COL865: Special Topics in Computer Applications

Physics-Based Animation

13 — Fluid simulation on grids

Review

Navier-Stokes equations for fluid velocity $\mathbf{u}(\mathbf{x}, t)$:

$$\partial \mathbf{u}/\partial t + (\mathbf{u} \cdot \nabla)\, \mathbf{u} = \rho^{-1} (-\nabla p + \mu\, \nabla^2 \mathbf{u} + \mathbf{f}_{\text{ext}})$$

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

Solve on grid via ***splitting***:

- ***Advection***: $\mathbf{u}^{(1)} = \text{advect}(\mathbf{u}^n, \mathbf{u}^n, \Delta t)$
- ***Body forces***: $\mathbf{u}^{(2)} = \mathbf{u}^{(1)} + \mathbf{f}_{\text{ext}}\, \Delta t$
- ***Viscosity***: $\mathbf{u}^{(3)} = \mathbf{u}^{(2)} + \nu\, \nabla^2 \mathbf{u}\, \Delta t$
- ***Pressure***: $\mathbf{u}^{n+1} = \mathbf{u}^{(3)} - \nabla p\, \Delta t$ so that $\nabla \cdot \mathbf{u}^{n+1} = 0$

Advection

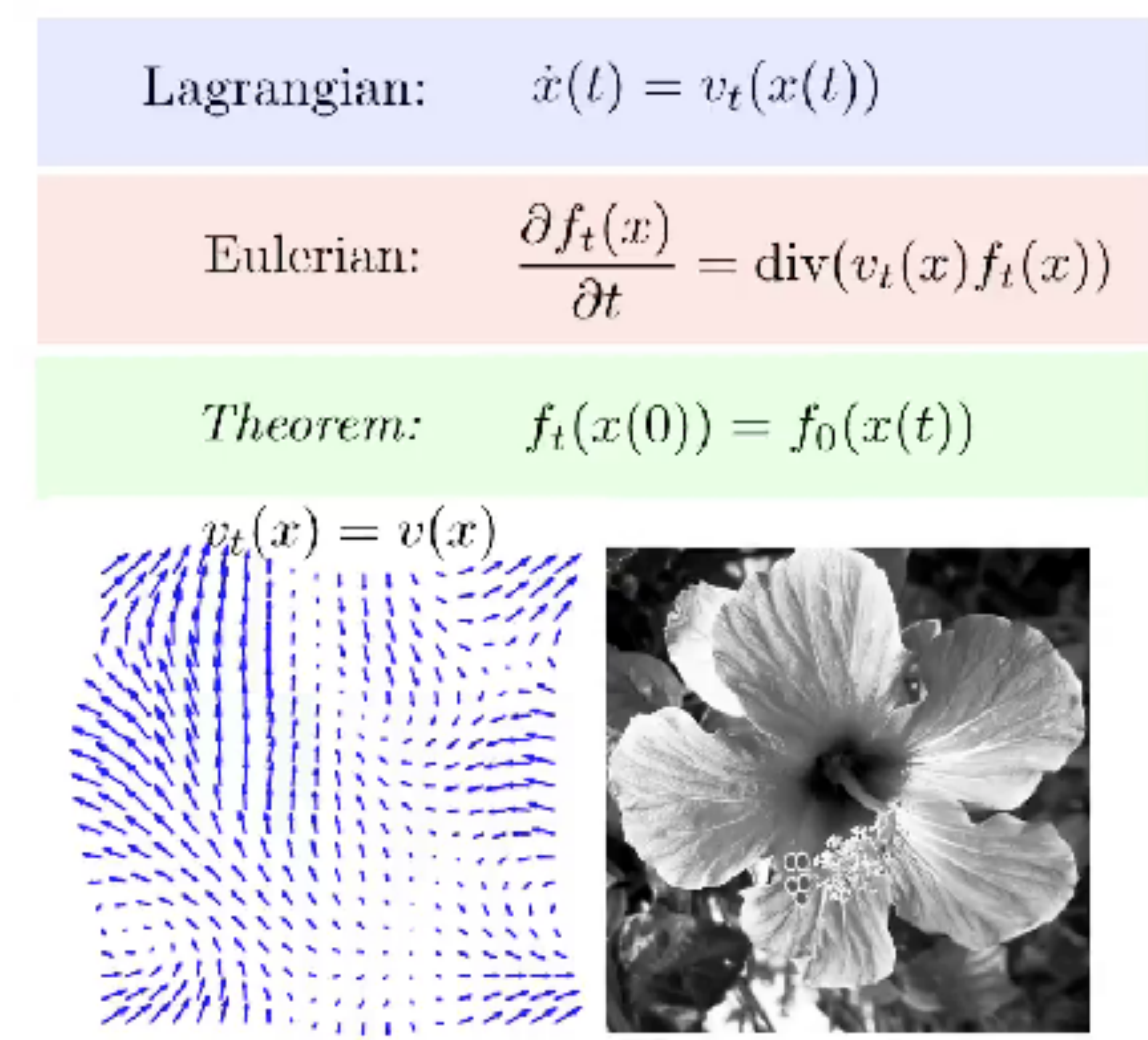
Advection

Advection of passive scalar c by velocity field $\mathbf{u}$:

$$\partial c/\partial t + \mathbf{u} \cdot \nabla c = 0$$

Given $c^n = c(\mathbf{x}, t^n)$, solve for $c^{n+1} = c(\mathbf{x}, t^{n+1})$

$$c^{n+1} = \mathrm{advect}(c^n, \mathbf{u}, \Delta t)$$



[Peyré 2018]

