Lecture XI: Fluid Simulation

(based on “Fluid Simulation” course notes from SIGGRAPH 2007)

Example: Fluid Simulation

https://youtu.be/F5KuP6qEuew

Coupled Particle System

- Particles interact with each other depending on their spatial relationship.
  - these relationships are dynamic, so geometric and topological changes can take place.
- Each particle p_i has a potential energy E_{Pi}.
  - The sum of the pairwise potential energies between the particle p_i and the other particles.

$$E_{Pi} = \sum_{j \neq i} E_{Pij}$$

Coupled Particle System

- The force f_i applied on the particle at position p_i is

$$f_i = -\nabla_{p_i E_{Pi}} = -\sum_{j \neq i} \nabla_{p_i E_{Pij}}$$

where $\nabla_{p_i E_{Pi}} = \left(\frac{dE_{Pi}}{dx_i}, \frac{dE_{Pi}}{dy_i}, \frac{dE_{Pi}}{dz_i}\right)$

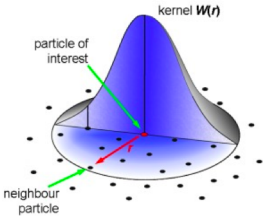
- Reducing computational costs by localizing.
  - potential energies weighted according to distance to particle.

Smoothed Particle Hydrodynamics (SPH)

- Any quantity A at any point r is given by

$$A(r) = \sum_j m_j \frac{A_j}{\rho_j} W(|r - r_j|, h)$$

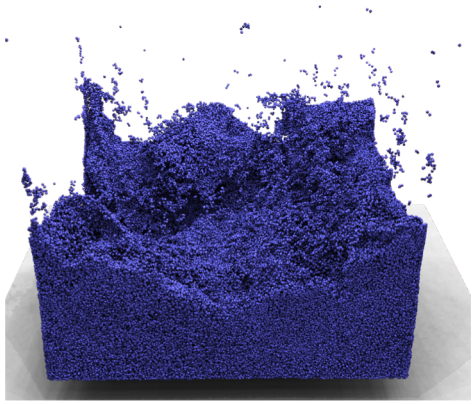
  - W: smoothing kernel
  - ρ_j: particle density
    - usually Gaussian function or cubic spline.
  - h: smoothing length.



http://www.nuigalway.ie/media/publicsub-sites/engineering/images/bio_sphkernel1.jpg

- Example: the density can be calculated as

$$\rho(r) = \sum_j m_j W(|r - r_j|, h)$$



http://rnd-zimmer.de/images/sph_particles2.png

- Applied to pressure and viscosity forces.
- External forces are applied directly to the particles.

Smoothed Particle Hydrodynamics

- Derivatives of quantities: by derivatives of W:

$$\nabla A(r) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(|r - r_j|, h)$$

- Varying h ⇔ tunes the resolution of a simulation locally.
  - Typically use a large length in low particle density regions and vice versa.
- Pro: easy to conserve mass (constant number of particles).
- Con: difficult to maintain material incompressibility.

Reminder: Navier-Stokes Equations

- Representing the conservation of mass and momentum for an incompressible fluid ($\nabla \cdot \vec{u} = 0$):

$$\overbrace{\rho(\underbrace{\vec{u}_t}_{\text{Unsteady acceleration}} + \underbrace{\vec{u} \cdot \nabla \vec{u}}_{\text{Convective acceleration}})}^{\text{Material Derivative } \frac{D\vec{u}}{Dt}} = \underbrace{\nu \nabla \cdot (\nabla \vec{u})}_{\text{Viscosity}} \underbrace{- \nabla p}_{\text{Pressure gradient}} + \underbrace{\vec{f}}_{\text{External body forces}}$$

  - p: pressure field
  - ν: kinematic viscosity.
  - f: body force per density (usually just gravity ρg).

Reduced form

- No Viscosity:

$$\rho(\vec{u}_t + \vec{u} \cdot \nabla \vec{u}) = -\nabla p + \vec{f}$$

- Incompressibility:

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

- Denoted as Euler equations.
- Solid-wall boundary conditions: $\vec{u} \cdot \hat{n} = 0$
  - There is no velocity on the boundary!
  - Bouncing creates force\impulse.
  - Relate: shooting up and falling own.

