Received: 12 August 2021 | Revised: 30 September 2021 | Accepted: 27 November 2021

DOI: 10.1002/cnm.3558

BASIC RESEARCH

WILEY

Projection-based model reduction for the immersed boundary method

Yushuang Luo | **Xiantao Li** | **Wenrui Hao**

Department of Mathematics, The Pennsylvania State University, University Park, Pennsylvania, USA

Correspondence
Wenrui Hao, Department of Mathematics, The Pennsylvania State University, University Park, PA 16802, USA.
Email: wxh64@psu.edu

Funding information
National Science Foundation, Grant/Award Numbers: DMS-1953120, DMS-2052685

Abstract

Fluid–structure interactions are central to many biomolecular processes, and they impose a great challenge for computational and modeling methods. In this paper, we consider the immersed boundary method (IBM) for biofluid systems, and to alleviate the computational cost, we apply reduced-order techniques to eliminate the degrees of freedom associated with the large number of fluid variables. We show how reduced models can be derived using Petrov–Galerkin projection and subspaces that maintain the incompressibility condition. More importantly, the reduced-order model (ROM) is shown to preserve the Lyapunov stability. We also address the practical issue of computing coefficient matrices in the ROM using an interpolation technique. The efficiency and robustness of the proposed formulation are examined with test examples from various applications.

KEYWORDS

fluid–structure interaction, immersed boundary method, model reduction

1 | INTRODUCTION

Biofluid dynamics, the study of cellular movement in biological fluid flow, is essential for understanding how the cellular behavior changes within living tissues.[1] With the rapid development of scientific computing algorithms, mathematical modeling and numerical simulations have become an indispensable approach for studying biofluid dynamics. Specifically, the interaction between cell structures and the surrounding fluid flow is of the utmost importance. Mathematically speaking, this belongs to a large class of problems known as the fluid–structure interactions (FSI), often described by coupling the incompressible Navier–Stokes equations with solid equations. A variety of computational and modeling techniques has been developed for FSIs, and they have been successfully implemented in studying (among many other applications) biology and biomedical diseases.[2–9] In order to numerically solve FSI problems, several numerical methods have been developed to represent/track the interface movement explicitly, such as the boundary element method (BEM),[10–12] the immersed boundary method (IBM),[13–16] the immersed interface method (IIM),[17] the fictitious domain method (FDM),[18,19] and the front tracking method (FTM).[20,21] Another alternative approach to solve the FSI problem is to capture the interface dynamics implicitly by evolving a scalar function defined on the whole domain. The level-set method,[22] the phase-field method,[23] and the implicit boundary integral method[24] are important examples. However, direct simulations based on these methods tend to be time-consuming and computationally expensive for the prediction and analysis of long-term dynamics (although the short-term prediction is certainly feasible). Often of interest in biology, is the structure dynamics, which, due to its observability, is easy to validate either experimentally or computationally.[25] In addition, there are many important scenarios where the cell structure is immersed in a large fluid environment, and simulating the entire system becomes computationally challenging.

The purpose of this paper is to explore an alternative to reduce the computational cost using reduced-order techniques, which have been applied to a wide variety of problems in science and engineering.[26–29] Reduced-order modeling is concerned with large-dimensional dynamical systems with low-dimensional input and output, and the main objective is to construct reduced models that can approximate the mapping from the input directly to the output. The present FSI problem will be formulated as a reduced-order problem, where the input is the force exerted from the structure and the output is the local velocity of the structure. As a proof-of-concept, we utilize the conventional IBM model.[13] Specifically, incompressible unsteady Stokes flows are considered, together with the no-slip interface condition enforced on the immersed structure. But it is also important to point out that there have been many extensions of the original IBM framework with different treatments for the Lagrangian equations of motion or the fluid dynamics,[30–32] and reduced-order modeling can be considered in those settings as well. Our starting point is a semi-discrete representation of the IBM model, so that the dynamics of fluid and structure motion can be expressed as coupled ODEs, which can then be placed in the reduced-order modeling framework. Then we derive the effective mapping from the structure force to the local velocity, which completely eliminates the fluid variables.

To construct specific reduced models that do not involve the fluid dynamics explicitly, we first construct subspaces that preserve the incompressibility condition, followed by a Petrov–Galerkin projection. We show that the choice of the subspaces ensures certain interpolation conditions on the underlying transfer function. An important departure from standard reduced-order problems is that in IBM, the structure is also evolving continuously. As a result, the subspaces are varying in time. This poses some challenges as the coefficient matrices of the reduced models need to be updated frequently. To circumvent this issue, we observe the connection between the those matrices and the Green's function of the Laplace equation. More specifically, the entries of those matrices are tied to the nodal points on the structure. When the two points are far apart, the corresponding entry can be well approximated by the Green's function. On the other hand, for points that are within some cut-off distance, the computation can be done in advance, and then in the simulation, those entries can be computed by interpolation. We show that such a strategy avoids repetitive computation of those coefficient matrices and it can speed up the computation considerably.

The remaining part of the paper is organized as follows: in Section 2, we introduce the full-order model (FOM) in the IBM setup; in Section 3, we formulate the reduced-order model (ROM); several numerical examples are used to compare both FOM and ROM in Section 4; then the conclusion is drawn in Section 5.

2 | FULL-ORDER MODEL

In this section, we briefly review the mathematical formulation of the IBM and derive its semi-discrete representation, which will serve as the FOM.

2.1 | Mathematical formulation of the IBM

The IBM is intended for the computer simulation of FSI, especially in biological fluid dynamics. It is mathematically defined by a set of differential equations involving a mixture of Eulerian and Lagrangian descriptions, linked by the Dirac delta function. The dynamics of the fluid is described in terms of the velocity $\boldsymbol{u}(\boldsymbol{x},t)$ and the pressure $p(\boldsymbol{x},t)$ on an Eulerian coordinate for $\boldsymbol{x}\in\Omega$, where $\Omega\subset\mathbb{R}^d$, $d=2\text{or}3$, represents the fluid domain. The immersed structures, on the other hand, are handled in a Lagrangian coordinate as a parametric curve or surface $X(\boldsymbol{s},t)$. Specifically, $X(\boldsymbol{s},t)$ represents the position at time t in Cartesian coordinates of the structure point labeled by $\boldsymbol{s}\in\Gamma$, where $\Gamma\subset\mathbb{R}^{d-1}$ is the parameter space. In this work, we focus on two-dimensional flows where the structure is described as a parametric curve. In this case $\boldsymbol{x}\in\Omega\subset\mathbb{R}^2$ and s is a scalar parameter. The formulation is mostly algebraic. Therefore, the extension to high dimensional cases is straightforward. Assuming constant density, the time-dependent Stokes equation is used to model the incompressible flow

$$\rho\frac{\partial\boldsymbol{u}}{\partial t}=-\nabla p+\mu\nabla^2\boldsymbol{u}+\boldsymbol{f}, \tag{1}$$

$$\nabla\cdot\boldsymbol{u}=0, \tag{2}$$

where ρ and μ are the fluid density and viscosity, respectively. The body force $\boldsymbol{f}$ exerted by the structure on the fluid is defined as

$$\boldsymbol{f}(\boldsymbol{x},t) = \int_{\Gamma} \boldsymbol{F}(s,t)\delta(\boldsymbol{x}-\boldsymbol{X}(s,t))\,ds, \tag{3}$$

where $\delta(\boldsymbol{x})$ is the Dirac delta function. In addition, $\boldsymbol{F}(s,t)$ denotes the force density on the immersed structure, defined as

$$\boldsymbol{F}(s,t) = \mathcal{F}[\boldsymbol{X}(s,t)], \tag{4}$$

where $\mathcal{F}$ is a functional of the IBM configuration. Spring forces, bending resistance or any other type of behavior (area and volume conservation constraints) can be built into this functional to embody the physics of the immersed structure under different circumstances.[19,33] We give detailed description of the force density in Section 4 in the numerical examples for various systems.

Assuming an over-damped structure, the immersed boundary must move with the local fluid velocity:

$$\frac{\partial \boldsymbol{X}(s,t)}{\partial t} = \boldsymbol{u}(\boldsymbol{X}(s,t),t) = \int_{\Omega} \boldsymbol{u}(\boldsymbol{x},t)\delta(\boldsymbol{x}-\boldsymbol{X}(s,t))\,d\boldsymbol{x}. \tag{5}$$

This last equation is nothing other than the no-slip condition written as a delta function convolution.

2.2 | The semi-discrete equations

To derive a semi-discrete representation of the IBM, we use the finite difference discretization.[13] Other numerical methods can also be applied to discretize the IBM, for example, the finite element method[34] and the finite volume method,[35] in which the state space consists of nodal values.

In this work, fluid variables are discretized on a uniform staggered Eulerian grid, denoted Ω_h; and the structures are discretized on an independent Lagrangian grid, denoted by Γ_h (Figure 1). The Eulerian grid points are of the form $\boldsymbol{x} = \boldsymbol{j}h$, where $\boldsymbol{j} = (j_1, j_2)$ is a two-dimensional vector with integer components and h is the Eulerian grid size. The Lagrangian grid is a set of s of the form $k\Delta s$, where k has integer components. The following restriction is imposed to avoid leak,[13]

$$|\,\boldsymbol{X}(s+\Delta s,t) - \boldsymbol{X}(s,t)\,| < \frac{h}{2}, \tag{6}$$

for all s. First, the semi-discrete equations for (1) and (2) form a system of linear differential-algebraic equations (DAEs)

$$\rho\dot{\boldsymbol{u}}_h(t) = -N\boldsymbol{p}_h(t) + \mu G\boldsymbol{u}_h(t) + \boldsymbol{f}_h(t), \tag{7}$$

$$0 = M\boldsymbol{u}_h(t), \tag{8}$$

where

$$\boldsymbol{u}_h(t) = \begin{bmatrix} \vdots \\ u_h(\boldsymbol{x},t) \\ \vdots \end{bmatrix} \in \mathbb{R}^{n_u}, \quad \boldsymbol{p}_h(t) = \begin{bmatrix} \vdots \\ p_h(\boldsymbol{x},t) \\ \vdots \end{bmatrix} \in \mathbb{R}^{n_p}, \quad \boldsymbol{f}_h(t) = \begin{bmatrix} \vdots \\ f_h(\boldsymbol{x},t) \\ \vdots \end{bmatrix} \in \mathbb{R}^{n_u}, \quad \forall \boldsymbol{x} \in \Omega_h,$$

