Computing viscous flow along a 2D open channel using the immersed interface method

Sarah E. Patterson1 | Anita T. Layton2

1Applied Mathematics Department, Virginia Military Institute, Lexington, Virginia, USA
2Departments of Applied Mathematics and Biology, Cheriton School of Computer Science, and School of Pharmacy, University of Waterloo, Waterloo, Ontario, Canada

Correspondence
Sarah Patterson, Applied Mathematics Department, Virginia Military Institute, Lexington, VA, USA.
Email: pattersonse@vmi.edu

Funding information
Canada 150 Research Chair program; National Institute of Diabetes and Digestive and Kidney Diseases, Grant/Award Number: R01DK106102

Abstract
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
finite difference, fluid dynamics, fluid–structure interaction, immersed boundary problem, Navier–Stokes, 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. Small changes in the diameter of the channel can cause drastically different fluid behavior. For example, the main mechanism for blood pumping in a valveless heart is thought to be peristalsis, a wave-like contraction.1-3 Deformable channels or tubes are also used to model obstructed ureters,4 food mixing in the intestine,5 and blood flow through a vessel6,7 or renal tubules.8,9

A natural way to model flow along a compliant channel is to frame it as an immersed boundary problem.10-14 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. The immersed boundary applies a singular force to the underlying fluid, which is often described by Navier–Stokes flow, but can be described by Stokes flow for highly viscous fluids. The immersed boundary method transfers the singular boundary forces onto the underlying fluid using approximate (smooth) Dirac delta functions typically with $O(h)$ 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.

If better accuracy is desired, especially near the immersed boundary, one may use the immersed interface method developed by LeVeque and Li. The immersed interface method captures the sharp jumps in the solution and its derivatives 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.

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, 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.

This study develops 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 accomplish this, 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 domain given by $\Psi = [0, L] \times [-H, H]$ with length $L$ and width $2H$. Let $\Gamma$ represent the compliant walls of the channel immersed in $\Psi$. Specifically, $\Gamma$ is an interface immersed in the fluid formed by two distinct smooth curves $X = G_1(s)$ 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 $X = G_2(s)$ 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 $\Gamma$ oriented toward $\partial \Psi$ away from the channel.

![Figure 1](image-url)
2.2 | Boundary conditions

To properly state the problem, appropriate boundary conditions must be imposed on fluid velocity and pressure. Biperiodic 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 \( P_{\text{diff}} \), at the inlet and outlet of the channel

\[
p(0, y) = p(L, y) + \begin{cases} 
  P_{\text{diff}} & a_1 \leq y \leq a_2 \\
  0 & \text{else} 
\end{cases}.
\]

(1)

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

\[
\frac{\partial p}{\partial x} \bigg|_{(x=0,y)} = \frac{\partial p}{\partial x} \bigg|_{(x=L,y)}.
\]

(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.

Note that boundary conditions at the inflow and outflow of a channel are a rich research area.\(^{27-29}\) These boundary conditions were chosen for simplicity, but this method can be used with a wide range of boundary conditions that would allow for the accumulation and release of fluid in the channel.

2.3 | Fluid–structure interactions

The Navier–Stokes equations are given by

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

(3)

\[\nabla \cdot \mathbf{u} = 0\]

(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; \( \mu \) is viscosity; and \( \mathbf{F} \) is the interfacial force, which is singularly supported along \( \Gamma \) (see below). We consider also the zero Reynolds number regimes given by the Stokes momentum equation:

\[\mu \Delta \mathbf{u} - \nabla p + \mathbf{F} = 0\]

(5)

and the continuity equation (Equation (4)).

The force exerted by the interface \( \Gamma \) can be written as

\[\mathbf{F}(\mathbf{x}, t) = \int_{\Gamma} f(\alpha) \delta(\mathbf{x} - \mathbf{X}(\alpha)) d\alpha,\]

(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 \( t \), \( f(\alpha) \) is the force strength at the point \( \mathbf{X}(\alpha) \), and \( \delta \) is the Dirac delta function.

The force strength \( f(\alpha) \) from Equation (6) has two major components, an elastic force \( f_E \) and a tether force \( f_T \):

\[f(\alpha) = f_E(\alpha) + f_T(\alpha).\]

(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
\[ \mathbf{f}_E(s, t) = \frac{\partial}{\partial s} (T(s, t) \mathbf{r}(s, t)) \]  

(8)

where the unit tangent vector to \( \Gamma \) is given by

\[ \mathbf{r}(s, t) = \frac{\partial \mathbf{X}/\partial s}{|\partial \mathbf{X}/\partial s|}. \]  

(9)

The tension \( T(s, t) \) in Equation (8) is given by

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

(10)

where \( a_E \) controls the stiffness of the interface.

The interface is tethered to a set of anchor points evenly distributed along the length of the interface. Interface movement can either be restricted by using stationary anchor points or be induced by moving the position of the anchor points. The second force component \( \mathbf{f}_T \) arises from the displacement of the interface control knots from their anchor points.