Finite differences

$$\partial c/\partial t + \mathbf{u} \cdot \nabla c = 0$$

Directly discretize $\partial c/\partial t$, ∇c with standard FD formulas

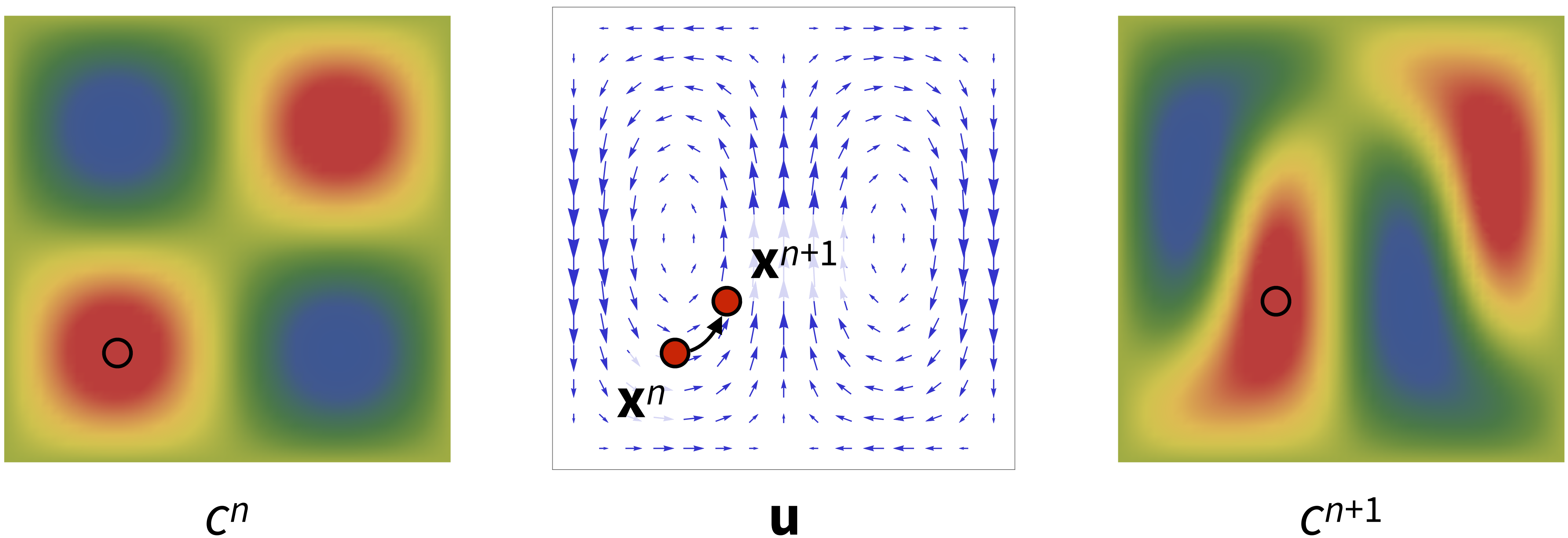
- Upwinding, Lax-Friedrichs, Lax-Wendroff, … [Trefethen Ch. 3.2]

Explicit schemes limited by CFL condition:

$$\Delta t \leq a\, \Delta x / \|\mathbf{u}\| \text{ for some constant } a$$

Semi-Lagrangian advection

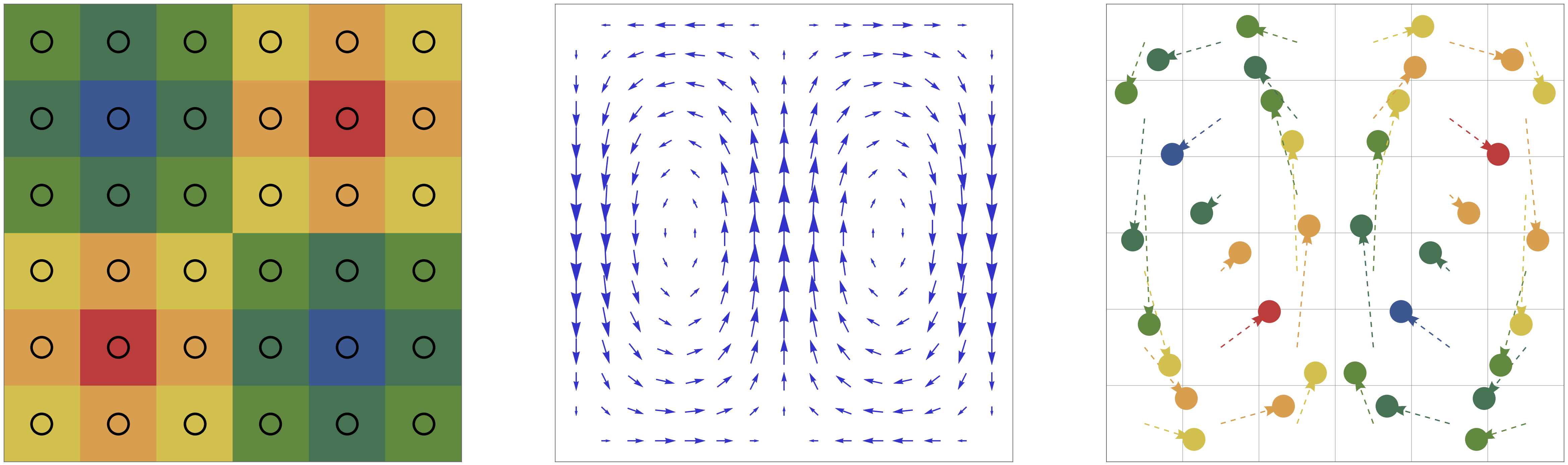
Particle view of advection:



