GAMES103: Intro to Physics-Based Animation

Eulerian Fluids
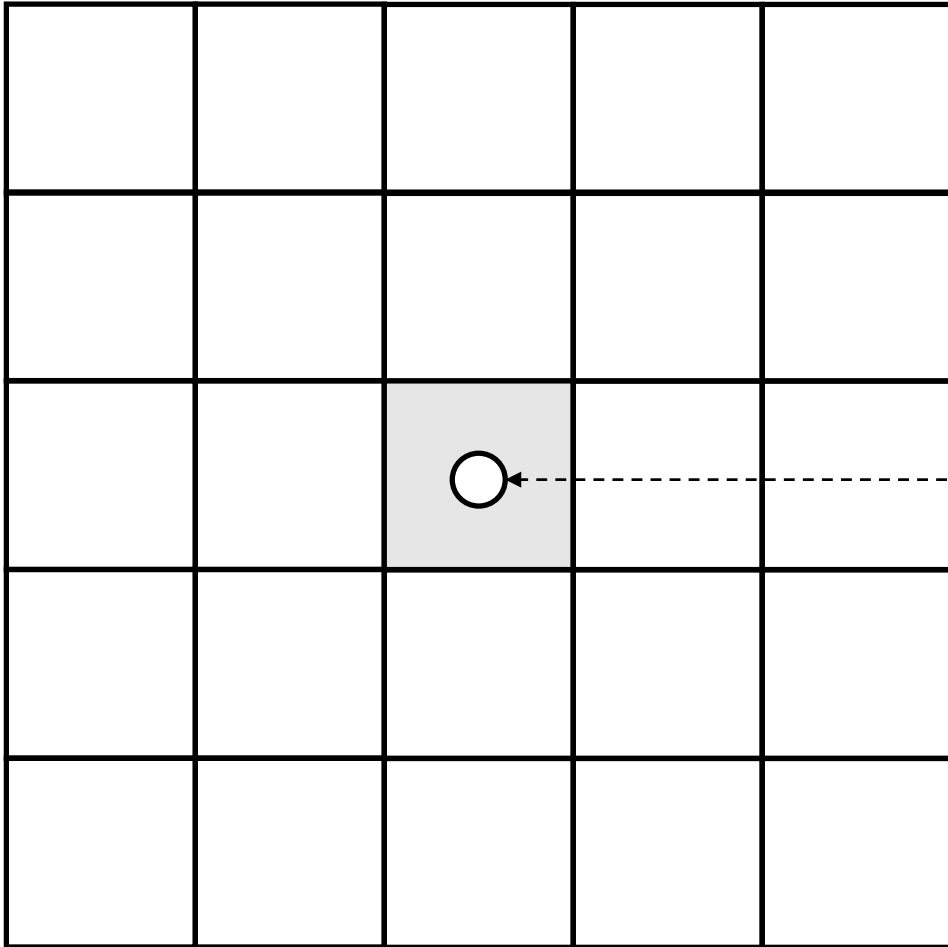
Huamin Wang

Dec 2021

Topics for the Day

- A Grid Representation and Finite Differencing

- Incompressible, Viscous Navier Stokes' equations

- Air and liquid

A Grid Representation and Finite Differencing

A Regular Grid Representation

In an Eulerian grid representation, we store physical quantities over a grid. The most popular grid is a regular one.

- Scalars
  - Density/color
  - Pressure
  - Temperature
  - …
- Vectors
  - Velocities
- The grid can be viewed as a scalar/vector function, also known as a scalar/vector field.

Central Differencing

Recall that we can use central differencing to estimate the derivative.

$$f(t_0 - \Delta t) \qquad f(t_0) \qquad f(t_0 + \Delta t)$$

$$t_0 - \Delta t \qquad t_0 \qquad t_0 + \Delta t$$
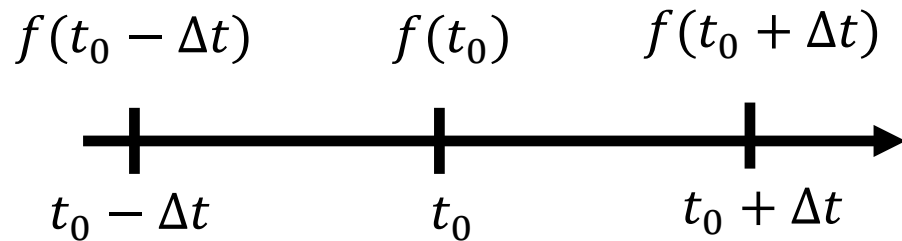
$$f(t_0 + \Delta t) = f(t_0) + \Delta t \frac{df(t_0)}{dt} + \frac{\Delta t^2}{2}\frac{d^2 f(t_0)}{dt^2} + \cdots$$
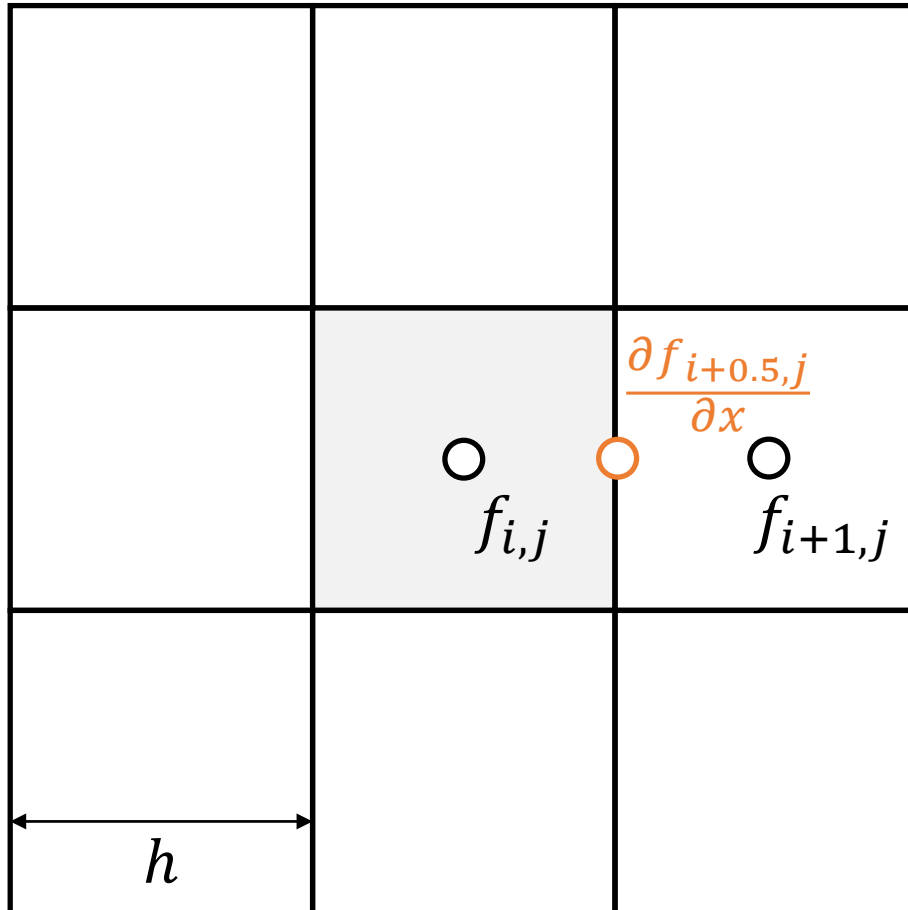
$$f(t_0 - \Delta t) = f(t_0) - \Delta t \frac{df(t_0)}{dt} + \frac{\Delta t^2}{2}\frac{d^2 f(t_0)}{dt^2} + \cdots$$

Central differencing (second-order)

$$\frac{df(t_0)}{dt} \approx \frac{f(t_0 + \Delta t) - f(t_0 - \Delta t)}{2\Delta t}$$
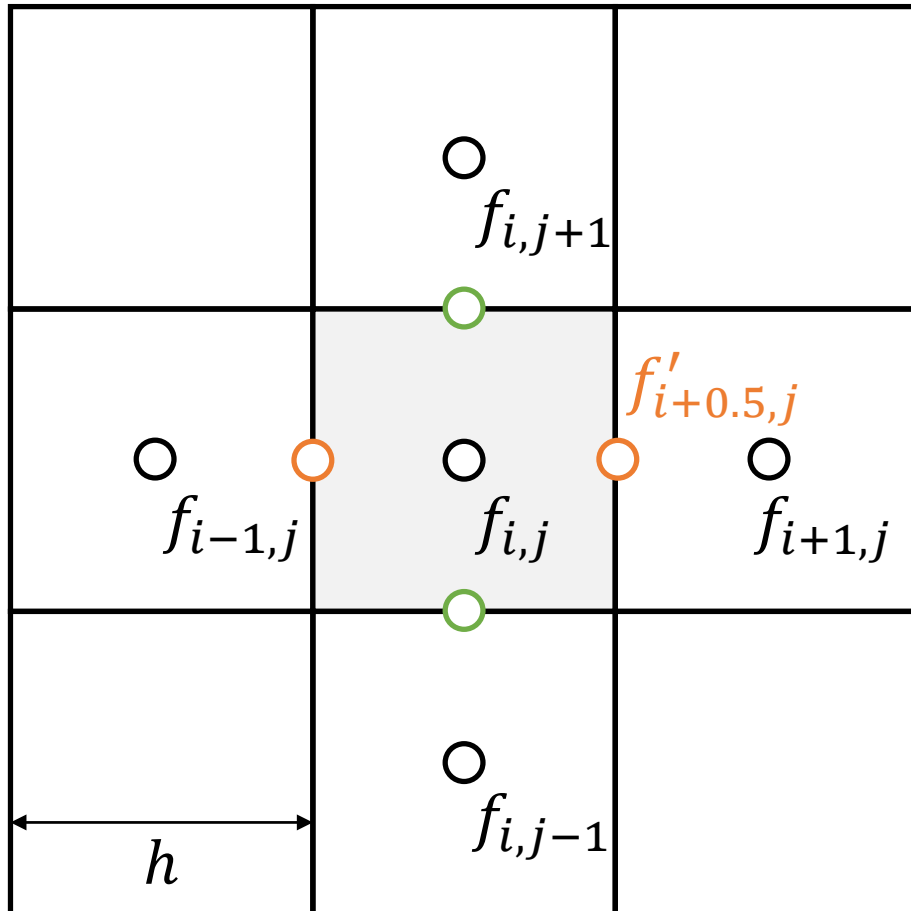
Finite Differencing on Grid

