Immersed Interface Methods for Fluid Dynamics Problems

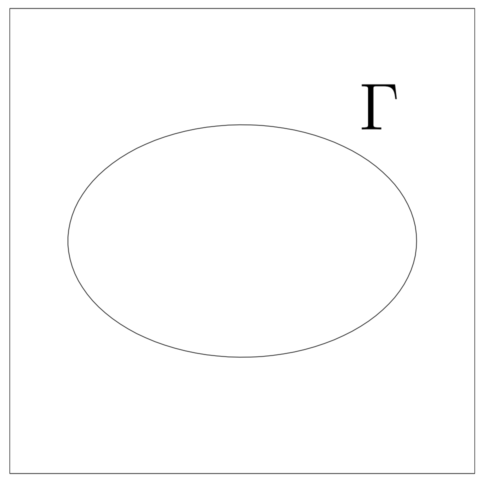
Randall J. LeVeque
Department of Applied Mathematics
University of Washington

Joint work with
Zhilin Li, NCSU
Long Lee, UNC

Supported in part by NSF, DOE

Incompressible Navier-Stokes

$$
\begin{aligned}
u_t + (u \cdot \nabla)u + \nabla p &= \mu \nabla^2 u + f \\
\nabla \cdot u &= 0
\end{aligned}
$$



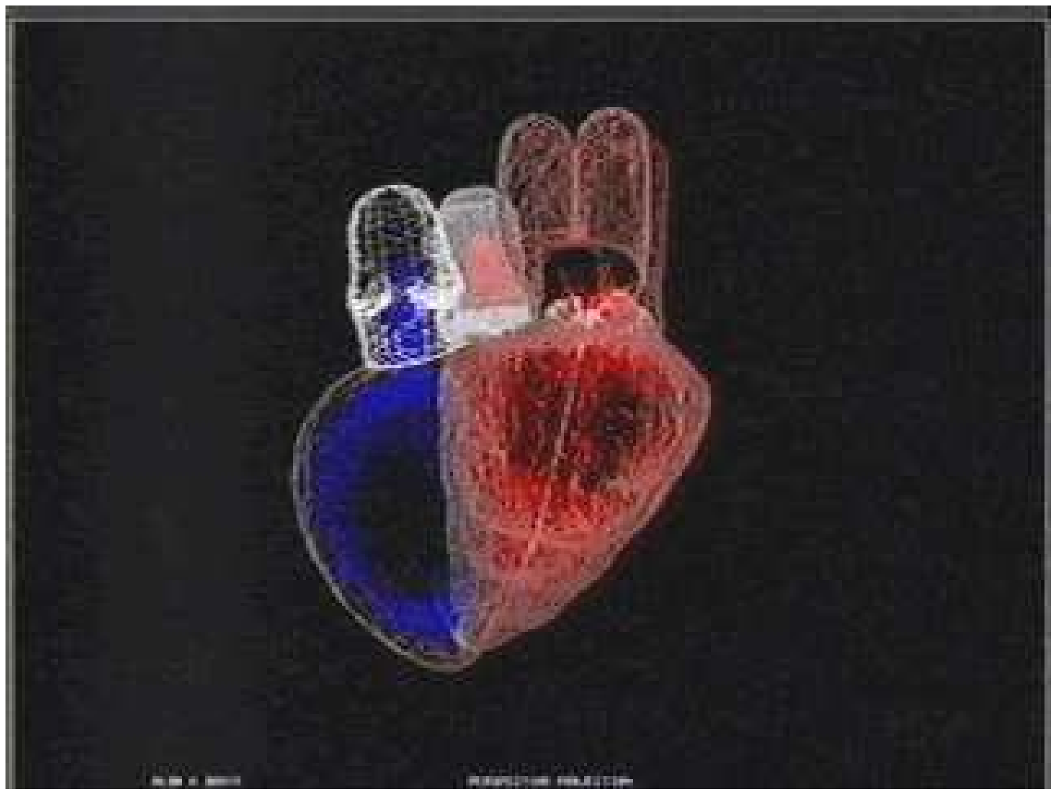
An immersed elastic membrane Γ exerts a singular force on the fluid,

$$
f(x, y) = \int_\Gamma F(s)\, \delta(x - X(s))\, \delta(y - Y(s))\, ds,
$$

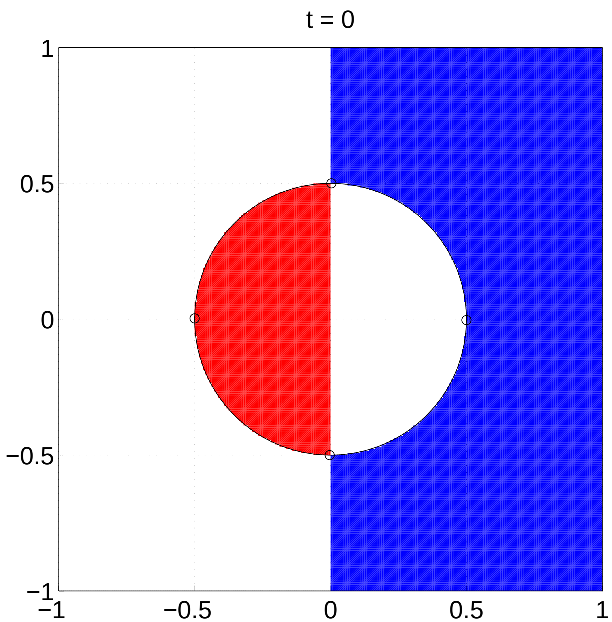
and moves with the fluid.

