Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

Computing Viscous Flow Along a 2D Open Channel Using the Immersed Interface Method

Sarah E. Patterson*[1] | Anita T. Layton[2]

[1]Applied Mathematics Department, Virginia Military Institute, Virginia, United States
[2]Departments of Applied Mathematics and Biology, Cheriton School of Computer Science, and School of Pharmacy, University of Waterloo, Ontario, Canada

Correspondence
*Sarah Patterson, Applied Mathematics Department, Virginia Military Institute, Virginia, United States. Email: pattersonse@vmi.edu

Summary

We present a numerical method for simulating 2D flow through a channel with deformable walls. The fluid is assumed to be incompressible and viscous. We consider the highly viscous regime, where fluid dynamics are described by the Stokes equations, and the less viscous regime described by the Navier-Stokes equations. The model is formulated as an immersed boundary problem, with the channel defined by compliant walls that are immersed in a larger computational fluid domain. The channel traverses through the computational domain, and the walls do not form a closed region. When the walls deviate from their equilibrium position, they exert singular forces on the underlying fluid. We compute the numerical solution to the model equations using the immersed interface method, which preserves sharp jumps in the solution and its derivatives. The immersed interface method typically requires a closed immersed interface, a condition that is not met by the present configuration. Thus, a contribution of the present work is the extension of the immersed interface method to immersed boundary problems with open interfaces. Numerical results indicate that this new method converges with second-order accuracy in both space and time, and can sharply capture discontinuities in the fluid solution.

KEYWORDS:
Fluid-structure interaction, immersed boundary problem, Navier-Stokes, fluid dynamics, finite difference, open interface

1 | INTRODUCTION

A detailed and accurate description of fluid flow through a channel with compliant or actively moving walls is of interest in many biological applications, including pumping via peristalsis in a valveless heart[1], food mixing in the intestine[2], and blood flow through a vessel[3,4] or renal tubules[5,6,7,8]. A natural way to model flow along a compliant channel is to frame it as an immersed boundary problem[9,10]. Immersed boundary problems are typically a subset of fluid-structure interaction problems in which a thin structure or physical boundary is present in the fluid[11]. The immersed boundary method transfers the singular boundary forces onto the underlying fluid using approximate (smooth) Dirac delta functions typically with $\mathcal{O}(h)$ support. Because the delta function is smoothed, this approach does not capture the jump discontinuity in the solution (e.g., pressure) at the immersed boundary, but rather approximates the solution as a continuous function with a large gradient. In general, the immersed boundary method computes approximations with first-order spatial accuracy. Most immersed boundary problems are formulated for Navier–Stokes flows, although it is applicable to Stokes flows as well.

If better accuracy is desired, especially near the immersed boundary, one may use the immersed interface method developed by LeVeque and Li[12,13]. The immersed interface method captures the jumps in the solution and its derivatives sharply, and generates approximations with second-order accuracy. The key idea in the immersed interface method is the incorporation of known jumps in the solution or its derivatives into the finite difference schemes. The immersed interface method has been applied to a variety of applications to compute the coupled motion of a viscous fluid and a thin closed interface with second-order accuracy[14,15,16,17,18]. However, the immersed interface method requires that the immersed interfaces be closed. Thus, to simulate flow through a channel, one typically approximates the channel as a closed, elongated interface with capped ends[9,17,19,20,21,22], immersed in a larger computational domain. The flow is then driven by a pair of fluid source and sink located at the opposing ends of the channel. A major downside of this setup is the unrealistic flow near the source and sink. As a remedy, a longer channel is modeled, and only the flow sufficiently far from the source and sink is considered.

The goal of this study is to develop a modified version of the immersed interface method that can be applied directly to an open channel without the representation of (unrealistic) fluid source and sink. To achieve that goal, we derive the jump conditions for the open interface and apply the method to the Stokes equations as well as the Navier-Stokes equations. Numerical examples indicate that the method captures the sharp jumps in the solutions, and achieves second-order accuracy in both time and space.

2 | PROBLEM FORMULATION

2.1 | Computational domain and immersed interface

We formulate a model that simulates fluid flow through an open channel with compliant walls, extending from one side of a rectangular computational domain to the opposite side. The model is formulated in 2D rectangular coordinates. To define the model, consider a rectangular fluid and computational domain given by $\Psi = [0, L] \times [-H, H]$. Let Γ represent the compliant walls of the channel immersed in Ψ. Specifically, Γ is an interface immersed in the fluid formed by two distinct smooth curves $\mathbf{X} = G_1(s)$ for $s \in [0, L_1]$ such that L_1 is the resting length of the curve G_1 which intersects $\partial\Psi$ at $(0, a_1)$ and (L, b_1) and $\mathbf{X} = G_2(s)$ for $s \in [0, L_2]$ such that L_2 is the resting length of G_2 which intersects $\partial\Psi$ at $(0, a_2)$ and (L, b_2), as shown in Figure 1. We are interested in tracking flow through the channel (i.e., between G_1 and G_2), driven by a pressure gradient between $x = 0$ and $x = L$. Let $\mathbf{n}$ be the unit vector normal to Γ oriented towards $\partial\Psi$ away from the channel.

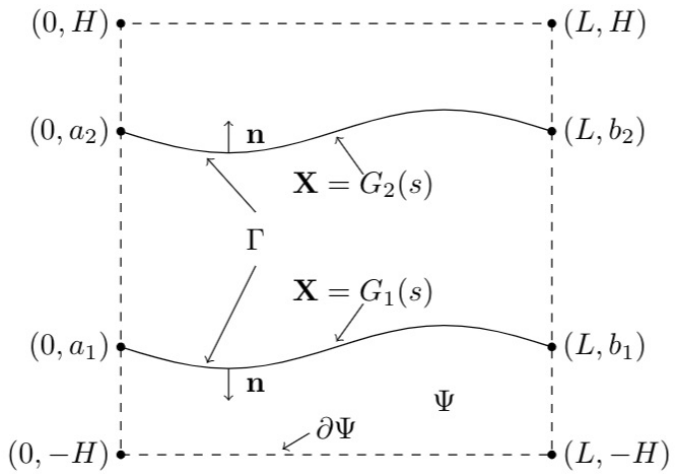

FIGURE 1 The open immersed interface, Γ, spans the 2D computational domain, $\Psi = [0, L] \times [-H, H]$. The interface Γ consists of two separate curves given by G_1 and G_2 that intersect the $x = 0$ and $x = L$ on $\partial\Psi$ at $(x, y) = (0, a_1)$, $(x, y) = (0, b_1)$, and $(x, y) = (0, a_2)$, $(x, y) = (0, b_2)$ respectively. The unit normal, defined to be $\mathbf{n}$, is oriented towards $\partial\Psi$.