Splitting

- Consider PDE:

$$\frac{dq}{dt} = a(q) + b(q)$$

- Solution mechanism: one after the other:

$$q'(t) = q(t) + \Delta t a(q)$$
$$q'(t + \Delta t) = q'(t) + \Delta t b(q')$$

- First-order accurate

- Good for: when solving separately is much easier.

$$\rho\left(\frac{D\vec{u}}{Dt}\right) = -\nabla p + \vec{f}$$

Discrete inviscid NS Equations

- Three split steps in each iteration:
- Advection: $\frac{D\vec{u}}{Dt} = 0.$
- Body forces: $u_t = \vec{f}$.
- Pressure and incompressibility:

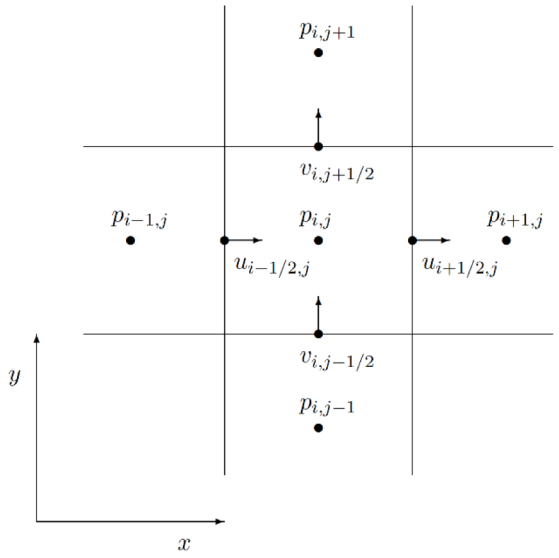
$$\rho u_t + \nabla p = 0 \text{ so that}$$
$$\nabla \cdot \vec{u} = 0$$

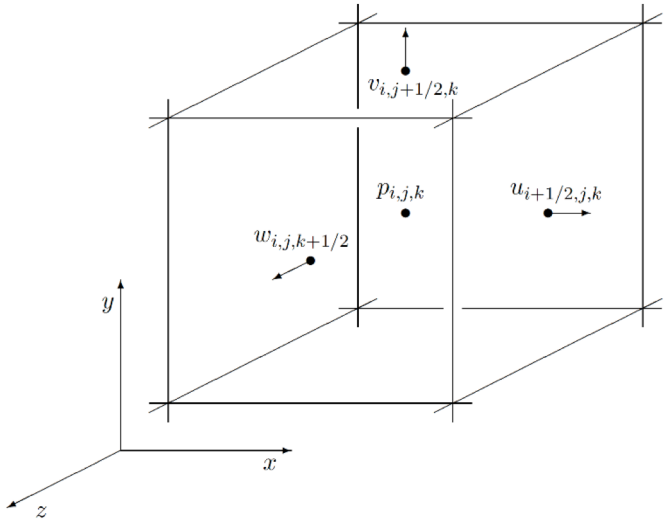
Discrete inviscid NS Equations

- For each time step:
- $U(t)$: all discrete values of $\vec{u}$. F for $\overrightarrow{f}$.
  - $U^A = \mathbf{advect}(U(t), \Delta t, U(t))$ (Advection)
  - $U^B = U^A + \Delta t F$ (Body forces)
  - $U(t + \Delta t) = \mathbf{project}(U^B, \Delta t)$ (Pressure and incompressibility)

MAC Grid

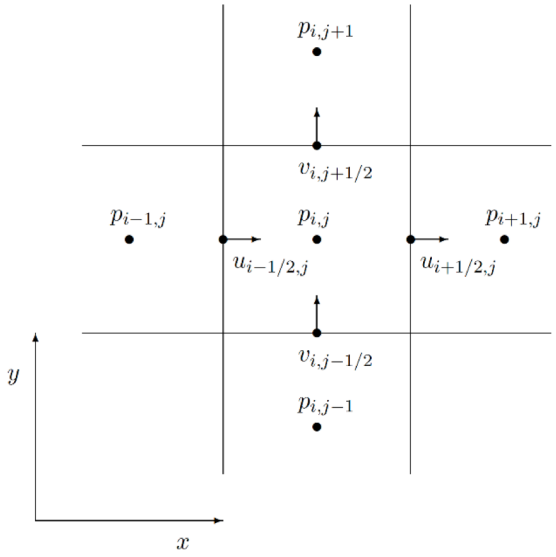
- Marker-and-Cell
- Different values are stored in different locations
- Advantages:
  - Stability
  - Staggered instead of collocated.

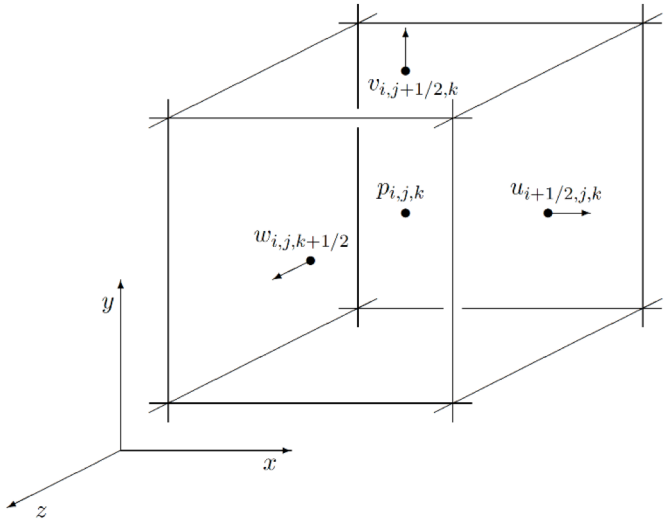




MAC Grid

- In our case:
  - Velocities: on mid-edges
    - Each dimension $\vec{u} = (u, v, w)$ on each type of mid-edge!
  - Pressures: in mid cells.
  - Interpolation: by bilinear\trilinear functions.