Particle moves through velocity field: $\mathrm{d}\mathbf{x}_i/\mathrm{d}t = \mathbf{u}(\mathbf{x}_i)$

$$c(\mathbf{x}_i^{n+1}, t^{n+1}) = c(\mathbf{x}_i^n, t^n)$$

Semi-Lagrangian advection



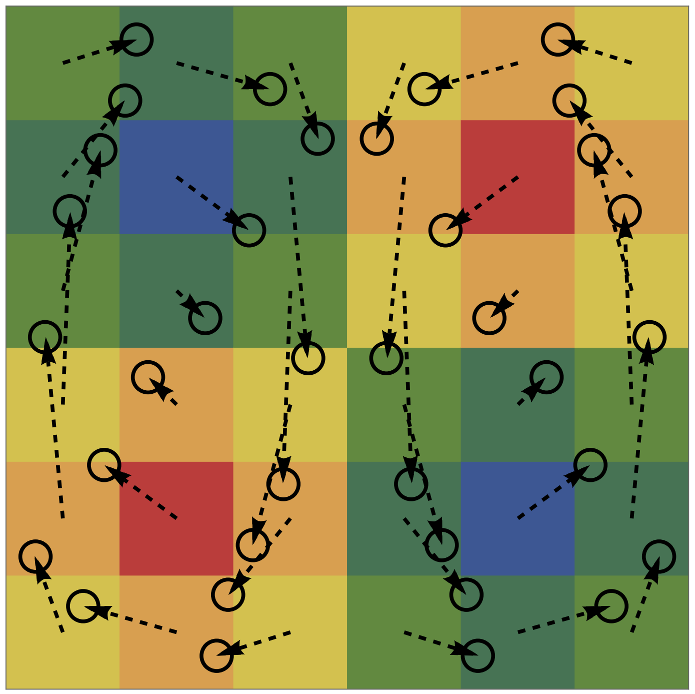
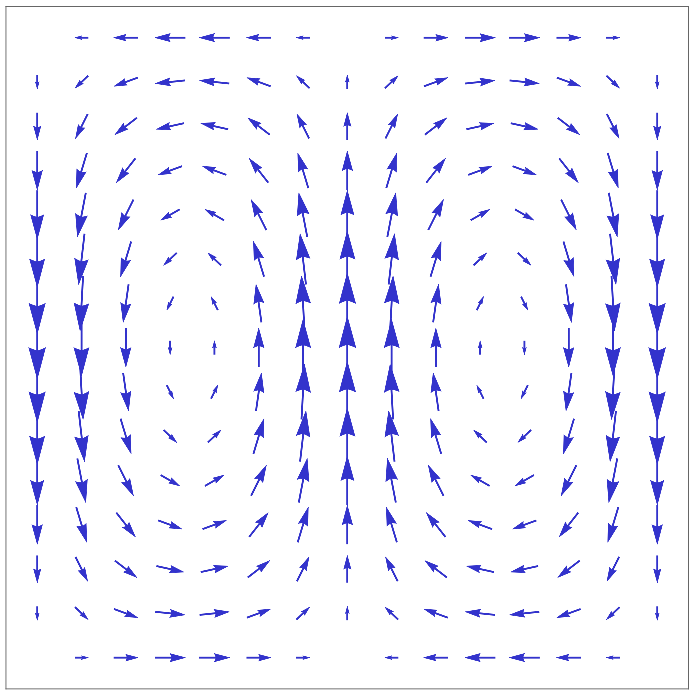
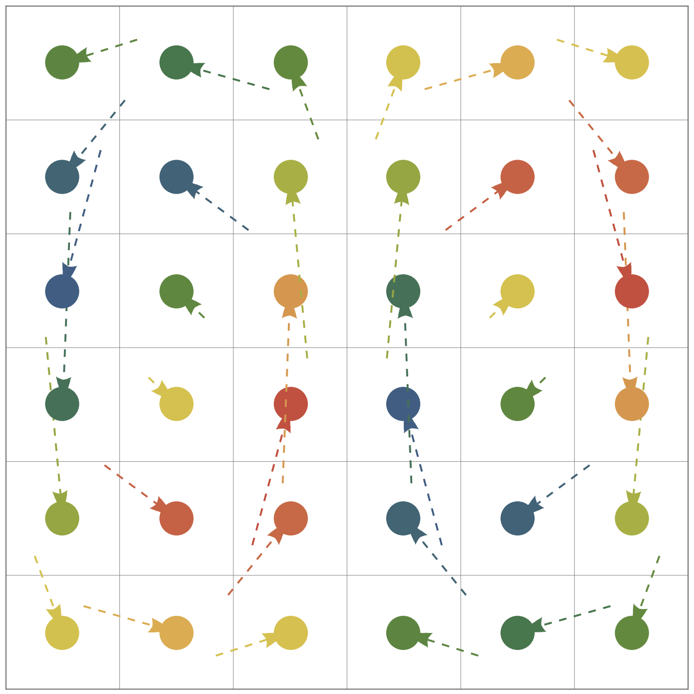
Idea:

1. For each grid node of c^n, create a particle

2. Trace particles forward with $d\mathbf{x}_i/dt = \mathbf{u}(\mathbf{x}_i)$ over Δt

But particles don't land on grid nodes of c^{n+1}

Semi-Lagrangian advection

Simple fix [Stam 1999]:

1. For each grid node of c^{n+1}, create a particle
2. Trace particles ***backwards*** over $-\Delta t$, look up (interpolated) value in c^n, write into c^{n+1}

Semi-Lagrangian advection

Advantage: Unconditionally stable

Limitation: Numerical diffusion

- Monotone cubic interpolation [Fedkiw et al. 2001, App. B]
- Higher-order correction schemes [Kim et al. 2005, Selle et al. 2006]

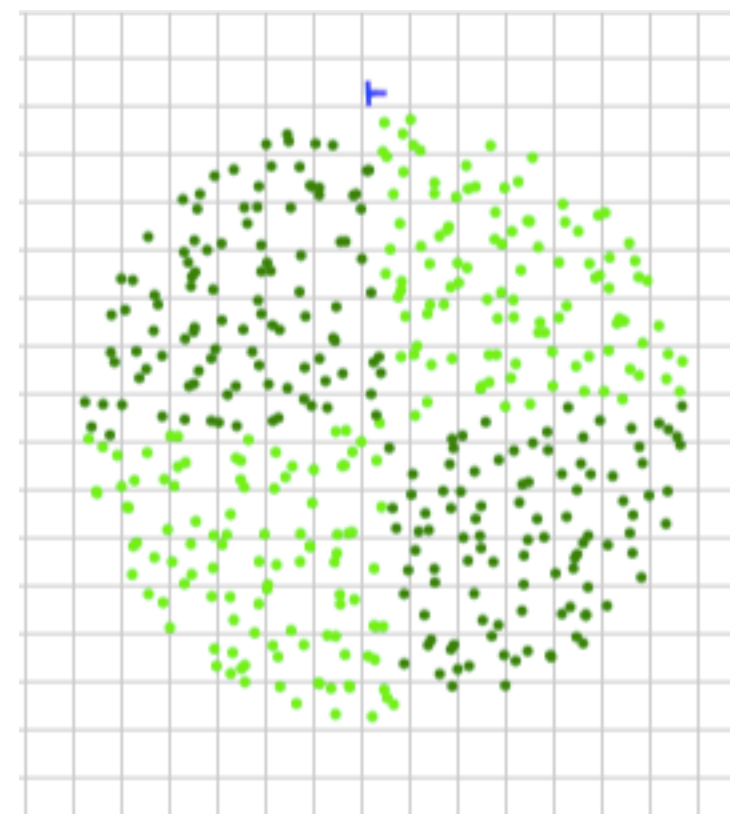
[Fedkiw et al. 2001]

Particle advection

Keep c stored on particles that persist across time steps [Zhu and Bridson 2005]

1. Trace particles forward with $\mathrm{d}\mathbf{x}_i/\mathrm{d}t = \mathbf{u}_{\text{grid}}(\mathbf{x}_i)$ as usual
2. Transfer c_i values to grid nodes: weighted average using grid interpolation weights

Diffusion doesn't accumulate over time steps



[Jiang et al. 2015]

Particle advection

$\mathbf{u}^{(1)} = \text{advect}(\mathbf{u}^n, \mathbf{u}^n, \Delta t)$

- Move particles using $\mathbf{u}_{\text{grid}}$
- Transfer particle $\mathbf{u}_i$ to grid