Suppose the interface \( \mathbf{X} \) is discretized by a set of interface control knots \( \{ \mathbf{X}_i \}_{i=1}^{N_b} \) where \( N_b \) is the total number of interface control knots. Each control knot in \( \{ \mathbf{X}_i \}_{i=1}^{N_b} \) is attached to an anchor point in \( \{ \mathbf{X}_i \}_{i=1}^{N_b} \) by a spring with resting length 0. Figure 2 shows three interface control knots \( \mathbf{X}_{i-1}, \mathbf{X}_i, \) and \( \mathbf{X}_{i+1} \) connected to the tether points \( \mathbf{X}_{i-1}, \mathbf{X}_i, \) and \( \mathbf{X}_{i+1}, \) respectively, by a spring. Let \( a_T \) be the spring force constant. Then the tether force is given by

\[ \mathbf{f}_T = a_T (\mathbf{X} - \mathbf{X}). \]  

(11)

Thus, if the interface knots move away from their anchor points, restorative forces are generated.

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

\[ \frac{d\mathbf{X}}{dt} = \mathbf{u}(\mathbf{X}). \]  

(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 \( \Gamma \) to a fictitious closed and piecewise smooth curve \( \Gamma_c \) by taking the union of \( \Gamma \), 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 Equations (13)–(16) are the same for the Stokes equations and for the Navier–Stokes equations. This follows from the no-slip condition assumed for the interfacial motion, which generates a continuous velocity field and material derivative across the interface.30

\[ [p] = \mathbf{f} \cdot \mathbf{n} \]  

(13)

\[ [p_n] = \frac{\partial}{\partial s} (\mathbf{f} \cdot \mathbf{r}) \]  

(14)
\[ [u] = 0 \]
\[ \mu [u_n] = -(F \cdot \tau) \tau, \]
where the jump \([ \cdot ]\) of a quantity, say \(q(X, t)\), at time \(t\) and point \(X \in \Gamma\) is
\[ [q(X, t)] = \lim_{\varepsilon \to 0^+} q(X + \varepsilon n, t) - \lim_{\varepsilon \to 0^-} q(X + \varepsilon n, t). \]

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 (Equations (4) and (5)), we compute the fluid velocity and pressure on a fixed Eulerian grid:
\[ \Omega_h = \left\{ x_{ij} = \left( jh, ih - \frac{H}{2} \right) \bigg| i \in 1, \ldots, N_x \text{ and } j \in 1, \ldots, N_y \right\}, \]
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 \(\Gamma\). The interface position at time \(t_n = n \Delta t\) is tracked by \(N_b\) boundary markers \(X^n = \{X^n_i\}_{i=1}^{N_b}\) that are connected by a cubic spline.

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 Equation (5) to yield
\[ \Delta p = \nabla \cdot F, \]
the other two being the two velocity components of Equation (4). Specifically, Equations (4) and (18) are discretized using second-order finite difference, with jump conditions Equations (13)–(16) incorporated into the finite difference stencil as correction terms. For details on computing the correction terms, see Reference 16.

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. This approach leverages the fact that, for a given singular interfacial force, the jumps in the solutions (see Equations (13)–(16)) are identical for both the Stokes equations and the Navier–Stokes equations.\(^{16,17}\)

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 \(u_s\) and \(p_s\)), and a regular part (\(u_r\) and \(p_r\)):
\[ u = u_s + u_r, \quad p = p_s + p_r \]

Taking the difference between Equations (3) and (5), we determine that the regular part, \(u_r\) and \(p_r\), satisfies
\[ \rho \left( \frac{\partial u_r}{\partial t} + u \cdot \nabla u_r \right) = \mu \Delta u_r - \nabla p_r + F_b \]
where \(F_b\) is a body force
\[ F_b = -\rho \left( \frac{\partial u_s}{\partial t} + u \cdot \nabla u_s \right). \]
Note that the full velocity \( \mathbf{u} \) is used in the transport of \( \mathbf{u}_r \) and in \( \mathbf{F}_b \). Then substituting Equation (19) into Equation (4) yields

\[
\nabla \cdot \mathbf{u} = 0
\]  
(22)

To advance the full solution from \( t_n \) to \( t_{n+1} \), the Stokes part \( (\mathbf{u}_r^{n+1}, p_r^{n+1}) \) 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 20 and 22 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 \( \mathbf{u}_r^{n,n+1} \) using Equation (20) alone and then project \( \mathbf{u}_r^{n,n+1} \) onto the divergence free space to yield \( \mathbf{u}_r^{n+1} \) and \( p_r^{n+1} \).

To compute \( \mathbf{u}_r^{n,n+1} \), we first rewrite Equation (20) in terms of the material derivative as follows

\[
\rho \frac{D\mathbf{u}_r}{Dt} = -\nabla p_r + \mu \nabla^2 \mathbf{u}_r + \mathbf{F}_b.
\]  
(23)

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 performed by integrating

\[
\frac{dx(t)}{dt} = \mathbf{u}(\mathbf{x}(t), t), \quad \mathbf{x}(t_{n+1}) = \mathbf{x}_0,
\]  
(24)

backward 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 \( \mathbf{u}_r^n = \mathbf{u}_r(\mathbf{x}^n, t_n) \) and \( \mathbf{u}_r^{n-1} = \mathbf{u}_r(\mathbf{x}^{n-1}, t_{n-1}) \) are approximated using spatial interpolation. The material derivative is then approximated using the second-order backward difference formula,

\[
\rho \left( \frac{3\mathbf{u}_r^{n,n+1} - 4\mathbf{u}_r^n + \mathbf{u}_r^{n-1}}{2\Delta t} \right) + \nabla p_r^n = \mu \Delta \mathbf{u}_r^{n,n+1} + \mathbf{F}_b^{n+1}.
\]  
(25)

