Introduction to Water Simulation II
---Numerical Solution

Mingsong Dou
Nov.17, 2010

Outline

- Splitting the Navier-Stokes Equation
- MAC Grid, a staggered grid
- Algorithms details
  - Advection
  - Add body forces
  - Make water incompressible
- From velocity field to water surface

A close look at the Navier-Stokes Equation

$$\boxed{\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u}} + \frac{1}{\rho}\nabla p = \vec{g} + \boxed{\nu \nabla \cdot \nabla \vec{u}}$$

$$= \frac{D\vec{u}}{Dt}$$

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

- Unknowns: $\vec{u}$ and p. $\vec{u}$ is what we want.
- p is constrained by the incompressibility term
- Viscosity term could be dropped
- Still too complicated to solve it in one step. It would be nice to split it into several steps.

A close look at the Navier-Stokes Equation

$$\frac{D\vec{u}}{Dt} + \frac{1}{\rho}\nabla p = \vec{g}$$

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

- Unknowns: $\vec{u}$ and p. $\vec{u}$ is what we want.
- p is constrained by the incompressibility term
- Viscosity term could be dropped
- Still too complicated to solve it in one step. It would be nice to split it into several steps.

Dived and Conquer

- A simple example

$$\frac{dq}{dt} = f(q) + g(q)$$

- Euler Solution

$$q^{n+1} = q^n + \Delta t\,[\,f(q) + g(q)]$$

- Split it into two ODEs

$$\frac{dq}{dt} = f(q)$$

$$\frac{dq}{dt} = g(q)$$

- Solve them sequentially

$$\tilde{q} = q^n + \Delta t f(q^n)$$

$$q^{n+1} = \tilde{q} + \Delta t g(\tilde{q})$$

Splitting the Fluid Equations

$$\frac{\partial \vec{u}}{\partial t} + \underbrace{\vec{u} \cdot \nabla \vec{u}}_{= f_1(\vec{u})} + \underbrace{\frac{1}{\rho} \nabla p}_{= f_3(\vec{u})} = \underbrace{\vec{g}}_{= f_2(\vec{u})}$$

- $$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = 0 \Rightarrow \frac{D\vec{u}}{Dt} = 0$$ Advection/Transportation
- $$\frac{\partial \vec{u}}{\partial t} = \vec{g}$$ Body Forces
- $$\frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho} \nabla p = 0$$ Pressure/Incompressiblity

  $$\text{s.t.} \quad \nabla \cdot \vec{u} = 0$$

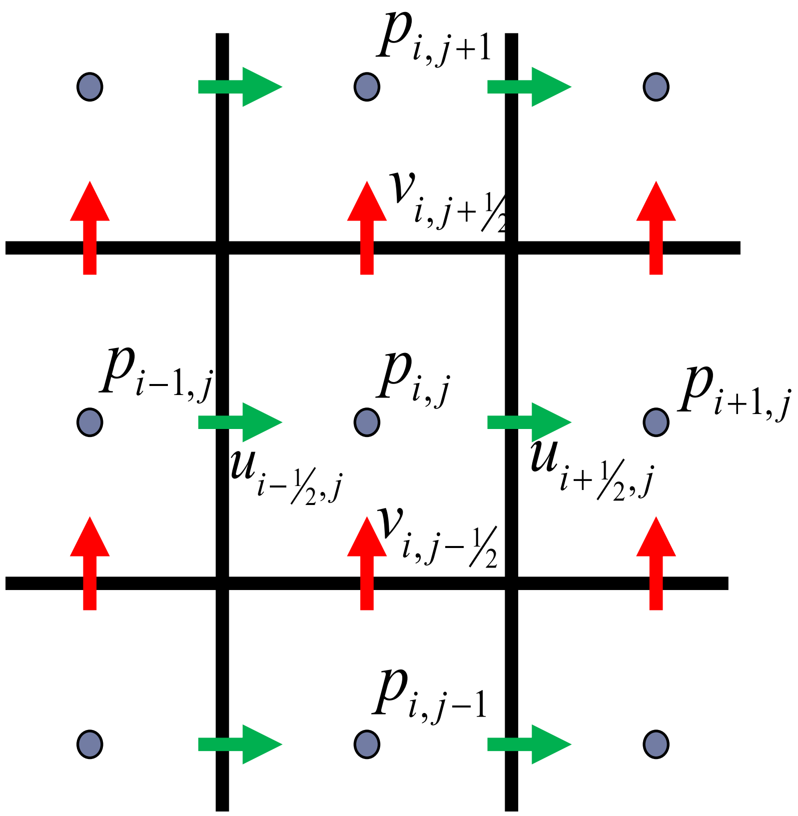
Splitting the Fluid Equations

$$\frac{\partial \vec{u}}{\partial t} + \underbrace{\vec{u} \cdot \nabla \vec{u}}_{= f_1(\vec{u})} + \underbrace{\frac{1}{\rho} \nabla p}_{= f_3(\vec{u})} = \underbrace{\vec{g}}_{= f_2(\vec{u})}$$

