

A stabilized formulation for the solution of the incompressible unsteady Stokes equations in the frequency domain

Mahdi Esmaily
Cornell University

Abstract

A stabilized finite element method is introduced for the simulation of time-periodic creeping flows, such as those found in the cardiorespiratory systems. The new technique, which is formulated in the frequency rather than time domain, strictly uses real arithmetics and permits the use of similar shape functions for pressure and velocity for ease of implementation. It involves the addition of the Laplacian of pressure to the continuity equation with a complex-valued stabilization parameter that is derived systematically from the momentum equation. The numerical experiments show the excellent accuracy and robustness of the proposed method in simulating flows in complex and canonical geometries for a wide range of conditions. The present method significantly outperforms a traditional solver in terms of both computational cost and scalability, which lowers the overall solution turnover time by several orders of magnitude.

1 Introduction

Simulation of time-periodic creeping flows, such as cardiorespiratory flows in smaller vessels, can significantly benefit from a formulation that is expressed in the frequency rather than the time domain. Firstly, the boundary conditions in these problems, which typically vary smoothly in time, can be represented via a handful of Fourier modes. By solving for those few selected Fourier modes rather than integrating over thousands of time steps, a frequency formulation reduces the cost of a simulation by orders of magnitude. The absence of homogeneous solution in the frequency formulation presents a second major cost advantage. In a standard time formulation, the transient solution obtained at the beginning of the simulation has little significance, yet it must be computed before the particular solution can be obtained. Since a frequency formulation is independent of the initial conditions, it directly produces the particular solution, thus avoiding this costly and unnecessary computation. Thirdly, while the frequency formulation is embarrassingly parallelizable, a standard time formulation can be hardly parallelized in time. Thus, computations that are based on frequency formulation can be scaled to a much larger number of processors, permitting a much shorter solution turnover time. Fourthly, the time integration error present in the standard time formulation is absent in the frequency formulation. The truncation error associated with the selection of a finite number of Fourier modes is, however, present in the frequency formulation. Lastly, the stability consideration associated with the time integrator is no longer a concern in a frequency formulation as the solution is obtained from solving a boundary value problem.

The advantages of the frequency formulation for the solution of the incompressible unsteady Stokes equations enumerated above were shown in practice in a recent article [1]. That included one to two orders of magnitude reduction in cost and improvement in scalability by the number of computed modes. Despite these attractive results, the formulation presented in [1] had several shortcomings. Firstly, it relies on complex arithmetics and thus was hard to implement by requiring a significant change in the existing implementation

of both the fluid and linear solver. Furthermore, a Bubnov-Galerkin formulation was employed in that case to satisfy the inf-sup condition [2, 3, 4]. That led to two additional shortcomings, namely the requirement to use mixed shape functions for pressure and velocity and also the stiffness of the tangent matrix. The use of linear shape function for pressure and quadratic shape function for velocity is not a convenient choice, particularly in the case of complex geometries. The issue associated with the stiffness of the tangent matrix is caused by a zero block on the diagonal of the stiffness matrix (given that the continuity equation does not depend on the pressure), which delays the convergence of the iterative linear solver. It was shown that this slow convergence contributes to an order of magnitude increase in cost, particularly at higher modes where the condition number of the stiffness matrix increases.

The present article introduces a stabilized formulation to overcome the aforementioned issues. Namely, the new formulation uses purely real arithmetics, permits the use of equal order shape functions, and avoids the zero block in the tangent matrix for faster convergence of the linear solver. The article is organized as follows. In Section 2, the stabilized formulation is presented. Then in Section 3, the proposed formulation is tested using a canonical and a complex patient-specific geometry. Lastly, Section 4 contains the concluding remarks.

2 A time-spectral stabilized formulation for Stokes equation

Creeping flows, such as those found in cardiorespiratory flows, can be modeled using the incompressible unsteady Stokes equation. Taking the temporal Fourier transformation of the unsteady Stokes equations produces a boundary value problem that is stated as

$$\begin{aligned}
\hat{j}\rho\omega\mathbf{u} &= -\nabla p + \nabla \cdot (\mu\nabla\mathbf{u}) && \text{in } \Omega, \\
\nabla \cdot \mathbf{u} &= 0 && \text{in } \Omega, \\
\mathbf{u} &= \mathbf{g} && \text{on } \Gamma_g, \\
(-p\mathbf{I} + \mu\nabla\mathbf{u}) \cdot \mathbf{n} &= \mathbf{h} && \text{on } \Gamma_h.
\end{aligned} \tag{1}$$

where $\hat{j} = \sqrt{-1}$, ω is the oscillation frequency, \mathbf{x} is position, $\mathbf{u}(\omega, \mathbf{x})$ is the velocity, $p(\omega, \mathbf{x})$ is pressure, $\mathbf{g}(\omega, \mathbf{x})$ is the imposed velocity on the Dirichlet boundary Γ_g , and $\mathbf{h}(\omega, \mathbf{x})$ is the imposed traction on the Neumann boundary Γ_h . Note that the time derivative term in the Stokes equations is transformed to a complex-valued source term in Eq. (1). Thus, the unsteady Stokes equations, when expressed in the frequency domain, behave similarly to the steady Stokes equations with a nonzero complex source term.

