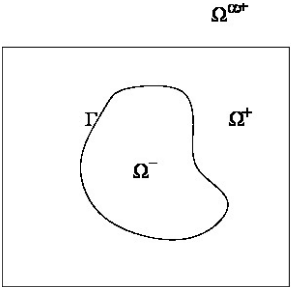
# Solving the Immersed Interface Problem Using the Decomposition with Boundary Integral Approach

Anita Layton

June 28, 2010

# Model Formulation




Dirihclet BC

$$\mathbf{u}\big|_{\partial\Omega} = \mathbf{u}_b$$

Navier-Stokes flows:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\nabla\mathbf{p} + \mu\nabla^2\mathbf{u} + \mathbf{f},$$

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

Elastic boundary force:

$$f_i(\mathbf{x}, t) = \int_0^L F_i(s, t)\delta(\mathbf{x} - \mathbf{X})ds,$$

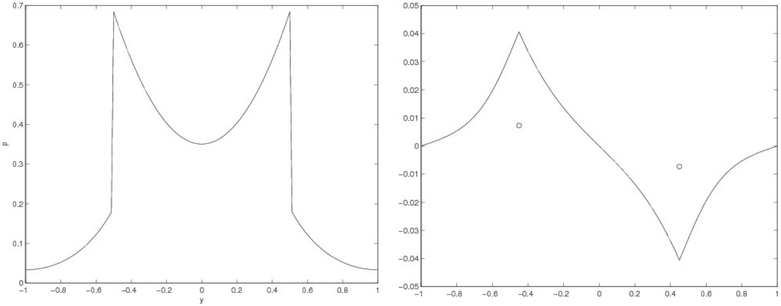
$$\mathbf{F}(s, t) = \frac{\partial}{\partial s}\left(T(s, t)\tau(s, t)\right),$$

$$T(s, t) = T_0\left(\left|\frac{\partial \mathbf{X}}{\partial \alpha}\right| - 1\right).$$

# Numerical Challenges



Singular boundary force introduces discontinuities in fluid solution.




Standard finite difference approximations have large errors.

# Solution Approaches



**Immersed boundary method**: approximate $\delta$
**Immersed interface method**: (Mayo, *SINUM 84*; Li and Lai, *JCP*, 2001)

$$
\begin{aligned}
\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + (\mathbf{u} \cdot \nabla_h \mathbf{u})^{n+\frac{1}{2}} &= -\nabla_h p^{n-\frac{1}{2}} + \frac{\mu}{2}\left(\nabla_h^2 \mathbf{u}^* + \nabla_h^2 \mathbf{u}^n\right) \\
&\quad + \mathbf{C}_1 \\
(\mathbf{u} \cdot \nabla_h \mathbf{u})^{n+\frac{1}{2}} &= \frac{3}{2}(\mathbf{u}^n \cdot \nabla_h)\mathbf{u}^n - \frac{1}{2}(\mathbf{u}^{n-1} \cdot \nabla_h)\mathbf{u}^{n-1} + \mathbf{C}_2^n \\
\nabla_h^2 \phi^{n+1} \phi &= \frac{\nabla_h \cdot \mathbf{u}^*}{\Delta t} + C_3 \\
\mathbf{u}^{n+1} &= \mathbf{u}^* - \Delta t \nabla_h \phi^{n+1} + \mathbf{C}_4 \\
\nabla_h p^{n+\frac{1}{2}} &= \nabla_h p^{n-\frac{1}{2}} + \nabla_h \phi^{n+1} + \mathbf{C}_5
\end{aligned}
$$

# The Velocity Decomposition Approache



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.
  - Beale and Layton, *JCP*, 228: 3358–3367, 2009.
- 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:

$$\nabla p_s = \mu \nabla^2 \mathbf{u}_s + \mathbf{f}$$
$$\nabla \cdot \mathbf{u}_s = 0$$

The jump conditions would be

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

Solve three Poisson problems (How? Later)

$$\nabla^2 p_s = 0, \quad \mu \nabla^2 \mathbf{u}_s = \nabla p_s$$

# Velocity Decomposition



**Key observation:** the jump conditions in $p$ and $\mathbf{u}_n$ 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:

$$\mathbf{u} = \mathbf{u}_s + \mathbf{u}_r + \mathbf{u}_{bc}, \quad p = p_s + p_r + p_{bc}$$

- Stokes solution include singular $\mathbf{f}$ (still easy to solve); remainder solution doesn't.
- Thus, Stokes solution is *singluar*, and the regular solution is *regular* (also "easy" to solve!).
- Regular and boundary correction parts: Later