Peskin's Heart Model

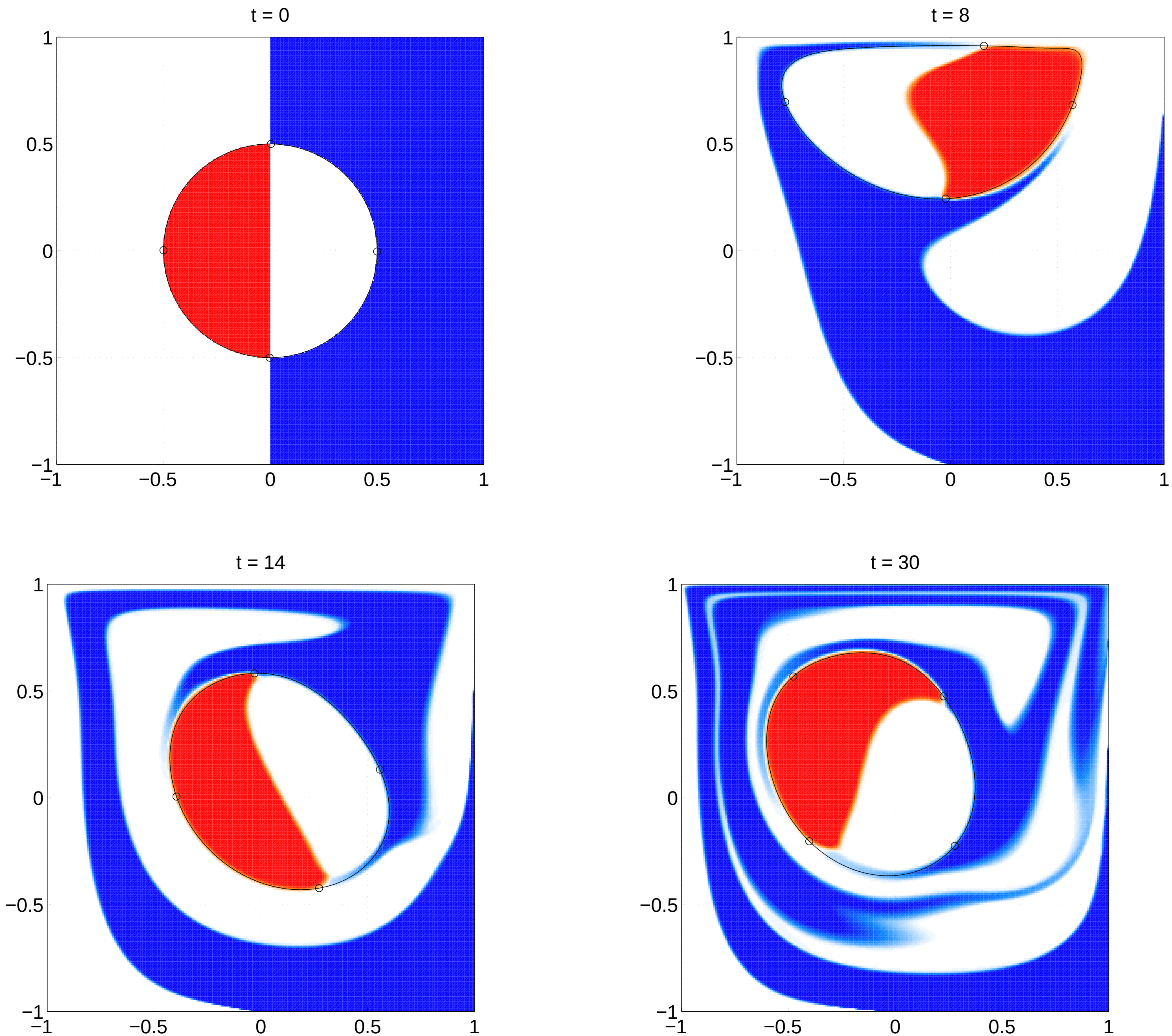
Originally developed to model blood flow in a beating heart and the operation of artificial heart valves.



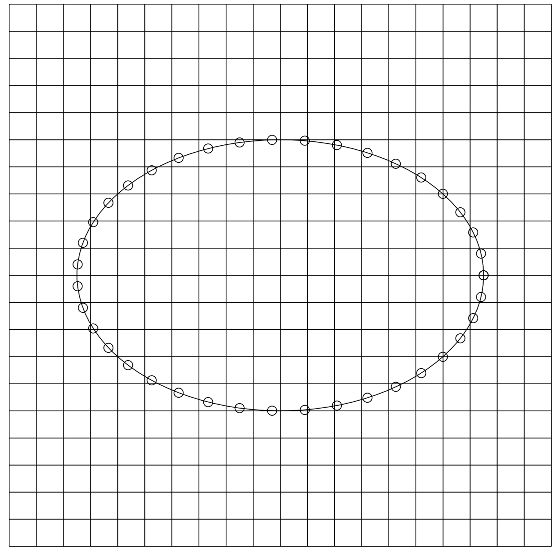
Balloon in a driven cavity



Balloon in a driven cavity



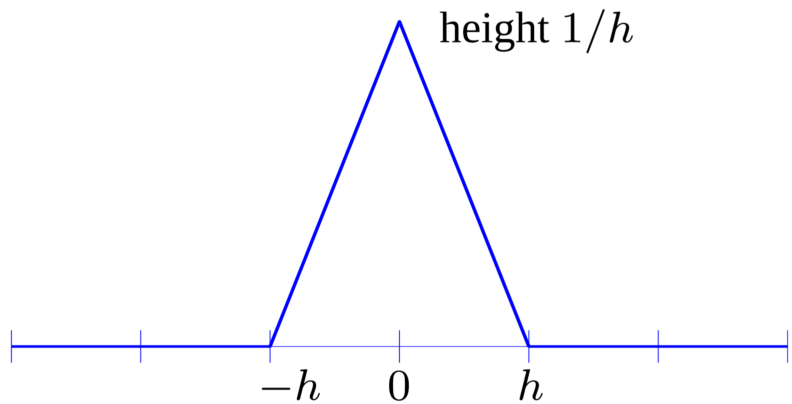
Peskin's Immersed Boundary Method



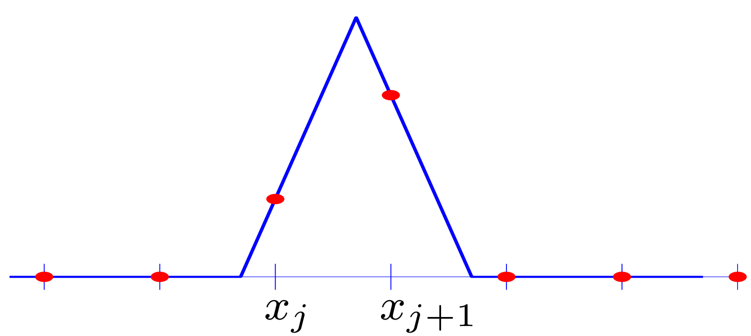
- Membrane represented by discrete control points X_k^n.
- Calculate force strength F_k^n at each control point.
- Use discrete delta function to spread forces to nearby Cartesian grid points, yielding nonzero f_{ij} at points near the interface.
- Advance the fluid equations on the uniform grid.
- Interpolate resulting velocity field u_{ij}^{n+1} to control points to obtain U_k^{n+1}.
- Move control points by $X_k^{n+1} = X_k^n + \Delta t\, U_k^{n+1}$.
- Implicit or semi-implicit approach may be needed for stability.

