A MultiMesh Finite Element Method for the Stokes Problem

August Johansson, Mats G. Larson, and Anders Logg

Abstract The multimesh finite element method enables the solution of partial differential equations on a computational mesh composed by multiple arbitrarily overlapping meshes. The discretization is based on a continuous–discontinuous function space with interface conditions enforced by means of Nitsche's method. In this contribution, we consider the Stokes problem as a first step towards flow applications. The multimesh formulation leads to so called cut elements in the underlying meshes close to overlaps. These demand stabilization to ensure coercivity and stability of the stiffness matrix. We employ a consistent least-squares term on the overlap to ensure that the inf-sup condition holds. We here present the method for the Stokes problem, discuss the implementation, and verify that we have optimal convergence.

1 Introduction

Consider the Stokes problem

$$-\Delta \boldsymbol{u} + \nabla p = \boldsymbol{f} \qquad \text{in } \Omega, \tag{1}$$

$$\text{div}\,\boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{2}$$

$$\boldsymbol{u} = \boldsymbol{0} \qquad \text{on } \partial\Omega, \tag{3}$$

August Johansson

Simula Research Laboratory, P.O. Box 134, 1325 Lysaker, Norway, e-mail: `august.johansson@gmail.com`

Mats G. Larson

Department of Mathematics, Umeå University, 90187, Umeå, Sweden, e-mail: `mats.larson@math.umu.se`

Anders Logg

Department of Mathematical Sciences, Chalmers University of Technology and University of Gothenburg, 41296 Göteborg, Sweden, e-mail: `logg@chalmers.se`.

for the velocity $\boldsymbol{u} : \Omega \to \mathbb{R}^d$ and pressure $\boldsymbol{p} : \Omega \to \mathbb{R}$ in a polytopic domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$.

The Stokes problem is considered here as a first step towards a multimesh formulation for multi-body flow problems, and ultimately fluid–structure interaction problems, in which each body is discretized by an individual boundary-fitted mesh and the boundary-fitted meshes move freely on top of a fixed background mesh. The applications for such a formulation are many, e.g., the simulation of blood platelets in a blood stream, the optimization of the configuration of an array of wind turbines, or the investigation of the effect of building locations in a simulation of urban wind conditions and pollution. Common to these applications is that the multimesh method removes the need for costly mesh (re)generation and allows the platelets, wind turbines or buildings to be moved around freely in the domain, either in each timestep as a part of a dynamic simulation, or in each iteration as part of an optimization problem.

The multimesh formulation presented here is a generalization of the formulation presented and analyzed in [7] for two domains. For comparison, the multimesh discretization of the Poisson problem for arbitrarily many intersecting meshes is presented in [5] and analyzed in [8].

2 Notation

We first review the notation for domains, interfaces, meshes and overlaps used to formulate the multimesh finite element method. For a more detailed exposition, we refer to [5].

Notation for domains

- -

Let $\Omega = \widehat{\Omega}_0 \subset \mathbb{R}^d$, $d = 2, 3$, be a domain with polytopic boundary (the background domain).

Let $\widehat{\Omega}_i \subset \widehat{\Omega}_0$, $i = 1, \ldots, N$ be the so-called *predomains* with polytopic boundaries (see Figure 1).

Let $\Omega_i = \widehat{\Omega}_i \setminus \bigcup_{j=i+1}^N \widehat{\Omega}_j$, $i = 0, \ldots, N$ be a partition of Ω (see Figure 2).
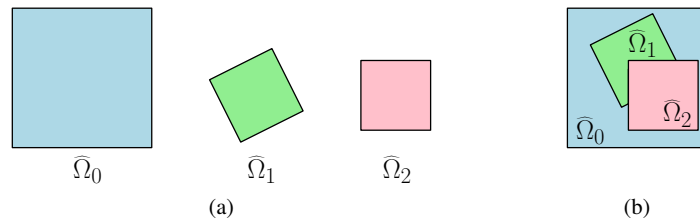

Fig. 1 (a) Three polygonal predomains. (b) The predomains are placed on top of each other in an ordering such that $\widehat{\Omega}_0$ is placed lowest, $\widehat{\Omega}_1$ is in the middle and $\widehat{\Omega}_2$ is on top.

