Solving the Immersed Interface Problem Using the Decomposition with Boundary Integral Approach

Anita Layton

June 28, 2010
Model Formulation

Navier-Stokes flows:
\[
\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla \mathbf{p} + \mu \nabla^2 \mathbf{u} + \mathbf{f},
\]
\[
\nabla \cdot \mathbf{u} = 0
\]

Elastic boundary force:
\[
f_i(x, t) = \int_0^L F_i(s, t) \delta(x - X) ds,
\]
\[
\mathbf{F}(s, t) = \frac{\partial}{\partial s} \left( T(s, t) \tau(s, t) \right),
\]
\[
T(s, t) = T_0 \left( \left| \frac{\partial \mathbf{X}}{\partial \alpha} \right| - 1 \right).
\]

Dirichlet BC

\[
\mathbf{u} \big|_{\partial \Omega} = \mathbf{u}_b
\]
Solving the Immersed Interface Problem Using the Decomposition with Boundary Integral Approach

Anita Layton

Introduction

Solution approaches

Velocity Decomposition

Numerical Results

Resolving Boundary Layers

Summary

Numerical Challenges

Singular boundary force introduces discontinuities in fluid solution.

Standard finite difference approximations have large errors.
**Immersed boundary method**: approximate $\delta$

**Immersed interface method**: (Mayo, *SINUM* 84; Li and Lai, *JCP*, 2001)

\[
\frac{u^* - u^n}{\Delta t} + (u \cdot \nabla_h u)^{n+\frac{1}{2}} = -\nabla_h p^{n-\frac{1}{2}} + \frac{\mu}{2} \left( \nabla_h^2 u^* + \nabla_h^2 u^n \right) + C_1
\]

\[
(u \cdot \nabla_h u)^{n+\frac{1}{2}} = \frac{3}{2} (u^n \cdot \nabla_h) u^n - \frac{1}{2} (u^{n-1} \cdot \nabla_h) u^{n-1} + C_2^n
\]

\[
\nabla_h^2 \phi^{n+1} \phi = \frac{\nabla_h \cdot u^*}{\Delta t} + C_3
\]

\[
u^{n+1} = u^* - \Delta t \nabla_h \phi^{n+1} + C_4
\]

\[
\nabla_h p^{n+\frac{1}{2}} = \nabla_h p^{n-\frac{1}{2}} + \nabla_h \phi^{n+1} + C_5
\]
The Velocity Decomposition Approach

Like immersed interface method, computes sharp, second-order approximations.

Need fewer corrections—or none at all!

- First-generation code:
  - Corrections required for a Stokes problem only;
  - Biperiodic boundary conditions.

- New and improved version:
  - No correction needed, uses boundary integrals instead;
  - Dirichlet boundary conditions.
If We Were Bacteria...

Then we would be in the zero-Reynolds number regime:

\[ \nabla p_s = \mu \nabla^2 u_s + f \]
\[ \nabla \cdot u_s = 0 \]

The jump conditions would be

\[
[p_s] = f \cdot n, \quad [p_{sn}] = \frac{\partial}{\partial s} (f \cdot \tau),
\]
\[
[u_s] = 0, \quad \mu \left[ \frac{\partial u_s}{\partial n} \right] = -(f \cdot \tau) \tau.
\]

Solve three Poisson problems (How? Later)

\[ \nabla^2 p_s = 0, \quad \mu \nabla^2 u_s = \nabla p_s \]
Key observation: the jump conditions in $p$ and $u_n$ are the same for Navier-Stokes and Stokes flows.

- Assume Dirichlet boundary conditions.
- Split solution into a Stokes part, a regular part, and a boundary correction part:

$$u = u_s + u_r + u_{bc}, \quad p = p_s + p_r + p_{bc}$$

- Stokes solution include singular $f$ (still easy to solve); remainder solution doesn’t.
- Thus, Stokes solution is singular, and the regular solution is regular (also “easy” to solve!).
- Regular and boundary correction parts: Later
Boundary Integral Solution

\[ \nabla p_s = \mu \nabla^2 u_s + f, \quad \nabla \cdot u_s = 0 \]

Stokes solutions are given by the boundary integrals

\[ p(x) = \int_{\Gamma} \nabla G(x - y)f(y)ds(y) \]

\[ u(x) = \int_{\Gamma} V(x - y)f(y)ds(y). \]

\[ \nabla G \] and \[ V \] are determined by the spatial dimensions and boundary conditions. For 2D free space,

\[ \nabla G(x) = \frac{x}{2\pi|x|^2} \]

\[ V(x) = \frac{1}{4\pi} \left[ \begin{array}{ccc} - \log|x| + \frac{x_1^2}{|x|^2} & \frac{x_1 x_2}{|x|^2} \\ \frac{x_1 x_2}{|x|^2} & - \log|x| + \frac{x_2^2}{|x|^2} \end{array} \right] \]
Using boundary integrals eliminate the need for corrections, but...