Discrete delta function in 1D

Example: Hat function



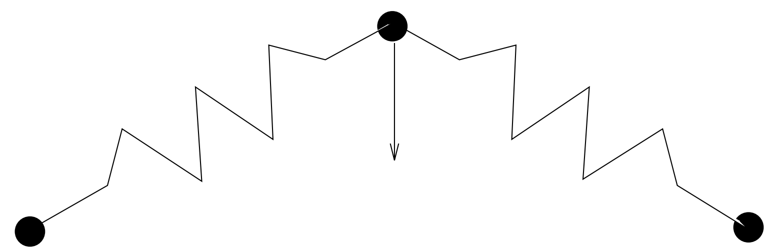
Singular force $F\delta(x-\alpha) \approx F d_h(x_i - \alpha)$ on the grid.
This is nonzero at only two points $(x_j < \alpha < j_{j+1})$:



Spring model of forces

The force $\vec{F}_k$ at $\vec{X}_k$ is computed based on the shape of the boundary.

Example: Spring model

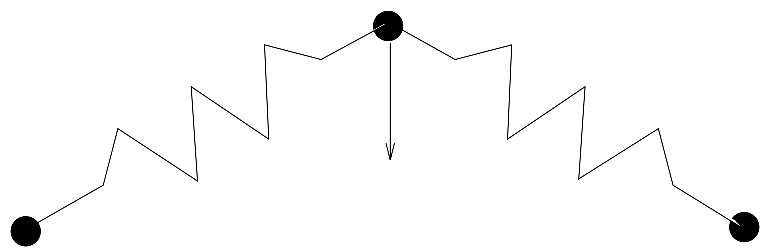


$$\vec{F}_k = \sigma_{k+1/2}(\vec{X}_{k+1} - \vec{X}_k) - \sigma_{k-1/2}(\vec{X}_k - \vec{X}_{k-1}).$$

Spring model of forces

The force $\vec{F}_k$ at $\vec{X}_k$ is computed based on the shape of the boundary.

Example: Spring model



$$\vec{F}_k = \sigma_{k+1/2}(\vec{X}_{k+1} - \vec{X}_k) - \sigma_{k-1/2}(\vec{X}_k - \vec{X}_{k-1}).$$

For $\vec{X}(s)$ parameterized by unstretched length,

$$f(s,t) = \frac{\partial}{\partial s}(T(s,t)\tau(s,t)),$$

where

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

Jump conditions

With mass density $m(s)$:

$$m(s)X_{tt}(s,t) = f(s,t) - [\![p]\!]\vec{n} + \mu\left[\frac{\partial u}{\partial n}\right].$$

Massless membrane: $m(s) = 0$

$$\begin{aligned} f &= \text{elastic force (computed from } X^n\text{)} \\ &= f_n\vec{n} + f_\tau\tau \\ [\![p]\!] &= f_n \\ \mu\left[\frac{\partial u}{\partial n}\right] &= -f_\tau\tau \end{aligned}$$

Projection Method (one form)

$$\begin{aligned} u_t + (u \cdot \nabla)u + \nabla p &= \mu \nabla^2 u + f \\ \nabla \cdot u &= 0 \end{aligned}$$

$\boxed{1}$ $U^n \longrightarrow U^*$ by solving

$$u_t + (u \cdot \nabla)u = \mu \nabla^2 u + f$$

$\boxed{2}$ $U^* \longrightarrow U^{n+1}$ by solving

$$u_t + \nabla p = 0$$

and requiring $\nabla \cdot U^{n+1} = 0$:

$$\frac{U^{n+1} - U^*}{\Delta t} + \nabla p = 0$$

$$\Longrightarrow \quad \Delta t \, \nabla^2 p = \nabla \cdot U^*$$

$\boxed{1}\ U^n \rightarrow U^*$ by solving $u_t + (u \cdot \nabla)u = \mu \nabla^2 u + f$

$\boxed{2}\ U^* \rightarrow U^{n+1}$ by solving $\Delta t\, \nabla^2 p = \nabla \cdot U^*$

True solution:

- p should be discontinuous across Γ
- u should be continuous but not smooth

Numerical solution:

- Singular source in $\boxed{1}$ leads to "delta function" in U^*
- $\nabla \cdot U^*$ gives "dipole source" for $\nabla^2 p$
- Results in "discontinuity" in p.

Immersed Interface Approach

$$
\begin{aligned}
u_t + (u \cdot \nabla) u + \nabla p &= \mu \nabla^2 u + f \\
\nabla \cdot u &= 0
\end{aligned}
$$

$\boxed{1}$ $U^n \longrightarrow U^*$ by solving

$$u_t + (u \cdot \nabla) u = \mu \nabla^2 u + f_\tau$$

$\boxed{2}$ $U^* \longrightarrow U^{n+1}$ by solving

$$u_t + \nabla p = f_n$$

and requiring $\nabla \cdot U^{n+1} = 0$:

$$\frac{U^{n+1} - U^*}{\Delta t} + \nabla p = f_n$$

$$\Longrightarrow \quad \Delta t \, \nabla^2 p = \nabla \cdot U^* + \Delta t \, \nabla \cdot f_n$$

$\boxed{1}$ $U^n \longrightarrow U^*$ by solving $u_t + (u \cdot \nabla)u = \mu \nabla^2 u + f_\tau$

$\boxed{2}$ $U^* \longrightarrow U^{n+1}$ by solving $\Delta t\, \nabla^2 p = \nabla \cdot U^* + \Delta t\, \nabla \cdot f_n$

Numerical solution:

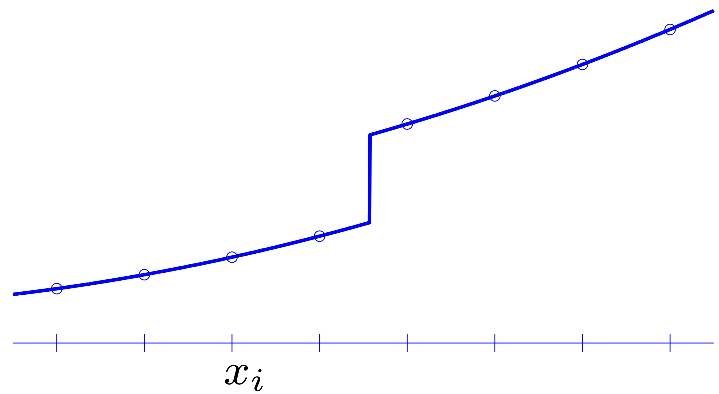
- Singular source in $\boxed{1}$ is tangential to interface, so u remains bounded
- $\nabla \cdot f_n$ gives correct dipole source for $\nabla^2 p$
- Use jump conditions on p and $\partial p / \partial n$ while solving

$$\Delta t\, \nabla^2 p = \nabla \cdot U^*$$

using an immersed interface method.

Simple 1D example: $p_{xx} = f(x)$ with boundary conditions and jump condition $[\![p]\!] = c$ at $x = \alpha$.