Since \mathbf{u} and p are complex-valued variables in general, numerical simulation of Eq. (1) will require complex arithmetics [1]. For a real-valued formulation, the state variables as well as boundary conditions can be expressed as

$$\begin{aligned}
\mathbf{u} &= \mathbf{u}_r + \hat{j}\mathbf{u}_i, \\
p &= p_r + \hat{j}p_i, \\
\mathbf{g} &= \mathbf{g}_r + \hat{j}\mathbf{g}_i, \\
\mathbf{h} &= \mathbf{h}_r + \hat{j}\mathbf{h}_i.
\end{aligned} \tag{2}$$

Using these changes of variables, Eq. (1) can be rewritten in the real domain as

$$\begin{aligned}
-\rho\omega\mathbf{u}_i &= -\nabla p_r + \nabla \cdot (\mu\nabla\mathbf{u}_r), && \rho\omega\mathbf{u}_r = -\nabla p_i + \nabla \cdot (\mu\nabla\mathbf{u}_i) && \text{in } \Omega, \\
\nabla \cdot \mathbf{u}_r &= 0, && \nabla \cdot \mathbf{u}_i &= 0 && \text{in } \Omega, \\
\mathbf{u}_r &= \mathbf{g}_r, && \mathbf{u}_i &= \mathbf{g}_i && \text{on } \Gamma_g, \\
(-p_r\mathbf{I} + \mu\nabla\mathbf{u}_r) \cdot \mathbf{n} &= \mathbf{h}_r, && (-p_i\mathbf{I} + \mu\nabla\mathbf{u}_i) \cdot \mathbf{n} &= \mathbf{h}_i && \text{on } \Gamma_h.
\end{aligned} \tag{3}$$

To weak form of Eq. (3) can be expressed as finding $\mathbf{u}_r \in \mathcal{S}_r$, $\mathbf{u}_i \in \mathcal{S}_i$ and $p_r, p_i \in L^2$ such that for any $\mathbf{w}_r, \mathbf{w}_i \in \mathcal{W}$ and $q_r, q_i \in L^2$ the following must hold

$$B_G = \int_{\Omega} \left[-\rho\omega \mathbf{w}_r \cdot \mathbf{u}_i + \nabla \mathbf{w}_r : (-p_r I + \mu \nabla \mathbf{u}_r) - q_r \nabla \cdot \mathbf{u}_r \right. \\ \left. - \rho\omega \mathbf{w}_i \cdot \mathbf{u}_r - \nabla \mathbf{w}_i : (-p_i I + \mu \nabla \mathbf{u}_i) + q_i \nabla \cdot \mathbf{u}_i \right] d\Omega, \quad (4)$$

$$F_G = \int_{\Gamma_h} \left[\mathbf{w}_r \cdot \mathbf{h}_r - \mathbf{w}_i \cdot \mathbf{h}_i \right] d\Gamma.$$

In this equation, \mathbf{w}_r , \mathbf{w}_i , q_r and q_i are test functions for velocity and pressure and

$$\begin{aligned} \mathcal{S}_r &= \{ \mathbf{u}_r | \mathbf{u}_r \in (H^1)^{n_{sd}}, \mathbf{u} = \mathbf{g}_r \text{ on } \Gamma_g \}, \\ \mathcal{S}_i &= \{ \mathbf{u}_i | \mathbf{u}_i \in (H^1)^{n_{sd}}, \mathbf{u} = \mathbf{g}_i \text{ on } \Gamma_g \}, \\ \mathcal{W} &= \{ \mathbf{w} | \mathbf{w} \in (H^1)^{n_{sd}}, \mathbf{w} = \mathbf{0} \text{ on } \Gamma_g \}. \end{aligned} \quad (5)$$

In above, L^2 denotes the space of scalar-valued functions that are square-integrable on Ω . Also, $(H^1)^{n_{sd}}$ denotes the space of vector-valued functions with square-integrable derivatives on Ω .

In obtaining B_G and F_G in Eq. (4), the imaginary part of the momentum equation and real part of the continuity equation were multiplied by -1 . The sign of those equations was changed to ensure the tangent matrix remains symmetric.

To stabilize Eq. (4) for equal order shape functions for \mathbf{u} and p and relate the continuity equation to pressure, the Laplacian of pressure is added to the continuity equation. To systematically derive this term, consider the divergence of the momentum equation from Eq. (1) that is

$$\hat{j} \rho \omega \nabla \cdot \mathbf{u} = -\nabla^2 p + \nabla \cdot [\nabla \cdot (\mu \nabla \mathbf{u})]. \quad (6)$$

Since the last term in Eq. (6) involves the third derivative of velocity, it will vanish in the interior of an element when linear shape functions are employed. It is thus approximated using the characteristic size of the element H as $\nabla \cdot [\nabla \cdot (\mu \nabla \mathbf{u})] \approx (\mu/H^2) \nabla \cdot \mathbf{u}$. Thus, Eq. (6) can be written as

$$\nabla \cdot \mathbf{u} - \tau \nabla^2 p = 0, \quad (7)$$

where

$$\tau = \frac{1}{(\mu/H^2) - \rho \omega \hat{j}}. \quad (8)$$

The numerical experiments show that incorporating a small constant in the definition of τ improves the solution accuracy and reduces the number of linear solver iterations. They also show $\boldsymbol{\xi} \in \mathbb{R}^{n_{sd}} \times \mathbb{R}^{n_{sd}}$, which is the covariant tensor obtained from a mapping between the physical and parent elements, provides a good approximation for H [5]. Thus, in practice, the real and imaginary component of τ are defined as

$$\tau_r = \frac{c\mu\sqrt{\boldsymbol{\xi} : \boldsymbol{\xi}}}{(\rho\omega)^2 + \mu^2 \boldsymbol{\xi} : \boldsymbol{\xi}}, \quad \tau_i = \frac{c\rho\omega}{(\rho\omega)^2 + \mu^2 \boldsymbol{\xi} : \boldsymbol{\xi}}. \quad (9)$$

The numerical simulation involving 2D triangular elements and 3D tetrahedral elements show $c = 2^{-5} \approx 0.03$ produces satisfactory results. This value of c is utilized for all the computations reported in Section 3.