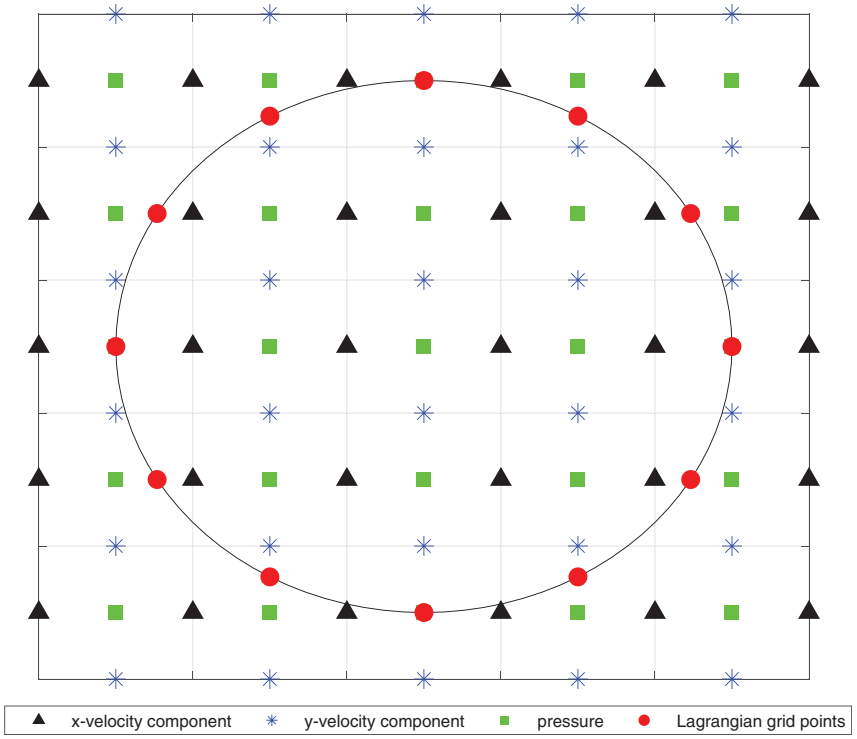

FIGURE 1 The fluid field variables are defined on a regular staggered grid. The structure variables are defined on an Lagrangian grid

are vectors of discrete velocity field, pressure and body force, respectively. $G \in \mathbb{R}^{n_u,n_u}$ is the discrete Laplace operator. Matrices $N \in \mathbb{R}^{n_u,n_p}$ and $M \in \mathbb{R}^{n_p,n_u}$ are the discrete gradient and divergence operators, respectively.

Second, the integrals in (3) and (5) are replaced by the following sums over the appropriate grid points,

$$f_h(\boldsymbol{x},t) = \sum_{s \in \Gamma_h} F_h(s,t)\ \delta_r(\boldsymbol{x} - X_h(s,t))\ \Delta s \quad \forall \boldsymbol{x} \in \Omega_h, \tag{9}$$

$$\dot{X}_h(s,t) = \sum_{\mathbf{x} \in \Omega_h} u_h(\boldsymbol{x},t)\ \delta_r(\boldsymbol{x} - X_h(s,t))\ h^2 \quad \forall s \in \Gamma_h, \tag{10}$$

where $F_h(s,t)$ is the discrete Lagrangian force density associated with the structure point labeled s, obtained by discretizing (4). In addition, a function $\delta_r(\boldsymbol{x})$ that is nonsingular for each r but approaches $\delta(\boldsymbol{x})$ as $r \to 0$ is needed. There are many ways to construct such δ_r. We choose a radially symmetric function with compact support as follows,[36]

$$\delta_r(\boldsymbol{x}) = \begin{cases} C_r\left[1 + \dfrac{|\mathbf{x}|^2}{r^3}(2|\boldsymbol{x}| - 3r)\right] & |\boldsymbol{x}| \le r, \\ 0 & |\boldsymbol{x}| > r, \end{cases} \tag{11}$$

where the normalizing constant $C_r = \frac{10}{3\pi} r^{-2}$ depends on r and the space dimension ($C_r = \frac{15}{4\pi} r^{-3}$ in 3D). For computational efficiency, we choose $r = 2h$ in all our numerical experiments, as suggested for IBM.[13]

Meanwhile, Equations (9) and (10) can be put into matrix–vector form:

$$\boldsymbol{f}_h(t) = B\boldsymbol{F}_h(t), \tag{12}$$

$$\dot{\boldsymbol{X}}_h(t) = h^2 B^T \boldsymbol{u}_h(t), \tag{13}$$

where

$$\boldsymbol{F}_h(t) = \begin{bmatrix} \vdots \\ F_h(s,t)\ \Delta s \\ \vdots \end{bmatrix} \in \mathbb{R}^{n_s}, \quad \boldsymbol{X}_h(t) = \begin{bmatrix} \vdots \\ X_h(s,t) \\ \vdots \end{bmatrix} \in \mathbb{R}^{n_s}, \quad \forall s \in \Gamma_h,$$

are the discrete representations of the structure position and the Lagrangian force density. Using natural arrangement of the fluid variables, $B \in \mathbb{R}^{n_u,n_s}$ can be constructed as a block matrix

$$B = \begin{bmatrix} B_1 & 0 \\ 0 & B_2 \end{bmatrix} \quad \left(\begin{bmatrix} B_1 & 0 & 0 \\ 0 & B_2 & 0 \\ 0 & 0 & B_3 \end{bmatrix} \quad \text{in 3D} \right). \tag{14}$$

A column of each B_k consists of evaluations of $\delta_r(\boldsymbol{x} - X(s,t))$ for a fixed $X(s,t)$ on grid points $\boldsymbol{x}$ that store one component of the fluid velocity variables. For example, the i,j-entry of B_1 is $\delta_r(\boldsymbol{x}_i - X(s_j,t))$, where $\boldsymbol{x}_i$ is the grid point that stores the ith fluid velocity in the x-direction. Note that B_k's are not identical because they correspond to different Eulerian grid points $\boldsymbol{x}$. For example in Figure 1, $\boldsymbol{x}$ in B_1 are the points marked by filled triangles, while $\boldsymbol{x}$ in B_2 are marked by stars. We also point out that B is time dependent due to its dependence on $\boldsymbol{X}_h(t)$.

Lastly, by substituting (12) into (7) for $\boldsymbol{f}_h(t)$, we get the following DAE system which we shall refer to as the FOM

$$\rho \dot{\boldsymbol{u}}_h(t) = -N\boldsymbol{p}_h(t) + \mu G\boldsymbol{u}_h(t) + B\boldsymbol{F}_h(t), \tag{15}$$

$$0 = M\boldsymbol{u}_h(t), \tag{16}$$

$$\dot{\boldsymbol{X}}_h(t) = h^2 B^T \boldsymbol{u}_h(t). \tag{17}$$

In general, the number of structure variables is much less than the number of fluid variables, that is, $n_s \ll n_u$. In fact, δ_r having compact support means only a small fraction of the Eulerian grid points are directly interacting with the structure. If one is only interested in the motion of the structure, that is, $\boldsymbol{X}_h(t)$, solving the system (15)–(17) becomes a reduced-order problem,[37] where $\boldsymbol{F}_h(t)$ is the low-dimensional input and $\dot{\boldsymbol{X}}_h(t)$ is the low-dimensional output.

3 | REDUCED-ORDER MODEL

To construct our ROM, we start by transforming the DAE system (15)–(17) to a coupled ODE system. Multiplying (15) by M to the left and using (16), we rewrite (15) as:

$$0 = -MN\boldsymbol{p}_h(t) + \mu MG\boldsymbol{u}_h(t) + MB\boldsymbol{F}_h(t). \tag{18}$$

Assuming MN is nonsingular, it follows that

$$\boldsymbol{p}_h(t) = (MN)^{-1}(\mu MG\boldsymbol{u}_h(t) + MB\boldsymbol{F}_h(t)). \tag{19}$$

Substituting (19) into (15) for $\boldsymbol{p}_h(t)$, one gets

$$\rho \dot{\boldsymbol{u}}_h(t) = \mu QG\boldsymbol{u}_h(t) + QB\boldsymbol{F}_h(t), \tag{20}$$

where

$$Q = I - N(MN)^{-1}M \tag{21}$$

is an oblique projection. It is worth emphasizing here that the discrete gradient operator N and the discrete divergence operator M are adjoint of each other with different dimensions.

An ODE system is then obtained from (15),

$$\dot{\boldsymbol{u}}_h(t) = \frac{\mu}{\rho} QG\mathbf{u}_h(t) + \frac{1}{\rho} QB\boldsymbol{F}_h(t), \tag{22}$$

$$\dot{\boldsymbol{X}}_h(t) = h^2 B^T \boldsymbol{u}_h(t). \tag{23}$$

We assume $\boldsymbol{u}_h(0) = \mathbf{0}$ in the rest of this section. Nonzero initial values can be handled by linear superposition

$$\boldsymbol{u}_h(t) = \boldsymbol{u}_h^{(0)}(t) + \boldsymbol{u}_h^{(1)}(t), \tag{24}$$

in which $\boldsymbol{u}_h^{(0)}(0) = \mathbf{0}$ and $\boldsymbol{u}_h^{(1)}(0) = \boldsymbol{u}_h(0)$. Then one can decompose (22) and (23) to

$$\dot{\boldsymbol{u}}_h^{(0)}(t) = \frac{\mu}{\rho} QG\boldsymbol{u}_h^{(0)}(t) + \frac{1}{\rho} QB\boldsymbol{F}_h(t), \tag{25}$$

$$\dot{\boldsymbol{u}}_h^{(1)}(t) = \frac{\mu}{\rho} QG\boldsymbol{u}_h^{(1)}(t), \tag{26}$$

$$\dot{\boldsymbol{X}}_h(t) = h^2 B^T \left(\boldsymbol{u}_h^{(0)}(t) + \boldsymbol{u}_h^{(1)}(t) \right). \tag{27}$$

The dynamics of $\boldsymbol{u}_h^{(1)}(t)$, which has nonzero initial value, is described by a first-order linear ODE, without interactions with the immersed structure. It can be solved separately in advance, or in some cases, it can be resolved analytically.