Or: $p_{xx} = f(x) + c\delta'(x - \alpha)$.



Want to set $p_{xx}(x_i) = f(x_i)$.

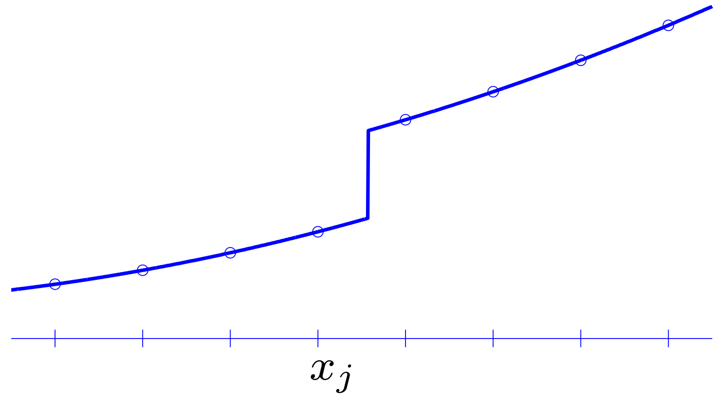
$$p(x_{i-1}) = p(x_i) - hp_x(x_i) + \frac{1}{2}h^2 p_{xx}(x_i) - \cdots$$
$$p(x_{i+1}) = p(x_i) + hp_x(x_i) + \frac{1}{2}h^2 p_{xx}(x_i) + \cdots$$

$$\Longrightarrow \qquad p_{xx}(x_i) \approx \frac{p(x_{i-1}) - 2p(x_i) + p(x_{i+1})}{h^2}$$

So that

$$\frac{p_{i-1} - 2p_i + p_{i+1}}{h^2} = f_i$$

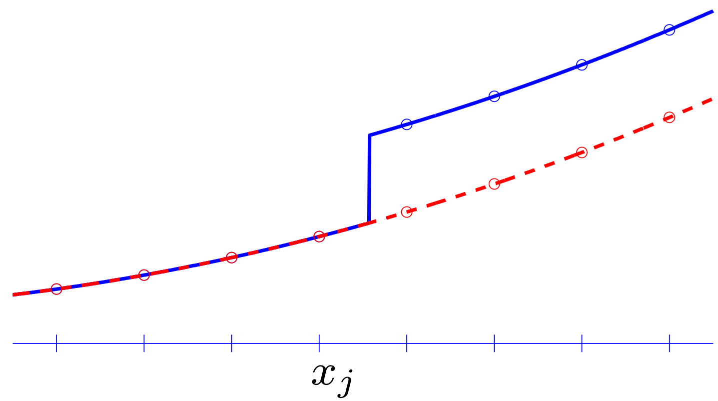
$$p_{xx} = f(x) + c\delta'(x - \alpha).$$



Want to set $p_{xx}(x_j) = f(x_j)$.

$$p(x_{j-1}) = p(x_j) - hp_x(x_j) + \frac{1}{2}h^2 p_{xx}(x_j) - \cdots$$

$$p_{xx} = f(x) + c\delta'(x-\alpha).$$



Want to set $p_{xx}(x_j) = f(x_j)$.

$$p(x_{j-1}) = p(x_j) - hp_x(x_j) + \tfrac{1}{2}h^2 p_{xx}(x_j) - \cdots$$
$$p(x_{j+1}) = \big(p(x_j) + hp_x(x_j) + \tfrac{1}{2}h^2 p_{xx}(x_j) + \cdots\big) + c$$

$$\Longrightarrow \qquad p_{xx}(x_j) \approx \frac{p(x_{j-1}) - 2p(x_j) + p(x_{j+1})}{h^2} - \frac{c}{h^2}$$

So that

$$\frac{p_{j-1} - 2p_j + p_{j+1}}{h^2} = f_j + \frac{c}{h^2}$$

and similarly

$$\frac{p_j - 2p_{j+1} + p_{j+2}}{h^2} = f_{j+1} - \frac{c}{h^2}$$

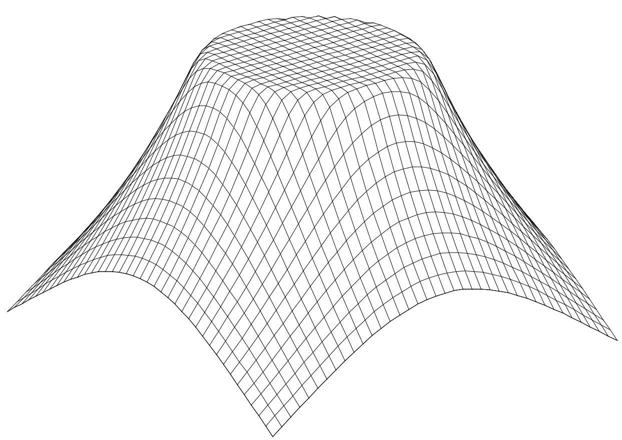
Example: $u_{xx} + u_{yy} = \int_\Gamma \delta(x - X(s))\, \delta(y - Y(s))\, ds$

We use the Dirichlet boundary condition which is determined from the exact solution

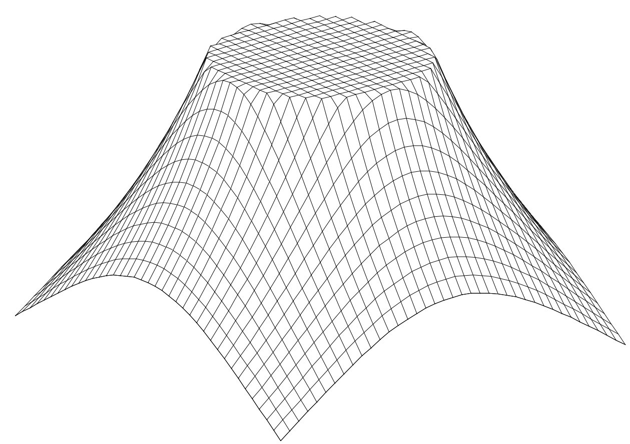
$$u(x,y) = \begin{cases} 1 & \text{if } r \le 0.5 \\ 1 + \log(2r) & \text{if } r > 0.5 \end{cases}$$

where $r = \sqrt{x^2 + y^2}$.