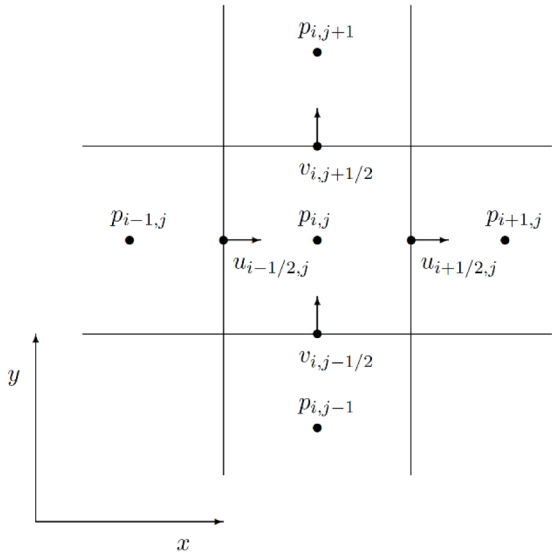


Derivatives

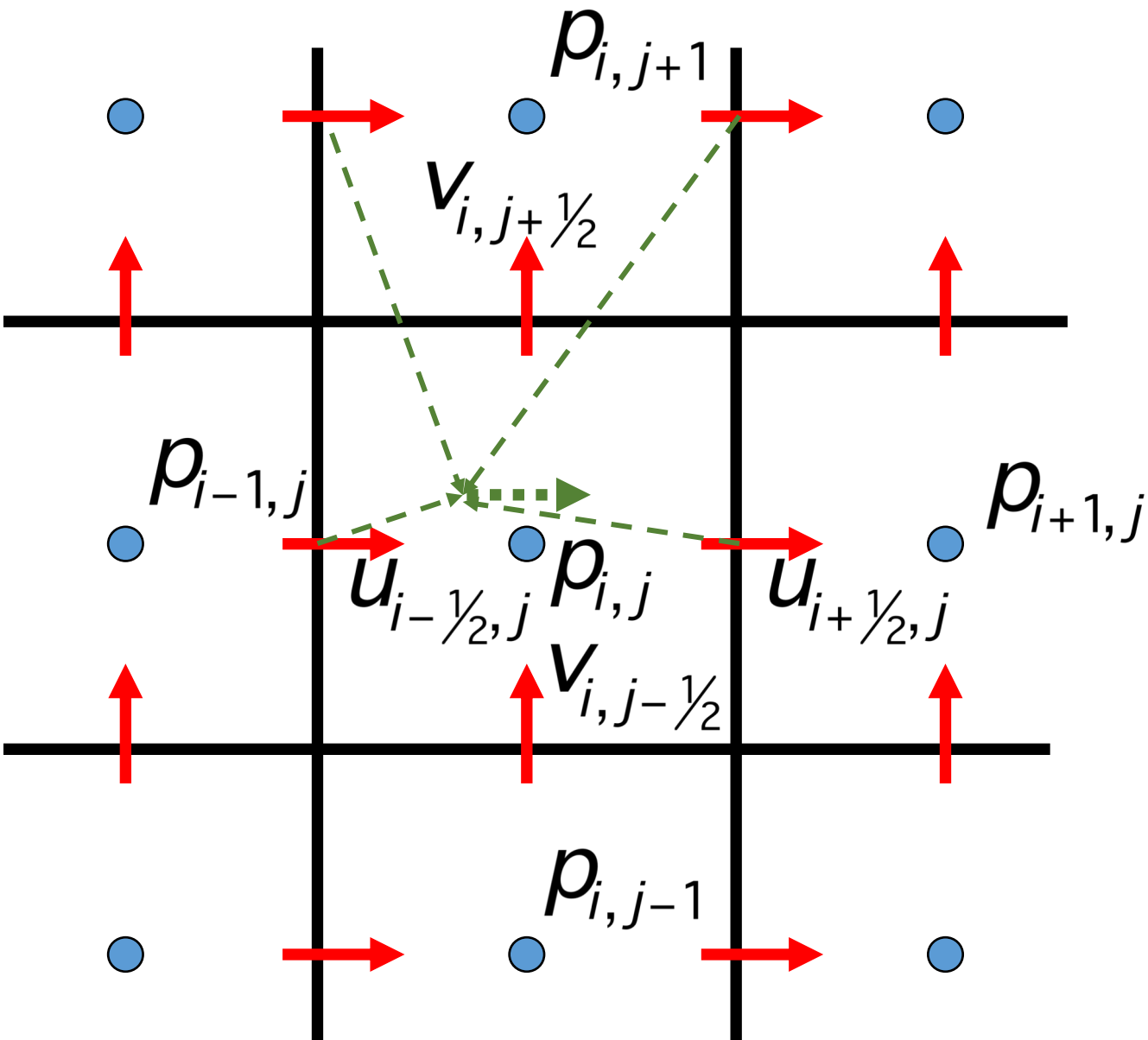
- Central differences:

$$\frac{\partial u}{\partial x} \approx \frac{u_{i+1/2} - u_{i-1/2}}{\Delta x}$$

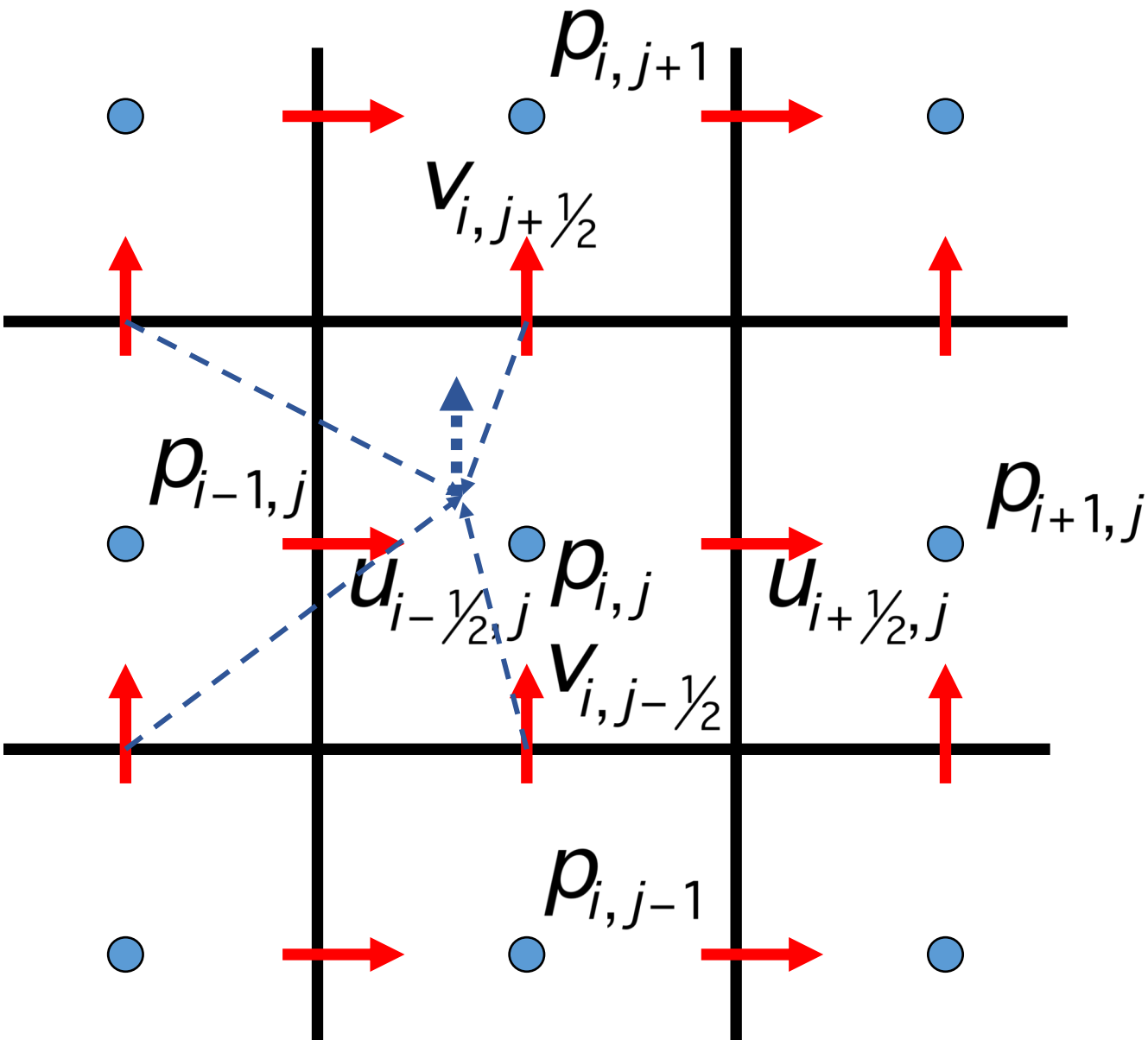
- Derivatives of mid-edges live on grid centers.
- And vice versa.