We consider a general Galerkin projection of (22), motivated by its success in reduced-order problems.[26,37] More specifically, we seek $\widetilde{\boldsymbol{u}}_h(t)$ in a subspace, spanned by the columns of a tall matrix V, as an approximation for $\boldsymbol{u}_h(t)$, such that for any $\boldsymbol{w}(t)$ in a test space, spanned by the columns of a tall matrix W, we have

$$\left(\dot{\widetilde{\boldsymbol{u}}}_h(t) - \frac{\mu}{\rho} QG\widetilde{\boldsymbol{u}}_h(t) - \frac{1}{\rho} QB\boldsymbol{F}_h(t), \boldsymbol{w}(t) \right) = 0. \tag{28}$$

Note that the subspaces are not necessarily fixed, which means that the matrices V and W are generally time-dependent. This point will be addressed in Section 3.2.

In a matrix–vector form, the approximate solution is written as

$$\widetilde{\boldsymbol{u}}_h(t) = V\boldsymbol{z}(t). \tag{29}$$

Then the Galerkin projection yields a reduced-order equation

$$W^T(\dot{V\boldsymbol{z}})(t) = \frac{\mu}{\rho} W^T QGV\boldsymbol{z}(t) + \frac{1}{\rho} W^T QB\boldsymbol{F}_h(t). \tag{30}$$

Thus we obtain an ROM of (22) and (23):

$$\dot{\boldsymbol{z}}(t) = M_0^{-1} M_1 \boldsymbol{z}(t) + M_0^{-1} M_2 \boldsymbol{F}_h(t), \tag{31}$$

$$\dot{\boldsymbol{X}}_h(t) \approx h^2 B^T V \boldsymbol{z}(t), \tag{32}$$

where the matrices are given by,

$$M_0 = W^T V, \quad M_1 = \frac{\mu}{\rho} W^T Q G V - W^T \dot{V}, \quad M_2 = \frac{1}{\rho} W^T Q B, \tag{33}$$

assuming M_0 is nonsingular. The computation of the coefficient matrices M_0, M_1, and M_2 depends on V and W. In the rest of this section, we first discuss our choice for the subspaces V, W, and their properties. Then we demonstrate how an interpolation procedure can help accelerate the computation of the coefficients by exploiting the connection between the matrix entries and the Green's function.

3.1 | Subspace selection

We propose the following choice of V and W,

$$V = QB, \quad W = B. \tag{34}$$

For later reference, note that both subspaces vary in time. The resulting coefficient matrices are given by

$$M_0 = B^T Q B, \quad M_1 = \frac{\mu}{\rho} B^T Q G Q B - B^T Q \dot{B}, \quad M_2 = \frac{1}{\rho} B^T Q B = \frac{1}{\rho} M_0. \tag{35}$$

In principle, one can use higher dimensional Krylov subspaces (V and W with more columns), followed by Lanczos orthogonalization algorithms,[26,37,38] to improve the accuracy of the ROM. Specifically, we shall see in the following discussion that our choice satisfies two interpolation conditions. Higher dimensional Krylov subspaces are able to interpolate the transfer function more accurately by enforcing more interpolation conditions, but at the expense of a reduced computational speedup. From the numerical tests, our observation is that the current subspaces achieve a good balance between accuracy and efficiency.

3.1.1 | Transfer function approximation

We first show the accuracy property of our choice of subspaces. This can be understood by solving the linear ODE (22) analytically for $\boldsymbol{u}_h$, which yields,

$$\boldsymbol{u}_h(t) = \frac{1}{\rho} \int_0^t \exp\left[\frac{\mu}{\rho}(t-\tau) Q G\right] Q B \boldsymbol{F}_h(\tau)\, d\tau. \tag{36}$$

Note that we assume zero initial condition as discussed before. Plugging (36) into (23) gives

$$\dot{\boldsymbol{X}}_h(t) = \int_0^t \phi(t-\tau) \boldsymbol{F}_h(\tau)\, d\tau, \tag{37}$$

where $\phi(t)$ denotes the transfer function,

$$\phi(t) = \frac{h^2}{\rho} B^T \exp\left[\frac{\mu}{\rho} tQG\right] QB. \tag{38}$$

A similar calculation for the ROM (31) and (32) shows that:

$$\dot{\boldsymbol{X}}_h(t) \approx \int_0^t \phi_{\text{red}}(t-\tau)\boldsymbol{F}_h(\tau)\, d\tau, \tag{39}$$

where the transfer function $\phi_{\text{red}}(t)$ of the ROM is given by,

$$\begin{aligned} \phi_{\text{red}}(t) &= h^2 B^T V \exp\left[tM_0^{-1}M_1\right] M_0^{-1} M_2, \\ &= \frac{h^2}{\rho} B^T QB \exp\left[t\left(B^T QB\right)^{-1}\left(\frac{\mu}{\rho} B^T QGQB - B^T Q\dot{B}\right)\right]. \end{aligned} \tag{40}$$

$\phi_{\text{red}}(t)$ is expected to approximate $\phi(t)$ in the sense that,

$$\phi_{\text{red}}(0) = \phi(0), \tag{41}$$

$$\dot{\phi}_{\text{red}}(0) = \dot{\phi}(0). \tag{42}$$

The equality (41) follows immediately from evaluating (38) and (40) at $t = 0$. Differentiating (38) and (40) at $t = 0$ yields

$$\dot{\phi}_{\text{red}}(0) = \frac{\mu h^2}{\rho^2} B^T QB\left(B^T QB\right)^{-1} B^T QGQB = \frac{\mu h^2}{\rho^2} B^T QGQB = \dot{\phi}(0). \tag{43}$$

In the above calculation, we have treated B as a constant matrix. The reason is that we are only concerned with a small time interval $[0, t]$, typically with the size of one time step. In numerical simulations, the matrix B is usually treated as constant when advancing one time step.

3.1.2 | Enforcing incompressibility

Another essential property of the full model is the incompressibility of the fluid. Recall that M is the discrete divergence operator. The approximate fluid solution,

$$\widetilde{\boldsymbol{u}}_h(t) = V\boldsymbol{z}(t), \tag{44}$$

is incompressible if

$$M\widetilde{\boldsymbol{u}}_h(t) = MV\boldsymbol{z}(t) = 0. \tag{45}$$

A quick calculation verifies that our choice of $V = QB$ satisfies this constraint:

$$MV = MQB = M\left(I - N(MN)^{-1}M\right)B = \left(M - MN(MN)^{-1}M\right)B = 0. \tag{46}$$

Therefore, the incompressibility property is preserved in the ROM.

3.1.3 | Lyapunov stability

The ROM also preserves Lyapunov stability of the FOM with our choice of subspaces. We first show the stability of the FOM. We assume the discrete Lagrangian force density $\boldsymbol{F}_h$ is given by an energy functional $W(\boldsymbol{X}_h)$ of the structure configuration, that is,

$$\boldsymbol{F}_h(t) = -\nabla_{X_h} W(\boldsymbol{X}_h(t)). \tag{47}$$

We also assume the discrete gradient and divergence operators satisfy

$$M = N^T, \tag{48}$$

such that

$$Q = I - \Sigma = I - N\left(N^T N\right)^{-1} N^T \tag{49}$$

is an orthogonal projection. We now define the following Lyapunov functional for the FOM consisting of the kinetic and the elastic energy,

$$V(\boldsymbol{u}_h(t), \boldsymbol{X}_h(t)) = \frac{1}{2}\boldsymbol{u}_h(t)^T Q \boldsymbol{u}_h(t) + \frac{1}{h^3\rho} W(\boldsymbol{X}_h(t)). \tag{50}$$

We have $V(\boldsymbol{u}_h(t), \boldsymbol{X}_h(t)) \geq 0$ because Q, as a projection, is positive semidefinite with eigenvalues 0 or 1. In particular, notice that $Q^2 = Q$ and $Q^T = Q$. In addition, the divergence-free condition implies that $Q\boldsymbol{u}_h(t) = \boldsymbol{u}_h(t)$. A direct calculation shows that

$$\begin{aligned} \dot{V}(\boldsymbol{u}_h(t), \boldsymbol{X}_h(t)) &= \boldsymbol{u}_h(t)^T Q^2 \left(\frac{\mu}{\rho} G \boldsymbol{u}_h(t) + \frac{1}{\rho} B \boldsymbol{F}_h(t)\right) - \frac{1}{\rho}\boldsymbol{u}_h(t)^T B \boldsymbol{F}_h(t)) \\ &= \frac{\mu}{\rho}\boldsymbol{u}_h(t)^T Q G \boldsymbol{u}_h(t) + \frac{1}{\rho}\boldsymbol{u}_h(t)^T Q B \boldsymbol{F}_h(t)) - \frac{1}{\rho}\boldsymbol{u}_h(t)^T B \boldsymbol{F}_h(t)) \\ &= \frac{\mu}{\rho}\boldsymbol{u}_h(t)^T G \boldsymbol{u}_h(t) \leq 0, \end{aligned} \tag{51}$$

since the discrete Laplace operator G is negative semidefinite. This implies the Lyapunov stability of the FOM.

The Lyapunov functional for the ROM is defined as follows

$$V_r(\boldsymbol{X}_h(t)) = \frac{h^2}{\rho} W(\boldsymbol{X}_h(t)) + \frac{1}{2}\dot{\boldsymbol{X}}_h(t)^T \left(B^T Q B\right)^{-1} \dot{\boldsymbol{X}}_h(t). \tag{52}$$

It is now clear that $V_r(\boldsymbol{X}_h(t)) \geq 0$ holds for all t since $B^T Q B$ is positive semidefinite.