$\mathbf{u}^{(2)} = \mathbf{u}^{(1)} + \mathbf{f}_{\text{ext}}\,\Delta t$

$\mathbf{u}^{(3)} = \mathbf{u}^{(2)} + \nu\,\nabla^2\mathbf{u}\,\Delta t$

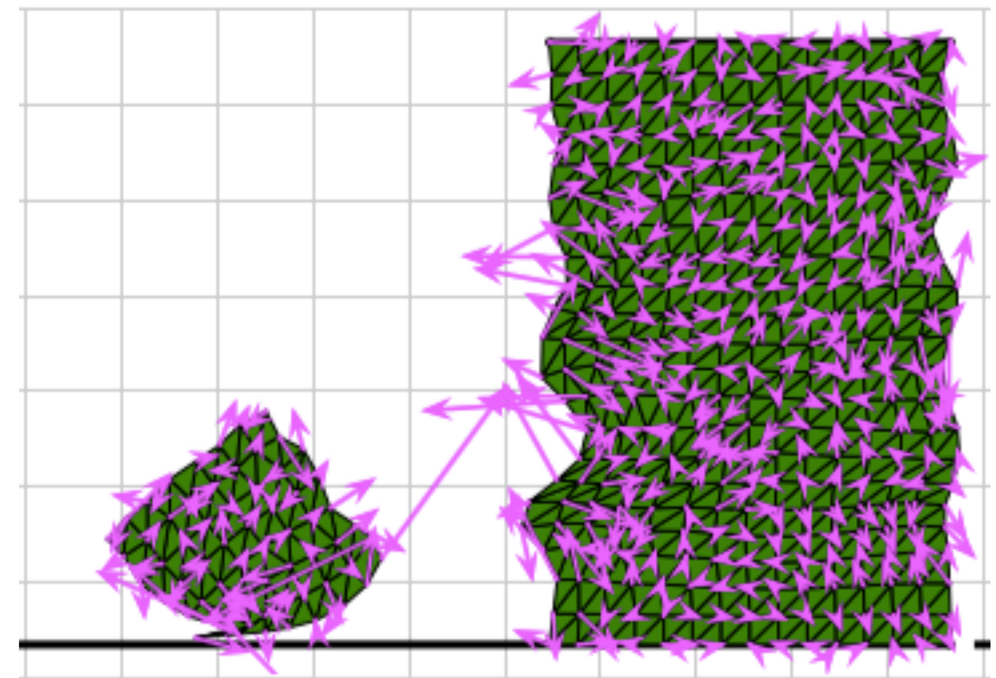
$\mathbf{u}^{n+1} = \mathbf{u}^{(3)} - \nabla p\,\Delta t,\ \ \nabla \cdot \mathbf{u}^{n+1} = 0$

At next time step, **u** on grid will have changed

- ***Particle-in-cell (PIC)***:
  First transfer values $\mathbf{u}_i = \mathbf{u}(\mathbf{x}_i)$
  - Problem: diffusion
- ***Fluid implicit particle (FLIP)***:
  Only transfer ***change*** in **u**
  (i.e. effect of forces)

See Zhu and Bridson [2005] for details

Particle advection



[Jiang et al. 2015]

Pure FLIP is unstable

- ***Fix***: Blend with small amount (1%-5%) of PIC
- ***Better fix***: Use APIC [Jiang et al. 2015], PolyPIC [Fu et al. 2017]

Pressure

Pressure

- $\mathbf{u}^{(1)} = \text{advect}(\mathbf{u}^n, \mathbf{u}^n, \Delta t)$
- $\mathbf{u}^{(2)} = \mathbf{u}^{(1)} + \mathbf{f}_{\text{ext}} \Delta t$
- $\mathbf{u}^{(3)} = \mathbf{u}^{(2)} + \nu \nabla^2 \mathbf{u} \Delta t$

After these steps, we have intermediate velocity $\tilde{\mathbf{u}} = \mathbf{u}^{(3)}$

$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \nabla p \, \Delta t,$$
$$\nabla \cdot \mathbf{u}^{n+1} = 0$$

“Project out” the divergence in $\tilde{\mathbf{u}}$

Pressure as decomposition

Helmholtz-Hodge decomposition:
Decompose $\tilde{\mathbf{u}}$ into divergence-free and curl-free components

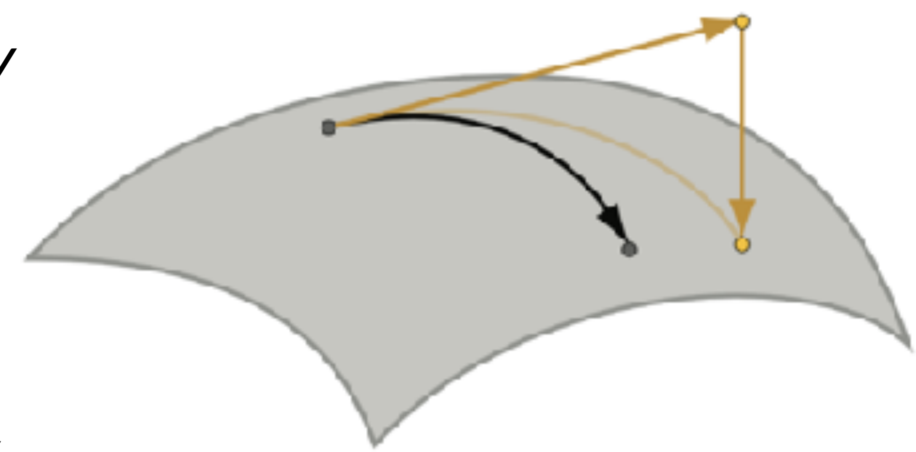
$$\tilde{\mathbf{u}} = \mathbf{u}^{n+1} + \nabla p$$

[Tong et al. 2003]

Pressure as projection

Given $\tilde{\mathbf{u}}$, find "nearest" vector $\mathbf{u}^{n+1}$ in divergence-free subspace

$$\mathbf{u}^{n+1} = \underset{\nabla \cdot \mathrm{u}=0}{\arg\min} \iiint \rho \|\mathbf{u} - \tilde{\mathbf{u}}\|^2 \, \mathrm{d}V$$