Remark 1. To simplify the presentation, the domains $\Omega_1, \ldots, \Omega_N$ are not allowed to intersect the boundary of Ω.
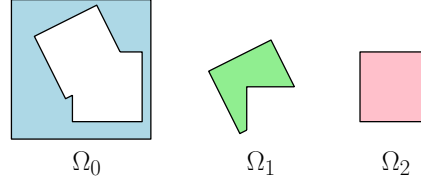

Fig. 2 Partition of $\Omega = \Omega_0 \cup \Omega_1 \cup \Omega_2$. Note that $\Omega_2 = \widehat{\Omega}_2$.

Notation for interfaces

Let the *interface* Γ_i be defined by $\Gamma_i = \partial\widehat{\Omega}_i \setminus \bigcup_{j=i+1}^{N} \widehat{\Omega}_j$, $i = 1, \ldots, N-1$ (see Figure 3a).

Let $\Gamma_{ij} = \Gamma_i \cap \Omega_j$, $i > j$ be a partition of Γ_i (see Figure 3b).
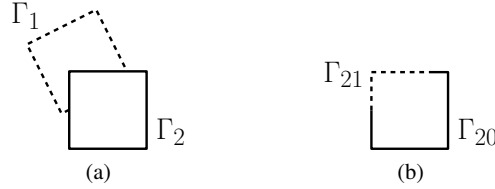

Fig. 3 (a) The two interfaces of the domains in Figure 1: $\Gamma_1 = \partial\widehat{\Omega}_1 \setminus \widehat{\Omega}_2$ (dashed line) and $\Gamma_2 = \partial\widehat{\Omega}_2$ (filled line). Note that Γ_1 is not a closed curve. (b) Partition of $\Gamma_2 = \Gamma_{20} \cup \Gamma_{21}$.

Notation for meshes

Let $\widehat{\mathscr{K}}_{h,i}$ be a quasi-uniform [3] *premesh* on $\widehat{\Omega}_i$ with mesh parameter $h_i = \max_{K \in \widehat{\mathscr{K}}_{h,i}} \mathrm{diam}(K)$, $i = 0, \ldots, N$ (see Figure 4a).

Let $\mathscr{K}_{h,i} = \{K \in \widehat{\mathscr{K}}_{h,i} : K \cap \Omega_i \neq \emptyset\}$, $i = 0, \ldots, N$ be the *active meshes* (see Figure 4b).

The *multimesh* is formed by the active meshes placed in the given ordering (see Figure 5b).

Let $\Omega_{h,i} = \bigcup_{K \in \mathscr{K}_{h,i}} K$, $i = 0, \ldots, N$ be the *active domains*.

Notation for overlaps

Let $\mathscr{O}_i$ denote the *overlap* defined by $\mathscr{O}_i = \Omega_{h,i} \setminus \Omega_i$, $i = 0, \ldots, N-1$.

Let $\mathscr{O}_{ij} = \mathscr{O}_i \cap \Omega_j = \Omega_{h,i} \cap \Omega_j$, $i < j$ be a partition of $\mathscr{O}_i$.
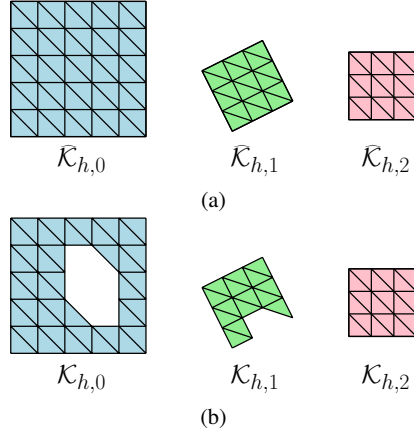

$\bar{\mathcal{K}}_{h,0}$ $\bar{\mathcal{K}}_{h,1}$ $\bar{\mathcal{K}}_{h,2}$

(a)

$\mathcal{K}_{h,0}$ $\mathcal{K}_{h,1}$ $\mathcal{K}_{h,2}$

(b)

Fig. 4 (a) The three premeshes. (b) The corresponding active meshes (cf. Figure 1).