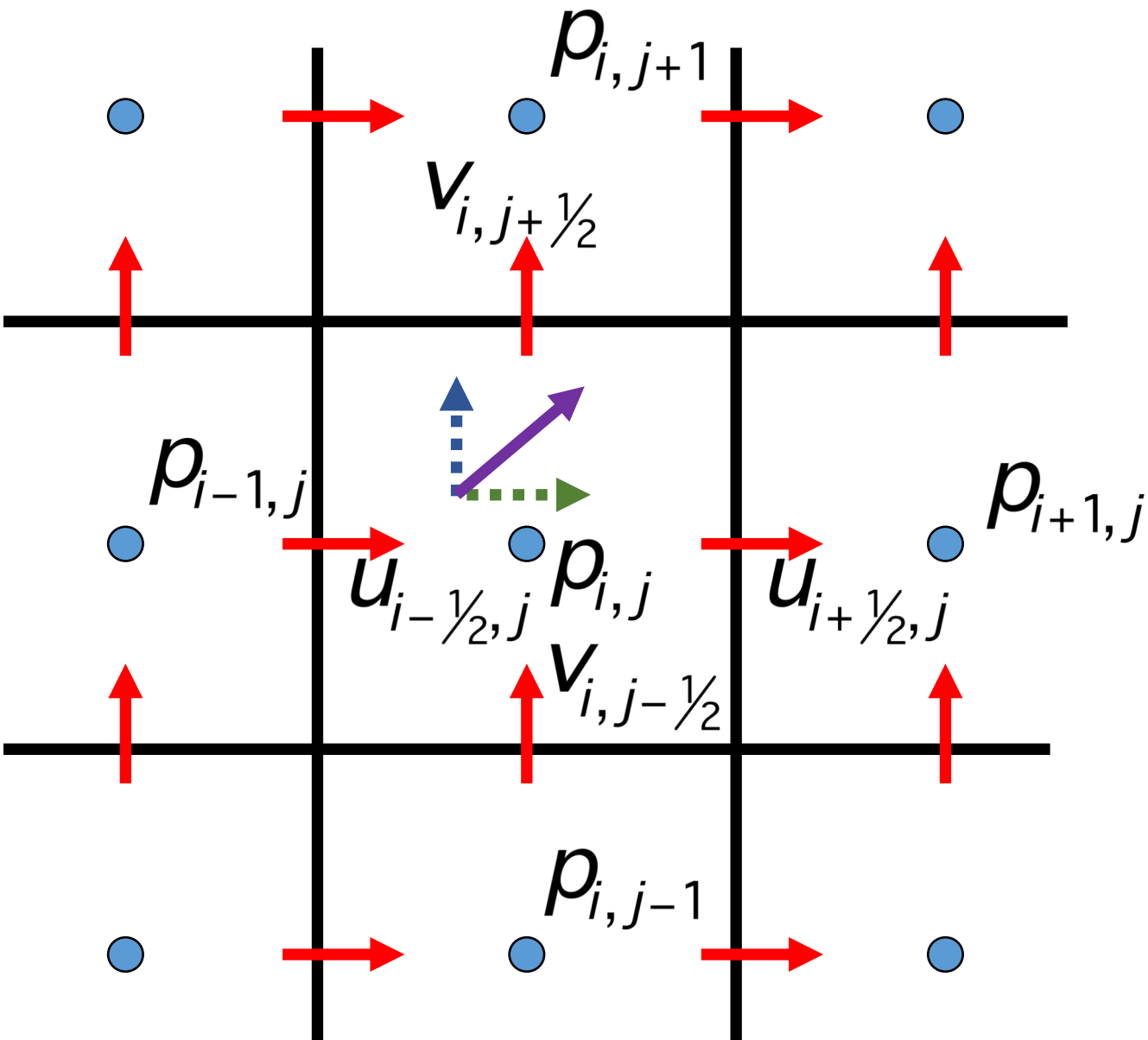
Interpolation



Interpolation



Interpolation



Advection

- Continuous equation:

$$\frac{Dq}{Dt} = q_t + \vec{u} \cdot \nabla q = 0$$

- Naïve discretization:

$$\frac{q_i(t + \Delta t) - q_i(t)}{\Delta t} + (u_i, v_i, w_i) \frac{q_{i+1}(t) - q_{i-1}(t)}{2\Delta x}$$

- Conditionally unstable!
  - Will blow up for every Δt.

