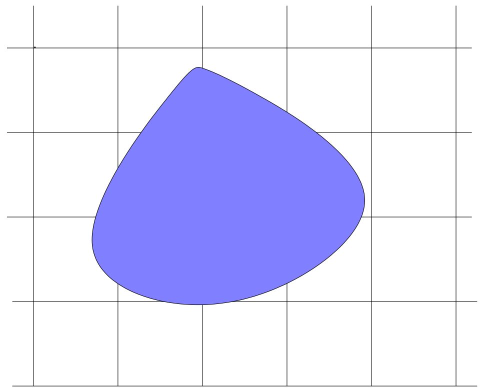
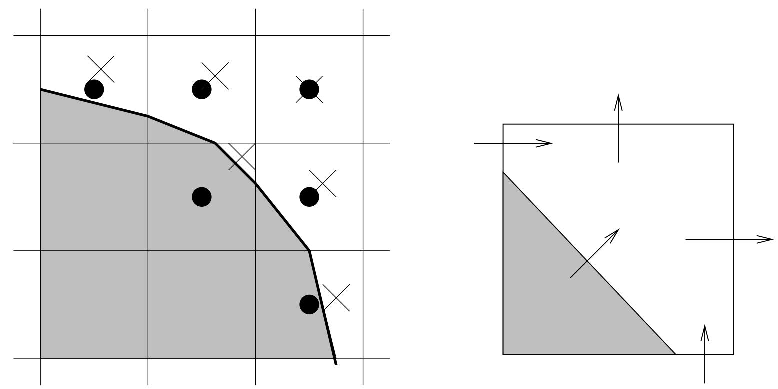
Volume-of-Fluid Discretization Methods for PDE in Irregular Domains

Phillip Colella
Computing Sciences Directorate
Lawrence Berkeley National Laboratory
Berkeley, CA

Cartesian Grid Representation of Irregular Boundaries

Based on nodal-point representation (Shortley and Weller, 1938) or finite-volume representation (Noh, 1964).





Advantages of underlying rectangular grid:

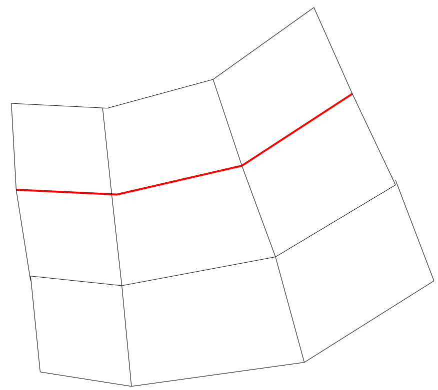
• Grid generation is tractable (Aftosmis, Berger, and Melton, 1998).

• Good discretization technology, e.g. well-understood consistency theory for finite differences, geometric multigrid for solvers.

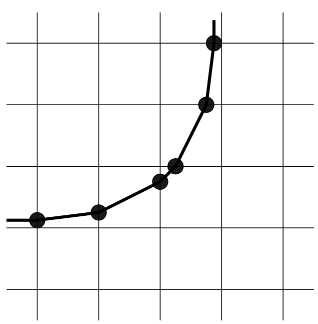
• Straightforward coupling to structured AMR (Chern and Colella, 1987; Young et. al., 1990; Berger and Leveque, 1991).

Lagrangian vs. Eulerian Representations of Free Surfaces

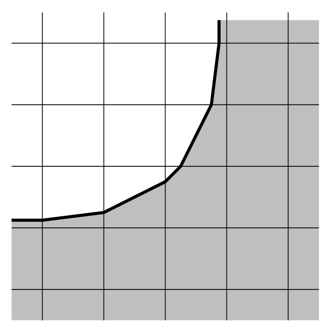
Lagrangian:



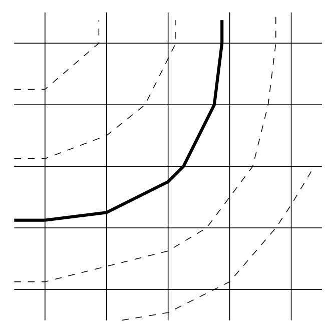
Eulerian:



Polygonal
(LANL, 1950s)



Volume of fluid
(LANL, LLNL, 1960s)