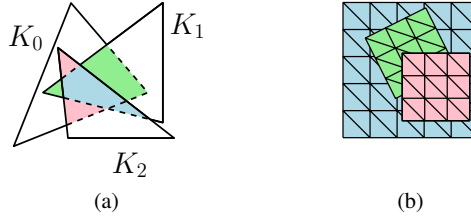K_0 K_1

K_2

(a)   (b)

Fig. 5 (a) Given three ordered triangles K_0, K_1 and K_2, the overlaps are $\mathcal{O}_{01}$ in green, $\mathcal{O}_{02}$ in red and $\mathcal{O}_{12}$ in blue. (b) The multimesh of the domains in Figure 1b consists of the active meshes in Figure 4b.

3 MultiMesh Finite Element Method

To formulate the multimesh finite element for the Stokes problem (1)–(2), we assume for each (active) mesh $\mathcal{K}_{h,i}$ the existence of a pair of inf-sub stable spaces $V_{h,i} \times Q_{h,i}$, $i = 0, 1, \ldots, N$, away from the interface. To be precise, we assume inf-sup stability in $\omega_{h,i} \subset \Omega_{h,i}$ in the sense of (5) below, where $\omega_{h,i}$ is close to Ω_i in the sense that $\Omega_{h,i} \setminus \omega_{h,i} \subset U_\delta(\Gamma_i)$, where

$$U_\delta(\Gamma_i) = \bigcup_{\boldsymbol{x} \in \Gamma_i} B_\delta(\boldsymbol{x}) \tag{4}$$

and $B_\delta(\boldsymbol{x})$ is a ball of radius δ centered at $\boldsymbol{x}$. In other words, $U_\delta(\Gamma_i)$ is the tubular neighborhood of Γ with thickness δ. In the numerical examples, we let $\omega_{h,i}$ be the union of elements in $\mathcal{K}_{h,i}$ with empty intersection with Γ_{ij}, $j > i$.

The inf-sup condition may expressed on each submesh $\omega_{h,i}$ by

$$\|p_i - \lambda_{\omega_{h,i}}(p)\|_{\omega_{h,i}} \lesssim \sup_{\mathbf{v} \in \mathbf{W}_{h,i}} \frac{(\operatorname{div} \mathbf{v}, p)_{\omega_{h,i}}}{\|D\mathbf{v}\|_{\omega_{h,i}}}, \tag{5}$$

where $p_i = p|_{\Omega_{h,i}}$, $\lambda_{\omega_{h,i}}(p)$ is the average of p over $\omega_{h,i}$ and $\mathbf{W}_{h,i}$ is the subspace of $\mathbf{V}_{h,i}$ defined by

$$\mathbf{W}_{h,i} = \{\mathbf{v} \in \mathbf{V}_{h,i} : \mathbf{v} = \mathbf{0} \text{ on } \overline{\Omega_{h,i} \setminus \omega_{h,i}}\}. \tag{6}$$

We now define the multimesh finite element space as the direct sum

$$\mathbf{V}_h \times Q_h = \bigoplus_{i=0}^{N} \mathbf{V}_{h,i} \times Q_{h,i}, \tag{7}$$

where $\mathbf{V}_h$ and Q_h consist of piecewise polynomial of degree k and l, respectively. This means that an element $\mathbf{v} \in \mathbf{V}_h$ is a tuple $(\mathbf{v}_0, \ldots, \mathbf{v}_N)$, and the inclusion $\mathbf{V}_h \hookrightarrow L^2(\Omega)$ is defined by $\mathbf{v}(\mathbf{x}) = \mathbf{v}_i(\mathbf{x})$ for $\mathbf{x} \in \Omega_i$. A similar interpretation is done for $q \in Q_h$. We consider here Taylor-Hood elements [3] with $k \geq 2, l = k - 1$, for which the condition (5) is fulfilled, but non-conforming elements are also possible.