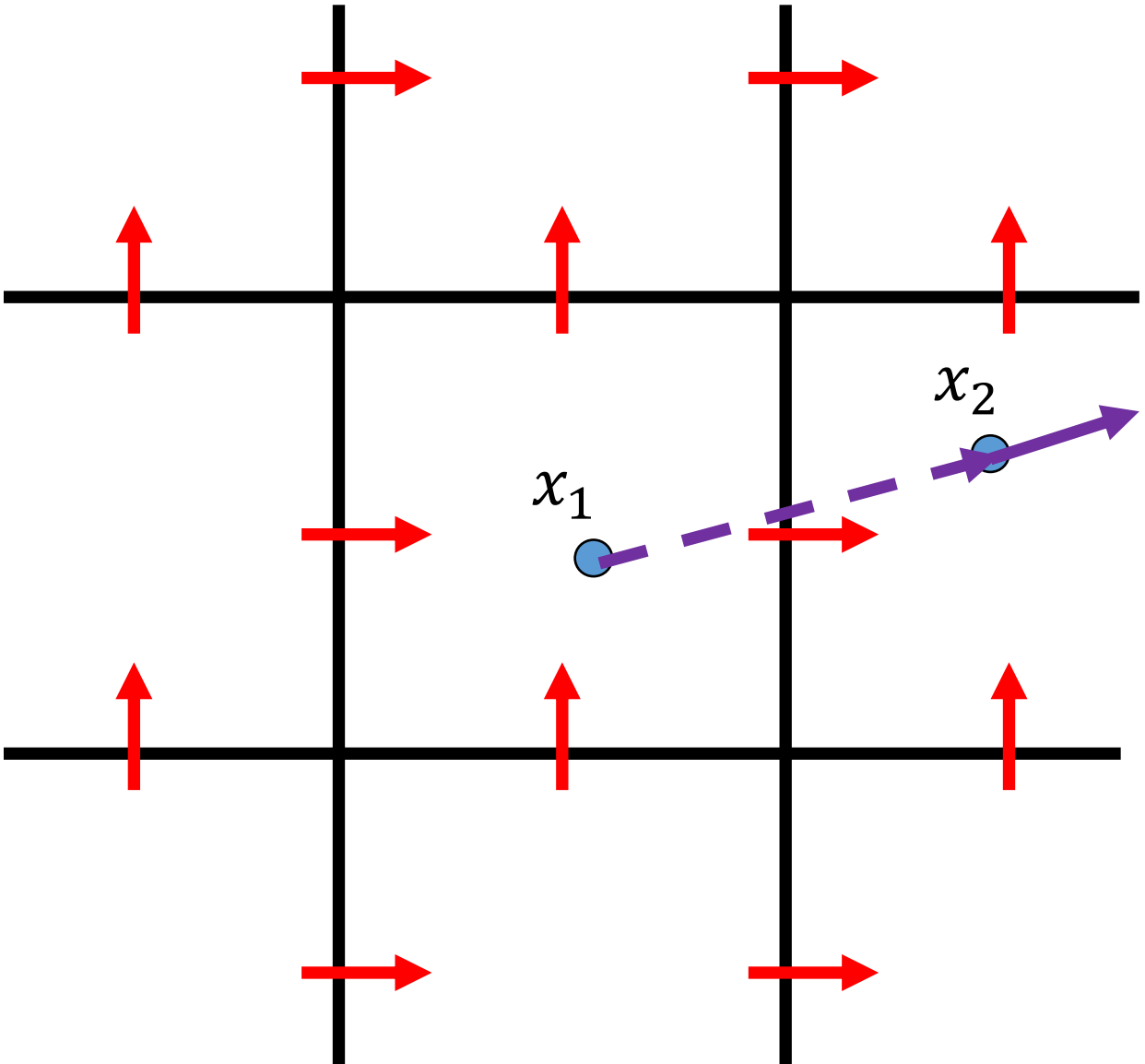
Semi-Lagrangian

- Recall that $\frac{Dq}{Dt} = 0$ is trivial for Lagrangian particles.
- Insight: if we have $q(t)$ at some grid point x_2 with velocity $\vec{u}$ it was advected from another point x_1 in the grid:

$$\vec{u} = \frac{x_2 - x_1}{\Delta t}$$

- Value $q(t)$ at x_2 = value $q(t - \Delta t)$ in x_1.
- Getting $q(t - \Delta t)$ in x_1 from bi\trilinear interpolation.

Semi-Lagrangian



Extrapolation

- Position x_1 might be outside of the fluid
- We need to extrapolate the flow field to be defined everywhere.
- Grid boundaries: ”no-stick” condition: $\vec{u} \cdot \hat{n} = 0$
  - More simply in our case: just $\vec{u} = 0$ outside.
- In grid, outside of fluid: more complicated…
  - Basic idea: closest-neighbor interpolation.