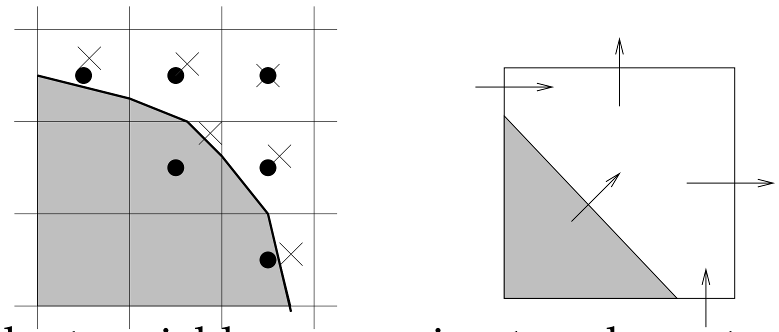


Level Set
(Osher & Sethian, 1988)

Finite-Volume Discretization - Fixed Boundaries

Consider PDEs written in conservation form:

$$\nabla \cdot (\nabla \phi) = \rho \qquad \frac{\partial U}{\partial t} + \nabla \cdot \vec{F}(U) = 0$$



• Primary dependent variables approximate values at centers of Cartesian cells. Extension of smooth functions to covered region exists, and extension operator is a bounded operator on the relevant function spaces.

• Divergence theorem over each control volume leads to "finite volume" approximation for $\nabla \cdot \vec{F}$:

$$\nabla \cdot \vec{F} \approx \frac{1}{\kappa h^d} \int \nabla \cdot \vec{F} dx = \frac{1}{\kappa h} \sum \alpha_s \vec{F}_s \cdot \vec{n}_s + \alpha_B \vec{F} \cdot \vec{n}_B \equiv D \cdot \vec{F}$$

• Away from the boundaries, method reduces to standard conservative finite difference discretization.

• If $\vec{F}_s \cdot \vec{n}_s$ approximates the value at the centroid to $O(h^2)$, then the truncation error $\tau = D \cdot \vec{F} - \nabla \cdot \vec{F}$ is given by

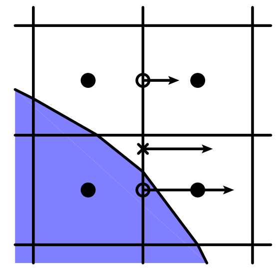
$$\begin{aligned}\tau &= O(h^2) \text{ at interior cells (if approximation is smooth).}\\ &= O(\frac{h}{\kappa}) \text{ at irregular control volumes.}\end{aligned}$$

Poisson's Equation

$$\Delta\phi = \rho \Rightarrow L^h \phi^h = \rho^h$$

$$L^h = D\vec{F}, \quad \vec{F} \approx \nabla\phi$$

$\vec{F}$ computed using linear interpolation of centered difference approximations to derivatives of ϕ.



$$L^h(\phi^h)_{\boldsymbol{i}} = \frac{1}{\kappa_{\boldsymbol{i}}} \sum_{\boldsymbol{s} \in \mathcal{S}_{\boldsymbol{i}}} a_{\boldsymbol{s}} \phi^h_{\boldsymbol{s}}$$

$$a_{\boldsymbol{s}} = O(\frac{1}{h^2}) \text{ uniformly w.r.t. } \kappa$$

The small denominator can be eliminated by diagonal scaling, eliminating the obvious potential conditioning problem: we solve

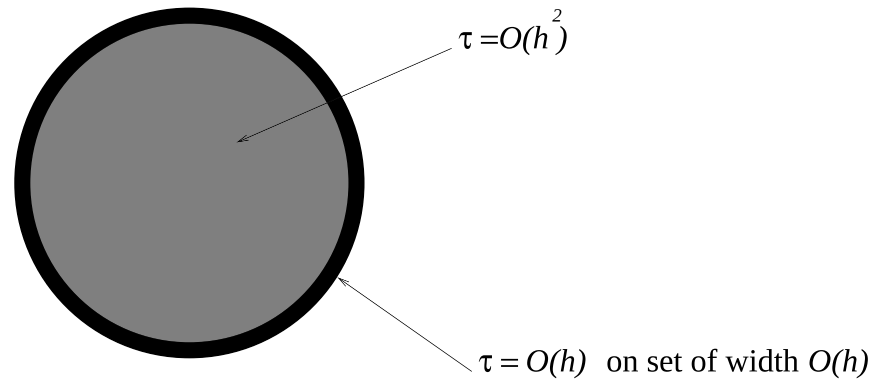
$$\kappa_{\boldsymbol{i}} L^h(\phi^h)_{\boldsymbol{i}} = \kappa_{\boldsymbol{i}} \rho^h_{\boldsymbol{i}}$$

Modified Equation Analysis

Error equation: $\phi^{exact,h} = \phi^h + (L^h)^{-1}(\tau)$

Modified equation: $\epsilon = (L^h)^{-1}(\tau) \approx \Delta^{-1}\tilde{\tau}$

where $\tilde{\tau}$ is some extension of τ, e.g. piecewise constant on each control volume.