2.2 | Boundary conditions

To properly state the problem, appropriate boundary conditions must be imposed on fluid velocity and pressure. Bi-periodic boundary conditions are assumed for fluid velocity, which implies that the volume of the fluid in the channel remains constant in time. To drive flow, we prescribe a pressure gradient inside the channel. For simplicity, we assume that $a_1 = b_1$ and $a_2 = b_2$. Thus, the width of the inlet and outlet is the same. We require there to be a constant difference in pressure, denoted P_{diff}, at the inlet and outlet of the channel

$$p(0, y) = p(L, y) + \begin{cases} P_{\text{diff}} & a_1 \leq y \leq a_2 \\ 0 & \text{else} \end{cases} \tag{1}$$

The derivatives of pressure on the $x = 0$ and $x = L$ boundaries are required to be equal

$$\left.\frac{\partial p}{\partial x}\right|_{(x=0,y)} = \left.\frac{\partial p}{\partial x}\right|_{(x=L,y)}. \tag{2}$$

We will refer to this setup as "inhomogeneous periodic boundary conditions." Not only does this setup generate a pressure gradient along the channel, but it also forces the pressure gradient to be periodic in x, which is consistent with the periodic boundary conditions imposed for velocity. Periodic boundary conditions are imposed for pressure at the $y = \pm H$ boundaries.

2.3 | Fluid structure interactions

The Navier-Stokes equations are given by

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}\right) = \mu \Delta \mathbf{u} - \nabla p + \mathbf{F} \tag{3}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{4}$$

where $\mathbf{u} = (u, v)$, where u and v denote fluid velocity components in the x and y directions, respectively; p is the pressure; μ is viscosity; and $\mathbf{F}$ is the interfacial force, which is singularly supported along Γ (see below). We consider also the zero Reynolds number regimes given by the Stokes momentum equation:

$$\mu \Delta \mathbf{u} - \nabla p + \mathbf{F} = 0 \tag{5}$$

and the continuity equation (Eq. 4).

The force exerted by the interface Γ can be written as

$$\mathbf{F}(\mathbf{x}, t) = \int_{\Gamma} \mathbf{f}(\alpha) \delta(\mathbf{x} - \mathbf{X}(\alpha)) d\alpha, \tag{6}$$

where $\mathbf{X}$ denotes the position of the interface, $\alpha = \alpha(s, t)$ is the material coordinate(s) that parameterize the interface curve at time t, $\mathbf{f}(\alpha)$ is the force strength at the point $\mathbf{X}(\alpha)$, and δ is the Dirac delta function.

The force strength $\mathbf{f}(\alpha)$ from Eq. 6 has two major components, an elastic force $\mathbf{f}_E$ and a tether force $\mathbf{f}_T$:

$$\mathbf{f}(\alpha) = \mathbf{f}_E(\alpha) + \mathbf{f}_T(\alpha). \tag{7}$$

Since the interface is elastic, any deviation from its resting configuration generates a restorative force. The elastic force can be modeled using Hooke's law

$$\mathbf{f}_E(s, t) = \frac{\partial}{\partial s}(T(s, t)\boldsymbol{\tau}(s, t)) \tag{8}$$

where the unit tangent vector to Γ is given by

$$\boldsymbol{\tau}(s, t) = \frac{\partial \mathbf{X}/\partial s}{|\partial \mathbf{X}/\partial s|}. \tag{9}$$

The tension $T(s, t)$ in Eq. 8 is given by

$$T(s, t) = a_E\left(\left|\frac{\partial \mathbf{X}}{\partial s}\right| - 1\right). \tag{10}$$

where a_E controls the stiffness of the interface.

The second force component $\mathbf{f}_T$ arises from the displacement of the tethers from their equilibrium positions or anchor points. The interface control knots are tethered to anchor points in the fluid domain by a spring with resting length 0. Suppose the interface $\mathbf{X}$ is anchored to $\bar{\mathbf{X}}$. Let a_T be the spring force constant. Then the tether force is given by

$$\mathbf{f}_T = a_T(\bar{\mathbf{X}} - \mathbf{X}). \tag{11}$$

Thus, if the interface knots move away from their anchor points, restorative forces are generated. Interface movement can either be restricted by using stationary anchor points, or be induced by moving the position of the anchor points.

The interface is deformable and moves at the same speed as the local fluid. The no-slip condition

$$\frac{d\mathbf{X}}{dt} = u(\mathbf{X}). \tag{12}$$

describes this motion.

2.4 | Jump conditions

The derivation of the jump conditions normally requires that the interface be a closed curve. Here we extend the derivation to an open channel. Details are explained in the Appendix. Briefly, we extend Γ to a fictitious closed and piecewise smooth curve Γ_c by taking the union of Γ, the line joining the points $\{(0, a_1), (0, a_2)\}$, and the line joining the points $\{(L, b_1), (L, b_2)\}$. The rest of the derivation is similar to the closed curve case. The jump conditions given in Eqs. 13-16 are the same for the Stokes equations and for the Navier-Stokes equations. This follows due to the no-slip condition assumed for the interfacial motion, which generates a continuous velocity field and material derivative across the interface[23].

$$[p] = \mathbf{f} \cdot \mathbf{n} \tag{13}$$

$$[p_n] = \frac{\partial}{\partial s}(\mathbf{f} \cdot \boldsymbol{\tau}) \tag{14}$$

$$[\mathbf{u}] = 0 \tag{15}$$

$$\mu[\mathbf{u}_n] = -(\mathbf{f} \cdot \boldsymbol{\tau})\boldsymbol{\tau}. \tag{16}$$

Below we refer to this extended immersed interface method as simply the immersed interface method.

3 | NUMERICAL METHODS

3.1 | Stokes problem

To solve the Stokes equations with a singular force (Eqs. 4 and 5), we compute the fluid velocity and pressure on a fixed Eulerian grid:

$$\Omega_h = \left\{ \mathbf{x}_{i,j} = \left(jh, ih - \frac{H}{2} \right) \middle| \, i \in 1, \ldots, N_y \text{ and } j \in 1, \ldots, N_x \right\},$$

where h is the grid spacing and $N_x = \frac{L}{h} + 1$ and $N_y = \frac{H}{h} + 1$ are the number of grid points in the x and y directions respectively. A moving Lagrangian frame of reference is used to track the location of the interface Γ. The interface position at time $t_n = n\Delta t$ is tracked by N_b boundary markers $\mathbf{X}^n = \{\mathbf{X}_i^n\}_{i=1}^{N_b}$ that are connected by a cubic spline[24].