- Start with an initial divergence-free velocity field $\vec{u}^{(0)}$.
- For time step $n = 0, 1, 2, \ldots$
  - Determine a good time step Δt to go from time t_n to time t_{n+1}
  - Set $\vec{u}^A = \texttt{advect}(\vec{u}^n, \Delta t, \vec{u}^n)$ $\qquad \frac{D\vec{u}}{Dt} = 0$
  - Add $\vec{u}^B = \vec{u}^A + \Delta t \vec{g}$ $\qquad \frac{\partial \vec{u}}{\partial t} = \vec{g}$
  - Set $\vec{u}^{n+1} = \texttt{project}(\Delta t, \vec{u}^B)$ $\qquad \frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho}\nabla p = 0 \quad \text{s.t.} \quad \nabla \cdot \vec{u} = 0$

Discretization In Space

- MAC Grid (staggered grid)



- Benefits
  - Accurate central difference
- Downsides
  - Variables spread up, interpolation needed.

The Disaster of Simple Grid When Evaluating the Derivatives

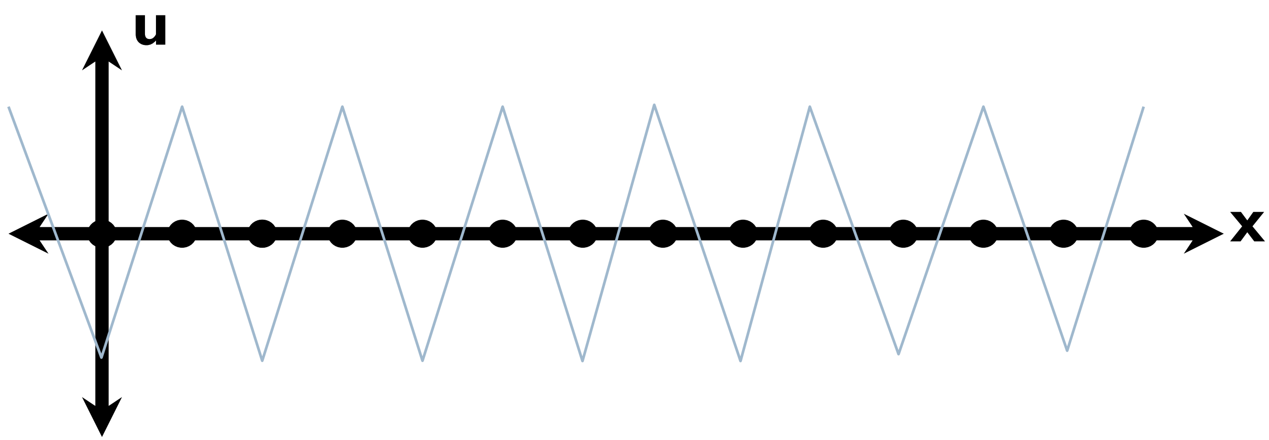
- forward or backward difference

$$\left(\frac{\partial q}{\partial x}\right)_i \approx \frac{q_{i+1} - q_i}{\Delta x}$$

- central difference

$$\left(\frac{\partial q}{\partial x}\right)_i \approx \frac{q_{i+1} - q_{i-1}}{2\Delta x}$$

- A bad situation of central difference, zero derivative everywhere.



The Disaster of Simple Grid When Evaluating the Derivatives

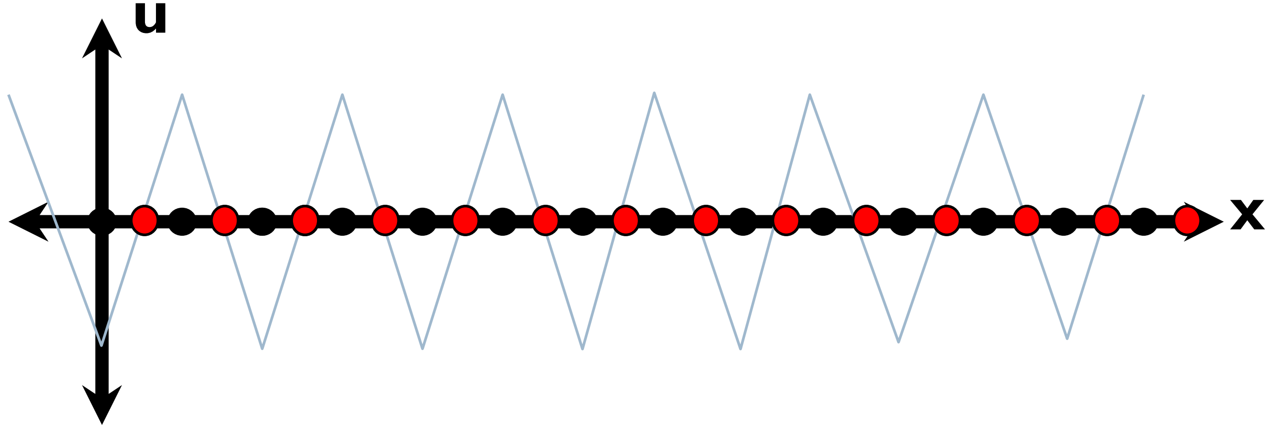
- forward or backward difference

$$\left(\frac{\partial q}{\partial x}\right)_i \approx \frac{q_{i+1} - q_i}{\Delta x}$$