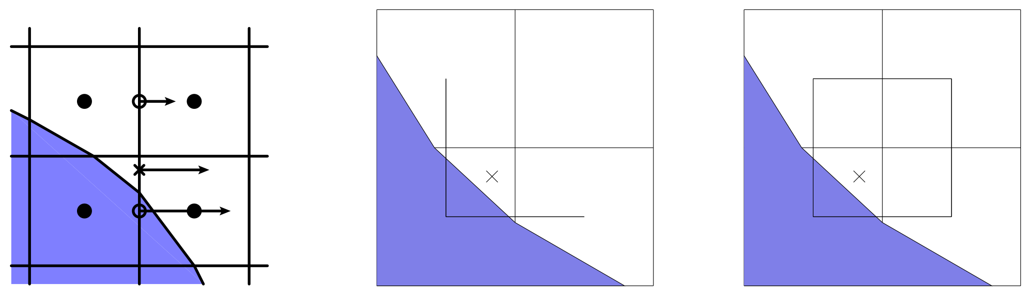


Smoothing of truncation error leads to a solution error that is $O(h^2)$ in max norm.

Extension to Three Dimensions

Our matrices aren't symmetric, nor are they M-matrices.

There are two obvious ways to extend the $O(h^2)$ flux calculation in 2D to 3D:

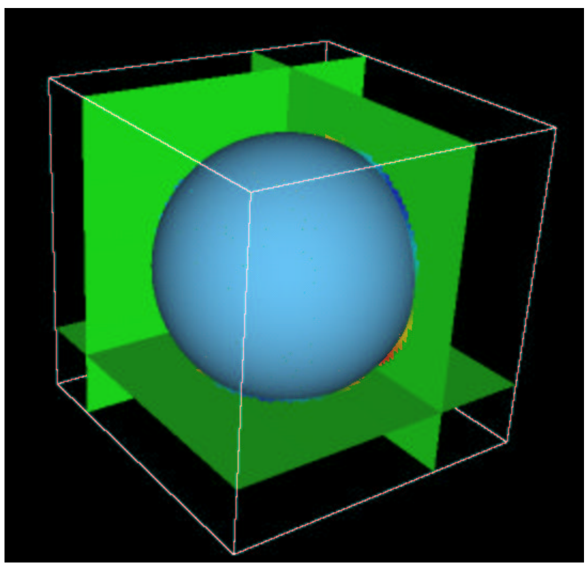


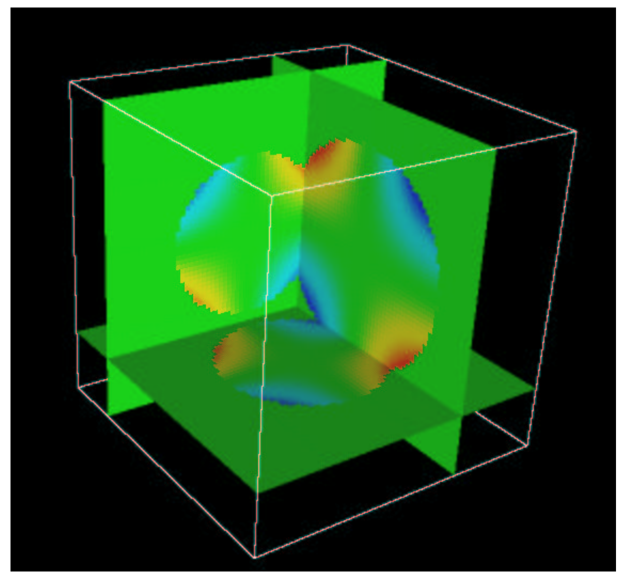
For intermittent configurations of adjacent small control volumes, linear interpolation is unstable (point Jacobi diverges), while bilinear interpolation appears to always be stable. Also, the inconsistent method coming from piecewise-constant interpolation is stable.

Unstable cases correspond to problems where small subproblems have eigenvalues of the wrong sign: the spectrum of PL^hP^t has elements in the right half-plane, where P is the projection onto a small set (2-8) of contiguous irregular control volumes.