The grid is very friendly with central differencing.

$$\frac{\partial f_{i+0.5,j}}{\partial x} \approx \frac{f_{i+1,j} - f_{i,j}}{h}$$

Finite Differencing on Grid
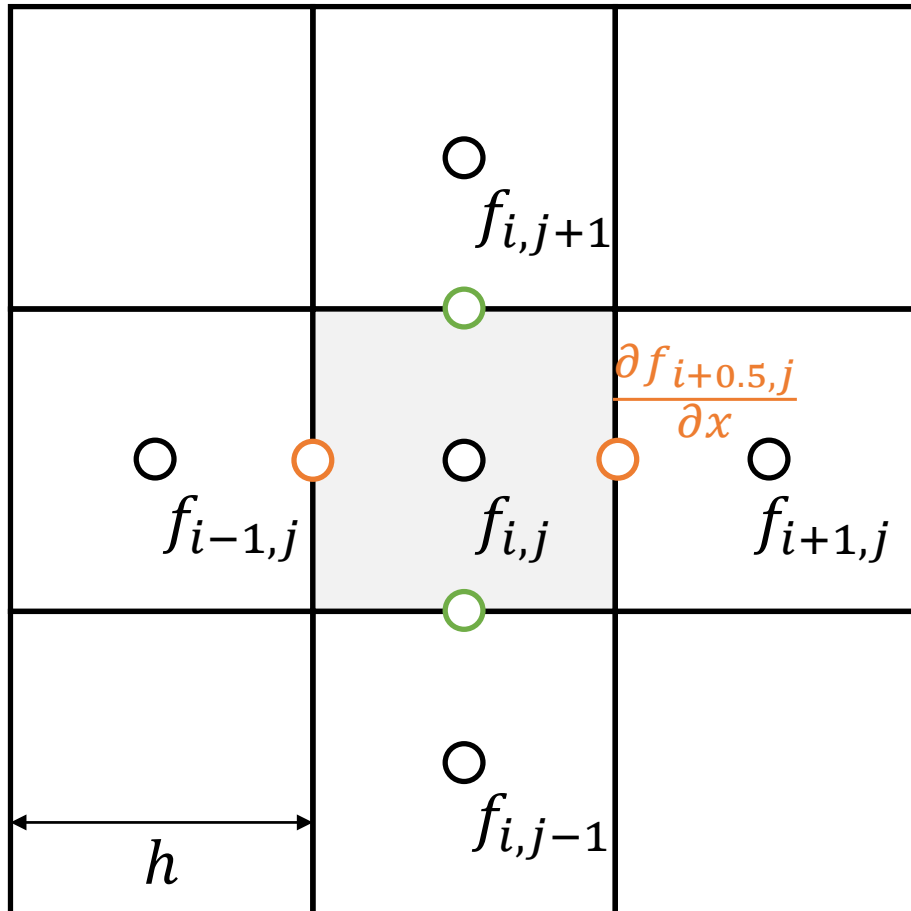
The grid is very friendly with central differencing.

$$\frac{\partial f_{i-0.5,j}}{\partial x} \approx \frac{f_{i,j} - f_{i-1,j}}{h}$$

$$\frac{\partial f_{i+0.5,j}}{\partial x} \approx \frac{f_{i+1,j} - f_{i,j}}{h}$$

$$\frac{\partial^2 f_{i,j}}{\partial x^2} \approx \frac{\frac{\partial f_{i+0.5,j}}{\partial x} - \frac{\partial f_{i-0.5,j}}{\partial x}}{h}$$

$$\approx \frac{f_{i-1,j} + f_{i+1,j} - 2f_{i,j}}{h^2}$$

$$\frac{\partial f_{i,j-0.5}}{\partial y} \approx \frac{f_{i,j} - f_{i,j-1}}{h}$$

$$\frac{\partial f_{i,j+0.5}}{\partial y} \approx \frac{f_{i,j+1} - f_{i,j}}{h}$$

$$\frac{\partial^2 f_{i,j}}{\partial y^2} \approx \frac{\frac{\partial f_{i,j+0.5}}{\partial y} - \frac{\partial f_{i,j-0.5}}{\partial y}}{h}$$

$$\approx \frac{f_{i,j-1} + f_{i,j+1} - 2f_{i,j}}{h^2}$$

Discretized Laplacian

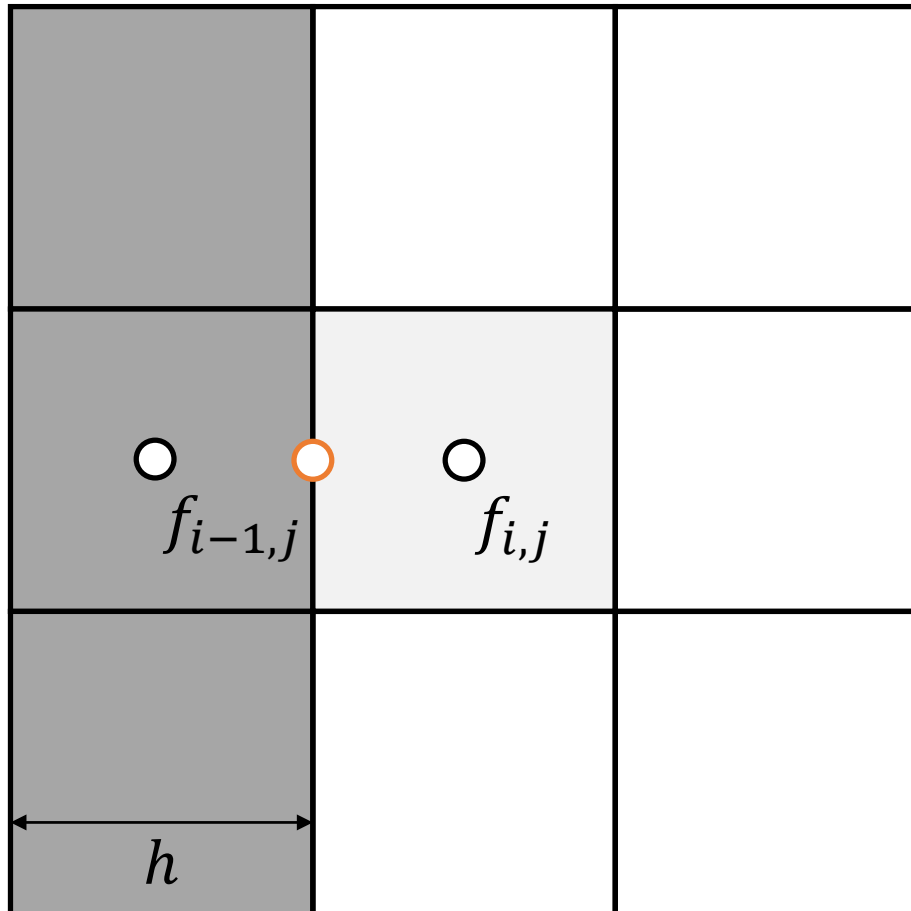We can then obtain the discretized Laplacian operator on grid.

$$\frac{\partial^2 f_{i,j}}{\partial x^2} \approx \frac{\frac{\partial f_{i-0.5,j}}{\partial x} - \frac{\partial f_{i+0.5,j}}{\partial x}}{h} \approx \frac{f_{i-1,j} + f_{i+1,j} - 2f_{i,j}}{h^2}$$

$$\frac{\partial^2 f_{i,j}}{\partial y^2} \approx \frac{\frac{\partial f_{i,j+0.5}}{\partial y} - \frac{\partial f_{i,j-0.5}}{\partial y}}{h} \approx \frac{f_{i,j-1} + f_{i,j+1} - 2f_{i,j}}{h^2}$$

$$\Delta f_{i,j} = \frac{\partial^2 f_{i,j}}{\partial x^2} + \frac{\partial^2 f_{i,j}}{\partial y^2} \approx \frac{f_{i-1,j} + f_{i+1,j} + f_{i,j-1} + f_{i,j+1} - 4f_{i,j}}{h^2}$$

Boundary Conditions

The boundary condition specifies $f_{i-1,j}$ if it's outside.
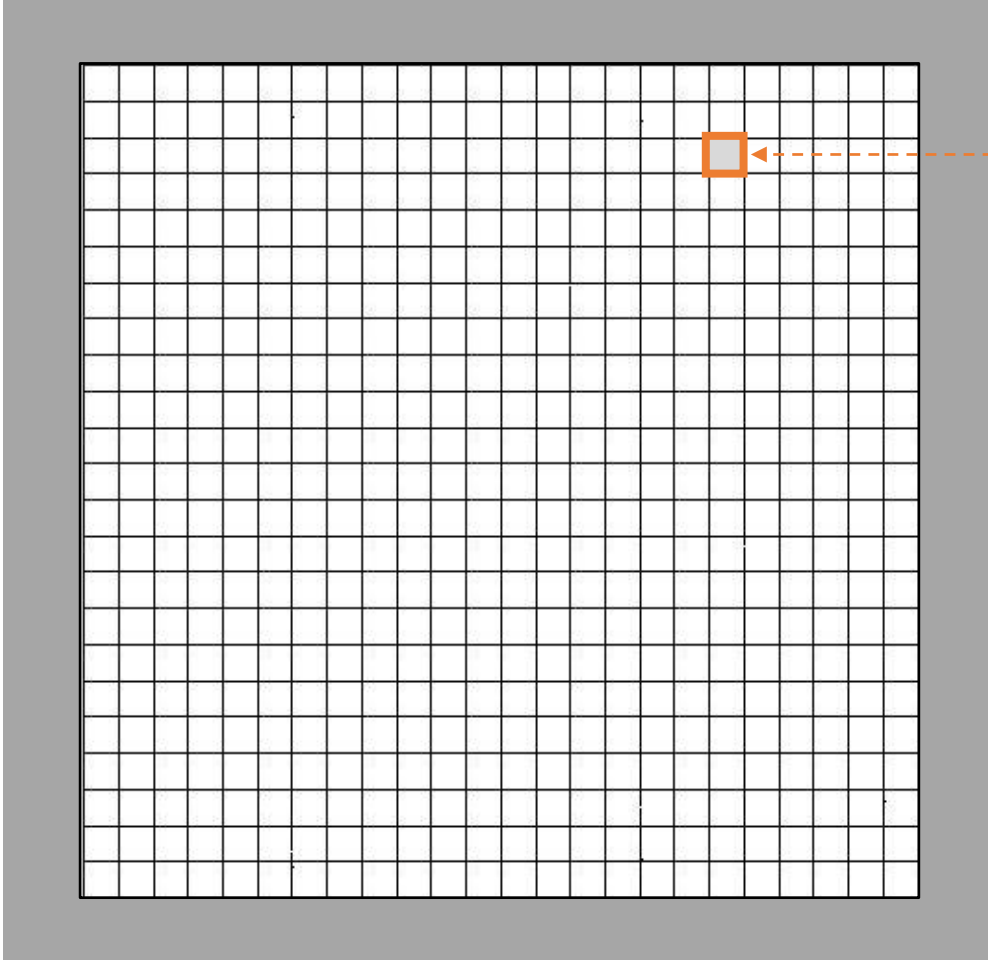