The Stokes solution is computed by applying the immersed interface method to solve a series of three Poisson equations: one obtained by taking the divergence of Eq. 5 to yield

$$\Delta p = \nabla \cdot \mathbf{F}, \tag{17}$$

the other two being the two velocity components of Eq. 4. Specifically, Eqs. 4 and 17 are discretized using second-order finite difference, with jump conditions Eqs. 13–16 incorporated into the finite difference stencil as correction terms. For details see Refs. [12,14].

3.2 | Navier-Stokes problem

A large number of correction terms are involved in the application of the immersed interface method directly to the Navier-Stokes equations. As an alternative, we adopt the velocity decomposition approach[25]. This approach leverages the fact that, for a given singular interfacial force, the jumps in the solutions are identical for both the Stokes equations and the Navier-Stokes equations (see Eqs. 13–16)[12,13].

To apply the velocity decomposition approach, we split the Navier-Stokes equations into two parts: a singular part that satisfies the Stokes equations including the singular force (denoted by $\mathbf{u}_s$ and p_s), and a regular part ($\mathbf{u}_r$ and p_r):

$$\mathbf{u} = \mathbf{u}_s + \mathbf{u}_r, \quad p = p_s + p_r \tag{18}$$

Taking the difference between Eqs. 3 and 5, we arrive at that the regular part, $\mathbf{u}_r$ and p_r, satisfies

$$\rho\left(\frac{\partial \mathbf{u}_r}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}_r\right) = \mu\Delta\mathbf{u_r} - \nabla p_r + \mathbf{F_b} \tag{19}$$

where $\mathbf{F}_b$ is a body force

$$\mathbf{F}_b = -\rho\left(\frac{\partial \mathbf{u}_s}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}_s\right). \tag{20}$$

Note that the full velocity $\mathbf{u}$ is used in the transport of $\mathbf{u}_r$ and in $\mathbf{F}_b$. Then substituting Eq. 18 into Eq. 4 yields

$$\nabla\cdot\mathbf{u}_r = 0 \tag{21}$$

To advance the full solution from t_n to t_{n+1}, the Stokes part $(\mathbf{u}_s^{n+1}, p_s^{n+1})$ is first computed, following the procedures described in Sect. 3.1. This allows us to update $\mathbf{F}_b^{n+1}$ by integrating along the fluid trajectory backward in time (more below). Then the regular solution $(\mathbf{u}_r^{n+1}, p_r^{n+1})$ is computed. Equations 19 and 21 are essentially the Navier-Stokes equations with a body force and are solved using the projection method. That is, we first compute the intermediate solution $\mathbf{u}_r^{*,n+1}$ using Eq. 19 alone and then project $\mathbf{u}_r^{*,n+1}$ onto the divergence free space to yield $\mathbf{u}_r^{n+1}$ and p_r^{n+1}.

To compute $\mathbf{u}_r^{*,n+1}$, we first rewrite Eq. 19 in terms of the material derivative as follows

$$\rho\frac{D\mathbf{u}_r}{Dt} = -\nabla p_r + \mu\nabla^2\mathbf{u}_r + \mathbf{F}_b. \tag{22}$$

Note that $\mathbf{u}_r$ and its material derivatives are smooth along the fluid trajectories. This motivates us to use the semi-Lagrangian time-discretization method. The semi-Lagrangian method computes the solution at fixed Eulerian grid points $\mathbf{x}$ by integrating the solution backward in time along the trajectories of fluid particles that pass through those grid points at t_{n+1}. In our case, the semi-Lagrangian discretization is applied to the material derivative $D\mathbf{u}_r/Dt$. To proceed, we first compute the upstream positions of the particle, $\mathbf{x^n}$ at t_n and $\mathbf{x^{n-1}}$ at t_{n-1}, which can be done by integrating

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{u}(\mathbf{x}(t), t), \quad \mathbf{x}(t_{n+1}) = \mathbf{x}_0, \tag{23}$$

backwards in time. Once $\mathbf{x}^n$ and $\mathbf{x}^{n-1}$ are found, we compute $\mathbf{u}_r$ at these locations and times. It is unlikely that the upstream positions coincide with grid points. Therefore these upstream velocity values $\tilde{\mathbf{u}}_r^n = \mathbf{u}_r(\mathbf{x}^n, t_n)$ and $\tilde{\mathbf{u}}_r^{n-1} = \mathbf{u}_r(\mathbf{x}^{n-1}, t_{n-1})$ are approximated using spatial interpolation. The material derivative is then approximated using the second-order backwards difference formula,

$$\rho\left(\frac{3\mathbf{u}_r^{*,n+1} - 4\tilde{\mathbf{u}}_r^n + \tilde{\mathbf{u}}_r^{n-1}}{2\Delta t}\right) + \nabla p_r^n = \mu\Delta\mathbf{u}_r^{*,n+1} + \mathbf{F}_b^{n+1}. \tag{24}$$

The intermediate solution $\mathbf{u}_r^{*,n+1}$ is then projected, $P(\mathbf{u}_r^{*,n+1}) = \mathbf{u}_r^{n+1}$, into the subspace of divergence-free vector fields so that $\nabla\cdot\mathbf{u}_r^{n+1} = 0$. Specifically, we solve

$$\Delta\phi^{n+1} = \frac{1}{\Delta t}\nabla\cdot\mathbf{u}_r^{*,n+1}. \tag{25}$$

and update $\mathbf{u}_r^{n+1}$ and p_r^{n+1} by

$$\mathbf{u}_r^{n+1} = \mathbf{u}_r^{*,n+1} - \Delta t\nabla\phi^{n+1} \tag{26}$$

$$\nabla p_r^{n+1} = \nabla p_r^n + \frac{3}{2}\nabla\phi^{n+1} - \mu\Delta t\nabla^3\phi^{n+1}. \tag{27}$$

The full solution is given by Eq. 18. Details for the velocity decomposition method can be found in Ref.[25].

3.3 | Boundary condition and solution decomposition

In the velocity decomposition method, boundary conditions must be imposed for both the Stokes and regular parts such that the boundary conditions for the full solution are still satisfied. Recall that bi-periodic boundary conditions are prescribed for the full velocity solution. This is achieved by imposing bi-periodic boundary conditions for both $\mathbf{u}_s$ and $\mathbf{u}_r$ as well as the intermediate solution $\mathbf{u}_r^*$.