Solution Error for Poisson's Equation in 3D

grid	$\|\epsilon\|_\infty$	p_∞	$\|\epsilon\|_2$	p_2	$\|\epsilon\|_1$	p_1
16^3	4.80×10^{-4}	—	5.17×10^{-5}	—	1.83×10^{-5}	—
32^3	1.06×10^{-4}	2.17	1.25×10^{-5}	2.05	4.41×10^{-6}	2.05
64^3	2.43×10^{-5}	2.13	3.07×10^{-6}	2.02	1.09×10^{-6}	2.02





Discretization of Hyperbolic Problems

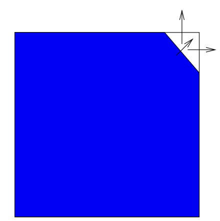
$$U^{n+1,h} = U^{n,h} - \Delta t D \cdot \vec{F}^{n+\frac{1}{2}}$$

Truncation error on irregular cells:

$$\tau \equiv \frac{U^{n+1,exact} - U^{n,exact}}{\Delta t} + D \cdot \vec{F}(U^{exact}) = O(h) + O(\Delta t^2) + O(\frac{h}{\kappa})$$

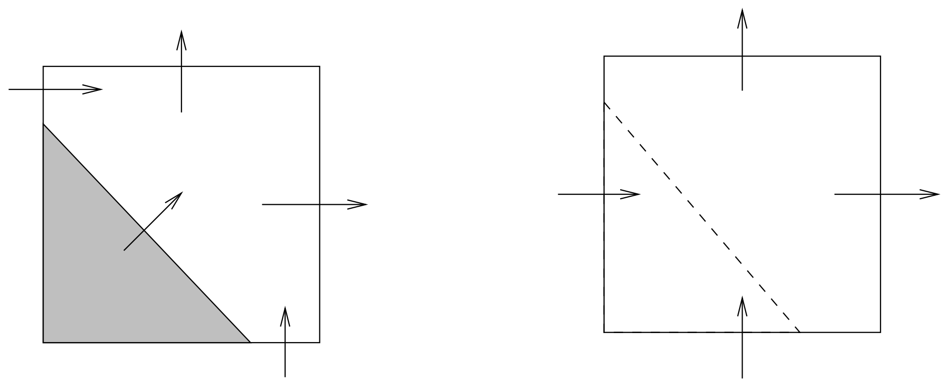
Want to use a time step given by the CFL for cells without the boundary.

$$\begin{aligned} U^{n+1} &= U^n - \Delta t D \cdot \vec{F} \\ &= U^n - \frac{\Delta t}{{\color{red}\kappa} h}\Big(\sum_{s \in faces} \alpha_s \vec{F}_s \cdot \vec{n}_s + \alpha_B \vec{F} \cdot \vec{n}_B\Big) \end{aligned}$$



Flux Difference Redistribution

In irregular cells, we hybridize the conservative update $(D \cdot \vec{F})$ with a nonconservative, but stable scheme $(\nabla \cdot \vec{F})^{NC}$, and redistribute the nonconservative increment to nearby cells.



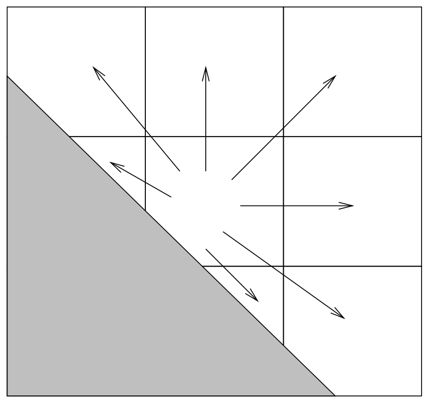
$$U^{n+1} = U^n - \Delta t(D \cdot \vec{F})^{NC} - w\Delta t((D \cdot \vec{F}) - (D \cdot \vec{F})^{NC})$$

The weight w is chosen so that, as $\kappa \to 1$, $w \to 1$, and $w = O(\kappa)$.

The amount of mass lost from each cell is

$$\delta M = -(1-w)\kappa((\nabla \cdot \vec{F})^{C} - (\nabla \cdot \vec{F})^{NC}) = O(h)$$

We redistribute that mass to nearby cells in a volume-weighted way.

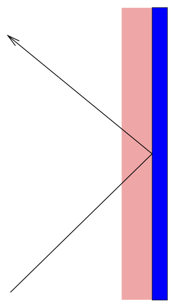


The truncation error for this method is $\tau = O(h)$ in cells sufficiently close to irregular cells, $\tau = O(h^2)$ otherwise.