- **Accuracy.** The kernels $V$ and $\nabla G$ are singular!

\[
\nabla G(x) = \frac{x}{2\pi|x|^2}, \quad V(x) = \frac{1}{4\pi} \left[ -\log|x| + \frac{x_1^2}{|x|^2} + \frac{x_1 x_2}{|x|^2} \right]
\]

Near the immersed interface, nearly singular integrals give rise to large quadrature errors.

- **Efficiency.** Using boundary integrals to compute solution values at $N^2$ grid-points takes $O(N^3)$ time.
Accuracy: Modified Stokeslets

Replace point source by a “blob” (Cortez, *SISC*, 2001)

\[ \phi_\epsilon(r) = \frac{3\epsilon^3}{2\pi(r^2 + \epsilon^2)^{5/2}} \]

Then the Green’s function of \( \Delta G = \delta \) becomes regularized:

\[ G(r) = \frac{1}{2\pi} \log(r) \Rightarrow G_\epsilon(r) = \frac{1}{2\pi} \left( \log(\sqrt{r^2 + \epsilon^2 + \epsilon}) - \frac{\epsilon}{\sqrt{r^2 + \epsilon^2}} \right) \]

where \( r = |x| \). Stokes solutions are given by the boundary integrals, e.g.,

\[ p(x) = \int_{\Gamma} \nabla G_\epsilon(x - y) f(y) ds(y) \]
Accuracy: Modified Stokeslets

But Stokes solutions computed using the regularized Green’s function are smooth near the boundary, i.e., the jump discontinuities in $p$ and in the derivatives of $u$ are not preserved.

This leads to $O(h)$ errors in $u$ and $O(1)$ errors in $p$.

To achieve better accuracy, corrections can be added:

$$p(x) = \sum_{k} \nabla G_\epsilon(x - s_k)f(s_k)\Delta s + T_1(x) + T_2(x) + O(\Delta s^2 + \epsilon^2)$$

where $T_1$ and $T_2$ correct for the quadrature and regularization errors (Beale and Lai, *SINUM*, 2001).
Efficiency: Hybrid Approach

- Combine boundary integrals with mesh-based solver.
- Compute integral solutions only near boundary (cost: $O(N^2)$), enough values to form discrete five-point Laplacian at irregular grid points.
Hybrid Approach

Solve three Poisson problems:

\[
\Delta_h p_s = \begin{cases} 
\frac{p_{s,i+1,j} + p_{s,i,j+1} - 4p_{s,i,j} + p_{s,i-1,j} + p_{s,i,j-1}}{h^2}, & \text{irreg points} \\
0, & \text{reg points}
\end{cases}
\]

\(p_s\) values in RHS Laplacian computed using modified Stokeslets.

\[
\Delta_h u_s = \begin{cases} 
\frac{u_{s,i+1,j} + u_{s,i,j+1} - 4u_{s,i,j} + u_{s,i-1,j} + u_{s,i,j-1}}{h^2}, & \text{irreg points} \\
\nabla p_s, & \text{reg points}
\end{cases}
\]

\(u_s\) values in RHS Laplacian computed using modified Stokeslets. \(\nabla p_s\) computed using finite difference.

- Solve Poisson problems using FFT, \(O(N^2 \log N)\).
The Regular Solution

\[ u = u_s + u_r + u_{bc} \]

The Stokes solution satisfies

\[ \nabla p_s = \mu \nabla^2 u_s + f, \quad \nabla \cdot u_s = 0 \]

Substituting into the Navier-Stokes equations:

\[
\frac{\partial (u_s + u_r + u_{bc})}{\partial t} + u \cdot \nabla (u_s + u_r + u_{bc}) = -\nabla (p_s + p_r + p_{bc}) \\
+ \mu \nabla^2 (u_s + u_r + u_{bc}) + f,
\]

\[
\Rightarrow \frac{\partial u_r}{\partial t} + u \cdot \nabla u_r = -\nabla p_r + \mu \nabla^2 u_r + f_b,
\]

\[
\Rightarrow \frac{\partial u_{bc}}{\partial t} + u \cdot \nabla u_{bc} = -\nabla p_{bc} + \mu \nabla^2 u_{bc},
\]

\[ \nabla \cdot u_r = 0, \quad \nabla \cdot u_{bc} = 0, \]

\[ f_b = -\frac{\partial u_s}{\partial t} - u \cdot \nabla u_s \]
The Regular Solution

\[
\mathbf{u} = \mathbf{u}_s + \mathbf{u}_r + \mathbf{u}_{bc}
\]

The Stokes solution satisfies

\[
\nabla p_s = \mu \nabla^2 \mathbf{u}_s + \mathbf{f}, \quad \nabla \cdot \mathbf{u}_s = 0
\]