From Eqs. (7) and (9), the stabilization terms are computed as

$$B_S = \sum_e \int_{\Omega_e} \left[-\nabla q_r \cdot (\tau_r \nabla p_r - \tau_i \nabla p_i) + \nabla q_i \cdot (\tau_r \nabla p_i + \tau_i \nabla p_r) \right] d\Omega, \quad (10)$$

where the signs are selected to be consistent with those in Eq. (4). These integrals are added to the Galerkin's weak form from Eq. (4) to obtain

$$B_G + B_S = F_G. \quad (11)$$

Derivation of the matrix form of Eq. (11) follows a standard process and, thus, not included here in detail. The result is

$$\begin{bmatrix} \mu L_{AB} \boldsymbol{\delta} & -\mathbf{G}_{AB} & -\rho\omega M_{AB} \boldsymbol{\delta} & \mathbf{0} \\ -\mathbf{D}_{AB} & -\tau_r L_{AB} & \mathbf{0} & \tau_i L_{AB} \\ -\rho\omega M_{AB} \boldsymbol{\delta} & \mathbf{0} & -\mu L_{AB} \boldsymbol{\delta} & \mathbf{G}_{AB} \\ \mathbf{0} & \tau_i L_{AB} & \mathbf{D}_{AB} & \tau_r L_{AB} \end{bmatrix} \begin{bmatrix} \mathbf{U}_{rB} \\ P_{rB} \\ \mathbf{U}_{iB} \\ P_{iB} \end{bmatrix} = - \begin{bmatrix} \mathbf{R}_{rA}^m \\ R_{rA}^c \\ \mathbf{R}_{iA}^m \\ R_{iA}^c \end{bmatrix} \quad (12)$$

where

$$\begin{aligned} L_{AB} &= \int_{\Omega} \nabla N_A \cdot \nabla N_B d\Omega, \\ \mathbf{G}_{AB} &= \int_{\Omega} \nabla N_A N_B d\Omega, \\ \mathbf{D}_{AB} &= \int_{\Omega} N_A \nabla N_B d\Omega, \\ M_{AB} &= \int_{\Omega} N_A N_B d\Omega, \end{aligned} \quad (13)$$

and

$$\begin{aligned} \mathbf{R}_{rA}^m &= - \int_{\Gamma_h} N_A \mathbf{h}_r d\Gamma + \mu L_{AB} \mathbf{g}_{rB} - \rho\omega M_{AB} \mathbf{g}_{iB}, \\ \mathbf{R}_{iA}^m &= \int_{\Gamma_h} N_A \mathbf{h}_i d\Gamma - \mu L_{AB} \mathbf{g}_{iB} - \rho\omega M_{AB} \mathbf{g}_{rB}, \\ R_{rA}^c &= -\mathbf{D}_{AB} \mathbf{g}_{rB}, \\ R_{iA}^c &= \mathbf{D}_{AB} \mathbf{g}_{iB}. \end{aligned} \quad (14)$$

In Eq. (12), \mathbf{U}_{rB} and \mathbf{U}_{iB} are the real and imaginary component of velocity vector at node B , respectively, and P_{rB} and P_{iB} are the real and imaginary component of pressure at node B , respectively.

Remarks on Eq. (12)

1. The tangent matrix in Eq. (12) is independent of the solution, thus, there is no need for Newton-Raphson iterations when this scheme is implemented. This linear property, which is a result of the linearity of the Stokes equations and that of the designed stabilization scheme, permits one to obtain the final solution via a single linear solution.
2. There is a linear relationship between the number of nonzero blocks in the tangent matrix and the number of spatial dimensions n_{sd} . In total, the tangent matrix contains $4(n_{sd} + 1)^2$ blocks from which only $8n_{sd} + 4$ are nonzero. That translates to less than half of all blocks for 3D problems (28 out of 64 are nonzero). Despite its relatively large size, this matrix is symmetric and can be solved using an efficient iterative solver.

3. It is possible to design a linear solver with a specialized library for the matrix-vector product that only operates on the nonzero blocks of the tangent matrix. Such an implementation is expected to roughly reduce the cost of 3D computations by half. What is presented below, however, does not take advantage of this optimization and is based on a standard matrix-vector product library that is developed in-house [6].
4. A set of iterative techniques are tested for solving this linear system including, conjugate gradient, successive over-relaxation, bi-conjugate gradient, generalized minimal residual [7], and bi-partitioned methods [8]. A symmetric Jacobi preconditioner was used for all these cases, producing diagonal entries with a mod of one. The results show that the conjugate gradient is the most efficient technique for solving the linear system in Eq. (12) and thus used for all the cases presented in Section 3. This superior performance is despite the nonmonotonic convergence of the conjugate gradient and the fact that the tangent matrix is symmetric but indefinite. Nevertheless, the conjugate gradient successfully converges for all cases considered below while being the least costly method among all techniques enumerated above.
5. The introduced scheme only requires the use of a single complex-valued stabilization parameter, i.e. τ defined in Eq. (9). There is an arbitrary constant c that is incorporated into the definition of τ . The numerical results involving 2D and 3D elements show that using a different value for c has a minimal effect on the accuracy of this stabilized formulation. Also, the numerical experiments show this formulation is fairly robust if the element length scale in Eq. (9) is based on a parameter other than ξ (e.g., the Jacobian of element mapping or its volume).
6. The imaginary component of τ , namely τ_i , is crucially necessary at high frequencies. This scheme will struggle to converge for $\omega H^2/\nu \gg 1$ if one ignores the contribution of the off-diagonal terms associated with τ_i in the tangent matrix. Similarly at the steady state limit where $\omega \rightarrow 0$, $\tau_i \rightarrow 0$ and $\tau_r \rightarrow c/(\mu\sqrt{\xi:\xi})$. At this limit, the form of τ_r becomes identical to the existing stabilization parameters that is utilized in the residual-based variational multiscale method at the limit of $\Delta t \rightarrow \infty$ and $\|\mathbf{u}\| \rightarrow 0$ [9, 10].
7. The real and imaginary unknowns are coupled via two sets of off-diagonal blocks in the tangent matrix, both of which are proportional to ω . As $\omega \rightarrow 0$, the two sets of unknowns become decoupled. That is when \mathbf{u}_r and p_r depend only on \mathbf{h}_r and \mathbf{g}_r and not on \mathbf{h}_i and \mathbf{g}_i and vice versa. This limit is physically known as the quasi steady limit where the acceleration term in the Stokes equations is negligible compared to the viscous and pressure terms.
8. Once the velocity and pressure unknowns are computed in the frequency domain, their temporal counterparts can be simply computed using Eq. (2) and

$$\begin{aligned}
\hat{\mathbf{u}}(\mathbf{x}, t) &= \sum_{\omega} \mathbf{u}(\mathbf{x}, \omega) e^{\hat{j}\omega t}, \\
\hat{p}(\mathbf{x}, t) &= \sum_{\omega} p(\mathbf{x}, \omega) e^{\hat{j}\omega t}.
\end{aligned}
\tag{15}$$

3 Results

Two sets of tests cases are considered in this section: 1) a canonical case of an unsteady pipe flow, where the analytical Womersley solution is available for establishing the accuracy of the proposed scheme, and 2) a patient-specific geometry for evaluating the performance of this method on more complex geometries.