Modified Equation Analysis

$$U^h = U + \epsilon \approx U^{mod}$$

$$\frac{\partial U^{mod}}{\partial t} + \nabla \cdot \vec{F}(U^{mod}) = \tilde{\tau}$$



If the boundary is noncharacteristic, the large forcing on the boundary can only act for a short time: $\frac{dU}{dt} = \tilde{\tau}$, but the characteristic path is in the region where $\tilde{\tau} = O(h)$ for only a time $O(h/\lambda)$, where λ is the characteristic speed. In that case,

$$U^h = U + O(h^2)$$

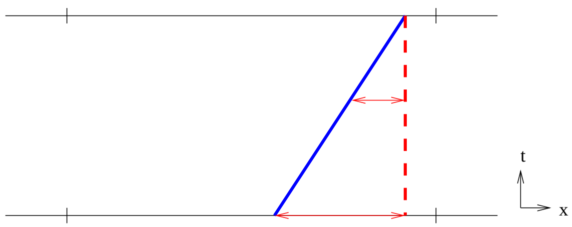
uniformly in x. If the boundary is characteristic, then we observe

$$U^h = U + O(h) \text{ in } L^\infty \text{ norm}$$

$$U^h = U + O(h^2) \text{ in } L^1 \text{ norm}$$

Diffusion in a Time-Dependent Domain

$$\frac{\partial T}{\partial t} = \Delta T + f \text{ on } \Omega(t)$$
$$\frac{\partial T}{\partial n} = \dot{m} + sT \text{ on } \partial\Omega(t)$$



In order to use a second-order accurate implicit time discretization, it is necessary to convert the moving boundary problem into a sequence of fixed boundary problems.

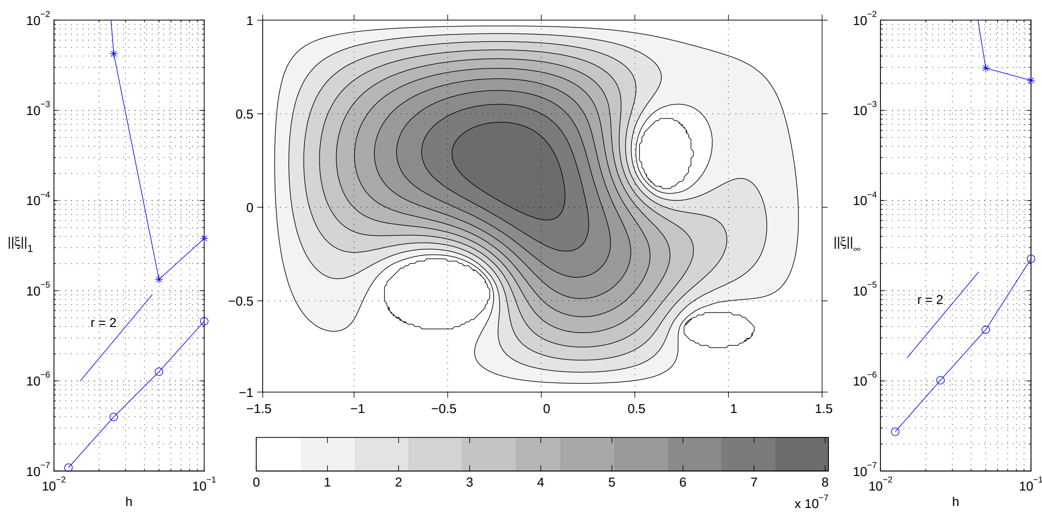
• Move the boundary, updating cells that are uncovered by appropriate extrapolation.

• Solve the heat equation on a fixed domain for one time step, using extrapolated boundary conditions.

