method 1. It is evident from the left column of Table 6, that the multiscale flux basis saves roughly 95 percent in maximum solves and 30 percent in runtime. Comparing results in the right column, Method 4 saves roughly 60 percent in maximum solves and 20 percent in runtime from Method 2. If there were more levels of mesh refinement, there would be more significant saving in both numbers, since the computational saving due to the multiscale flux basis occurs on every refinement level.

5. Conclusion

In this paper, we develop a multiscale flux basis algorithm for the MMMFEM discretization of the coupled Stokes–Darcy problem and compare it to the original implementation. The new method precomputes the basis by solving a fixed number of subdomain solves, which depend on the number of coarse scale mortar degrees of freedom per subdomain. The basis is used to replace the subdomain solves in the global interface iteration, completely eliminating the dependency between the number of solves and the number of iterations. The numerical examples for a variety of test cases show that the multiscale flux basis implementation is more computationally efficient than

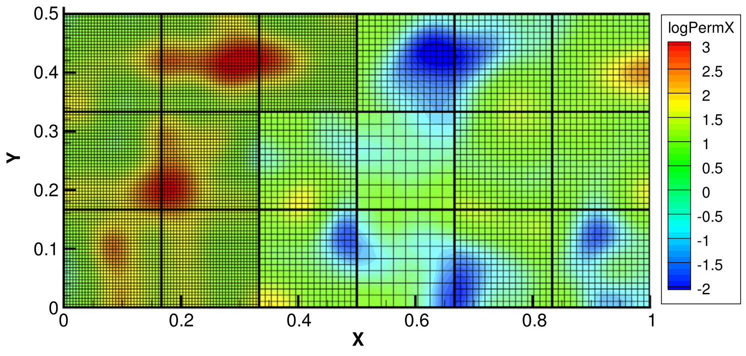

Fig. 6. Example 3, permeability field on the last mesh refinement level.

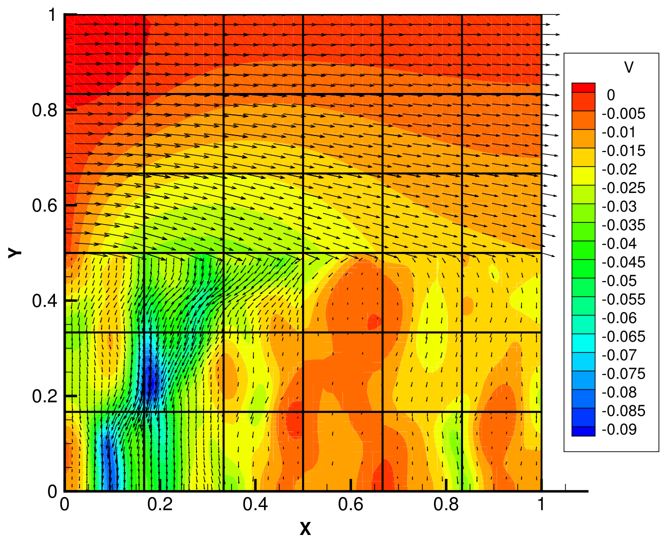

Fig. 7. Example 3, velocity solution on the last mesh refinement level.

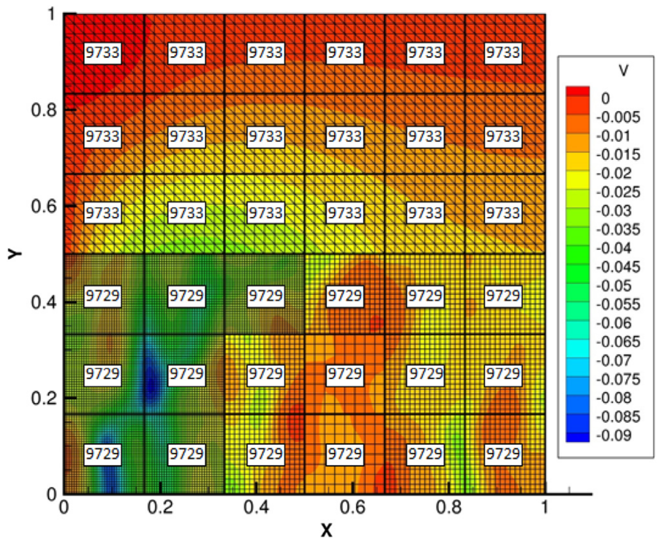

Fig. 8. Number of solves for Method 1, shown on each subdomain.