A Dirichlet boundary: $f_{i-1,j} = C$

$$\Delta f_{i,j} \approx \frac{C + f_{i+1,j} + f_{i,j-1} + f_{i,j+1} - 4f_{i,j}}{h^2}$$

A Neumann boundary: $f_{i-1,j} = f_{i,j}$

$$\Delta f_{i,j} \approx \frac{f_{i+1,j} + f_{i,j-1} + f_{i,j+1} - 3f_{i,j}}{h^2}$$

Example: Laplace's Equation

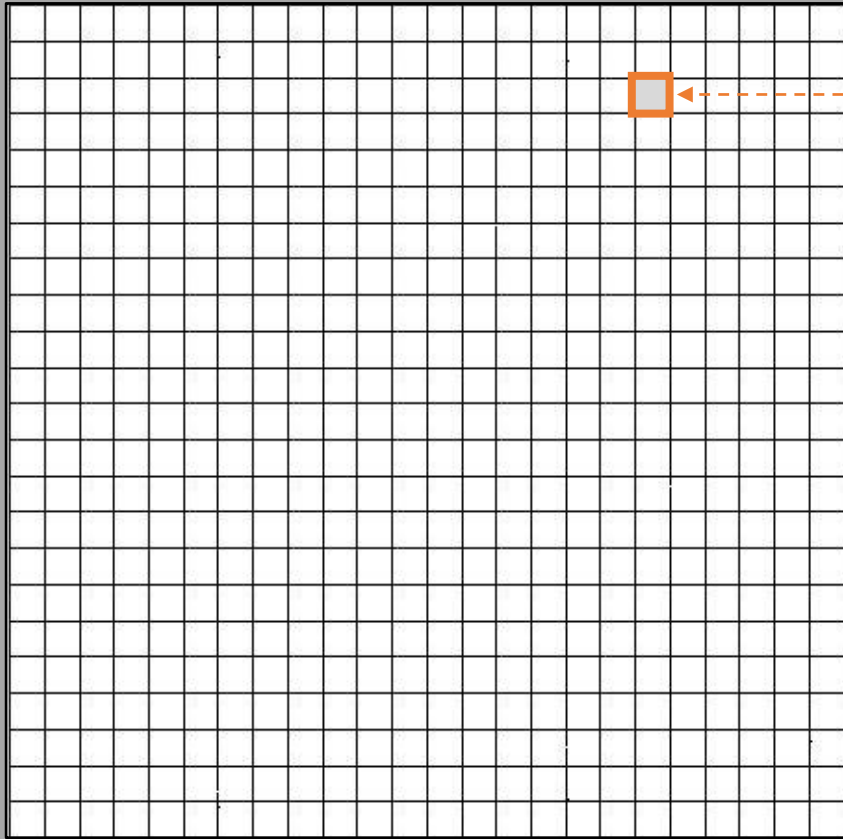With the grid, we can discretize Laplace's equation: $\Delta f = 0$.

$$f_{i-1,j} + f_{i+1,j} + f_{i,j-1} + f_{i,j+1} - 4f_{i,j} = 0$$

with boundary conditions

Note: At least one condition should be Neumann. (Why?)

Example: Laplace's Equation

To solve Laplace's equation, we can use the Jacobi method.

$$f_{i-1,j} + f_{i+1,j} + f_{i,j-1} + f_{i,j+1} - 4f_{i,j} = 0$$

The Jacobi Method

Initialize f

For $k = 0 \dots K$

    For i, j

$$f_{ij}^{\mathrm{new}} \leftarrow f_{ij} + \alpha\big(4f_{ij} - f_{i-1,j} - f_{i+1,j} - f_{i,j-1} - f_{i,j+1}\big)$$

    For i, j

$$f_{ij} \leftarrow f_{ij}^{\mathrm{new}}$$

Laplacian Smoothing

Diffusion

The process of applying Laplacian smoothing is called diffusion. See in-class demo.

Problem with Central Differencing

Central differencing gives the derivative in the middle.
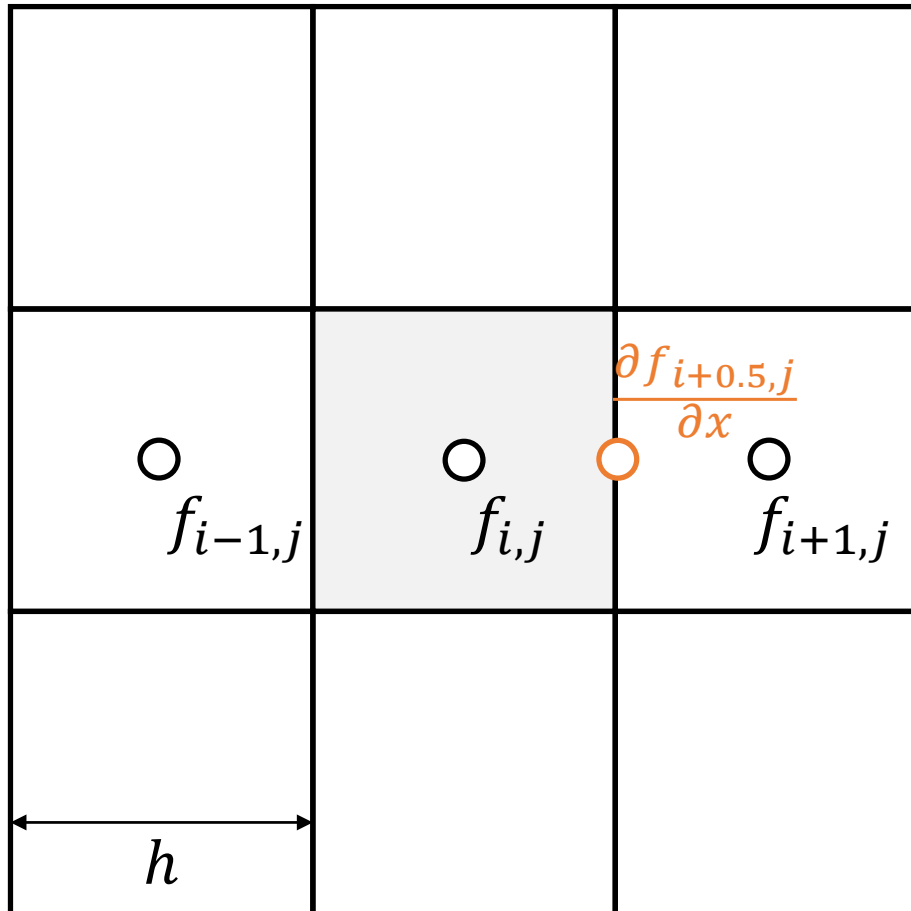
- The cell doesn't exist at (i+0.5, j).

- To get $\frac{\partial f_{i,j}}{\partial x}$, we need $f_{i-1,j}$ and $f_{i+1,j}$. But this is weird, because $f_{i,j}$ is unused.

Solution: Staggered Grid

We define some physical quantities on faces, specifically velocities.
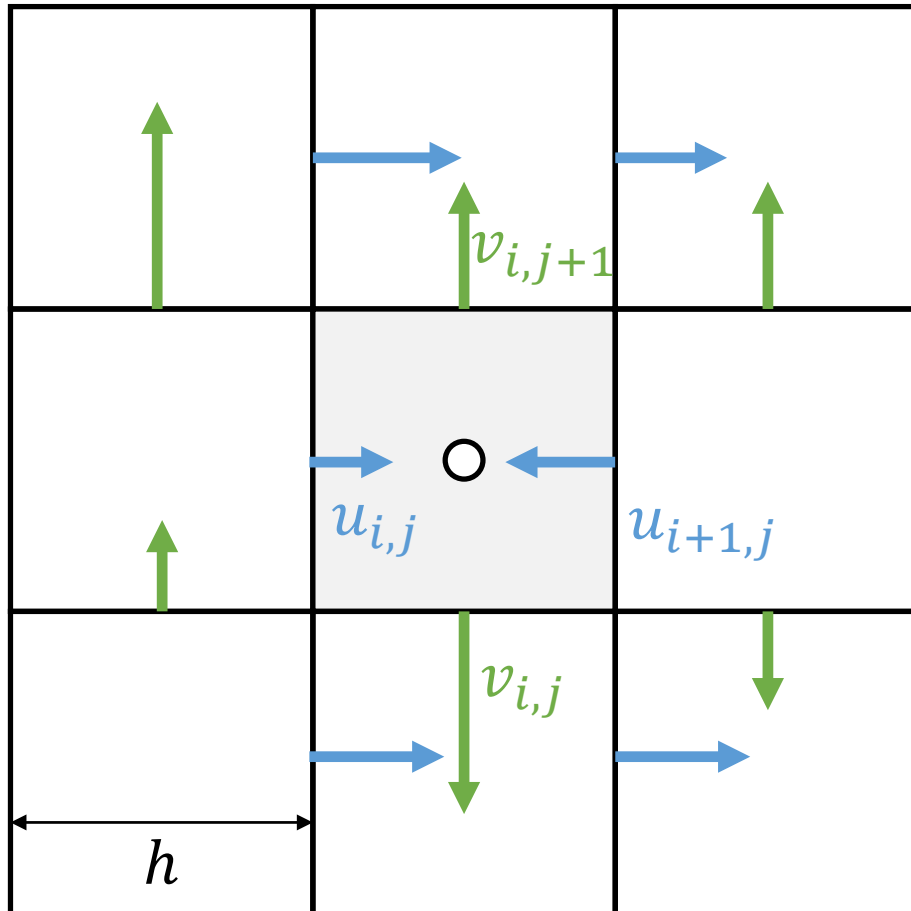
- The x-part of the velocity is defined on vertical faces.