[Elcott et al. 2007]

Orthogonal projection

⇒ always reduces energy $\iiint \rho \, \|\mathbf{u}\|^2 \, \mathrm{d}V$

⇒ unconditionally stable

Computing the pressure

Just plug it in:

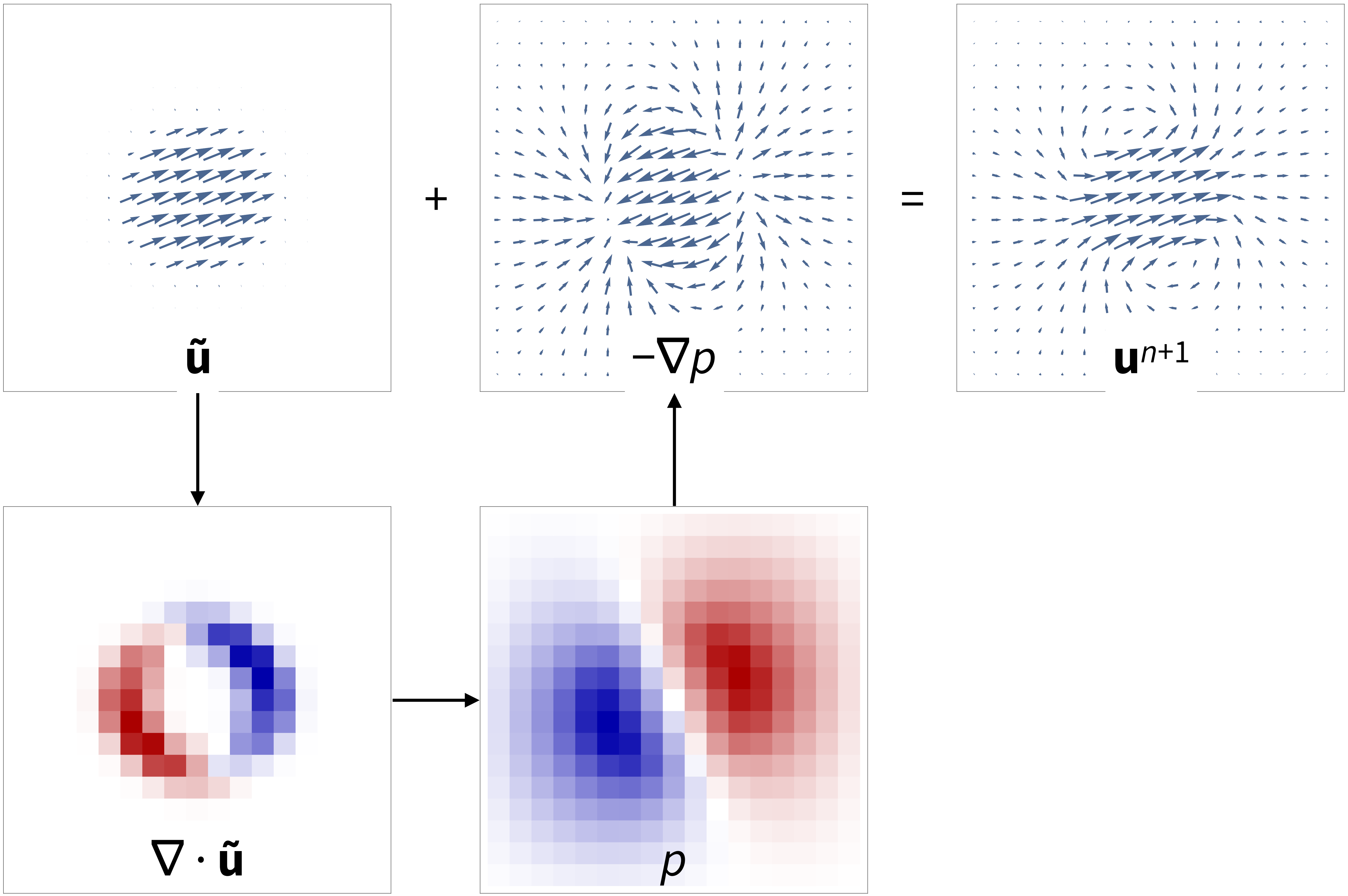
$$\mathbf{u}^{n+1} = \tilde{\mathbf{u}} - \nabla p \, \Delta t,$$
$$\nabla \cdot \mathbf{u}^{n+1} = 0$$

$$\Rightarrow \quad \nabla \cdot \tilde{\mathbf{u}} - \nabla^2 p \, \Delta t = 0$$

1. Compute $\nabla \cdot \tilde{\mathbf{u}}$
2. Solve PDE: $\nabla^2 p \, \Delta t = \nabla \cdot \tilde{\mathbf{u}}$ for p
3. Apply force $\nabla p \, \Delta t$ to get $\mathbf{u}^{n+1}$

Finding scalar field with specified Laplacian: ***Poisson problem***

Pressure projection



Spatial discretization

Issues discretizing grad, div:

- Forward, backward diff: directional bias
- Centered diff: ***null space problem***