Although not presented here for the sake of brevity, simulation involving 2D channel flow has also been performed, where the results are in line with the above 3D cases.

The method described above is implemented in an in-house finite element solver. This solver is parallelized using a message passing interface (MPI). The workload is only parallelized using spatial partitioning by employing ParMETIS library [11]. Further parallelization across different frequencies is not considered here as it involves a trivial process of running a series of simulations with different ω . All computations are performed on a cluster of AMD OpteronTM 6378 processors that are interconnected via a QDR Infiniband.

Unless stated otherwise, a tolerance of $\epsilon_{LS} = 10^{-3}$ is used for the conjugate gradient to solve the linear systems.

3.1 Oscillatory pipe flow

An oscillatory laminar pipe flow is considered for the first test case. A pipe with a length to radius ratio of $L/R = 15$ is considered with an oscillatory unit inlet and zero outlet Neumann boundary condition (i.e., $\mathbf{h}_r = \mathbf{1n}$ and $\mathbf{h}_i = \mathbf{0}$ on the inlet and $\mathbf{h} = \mathbf{0}$ on the outlet). The oscillation frequency ω is varied to simulate flow at eleven Womersley numbers $\alpha = R\sqrt{\rho\omega/\mu} = 0, \sqrt{2}, 2, \dots, 2^5$. Three tetrahedral meshes (M1, M2, and M3) are utilized for spatial discretization (Table 1). All these computations are performed using 16 processors unless stated otherwise.

Table 1: Tetrahedral meshes used for discretization of the 3D pipe flow (M1–M3). N_{ele} and N_{nds} denote the numbers of elements and nodes, respectively.

	M1	M2	M3
N_{ele}	24,450	207,063	728,922
N_{nds}	5,462	37,401	122,291

An analytical solution is available for an oscillatory flow in a pipe, that is expressed in the frequency domain as [12]

$$u_{\text{ref}}(r, \alpha) = \begin{cases} \frac{h}{4\mu L}(R^2 - r^2), & \alpha = 0, \\ -\frac{\hat{j}hR^2}{L\mu\alpha^2} \left[1 - J_0(\hat{j}^{\frac{3}{2}}\alpha)^{-1} J_0(\hat{j}^{\frac{3}{2}}\alpha \frac{r}{R}) \right], & \alpha \neq 0, \end{cases} \quad (16)$$

where J_0 is the zero order Bessel function of the first kind and h is the magnitude of the imposed Neumann boundary condition, which is one in this case. The Womersley solution from Eq. (16) is used as the reference solution to evaluate the performance of the proposed solver.

All the results are normalized using the steady centerline velocity from the reference solution, $u_{\text{ref}}(0, 0)$. An overall good agreement is observed between the simulation and the reference results (Figure 1), particularly at smaller Womersley numbers. Accurate computation of the velocity profile at larger Womersley numbers requires the use finer grids as sharper gradients are developed in those regimes.

The real and imaginary components of the predicted flow rate are compared against the analytical solution of Womersley that is

$$q_{\text{ref}}(\alpha) = \begin{cases} \frac{h\pi R^4}{8\mu L}, & \alpha = 0, \\ \frac{-\hat{j}\pi h R^4}{L\mu\alpha^2} \left[1 + \frac{2\hat{j}^{\frac{1}{2}} J_1(\hat{j}^{\frac{3}{2}}\alpha)}{\alpha J_0(\hat{j}^{\frac{3}{2}}\alpha)} \right], & \alpha \neq 0, \end{cases} \quad (17)$$