The full pressure solution must satisfy the inhomogeneous periodic boundary conditions given by Eqs. 1–2. This is achieved by imposing the inhomogeneous periodic conditions on p_s along x and periodic boundary conditions along y, and by imposing the regular bi-periodic boundary conditions on p_r. Bi-periodic boundary conditions are assumed for the auxillary variable ϕ.

4 | NUMERICAL RESULTS

4.1 | Stokes Results

We first conduct a grid refinement study on the Stokes problem to test the spatial convergence of the method in the zero Reynolds number regime. In this test, the interface is initially displaced and allowed to return to its resting position as a channel with a constant width. The computational domain is taken to be $\Omega = [0\text{cm}, 4.25\text{cm}] \times [-1.625\text{cm}, 1.625\text{cm}]$. The interface is tethered to its resting position as a channel with constant width $R = 1.42$ centered at $y = 0$. The interface is initially displaced in the y direction according to the function

$$Y(x) = .2e^{-\left(x-\frac{L}{2}\right)^2} + 0.5R. \tag{28}$$

The pressure drop along the channel (in x) is taken to be $P_{\text{diff}} = 2.544$ gm/(s^{2}cm). Model parameters for this simulation are presented in Table 1.

TABLE 1 Parameters used for the spatial convergence study for 2D Stokes flow in rectangular coordinates.

Parameter	Symbol	Value
Viscosity	μ	.1 gm/(cm·s)
Density	ρ	.1 gm/cm^3
Domain length	L	4.25 cm
Domain height	H	3.25 cm
Resting channel width	R	0.7 cm
Interface control points	N_b	100
Simulation length	T	120 ms
Time steps	N_t	60
Time step size	Δt	$2e-3$ ms
Tether force constant	a_{tether}	5 gm/s^2
Elastic force constant	a_{elastic}	1 gm/s^2
Pressure difference	P_{diff}	2.544 gm/(s^{2}cm)

A spatial grid refinement study was conducted at $t = 0$. Computed solutions were compared to an $N_x = 1089$ by $N_y = 1666$ high resolution numerical solution. Results in Table 2 indicate that second-order spatial convergence was achieved for pressure and both components of velocity. Furthermore, the jump discontinuity in p and in the normal derivative of u are captured robustly. Solution profiles are not shown for these results, because the sharp-interface feature of the method is also captured and illustrated in the Navier-Stokes example below.

TABLE 2 Spatial convergence results for the Stokes problem. The interface is initially displaced from its resting position and allowed to relax. These results shown are computed at $t = 0$.

Grid size		p		u		v	
N_x	N_y	$\|\|\cdot\|\|_\infty$	Order	$\|\|\cdot\|\|_\infty$	Order	$\|\|\cdot\|\|_\infty$	Order
35	27	1.793		5.476		2.723	
69	53	0.560	1.68	1.703	1.68	0.764	1.83
137	105	0.144	1.96	0.357	2.26	0.187	2.03
273	209	0.033	2.12	0.074	2.27	0.046	2.03
545	417	0.007	2.20	0.015	2.28	0.009	2.38

4.2 | Navier-Stokes Equations

For the second example, we solve the Navier-Stokes equations to assess the spatial and temporal accuracy in the non-zero Reynolds number regime.

Spatial convergence and steady-state profile. We start with a fluid that is at rest throughout the domain and a channel with uniform width $W = 10.1$. The axial pressure gradient is then gradually increased during the first 0.125s and then held constant according to the function:

$$P_{\text{diff}}(t) = \begin{cases} \frac{3\mu L}{R^2} \sin(\frac{\pi}{2}\frac{t}{1/8}) & t < 1/8 \\ \frac{3\mu L}{R^2} & t \geq 1/8. \end{cases} \tag{29}$$

where $R = W/2$ The channel boundaries are tethered to its initial and resting position, i.e., a channel with a width of W. Table 3 shows the remainder of the parameters.

TABLE 3 Parameters used for the spatial convergence study of the Navier-Stokes problem.

Parameter	Symbol	Value
Viscosity	μ	0.0175 gm/(cm·s)
Density	ρ	1.055 gm/cm^3
Domain length	L	32 μm
Domain height	H	16 μm
Resting channel width	W	10.1 μm
Simulation interval	T	1 s
Number of time steps	N_t	2560
Time step size	Δt	4e-4 s
Tether force constant	a_{Tether}	1e-4 gm/s^2
Elastic force constant	a_{Elastic}	1e-4 gm/s^2
Max pressure difference	P_{diff}	6.6e-5 gm/(s$^2\mu$m)

At steady state, pressure gradient and velocity are described by 2D Poiseuille flow, i.e.,

$$u = -\frac{dp}{dx}\frac{1}{2\mu}(R^2 - y^2), \quad v = -\frac{dp}{dy} = 0. \tag{30}$$

It follows that pressure decreases linearly along the channel and u has a parabolic profile given by

$$p = \begin{cases} c - \frac{3\mu\bar{u}x}{R^2} & |y| \leq R \\ 0 & R \leq |y| \end{cases}, \qquad u = \begin{cases} \frac{3\bar{u}(R^2-y^2)}{2R^2} & |y| \leq R \\ 0 & R \leq |y| \end{cases}, \qquad v = 0.$$

In Figure 2, left panel, a solid line shows the expected Poiseuille velocity profile given the parameters of the simulation. The computed u-component of the velocity profile on the slice where $x = 16\mu$m at $t = 1$s is denoted by the + symbol. The computed u exhibits an approximate parabolic profile.

The computed p is shown in Figure 2, right panel. Note, interface segments are located at approximately $y = \pm 5.05\mu$m. It can be seen that p decreases approximately linearly decreasing along the channel. Outside of the channel, the pressure is approximately zero. There is a sharp discontinuity in pressure across the interface.

To test the spatial convergence of the method, the solution was computed at various spatial resolutions. The solutions are compared with high-resolution solutions computed on a 1025×513 grid at time $t = 0.1$s when the solution is still transitioning from its initial solution to its steady-state solution. The number of interface control points for each segment of the interface is $N_b = \frac{N_x+1}{2}$. Convergence results indicate that the method is approximately second-order accurate in space; see Table 4.