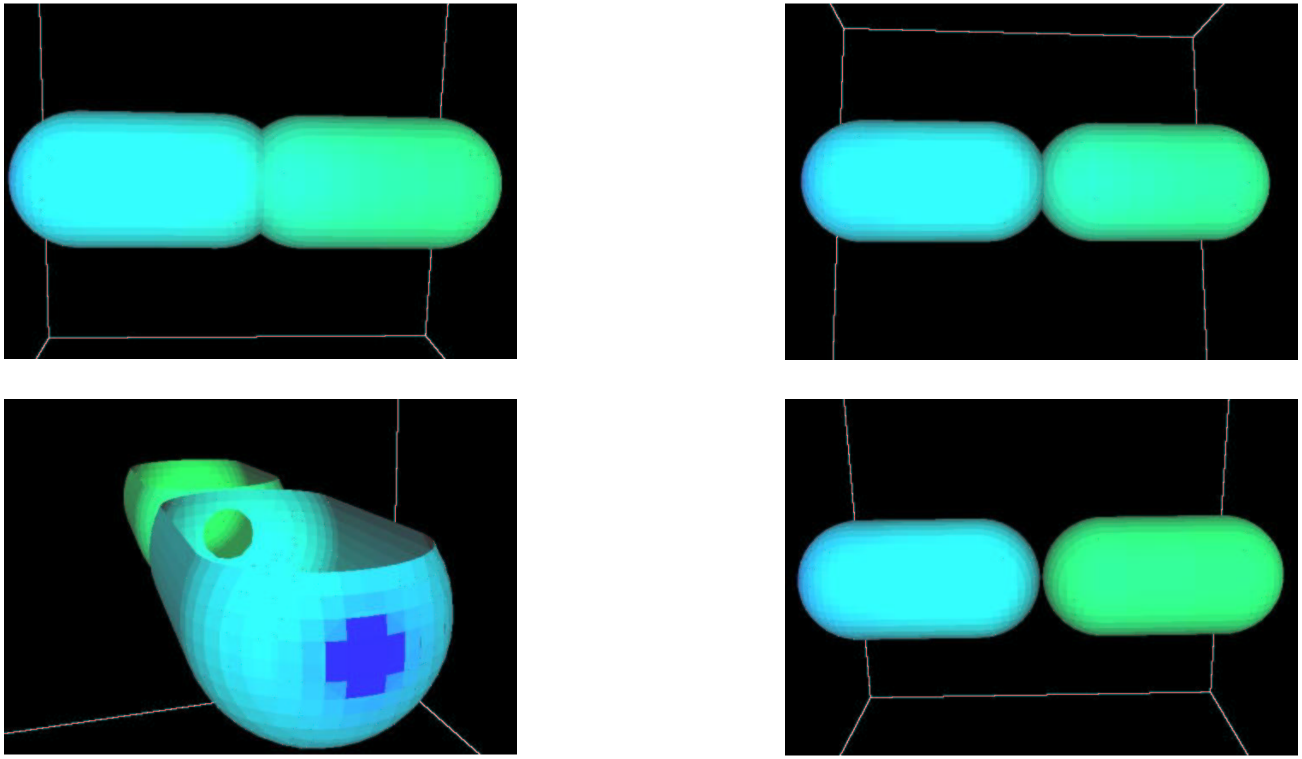
If we use Crank-Nicolson for the second step, the resulting method is unstable. To obtain a stable, second-order accurate method, must use an implicit Runge-Kutta method with better stability properties.

$$(I - r_1\Delta)(I - r_2\Delta)^{n+1} = (I + a\Delta)^{n} + \Delta t(I + r_4\Delta)f^{n+\frac{1}{2}}$$

$$r_1 + r_2 + a = \Delta t\,,\, r_1 + r_2 + r_4 = \frac{\Delta t}{2}\,,\, r_1 + r_2 > \frac{\Delta t}{2}$$



Moving Boundary Calculation in Three Dimensions



To treat more complex problems, we

• Decompose them into pieces, each one of which is well-understood, and between which the coupling is not too strong;

• Use numerical methods based on our understanding of the components, coupled together using predictor-corrector methods in time.

Example: Incompressible Navier-Stokes equations

$$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \nabla p = \nu \Delta \vec{u}$$
$$\nabla \cdot \vec{u} = 0$$

These equations can be splitting into three pieces:

$$\begin{aligned}
&\text{Hyperbolic:} && \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = 0 \\
&\text{Parabolic:} && \frac{\partial \vec{u}}{\partial t} = \nu \Delta \vec{u} \\
&\text{Elliptic:} && \Delta p = \nabla \cdot (-\vec{u} \cdot \nabla \vec{u} + \nu \Delta \vec{u})
\end{aligned}$$