We now consider the following asymmetric finite element method: Find $(\mathbf{u}_h, p_h) \in \mathbf{V}_h \times Q_h$ such that $A_h((\mathbf{u}_h, p_h), (\mathbf{v}, q)) = l_h(\mathbf{v})$ for all $(\mathbf{v}, q) \in \mathbf{V}_h \times Q_h$, where

$$A_h((\mathbf{u}, p), (\mathbf{v}, q)) = a_h(\mathbf{u}, \mathbf{v}) + s_h(\mathbf{u}, \mathbf{v}) + b_h(\mathbf{u}, q) + b_h(\mathbf{v}, p) + d_h((\mathbf{u}, p), (\mathbf{v}, q)),$$

$$\tag{8}$$

$$a_h(\mathbf{u}, \mathbf{v}) = \sum_{i=0}^{N} (D\mathbf{u}_i, D\mathbf{v}_i)_{\Omega_i} \tag{9}$$

$$- \sum_{i=1}^{N} \sum_{j=0}^{i-1} \left((\langle (D\mathbf{u}) \cdot \mathbf{n}_i \rangle, [\mathbf{v}])_{\Gamma_{ij}} + ([\mathbf{u}], \langle (D\mathbf{v}) \cdot \mathbf{n}_i \rangle)_{\Gamma_{ij}} \right)$$

$$+ \sum_{i=1}^{N} \sum_{j=0}^{i-1} \beta_0 h^{-1} ([\mathbf{u}], [\mathbf{v}])_{\Gamma_{ij}},$$

$$s_h(\mathbf{u}, \mathbf{v}) = \sum_{i=0}^{N-1} \sum_{j=i+1}^{N} \beta_1 ([D\mathbf{u}_i], [D\mathbf{v}_i])_{\mathcal{O}_{ij}}, \tag{10}$$

$$b_h(\mathbf{u}, q) = -\sum_{i=0}^{N} (\operatorname{div} \mathbf{u}_i, q_i)_{\Omega_i} + \sum_{i=1}^{N} \sum_{j=0}^{i-1} ([\mathbf{n}_i \cdot \mathbf{u}], \langle q \rangle)_{\Gamma_{ij}}, \tag{11}$$

$$d_h((\mathbf{u}, p), (\mathbf{v}, q)) = \sum_{i=0}^{N} \delta h^2 (\Delta \mathbf{u}_i - \nabla p_i, \Delta \mathbf{v}_i + \nabla q_i)_{\Omega_{h,i} \setminus \omega_{h,i}}, \tag{12}$$

$$l_h(\mathbf{v}) = \sum_{i=0}^{N} (\mathbf{f}, \mathbf{v}_i)_{\Omega_i} - \sum_{i=0}^{N} \delta h^2 (\mathbf{f}, \Delta \mathbf{v}_i + \nabla q_i)_{\Omega_{h,i} \setminus \omega_{h,i}}. \tag{13}$$

Here, β_0 and β_1 are stabilization parameters that must be sufficiently large to ensure that the bilinear form A_h is coercive; cf. [7] for an analysis of the two-domain case.

For simplicity, we use the global mesh size h here and throughout the presentation. If the meshes are of substantially different sizes, it may be beneficial to introduce the individual mesh sizes h_i in (12) and the average $h_i^{-1} + h_j^{-1}$ in (9).

Note that since Γ_i is partitioned into interfaces Γ_{ij} relative to underlying meshes, the sums of the interface terms are over $0 \leq j < i \leq N$. In contrast, the sums of the overlap terms are over $0 \leq i < j \leq N$ since the overlap $\mathscr{O}_i$ is partitioned into overlaps $\mathscr{O}_{ij}$ relative to overlapping meshes.

The jump terms on $\mathscr{O}_{ij}$ and Γ_{ij} are defined by $[\boldsymbol{v}] = \boldsymbol{v}_i - \boldsymbol{v}_j$, where $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are the finite element solutions (components) on the active meshes $\mathscr{K}_{h,i}$ and $\mathscr{K}_{h,j}$. The average normal flux is defined on Γ_{ij} by

$$\langle \boldsymbol{n}_i \cdot \nabla \boldsymbol{v} \rangle = (\boldsymbol{n}_i \cdot \nabla \boldsymbol{v}_i + \boldsymbol{n}_i \cdot \nabla \boldsymbol{v}_j)/2. \tag{14}$$

Here, any convex combination is valid [4].