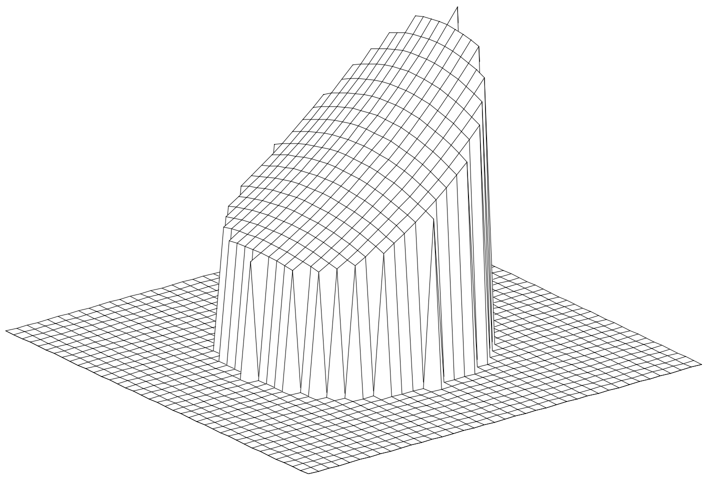
Using $d_h(x)$:

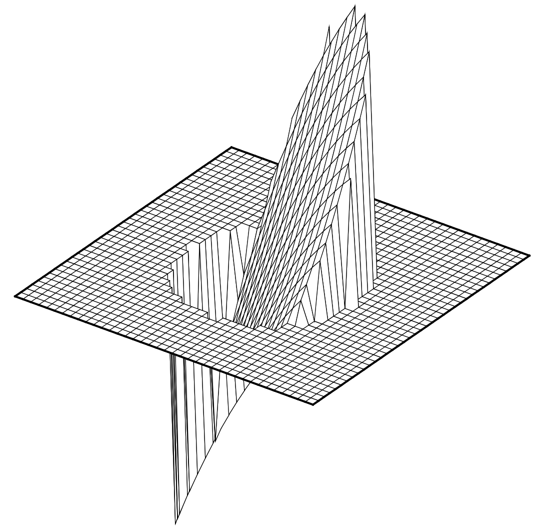


Using jump conditions:



Example with discontinuity in solution

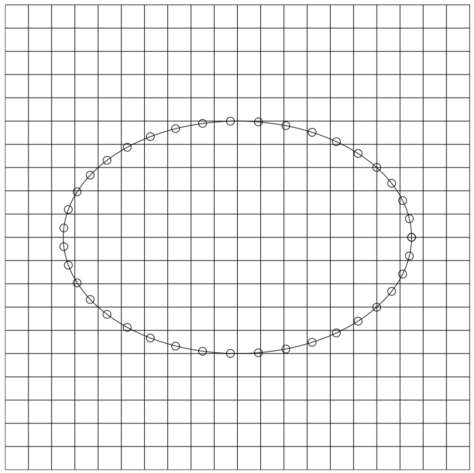
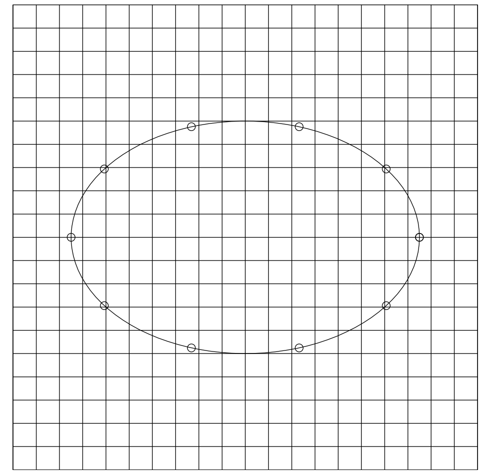




n	global error	ratio	local error	ratio
20	4.37883×10^{-4}		2.99215×10^{-2}	
40	1.07887×10^{-4}	4.0587	1.52546×10^{-2}	1.9615
80	2.77752×10^{-5}	3.8843	7.70114×10^{-3}	1.9808
160	7.49907×10^{-6}	3.7038	3.87481×10^{-3}	1.9875
320	1.74001×10^{-6}	4.3098	1.93917×10^{-3}	1.9982

Immersed Interface Method

- Represent interface by spline through smaller number of control points:

- Can compute force and hence pressure jump at any point on membrane.

$$f(s,t) = \frac{\partial}{\partial s}(T(s,t)\tau(s,t)),$$

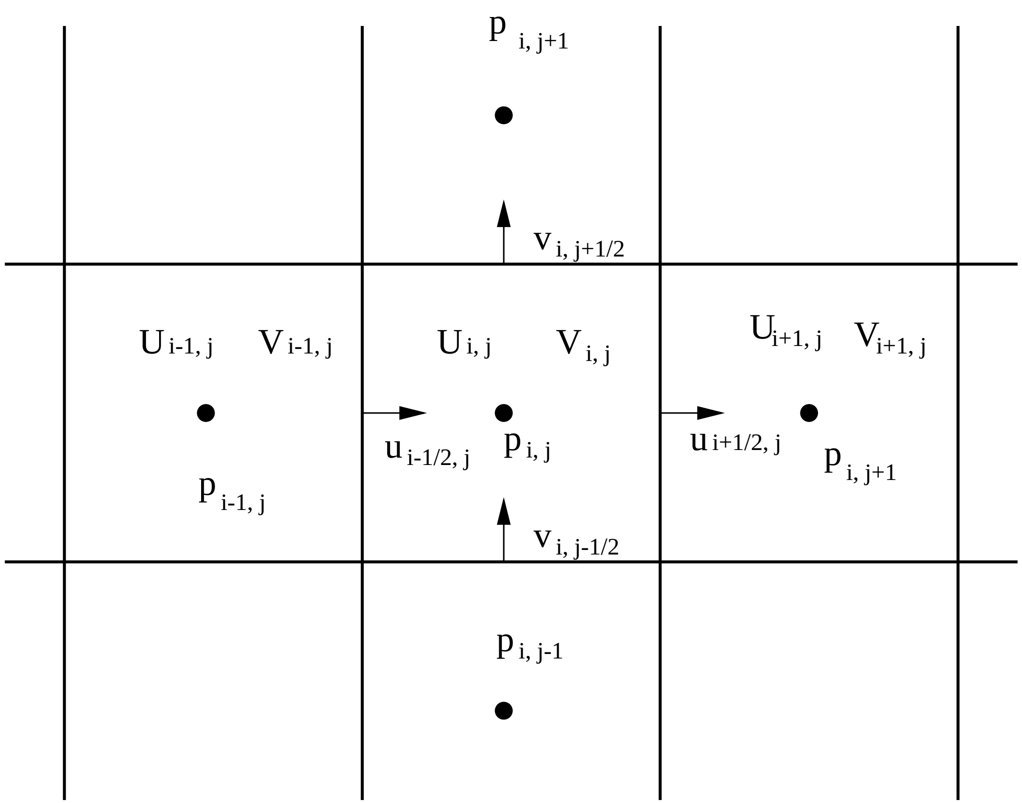
where

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

- Incorporate jump conditions into Taylor series expansion to derive finite-difference method that is pointwise second-order accurate.