Temporal convergence. The above configuration is not ideal for assessing temporal convergence in part because the interface position undergoes little deformation. Thus, we consider a different configuration in which the channel walls move. Specifically, the wall tether anchors are displaced in time, causing the walls to actively deform. The y-position of the tethers at position x at

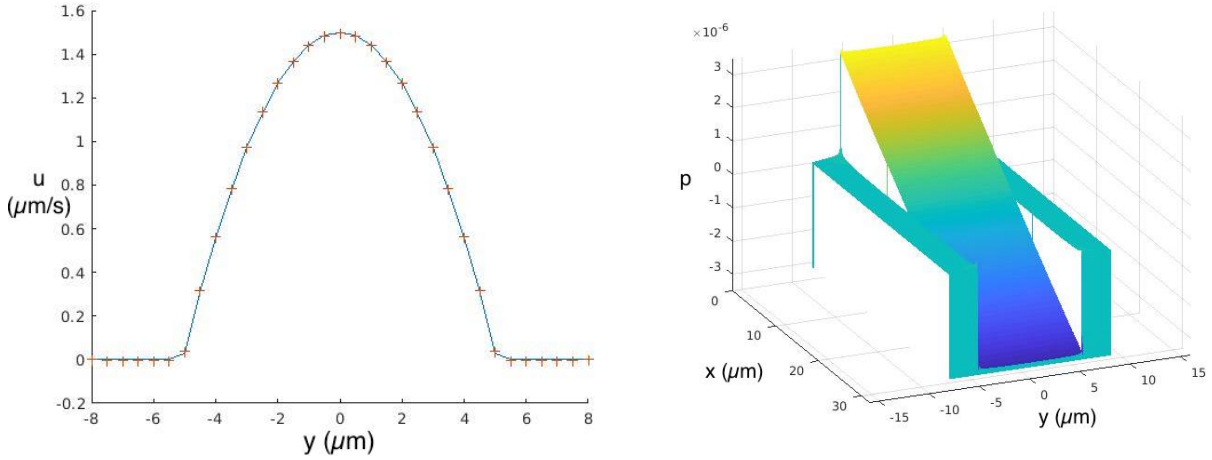

FIGURE 2 *Left:* Velocity profile (+) of computed solution at a cross-section compared with expected velocity profile of 2D Poiseuille flow. *Right:* Computed pressure in the computational domain. Pressure is linearly decreasing inside of the channel and approximately constant outside the channel with sharp discontinuities that occur across the interface located near $y = \pm 5.05\mu$m.

TABLE 4 Spatial convergence results for Navier-Stokes flow compared to a high resolution 1025×513 solution taken at time $t = .1$s.

Grid size		p		u		v	
N_x	N_y	$\|\|\cdot\|\|_\infty$	Order	$\|\|\cdot\|\|_\infty$	Order	$\|\|\cdot\|\|_\infty$	Order
		1.0e-04 ×					
65	33	0.134		2.47		1.4744	
129	65	0.021	2.70	0.447	2.46	0.235	2.64
257	129	0.009	1.16	0.193	1.21	0.096	1.30
513	257	0.002	2.28	0.032	2.58	0.014	2.75

time t for the upper and lower interface, denoted $Y_{\text{toptether}}(x,t)$ and $Y_{\text{bottom,tether}}(x,t)$, respectively, are given by

$$Y_{\text{top,tether}}(x,t) = R\left(1 - .1\sin\left(\frac{\pi t}{4T}\right)\sin\left(\frac{2\pi x}{L}\right)\right) \tag{31}$$

$$Y_{\text{bottom,tether}}(x,t) = -R\left(1 + .1\sin\left(\frac{\pi t}{4T}\right)\sin\left(\frac{2\pi x}{L}\right)\right). \tag{32}$$

The fluid solutions are computed in the domain with a length of 61μm and width 21μm. The remainder of the parameter values for this simulation are given in Table 5. The temporal accuracy of the method is assessed by refining the time step and comparing the solutions at time $t = 12.5\mu$s to a high-resolution solution computed after 64 time steps of size $\Delta t = 1.8e-7$ when the solution is changing at a sufficiently high rate. Convergence results, given in Table 6, indicate that this method can achieve second-order accuracy in time.

5 | DISCUSSION

We have presented a numerical method for simulating viscous fluid flow through an open channel with deformable walls. The model is formulated as an immersed boundary problem, with the channel spanning from one end of the computational domain to the other. We apply the method to the Stokes equations and the Navier Stokes equations. The Stokes equations are solved using a method that is an extension of the immersed interface method, which originally requires the immersed interface to be closed.

TABLE 5 Parameters used for the temporal convergence study of the Navier-Stokes problem.

Parameter	Symbol	Value
Viscosity	μ	.0175 gm/(cm·s)
Density	ρ	1.055 gm/cm^3
Domain length	L	61 μm
Domain height	H	21 μm
X-grid points	N_x	489
Y-grid points	N_y	169
Spatial grid size	h	.012 μm
Initial channel radius	R	5.05 μm
Vessel control points	N_b	100
Simulation length	T	12.5 μs
Tether force constant	a_{Tether}	1e-4 gm/s^2
Elastic force constant	$a_{Elastic}$	1e-4 gm/s^2
Pressure difference	P_{diff}	6.6e-5 gm/(s$^2\mu$m)

TABLE 6 Temporal convergence results for the Navier-Stokes problem.

Time Step		p_x		p_y		u		v	
N_t	Δt	$\|\|\cdot\|\|_\infty$	Order	$\|\|\cdot\|\|_\infty$	Order	$\|\|\cdot\|\|_\infty$	Order	$\|\|\cdot\|\|_\infty$	Order
	1.0e-06 ×	1.0e-06 ×		1.0e-06 ×					
4	3.13	0.50		0.87		2.86		142.94	
8	1.56	0.17	1.54	0.29	1.58	0.70	2.03	31.51	2.18
16	0.78	0.05	1.88	0.08	1.87	0.16	2.14	6.84	2.20
32	0.39	0.01	2.24	0.02	2.18	0.03	2.34	1.30	2.39

This method gives second-order accurate values by incorporating known jumps for the solution and its derivatives into a finite difference method. The Navier-Stokes equations are solved using the velocity decomposition approach, which decomposes the velocity into a "Stokes" part and a "regular" part. The first part is determined by the Stokes equations and the singular interfacial force. The regular part of the velocity is given by the Navier–Stokes equations with a body force resulting from the Stokes part. The regular velocity is obtained using a time-stepping method that combines the semi-Lagrangian method with the backward difference formula. Numerical examples are presented to demonstrate that, for both the Stokes and Navier-Stokes models, the method converges with second-order spatial and temporal accuracy.