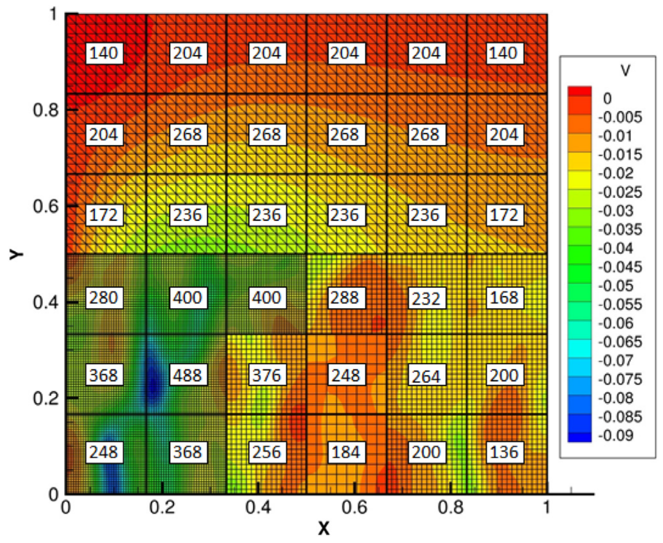

Fig. 9. Number of solves for Method 3, shown on each subdomain.

the original MMMFEM, and a greater savings occurs with increased number of subdomains and global mortar degrees of freedom. We also demonstrate the multiscale flux basis can be combined with the balancing preconditioner in Darcy to form a powerful solver for large-scale problems.

Acknowledgments

Third and fourth authors are partially supported by NSF grant DMS 1418947 and DOE grant DE-FG02-04ER25618.

References

[1] G.S. Beavers, D.D. Joseph, Boundary conditions at a naturally impermeable wall, J. Fluid. Mech 30 (1967) 197–207.
[2] P.G. Saffman, On the boundary condition at the surface of a porous media, Stud. Appl. Math. L (2) (1971) 93–101.
[3] W.J. Layton, F. Schieweck, I. Yotov, Coupling fluid flow with porous media flow, SIAM J. Numer. Anal. 40 (6) (2003) 2195–2218.
[4] M. Discacciati, E. Miglio, A. Quarteroni, Mathematical and numerical models for coupling surface and groundwater flows, Appl. Numer. Math. 43 (1–2) (2002) 57–74. 19th Dundee Biennial Conference on Numerical Analysis (2001).
[5] B. Rivière, I. Yotov, Locally conservative coupling of Stokes and Darcy flows, SIAM J. Numer. Anal. 42 (5) (2005) 1959–1977.
[6] J. Galvis, M. Sarkis, Non-matching mortar discretization analysis for the coupling Stokes-Darcy equations, Electron. Trans. Numer. Anal. 26 (2007) 350–384.
[7] V. Girault, D. Vassilev, I. Yotov, Mortar multiscale finite element methods for Stokes-Darcy flows, Numer. Math. 127 (2014) 93–165.
[8] P. Song, C. Wang, I. Yotov, Domain decomposition for Stokes-Darcy flows with curved interfaces, Proc. Comput. Sci. 18 (2013) 1077–1086.
[9] D. Vassilev, C. Wang, I. Yotov, Domain decomposition for coupled Stokes and Darcy flows, Comput. Methods Appl. Mech. Engrg. 268 (2014) 264–283.
[10] C. Bernardi, T.C. Rebollo, F. Hecht, Z. Mghazli, Mortar finite element discretization of a model coupling Darcy and Stokes equations, M2AN Math. Model. Numer. Anal. 42 (3) (2008) 375–410.
[11] M. Discacciati, A. Quarteroni, Analysis of a domain decomposition method for the coupling of Stokes and Darcy equations, in: Numerical Mathematics and Advanced Applications, Springer Italia, Milan, 2003, pp. 3–20.
[12] M. Discacciati, A. Quarteroni, Convergence analysis of a subdomain iterative method for the finite element approximation of the coupling of Stokes and Darcy equations, Comput. Vis. Sci. 6 (2–3) (2004) 93–103.
[13] M. Discacciati, A. Quarteroni, A. Valli, Robin-Robin domain decomposition methods for the Stokes-Darcy coupling, SIAM J. Numer. Anal. 45 (3) (2007) 1246–1268.
[14] R.H.W. Hoppe, P. Porta, Y. Vassilevski, Computational issues related to iterative coupling of subsurface and channel flows, Calcolo 44 (1) (2007) 1–20.
[15] J. Galvis, M. Sarkis, FETI and BDD preconditioners for Stokes-Mortar-Darcy systems, Commun. Appl. Math. Comput. Sci. 5 (2010) 1–30.
[16] C. Farhat, F.X. Roux, A method of finite element tearing and interconnecting and its parallel solution algorithm, Internat. J. Methods Engrg. 32 (1991) 1205–1227.
[17] A. Toselli, O. Widlund, Domain Decomposition Methods—Algorithms and Theory, Springer-Verlag Berlin Heidelberg, 2005.