# Boundary Integral Solution

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

Stokes solutions are given by the boundary integrals

$$p(\mathbf{x}) = \int_\Gamma \nabla G(\mathbf{x}-\mathbf{y})\mathbf{f}(\mathbf{y})ds(\mathbf{y})$$
$$\mathbf{u}(\mathbf{x}) = \int_\Gamma V(\mathbf{x}-\mathbf{y})\mathbf{f}(\mathbf{y})ds(\mathbf{y}).$$

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

$$\nabla G(\mathbf{x}) = \frac{\mathbf{x}}{2\pi|\mathbf{x}|^2}$$
$$V(\mathbf{x}) = \frac{1}{4\pi}\begin{bmatrix} -\log|\mathbf{x}| + \frac{x_1^2}{|\mathbf{x}|^2} & \frac{x_1 x_2}{|\mathbf{x}|^2} \\ \frac{x_1 x_2}{|\mathbf{x}|^2} & -\log|\mathbf{x}| + \frac{x_2^2}{|\mathbf{x}|^2} \end{bmatrix}$$

# Two Problems with Integral Solutions



Using boundary integrals eliminate the need for corrections, but...

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

$$\nabla G(\mathbf{x}) = \frac{\mathbf{x}}{2\pi|\mathbf{x}|^2}, \quad V(\mathbf{x}) = \frac{1}{4\pi}\begin{bmatrix} -\log|\mathbf{x}| + \frac{x_1^2}{|\mathbf{x}|^2} & \frac{x_1 x_2}{|\mathbf{x}|^2} \\ \frac{x_1 x_2}{|\mathbf{x}|^2} & -\log|\mathbf{x}| + \frac{x_2^2}{|\mathbf{x}|^2} \end{bmatrix}$$

  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 $\mathcal{O}(N^3)$ 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 $r = |\mathbf{x}|$. Stokes solutions are given by the boundary integrals, e.g.,

$$p(\mathbf{x}) = \int_\Gamma \nabla G_\epsilon(\mathbf{x}-\mathbf{y})\mathbf{f}(\mathbf{y})ds(\mathbf{y})$$

# 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 $\mathbf{u}$ are not preserved.

This leads to $\mathcal{O}(h)$ errors in $\mathbf{u}$ and $\mathcal{O}(1)$ errors in $p$.

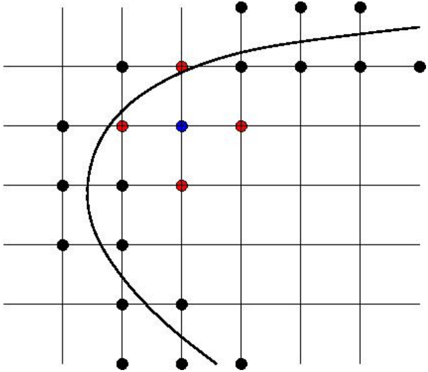
To achieve better accuracy, corrections can be added:

$$p(\mathbf{x}) = \sum_k \nabla G_\epsilon(\mathbf{x} - \mathbf{s}_k) f(\mathbf{s}_k) \Delta s + T_1(\mathbf{x}) + T_2(\mathbf{x}) + \mathcal{O}(\Delta s^2 + \epsilon^2)$$

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: $\mathcal{O}(N^2)$), enough values to form discrete five-point Laplacian at irregular grid points.



# Hybrid Approach



Solve three Poisson problems:

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

$p_s$ values in RHS Laplacian computed using modified Stokeslets.

$$\Delta_h u_s = \begin{cases} \frac{{\mathbf{u}_s}_{i+1,j} + {\mathbf{u}_s}_{i,j+1} - 4{\mathbf{u}_s}_{i,j} + {\mathbf{u}_s}_{i-1,j} + {\mathbf{u}_s}_{i,j-1}}{h^2}, & \text{irreg points} \\ \nabla p_s, & \text{reg points} \end{cases}$$

$\mathbf{u}_s$ values in RHS Laplacian computed using modified Stokeslets. $\nabla p_s$ computed using finite difference.

- Solve Poisson problems using FFT, $\mathcal{O}(N^2 \log N)$.