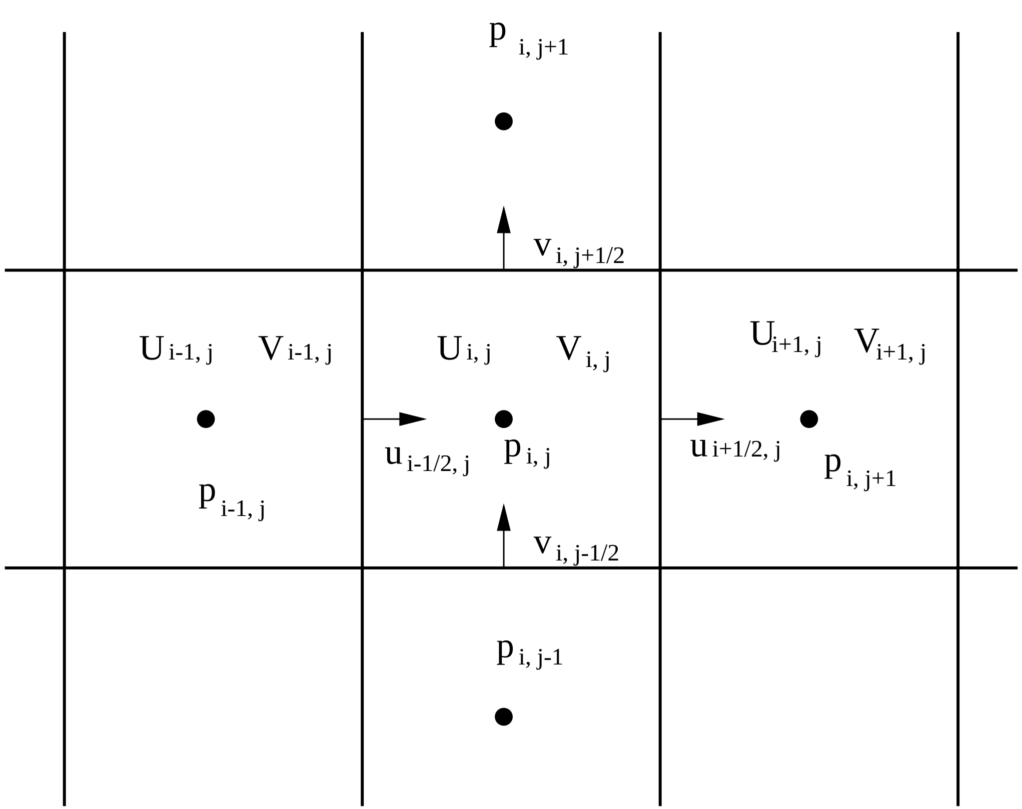
Hybrid Finite-Volume / Finite-Difference Method

- Fluid equations solved on finite-volume grid.
- Cell-centered velocity advected using high-resolution methods (CLAWPACK)
- Edge velocities needed in advection algorithm are obtained by averaging cell-centered values.



Hybrid Finite-Volume / Finite-Difference Method

- Divergence-free condition applied to the edge velocities.
- Finite-difference immersed interface method used to compute pressure at cell centers.
- Pressure correction is then applied to cell-centered velocities.



Explicit Method

Given X_k^n, U^n, u^n at start of time step.

Step 1. Solve $U_t + (u^n \cdot \nabla)U = 0$. Takes $U^n \; \rightarrow \; U^\dagger$.

Step 2. Solve $U_t = \mu \nabla^2 U + f_\tau$. Takes $U^\dagger \; \rightarrow \; U^*$.

Step 3. Average U^* from adjacent grid cells to obtain u^* at edges.

Step 4. Solve $\Delta t \, \nabla^2 p^{n+1} = \nabla \cdot u^*$ with $[\![p^{n+1}]\!] = f_n$ to obtain p^{n+1}.

Step 5. Update u^* based on p^{n+1} to get u^{n+1} (div free).

Step 6. Update U^* based on p^{n+1} to get U^{n+1}.

Step 7. Interpolate U^{n+1} to marker points X^n to obtain $\mathcal{U}^{n+1}(X^n)$. Move membrane using

$$X^{n+1} = X^n + \Delta t \, \mathcal{U}^{n+1}(X^n).$$

Trapezoidal Method

$\mathcal{U}^n(X^n)$ = velocities at marker points X^n, interpolated from U^n,

$\mathcal{U}^{n+1}(X^{n+1})$ = velocities at marker points X^{n+1}, interpolated from

U^{n+1}, determined by stepping forward from U^n with forces

$\frac{1}{2}(f(X^n) + f(X^{n+1}))$.

Would like:

$$X^{n+1} = X^n + \frac{1}{2}\Delta t\,(\mathcal{U}^n(X^n) + \mathcal{U}^{n+1}(X^{n+1})).$$

Problem: Implicit in X^{n+1} and computing $\mathcal{U}^{n+1}(X^{n+1}))$ requires solving fluid equations and Poisson problem.

Implicit Method

Given X_k^n, U^n, u^n at start of time step.
Apply Quasi-Newton method to solve for X^{n+1}.

Step I1. Apply Step 1, the advection step. $U^n \;\rightarrow\; U^\dagger$.

Step I2. Make a guess $X^{[0]}$ for X^{n+1} and set $I = 0$.

Step I3. Perform Steps 2–6 but replacing $f(X^n)$ by $\frac{1}{2}(f(X^n) + f(X^{[I]}))$
This gives provisional velocity field U^{n+1}.