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

\[
\Delta \phi^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}_r^{n,n+1}.
\]  
(26)

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

\[
\mathbf{u}_r^{n+1} = \mathbf{u}_r^{n,n+1} - \Delta t \nabla \phi^{n+1}
\]  
(27)

\[
\nabla p_r^{n+1} = \nabla p_r^n + \frac{3}{2} \nabla \phi^{n+1} - \mu \Delta t \nabla^3 \phi^{n+1}.
\]  
(28)

The full solution is given by Equation (19). Details for the velocity decomposition method can be found in Reference 32.

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 biperiodic boundary conditions are prescribed for the full velocity solution. This is achieved by imposing biperiodic boundary conditions for both \( \mathbf{u}_r \) and \( \mathbf{u}_r \) as well as the intermediate solution \( \mathbf{u}_r^n \).
The full pressure solution must satisfy the inhomogeneous periodic boundary conditions given by Equations (1) and (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 biperiodic boundary conditions on $p_r$. Biperiodic boundary conditions are assumed for the auxiliary variable $\phi$.

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 $\Omega = [0, 4.25 \, \mu m] \times [-1.625, 1.625 \, \mu m]$. The interface is tethered to its resting position as a channel with constant width $R = 1.42 \, \mu m$ centered at $y = 0$. The lower and upper segments of the interface are initially displaced in the $y$ direction according to the functions

\[ G_1(x) = -0.2e^{-(x - \frac{L}{2})^2} - 0.5R \]  
\[ G_2(x) = 0.2e^{-(x - \frac{L}{2})^2} + 0.5R \]

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

For the Stokes problem, the fluid solutions are steady and only indirectly depend on time because the position of the interface may be time dependent. Therefore, we conducted a spatial grid refinement study at $t = 0$. This goal of this study is to test the spatial convergence of method without added temporal errors (the spatial convergence with temporal errors are presented in Table 4). Computed solutions were compared with an $N_x = 1089$ by $N_y = 1666$ high-resolution numerical solution. The order of convergence for a solution component $w$ is found using the formula

\[ \text{Order} = \log_2 \left( \frac{||w_{\frac{h}{2}} - w^*||_\infty}{||w_h - w^*||_\infty} \right) \]  

where $w^*$ is the $N_x = 1089$ by $N_y = 1666$ high-resolution solution, $w_h$ is the solution computed with spatial grid size $h$, $w_{\frac{h}{2}}$ is the solution computed with spatial grid size $\frac{h}{2}$, and $|| \cdot ||_\infty$ is the infinity norm.

**Table 1** Parameters used for the spatial convergence study for 2D Stokes flow in rectangular coordinates

| Parameter                  | Symbol | Value                        |
|----------------------------|--------|------------------------------|
| Viscosity                  | $\mu$  | 0.1 gm/(cm·s)               |
| Density                    | $\rho$ | 0.1 gm/cm$^3$               |
| Domain length              | $L$    | 4.25 $\mu$m                 |
| Domain height              | $H$    | 3.25 $\mu$m                 |
| Resting channel width      | $R$    | 0.7 $\mu$m                  |
| Interface control points   | $N_b$  | 100                          |
| Simulation length          | $T$    | 120 ms                       |
| Time steps                 | $N_t$  | 60                           |
| Time step size             | $\Delta t$ | 2e − 3 ms                  |
| Tether force constant      | $a_{\text{tether}}$ | 5 gm/s$^2$             |
| Elastic force constant     | $a_{\text{elastic}}$ | 1 gm/s$^2$              |
| Reynolds number            | $Re$   | 1.4000e-08                  |
| Pressure difference        | $P_{\text{diff}}$ | 2.544e-4 gm/(s$^2 \, \mu m$) |
Results in Table 2 indicate that second-order spatial convergence was achieved for pressure and both components of velocity. Furthermore, the jump discontinuities 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.

## 4.2 Navier–Stokes results

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

### 4.2.1 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 \( 0.125 \) s and then held constant according to the function:

\[
P_{\text{diff}}(t) = \begin{cases} \frac{3\mu L}{R^2} \sin \left( \frac{\pi}{2} \frac{t}{1/8} \right) & t < 1/8 \\ \frac{3\mu L}{R^2} & t \geq 1/8. \end{cases}
\] (32)

where \( R = W/2 \). The channel boundaries are tethered to its initial and resting position, that is, a channel with a width of \( W \). Table 3 shows the remainder of the parameters.