To prove $\dot{V}_r(\boldsymbol{X}_h(t)) \leq 0$, we start by rewriting the ROM (31) and (32) as a second-order ODE of $\boldsymbol{X}_h$. Note that (32) and (34) imply $\boldsymbol{z}(t) = h^{-2}\left(B^T Q B\right)^{-1} \dot{\boldsymbol{X}}_h(t)$ in the ROM. Using the symmetry of Q, one has,

$$\begin{aligned} \ddot{\boldsymbol{X}}_h &= h^2[(\dot{B^T} Q B)\boldsymbol{z} + B^T Q B \dot{\boldsymbol{z}}(t)] = h^2\left(B^T Q \dot{B} + \frac{\mu}{\rho} B^T Q G Q B\right)\boldsymbol{z} + \frac{h^2}{\rho} B^T Q B \boldsymbol{F}_h(\boldsymbol{X}_h(t)) \\ &= \left(B^T Q \dot{B} + \frac{\mu}{\rho} B^T Q G Q B\right)\left(B^T Q B\right)^{-1} \dot{\boldsymbol{X}}_h + \frac{h^2}{\rho} B^T Q B \boldsymbol{F}_h(\boldsymbol{X}_h(t)). \end{aligned} \tag{53}$$

Then the following calculation shows that V_r is nonincreasing,

$$
\begin{aligned}
\dot{V}_r(\boldsymbol{X}_h(t)) &= -\frac{h^2}{\rho}\dot{\boldsymbol{X}}_h^T\boldsymbol{F}_h + \dot{\boldsymbol{X}}_h\left(B^TQB\right)\ddot{\boldsymbol{X}}_h + \frac{1}{2}\dot{\boldsymbol{X}}_h\left(\dot{B^TQB}\right)^{-1}\dot{\boldsymbol{X}}_h \\
&= \frac{\mu}{\rho}\dot{\boldsymbol{X}}_h^T\left(B^TQB\right)^{-1}\left(B^TQGQB\right)\left(B^TQB\right)^{-1}\dot{\boldsymbol{X}}_h - \frac{h^2}{\rho}\dot{\boldsymbol{X}}_h^T\boldsymbol{F}_h + \frac{h^2}{\rho}\dot{\boldsymbol{X}}_h^T\boldsymbol{F}_h \\
&+ \dot{\boldsymbol{X}}_h^T\left(B^TQB\right)^{-1}\left(B^TQ\dot{B}\right)\left(B^TQB\right)^{-1}\dot{\boldsymbol{X}}_h \\
&- \dot{\boldsymbol{X}}_h^T\left(B^TQB\right)^{-1}\left(B^TQ\dot{B}\right)\left(B^TQB\right)^{-1}\dot{\boldsymbol{X}}_h \\
&= \frac{\mu}{\rho}\dot{\boldsymbol{X}}_h^T\left(B^TQB\right)^{-1}\left(B^TQGQB\right)\left(B^TQB\right)^{-1}\dot{\boldsymbol{X}}_h \\
&= \frac{\mu}{\rho}\boldsymbol{Y}^TG\boldsymbol{Y} \le 0,
\end{aligned}
\tag{54}
$$

where we have defined $\boldsymbol{Y} := QB\left(B^TQB\right)^{-1}\dot{\boldsymbol{X}}_h$. The last inequality holds because G is negative semidefinite.

3.2 | Computing the coefficients using interpolation

Because the coefficient matrices in the ROM are in principle time dependent, they should be updated frequently during simulation. Direct matrix multiplication for this purpose is time consuming since Q is a dense matrix in $\mathbb{R}^{2n_u,2n_u}$. For example, computing M_1 in (35) has complexity $O\left(n_u^2\right)$. In the rest of this section, we propose a computationally cheaper approach using interpolation to approximate the coefficients.

We first approximate $\dot{B}$ by

$$
\dot{B}(t) \approx \frac{1}{\Delta t}(B(t) - B(t-\Delta t)). \tag{55}
$$

One could consider a higher order discretizations for $\dot{B}$ so that a method of order higher than one in time can be used to solve the ROM. Ultimately, this is a trade-off between accuracy and the offline interpolation efficiency. From the numerical tests, we will see such first order approximation of $\dot{B}$ together with the forward Euler's method provides acceptable accuracy for the numerical tests compared to the FOM.

The other observation is that the matrix M_1 is then approximated by

$$
M_1 \approx \frac{\mu}{\rho}B(t)^TQGQB(t) - \frac{1}{\Delta t}B(t)^TQB(t) + \frac{1}{\Delta t}B(t)^TQB(t-\Delta t). \tag{56}
$$

Together with $M_0 = B(t)^TQB(t)$, the following three matrices are needed for building our ROM

$$
B(t)^TQB(t), \quad B(t)^TQGQB(t), \quad B(t)^TQB(t-\Delta t). \tag{57}
$$

Since Q and G are constant matrices, the i,j-entry of any of the above matrices at time t is determined by the ith row of $B(t)^T$ and the jth column of $B(t)$ (or $B(t-\Delta t)$). Recall that each column of $B(t)$ (or row of $B(t)^T$) represents a smoothed delta function associated with a structure point. Suppose the ith column of $B(t)^T$ is associated with the Lagrangian grid point $X_l \in \mathbb{R}^d$ ($d = 2,3$) and the jth column of $B(t)^T$ (or $B(t-dt)$) is associated with $X_r \in \mathbb{R}^d$. Given the prescribed function δ_r and a fixed Eulerian grid, the i,j-entry of a coefficient matrix is uniquely determined by X_l and X_r, which can be viewed as a function from $\mathbb{R}^{2d}$ to $\mathbb{R}$. It is then natural to sample such functions before the simulation starts. As the simulation runs, coefficient matrices are updated by interpolation using precomputed samples. In this work, linear interpolation is used. Because each entry of the $2n_s - \text{by} - 2n_s$ coefficient matrix is obtained by evaluating a precomputed linear function, the complexity is typically $O\left(n_s^2\right)$, which is much smaller than the complexity of direct matrix multiplications $O\left(n_u^2\right)$, given $n_s \ll n_u$.

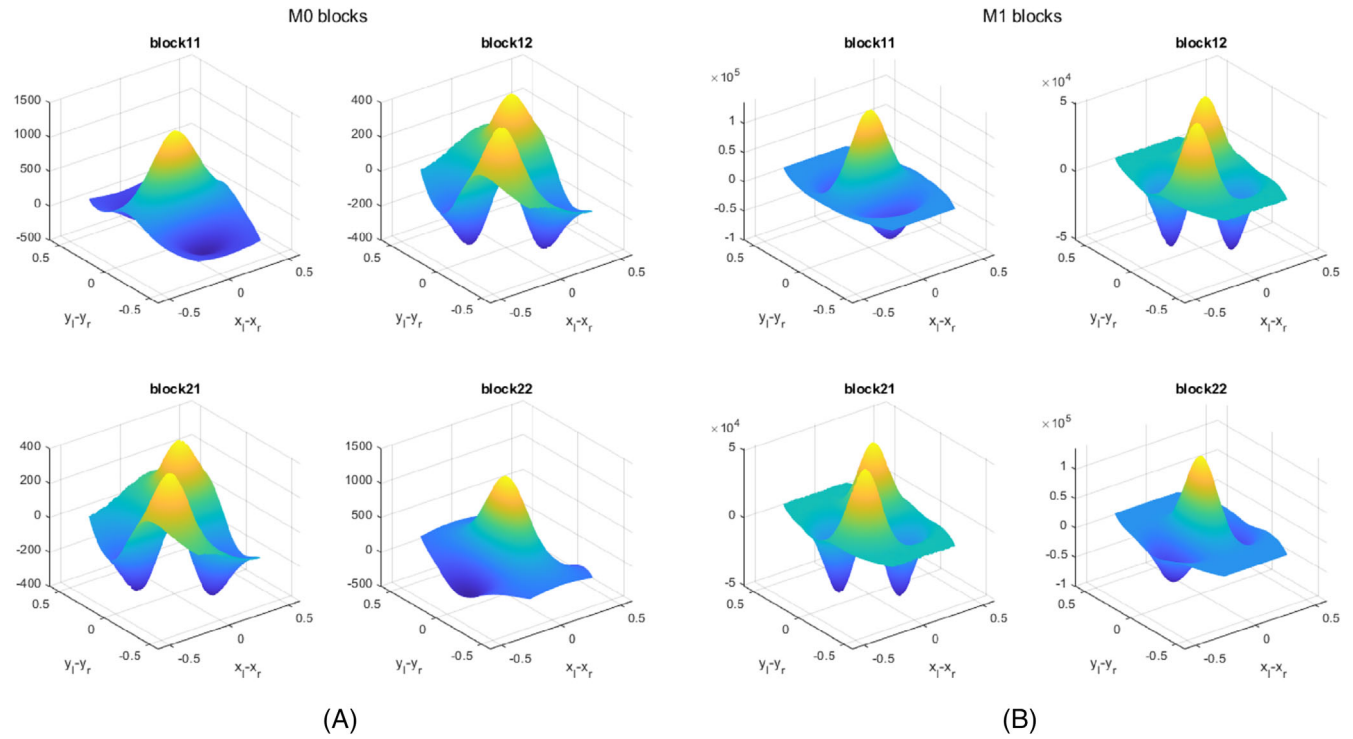

FIGURE 2 Surface plots of entries of (A) M_0 and (B) M_1 against $X_l - X_r = (x_l - x_r, y_l - y_r)$ in 2D case. Both matrices are 2-by-2 block matrices where each block corresponds to a function from $\mathbb{R}^4$ to $\mathbb{R}$. Large numbers of $(X_l, X_r) = (x_l, y_l, x_r, y_r)$ pairs are sampled so that many of them correspond to the same difference $X_l - X_r$. Then corresponding entries of M_0 and M_1 blocks are plotted against $X_l - X_r$. In each plot, we observe a single surface, indicating no multiple values. Therefore, these $\mathbb{R}^4$ functions can be considered as functions in $\mathbb{R}^2$ of $X_l - X_r$

Next, motivated by our numerical experiments illustrated in Figure 2, we show that the interpolated $2d$-dimensional functions of X_l and X_r can be well approximated by d-dimensional functions of $X_l - X_r$, that is, the relative position of the two points. Such low-dimensional approximation significantly reduces the number of samples needed for more accurate interpolations. Hence, the sampling process can also be accelerated.