The development of the present method is motivated by our interest in simulating biological problems with flows through biological tubes[26,27] or microfluidic devices[28]. The present model is formulated for 2D. Hence, the flow that it describes may differ significantly from flow through a tube. In future studies, the immersed interface method may be extended to compute fluid flows through a 3D open tube.

6 | ACKNOWLEDGEMENTS

Authors have no conflict of interest relevant to this article.

APPENDIX

A DERIVATION OF JUMP CONDITIONS

The derivation of the jump conditions normally requires that the interface be a closed curve. Therefore, we extend Γ to a fictitious closed and piecewise smooth curve Γ_c by taking the union of Γ, the line joining the points $\{(0, a_1), (0, a_2)\}$, and the line joining the points $\{(L, b_1), (L, b_2)\}$. See Fig. A1. Note that the jump conditions will be the same for the Stokes equations and for the Navier-Stokes equations.

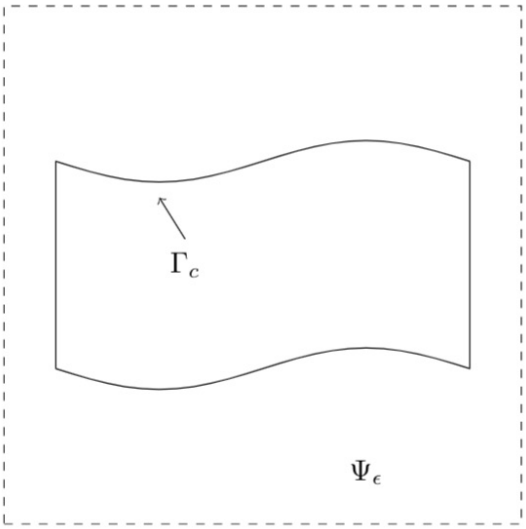

FIGURE A1 The extended computational domain, $\Psi_\epsilon = [-\epsilon, L+\epsilon] \times [-\epsilon, H+\epsilon]$ contains the fictitious closed immersed interface, Γ_c.

We first derive jump conditions for p. Let $\phi(\mathbf{x})$ be an arbitrary twice continuously differentiable test function defined on the extended domain $\Psi_\epsilon = [-\epsilon, L+\epsilon] \times [-\epsilon, H+\epsilon]$ (see Figure A1). Extend $\mathbf{F}$ to Γ_c by defining

$$\mathbf{F} = \int_{\Gamma_c} \mathbf{f}(s)\boldsymbol{\delta}(\mathbf{X}(s) - \mathbf{x}) ds \tag{A1}$$

where $\mathbf{f}$ is a piecewise smooth extension to Γ_c. Integrating the product of the divergence of the boundary force $\mathbf{F}$ and ϕ over the extended fluid domain Ψ_ϵ,

$$\iint_{\Psi_\epsilon} (\nabla \cdot \mathbf{F}) \phi dA = \iint_{\Psi_\epsilon} \left(\nabla \cdot \int_{\Gamma_c} \mathbf{f}(s)\boldsymbol{\delta}(\mathbf{X}(s) - \mathbf{x}) ds \right) \phi dA \tag{A2}$$

$$= \int_{\Gamma_c} \iint_{\Psi_\epsilon} \left(f_1(s) \frac{d}{dx}\boldsymbol{\delta}(\mathbf{X}(s) - \mathbf{x}) + f_2(s) \frac{d}{dy}\boldsymbol{\delta}(\mathbf{X}(s) - \mathbf{x}) \right) \phi ds dA$$

$$= -\int_{\Gamma_c} \left(f_1(s) \frac{d}{dx}\phi(\mathbf{X}(s)) + f_2(s) \frac{d}{dy}\phi(\mathbf{X}(s)) \right) ds. \tag{A3}$$

The last line can be obtained via integration by parts and noting that δ is zero away from Γ_c. This calculation holds on any subset of Ψ_ϵ which contains Γ_c. In particular, it holds in the belt domain $\Omega_\epsilon = \{\mathbf{x} \in \Psi_\epsilon | \min_{\mathbf{y} \in \Gamma_c} ||\mathbf{x} - \mathbf{y}|| < \epsilon\}$, which encloses the interface Γ_c. Let Γ_ϵ^+ and Γ_ϵ^- denote the inner and outer boundary of Ω_ϵ, respectively, as shown in Figure A2.

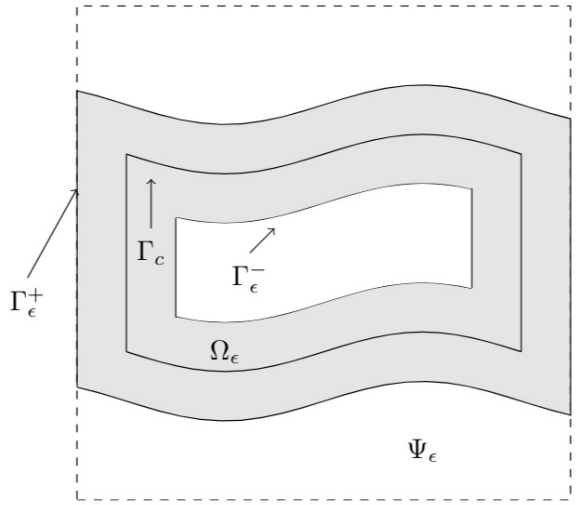

FIGURE A2 The belt domain, Ω_ϵ, is the area between the outer curve Γ_ϵ^+ and the inner curve Γ_ϵ^- that contains Γ_c.

By taking the divergence of the Stokes equation 5, one obtains a Poisson equation for pressure: $\Delta p = \nabla \cdot \mathbf{F}$. Combining this Poisson equation with Eq. A3, one obtains

$$\begin{aligned}\iint_{\Omega_\epsilon} (\Delta p)\,\phi(x,y)dA &= \iint_{\Omega_\epsilon} (\nabla \cdot \mathbf{F})\,\phi(x,y)dA \\ &= -\int_{\Gamma_c} \left(f_1(s)\frac{d}{dx}\phi(X(s),Y(s)) + f_2(s)\frac{d}{dy}\phi(X(s),Y(s)) \right) ds. \end{aligned} \tag{A4}$$

For any vector function $\mathbf{G}(x,y) = [G_1(x,y), G_2(x,y)]^T$, the Divergence Theorem states

$$\iint_{\Omega} \nabla \cdot (\phi(x,y)\mathbf{G}(x,y))dA = \int_{\partial\Omega} (\phi(x,y)\mathbf{G}(x,y)) \cdot \mathbf{n}ds. \tag{A5}$$