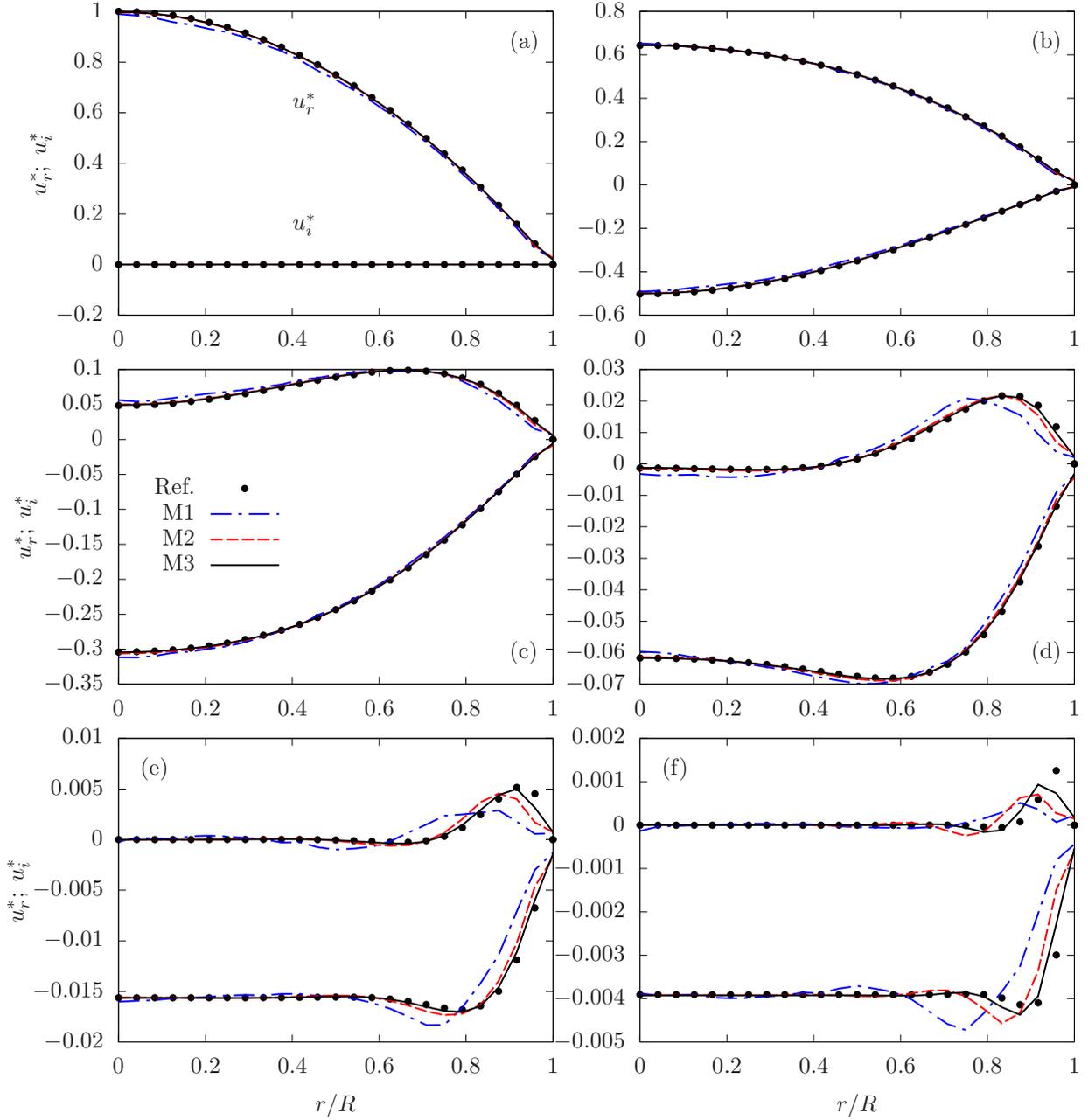


Figure 1: Axial velocity of a pulsating pipe flow as a function of radius computed from the proposed formulation on three meshes (Table 1 where M1, M2, and M3 are represented by dash-dot, dashed, and solid lines, respectively) and the analytical reference solution (Eq. (16) that is shown by dots). Different panels correspond to different Womersley number $\alpha = R\sqrt{\rho\omega/\mu}$. (a) $\alpha = 0$, (b) $\alpha = 2$, (c) $\alpha = 4$, (d) $\alpha = 8$, (e) $\alpha = 16$, and (f) $\alpha = 32$.

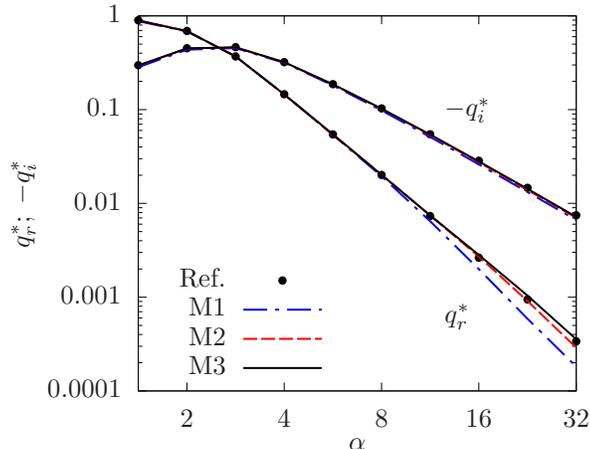


Figure 2: The normalized real (q_r^*) and imaginary (q_i^*) flow rate versus Womersley number $\alpha = R\sqrt{\rho\omega/\mu}$ for three meshes, M1 (dashed-dot), M2 (dashed), and M3 (solid) and their comparison against the reference results (dots) from Eq. (17).

in Figure 2. The results, which are normalized based on the reference steady flow rate ($q_{ref}(0)$), show a good agreement with the reference results. As correctly computed, the portion of the flow rate that is in phase with pressure drops faster at larger α than the out of phase component (i.e., $q_r \rightarrow 0$ faster than $q_i \rightarrow 0$ as $\alpha \rightarrow \infty$). Thus, the fact that pressure and flow are in phase and out of phase by $\pi/2$ at small and large α , respectively, are correctly captured by the proposed formulation.

The change in error as a function of Womersley number (or oscillation frequency ω) and mesh resolution is captured more explicitly in Figure 3(a). Similar to what was observed earlier, increasing the Womersley number or the element size increases the error in the solution. The increase in error with α only occurs beyond a certain threshold since the curvature in the fluid velocity field remains relatively unchanged at small α .

To better understand the computational cost of the proposed method, the number of iterations of the linear solver N_{itr} is plotted for these computations in Figure 3(b). These results show that $N_{itr} = \mathcal{O}(10^3)$ as α is increased from zero to 32, indicating the relative independence of the computational cost as the flow becomes more oscillatory. This independence is in contrast to the previous Bubnov-Galerkin formulation in [1] that showed an increase in cost at high Womersley numbers due to the ill-conditioning of the tangent matrix owing to its zero diagonal block. The higher number of iterations required for the finer grid in Figure 3-(b) is a consequence of the broader range of eigenvalues in the tangent matrix. Adopting a multigrid preconditioner in future studies should break the dependence of N_{itr} on L/H , thus leading to an overall cost that linearly scales with the number of nodes in the mesh.

As stated earlier, the proposed stabilized method has one adjustable parameter c that appeared in Eq. (9). One of the attractive properties of the proposed scheme is that its accuracy hardly depends on this parameter. As shown in Figure 4, changing c by orders of magnitude has little effect on the error. However, the number of linear solver iterations N_{itr} and thus the overall cost do depend on c . These results, which are obtained at $\alpha = 1$, are relatively independent of α . Therefore, it is expected that the proposed $c = 2^{-5}$ to produce accurate results across the board even if it produces a suboptimal convergence rate when an element type other than those tested here (i.e., bilateral and triangular) are used.