The proposed formulation (8) is identical to the one proposed in [7] with sums over all domains and interfaces. Also note the similarity with the multimesh formulation for the Poisson problem presented in [5], the difference being the additional least-squares term d_h (and the corresponding term in l_h) since we only assume inf-sup stability in $\omega_{h,i}$, see (5) and the discussion above. If we do not assume inf-sup stability anywhere (e.g. if we would use a velocity-pressure element of equal order), the least-squares term should be applied over the whole domain as in [10]. Please cf. [10] for the use of a symmetric d_h.

Other stabilization terms may be considered. By norm equivalence, the stabilization term $s_h(\boldsymbol{u}, \boldsymbol{v})$ may alternatively be formulated as

$$s_h(\boldsymbol{u}, \boldsymbol{v}) = \sum_{i=0}^{N-1} \sum_{j=i+1}^{N} \beta_2 h^{-2}([\boldsymbol{u}], [\boldsymbol{v}])_{\mathscr{O}_{ij}}. \tag{15}$$

where β_2 is a stabilization parameter; see [8].

Note that the finite element method weakly approximates continuity in the sense that $[\boldsymbol{u}_h] = \boldsymbol{0}$ and $[\boldsymbol{n}_i \cdot \nabla \boldsymbol{u}_h] = 0$ on all interfaces.

4 Implementation

We have implemented the multimesh finite element method as part of the software framework FEniCS [9, 1]. One of the main features of FEniCS is the form language UFL [2] which allows variational forms to be expressed in near-mathematical notation. However, to express the multimesh finite element method (8), a number of custom measures must be introduced. In particular, new measures must be introduced for integrals over cut cells, interfaces and overlaps. These measures are then

mapped to quadrature rules that are computed at runtime. An overview of these algorithms algorithms and the implementation is given in [6].

To express the multimesh finite element method, we let dX denote the integration over domains Ω_i, $i = 0,\dots,N$, including cut cells. Integration over Γ_{ij} and $\mathscr{O}_{ij}$ are expressed using the measures dI and dO, respectively. We let dC denote integration over $\Omega_{h,i} \setminus \omega_{h,i}$. Now the multimesh finite element method for the Stokes problem may be expressed as

```
a_h = inner(grad(u), grad(v))*dX \
      - inner(avg(grad(u)), tensor_jump(v, n))*dI \
      - inner(avg(grad(v)), tensor_jump(u, n))*dI \
      + beta_0/h * inner(jump(u), jump(v))*dI
s_h = beta_1 * inner(jump(grad(u)), jump(grad(v)))*dO
b_h = lambda v, q: inner(-div(v), q)*dX \
                 + inner(jump(v, n), avg(q))*dI
d_h = delta*h**2 * inner(-div(grad(u)) + grad(p), \
                         -div(grad(v)) - grad(q))*dC
```

This makes it easy to implement the somewhat lengthy form (8), as well as investigate the effect of different stabilization terms.

5 Numerical Results

To investigate the convergence of the multimesh finite element method, we solve the Stokes problem in the unit square with the following exact solution

$$\boldsymbol{u}(x,y) = 2\pi \sin(\pi x)\sin(\pi y) \cdot (\cos(\pi y)\sin(\pi x), -\cos(\pi x)\sin(\pi y)), \qquad (16)$$

$$p(x,y) = \sin(2\pi x)\sin(2\pi y), \qquad (17)$$

and corresponding right hand side. We use $P_k P_{k-1}$ Taylor–Hood elements with $k \in \{2,3,4\}$ and we use $N \in \{1,2,4,8,16,32\}$ randomly placed domains as in [5] (see Figure 6). Due to the random placement of domains, some domains are completely hidden and will not contribute to the solution. For $N = 8$, this is the case for one domain, for $N = 16$, three domains and for $N = 32$, four domains are completely hidden. This is automatically handled by the computational geometry routines. Convergence results are presented in Figure 7 as well as in Table 1.