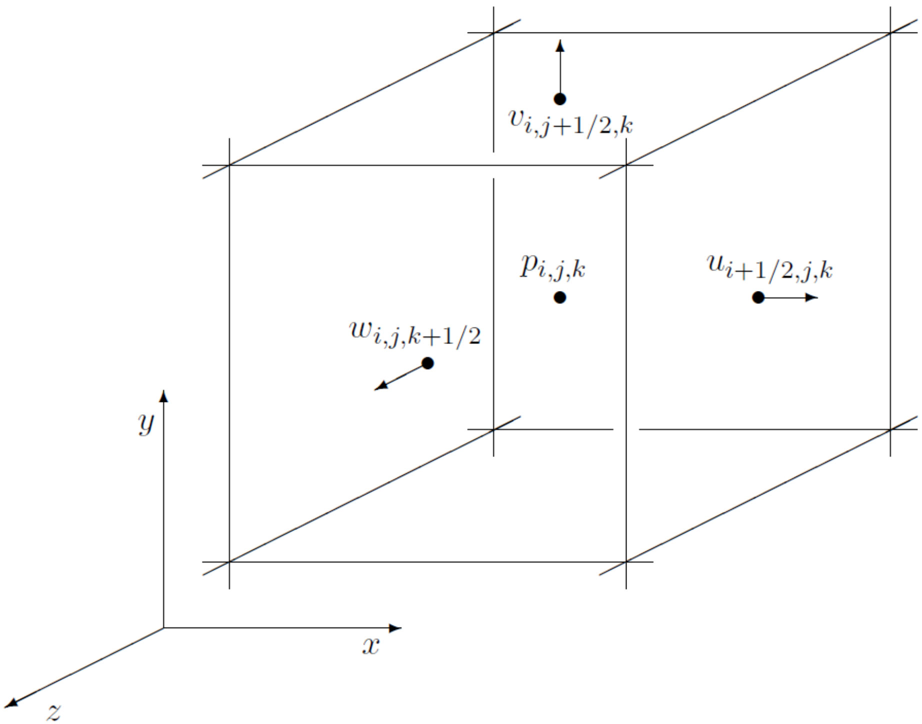
- central difference

$$\left(\frac{\partial q}{\partial x}\right)_i \approx \frac{q_{i+1} - q_{i-1}}{2\Delta x}$$

- Staggered grid hands it well



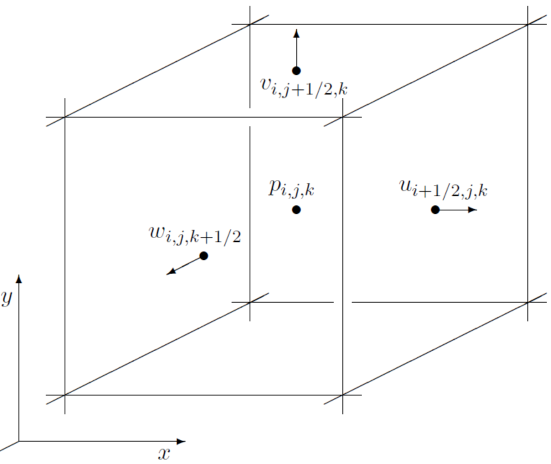
3D MAC Grid



- For each Cell
  - pressure locates at the center
  - Each component of the velocity takes up two faces
  - Each face only has one component of the velocity, interpolation needed.

- Derivatives of velocity at the center of the cell…
- Derivatives of pressure at the center of each facet…

3D MAC Grid



- Interpolate the velocity at the center of the cell and its facets.

$$\vec{u}_{i,j,k} = \left(\frac{u_{i-1/2,j,k} + u_{i+1/2,j,k}}{2}, \quad \frac{v_{i,j-1/2,k} + v_{i,j+1/2,k}}{2}, \quad \frac{w_{i,j,k-1/2} + w_{i,j,k+1/2}}{2} \right)$$

$$\vec{u}_{i+1/2,j,k} = \left(u_{i+1/2,j,k}, \quad \frac{\begin{array}{c} v_{i,j-1/2,k} + v_{i,j+1/2,k} \\ +v_{i+1,j-1/2,k} + v_{i+1,j+1/2,k} \end{array}}{4}, \quad \frac{\begin{array}{c} w_{i,j,k-1/2} + w_{i,j,k+1/2} \\ +w_{i+1,j,k-1/2} + w_{i+1,j,k+1/2} \end{array}}{4} \right)$$