# 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:

$$\begin{aligned}
\frac{\partial(\mathbf{u}_s + \mathbf{u}_r + \mathbf{u}_{bc})}{\partial t} + \mathbf{u} \cdot \nabla (\mathbf{u}_s + \mathbf{u}_r + \mathbf{u}_{bc}) = -\nabla(p_s + p_r + p_{bc}) \\
+\mu \nabla^2 (\mathbf{u}_s + \mathbf{u}_r + \mathbf{u}_{bc}) + \mathbf{f}, \\
\Rightarrow \frac{\partial \mathbf{u}_r}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}_r = -\nabla p_r + \mu \nabla^2 \mathbf{u}_r + \mathbf{f_b}, \\
\Rightarrow \frac{\partial \mathbf{u}_{bc}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}_{bc} = -\nabla p_{bc} + \mu \nabla^2 \mathbf{u}_{bc}, \\
\nabla \cdot \mathbf{u}_r = 0, \quad \nabla \cdot \mathbf{u}_{bc} = 0, \\
\mathbf{f}_b = -\frac{\partial \mathbf{u}_s}{\partial t} - \mathbf{u} \cdot \nabla \mathbf{u}_s
\end{aligned}$$

# 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:

$$\frac{\partial(\mathbf{u}_s + \mathbf{u}_r + \mathbf{u}_{bc})}{\partial t} + \mathbf{u} \cdot \nabla (\mathbf{u}_s + \mathbf{u}_r + \mathbf{u}_{bc}) = -\nabla(p_s + p_r + p_{bc})$$

$$+\mu \nabla^2 (\mathbf{u}_s + \mathbf{u}_r + \mathbf{u}_{bc}) + \mathbf{f},$$

$$\Rightarrow \frac{\partial \mathbf{u}_r}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u}_r = -\nabla p_r + \mu \nabla^2 \mathbf{u}_r + \mathbf{f_b},$$

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

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

$$\mathbf{f}_b = -\frac{\partial \mathbf{u}_s}{\partial t} - \mathbf{u} \cdot \nabla \mathbf{u}_s$$

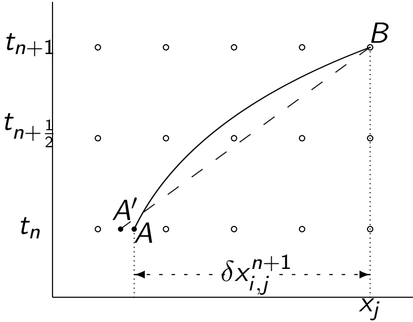
# Semi-Lagrangian Discretization

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

$$\frac{d\mathbf{u}_r}{dt} = -\nabla p_r + \mu \nabla^2 \mathbf{u}_r + \mathbf{f}_\mathbf{b},$$

where

$$\mathbf{f}_b = -\frac{d\mathbf{u}_s}{dt}$$



# Computing Regular Solution



Discretize total derivative using BDF:

$$\frac{3\mathbf{u}_r^{n+1} - 4\tilde{\mathbf{u}}_r^n + \tilde{\mathbf{u}}_r^{n-1}}{2\Delta t} + \nabla p_r^n = \mu \nabla^2 \mathbf{u}_r^{n+1} + \mathbf{f}_b^{n+1}$$

Then, in Fourier space, $\mathbf{u}_r^*$ is projected onto divergence-free space (orthogonal to $[\sin k_1 x, \sin k_2 y]$).

$$\mathbf{u}_r^{n+1} = \mathbb{P}\mathbf{u}_r^*$$
$$\nabla\phi = \mathbf{u}_r^* - \mathbf{u}_r^{n+1}$$

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



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 $\mathbf{u}_s$ near the boundary.
2. Solve the modified Poisson problems for $p_s$ and $\mathbf{u}_s$ everywhere.
3. Compute regular solution

$$\frac{d\mathbf{u}_r}{dt} = -\nabla p_r + \mu \nabla^2 \mathbf{u}_r - \frac{d\mathbf{u}_s}{dt}, \quad \nabla \cdot \mathbf{u}_r = 0$$

4. Compute boundary correction solution with

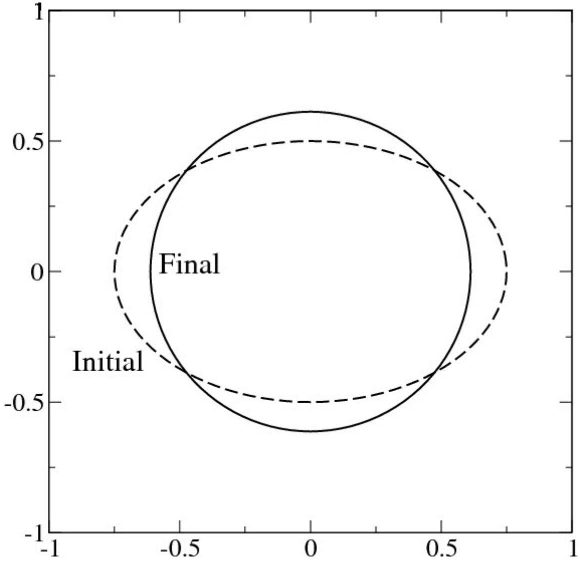
$$\mathbf{u}_{bc}|_{\partial\Omega} = \mathbf{u}_b - \mathbf{u}_s|_{\partial\Omega} - \mathbf{u}_r|_{\partial\Omega}$$