[18] B. Ganis, I. Yotov, Implementation of a Mortar Mixed Finite Element Method using a Multiscale Flux Basis, Comput. Methods Appl. Mech. Engrg. 198 (49–52) (2009) 3989–3998.
[19] T. Arbogast, G. Pencheva, M.F. Wheeler, I. Yotov, A multiscale mortar mixed finite element method, Multiscale Model. Simul. 6 (1) (2007) 319.
[20] T.J.R. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods, Comput. Methods Appl. Mech. Engrg. 127 (1995) 387–401.
[21] T. Arbogast, Analysis of a two-scale, locally conservative subgrid upscaling for elliptic problems, SIAM J. Numer. Anal. 42 (2004) 576–598.
[22] T.Y. Hou, X.H. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, J. Comput. Phys. 134 (1997) 169–189.
[23] Z. Chen, T.Y. Hou, A mixed multiscale finite element method for elliptic problems with oscillating coefficients, Math. Comp. 72 (2003) 541–576.
[24] 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.
[25] Y. Efendiev, J. Galvis, T.Y. Hou, Generalized multiscale finite element methods (GMsFEM), J. Comput. Phys. 251 (2013) 116–135.
[26] J. Galvis, Y. Efendiev, Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces, Multiscale Model. Simul. 8 (5) (2010) 1621–1644.
[27] I. Ambartsumyan, E. Khattatov, C. Wang, I. Yotov, Stochastic multiscale flux basis for Stokes-Darcy flows, preprint.
[28] R.G. Ghanem, P.D. Spanos, Stochastic Finite Elements: A Spectral Approach, Springer-Verlag, New York, 1991.
[29] L.C. Cowsar, J. Mandel, M.F. Wheeler, Balancing domain decomposition for mixed finite elements, Math. of Comp. 64 (211) (1995) 989–1015.
[30] J. Mandel, Balancing domain decomposition, Commun. Numer. Methods Eng. 9 (3) (1993) 233–241.
[31] G. Pencheva, I. Yotov, Balancing domain decomposition for mortar mixed finite element methods on non-matching grids, Numer. Linear Algebra Appl. 10 (2003) 159–180.
[32] C. Taylor, P. Hood, A numerical solution of the Navier-Stokes equations using the finite element technique, Internat. J. Comput. & Fluids 1 (1) (1973) 73–100.
[33] D.N. Arnold, F. Brezzi, M. Fortin, A stable finite element for the Stokes equations, Calcolo 21 (4) (1984) 337–344 (1985).
[34] C. Bernardi, G. Raugel, Analysis of some finite elements for the Stokes problem, Math. Comp. (1985) 71–79.
[35] R.A. Raviart, J.M. Thomas, A mixed finite element method for 2nd order elliptic problems, in: Mathematical Aspects of the Finite Element Method, in: Lecture Notes in Mathematics, vol. 606, Springer-Verlag, New York, 1977, pp. 292–315.
[36] F. Brezzi, J. Douglas Jr., L.D. Marini, Two families of mixed elements for second order elliptic problems, Numer. Math. 88 (1985) 217–235.
[37] F. Brezzi, J. Douglas Jr., R. Duràn, M. Fortin, Mixed finite elements for second order elliptic problems in three variables, Numer. Math. 51 (1987) 237–250.
[38] M.F. Wheeler, G. Xue, I. Yotov, A multiscale mortar multipoint flux mixed finite element method, ESAIM Math. Model. Numer. Anal. 46 (4) (2012) 759–796.
[39] M.F. Wheeler, I. Yotov, A posteriori error estimates for the mortar mixed finite element method, SIAM J. Numer. Anal. 43 (3) (2005) 1021–1042.