$$\vec{u}_{i,j+1/2,k} = \left(\frac{\begin{array}{c} u_{i-1/2,j,k} + u_{i+1/2,j,k} \\ +u_{i-1/2,j+1,k} + u_{i+1/2,j+1,k} \end{array}}{4}, \quad v_{i,j+1/2,k}, \quad \frac{\begin{array}{c} w_{i,j,k-1/2} + w_{i,j,k+1/2} \\ +w_{i,j+1,k-1/2} + w_{i,j+1,k+1/2} \end{array}}{4} \right)$$

$$\vec{u}_{i,j,k+1/2} = \left(\frac{\begin{array}{c} u_{i-1/2,j,k} + u_{i+1/2,j,k} \\ +u_{i-1/2,j,k+1} + u_{i+1/2,j,k+1} \end{array}}{4}, \quad \frac{\begin{array}{c} v_{i,j-1/2,k} + v_{i,j+1/2,k} \\ +v_{i,j-1/2,k+1} + v_{i,j+1/2,k+1} \end{array}}{4}, \quad w_{i,j,k+1/2} \right)$$

STEP I: Advect the Velocity

- $\boxed{\dfrac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = 0 \quad \text{or} \quad \dfrac{D\vec{u}}{Dt} = 0}$
- $$\frac{\partial \vec{u}}{\partial t} = \vec{g}$$
- $$\frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho} \nabla p = 0 \quad \text{s.t.} \quad \nabla \cdot \vec{u} = 0$$

Advecting Quantities

- The goal is to solve

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

  "the advection equation" for any grid quantity q
  - advect each component of velocity separately

- Intead of treating it in Euler fashion by directly solving

$$\frac{\partial q}{\partial t} + u\frac{\partial q}{\partial x} = 0$$

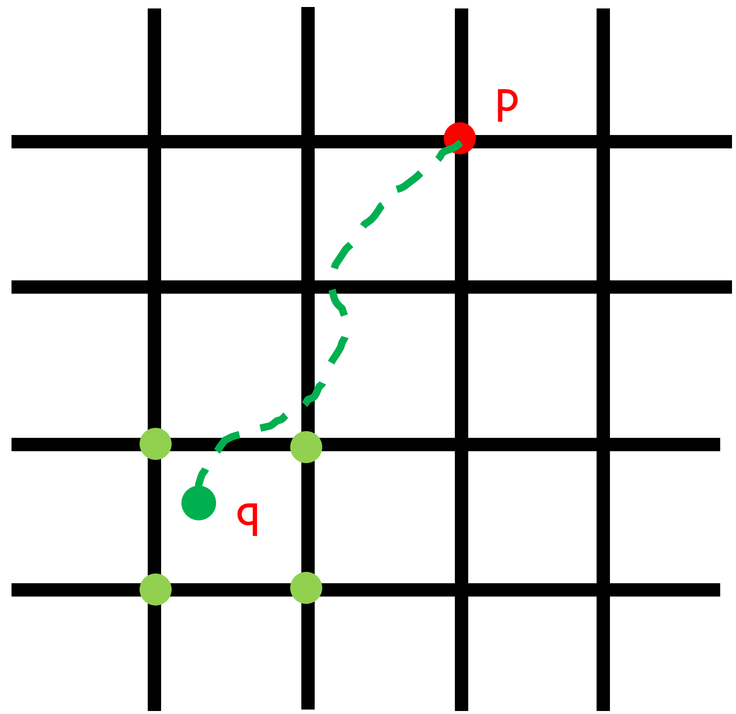
  we are using Lagrangian notion.

- We're on an Eulerian grid, though, so the result will be called "semi-Lagrangian".

Semi-Lagrangian Algorithm

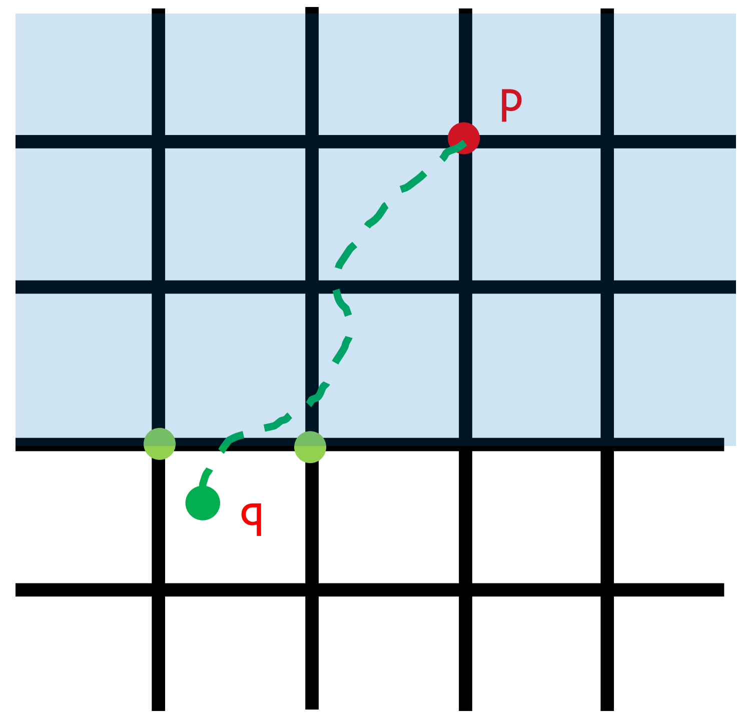
- For each grid point, find X^{old}, and use the quantity (of previous time step) at this position as the new quantity of the grid point.
- Interpolation may be necessary. Be careful when doing interpolation in staggered grid.
- Forward Euler is adequate to find the old position.