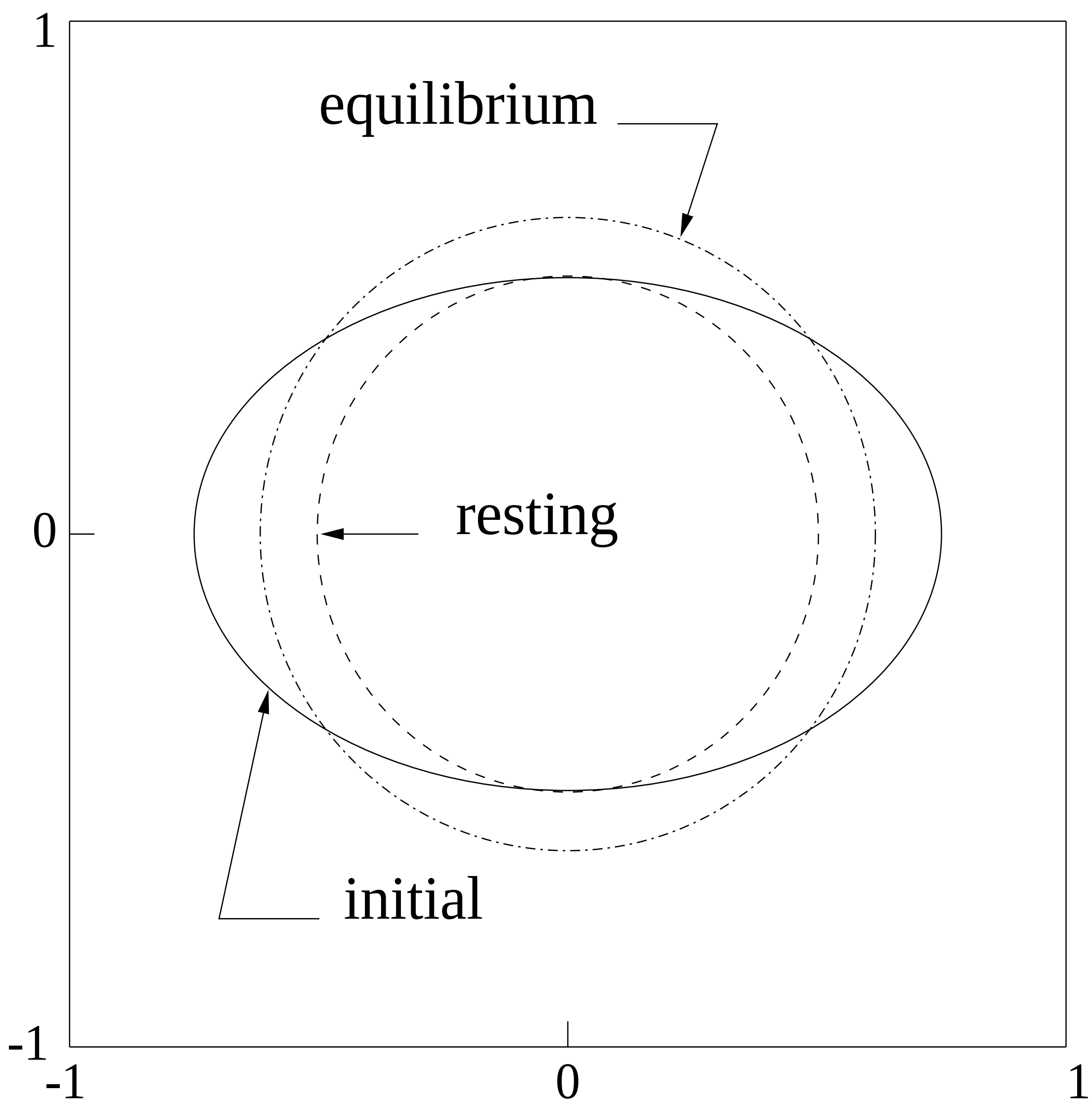
Step I4. Evaluate

$$g(X^{[I]}) = X^{[I]} - X^n - \frac{1}{2}\Delta t\,(\mathcal{U}^n(X^n) + \mathcal{U}^{n+1}(X^{[I]}))$$

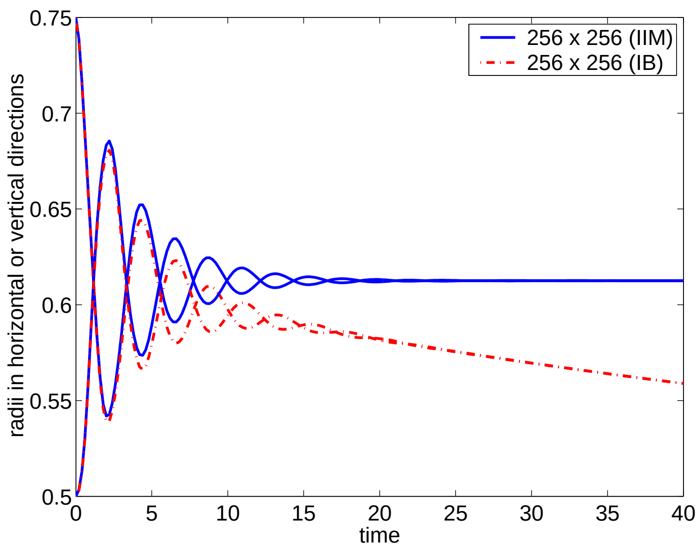
Step I5. Convergence check: If $\|g(X^{[I]})\| \leq \epsilon$ then set $X^{n+1} = X^{[I]}$, done.

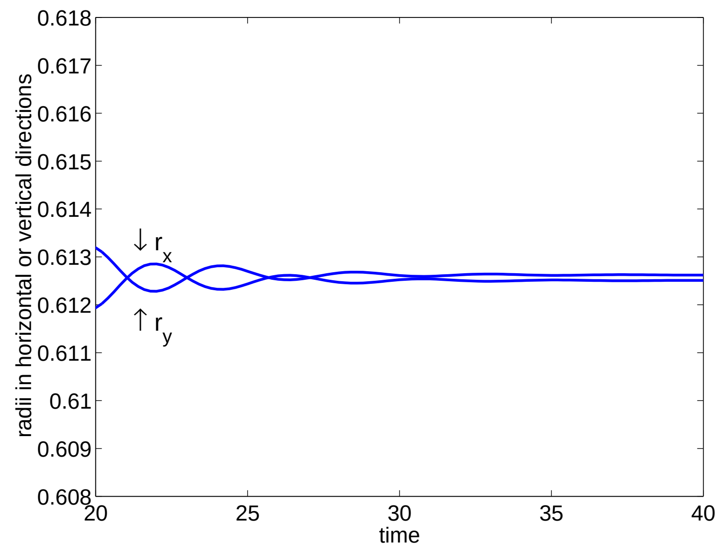
Otherwise, update $X^{[I]}$ to $X^{[I+1]}$, set $I = I + 1$, and go to step I3.