5. Combine to give overall solution:

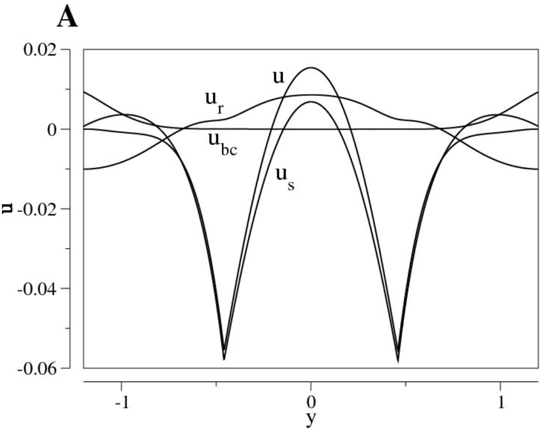
$$\mathbf{u} = \mathbf{u}_s + \mathbf{u}_r + \mathbf{u}_{bc}, \quad p = p_s + p_r + p_{bc},$$

# Example 1





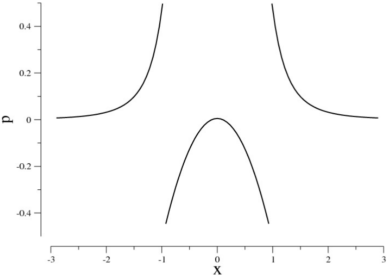
# Velocity




$v$, overall solution; $v_s$, Stokes part; $v_r$, regular part. $v$ and $v_s$ has a discontinuous normal derivative across the boundary, whereas $v_r$ does not.

# Resolving Boundary Layers

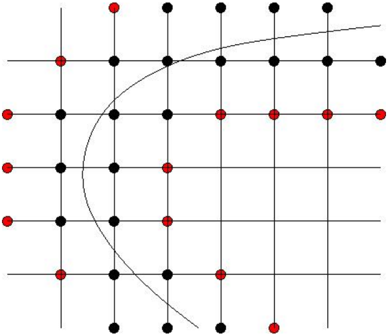
- Stiff boundary forces may generate a steep gradient in the solutions near the boundary.
- Even though the solutions are "smooth" away from the boundary, finite-difference approximation of the Laplacian may have large discretization errors.
- Remedy: Expanding the "band" where we use boundary integrals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



# Resolving Boundary Layers

$$\Delta_h p_s = \begin{cases} \frac{p_{s\,i+1,j} + p_{s\,i,j+1} - 4p_{s\,i,j} + p_{s\,i-1,j} + p_{s\,i,j-1}}{h^2}, & \text{inside band} \\ 0, & \text{outside band} \end{cases}$$

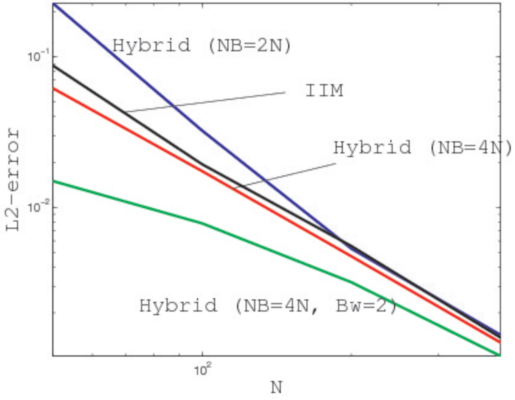
$$\Delta_h u_s = \begin{cases} \frac{\mathbf{u}_{s\,i+1,j} + \mathbf{u}_{s\,i,j+1} - 4\mathbf{u}_{s\,i,j} + \mathbf{u}_{s\,i-1,j} + \mathbf{u}_{s\,i,j-1}}{h^2}, & \text{inside band} \\ \nabla p_s, & \text{outside band} \end{cases}$$



# Stokes Example

Interface is a unit circle.

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



# 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?)