| Parameter                  | Symbol | Value                  |
|----------------------------|--------|------------------------|
| Viscosity                  | \( \mu \) | 0.0175 gm/(cm-s)     |
| Density                    | \( \rho \) | 1.055 gm/cm\(^3\)   |
| Average velocity           | \( \bar{u} \) | 1 \( \mu \)m/s       |
| Domain length              | \( L \) | 32 \( \mu \)m         |
| Domain height              | \( H \) | 16 \( \mu \)m         |
| Resting channel width      | \( W \) | 10.1 \( \mu \)m       |
| Simulation interval        | \( T \) | 1 s                    |
| Number of time steps       | \( N_t \) | 2560                   |
| Time step size             | \( \Delta t \) | 4e-4 s                 |
| Tether force constant      | \( a_{\text{tether}} \) | 1e-4 gm/s\(^2\)  |
| Elastic force constant     | \( a_{\text{elastic}} \) | 1e-4 gm/s\(^2\)  |
| Reynolds number            | \( \text{Re} \) | 5.9783e-06           |
| Max pressure difference    | \( P_{\text{diff}} \) | 6.6e-5 gm/(s\(^2\)\(\mu\)m) |
At steady state, pressure gradient and velocity are described by 2D Poiseuille flow, that is,

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

(33)

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

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

In Figure 3 a solid line shows the expected Poiseuille velocity profile given the parameters of the simulation. The computed \( u \)-component of the velocity profile is shown on three slices along the length of the channel at time \( t = 0.05 \) s. The velocity profile on the slice where \( x = 1, x = 16, \) and \( x = 29 \) \( \mu \)m are denoted by the \( \bigcirc \), \( \triangle \), and \( \square \) symbols, respectively. While the computed \( u \) exhibits an approximate parabolic profile along the entire length of the channel, it is particularly noteworthy that the results near the inlet and outlet of the channel exhibit this expected behavior.

The computed \( p \) is shown in Figure 4. The interface segments are located at approximately \( y = \pm 5.05 \) \( \mu \)m. It can be seen that \( p \) decreases linearly 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 number of interface control points for each segment of the interface is \( N_b = \frac{N_i + 1}{2} \). The solutions are compared with high-resolution solutions computed on a \( 1025 \times 513 \) grid at time \( t = 0.1 \) s when the solution is still transitioning from its initial solution to its steady-state solution. The order of convergence for a solution component \( w \) is found using the formula

![Figure 3](image-url)  
**Figure 3** The velocity profile of the computed solution for \( t = 0.5 \) s at the cross-section where \( x = 1, x = 16, \) and \( x = 31 \) \( \mu \)m are shown with \( \bigcirc \), \( \triangle \), and \( \square \) symbols, respectively. These are compared with expected velocity profile of 2D Poiseuille flow (solid blue line).

![Figure 4](image-url)  
**Figure 4** 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 (solid black line) located near \( y = \pm 5.05 \) \( \mu \)m.
where \( w^* \) is the \( N_x = 1025 \) by \( N_y = 513 \) high-resolution solution, \( w_h \) is the solution computed with spatial grid size \( h \), \( w_{h_2} \) is the solution computed with spatial grid size \( h/2 \), and \( || \cdot ||_\infty \) is the infinity norm. Convergence results indicate that the method is approximately second-order accurate in space; see Table 4.

### 4.2.2 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 \( x \) for the upper and lower interface, denoted \( Y_{\text{top, tether}}(x) \) and \( Y_{\text{bottom, tether}}(x) \), respectively, are given by

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

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

The fluid solutions are computed in the domain with a length of 61 \( \mu \)m and width 21 \( \mu \)m. 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

| Parameter          | Symbol | Value                      |
|--------------------|--------|---------------------------|
| Viscosity          | \( \mu \) | 0.0175 gm/(cm\( \cdot \)s) |
| Density            | \( \rho \) | 1.055 gm/cm\(^3\)         |
| Domain length      | \( L \) | 61 \( \mu \)m              |
| Domain height      | \( H \) | 21 \( \mu \)m              |
| X-grid points      | \( N_x \) | 489                       |
| Y-grid points      | \( N_y \) | 169                       |
| Spatial grid size  | \( h \) | 0.012 \( \mu \)m           |
| Initial channel radius | \( R \) | 5.05 \( \mu \)m          |
| Vessel control points | \( N_b \) | 100                       |
| Simulation length  | \( T \) | 12.5 \( \mu \)s            |
| Tether force constant | \( a_{\text{tether}} \) | 1e-4 gm/s\(^2\)       |
| Elastic force constant | \( a_{\text{elastic}} \) | 1e-4 gm/s\(^2\)    |
| Reynolds number    | \( Re \) | 5.9783e-06                |
| Pressure difference | \( P_{\text{diff}} \) | 6.6e-5 gm/(s\(^2\)\(\mu\)m) |
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. The order of convergence for a solution component $w$ is found using the formula

$$\text{Order} = \log_2 \left( \frac{||w_{\Delta t} - w^*||_\infty}{||w_{\frac{\Delta t}{2}} - w^*||_\infty} \right)$$

(37)

where $w^*$ is the high-resolution solution computed with time steps of size $\Delta t = 1.8e \, -7$, $w_{\Delta t}$ is the solution computed with time steps of size $\Delta t$, $w_{\frac{\Delta t}{2}}$ is the solution computed with time steps of size $\frac{\Delta t}{2}$, and $|| \cdot ||_\infty$ is the infinity norm. Convergence results, given in Table 6, indicate that this method can achieve second-order accuracy in time.