$$\vec{x}_q = \vec{x}_p - \Delta t \vec{u}_p$$



Boundary Conditions

- What if the particle flies out of the water boundary
  - due to numeric error, just extrapolate from nearest points on the boundary;
  - due to water flowing in from outside, …



Dissipation

- Interpolation cause smoothed velocity field. Small vortices will be phased out.
- It equals to simulate a fluid with viscosity.
- Will be covered by the following lecture.

STEP II: Add Body Forces

- $$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = 0 \quad \text{or} \quad \frac{D\vec{u}}{Dt} = 0$$
- $$\boxed{\frac{\partial \vec{u}}{\partial t} = \vec{g}}$$
- $$\frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho} \nabla p = 0 \quad \text{s.t.} \quad \nabla \cdot \vec{u} = 0$$

Integrating Body Forces

- Supper Easy!!!
- Just add the new term at each grid point

$$\vec{u}^{*} = \vec{u}^{advected} + \Delta t \vec{g}$$

STEP III: Making Fluid Incompressible

- $$\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} = 0 \quad \text{or} \quad \frac{D\vec{u}}{Dt} = 0$$
- $$\frac{\partial \vec{u}}{\partial t} = \vec{g}$$
- $$\boxed{\frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho}\nabla p = 0 \quad \text{s.t.} \quad \nabla \cdot \vec{u} = 0}$$

$$\frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho}\nabla p = 0$$
$$\text{s.t.} \quad \nabla \cdot \vec{u} = 0$$

The Continuous Version

- Space is continuous, but still assume the time space is discrete. Update the velocity,

$$\vec{u}^{n+1} = \vec{u}^* - \frac{\Delta t}{\rho}\nabla p$$

- To make it incompressible, the divergence should be zero.

$$\nabla \cdot (\vec{u}^* - \frac{\Delta t}{\rho}\nabla p) = 0$$

$$-\frac{\Delta t}{\rho}\nabla \cdot \nabla p = -\nabla \cdot \vec{u}^* \qquad \text{Laplace Equation}$$

- Solve Laplace Equation to get p, then substitute p into update equation. (Right now, let's leave out boundary conditions, and assume the water is boundless.)

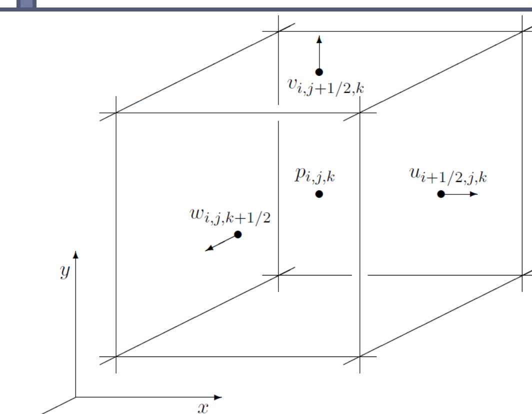
The Discrete Version

$$\frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho}\nabla p = 0$$

$$\text{s.t.} \quad \nabla \cdot \vec{u} = 0$$

- For clarity, discretize the pressure equation and the divergence constraint instead.

discretize $\vec{u}^{n+1} = \vec{u}^* - \frac{\Delta t}{\rho}\nabla p$	discretize $\nabla \cdot \vec{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$
$u^{n+1}_{i+1/2,j,k} = u_{i+1/2,j,k} - \Delta t \frac{1}{\rho} \frac{p_{i+1,j,k} - p_{i,j,k}}{\Delta x}$ $v^{n+1}_{i,j+1/2,k} = v_{i,j+1/2,k} - \Delta t \frac{1}{\rho} \frac{p_{i,j+1,k} - p_{i,j,k}}{\Delta x}$ $w^{n+1}_{i,j,k+1/2} = w_{i,j,k+1/2} - \Delta t \frac{1}{\rho} \frac{p_{i,j,k+1} - p_{i,j,k}}{\Delta x}$	$(\nabla \cdot \vec{u})_{i,j,k} \approx \frac{u_{i+1/2,j,k} - u_{i-1/2,j,k}}{\Delta x} + \frac{v_{i,j+1/2,k} - v_{i,j-1/2,k}}{\Delta x} + \frac{w_{i,j,k+1/2} - w_{i,j,k-1/2}}{\Delta x}$