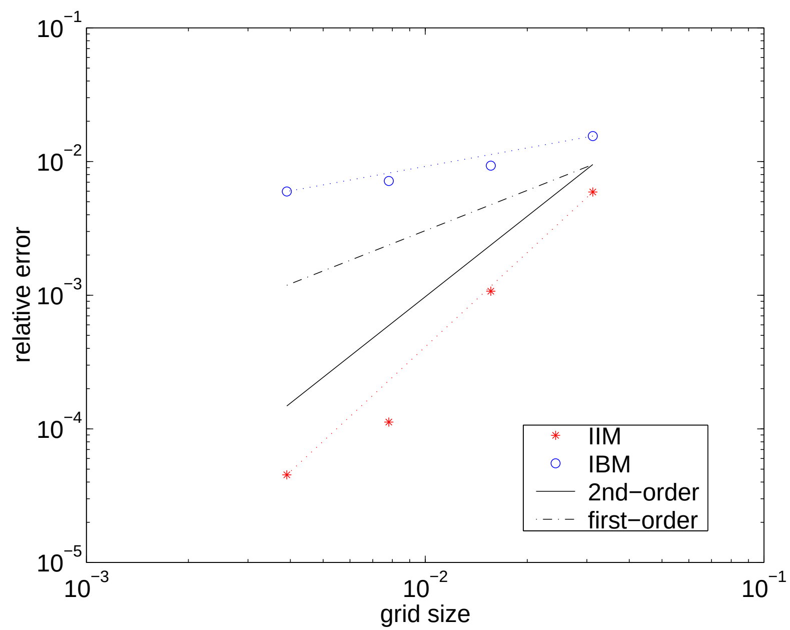
Oscillating balloon



Volume Conservation and Accuracy

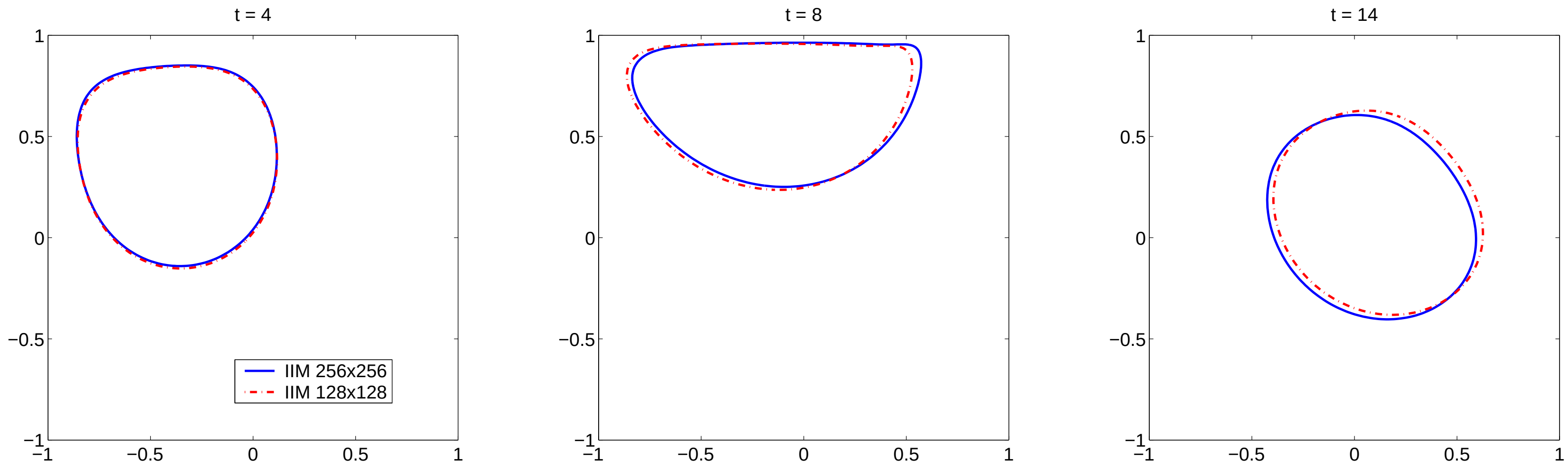




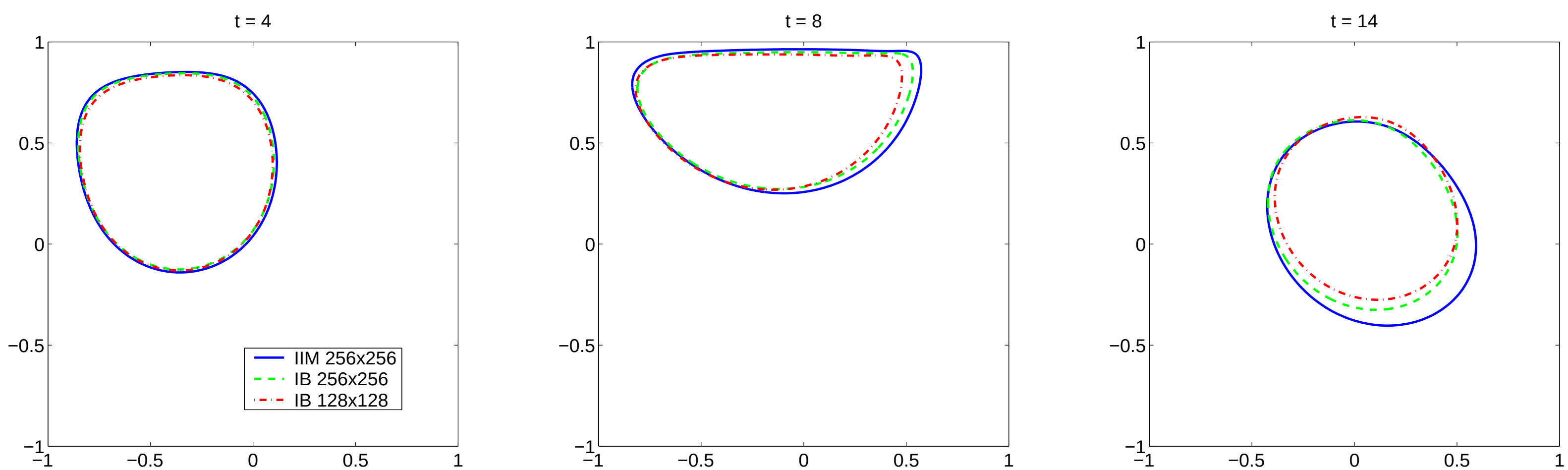


Balloon in driven cavity

IIM:



IBM:



Summary

- High-resolution finite volume for convective terms
- Finite difference immersed interface method for pressure Poisson problem
- Tangential component of force is spread using discrete delta functions
- Normal component of force gives jump in pressure, built into Poisson solver
- Easy modification of existing immersed boundary codes?
- Implicit method for moving the boundary
- Membrane mass can also be incorporated

Adding mass to the membrane

$$m(s)X_{tt}(s,t) = f(s,t) - [\![p]\!]\vec{n} + \mu\left[\frac{\partial u}{\partial n}\right].$$

Force now used in pressure and fluid solve:

$$\tilde{f}(s,t) = f(s,t) - m(s)X_{tt}(s,t).$$

Split this into normal and tangental components.

Easy to incorporate into implicit algorithm:

Step I3. Replace $\frac{1}{2}(f(X^n) + f(X^{[I]}))$ by

$$\frac{1}{2}(f(X^n) + f(X^{[I]})) - \frac{m(s)}{\Delta t^2}(X^{[I]} - 2X^n + X^{n-1}).$$