### 4.2.3 Stenosed channel

Finally, flow through a stenosed channel is presented to showcase a potential application of this method. The fluid solutions are computed in the domain with a length of 32 $\mu m$ and width 16 $\mu m$. In this example, the interface walls are tethered in their initial configuration as a straight channel with an indentation centered at $x = 21 \, \mu m$ which represents a stenosed channel. The $y$-position of the tethers at position $x$ at time $t$ for the upper and lower interface, denoted $Y_{\text{top, tether}}(x, t)$ and $Y_{\text{bottom, tether}}(x, t)$, respectively, are given by

$$Y_{\text{top, tether}}(x, t) = \begin{cases} R & x \leq 16 \\ R - 1.4 \left( \cos \left( \frac{\pi (x - 21)}{5} \right) + 1 \right) & 16 < x < 26 \\ 26 \leq x \end{cases}$$

(38)

$$Y_{\text{bottom, tether}}(x, t) = \begin{cases} -R & x \leq 16 \\ -R + 1.4 \left( \cos \left( \frac{\pi (x - 21)}{5} \right) + 1 \right) & 16 < x < 26 \\ -26 \leq x \end{cases}$$

(39)

The parameters for this application are identical to those presented in Table 3.

To test the spatial convergence of the method, the solution was computed at various spatial resolutions. The number of interface control points for each segment of the interface is $N_y = \frac{N_{\Delta t} + 1}{2}$. The solutions are compared with high-resolution solutions computed on a $513 \times 257$ grid at time $t = 0.1 \, s$. The order of convergence for a solution component $w$ is found using the formula

$$\text{Order} = \log_2 \left( \frac{||w_{\frac{h}{2}} - w^*||_\infty}{||w_h - w^*||_\infty} \right)$$

(40)

where $w^*$ is the $N_x = 513$ by $N_y = 257$ high-resolution solution, $w_h$ is the solution computed with spatial grid size $h$, $w_{\frac{h}{2}}$ is the solution computed with spatial grid size $\frac{h}{2}$, and $|| \cdot ||_\infty$ is the infinity norm. Convergence results indicate that the method is approximately second-order accurate in space; see Table 7.

| Time step | $p_x$ | $p_y$ | $u$ | $v$ |
|-----------|-------|-------|-----|-----|
| $N_t$ | $\Delta t$ | $|| \cdot ||_\infty$ | Order | $|| \cdot ||_\infty$ | Order | $|| \cdot ||_\infty$ | Order |
| 1.0e-06 $\times$ | 1.0e-06 $\times$ | 1.0e-06 $\times$ |
| 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 |
The results for this study indicate that even when the interface deviates substantially from a straight channel, the spatial convergence remains above first order, although second-order convergence was not quite achieved. The following images were generated from $129 \times 65$ resolution solution at time $t = 0.1$ s. The position of the interface at time $t = 0.1$ s is shown as a black line in Figure 5. The blue arrows indicate the direction and magnitude of the fluid velocity through the channel at time $t = 0.1$ s. Notice that flow is fastest near the center of the channel. Above the stenosed channel, the fluid pushes slightly toward the wall then slows down through the stenosed section of the channel. There is very little flow outside of the interface.

The pressure in the computational domain is shown in Figure 6. It is worth noting that the pressure decreases almost linearly throughout the channel. This constant pressure gradient drives the fluid flow through the channel. The pressure is approximately zero outside of the channel and exhibits a sharp jump across the interface.

### 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. 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 the method’s performance for both the Stokes and Navier–Stokes
problems. In the Stokes model, the interface is displaced from and allowed to return to its resting position as a straight
channel. Three numerical cases were run for the Navier–Stokes model: a straight channel, a straight channel that is
actively deformed by moving the tether anchor points, and a stenosed channel. These numerical cases indicate that the
method converges with second-order spatial and temporal accuracy when the interface undergoes small displacements
and near second-order spatial accuracy then the interface substantially deviates from a straight channel (as was the case
with the stenosed channel). In that study, there were points of high pressure near the intersection of the interface and the
inlet and outlet walls. This may be the cause of the loss of accuracy in the case of large interface displacement. Regardless,
this method has many potential applications for flow in biological tubes where the tube is pliable or actively moving, but
does not undergo drastic deformation. One such application is the myogenic response of vessels in the microcirculation
to sudden changes in blood pressure.33

All experiments in this article were in the low Reynolds number regime. Future work would also include optimizing
the implementation of this method and running experiments with higher Reynolds numbers.

The development of the present method is motivated by our interest in simulating biological problems with flows
through biological tubes34,35 or microfluidic devices.36 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. Additional work is also being conducted on using this method to
model congestion in the microcirculation and the myogenic response of vessels in the microcirculation to sudden changes
in blood pressure.

ACKNOWLEDGMENTS
This research was supported by the Canada 150 Research Chair program and the National Institutes of Health (National
Institute of Diabetes and Digestive and Kidney Diseases Grant R01DK106102).

PEER REVIEW INFORMATION
Engineering Reports thanks Mehdi Jabbarzadeh and other anonymous reviewers for their contribution to the peer review
of this work.

DATA AVAILABILITY STATEMENT
The data that support the findings of this study are available from the corresponding author upon reasonable request.

CONFLICT OF INTEREST
Authors have no conflict of interest relevant to this article.