- The y-part of the velocity is defined on horizonal faces.

- Intuitively, they represent the flow speed between two cells. For example, we write the volume changing speed at cell (i, j) as:

$$u_{i+1,j} + v_{i,j+1} - u_{i,j} - v_{i,j}$$

Divergence-Free Condition

No volume change is equal to say the fluid is incompressible. This can be formally written as a divergence-free velocity field.
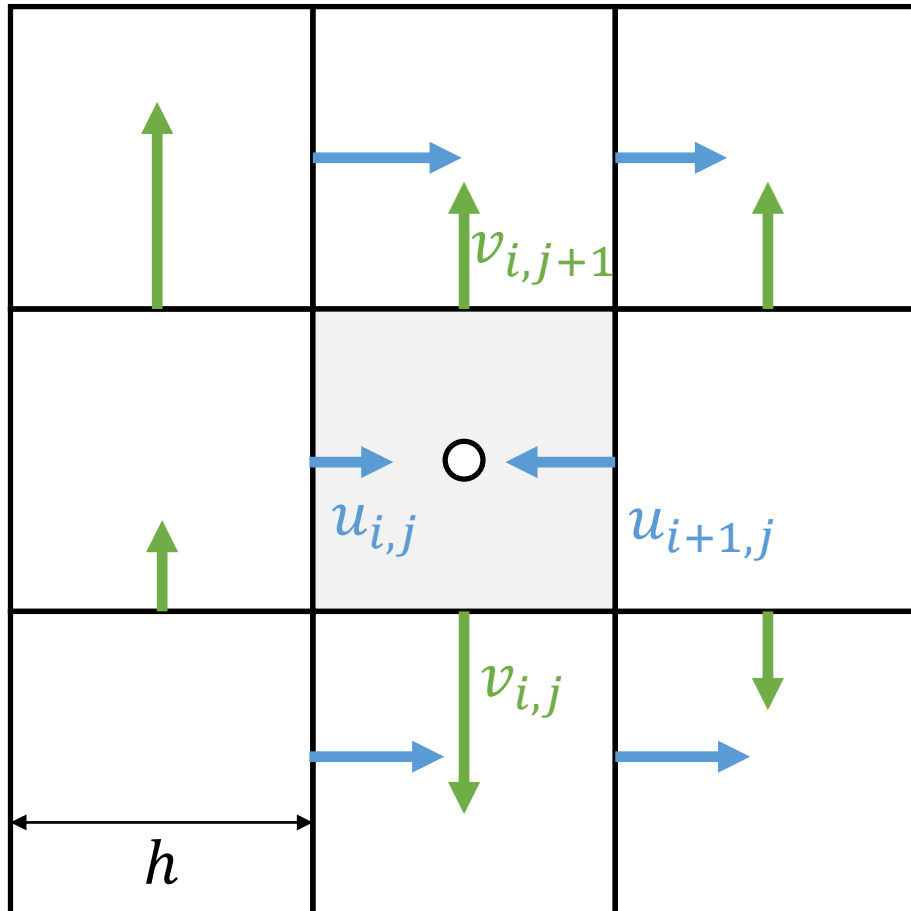
$$u_{i+1,j} + v_{i,j+1} - u_{i,j} - v_{i,j} = 0$$
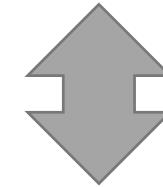
$$\nabla \cdot \mathbf{u}_{i,j} = \frac{\partial u_{i,j}}{\partial x} + \frac{\partial v_{i,j}}{\partial y} \approx \frac{u_{i+1,j} - u_{i,j}}{h} + \frac{v_{i,j+1} - v_{i,j}}{h} = 0$$

velocity field

Bilinear Interpolation

We use bilinear interpolation to interpolate physical quantities.

$$i \leftarrow \lfloor x \rfloor$$

$$j \leftarrow \lfloor y \rfloor$$
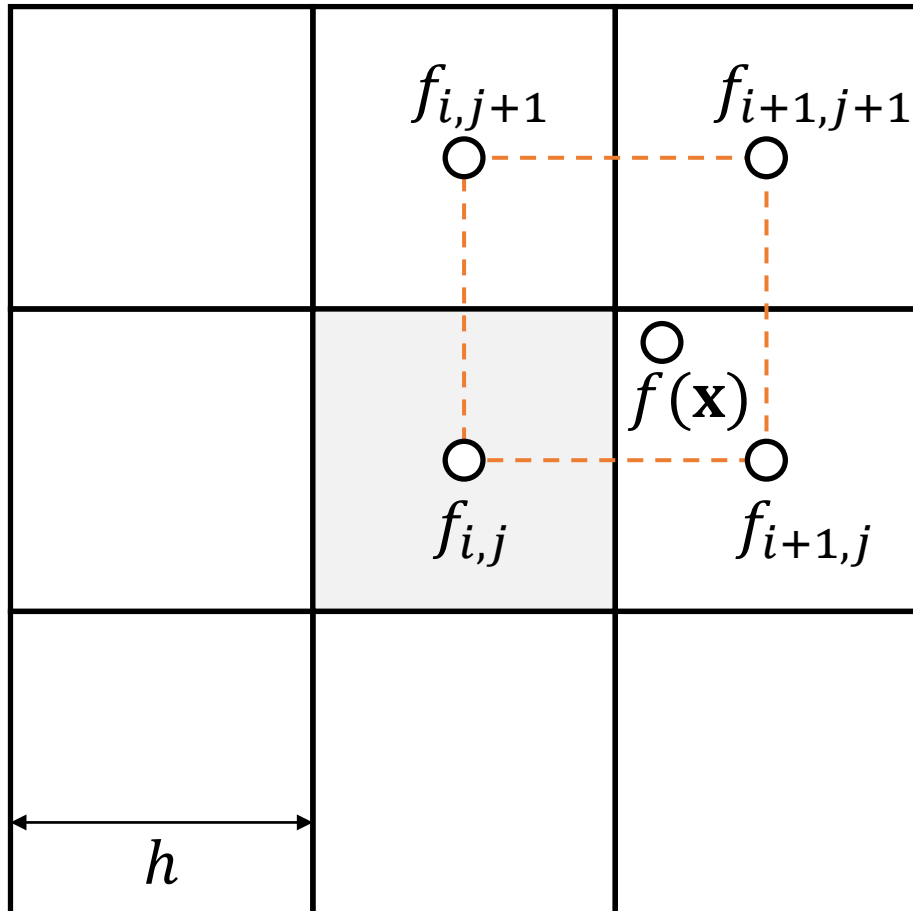
$$f(\mathbf{x}) \leftarrow (i + 1 - x)(j + 1 - y)f_{i,j}$$
$$+ (x - i)(j + 1 - y)f_{i+1,j}$$
$$+ (i + 1 - x)(y - j)f_{i,j+1}$$
$$+ (x - i)(y - j)f_{i+1,j+1}$$

Bilinear Interpolation

We use bilinear interpolation to interpolate staggered velocities as well.
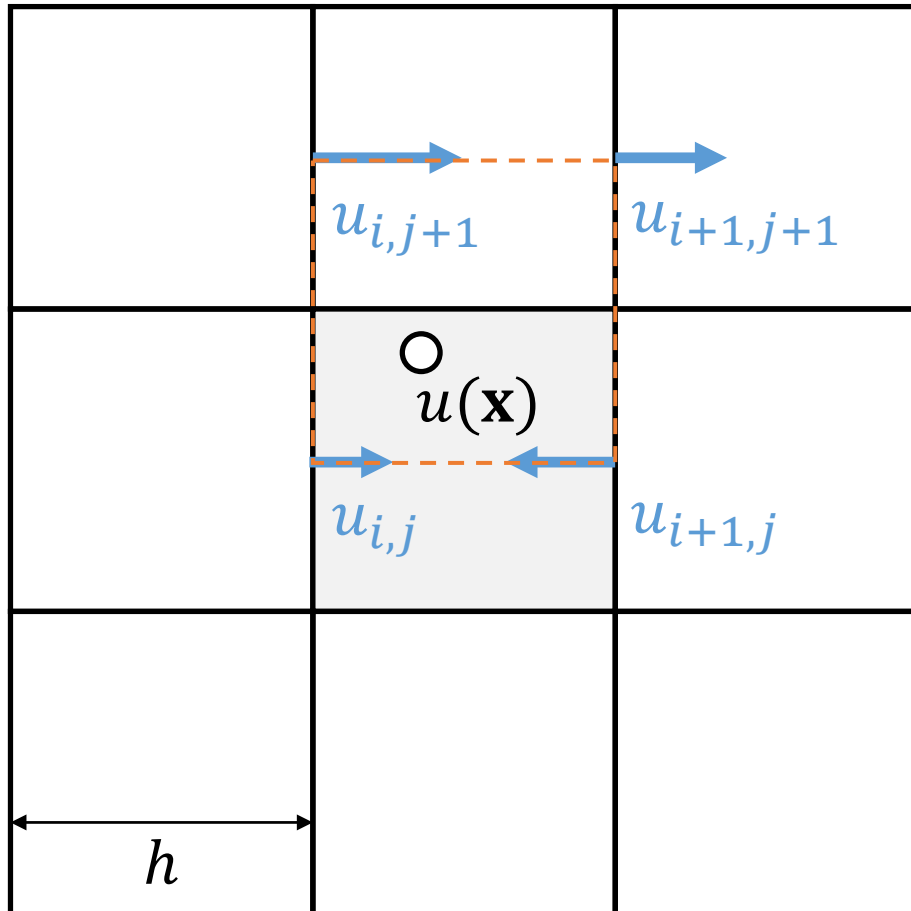
$$x \leftarrow x - 0.5$$
$$i \leftarrow \lfloor x \rfloor$$
$$j \leftarrow \lfloor y \rfloor$$

$$u(\mathbf{x}) \leftarrow (i + 1 - x)(j + 1 - y)u_{i,j}$$
$$+(x - i)(j + 1 - y)u_{i+1,j}$$
$$+(i + 1 - x)(y - j)u_{i,j+1}$$
$$+(x - i)(y - j)u_{i+1,j+1}$$