$$\frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho}\nabla p = 0$$

$$\text{s.t.} \quad \nabla \cdot \vec{u} = 0$$

The Discrete Version

discretize $\vec{u}^{\,n+1} = \vec{u}^{\,*} - \frac{\Delta t}{\rho}\nabla p$	discretize $\nabla \cdot \vec{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0$
$u^{n+1}_{i+1/2,j,k} = u_{i+1/2,j,k} - \Delta t \frac{1}{\rho} \frac{p_{i+1,j,k} - p_{i,j,k}}{\Delta x}$ $v^{n+1}_{i,j+1/2,k} = v_{i,j+1/2,k} - \Delta t \frac{1}{\rho} \frac{p_{i,j+1,k} - p_{i,j,k}}{\Delta x}$ $w^{n+1}_{i,j,k+1/2} = w_{i,j,k+1/2} - \Delta t \frac{1}{\rho} \frac{p_{i,j,k+1} - p_{i,j,k}}{\Delta x}$	$(\nabla \cdot \vec{u})_{i,j,k} \approx \frac{u_{i+1/2,j,k} - u_{i-1/2,j,k}}{\Delta x} + \frac{v_{i,j+1/2,k} - v_{i,j-1/2,k}}{\Delta x} + \frac{w_{i,j,k+1/2} - w_{i,j,k-1/2}}{\Delta x}$

Substitute left equations into the right one

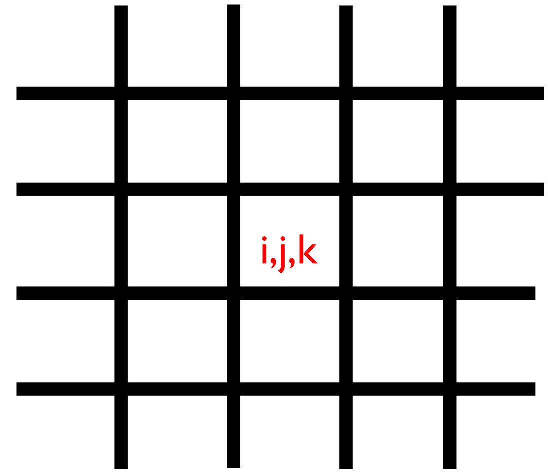
$$\frac{\Delta t}{\rho}\left(\frac{6p_{i,j,k} - p_{i+1,j,k} - p_{i,j+1,k} - p_{i,j,k+1} - p_{i-1,j,k} - p_{i,j-1,k} - p_{i,j,k-1}}{\Delta x^2}\right) = -\left(\frac{u_{i+1/2,j,k} - u_{i-1/2,j,k}}{\Delta x} + \frac{v_{i,j+1/2,k} - v_{i,j-1/2,k}}{\Delta x} + \frac{w_{i,j,k+1/2} - w_{i,j,k-1/2}}{\Delta x}\right)$$

$$-\frac{\Delta t}{\rho}\nabla \cdot \nabla p = -\nabla \cdot \vec{u}^{\,*}$$

Putting Them In Matrix-Vector Form

- Each cell has such a linear equation, combining them together we could get p at each cell.

$$\frac{\Delta t}{\rho}\left(\frac{6p_{i,j,k} - p_{i+1,j,k} - p_{i,j+1,k} - p_{i,j,k+1} - p_{i-1,j,k} - p_{i,j-1,k} - p_{i,j,k-1}}{\Delta x^2}\right) = -\left(\frac{u_{i+1/2,j,k} - u_{i-1/2,j,k}}{\Delta x} + \frac{v_{i,j+1/2,k} - v_{i,j-1/2,k}}{\Delta x} + \frac{w_{i,j,k+1/2} - w_{i,j,k-1/2}}{\Delta x}\right)$$



- Represent p of all the cells as a linear vector. Write all the linear equations into a matrix-vector form

$$\begin{bmatrix} \ddots & & & & & & & & & & & & \\ \cdots & -1 & \cdots & -1 & \cdots & -1 & 6 & -1 & \cdots & -1 & \cdots & -1 & \cdots \\ & & & & & & & & & & & & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ p_{i-1,j,k} \\ \vdots \\ p_{i,j-1,k} \\ \vdots \\ p_{i,j,k-1} \\ p_{i,j,k} \\ p_{i,j,k+1} \\ \vdots \\ p_{i,j+1,k} \\ \vdots \\ p_{i+1,j,k} \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \nabla \cdot \vec{u} \\ \vdots \end{bmatrix}$$

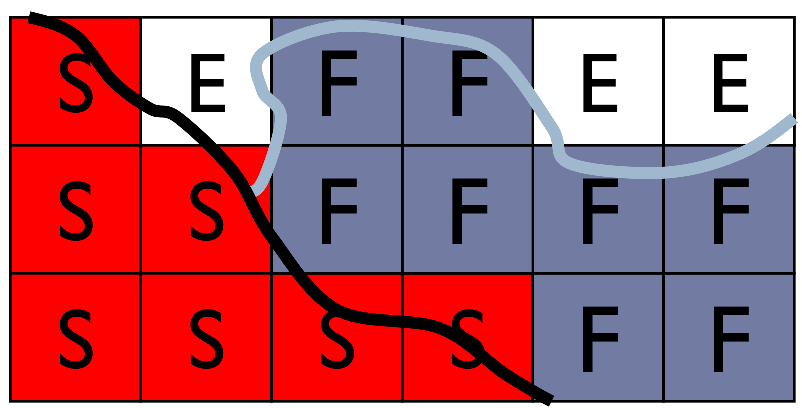
$$\mathbf{A}p = d$$

- A is a huge matrix. In a NxNxN grid, A is a N^3xN^3 matrix.
  - 100x100x100 grid results in a matrix with 10^{12} elements.
- A is sparse.