AUTHOR CONTRIBUTIONS
Sarah Patterson Conceptualization-Supporting; Data curation-Equal; Formal analysis-Equal; Investigation-
Equal; Methodology-Equal; Software-Lead; Validation-Lead; Visualization-Lead; Writing-original draft-Equal;
Writing-review & editing-Equal. Anita Layton Conceptualization-Equal; Formal analysis-Equal; Funding acquisition-
Lead; Investigation-Equal; Methodology-Equal; Project administration-Lead; Software-Supporting; Supervision-Lead;
Validation-Supporting; Writing-original draft-Equal; Writing-review & editing-Equal.

ORCID
Sarah E. Patterson https://orcid.org/0000-0002-4430-882X

REFERENCES
1. Santhanakrishnan A, Miller LA. Fluid dynamics of heart development. Cell Biochem Biophys. 2011;61(1):1-22.
2. Esser F, Masselter T, Speck T. Silent pumpers: a comparative topical overview of the peristaltic pumping principle in living nature,
   engineering, and biomimetics. Adva Intell Syst. 2019;1(2):1900009.
3. Waldrop L, Miller L. Large-amplitude, short-wave peristalsis and its implications for transport. Biomech Model Mechanobiol.
   2016;15(3):629-642.
4. Takaddus AT, Gautam P, Chandy AJ. A fluid-structure interaction (FSI)-based numerical investigation of peristalsis in an obstructed
   human ureter. Int J Numer Methods Biomed Eng. 2018;34(9):e3104.
5. Tharakan A, Norton I, Fryer P, Bakalis S. Mass transfer and nutrient absorption in a simulated model of small intestine. *J Food Sci.* 2010;75(6):E339-E346.

6. Mukhopadhyay S, Mandal MS, Mukhopadhyay S. Dynamic response of pulsatile flow of blood in a stenosed tapered artery. *Math Methods Appl Sci.* 2018;41(10):3885-3899.

7. Xiao L, Lin C, Chen S, Liu Y, Fu B, Yan W. Effects of red blood cell aggregation on the blood flow in a symmetrical stenosed microvessel. *Biomech Model Mechanobiol.* 2020;19(1):159-171.

8. Layton A. Solute and water transport along an inner medullary collecting duct undergoing peristaltic contractions. *Am J Physiol Renal Physiol.* 2019;317:F735-F742.

9. Layton A, Layton H. A computational model of epithelial solute and water transport along a human nephron. *PLoS Comput Biol.* 2019;15:e1006108.

10. Arthurs KM, Moore LC, Peskin CS, Pitman EB, Layton H. Modeling arteriolar flow and mass transport using the immersed boundary method. *J Comput Phys.* 1998;147(2):402-440.

11. Peskin CS. The immersed boundary method. *Acta Numerica.* 2002;11:479-517.

12. Iaccarino G, Verzicco R. Immersed boundary technique for turbulent flow simulations. *Appl Mech Rev.* 2003;56(3):331-347.

13. Chen ZL, Hickel S, Devesa A, Berland J, Adams NA. Wall modeling for implicit large-eddy simulation and immersed-interface methods. *Theor Comput Fluid Dyn.* 2014;28(1):1-21.

14. Ma M, Huang WX, Xu CX. A dynamic wall model for large eddy simulation of turbulent flow over complex/moving boundaries based on the immersed boundary method. *Phys Fluids.* 2019;31(11):115101.

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

16. LeVeque RJ, Li Z. Immersed interface methods for Stokes flow with elastic boundaries or surface tension. *SIAM J Sci Comput.* 1997;18(3):709-735.

17. Li Z, Lai MC. The immersed interface method for the Navier–Stokes equations with singular forces. *J Comput Phys.* 2001;171(2):822-842.

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

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

20. Le DV, Khoo BC, Peraire J. An immersed interface method for viscous incompressible flows involving rigid and flexible boundaries. *J Comput Phys.* 2006;220(1):109-138.

21. Li Y, Sgouralis I, Layton AT. Computing viscous flow in an elastic tube. *Numer Math Theory Methods Appl.* 2014;7(4):555-574.

22. Nganguia H, Young YN, Layton A, Lai MC, Hu WF. Electrohydrodynamics of a viscous drop with inertia. *Phys Rev E.* 2016;93(6):053114.

23. Li Y, Williams SA, Layton AT. A hybrid immersed interface method for driven Stokes flow in an elastic tube. *Numer Math Theory Methods Appl.* 2013;6(4):600-616.

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

25. Smith KM, Moore LC, Layton HE. Advection of nitric oxide in a mathematical model of the afferent arteriole. *Am J Physiol Renal Physiol.* 2003;284(5):F1080-F1096.

26. Rosar ME. A three-dimensional computer model for fluid flow through a collapsible tube. *New York Journal of Mathematics.* 1994;7:281-302.

27. Li Y, Yngvason J. A simple and efficient outflow boundary condition for the incompressible Navier–Stokes equations. *Eng Appl Comput Fluid Mech.* 2017;11(1):69-85.

28. Bruneau CH, Tancogne S. Far field boundary conditions for incompressible flows computation. *J Appl Anal Comput.* 2018;8(3):690-709.

29. Dong S, Kariotakakis GE, Chryssostomidis C. A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains. *J Comput Phys.* 2014;261:83-105.