Here, we provide justifications of this approach by making connections to the Green's functions. Recall that $Q = I - N(MN)^{-1}M$, where N and M are discrete gradient and divergence operators. Therefore, each entry of the matrix $B(t)^T QB(t)$ or $B(t)^T QB(t - \Delta t)$ is a numerical approximation of the integral

$$\begin{aligned} I_0 &= \int_\Omega \delta_r(\boldsymbol{x} - X_l)\left(\delta_{ij} - \partial_i \Delta^{-1} \partial_j\right)\delta_r(\boldsymbol{x} - X_r)\, d\boldsymbol{x} \\ &= \underbrace{\delta_{ij} \int_\Omega \delta_r(\boldsymbol{x} - X_l)\delta_r(\boldsymbol{x} - X_r)\, d\boldsymbol{x}}_{I_1} - \underbrace{\int_\Omega \delta_r(\boldsymbol{x} - X_l)\partial_i \Delta^{-1} \partial_j \delta_r(\boldsymbol{x} - X_r)\, d\boldsymbol{x}}_{I_2}, \end{aligned} \tag{58}$$

where $i, j = 1, \cdots, d$ and $d = 2$ or 3. δ_{ij} is the Kronecker delta function. I_1 only depends on $|X_l - X_r|$ due to our choice of δ_r. For I_2, we assume X_l and X_r are far from the boundary of Ω so the Green's function $G(\boldsymbol{x}, \boldsymbol{y})$ can be applied. Considering the limiting case of $r \to 0$, that is, $\delta_r \to \delta$, as $r \to 0$, and we arrive at,

$$\lim_{r \to 0} I_2 = -\partial_{x_i} \partial_{y_j} G(\boldsymbol{x}, \boldsymbol{y})\Big|_{\boldsymbol{x} = X_l, \boldsymbol{y} = X_r}, \tag{59}$$

which depends only on $X_l - X_r$.

Similarly, each entry of the matrix $B(t)^T QGQB(t)$ is a numerical approximation of the following integral

$$\begin{aligned} J_0 &= \int_\Omega \delta_r(\boldsymbol{x}-X_l)\left(\delta_{ij}-\partial_i\Delta^{-1}\partial_j\right)\Delta\left(1-\partial_j\Delta^{-1}\partial_j\right)\delta_r(\boldsymbol{x}-X_r)\,d\boldsymbol{x} \\ &= \underbrace{\int_\Omega \delta_r(\boldsymbol{x}-X_l)\left(\delta_{ij}-\partial_i\Delta^{-1}\partial_j\right)\Delta\delta_r(\boldsymbol{x}-X_r)\,d\boldsymbol{x}}_{J_1} - \underbrace{\int_\Omega \delta_r(\boldsymbol{x}-X_l)\left(\delta_{ij}-\partial_i\Delta^{-1}\partial_j\right)\Delta\partial_j\Delta^{-1}\partial_j\delta_r(\boldsymbol{x}-X_r)\,d\boldsymbol{x}}_{J_2}. \\ &= \underbrace{\delta_{ij}\int_\Omega \delta_r(\boldsymbol{x}-X_l)\Delta\delta_r(\boldsymbol{x}-X_r)\,d\boldsymbol{x}}_{J_{11}} - \underbrace{\int_\Omega \delta_r(\boldsymbol{x}-X_l)\partial_i\Delta^{-1}\partial_j\Delta\delta_r(\boldsymbol{x}-X_r)\,d\boldsymbol{x}}_{J_{12}} \\ &\quad \underbrace{-\delta_{ij}\int_\Omega \delta_r(\boldsymbol{x}-X_l)\partial_j^2\delta_r(\boldsymbol{x}-X_r)\,d\boldsymbol{x}}_{J_{21}} + \underbrace{\int_\Omega \delta_r(\boldsymbol{x}-X_l)\partial_i\Delta^{-1}\partial_j\Delta\partial_j\Delta^{-1}\partial_j\delta_r(\boldsymbol{x}-X_r)\,d\boldsymbol{x}}_{J_{22}}, \end{aligned} \tag{60}$$

where J_{11} and J_{21} depend on $X_l - X_r$ due to our choice of δ_r. For J_{12} and J_{22} we make the same assumptions as for I_2 and consider the limiting case. We obtain similar results

$$\lim_{r\to 0} J_{12} = -\partial_{x_i}\partial_{y_j}^3 G(\boldsymbol{x},\boldsymbol{y})\Big|_{\boldsymbol{x}=X_l,\boldsymbol{y}=X_r}, \tag{61}$$

$$\lim_{r\to 0} J_{22} = -\partial_{x_i}\Delta_y\partial_{y_j} G(\boldsymbol{x},\boldsymbol{y})\Big|_{\boldsymbol{x}=X_l,\boldsymbol{y}=X_r}. \tag{62}$$

So both terms depend only on $X_l - X_r$. However, the integrals I_0 and J_0 may not be further reduced to functions of $|X_l - X_r|$, as suggested by our numerical experiments, see Figure 3.

3.3 | Summary

We have shown our ROM preserves important properties of the FOM. Although the subspaces vary in time, interpolation technique can be applied to efficiently update coefficient matrices in the ROM.

4 | NUMERICAL RESULTS

In this section, we present three numerical examples to demonstrate the accuracy and speedup offered by our ROM. The finite difference method is used for both FOM and ROM. For the temporal discretization, we use the forward Euler method.

4.1 | Oscillation of an elliptical membrane

We consider the oscillations of a pressurized fiber. Initially, the stretched elastic fiber resides in the center of a resting fluid. The semi-major and semi-minor axes of the fiber are 0.4 and 0.2 μm, respectively. The fluid domain is $4\mu m \times 4\mu m$ with periodic boundary conditions on all edges. Fluid density and viscosity are chosen so that the Reynolds number is 0.01. The body force in this example is generated by an elastic energy functional,[13]

$$E = \int_\Gamma \varepsilon\left(\left|\frac{\partial \boldsymbol{X}}{\partial s}\right|\right) ds, \tag{63}$$

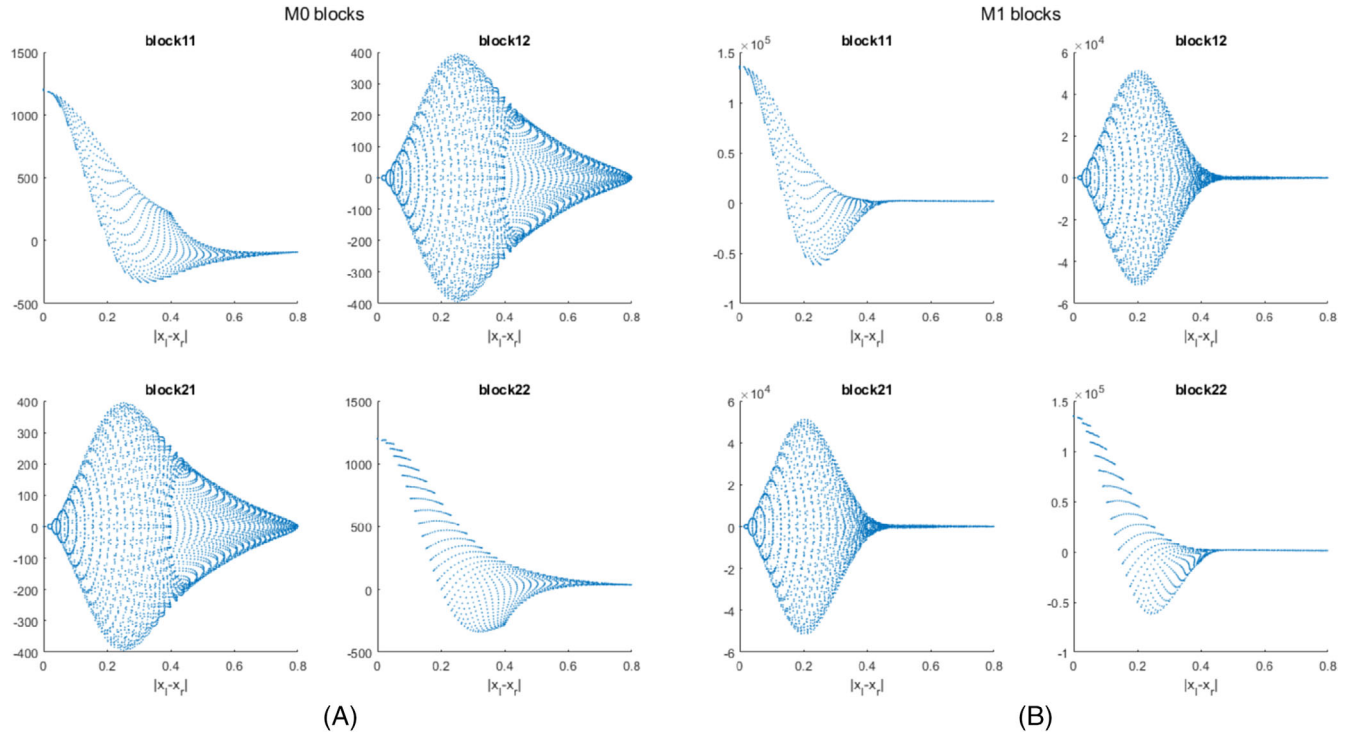

FIGURE 3 Scatter plots of entries of (A) M_0 and (B) M_1 against $|X_l - X_r|$ in 2D case. Multiple values exist, indicating these $\mathbb{R}^4$ functions may not be considered as functions in $\mathbb{R}$ of $|X_l - X_r|$

where ε is the local energy given by

$$\varepsilon(x) = \frac{\sigma}{2}(x - L)^2, \tag{64}$$

which corresponds to an elastic fiber having a "spring constant" σ and an equilibrium state where the elastic strain $|\partial \boldsymbol{X}/\partial s| \equiv L$. The force in (4) is then expressed as

$$F = -\nabla_X E = \sigma \frac{\partial}{\partial s}\left(\frac{\partial \boldsymbol{X}}{\partial s}\left(1 - \frac{L}{\left|\frac{\partial \boldsymbol{X}}{\partial s}\right|}\right)\right). \tag{65}$$

Since the fluid in the interior of the membrane is confined, the membrane will oscillate and eventually settle into a circular state. Membrane configurations simulated by the FOM and ROM are compared at different times (Figure 4A,B). The ROM simulation captures almost the same equilibrium state as the FOM. In addition, the membrane configurations are approximated accurately during the oscillation. We demonstrate that the ROM preserves the incompressibility by comparing the evolution of mass flux with that of the FOM. The mass flux is calculated by integrating the velocity over the membrane surface using the trapezoidal rule. The mass flux of the ROM is in close agreement with the FOM. Both are very close to zero up to a numerical error which keeps decreasing as the grid becomes finer, as shown in Figure 4C,D.