Problems Arising in Decomposition into Classical Components

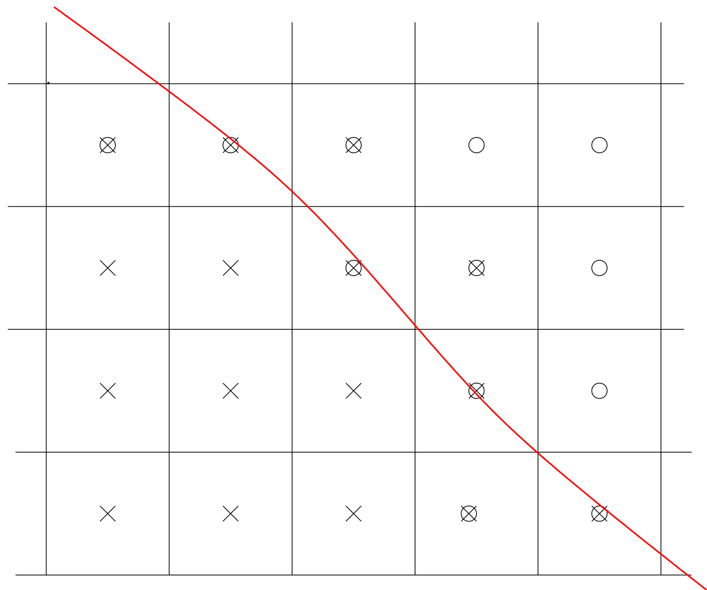
Using asymptotics to eliminate fast scales, or split slow and fast scales.

- Low Mach number asymptotics to eliminate acoustic scales: incompressible flow, low-Mach-number combustion, anelastic models for geophysical flows (Rehm and Baum, 1978; Majda and Sethian, 1985; Lai, Bell, Colella, 1993).
- Allspeed methods - splitting dynamics into vortical and compressive components (Colella and Pao, 1999).
- Methods for splitting the fast dielectric relaxation dynamics in charged-fluid models of "almost" quasineutral plasmas (Vitello and Graves, 1997; Colella, Dorr, and Wake, 1999).

All of these approaches lead to the introduction of redundant equations or constraints: $p = const.$, $\dot{q}_{net} = ...$. The presence of such constraints complicate the formulation of time-discretization methods.

Hyperbolic PDEs containing gauge constraints, such as ideal MHD ($\nabla \cdot \vec{B} = 0$) or solid mechanics, are well-posed only if the constraint is satisfied. Truncation error may cause the constraint to be violated (Miller and Colella, 2001; Crockett, et. al., to appear).

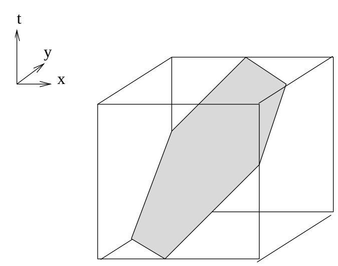
Cartesian Grid Discretization of Free Boundary Problems



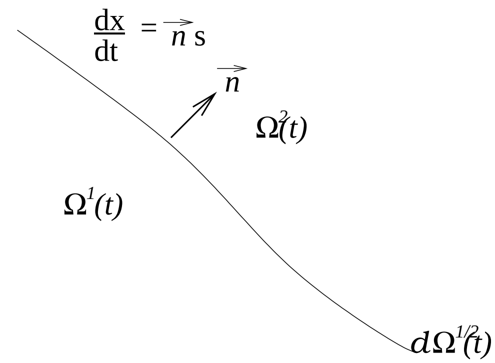
• Solution is double-valued on all cells intersecting the free boundary.

• Finite-volume discretization of conservation laws on each control volume on either side of the front.

• Motion of the front and discretization in the interior are coupled via the jump relations:
$\kappa^{n+1}U^{n+1} = \kappa^n U^n + \{$ sum of fluxes $\}$



Hyperbolic Free Boundary Problems



$$\frac{\partial U^q}{\partial t} + \nabla \cdot \vec{F}^q = 0 \text{ on } \Omega^q(t)\ ,\ q = 1, 2$$

$$[\vec{F} \cdot \vec{n} - sU] = g \text{ on } \partial\Omega^{1/2}(t)$$

$$[f] \equiv f^2 - f^1$$

• Discrete geometric quantities are a function of time, e.g., $\kappa = \kappa(t)$.

• Divergence theorem is applied in space-time to obtain discrete evolution equation:

$$0 = \int \frac{\partial U}{\partial t} + \nabla \cdot \vec{F} dxdt =$$

$$\bar{\kappa}^{n+1}\bar{U}^{n+1} - \bar{\kappa}^n \bar{U}^n + \frac{\Delta t}{h}\Big(\sum_{s \in faces} \bar{\alpha}_s \vec{F}_s \cdot \vec{n}_s + \bar{\alpha}_B(\vec{F} \cdot \vec{n}_B - sU)\Big)$$

Chenges to Fixed-Boundary Algorithm

- Riemann problem used to compute fluxes, speed of the front.
- Small-cell stability: mass increments are redistributed along characteristics in the direction normal to front.