Incompressible, Viscous Navier-Stokes Equations

Equation Fomulation

Incompressibility

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

Momentum

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{g} - \overbrace{(\mathbf{u} \cdot \nabla)\mathbf{u}}^{\text{advection}} + \underbrace{\mu \Delta \mathbf{u}}_{\text{(Laplacian) diffusion}} - \overbrace{\nabla p}^{\text{Internal pressure}}$$

external acceleration

- Method of Characteristics: solving a long partial differential equation (PDE) in steps
- Step 1: Update $\mathbf{u}$ by solving $\partial \mathbf{u} / \partial t = \mathbf{g}$
- Step 2: Update $\mathbf{u}$ by solving $\partial \mathbf{u} / \partial t = -(\mathbf{u} \cdot \nabla)\mathbf{u}$
- Step 3: Update $\mathbf{u}$ by solving $\partial \mathbf{u} / \partial t = v \Delta \mathbf{u}$
- Step 4: Update $\mathbf{u}$ by solving $\partial \mathbf{u} / \partial t = -\nabla p$

Step 1: External Acceleration

The Update of $\mathbf{u}$ by $\partial \mathbf{u}/\partial t = \mathbf{g}$ is straightforward, just add acceleration to u and v.
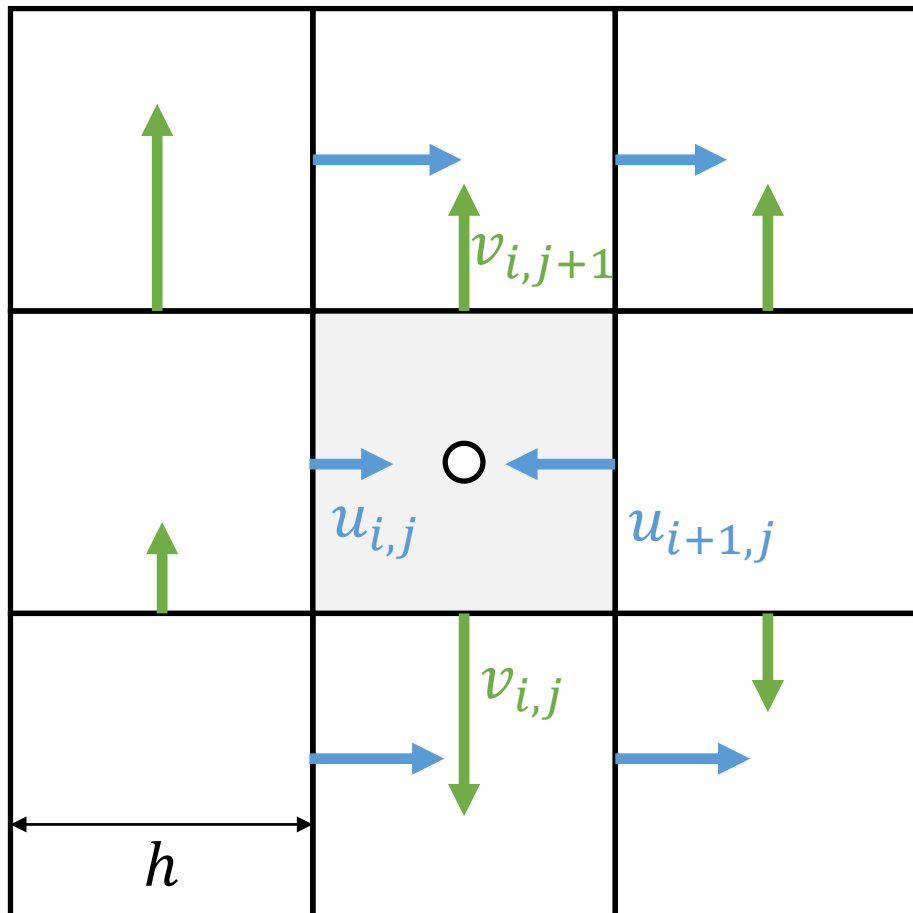
If gravity is the only external acceleration, we get:

$$v_{i,j}^{new} \leftarrow v_{i,j} + \underset{\text{time step}}{\underbrace{\Delta t}} \overset{\text{gravity}}{\overbrace{g}}$$

Step 2: Advection

Next we need to update $\mathbf{u}$ by solving $\partial\mathbf{u}/\partial t = -(\mathbf{u} \cdot \nabla)\mathbf{u}$.
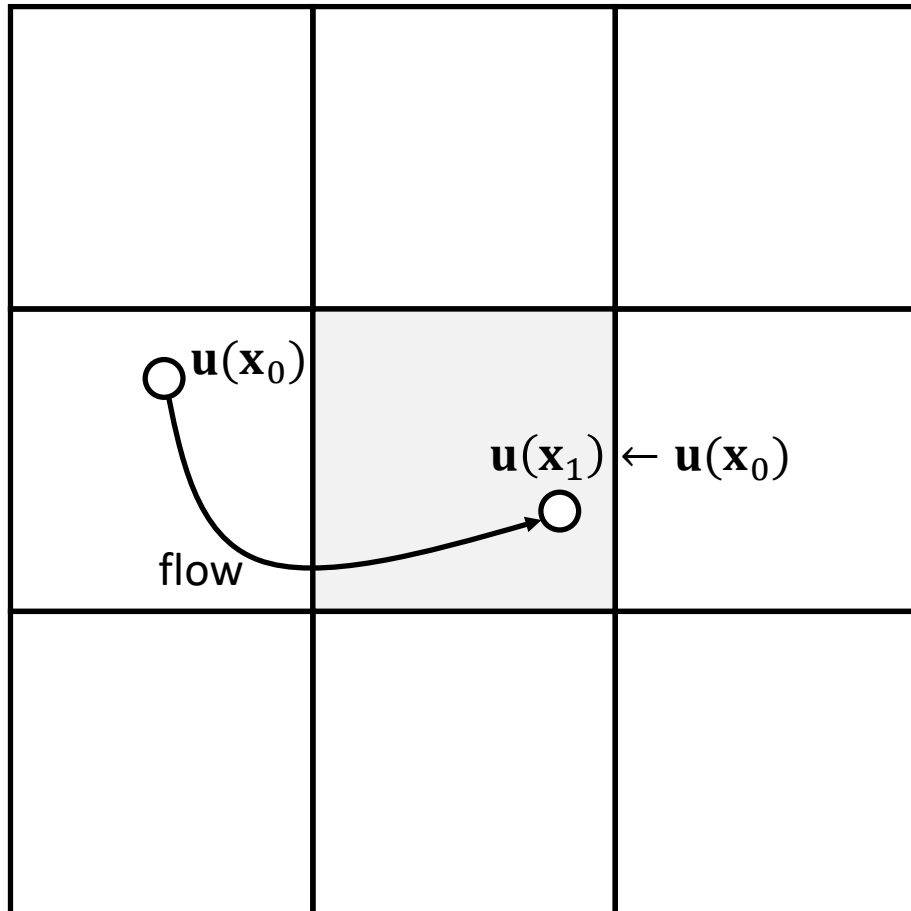
$$(\mathbf{u} \cdot \nabla)\mathbf{u} = \begin{bmatrix} u \cdot \dfrac{\partial u}{\partial x} + v \cdot \dfrac{\partial u}{\partial y} \\ u \cdot \dfrac{\partial v}{\partial x} + v \cdot \dfrac{\partial v}{\partial y} \end{bmatrix}$$

Solving this in an Eulerian way can be a source of instability.

To solve this problem, we come to realize that advection means to carry physical quantities by velocity.