The computational performance of the proposed scheme, both in terms of CPU-hours and solution

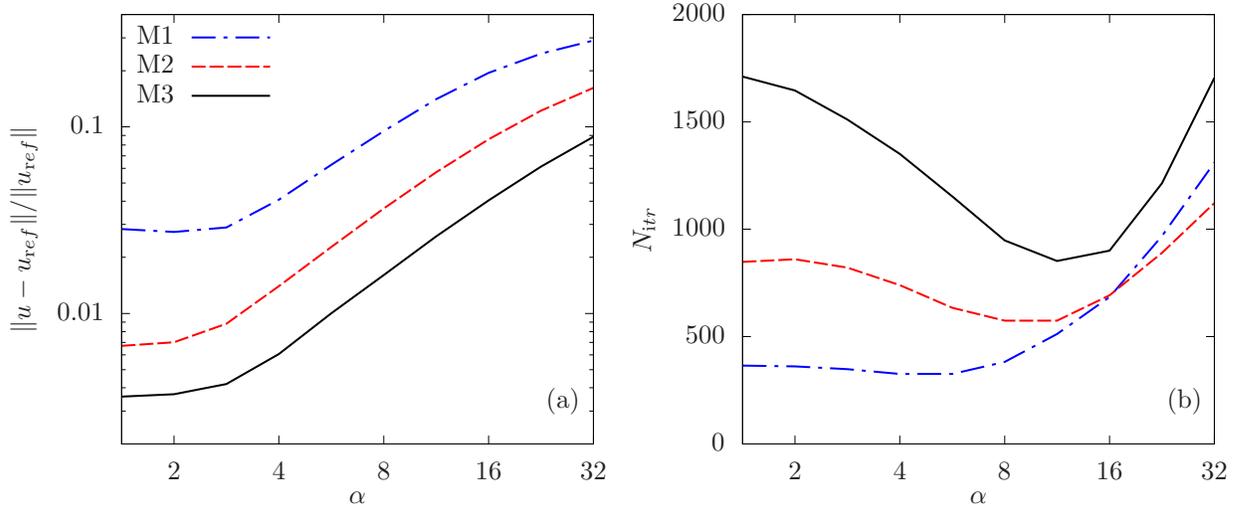


Figure 3: The error (a) and the number of conjugate gradient iterations (b) used for simulating the oscillating pipe flow versus Womersley number $\alpha = R\sqrt{\rho\omega/\mu}$ for three meshes, M1 (dashed-dot), M2 (dashed), and M3 (solid).

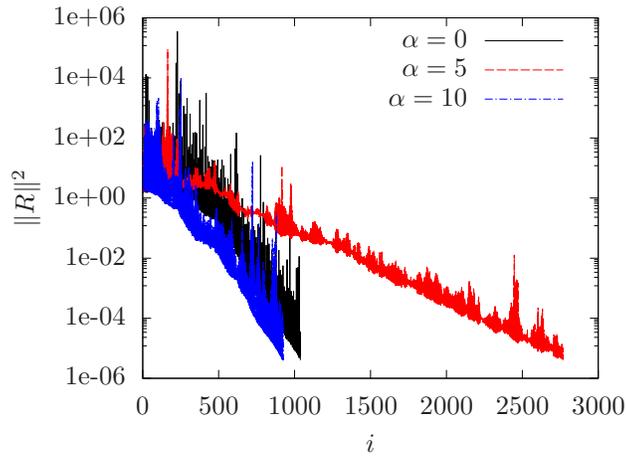


Figure 4: The error (a) and the number of conjugate gradient iterations (b) used for simulating the oscillating pipe flow versus the stabilization parameter c in Eq. (9). The results are obtained at $\alpha = 1$ for three meshes, M1 (dashed-dot), M2 (dashed), and M3 (solid). Similar behavior is observed at other values of α .

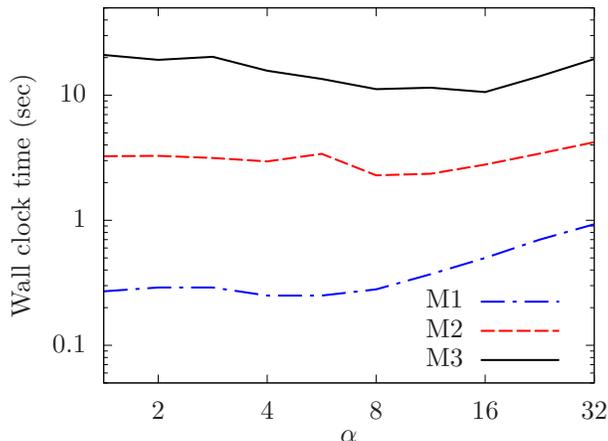


Figure 5: The simulation turnover time in second as a function of Womersley number $\alpha = R\sqrt{\rho\omega/\mu}$ for three meshes, M1 (dashed-dot), M2 (dashed), and M3 (solid).

turnover time, is far shorter than a standard CFD solver and the previously proposed Bubnov-Galerkin formulation. Taking mesh M2 at $\alpha^2 = 8\pi$ as an example, the present formulation takes 3 seconds to produce a solution using 32 processors, which is roughly 0.027 CPU-hours (Figure 5). Similar simulation performed using the Bubnov-Galerkin formulation and a standard CFD solver will require 1.25 and over 300 CPU-hours, respectively [1]. The four orders of magnitude reduction in cost in comparison to the standard CFD solver is roughly the same as the ratio between the number of time steps for the CFD solver and the number of solution modes for the present formulation, which is 10,000 to 1. Even if the number of modes required for an accurate representation of the boundary conditions is $\mathcal{O}(10)$, the present formulation will reduce the cost by three orders of magnitude. The solution turnover time of this case, on the other hand, will always be four orders of magnitude shorter than that of the conventional CFD solver given that all the modes can be computed in parallel for the linear Stokes equations.