Since $\nabla \cdot (\phi(x,y)\mathbf{G}(x,y)) = \phi(x,y)\nabla \cdot \mathbf{G}(x,y) + \nabla\phi(x,y)\mathbf{G}(x,y)$, we can rewrite the divergence theorem as

$$\iint_{\Omega} \phi(x,y)\nabla \cdot \mathbf{G}(x,y)dA = \int_{\partial\Omega} (\phi(x,y)\mathbf{G}(x,y)) \cdot \mathbf{n}ds - \iint_{\Omega} \nabla\phi(x,y)\mathbf{G}(x,y)dA. \tag{A6}$$

Applying the divergence theorem twice to the Poisson's equation for pressure yields

$$\begin{aligned}\iint_{\Omega_\epsilon} (\Delta p)\,\phi dxdy &= \int_{\Gamma_\epsilon^+} \phi p_n^+ ds - \int_{\Gamma_\epsilon^-} \phi p_n^- ds - \iint_{\Omega_\epsilon} \nabla\phi\nabla p dA \\ &= \int_{\Gamma_\epsilon^+} \phi p_n^+ ds - \int_{\Gamma_\epsilon^-} \phi p_n^- ds - \int_{\Gamma_\epsilon^+} \phi_n p^+ ds + \int_{\Gamma_\epsilon^-} \phi_n p^- ds + \iint_{\Omega_\epsilon} p\Delta\phi dA. \end{aligned} \tag{A7}$$

The superscripts + and − indicate the values taken from the outside and inside of the interface Γ_c, respectively. Notice ϕ is twice continuously differentiable and p is bounded and only discontinuous along the interface. So as ϵ approaches zero, we have

$$\iint_{\Omega_\epsilon} p\Delta\phi dA \longrightarrow 0. \tag{A8}$$

and

$$\iint_{\Omega_\epsilon} (\Delta p)\,\phi dxdy \longrightarrow \int_{\Gamma_c} \phi[p_n]ds - \int_{\Gamma_c} \phi_n[p]ds. \tag{A9}$$

We can express $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial y}$ in terms of the normal and tangential derivatives along the interface

$$\phi_n = \nabla\phi \cdot \mathbf{n} = \frac{\partial \phi}{\partial x}\cos\theta + \frac{\partial \phi}{\partial y}\sin\theta \tag{A10}$$

$$\phi_s = \nabla\phi \cdot \boldsymbol{\tau} = \frac{\partial \phi}{\partial x}\sin\theta + \frac{\partial \phi}{\partial y}\cos\theta. \tag{A11}$$

where θ is the angle between the outward normal and the x-axis, $\mathbf{n} = (\cos\theta, \sin\theta)$, and $\boldsymbol{\tau} = (-\sin\theta, \cos\theta)$. Solving this linear equation for $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial y}$ yields

$$\frac{\partial \phi}{\partial x} = \phi_n \cos\theta - \phi_s \sin\theta \tag{A12}$$

$$\frac{\partial \phi}{\partial y} = \phi_n \sin\theta + \phi_s \cos\theta. \tag{A13}$$

Then

$$\begin{aligned}\int_{\Gamma_c} \left(f_1 \frac{\partial \phi}{\partial x} + f_2 \frac{\partial \phi}{\partial y} \right) ds &= \int_{\Gamma_c} f_1(\phi_n \cos\theta - \phi_s \sin\theta) + f_2(\phi_n \sin\theta + \phi_s \cos\theta) ds \\ &= \int_{\Gamma_c} (f_1 \cos\theta + f_2 \sin\theta)\phi_n ds + \int_{\Gamma_c} \frac{\partial}{\partial s}(f_2 \cos\theta - f_1 \sin\theta)\phi ds. \end{aligned} \tag{A14}$$

For any twice continuously differentiable test function ϕ,

$$\int_{\Gamma_c} (f_1 \cos\theta + f_2 \sin\theta)\phi_n ds + \int_{\Gamma_c} \frac{\partial}{\partial s}(-f_1 \sin\theta + f_2 \cos\theta)\phi ds = \int_{\Gamma_c} \phi[p_n] ds - \int_{\Gamma_c} \phi_n [p] ds. \tag{A15}$$

We can conclude that

$$[p] = f_1 \cos\theta + f_2 \sin\theta \tag{A16}$$

and

$$[p_n] = \frac{\partial}{\partial s}(-f_1 \sin\theta + f_2 \cos\theta). \tag{A17}$$

Next, we derive the jump conditions for the u component of velocity. The derivation for v is similar. The velocity is continuous, so we only need to derive the jump conditions for the normal derivative, u_n. We start by multiplying the u component of Eq. 5 by test function ϕ and integrating.

$$\iint_{\Omega_\epsilon} \phi\mu\Delta u - \phi \frac{dp}{dx} dA = -\iint_{\Omega_\epsilon} \phi \int_{\Gamma_c} f_1(s)\delta(\mathbf{X}(s) - \mathbf{x}) ds dA \tag{A18}$$

$$\iint_{\Omega_\epsilon} \phi\mu\Delta u dA - \iint_{\Omega_\epsilon} \phi \frac{dp}{dx} dA = -\int_{\Gamma_c} \iint_{\Omega_\epsilon} \phi f_1(s)\delta(\mathbf{X}(s) - \mathbf{x}) dA ds \tag{A19}$$

$$\iint_{\Omega_\epsilon} \phi\mu\Delta u dA - \iint_{\Omega_\epsilon} \phi \frac{dp}{dx} dA = -\int_{\Gamma_c} \phi f_1(s) ds \tag{A20}$$

The last line follows from integration by parts. Applying the divergence theorem we get

$$\iint_{\Omega_\epsilon} \phi\mu\Delta u dA = \int_{\Gamma_\epsilon^+} \phi\mu^+ \nabla u^+ \cdot \mathbf{n} ds + \int_{\Gamma_\epsilon^-} \phi\mu^- \nabla u^- \cdot (-\mathbf{n}) ds - \iint_{\Omega_\epsilon} \mu(\nabla\phi\nabla u) dA \tag{A21}$$

$$= \int_{\Gamma_\epsilon^+} \mu^+ \phi u_n^+ ds - \int_{\Gamma_\epsilon^-} \mu^- \phi u_n^- ds - \iint_{\Omega_\epsilon} \mu(\nabla\phi\nabla u) dA \tag{A22}$$

$$\to \int_{\Gamma} \phi[\mu u_n] ds - 0 \quad \text{as } \epsilon \to 0. \tag{A23}$$