Solution: Semi-Lagrangian Method

The solution is to trace a virtual particle backward over time.
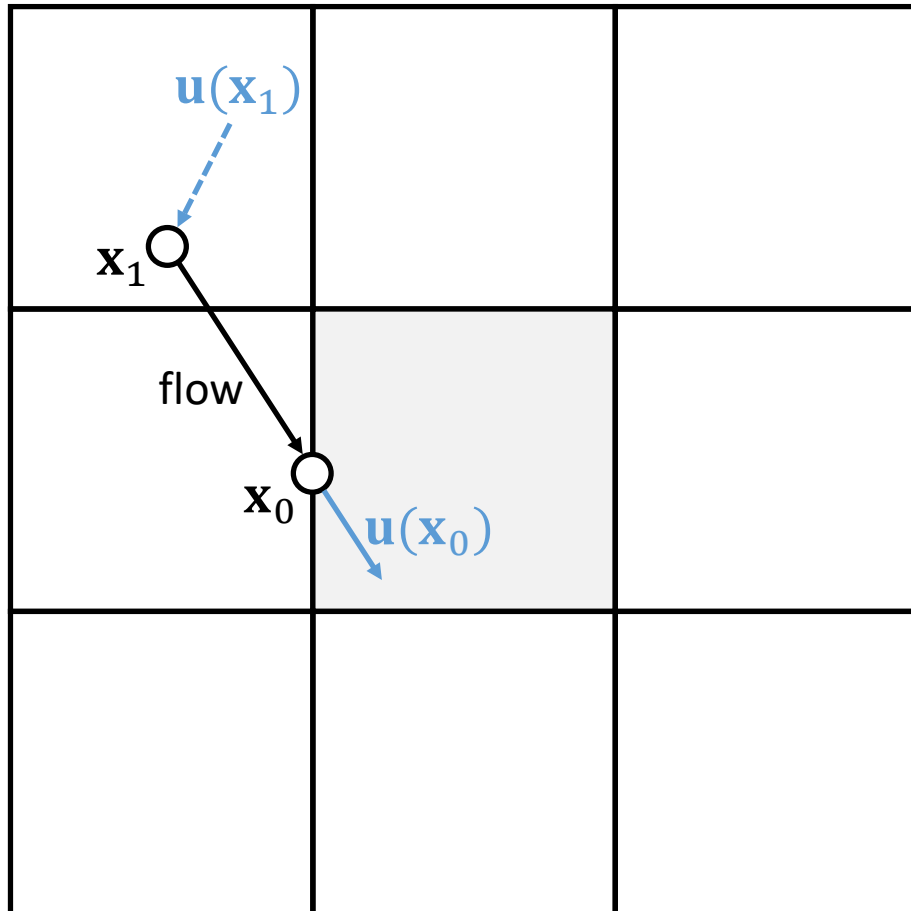
- Define $\mathbf{x}_0 \leftarrow (i - 0.5, j)$
- Compute $\mathbf{u}(\mathbf{x}_0)$
- $\mathbf{x}_1 \leftarrow \mathbf{x}_0 - \Delta t \, \mathbf{u}(\mathbf{x}_0)$
- Compute $\mathbf{u}(\mathbf{x}_1)$
- $u_{i,j}^{new} \leftarrow u(\mathbf{x}_1)$

Note that if the velocities are staggered, we need to do staggered bilinear interpolation.

Solution: Semi-Lagrangian Method

The solution is to trace a virtual particle backward over time.
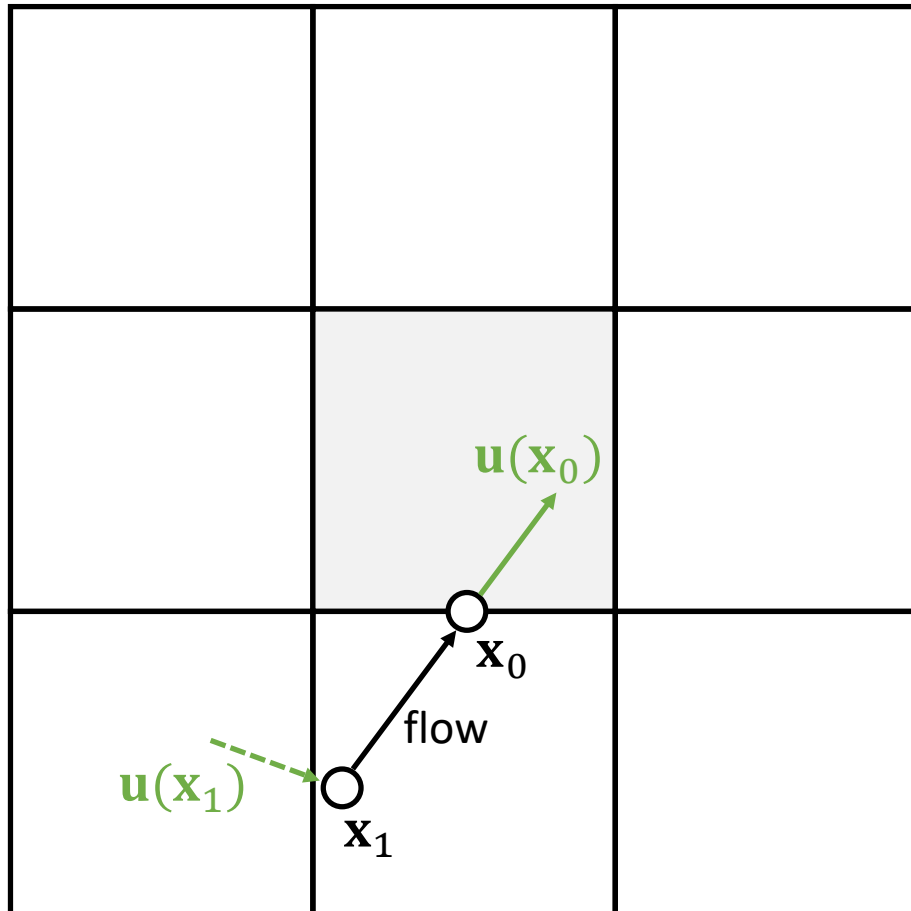
- Define $\mathbf{x}_0 \leftarrow (i, j - 0.5)$
- Compute $\mathbf{u}(\mathbf{x}_0)$
- $\mathbf{x}_1 \leftarrow \mathbf{x}_0 - \Delta t\, \mathbf{u}(\mathbf{x}_0)$
- Compute $\mathbf{u}(\mathbf{x}_1)$
- $v_{i,j}^{new} \leftarrow v(\mathbf{x}_1)$

Solution: Semi-Lagrangian Method

We could also subdivided the time step for better tracing.
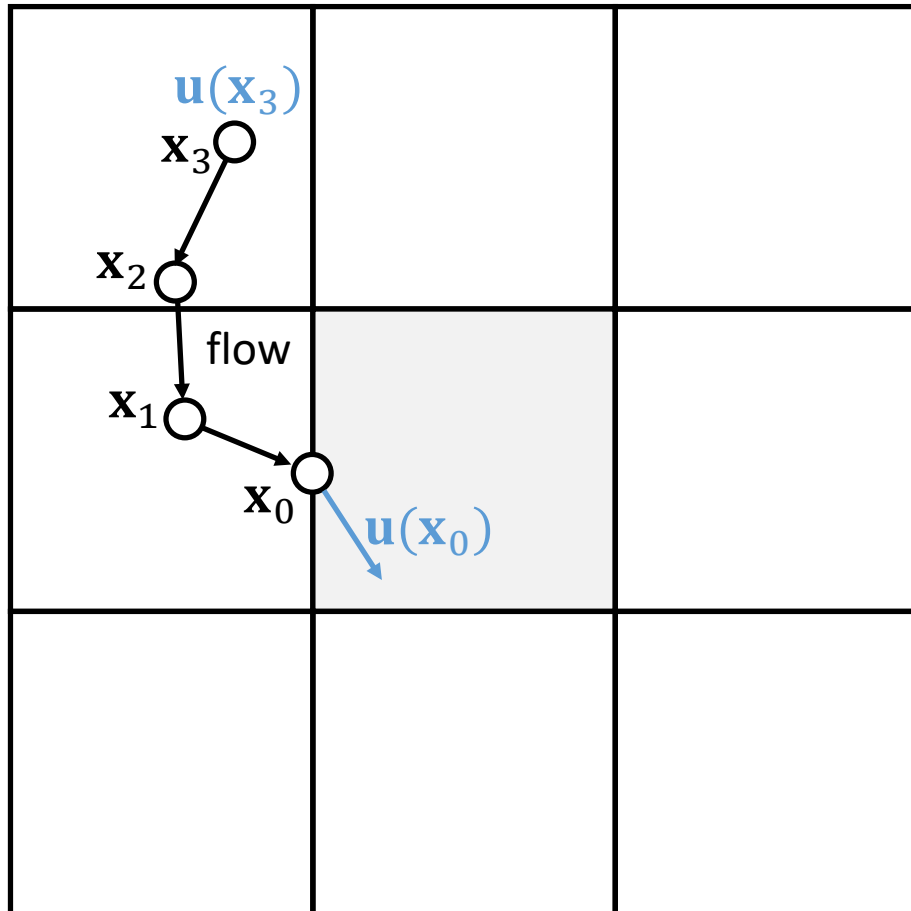
- Define $\mathbf{x}_0 \leftarrow (i - 0.5, j)$
- Compute $\mathbf{u}(\mathbf{x}_0)$
- $\boxed{\mathbf{x}_1 \leftarrow \mathbf{x}_0 - \frac{1}{3}\Delta t\, \mathbf{u}(\mathbf{x}_0)}$
- Compute $\mathbf{u}(\mathbf{x}_1)$
- $\boxed{\mathbf{x}_2 \leftarrow \mathbf{x}_1 - \frac{1}{3}\Delta t\, \mathbf{u}(\mathbf{x}_1)}$
- Compute $\mathbf{u}(\mathbf{x}_2)$
- $\boxed{\mathbf{x}_3 \leftarrow \mathbf{x}_2 - \frac{1}{3}\Delta t\, \mathbf{u}(\mathbf{x}_2)}$
- Compute $\mathbf{u}(\mathbf{x}_3)$
- $u_{i,j}^{new} \leftarrow u(\mathbf{x}_3)$