3.2 Patient-specific Glenn geometry

To demonstrate the applicability of the proposed method to more complex problems, a patient-specific geometry obtained from a patient undergoing Glenn operation is considered [13]. There are three inflow-outflow boundaries, namely superior vena cava (SVC) and the left and right pulmonary arteries (LPA and RPA). Zero Neumann boundary condition is imposed at the LPA and RPA and a non-slip boundary condition is imposed at the walls (Figure 6). To derive the flow, a unit oscillatory traction is imposed at the SVC with a frequency that is adjusted to obtain different Womersley numbers. The Womersley number, in this case, is defined as $\alpha = D_h\sqrt{\rho\omega/\mu}$, where D_h is the hydraulic diameter of the SVC. The geometry is discretized using 988,747 linear tetrahedral elements, resulting in 163,791 nodes.

The significant dependency of the solution to the Womersley number is shown in Figure 6 for the simulations performed at $\alpha = 0, 5, \text{ and } 10$. Despite geometrical complexity, the overall variation in the real and imaginary component of velocity follows that of the pipe flow case. For the steady case with $\alpha = 0$, $\mathbf{u}_i = \mathbf{0}$ and \mathbf{u}_r forms a parabolic profile at the annular cross sections. As α increases, \mathbf{u}_r develops peaks in the near wall region while \mathbf{u}_i grows in relative magnitude to create a bulk flow at the annular cross sections.

The convergence behavior of the conjugate gradient solver for the three simulations at $\alpha = 0, 5, \text{ and}$

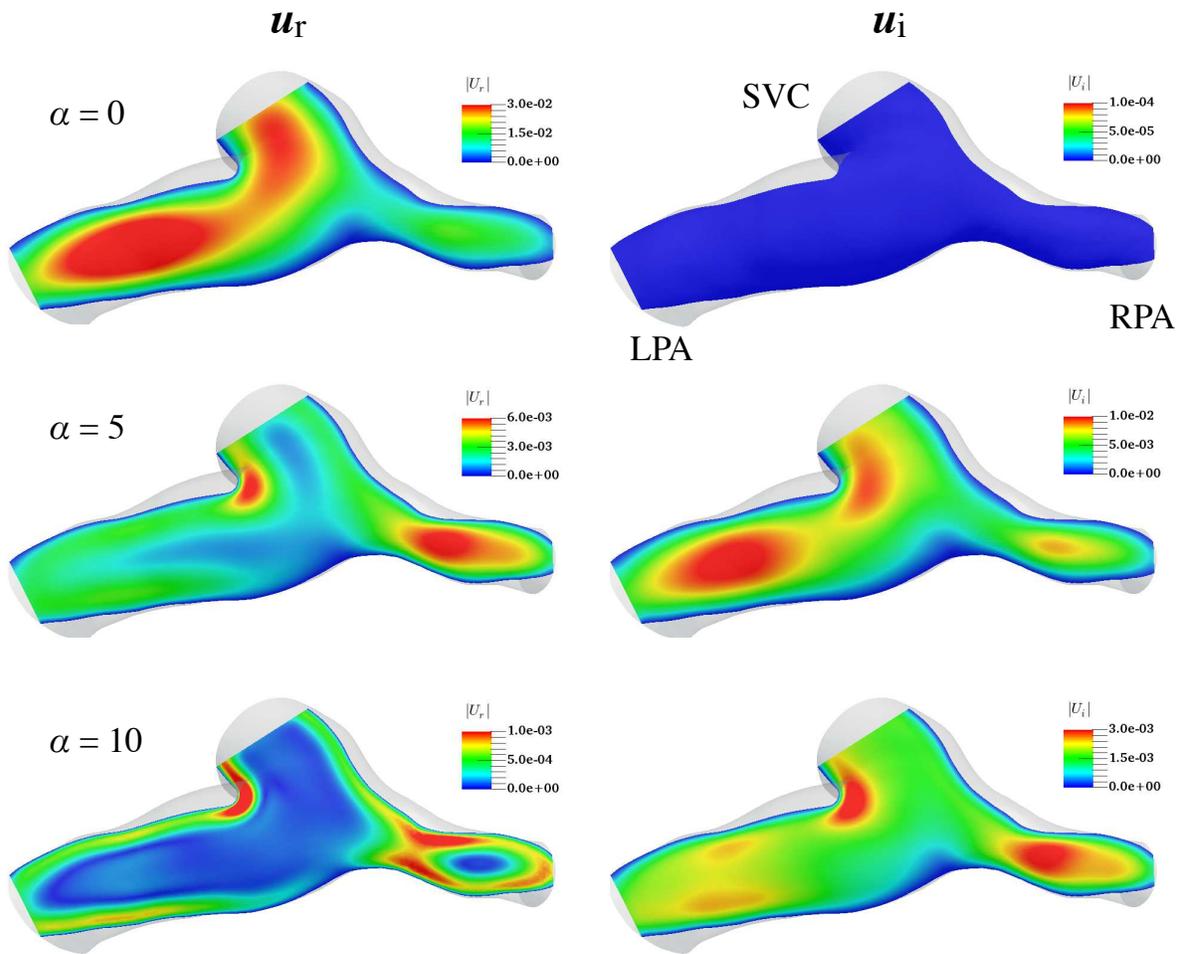


Figure 6: The real (left column) and imaginary (right column) components of velocity shown on a 2D slice of the Glenn geometry. The first row corresponds to the steady flow ($\alpha = 0$), the second row corresponds to moderately oscillatory flow ($\alpha = 5$), and the last row corresponds to a highly oscillatory flow ($\alpha = 10$).

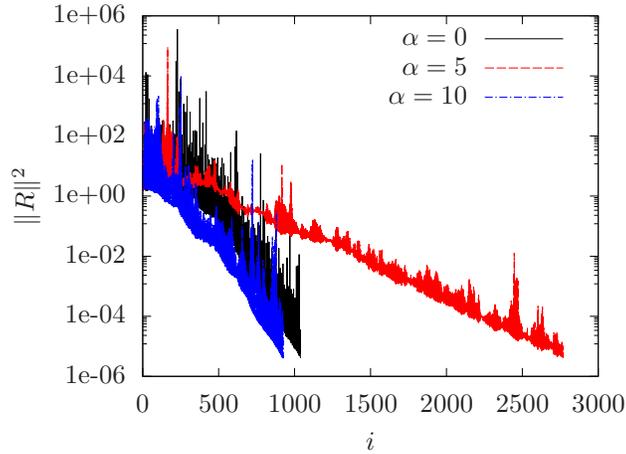


Figure 7: The norm of the residual in the conjugate gradient solver as a function of the iteration number. The linear system is obtained from the Glenn geometry at three different Womersley numbers (solid; $\alpha = 0$, dashed; $\alpha = 5$, and dashed-dot; $\alpha = 10$).