The one-step computation time of our ROM simulations with various grid sizes is compared to the one-step FOM simulation time in Table 1. There is a clear increase in the speedup factor as the grid spacing decreases. With 2D flow, the time complexities are $O(h^{-2})$ and $O(h^{-1})$ for the FOM and ROM simulations, respectively. The effect of the additional sampling cost at the beginning of the simulation is reported in Table 2. This overhead is less than 20 time steps of the FOM simulation. In this example, the total number of time steps is 1000. Therefore, the computational cost associated with the sampling process is negligible compared to the speedup during the simulation.

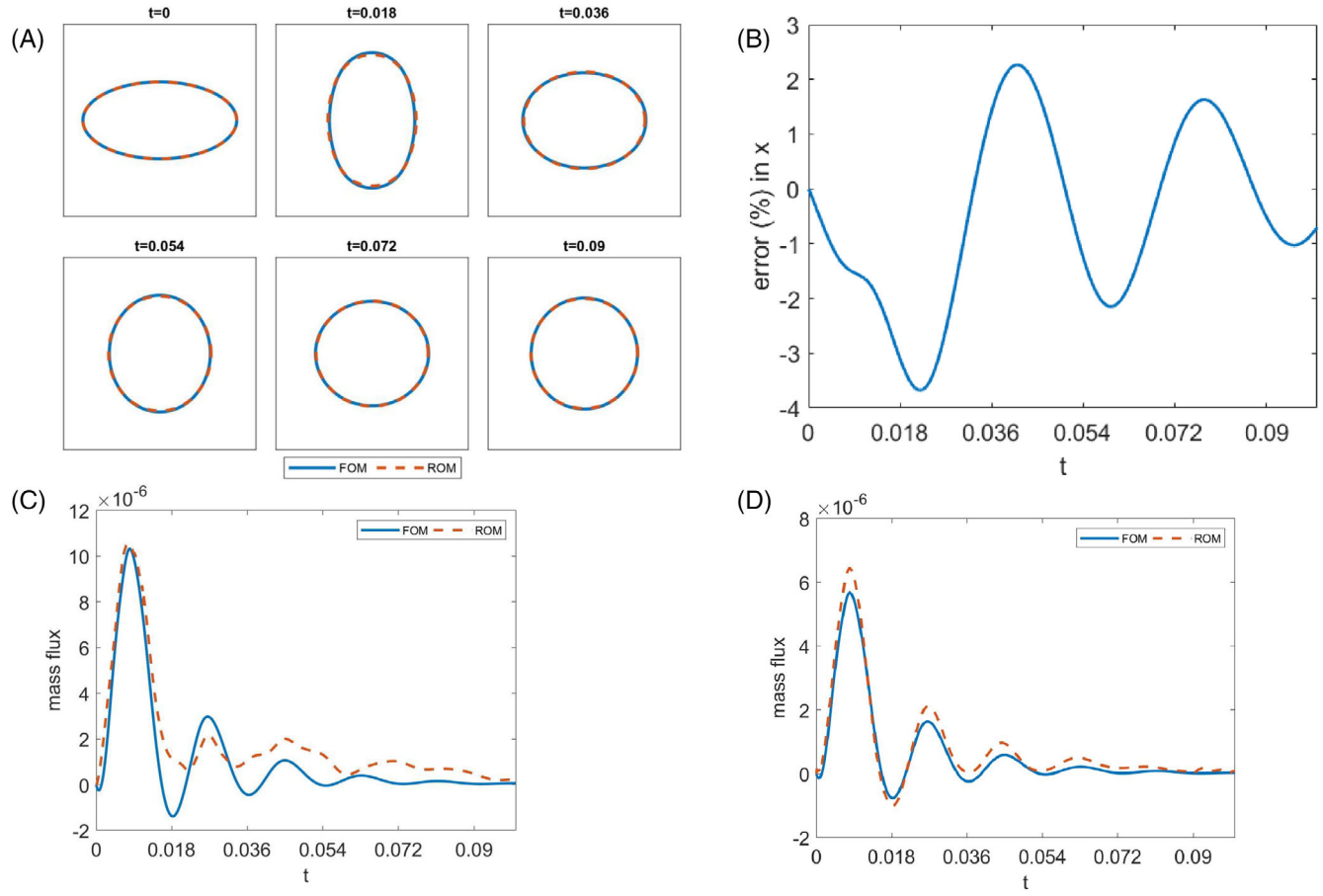

FIGURE 4 (A) Comparison between FOM and ROM of the elliptical membrane profiles at different times. (B) Relative difference between the x-coordinates of a reference structure point simulated by the FOM and ROM. (C) and (D) Evolution of mass flux across the membrane with fluid grid size $h = 1/8$ (C) and $h = 1/16$ (D)

TABLE 1 Membrane oscillation: Speedup of full order model and reduced-order model

	Model order		CPU time		
h	Full	Reduced	Full	Reduced	Speedup factor
1/6	1728	144	0.0118	0.0036	3.2778
1/8	3072	192	0.0309	0.0044	7.0227
1/12	6912	288	0.1391	0.007	19.871
1/16	12,288	384	0.3940	0.0166	24.735
1/20	19,200	480	0.9745	0.0275	35.436

TABLE 2 Membrane oscillation: Sampling cost and overall expected time saving (in seconds) for different numbers of total time steps

		$N_T = 15$			$N_T = 30$			$N_T = 50$		
h	Sampling time	FOM	ROM	Saving	FOM	ROM	Saving	FOM	ROM	Saving
1/8	0.053	0.464	0.119	0.345	0.927	0.185	0.742	10.55	0.273	1.28
1/12	0.170	2.09	0.275	1.81	4.17	0.380	3.79	6.95	0.520	6.43
1/16	0.748	5.91	0.997	4.94	11.8	1.246	10.5	19.7	1.58	18.1

We show the perimeter of the immersed structure at final time for various choices of the grid size (Figure 5). As the grid spacing reduces, the perimeter approaches an asymptotic zero-grid spacing value. We determine the order of convergence of the ROM based on these results,

$$\ln\left(\frac{1.7759-1.7863}{1.7718-1.7759}\right)/\ln(2)=1.3429.$$

4.2 | Rotation of an elliptical particle in shear flow

We study the problem of the motion of a rigid elliptical particle freely suspended in a shear flow. The fluid domain is $8\mu m \times 8\mu m$. The semi-minor and semi-major axes of the ellipse are $S_1 = 0.2\mu m$ and $S_2 = 0.3\mu m$, respectively. Initially, the ellipse is immersed in the center of a shear flow with its semi-major axis positioned along the y-axis. The maximum fluid velocity of the shear flow, fluid density, viscosity are chosen so that the Reynolds number is 0.01 (Figure 6).

It has been shown that the instantaneous inclination angle θ of the ellipse major axis with respect to the y-axis is

$$\tan(\theta)=\frac{S_2}{S_1}\tan\left(\frac{S_1S_2}{S_1^2+S_2^2}\dot{\gamma}t\right), \tag{66}$$

where t is the time variable.[39]

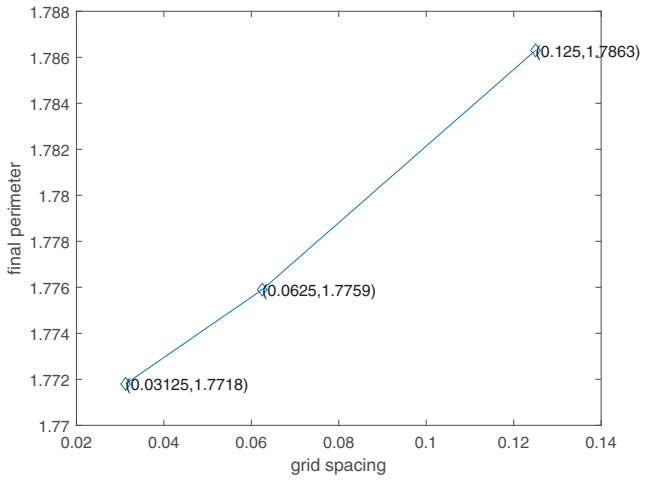

FIGURE 5 Convergence plot of ROM measured by final perimeter with varying grid spacings

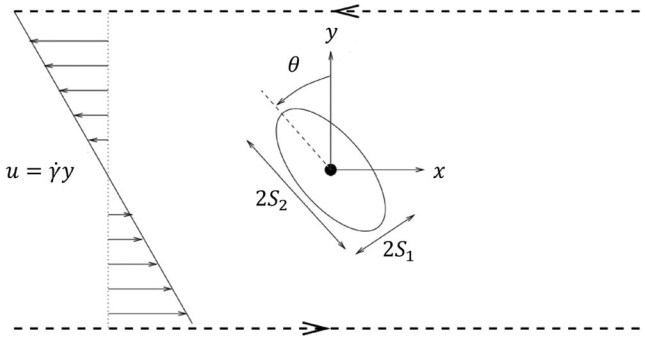

FIGURE 6 A rigid ellipse immersed in a shear flow

To preserve the elliptic shape of the rigid structure, the body force in this example is generated by a discrete bending energy.[33] Let θ_i^0 be the initial angle between the adjacent edges with the i-th Lagrangian grid point and θ_i be the current angle. The bending energy is given by

$$E_b = \sigma_b \sum_{i=1}^{n_s} \left(1 - \cos\left(\theta_i - \theta_i^0\right)\right), \tag{67}$$

where n_s is the number of Lagrangian grid points and σ_b is the bending coefficient. In this example, we choose $\sigma_b = 2000$ to increase the stiffness. The bending force generated on each structure point is given by,

$$\mathbf{F}_i = \left(F_{ix}, F_{iy}\right) = \left(-\frac{\partial E_b}{\partial x_i}, -\frac{\partial E_b}{\partial y_i}\right). \tag{68}$$

Figure 7 shows the simulated ellipse rotation rate and the analytical result (66). The rotation rate obtained by our ROM simulation is in close agreement with both the FOM simulation and the analytical solution. Table 3 shows the increase in the speedup factor as the grids become finer. Higher speedup factors are achieved for finer space grid.