Step 3: Diffusion

Next we need to update $\mathbf{u}$ by solving $\partial \mathbf{u}/\partial t = v\Delta\mathbf{u}$.
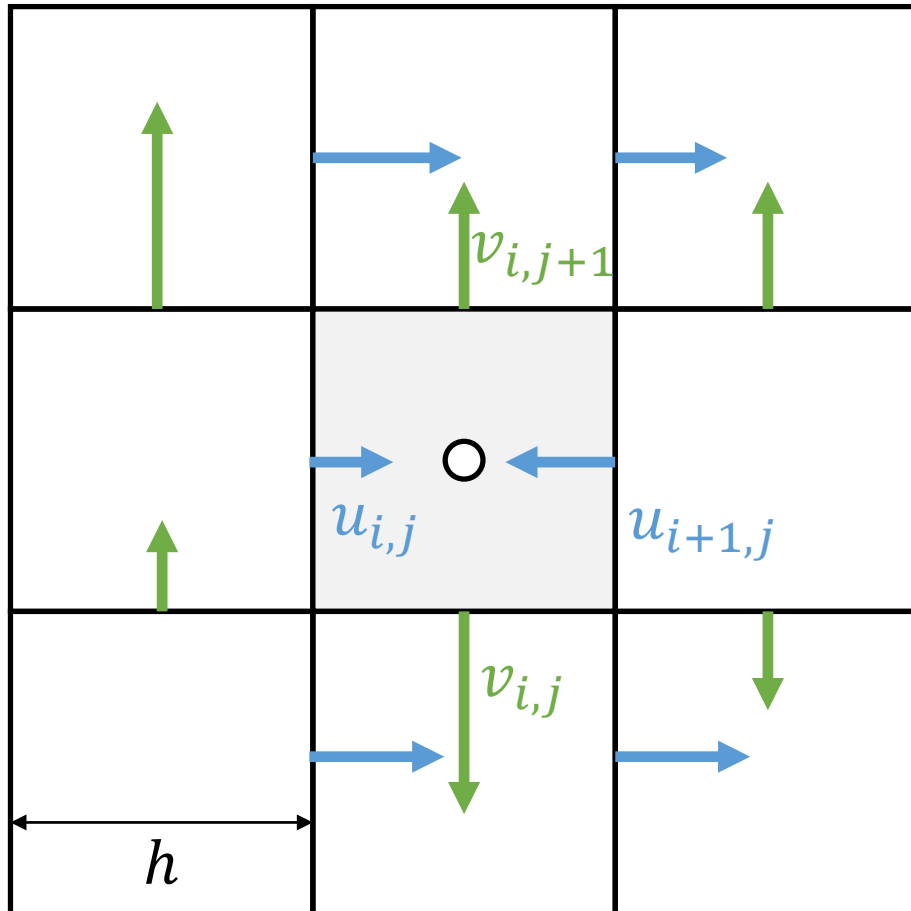
$$u_{i,j}^{new} \leftarrow u_{i,j} + v\Delta t \frac{u_{i-1,j}+u_{i+1,j}+u_{i,j-1}+u_{i,j+1}-4u_{i,j}}{h^2}$$

$$v_{i,j}^{new} \leftarrow v_{i,j} + v\Delta t \frac{v_{i-1,j}+v_{i+1,j}+v_{i,j-1}+v_{i,j+1}-4v_{i,j}}{h^2}$$

If $v\Delta t$ is large, the above formulae can be unstable.

Step 3: Diffusion

Next we need to update $\mathbf{u}$ by solving $\partial \mathbf{u}/\partial t = v \Delta \mathbf{u}$.
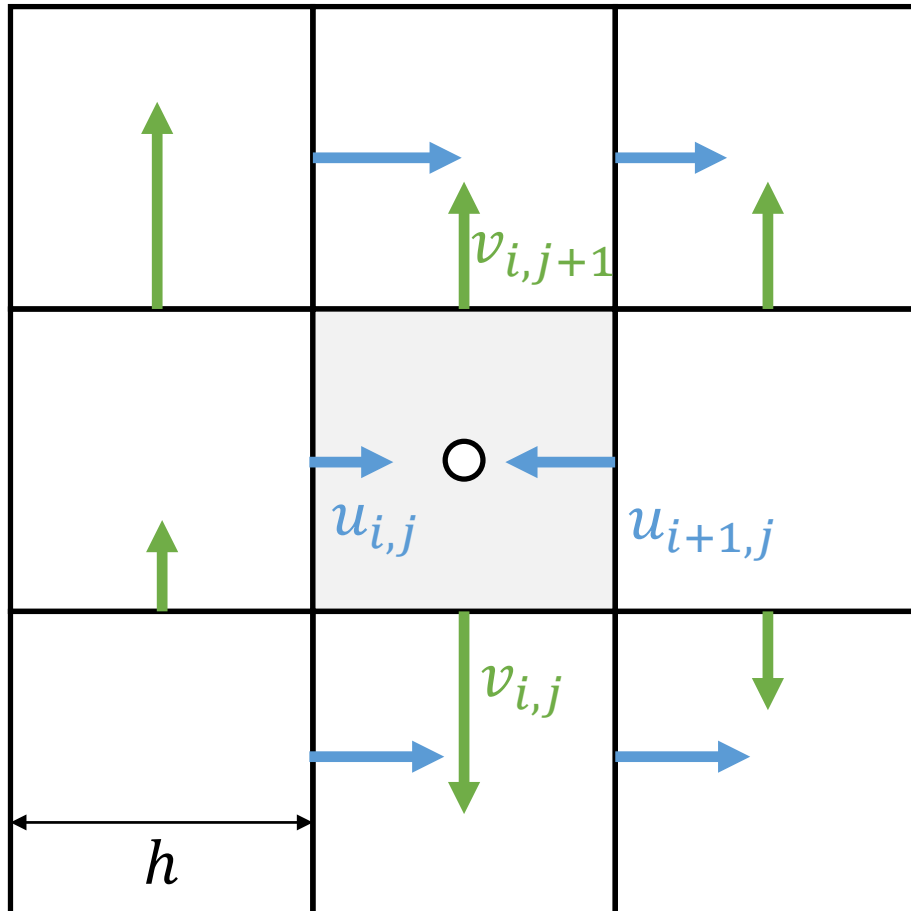
$$u_{i,j}^{temp} \leftarrow u_{i,j} + v \frac{\Delta t}{2} \frac{u_{i-1,j}+u_{i+1,j}+u_{i,j-1}+u_{i,j+1}-4u_{i,j}}{h^2}$$

$$u_{i,j}^{new} \leftarrow u_{i,j}^{temp} + v \frac{\Delta t}{2} \frac{u_{i-1,j}^{temp}+u_{i+1,j}^{temp}+u_{i,j-1}^{temp}+u_{i,j+1}^{temp}-4u_{i,j}^{temp}}{h^2}$$

We could also use even smaller sub-steps…

Step 4: Pressure Projection

Finally, we need to update $\mathbf{u}$ by solving $\partial \mathbf{u} / \partial t = -\nabla \boldsymbol{p}$.
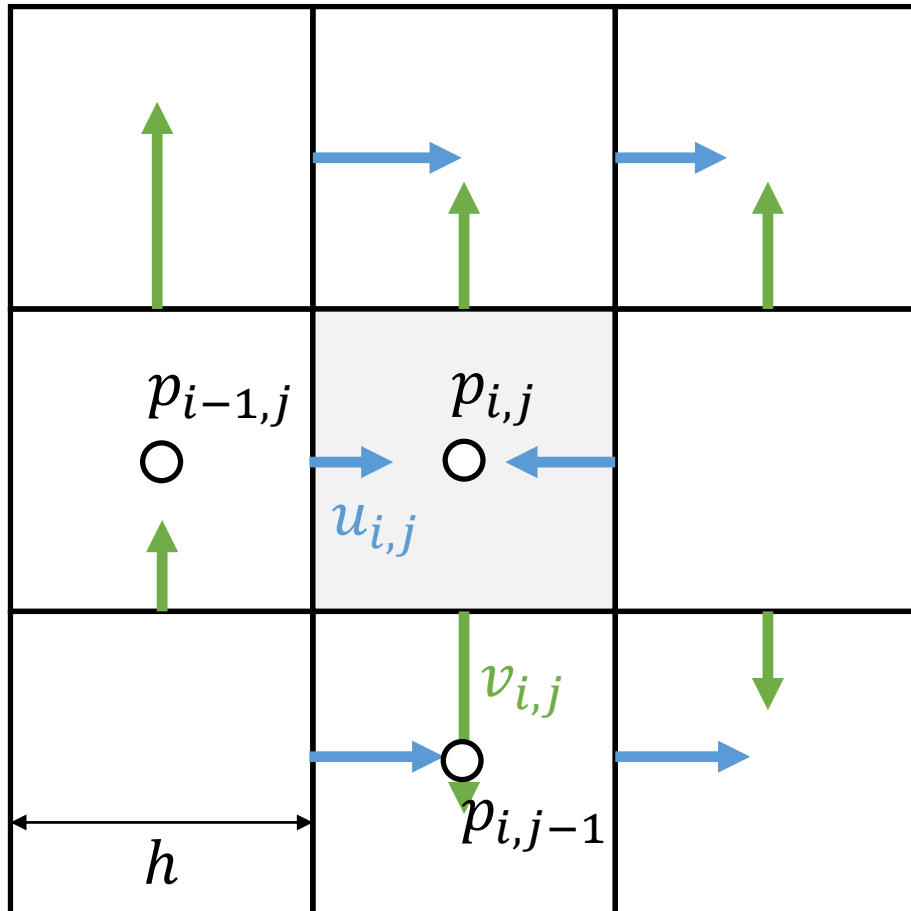
Staggering makes this very straightforward:

$$u_{i,j}^{new} \leftarrow u_{i,j} - \frac{\Delta t}{h}\left(p_{i,j} - p_{i-1,j}\right)$$

$$v_{i,j}^{new} \leftarrow v_{i,j} - \frac{\Delta t}{h}\left(p_{i,j} - p_{i,j-1}\right)$$

But what is $\boldsymbol{p}$?