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

31. Layton AT. Cubic spline collocation method for the shallow water equations on the sphere. *J Comput Phys.* 2002;179(2):578-592.

32. Beale JT, Layton AT. A velocity decomposition approach for moving interfaces in viscous fluids. *J Comput Phys.* 2009;228(9):3358-3367.

33. Hong KS, Kim K, Hill MA. Regulation of blood flow in small arteries: mechanosensory events underlying myogenic vasoconstriction. *J Exercise Rehabilit.* 2020;16(3):207.

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

35. 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.

36. 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.
APPENDIX A. DERIVATION OF JUMP CONDITIONS

The derivation of the jump conditions normally requires that the interface be a closed curve. Recall that the interface $\Gamma$ was formed by two distinct smooth curves that intersect the computational domain boundary $\partial \Psi$ at $(0, a_1)$ and $(L, b_1)$ and at $(0, a_2)$ and $(L, b_2)$, respectively. Since $\Gamma$ is not closed, we extend $\Gamma$ to a fictitious closed and piecewise smooth curve $\Gamma_c$ by taking the union of $\Gamma$, the line joining the points $\{(0, a_1), (0, a_2)\}$, and the line joining the points $\{(L, b_1), (L, b_2)\}$. See Figure A1. Note that the jump conditions will be the same for the Stokes equations and for the Navier–Stokes equations.

In this section, we use the notation $[\cdot]$ to represent the jump discontinuity of a quantity, say $q(X, t)$, at time $t$ and point $X \in \Gamma_c$, where

$$[q(X, t)] = \lim_{\epsilon \to 0^+} q(X + \epsilon n, t) - \lim_{\epsilon \to 0^-} q(X - \epsilon n, t).$$  \hspace{1cm} (A1)

We first derive jump conditions for $p$. Let $\phi(x)$ 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 $F$ to $\Gamma_c$ by defining

$$F = \int_{\Gamma_c} f(s) \delta(X(s) - x) ds \hspace{1cm} (A2)$$

where $f$ is a piecewise smooth extension to $\Gamma_c$. Integrating the product of the divergence of the boundary force $F$ and $\phi$ over the extended fluid domain $\Psi_\epsilon$,

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

$$= \int_{\Gamma^+} \int_{\Psi_\epsilon} \left( f_1(s) \frac{d}{dx} \delta(X(s) - x) + f_2(s) \frac{d}{dy} \delta(X(s) - y) \right) \phi ds dA$$

$$= - \int_{\Gamma^+} \left( f_1(s) \frac{d}{dx} \phi(X(s)) + f_2(s) \frac{d}{dy} \phi(X(s)) \right) ds. \hspace{1cm} (A4)$$

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

By taking the divergence of the Stokes equation 5, one obtains a Poisson equation for pressure: $\Delta p = \nabla \cdot F$. Combining this Poisson equation with Equation (A4), one obtains

\[ \text{FIGURE A1} \quad \text{The fictitious closed interface } \Gamma_c \text{ is the union of } \Gamma, \text{ the line joining the points } \{(0, a_1), (0, a_2)\}, \text{ and the line joining the points } \{(L, b_1), (L, b_2)\}. \]

\[ \text{The extended computational domain, } \Psi_\epsilon = [-\epsilon, L + \epsilon] \times [-\epsilon, H + \epsilon] \text{ contains the original computational domain, } \Psi = [0, L] \times [0, H] \text{ and the entire closed immersed interface} \]
The belt domain, $\Omega_\epsilon$, is the area between the outer curve $\Gamma_\epsilon^+$ and the inner curve $\Gamma_\epsilon^-$ that contains $\Gamma_\epsilon$.

\[
\int\int_{\Omega_\epsilon} (\Delta p)\phi(x,y) dA = \int\int_{\Omega} (\nabla \cdot \mathbf{F})\phi(x,y) dA
\]

\[
= -\int_{\Gamma_\epsilon} \left( f_1(s) \frac{d}{ds} \phi(X(s), Y(s)) + f_2(s) \frac{d}{dy} \phi(X(s), Y(s)) \right) ds.
\]  

(A5)

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

\[
\int\int_{\Omega} \nabla \cdot (\phi(x,y)\mathbf{G}(x,y)) dA = \int_{\partial\Omega} (\phi(x,y)\mathbf{G}(x,y)) \cdot \mathbf{n} ds.
\]  

(A6)

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

\[
\int\int_{\Omega} \phi(x,y)\nabla \cdot \mathbf{G}(x,y) dA = \int_{\partial\Omega} (\phi(x,y)\mathbf{G}(x,y)) \cdot \mathbf{n} ds - \int\int_{\Omega} \nabla \phi(x,y)\mathbf{G}(x,y) dA.
\]  

(A7)

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

\[
\int\int_{\Omega_\epsilon} (\Delta p)\phi(x,y) dA = \int_{\Gamma_\epsilon^+} \phi p_+ ds - \int_{\Gamma_\epsilon^-} \phi p_- ds - \int\int_{\Omega_\epsilon} \nabla \phi \nabla p dA
\]

\[
= \int_{\Gamma_\epsilon^+} \phi p_+ ds - \int_{\Gamma_\epsilon^-} \phi p_- ds - \int_{\Gamma_\epsilon^+} \phi p_+ ds + \int_{\Gamma_\epsilon^-} \phi p_- ds + \int\int_{\Omega_\epsilon} p \Delta \phi dA.
\]  