Incompressibility

- Let us take the divergence on $\rho u_t + \nabla p = 0$:

$$\nabla \cdot \rho \frac{\partial u}{\partial t} + \nabla \cdot \nabla p = 0 \rightarrow$$

$$\rho \frac{\partial}{\partial t} (\nabla \cdot u) + \nabla \cdot \nabla p = 0$$

- That's because "$\frac{\partial}{\partial t}$" and "$\nabla \cdot$" work on independent variables and therefore commute.

Incompressibility

- We need to solve for $\nabla \cdot \vec{u} = 0$
  - Pressure is solved for to make this happen.
- Idea: discretize $\rho \frac{\partial}{\partial t}(\nabla \cdot u) + \nabla \cdot \nabla p = 0$ as

$$\nabla \cdot \nabla p = -\rho \frac{(\nabla \cdot u)(t + \Delta t) - (\nabla \cdot u)(t)}{\Delta t}$$

- As we want $(\nabla \cdot u)(t + \Delta t) = 0$ we solve for:

$$\Delta p = \rho \frac{(\nabla \cdot u)(t)}{\Delta t}$$

The Discrete Divergence

- Lives in faces (like pressure)

$$\nabla \cdot \vec{u} \approx \frac{u_{i+1/2} - u_{i-1/2}}{\Delta x} + \frac{v_{j+1/2} - v_{j-1/2}}{\Delta x} + \cdots$$

- Laplacian of pressure is similar:

$$\Delta p \approx \frac{p_{i,j} - p_{i-1,j}}{\Delta x} + \frac{p_{i,j} - p_{i+1,j}}{2\Delta x} + \cdots$$

- Need to solve for pressure and consequently velocity.
  - Details are a bit nasty (Section 4.3 of notes)

Velocity from pressure

- The discretization of

$$u_t = -\frac{1}{\rho}\nabla p$$

- Becomes $u(t + \Delta t) = u(t) - \frac{\Delta t}{\rho}(p_{i+1} - p_{i+1})$
  - And similarly for every dimension.
- The advantage of MAC: gradient of pressures lives on mid-edges, like velocities!

Fluid Surface Conditions

- "Ghost" pressure for solid walls:

$$p_{i+1} = p_i + \frac{\rho \Delta x}{\Delta t} u_{i+1/2}$$

- Just inverting the pressure equation.
- For free boundary (water surface): $p_i = 0$.

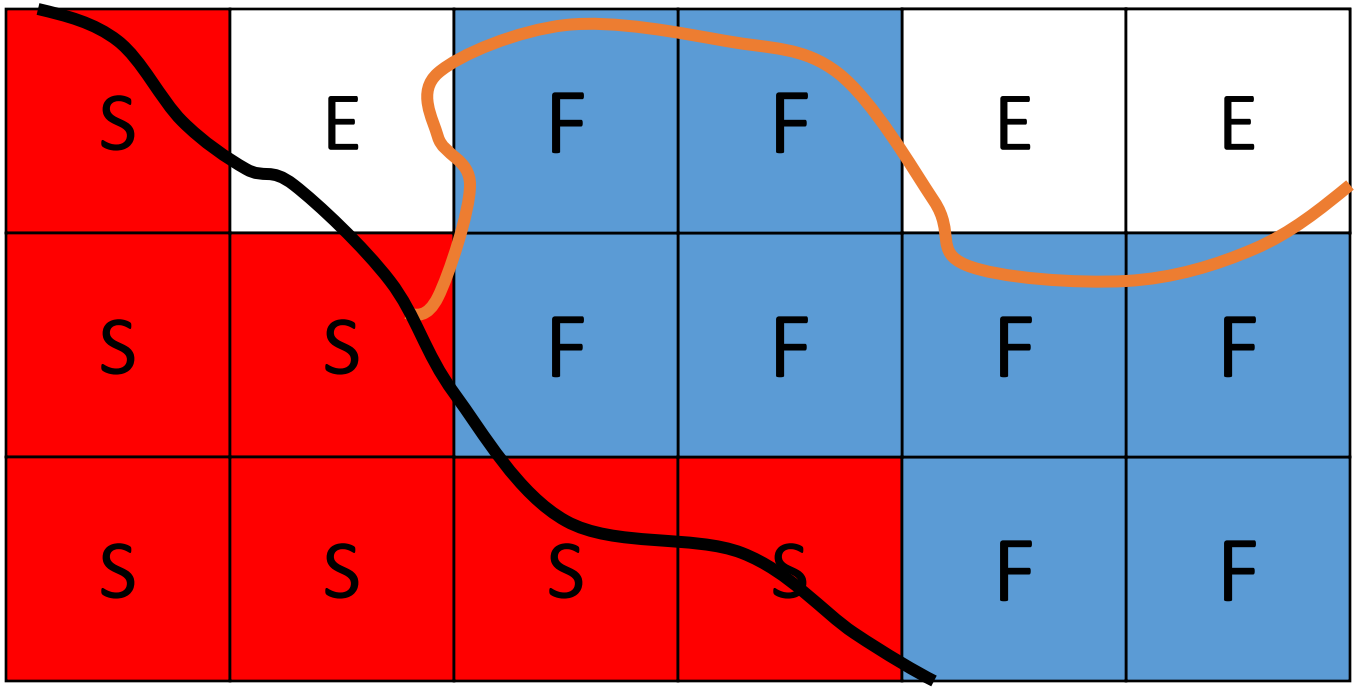
https://www.fxguide.com/wp-content/uploads/2011/08/realtimewaves.jpg