4.3 | Motion of two particles in laminar flow

In the last numerical test, we simulate the motion of two membranes in a 6μm × 15μm channel. The fluid is initially at rest, with inlet velocity profile given by, as depicted in Figure 8,

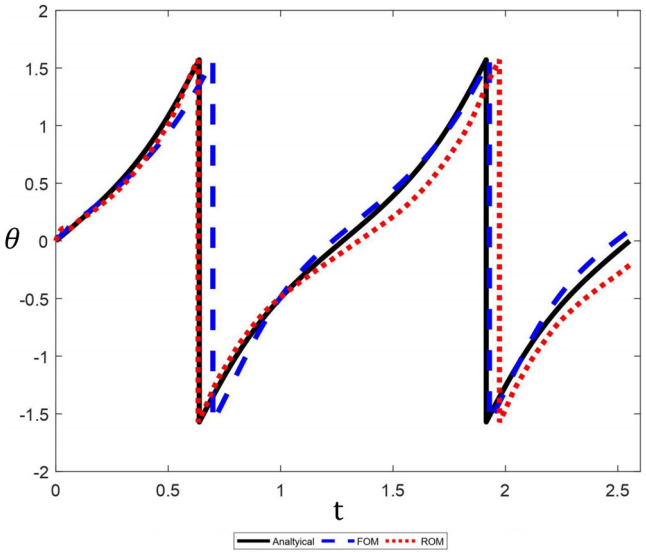

FIGURE 7 Ellipse rotation angles simulated by the full model and the reduced-order model compared with Jeffery's orbit. The variation in the angle θ relative to the ellipse major axis is plotted as a function of time t

TABLE 3 Particle rotation: Speedup of full order model and reduced-order model

	Model order		CPU time		
h	Full	Reduced	Full	Reduced	Speedup factor
3/16	2048	32	0.0214	0.0017	12.5882
1/8	4608	48	0.1012	0.0029	34.8966
3/32	8192	64	0.3065	0.0057	53.7719

$$U = U_0\left[1 - \left(\frac{y}{D}\right)^2\right], \quad -D \le y \le D. \tag{69}$$

At the beginning, the two membranes of the same elliptic shape are placed with horizontal semi-major axes and the same distance 0.6μm from its center to the x-axis. The initial semi-major axis and semi-minor axis are 0.3 and 0.2μm, respectively. Fluid density, viscosity, U_0 are chosen so that the Reynolds number is 0.01. Nonslip conditions are applied to the top and bottom boundaries.

The same bending force as in the previous example is applied to both membranes to prevent significant deformation. In addition, the two membranes interact with each other through a binding force and a repulsive force given respectively by,

$$F_{\text{binding}} = s(d - \lambda), \tag{70}$$

$$F_{\text{repulsion}} = ad + bd^3, \tag{71}$$

where d is the distance between two Lagrangian nodes on different cells and a, b, s, λ are parameters. These forces are developed to model the biochemical interactions between flowing melanoma tumor cells and substrate adherent polymorphonuclear neutrophils.[40] The attraction and repulsion forces yield oscillatory trajectories for both membranes, shown in Figure 9. Table 4 shows the increase in the speedup factor as the space grid becomes finer.

4.4 | Transport of circular capsule in a plain-Poiseuille flow

In this test case, the dynamics of a capsule within a plane-Poiseuille flow is considered. The setup of this example follows the test conducted by Coclite et al.[41] Initially, the capsule has a diameter of 7 μm and is immersed in a 2D channel

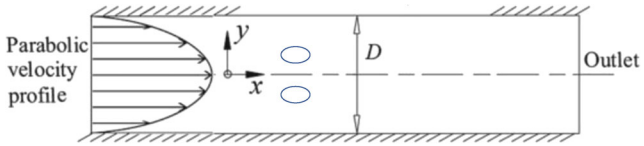

FIGURE 8 Two membranes interacting in a laminar channel flow

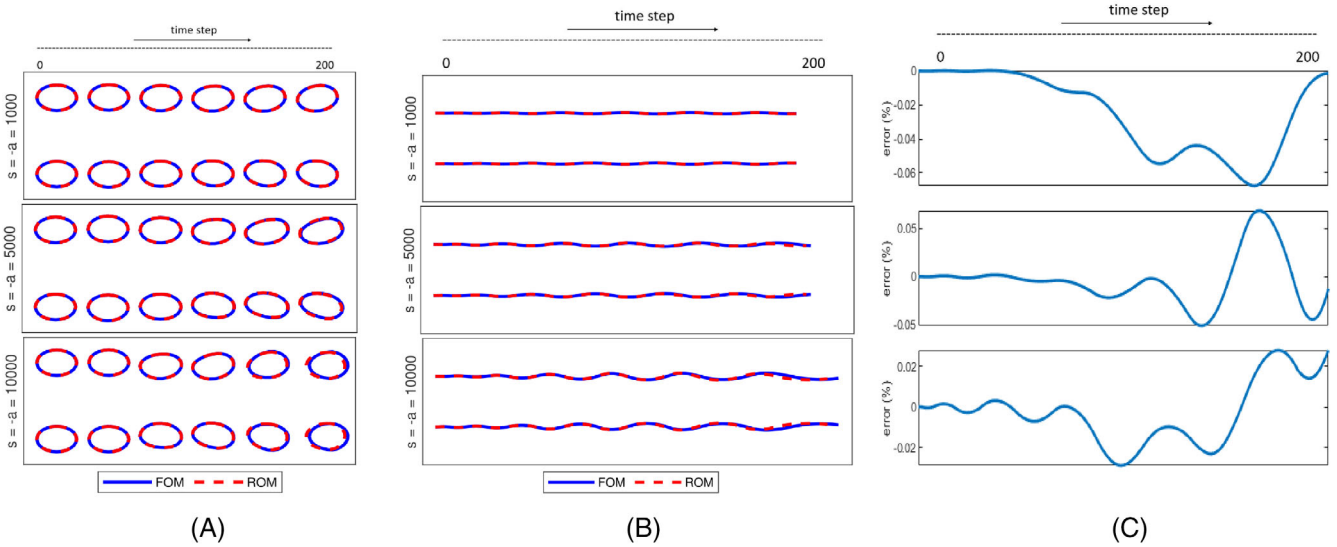

FIGURE 9 The increase of the attraction and repulsion forces (from top to bottom) as the magnitude of a and s increases. The other two parameter, b and λ, are fixed for ease of comparison. (A) Snapshots of two membranes at different times. (B) Trajectories of membrane centers. (C) Relative error in x-coordinates of a reference structure point of the upper membrane, between the FOM and ROM

TABLE 4 Two cells interaction: Speedup of full order model and reduced-order model

	Model order		CPU time		
h	**Full**	**Reduced**	**Full**	**Reduced**	**Speedup factor**
3/16	4096	64	0.0834	0.0080	10.425
1/8	9216	96	0.4272	0.0148	28.8649
3/32	16,384	128	1.2492	0.0378	33.0476

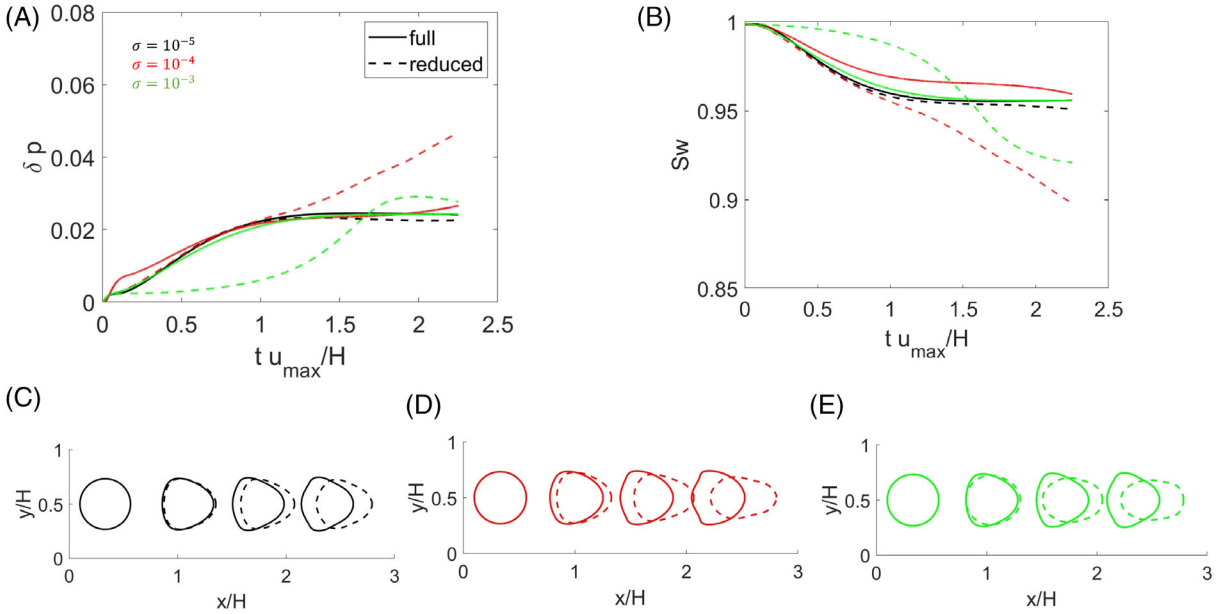

FIGURE 10 Transport of a circular capsule in plane-Poiseuille flow. (A) Variation of the capsule relative perimeter over time as function of the spring constant σ. (B) Variation of the capsule swelling ratio over time as function of the spring constant σ. (C–E) Capsule snapshots for $\sigma = 10^{-5}$ (C), $\sigma = 10^{-4}$ (D), and $\sigma = 10^{-3}$ (E) (Solid curves for FOM and dashed curves for ROM)

with a height $H = 15$ μm and length equal $3H$, centered at 7.5 μm away from the bottom of the lower wall. The fluid is initially at rest, the plane-Poiseuille flow with $u_{\max} = 10$ μm/s is then established by posing a linear pressure drop. Simulation is run at $Re = 0.01$, with $\rho = 100$ kgm^{-3}, $\mu = 10^{-5}$Pas. The body force on the capsule is the same as in Section 4.1, with three spring constant $\sigma = 10^{-5}$, 10^{-4}, and 10^{-3}Nμm^{-1}.

Following Coclite et al.,[41] we compare the results between the FOM and the ROM in terms of the capsule perimeter variation with respect to its original configuration, $\delta p(t) = \frac{p(t)-p_0}{p_0}$ (Figure 10A), and of the swelling ratio, $Sw = \frac{A(t)}{p^2(t)/4\pi}$, where $A(t)$ is the area associated with a circle of perimeter $p(t)$ (Figure 10B). The snapshots of FOM and ROM are also compared (Figure 10C–E).