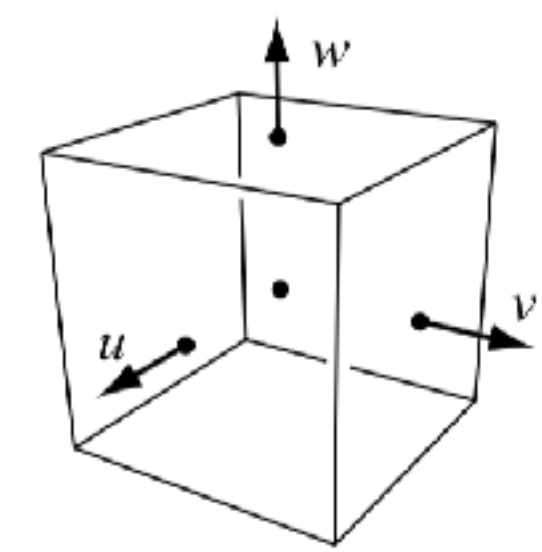
p=0	p=1	p=0	p=1	p=0

$\nabla p = 0$?!

→	←	→	→	→

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

Solution: ***Staggered grid***, a.k.a. ***marker-and-cell (MAC) grid***



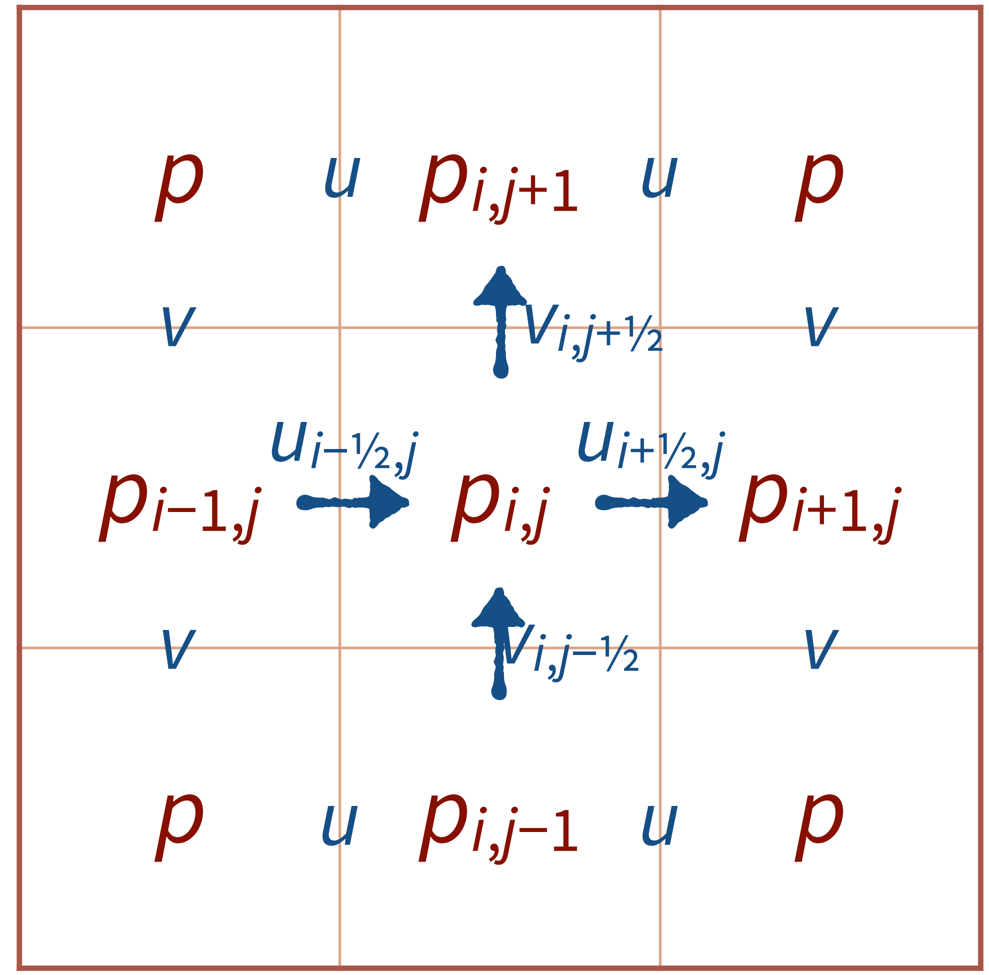
[Fedkiw et al. 2001]

Staggered grids

Scalars at cell centers, vector components on perpendicular faces

Fits nicely with grad, div:

- Value of $\nabla \cdot \mathbf{u}$ at cell center
- Components of ∇p on faces



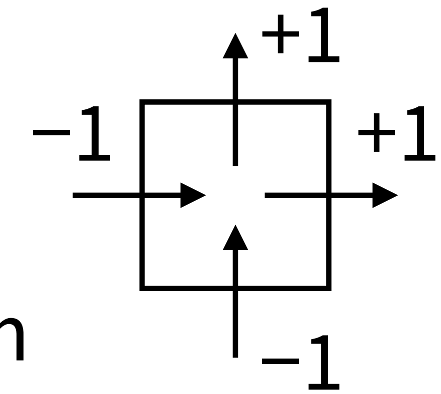
Implementation note: Be very careful about indexing!