The right most integral in Eq. A22 approaches zeros because ϕ is continuously differentiable and u_n is bounded and only discontinuous at the interface. Note that

$$\iint_{\Omega_\epsilon} \phi \frac{dp}{dx} dA = \iint_{\Omega_\epsilon} \phi \nabla \cdot \begin{bmatrix} p \\ 0 \end{bmatrix} dA \tag{A24}$$

$$= \int_{\Gamma_\epsilon^+} \phi([p^+, 0]^T \cdot \mathbf{n}) ds - \int_{\Gamma_\epsilon^-} \phi([p^-, 0]^T \cdot \mathbf{n}) ds - \iint_{\Omega_\epsilon} \nabla\phi \cdot \begin{bmatrix} p \\ 0 \end{bmatrix} dA \tag{A25}$$

$$\to \int_{\Gamma_c} \phi[p] \cos\theta ds + 0 \quad \text{as } \epsilon \to 0. \tag{A26}$$

This follows because ϕ is continuously differentiable and p is only discontinuous on the interface. Since ϕ is arbitrary, we must have

$$[\mu u_n] = [p]\cos\theta - f_1 \tag{A27}$$

$$= \cos\theta(f_1 \cos\theta + f_2 \sin\theta) - f_1 \tag{A28}$$

$$= \sin\theta(-f_1 \sin\theta + f_2 \cos\theta). \tag{A29}$$

Similarly, for v we can get

$$[\mu v_n] = [p]\sin\theta - f_2 \tag{A30}$$

$$= \cos\theta(f_1 \sin\theta - f_2 \cos\theta). \tag{A31}$$

This completes the derivation.

References

1. Santhanakrishnan A, Miller LA. Fluid dynamics of heart development. *Cell biochemistry and biophysics* 2011; 61(1): 1–22.
2. Tharakan A, Norton I, Fryer P, Bakalis S. Mass transfer and nutrient absorption in a simulated model of small intestine. *Journal of Food Science* 2010; 75(6): E339–E346.
3. Edwards A, Layton AT. Calcium dynamics underlying the myogenic response of the renal afferent arteriole. *American Journal of Physiology-Renal Physiology* 2014; 306(1): F34-F48.
4. Tang D, Yang C, Walker H, Kobayashi S, Ku DN. Simulating cyclic artery compression using a 3D unsteady model with fluid–structure interactions. *Computers & Structures* 2002; 80(20-21): 1651–1665.
5. Edwards A, Castrop H, Laghmani K, Vallon V, Layton A. Effects of NKCC2 isoform regulation on NaCl transport in thick ascending limb and macula densa: a modeling study. *Am J Physiol Renal Physiol* 2014; 307: F137–F146.
6. Layton A, Vallon V, Edwards A. A computational model for simulating solute transport and oxygen consumption along the nephrons. *Am J Physiol Renal Physiol* 2016; 311: F1378–F1390.
7. Layton A. Solute and water transport along an inner medullary collecting duct undergoing peristaltic contractions. *Am J Physiol Renal Physiol* 2019; 317: F735–F742.
8. Layton A, Layton H. A computational model of epithelial solute and water transport along a human nephron. *PLoS Comput Biol* 2019; 15: e1006108.
9. Arthurs KM, Moore LC, Peskin CS, Pitman EB, Layton H. Modeling arteriolar flow and mass transport using the immersed boundary method. *Journal of Computational Physics* 1998; 147(2): 402–440.
10. Peskin CS. The immersed boundary method. *Acta numerica* 2002; 11: 479–517.

11. Hou G, Wang J, Layton A. Numerical methods for fluid-structure interaction—a review. *Comm Comput Phys* 2012; 12: 337–377.

12. LeVeque RJ, Li Z. Immersed interface methods for Stokes flow with elastic boundaries or surface tension. *SIAM Journal on Scientific Computing* 1997; 18(3): 709–735.

13. Li Z, Lai MC. The immersed interface method for the Navier–Stokes equations with singular forces. *Journal of Computational Physics* 2001; 171(2): 822–842.

14. Layton A. An efficient numerical method for the two-fluid Stokes equations with a moving immersed boundary. *Computer Methods in Appl Mech Eng* 2008; 197: 2147–2155.

15. Layton A. Using integral equations and the immersed interface method to solve immersed boundary problems with stiff forces. *Computers Fluids* 2009; 38: 266–272.

16. Le DV, Khoo BC, Peraire J. An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries. *Journal of Computational Physics* 2006; 220(1): 109–138.

17. Li Y, Sgouralis I, Layton AT. Computing Viscous Flow in an Elastic Tube.. *Numerical Mathematics: Theory, Methods & Applications* 2014; 7(4).

18. Nganguia H, Young YN, Layton A, Lai MC, Hu WF. Electrohydrodynamics of a viscous drop with inertia. *Phys Rev E* 2016; 93: 053114.

19. Li Y, Williams SA, Layton AT. A hybrid immersed interface method for driven Stokes flow in an elastic tube. *Numerical Mathematics: Theory, Methods and Applications* 2013; 6(4): 600–616.

20. Rosar M, Peskin CS. Fluid flow in collapsible elastic tubes: a three-dimensional numerical model. *New York J. Math* 2001; 7: 281–302.

21. Smith KM, Moore LC, Layton HE. Advective transport of nitric oxide in a mathematical model of the afferent arteriole. *American Journal of Physiology-Renal Physiology* 2003; 284(5): F1080–F1096.

22. Rosar ME. A three-dimensional computer model for fluid flow through a collapsible tube. *New York Journal of Mathematics* 1994.

23. Lai MC, Li Z. A remark on jump conditions for the three-dimensional Navier-Stokes equations involving an immersed moving membrane. *Applied mathematics letters* 2001; 14(2): 149–154.

24. Layton AT. Cubic spline collocation method for the shallow water equations on the sphere. *Journal of Computational Physics* 2002; 179(2): 578–592.

25. Beale JT, Layton AT. A velocity decomposition approach for moving interfaces in viscous fluids. *Journal of Computational Physics* 2009; 228(9): 3358–3367.

26. Layton A. Feedback-mediated dynamics in a model of a compliant thick ascending limb. *Math Biosci* 2010; 228: 185–194.

27. Chen J, Edwards A, Layton A. Effects of pH and medullary blood flow on oxygen transport and sodium reabsorption in the rat outer medulla. *Am J Physiol Renal Physiol* 2010; 298: F1369–F1383.

28. Mandal HS, Su Z, Ward A, Tang XS. Carbon nanotube thin film biosensors for sensitive and reproducible whole virus detection. *Theranostics* 2012; 2(3): 251.