6 Discussion

The results presented in Table 1 and Figure 7 show the expected order of convergence for the velocity in the $L^2(\Omega)$ norm $(k+1)$, for the velocity in the $H_0^1(\Omega)$ norm (k), and for the pressure in the $L^2(\Omega)$ norm (k).
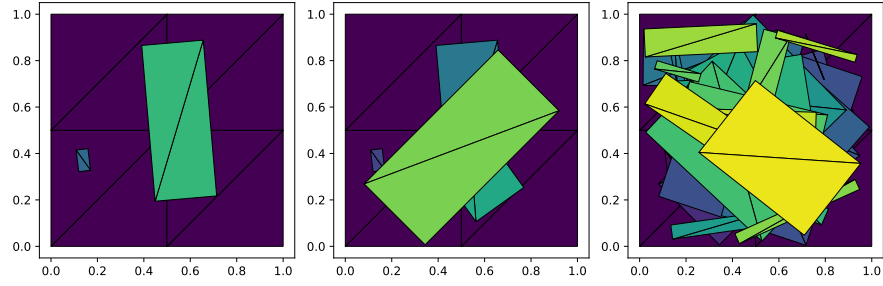
Fig. 6 A sequence of N meshes are randomly placed on top of a fixed background mesh of the unit square shown here for $N = 2, 4$ and 32 using the coarsest refinement level.
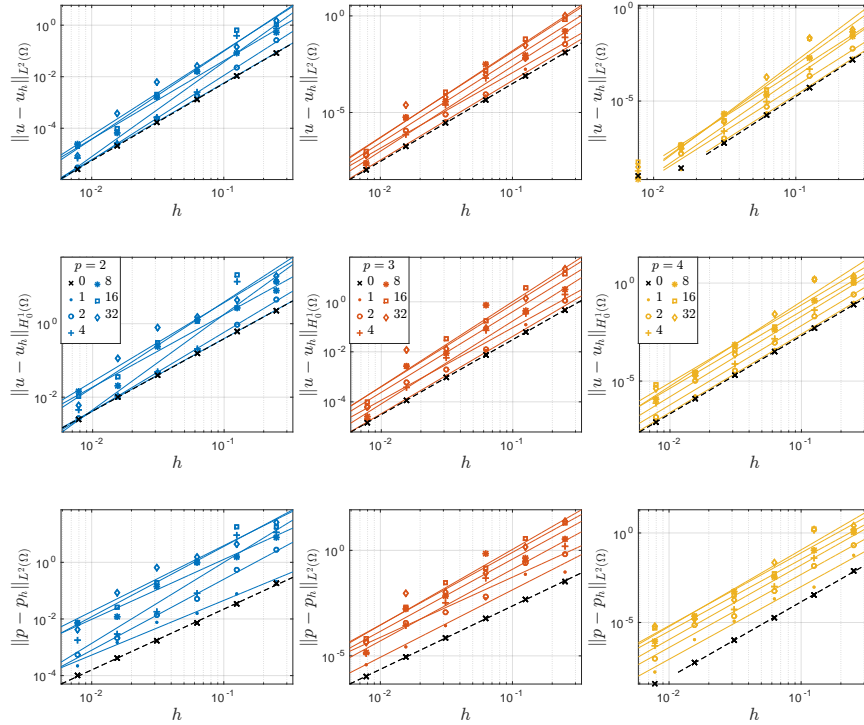

Fig. 7 Convergence results for $k = 2$, 3 and 4 (left to right) using up to 32 meshes (single mesh results are $N = 0$). From top to bottom we have the velocity error in the $L^2(\Omega)$ norm, the velocity error in the $H_0^1(\Omega)$ norm, and the pressure error in the $L^2(\Omega)$ norm. Results less than 10^{-8} are not included in the convergence lines due to limits in floating point precision.

Table 1 Error rates for $e_{L^2} = \|u - u_h\|_{L^2(\Omega)}$, $e_{H_0^1} = \|u - u_h\|_{H_0^1(\Omega)}$ and $e_{L^2} = \|p - p_h\|_{L^2(\Omega)}$.