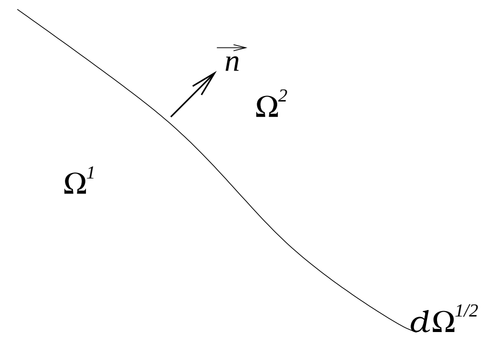
$$\delta M = \delta M^{+} + \delta M^{-}$$

$$\delta M^{+} = \sum_{\lambda_k \geq 0} (l_k \cdot \delta M) r_k$$

  δM^{+} remains on the same side of the front as it was generated on, while δM^{-} is redistributed across the front.

- Accuracy: for genuinely nonlinear waves, free boundary is noncharacteristic, so solution error is one order smaller that truncation error in max norm.

Elliptic Free Boundary Problems



$$\beta \Delta \phi^q = \rho^q \text{ on } \Omega^q \ , \ q = 1, 2$$

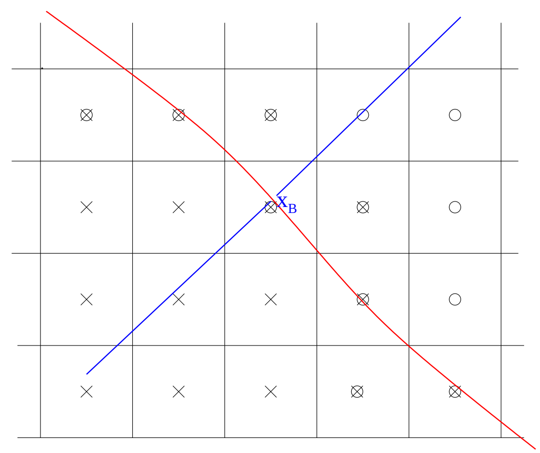
$$[\beta \frac{\partial \phi}{\partial n}] = g_N \ , \ [\phi] = g_D \text{ on } \partial \Omega^{1/2}$$

Given the values at the cell centers, the algorithm for the fixed boundary can be used to evaluate the operator, provided that one can find the values for ϕ_B^q. The jump relations lead to a pair of linear equations for ϕ_B^q:

$$\phi_B^1 - \phi_B^2 = g_D(\vec{x}_B)$$

$$\beta^1 \frac{d\Phi^1}{dr} - \beta^2 \frac{d\Phi^2}{dr} = g_N(\vec{x}_B)$$

Where $\Phi^q(r)$ are the interpolating polynomials along the normal directions from $\vec{x}_B$.



Free Boundary Problems for Diffusion

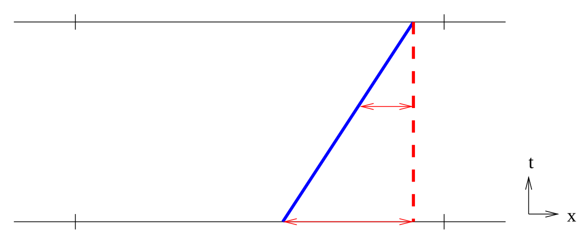
$$\frac{\partial T^\alpha}{\partial t} = D^\alpha \Delta T^\alpha + f^\alpha \text{ on } \Omega^\alpha(t)\text{ , } \alpha = 1,2$$

$$[D\frac{\partial T}{\partial n}] = g_N\text{ , } [T] = g_D \text{ on } \partial\Omega^{1/2}(t)$$

s is prescribed (*not* the Stefan problem).
As before, we convert a moving boundary problem into a sequence of problems on fixed boundaries.

$$[D\frac{\partial T}{\partial n}] = g_N + \vec{\delta}\cdot[D\nabla\frac{\partial T}{\partial n}]$$

$$[T] = g_D + \vec{\delta}\cdot[\nabla T] + \frac{1}{2}\vec{\delta}\cdot[\nabla\nabla T]\cdot\vec{\delta}$$



Future Work and Open Questions

- Adaptive mesh refinement.

- Software infrastructure.

- Decomposition into classical components: phase change boundaries, surface tension.

- Consistent discretization methods for free-boundary case.

- Other applications: magnetic fusion, combustion, cell modeling, bio-MEMS.