For $\sigma = 10^{-5}$, the ROM is a fair approximation of the FOM. As the force coefficient increases, the system becomes more stiff. Consequently, the ROM simulation does not approximate the FOM well. We emphasize that our result is not in full agreement with the data published in Coclite et al.[41] for two reasons. First, the time-dependent Stokes equations are considered in this work instead of Navier–Stokes equations. Secondly, the force we applied to the cell model is different.

5 | CONCLUSION

In this paper, we develop a reduced-order modeling framework for FSI problems. Using the IBM as an example, we discussed the transfer function and its approximations. This proposed ROM formulation enforces the impressibility condition and also preserves the Lyapunov stability. An efficient interpolation technique is applied to efficiently update the

time-dependent coefficient matrices. The proposed model reduction technique is applied to several biological applications involving linear incompressible Stokes flows, as demonstrated by the examples. Compared to other traditional methods, this new method has the following two advantages: (1) the fluid variables are the most time-consuming part in the traditional methods, such as IBM, IIM, and FDM. But they are not explicitly involved in our ROM; (2) the structure equation is derived explicitly. It does not require special discretization techniques, for example, those for singular integrals used in the BEM. Recently, there have been growing interest in combining the reduced-order technique and data-driven methods. In this scenario, rather than the direct access to the FOM, one works with observations, for example, structure conformations, in the form of time series. The problem is then reduced to inferring parameters in the ROM. This work is underway.

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation Grants DMS-1953120 (X.L.) and DMS-2052685 (W.H.).

CONFLICT OF INTEREST

We have no conflict of interest to report.

DATA AVAILABILITY STATEMENT

Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

ORCID

Wenrui Hao https://orcid.org/0000-0002-6925-7424

REFERENCES

1. Kleinstreuer C. *Biofluid Dynamics: Principles and Selected Applications*. CRC Press; 2006.
2. Canic S. Blood flow through compliant vessels after endovascular repair: wall deformations induced by the discontinuous wall properties. *Comput Visual Sci*. 2002;4(3):147-155. doi:10.1007/s007910100066
3. Canic S, Mirkovic D. A hyperbolic system of conservation laws in modeling endovascular treatment of abdominal aortic aneurysm. In: Freistühler H, Warnecke G, eds. *Hyperbolic Problems: Theory, Numerics, Applications*. Vol 140. Birkhäuser Basel; 2001:227-236.
4. Lei H, Karniadakis GE. Predicting the morphology of sickle red blood cells using coarse-grained models of intracellular aligned hemoglobin polymers. *Soft Matter*. 2012;8(16):4507-4516. doi:10.1039/C2SM07294G
5. Li X, Peng Z, Lei H, Dao M, Karniadakis GE. Probing red blood cell mechanics, rheology and dynamics with a two-component multi-scale model. *Philos Trans Roy Soc*. 2014;372(2021):20130389. doi:10.1098/rsta.2013.0389
6. Wang Q. A hydrodynamic theory for solutions of nonhomogeneous nematic liquid crystalline polymers of different configurations. *J Chem Phys*. 2002;116(20):9120-9136. doi:10.1063/1.1452722
7. Yang X, Forest G, Mullins W, Wang Q. Dynamic defect morphology and hydrodynamics of sheared nematic polymers in two space dimensions. *J Rheol*. 2009;53(3):589-615. doi:10.1122/1.3089622
8. Yu Y. Fluid-structure interaction modeling in 3D cerebral arteries and aneurysms. In: Wriggers P, Lenarz T, eds. *Biomedical Technology: Modeling, Experiments and Simulation*. Springer; 2018:123-146.
9. Zhao J, Yang X, Shen J, Wang Q. A decoupled energy stable scheme for a hydrodynamic phase-field model of mixtures of nematic liquid crystals and viscous fluids. *J Comput Phys*. 2016;305:539-556. doi:10.1016/j.jcp.2015.09.044
10. Hall WS. *The Boundary Element Method*. 1st ed. Springer; 1994:27.
11. Li S, Li X. A boundary integral method for computing the dynamics of an epitaxial Island. *SIAM J Sci Comput*. 2011;33(6):3282-3302. doi:10.1137/100814871
12. Everstine GC, Henderson FM. Coupled finite element/boundary element approach for fluid–structure interaction. *J Acoust Soc Am*. 1990;87(5):1938-1947. doi:10.1121/1.399320
13. Peskin CS. The immersed boundary method. *Acta Numer*. 2002;11:479-517. doi:10.1017/S0962492902000077
14. Lai MC, Peskin CS. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *J Comput Phys*. 2000;160(2):705-719. doi:10.1006/jcph.2000.6483
15. Atzberger PJ, Kramer PR, Peskin CS. A stochastic immersed boundary method for fluid-structure dynamics at microscopic length scales. *J Comput Phys*. 2007;224(2):1255-1292. doi:10.1016/j.jcp.2006.11.015
16. Sotiropoulos F, Yang X. Immersed boundary methods for simulating fluid–structure interaction. *Prog Aerosp Sci*. 2014;65:1-21. doi:10.1016/j.paerosci.2013.09.003
17. Leveque RJ, Li Z. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM J Num Anal*. 1994;31(4):1019-1044.
18. Glowinski R, Pan TW, Periaux J. A fictitious domain method for Dirichlet problem and applications. *Comput Methods Appl Mech Eng*. 1994;111(3–4):283-303. doi:10.1016/0045-7825(94)90135-X

19. Hao W, Xu Z, Liu C, Lin G. A fictitious domain method with a hybrid cell model for simulating motion of cells in fluid flow. *J Comput Phys*. 2015;280:345-362. doi:10.1016/j.jcp.2014.09.020
20. Tryggvason G, Bunner B, Esmaeeli A, et al. A front-tracking method for the computations of multiphase flow. *J Comput Phys*. 2001; 169(2):708-759. doi:10.1006/jcph.2001.6726
21. Glimm J, Li X, Liu Y, Xu Z, Zhao N. Conservative front tracking with improved accuracy. *SIAM J Num Anal*. 2003;41(5):1926-1947.
22. Cottet GH, Maitre E. A level set method for fluid-structure interactions with immersed surfaces. *Math Model Method Appl Sci*. 2006; 16(03):415-438. doi:10.1142/S0218202506001212
23. Du Q, Liu C, Wang X. A phase field approach in the numerical study of the elastic bending energy for vesicle membranes. *J Comput Phys*. 2004;198(2):450-468. doi:10.1016/j.jcp.2004.01.029
24. Kublik C, Tanushev NM, Tsai R. An implicit interface boundary integral method for Poisson's equation on arbitrary domains. *J Comput Phys*. 2013;247:279-311. doi:10.1016/j.jcp.2013.03.049
25. Babuska I, Oden JT. Verification and validation in computational engineering and science: basic concepts. *Comput Methods Appl Mech Eng*. 2004;193:4057-4066. doi:10.1016/j.cma.2004.03.002
26. Bai Z. Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. *Appl Num Math*. 2002;43(1–2):9-44. doi: 10.1016/S0168-9274(02)00116-2
27. Benner P, Gugercin S, Willcox K. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Rev*. 2015;57(4):483-531. doi:10.1137/130932715
28. Gugercin S, Antoulas A, Beattie C. H2 model reduction for large-scale linear dynamical systems. *SIAM J Matrix Anal Appl*. 2008;30(2): 609-638. doi:10.1137/060666123
29. Anic B. An Interpolation-Based Approach to the Weighted H2 Model Reduction Problem. PhD thesis. Virginia Polytechnic Institute and State University, Virginia; 2008.
30. Bao Y, Donev A, Griffith BE, McQueen DM, Peskin CS. An immersed boundary method with divergence-free velocity interpolation and force spreading. *J Comput Phys*. 2017;347:183-206. doi:10.1016/j.jcp.2017.06.041
31. Nitti A, Kiendl J, Reali A, Tullio DMD. An immersed-boundary/isogeometric method for fluid–structure interaction involving thin shells. *Comput Methods Appl Mech Eng*. 2020;364:112977. doi:10.1016/j.cma.2020.112977
32. Gilmanov A, Le TB, Sotiropoulos F. A numerical approach for simulating fluid structure interaction of flexible thin shells undergoing arbitrarily large deformations in complex domains. *J Comput Phys*. 2015;300:814-843. doi:10.1016/j.jcp.2015.08.008
33. Pivkin IV, Karniadakis GE. Accurate coarse-grained modeling of red blood cells. *Phys Rev Lett*. 2008;101(11):118105. doi: 10.1103/PhysRevLett.101.118105
34. Boffi D, Gastaldi L. A finite element approach for the immersed boundary method. *Comput Struct*. 2003;81(8–11):491-501. doi: 10.1016/S0045-7949(02)00404-2
35. Kim J, Kim D, Choi H. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *J Comput Phys*. 2001;171(1):132-150. doi:10.1006/jcph.2001.6778
36. Yang JZ, Wu X, Li X. A generalized Irving–Kirkwood formula for the calculation of stress in molecular dynamics models. *J Chem Phys*. 2012;137(13):134104. doi:10.1063/1.4755946
37. Freund RW. Krylov-subspace methods for reduced-order modeling in circuit simulation. *J Comput Appl Math*. 2000;123(1–2):395-421. doi:10.1016/S0377-0427(00)00396-4
38. Ma L, Li X, Liu C. Coarse-graining Langevin dynamics using reduced-order techniques. *J Comput Phys*. 2019;380:170-190.
39. Jeffery GB. The motion of ellipsoidal particles immersed in a viscous fluid. *Proc Roy Soc Lond Ser A*. 1922;102(715):161-179. doi: 10.1098/rspa.1922.0078
40. Behr J, Gaskin B, Fu C, Dong C, Kunz R. Localized modeling of biochemical and flow interactions during cancer cell adhesion. *PLoS One*. 2015;10(9):e0136926. doi:10.1371/journal.pone.0136926
41. Coclite A, Ranaldo S, de Tullio M, Decuzzi P, Pascazio G. Kinematic and dynamic forcing strategies for predicting the transport of inertial capsules via a combined lattice Boltzmann—immersed boundary method. *Comput Fluids*. 2019;180:41-53. doi:10.1016/j.compfluid.2018.12.014

How to cite this article: Luo Y, Li X, Hao W. Projection-based model reduction for the immersed boundary method. *Int J Numer Meth Biomed Engng*. 2021;e3558. doi:10.1002/cnm.3558