		$k = 2$			$k = 3$			$k = 4$	
N	e_{L^2}	$e_{H_0^1}$	e_{L^2}	e_{L^2}	$e_{H_0^1}$	e_{L^2}	e_{L^2}	$e_{H_0^1}$	e_{L^2}
0	2.9952	1.9709	2.1466	4.0289	2.9966	3.0028	4.9508	3.9844	4.2286
1	2.9750	1.9658	1.9291	4.1153	3.0912	3.1932	4.8861	4.0006	4.0587
2	3.2764	2.1472	2.5036	3.9087	2.9021	2.8489	4.8677	4.0416	4.0832
4	3.6666	2.5971	2.8597	4.3996	3.3125	3.3957	5.2741	4.0966	4.1609
8	3.0359	1.9697	2.1163	4.3412	3.2258	3.4213	4.8840	3.9409	4.0169
16	3.4131	2.3298	2.4794	4.5033	3.3907	3.5910	5.4702	3.9664	4.0729
32	3.2832	2.1505	2.3255	4.4196	3.2922	3.4362	5.7538	4.3191	4.2848

A detailed inspection of Figure 7 reveals that, as expected, the multimesh discretization yields larger errors than the single mesh discretization (standard Taylor–Hood on one single mesh). The errors introduced by the multimesh discretization are one to two orders of magnitude larger than the single mesh error. However, the convergence rate is optimal and it should be noted that the results presented here are for an extreme scenario where a large number of meshes are simultaneously overlapping; see Figure 6. For a normal application, such as the simulation of flow around a collection of objects, each object would be embedded in a boundary-fitted mesh and only a small number of meshes would simultaneously overlap (in addition to each mesh overlapping the fixed background mesh), corresponding to the situation when two or more objects are close.

The presented method and implementation demonstrate the viability of the multimesh method as an attractive alternative to existing methods for discretization of PDEs on domains undergoing large deformations. In particular, the discretization and the implementation are robust to thin intersections and rounding errors, both of which are bound to appear in a simulation involving a large number of meshes, timesteps or configurations.

References

1. Alnæs, M.S., Blechta, J., Hake, J., Johansson, A., Kehlet, B., Logg, A., Richardson, C., Ring, J., Rognes, M.E., Wells, G.N.: The FEniCS Project Version 1.5. Archive of Numerical Software **3**(100) (2015). DOI 10.11588/ans.2015.100.20553
2. Alnæs, M.S., Logg, A., Oelgaard, K.B., Rognes, M.E., Wells, G.N.: Unified Form Language: A domain-specific language for weak formulations of partial differential equations. ACM

Transactions on Mathematical Software **40** (2014). DOI 10.1145/2566630

3. Brenner, S.C., Scott, L.R.: The Mathematical Theory of Finite Element Methods. Springer, New York (2008). DOI 10.1007/978-0-387-75934-0

4. Hansbo, A., Hansbo, P., Larson, M.G.: A finite element method on composite grids based on Nitsche's method. ESAIM-Math. Model. Num. **37**(3), 495–514 (2003). DOI 10.1051/m2an:2003039

5. Johansson, A., Kehlet, B., Larson, M.G., Logg, A.: MultiMesh Finite Element Methods I: Solving PDEs on Multiple Intersecting Meshes. ArXiv e-prints (2018)

6. Johansson, A., Kehlet, B., Logg, A.: Construction of quadrature rules on general polygonal and polyhedral domains in cut finite element methods. In preparation (2018)

7. Johansson, A., Larson, M.G., Logg, A.: High order cut finite element methods for the Stokes problem. Advanced Modeling and Simulation in Engineering Sciences **2**(1), 1–23 (2015). DOI 10.1186/s40323-015-0043-7

8. Johansson, A., Larson, M.G., Logg, A.: MultiMesh Finite Element Methods II: Analysis. ArXiv e-prints (2018)

9. Logg, A., Mardal, K.A., Wells, G.N., et al.: Automated Solution of Differential Equations by the Finite Element Method. Springer (2012). DOI 10.1007/978-3-642-23099-8

10. Massing, A., Larson, M.G., Logg, A., Rognes, M.E.: A stabilized Nitsche overlapping mesh method for the Stokes problem. Numerische Mathematik pp. 1–29 (2014). DOI 10.1007/s00211-013-0603-z