Substituting into the Navier-Stokes equations:

\[
\frac{\partial (\mathbf{u}_s + \mathbf{u}_r + \mathbf{u}_{bc})}{\partial t} + \mathbf{u} \cdot \nabla (\mathbf{u}_s + \mathbf{u}_r + \mathbf{u}_{bc}) = -\nabla (p_s + p_r + p_{bc}) + \mu \nabla^2 (\mathbf{u}_s + \mathbf{u}_r + \mathbf{u}_{bc}) + \mathbf{f},
\]

\[
\Rightarrow \frac{\partial \mathbf{u}_r}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}_r = -\nabla p_r + \mu \nabla^2 \mathbf{u}_r + \mathbf{f}_b,
\]

\[
\Rightarrow \frac{\partial \mathbf{u}_{bc}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}_{bc} = -\nabla p_{bc} + \mu \nabla^2 \mathbf{u}_{bc},
\]

\[
\nabla \cdot \mathbf{u}_r = 0, \quad \nabla \cdot \mathbf{u}_{bc} = 0,
\]

\[
\mathbf{f}_b = -\frac{\partial \mathbf{u}_s}{\partial t} - \mathbf{u} \cdot \nabla \mathbf{u}_s
\]
Semi-Lagrangian Discretization

**Key observation**: solution is smooth along fluid trajectories ⇒ semi-Lagrangian time discretization:

\[
\frac{du_r}{dt} = -\nabla p_r + \mu \nabla^2 u_r + f_b,
\]

where

\[
f_b = -\frac{du_s}{dt}
\]
Computing Regular Solution

Discretize total derivative using BDF:

\[ \frac{3u_r^{n+1} - 4\tilde{u}_r^n + \tilde{u}_r^{n-1}}{2\Delta t} + \nabla p_r^n = \mu \nabla^2 u_r^{n+1} + f_b^{n+1} \]

Then, in Fourier space, \( u_r^* \) is projected onto divergence-free space (orthogonal to \([\sin k_1x, \sin k_2y]\)).

\[ u_r^{n+1} = P u_r^* \]
\[ \nabla \phi = u_r^* - u_r^{n+1} \]

In grid space, the pressure is updated accordingly:

\[ \nabla p_r^{n+1} = \nabla p_r^n + \frac{3}{2} \nabla \phi - \mu \Delta t \nabla^3 \phi \]
The boundary correction solution satisfies

$$\frac{\partial \mathbf{u}_{bc}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}_{bc} = - \nabla p_{bc} + \mu \nabla^2 \mathbf{u}_{bc},$$

$$\nabla \cdot \mathbf{u}_{bc} = 0,$$

and the Dirichlet BCs

$$\mathbf{u}_{bc}|_{\partial \Omega} = \mathbf{u}_b - \mathbf{u}_s|_{\partial \Omega} - \mathbf{u}_r|_{\partial \Omega}$$

The boundary correction solution is smooth; use a second-order Euler time-stepping method.
The Algorithm

1. Use modified Stokeslet to compute $p_s$ and $u_s$ near the boundary.

2. Solve the modified Poisson problems for $p_s$ and $u_s$ everywhere.

3. Compute regular solution

$$\frac{du_r}{dt} = -\nabla p_r + \mu \nabla^2 u_r - \frac{du_s}{dt}, \quad \nabla \cdot u_r = 0$$

4. Compute boundary correction solution with

$$u_{bc}|_{\partial\Omega} = u_b - u_s|_{\partial\Omega} - u_r|_{\partial\Omega}$$

5. Combine to give overall solution:

$$u = u_s + u_r + u_{bc}, \quad p = p_s + p_r + p_{bc},$$
Solving the Immersed Interface Problem Using the Decomposition with Boundary Integral Approach

Anita Layton

Introduction

Solution approaches

Velocity Decomposition

Numerical Results

Resolving Boundary Layers

Summary

Velocity $\mathbf{v}$, overall solution; $\mathbf{v}_s$, Stokes part; $\mathbf{v}_r$, regular part. $\mathbf{v}$ and $\mathbf{v}_s$ has a discontinuous normal derivative across the boundary, whereas $\mathbf{v}_r$ does not.
Resolving Boundary Layers

- Stiff boundary forces may generate a steep gradient in the solutions near the boundary.
- Even though the solutions are “smooth” away from the boundary, finite-difference approximation of the Laplacian may have large discretization errors.
- Remedy: Expanding the “band” where we use boundary integrals to compute the discrete Laplacian.
Resolving Boundary Layers

\[
\Delta h p_s = \begin{cases} 
\frac{p_{si+1,j} + p_{si,j+1} - 4p_{si,j} + p_{si-1,j} + p_{si,j-1}}{h^2}, & \text{inside band} \\
0, & \text{outside band}
\end{cases}
\]

\[
\Delta h u_s = \begin{cases} 
\frac{u_{si+1,j} + u_{si,j+1} - 4u_{si,j} + u_{si-1,j} + u_{si,j-1}}{h^2}, & \text{inside band} \\
\nabla p_s, & \text{outside band}
\end{cases}
\]
Stokes Example

Interface is a unit circle.

\[ f(\theta) = 14 \sin(7\theta) x(\theta), \quad p(r, \theta) = \begin{cases} r^{-7} \sin(7\theta), & r \geq 1 \\ -r^7 \sin(7\theta), & r < 1 \end{cases} \]
Summary

- Advantages of velocity-decomposition approach:
  - Jump corrections not needed
  - Can resolve boundary layers

- Ongoing work: (with Tom Beale) A more semi-implicit method:
  - Implicit time-stepping for Stokes solution (stiff?)
  - Explicit time-stepping for regular and boundary correction solutions (less stiff?)