Linear Equations

- End up with a sparse set of linear equations to solve for pressure
  - Matrix is symmetric positive (semi-)definite
- In 3D on large grids, direct methods unsatisfactory
- Instead use Preconditioned Conjugate Gradient, with Incomplete Cholesky preconditioner

Discrete Boundaries

- Grid cells are either solid, fluid, or empty.



Voxelization is Suboptimal

- “8-bit water”
  - Waves less than a grid cell high aren’t “seen” by the fluid solver – thus they don’t behave right
  - Strangely textured surface
- Solid wall artifacts:
  - If boundary not grid-aligned, noticeable geometric error.
  - Slopes are turned into stairs.

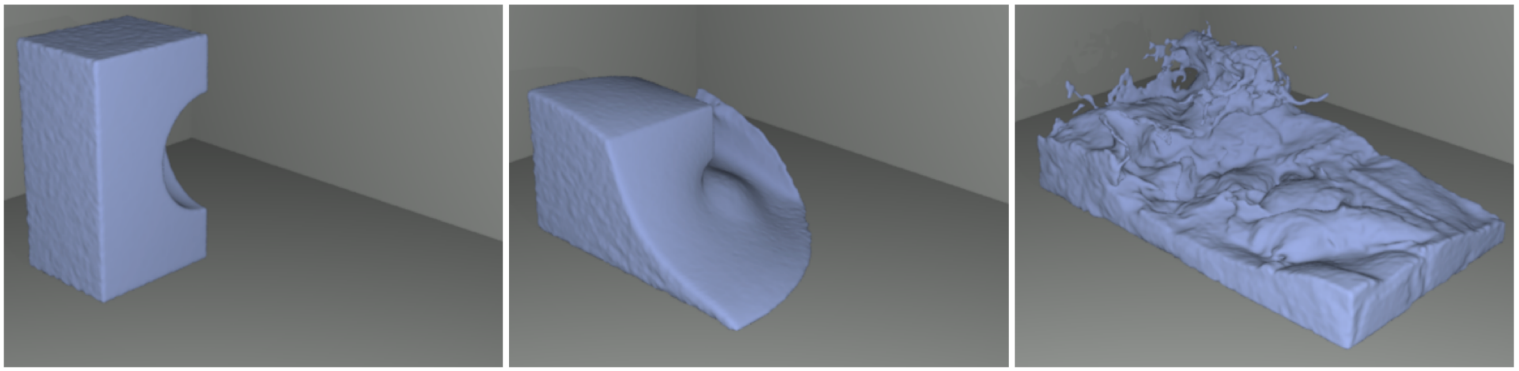
Marker Particles

- Simplest approach is to use marker particles
- Sample fluid volume with particles $\{x_p\}$
- At each time step, mark grid cells containing any marker particles as Fluid, rest as Empty or Solid
- Advect particles with velocity:

$$\frac{Dx_p}{Dt} = 0$$

Rendering Marker Particles

- Need to wrap a surface around the particles
  - e.g. blobbies, or Zhu & Bridson ‘05
- Problem: rendered surface has bumpy detail that the fluid solver doesn’t have access to
  - The simulation can’t animate that detail properly if it can’t “see” it.
- Result: looks fine for splashy, noisy surfaces, but fails on smooth, glassy surfaces.



Level Sets

- Model a function $\phi(x)$ on the grid so that the fluid boundaries are at $\phi(x) = 0$
- In particular, best behaved type of implicit surface function is signed distance:

$$\phi(x) = \begin{cases} dist(x, surface) & x \text{ outside} \\ -dist(x, surface) & x \text{ inside} \end{cases}$$

- e.g. the book [Osher&Fedkiw'02]

Surface Capturing

- Evolving $\phi(x)$ on the grid by advection as well.
- The surface ($\phi(x)$ =0) moves at the velocity of the fluid
- Topological changes automatically inferred.

$$\frac{D\phi}{Dt} = 0$$

Reinitializing

- Advecting $\phi(x)$ this way doesn't preserve signed distance property
  - Eventually gets badly distorted
  - Causes problems particularly for extrapolation, also for accuracy in general
- Reinitialize: recompute distance to surface every once in a while.
  - Ideally shouldn't change surface at all, just values in the volume