(A8)

The superscripts $+$ and $-$ indicate the values taken from the outside and inside of the interface $\Gamma_\epsilon$, respectively. Notice $\phi$ is twice continuously differentiable and $p$ is bounded and only discontinuous along the interface. So as $\epsilon$ approaches zero, we have

\[
\int\int_{\Omega_\epsilon} p \Delta \phi dA \to 0.
\]  

(A9)

and

\[
\int\int_{\Omega_\epsilon} (\Delta p)\phi dxdy \to \int_{\Gamma_\epsilon^+} \phi[p_+] ds - \int_{\Gamma_\epsilon^-} \phi[p_-] ds.
\]  

(A10)

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

\[
\phi_n = \nabla \phi \cdot \mathbf{n} = \frac{\partial \phi}{\partial x} \cos \theta + \frac{\partial \phi}{\partial y} \sin \theta
\]  

(A11)
\begin{equation}
\phi_s = \nabla \phi \cdot \boldsymbol{\tau} = \frac{\partial \phi}{\partial x} \sin \theta + \frac{\partial \phi}{\partial y} \cos \theta.
\end{equation}

where \( \theta \) 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

\begin{equation}
\frac{\partial \phi}{\partial x} = \phi_n \cos \theta - \phi_s \sin \theta
\end{equation}

\begin{equation}
\frac{\partial \phi}{\partial y} = \phi_n \sin \theta + \phi_s \cos \theta.
\end{equation}

Then

\begin{equation}
\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
\end{equation}

\begin{equation}
= \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{equation}

For any twice continuously differentiable test function \( \phi \),

\begin{equation}
\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] ds.
\end{equation}

We can conclude that

\[ [p] = f_1 \cos \theta + f_2 \sin \theta \]

and

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

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 Equation (5) by test function \( \phi \) and integrating.

\begin{equation}
\int \int_{\Omega} \phi \mu \Delta u - \phi \frac{dp}{dx} dA = - \int \int_{\Omega} \phi \int_{\Gamma_c} f_1(s) \delta(X(s) - x) ds dA
\end{equation}

\begin{equation}
\int \int_{\Omega} \phi \mu \Delta u dA - \int \int_{\Omega} \phi \frac{dp}{dx} dA = - \int_{\Gamma_c} \int \int_{\Omega} \phi f_1(s) \delta(X(s) - x) dA ds
\end{equation}

\begin{equation}
\int \int_{\Omega} \phi \mu \Delta u dA - \int \int_{\Omega} \phi \frac{dp}{dx} dA = - \int_{\Gamma_c} \phi f_1(s) ds
\end{equation}

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

\begin{equation}
\int \int_{\Omega} \phi \mu \Delta u dA = \int_{\Gamma_c^+} \phi u^+ \cdot \mathbf{n} ds + \int_{\Gamma_c^-} \phi \mu^- \nabla u^- \cdot (-\mathbf{n}) ds - \int \int_{\Omega} \mu(\nabla \phi \nabla u) dA
\end{equation}

\begin{equation}
\quad = \int_{\Gamma_c^+} \mu^+ \phi u^+_n ds - \int_{\Gamma_c^-} \mu^- \phi u^-_n ds - \int \int_{\Omega} \mu(\nabla \phi \nabla u) dA
\end{equation}

\begin{equation}
\quad \to \int_{\Gamma} \phi[\mu u_n] ds - 0 \text{ as } \epsilon \to 0.
\end{equation}
The right most integral in Equation (A23) approaches zeros because $\phi$ is continuously differentiable and $u_n$ is bounded and only discontinuous at the interface. Note that

$$\int \int_{\Omega} \phi \frac{dp}{dx} dA = \int \int_{\Omega} \phi \nabla \cdot \begin{bmatrix} p \\ 0 \end{bmatrix} dA$$  \hspace{1cm} (A25)

$$= \int_{\Gamma^+} \phi([p^+, 0]^T \cdot n) ds - \int_{\Gamma^-} \phi([p^-, 0]^T \cdot n) ds - \int \int_{\Omega} \nabla \phi \cdot \begin{bmatrix} p \\ 0 \end{bmatrix} dA$$  \hspace{1cm} (A26)

$$\rightarrow \int_{\Gamma_c} \phi[p] \cos \theta ds + 0 \text{ as } \epsilon \rightarrow 0.$$  \hspace{1cm} (A27)

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

$$[\mu u_n] = [p] \cos \theta - f_1$$  \hspace{1cm} (A28)

$$= \cos \theta (f_1 \cos \theta + f_2 \sin \theta) - f_1$$  \hspace{1cm} (A29)

$$= \sin \theta (-f_1 \sin \theta + f_2 \cos \theta).$$  \hspace{1cm} (A30)

Similarly, for $v$ we can get

$$[\mu v_n] = [p] \sin \theta - f_2$$  \hspace{1cm} (A31)

$$= \cos \theta (f_1 \sin \theta - f_2 \cos \theta).$$  \hspace{1cm} (A32)

This completes the derivation.