10 are shown in Figure 7. Even though there are intermittent increases in the error, all cases show an exponential convergence rate with $\alpha = 5$ case requiring the largest number of iterations.

Since there is no closed-form solution available for the assessment of the accuracy of this case, the net flow through three branches (SVC, LPA, and RPA) is computed to obtain a measure of the error in the solution of the continuity equation (Figure 8). It is observed that this error is linearly proportional to the tolerance by which the linear system of equations is solved. Varying the Womersley number has little effect on this error, demonstrating the robustness of the proposed method for a wide range of oscillatory flows.

In terms of cost, it takes a full day and 3,000 CPU-hours to simulate this case on a 128 processor cluster using a conventional CFD solver [1]. These numbers are reduced to 5.75 seconds and 0.2 CPU-hours when using the proposed formulation (considering the most expensive case of $\alpha = 5$). This four orders of magnitude reduction in wall clock time and computational cost is roughly equal to the number of time steps for the conventional solver, which is 10,000 (to simulate a total of 5 cycles to ensure cycle-to-cycle convergence). As stated earlier, this large gap in wall-clock time is independent of the number of computed modes as these computations are embarrassingly parallelizable across different modes.

4 Conclusions

A new technique was proposed for solving the Stokes equations in the frequency domain. This method only relied on real arithmetics and allowed for the use of similar order shape functions for pressure and velocity unknowns. The method was built around the addition of a stabilization term in the form of the Laplacian of pressure to the continuity equation (Eq. (7)). This stabilization term that was derived systematically from the momentum equation produced a complex-valued stabilization parameter τ (Eq. (9)). Varying the single adjustable parameter that appeared in its definition showed that it had little effect on the accuracy of the overall scheme. The results also showed the robustness of the proposed method against variation in the Womersley number, mesh size, and geometrical complexity. The proposed method was very computationally efficient, enabling a typical simulation, which otherwise takes hours to simulate using a conventional solver, to be simulated in seconds.

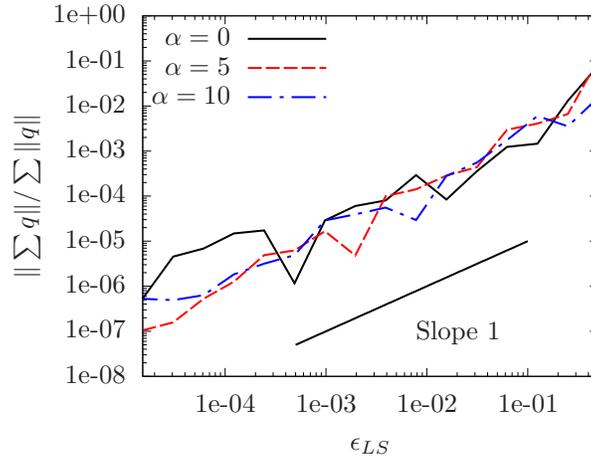


Figure 8: The imbalance between the total flow going in and out of the Glenn geometry (i.e., a measure of error in satisfying the conservation of mass) as a function of linear solver tolerance at three different Womersley numbers (solid; $\alpha = 0$, dashed; $\alpha = 5$, and dashed-dot; $\alpha = 10$).

References

- [1] Chenwei Meng, Anirban Bhattacharjee, and Mahdi Esmaily. A scalable spectral Stokes solver for simulation of time-periodic flows in complex geometries. *Journal of Computational Physics*, 445:110601, 2021.
- [2] Olga Alexandrovna Ladyzhenskaya. *The mathematical theory of viscous incompressible flow*, volume 2. Gordon and Breach New York, 1969.
- [3] Ivo Babuska. Error-bounds for finite element method. *Numerische Mathematik*, 16(4):322–333, 1971.
- [4] Franco Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 8(R2):129–151, 1974.
- [5] Yuri Bazilevs, Kenji Takizawa, and Tayfun E Tezduyar. *Computational fluid-structure interaction: methods and applications*. John Wiley & Sons, 2013.
- [6] Mahdi Esmaily, Yuri Bazilevs, and Alison Marsden. Impact of data distribution on the parallel performance of iterative linear solvers with emphasis on CFD of incompressible flows. *Computational Mechanics*, 55(1):93–103, 2015.
- [7] Youcef Saad and Martin H Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.
- [8] Mahdi Esmaily, Yuri Bazilevs, and Alison Marsden. A bi-partitioned iterative algorithm for solving linear systems arising from incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering*, 286(1):40–62, 2015.

- [9] Farzin Shakib, Thomas JR Hughes, and Zdeněk Johan. A new finite element formulation for computational fluid dynamics: X. the compressible euler and navier-stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 89(1-3):141–219, 1991.
- [10] Y Bazilevs, VM Calo, JA Cottrell, TJR Hughes, A Reali, and G Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197(1-4):173–201, 2007.
- [11] George Karypis and Vipin Kumar. MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 4.0. <http://www.cs.umn.edu/~metis>, 2009.
- [12] John R Womersley. Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known. *The Journal of Physiology*, 127(3):553, 1955.
- [13] Gregory Arbia, Chiara Corsini, Mahdi Esmaily, Alison L Marsden, Francesco Migliavacca, Giancarlo Pennati, Tain-Yen Hsia, Irene E Vignon-Clementel, Modeling of Congenital Hearts Alliance (MOCHA) Investigators, et al. Numerical blood flow simulation in surgical corrections: what do we need for an accurate analysis? *Journal of Surgical Research*, 186(1):44–55, 2014.