Step 4: Pressure Projection

The pressure is caused by incompressibility.
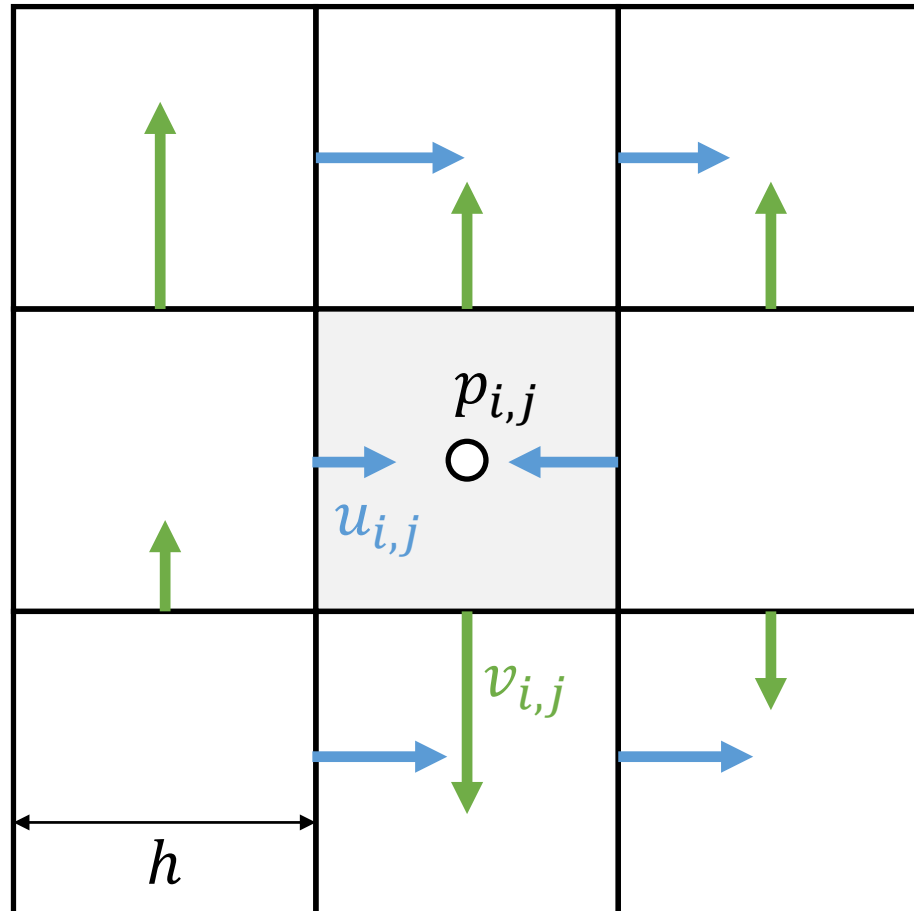
In other words, after this update by pressure, we should achieve:

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

which means

$$u_{i+,j}^{new} + v_{i,j+1}^{new} - u_{i,j}^{new} - v_{i,j}^{new} = 0$$

$$u_{i+1,j} - \frac{\Delta t (p_{i+1,j} - p_{i,j})}{h} + v_{i,j+1} - \frac{\Delta t (p_{i,j+1} - p_{i,j})}{h}$$
$$- u_{i,j} - \frac{\Delta t (p_{i,j} - p_{i-1,j})}{h} - v_{i,j} - \frac{\Delta t (p_{i,j} - p_{i,j-1})}{h} = 0$$

Step 4: Pressure Projection

The pressure is caused by incompressibility.
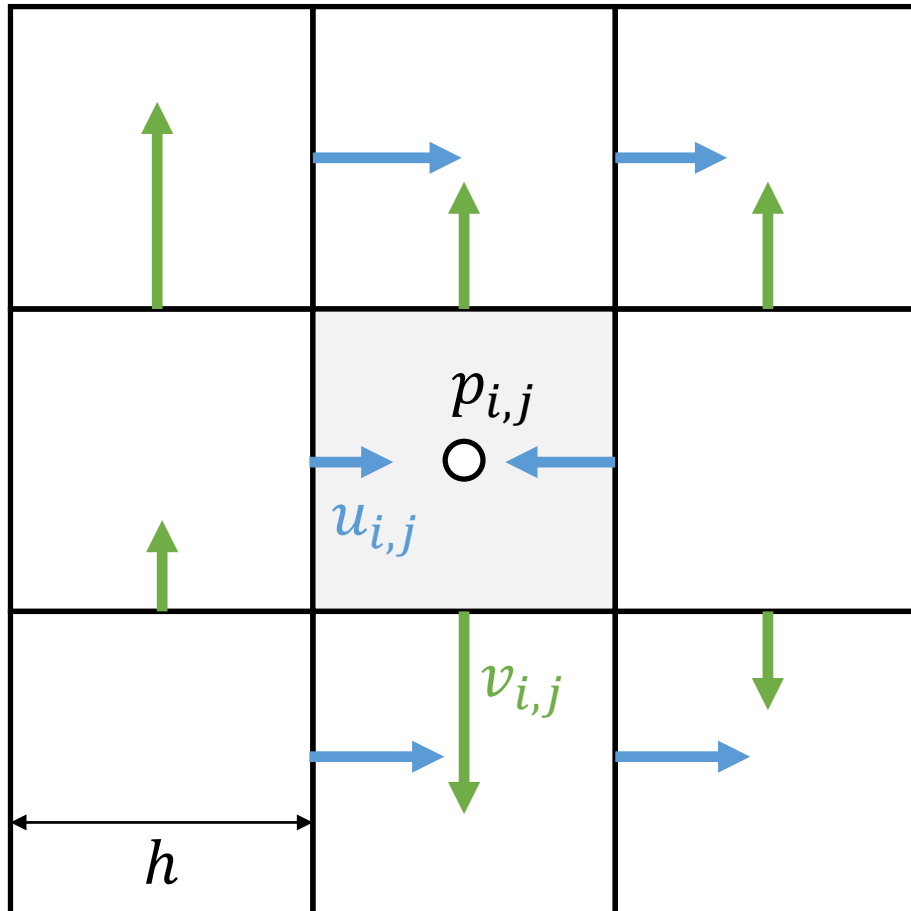
Eventually, we get a Poisson equation:

$$4p_{i,j} - p_{i-1,j} - p_{i+1,j} - p_{i,j-1} - p_{i,j+1} =$$
$$h(-u_{i+1,j} - v_{i,j+1} + u_{i,j} + v_{i,j})/\Delta t$$

with boundary conditions:

Dirichlet boundary (open) $p_{i-1,j} = P$

Neumann boundary (close) $p_{i-1,j} = p_{i,j}$

Once we solve $\boldsymbol{p}$, we update $\mathbf{u}$ and done.

After-Class Reading

Jos Stam. 1999. *Stable Fluids. TOG (SIGGRAPH).*

Air and Smoke

Air Simulation

- Air simulation is done in two steps.

- In Step 1, we update the flow (the velocity field) $\mathbf{u}$.

- In Step 2, we use semi-Lagrangian (page 22) advect all of the other physical quantities, i.e., density, temperature…

- Typically we use Dirichlet boundaries for an open space (or Neumann boundaries for a container.)

- We can use it to simulate underwater as well.

Water Simulation

- Two representations
  - Volume-of-fluid (as the name suggests…)
  - A signed distance function defined over the grid.

- How to advect?
  - Semi-Lagrangian (volume loss)
  - Level set method (volume loss)
  - Needs corrections.

But what if there is an air-water boundary???
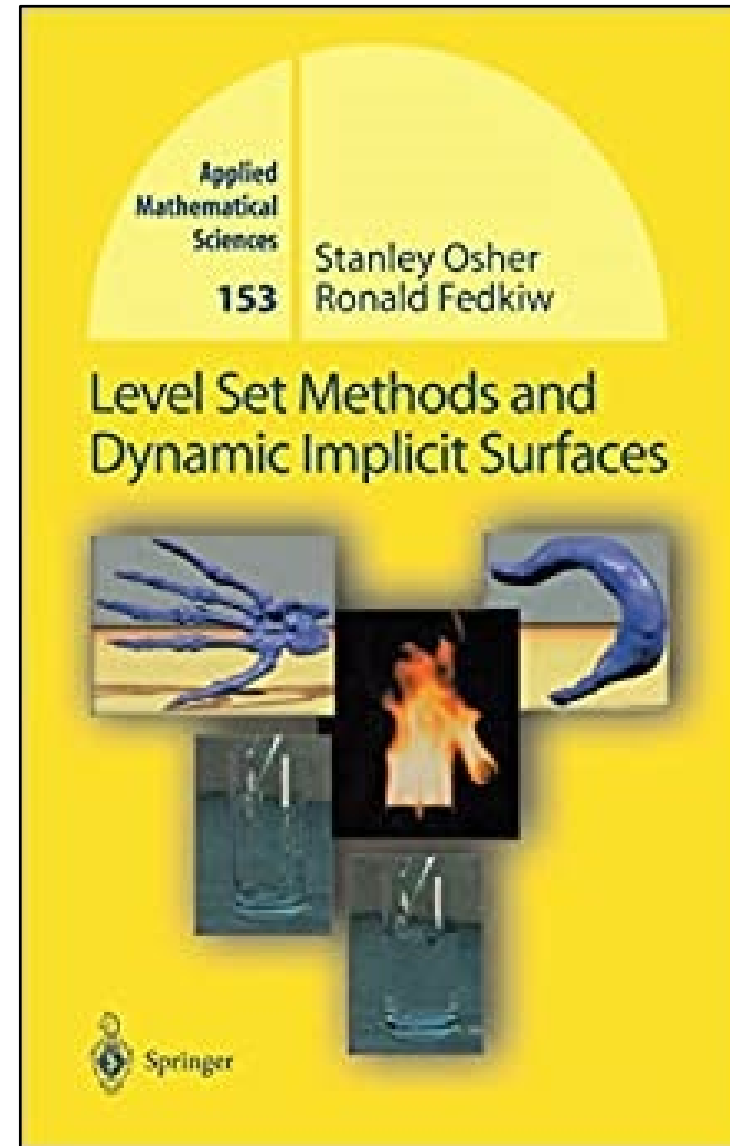
Water Simulation

- Two representations
  - Volume-of-fluid (as the name suggests...)
  - A signed distance function defined over the grid.

- How to advect?
  - Semi-Lagrangian (volume loss)
  - Level set method (volume loss)
  - Needs corrections.

But what if there is an air-water boundary???

After-Class Reading

Osher and Fedkiw.
Level Set Methods and Dynamic Implicit Surfaces.

A Summary For the Day

- The Eulerian grid presentation is very friendly with finite differencing. This makes calculus a lot easier.

- For velocity fields, we can use staggered grid.

- For low-speed, incompressible, viscous flow, we need to solve the Navier-Stokes equations.

- To solve the equations, we can do this in step-by-step (method of characteristics).

- To simulate air and water, we need to advect some physical quantities.
    - Smoke (density); water (volume-of-fluid, or signed distance function)
    - Volume loss issue in water (how to fix it?)
    - If you need to create a mesh from grid for rendering, you need something like <u>marching cube</u>.