- 3 separate arrays: $u_x(m+1, n, o)$, $u_y(m, n+1, o)$, $u_z(m, n, o+1)$

Pressure projection on staggered grids

1. Compute $\nabla \cdot \tilde{\mathbf{u}}$

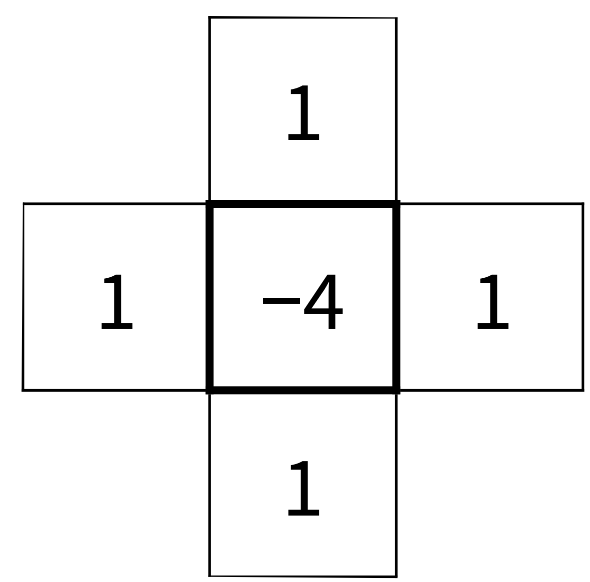
   $$(\nabla \cdot \mathbf{u})_{i,j} \approx (u_{i+½,j} - u_{i-½,j})/\Delta x + (v_{i,j+½} - v_{i,j-½})/\Delta x$$

   Looks like discrete version of divergence theorem
   $(\iiint \nabla \cdot \mathbf{u}\, \mathrm{d}V = \oiint \mathbf{u} \cdot \mathbf{n}\, \mathrm{d}A)$

2. Define Laplacian $\nabla^2 p$ as usual:

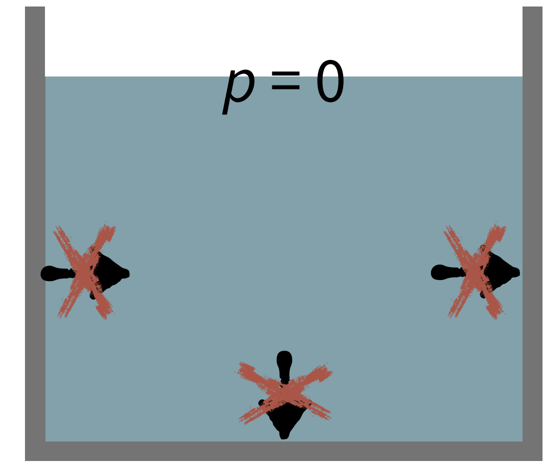
   $$(\nabla^2 p)_{i,j} \approx (p_{i-1,j} + p_{i+1,j} + p_{i,j-1} + p_{i,j+1} - 4\, p_{i,j})/\Delta x^2$$

Pressure boundary conditions

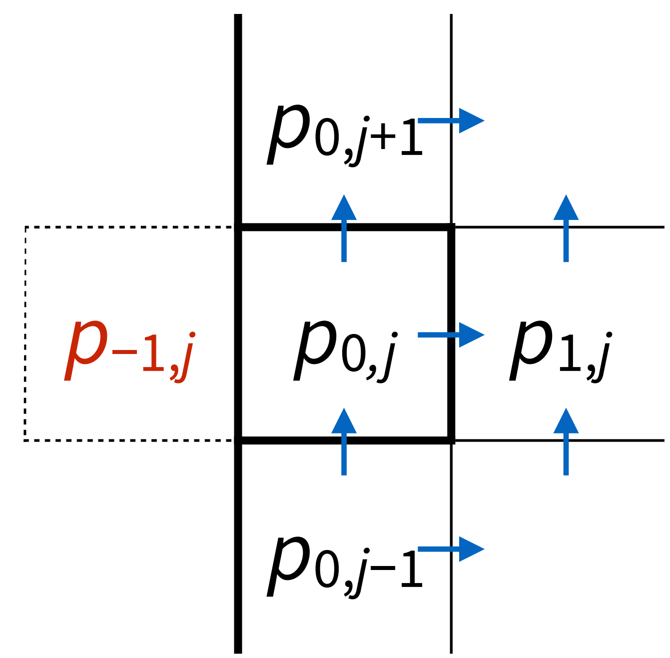
Solid boundaries:

- Fix $\tilde{\mathbf{u}} \cdot \mathbf{n} = 0$ (no-through boundary condition)
- Pressure shouldn't change this, so $\nabla p \cdot \mathbf{n} = 0$ (Neumann boundary)



$$p_{-1,j} = p_{0,j}$$

$$(\nabla^2 p)_{0,j} \approx (p_{-1,j} + p_{1,j} + p_{0,j-1} + p_{0,j+1} - 4\, p_{0,j})/\Delta x^2$$
$$= (p_{1,j} + p_{0,j-1} + p_{0,j+1} - 3\, p_{0,j})/\Delta x^2$$



Free surfaces: $p = 0$ (next class)