Boundary Conditions

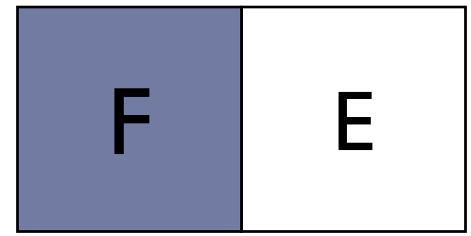
$$\frac{\Delta t}{\rho}\left(\frac{\begin{array}{r}6p_{i,j,k} - p_{i+1,j,k} - p_{i,j+1,k} - p_{i,j,k+1} \\ - p_{i-1,j,k} - p_{i,j-1,k} - p_{i,j,k-1}\end{array}}{\Delta x^2}\right) = -\left(\begin{array}{l}\frac{u_{i+1/2,j,k} - u_{i-1/2,j,k}}{\Delta x} + \frac{v_{i,j+1/2,k} - v_{i,j-1/2,k}}{\Delta x} \\ \qquad\qquad\qquad\quad + \frac{w_{i,j,k+1/2} - w_{i,j,k-1/2}}{\Delta x}\end{array}\right)$$

- At cell(i,j,k), the pressures from its 6 neighboring cells are needed. What if the its neighboring cell is not Fluid?
- At boundary cells, some modifications on the linear equation are needed.
- Each cell is either Fluid, Solid, or Empty. Since water is moving, thus the property of a cell may change (From a Empty to Fluid, or opposite) during simulating.



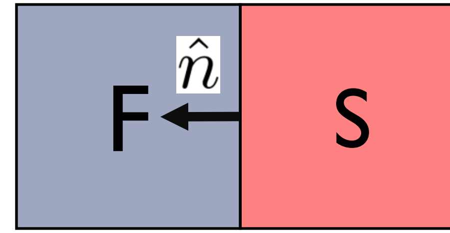
Empty Cell

- Assume the pressure of the Empty Cell is zero
- For example, if cell(i+1, j, k) is empty, then the linear equation at cell(i, j, k) should be:



$$\frac{\Delta t}{\rho}\left(\frac{\begin{array}{r}6p_{i,j,k} - \overset{0}{\overset{\shortparallel}{p_{i+1,j,k}}} - p_{i,j+1,k} - p_{i,j,k+1} \\ - p_{i-1,j,k} - p_{i,j-1,k} - p_{i,j,k-1}\end{array}}{\Delta x^2}\right) = -\left(\begin{array}{r}\dfrac{u_{i+1/2,j,k} - u_{i-1/2,j,k}}{\Delta x} + \dfrac{v_{i,j+1/2,k} - v_{i,j-1/2,k}}{\Delta x} \\ + \dfrac{w_{i,j,k+1/2} - w_{i,j,k-1/2}}{\Delta x}\end{array}\right)$$

Solid Cell



- The assumption
  - The water do not penetrate the solid, thus

$$\vec{u} \cdot \hat{n} = \vec{u}_{\text{solid}} \cdot \hat{n}$$

- If right neighboring cell (i+1,j,k) of cell(i, j,k) is Solid,

$$u^{n+1}_{i+1/2,j} = u_{i+1/2,j} - \Delta t \frac{1}{\rho} \frac{p_{i+1,j} - p_{i,j}}{\Delta x}$$

$$\vec{u}^{\,n+1} = \vec{u}^{\,*} - \frac{\Delta t}{\rho} \nabla p$$

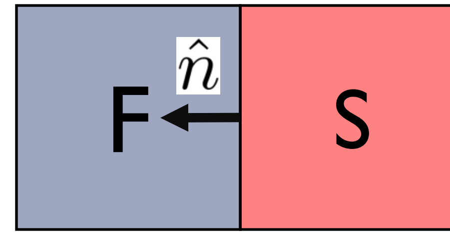
$$u^{n+1}_{i+1/2,j} = u_{\text{solid}}$$

$$(\vec{u}_{\text{solid}} \cdot \hat{n})$$

$$p_{i+1,j} = p_{i,j} + \frac{\rho \Delta x}{\Delta t} \left(u_{i+1/2,j} - u_{\text{solid}} \right)$$

Modification on Laplace Equation

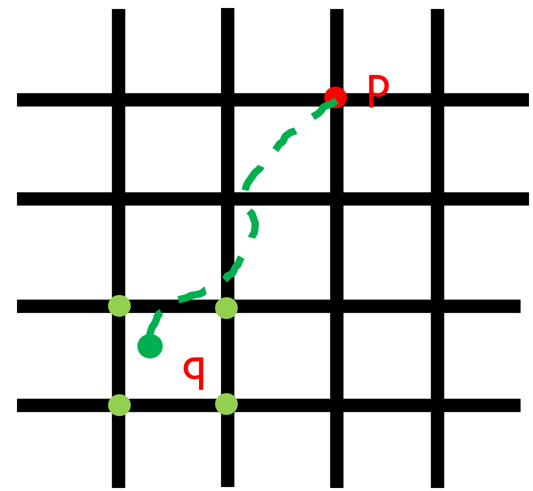
$$p_{i+1,j} = p_{i,j} + \frac{\rho \Delta x}{\Delta t}\left(u_{i+1/2,j} - u_{\text{solid}}\right)$$



$$\frac{\Delta t}{\rho}\left(\frac{\overset{\overset{5}{\shortparallel}}{6}\, p_{i,j,k} - \cancel{p_{i+1,j,k}} - p_{i,j+1,k} - p_{i,j,k+1} - p_{i-1,j,k} - p_{i,j-1,k} - p_{i,j,k-1}}{\Delta x^2}\right) = -\left(\frac{\overset{\overset{u_{\text{solid}}}{\shortparallel}}{u_{i+1/2,j,k}} - u_{i-1/2,j,k}}{\Delta x} + \frac{v_{i,j+1/2,k} - v_{i,j-1/2,k}}{\Delta x} + \frac{w_{i,j,k+1/2} - w_{i,j,k-1/2}}{\Delta x}\right)$$

Summary

$$\frac{\partial \vec{u}}{\partial t} + \underbrace{\vec{u} \cdot \nabla \vec{u}}_{f_1(\vec{u})} + \underbrace{\frac{1}{\rho} \nabla p}_{f_3(\vec{u})} = \underbrace{\vec{g}}_{f_2(\vec{u})}$$



- $\frac{D\vec{u}}{Dt} = 0$ Advection/Transportation
- $\frac{\partial \vec{u}}{\partial t} = \vec{g}$ Body Forces
- $\frac{\partial \vec{u}}{\partial t} + \frac{1}{\rho} \nabla p = 0$ Pressure/Incompressiblity

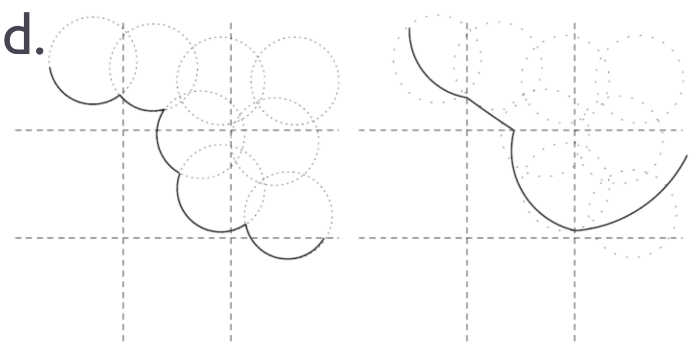
  s.t. $\nabla \cdot \vec{u} = 0$

$$\vec{u}^{*} = \vec{u}^{advected} + \Delta t \vec{g}$$

$$\boxed{\begin{gathered}\mathbf{A}p = d \\ \vec{u}^{n+1} = \vec{u}^{*} - \frac{\Delta t}{\rho} \nabla p\end{gathered}}$$

Where is the Water?

- Marker particles
  - initially each Fluid cell in the MAC grid will be allocated N particles (e.g. N = 4).
  - Move particles in the incompressible velocity field
  - update cell properties (the cell contains any particles is marked as Fluid).
- From particles to Water surface
  - Implicit function: f(x) = distance to the nearest particle – r
  - Sample f(x) with a high resolution grid.
  - Marching Cube to find Iso-surface



Water and Level Sets

- Represent the surface using an implicit function

$$\{\vec{x} \quad | \quad \phi(\vec{x}) = 0\}$$

  - One popular choice: Signed Distance Function
    - Distance to the nearest point on the surface
    - Positive outside, negative inside.
    - Some nice properties:

$$\nabla\phi \cdot \hat{n} = 1$$

$$\hat{n} = \nabla\phi$$

- Evolution of this function: advection

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

  - Above properties may not be preserved. Periodically recalculate the distance function.

Reference

- More details on Level Sets:
  - Book: "Level Set Methods and Dynamic Implicit Surface" by Stanley Osher and Ronald Fedkiw.
- A nice overview on Fluid Simulation
  - SIGGRAPH 2007 Course Notes, "Fluid Simulation" by Robert Bridson and Matthias Muller-Fischer.
- Libraries to Solve Sparse Linear System
  - SparseLib: http://math.nist.gov/sparselib++/
  - PARDISO or IML (Intel Mathematic Library).
- Surface Reconstruction
  - Marching Cube, http://local.wasp.uwa.edu.au/~pbourke/geometry/polygonise/
  - Poisson Surface Reconstruction, http://www.cs.jhu.edu/~misha/

Thanks!