

Animating fluids (I)

Stable Fluid

Solving Navier-Stokes on grid

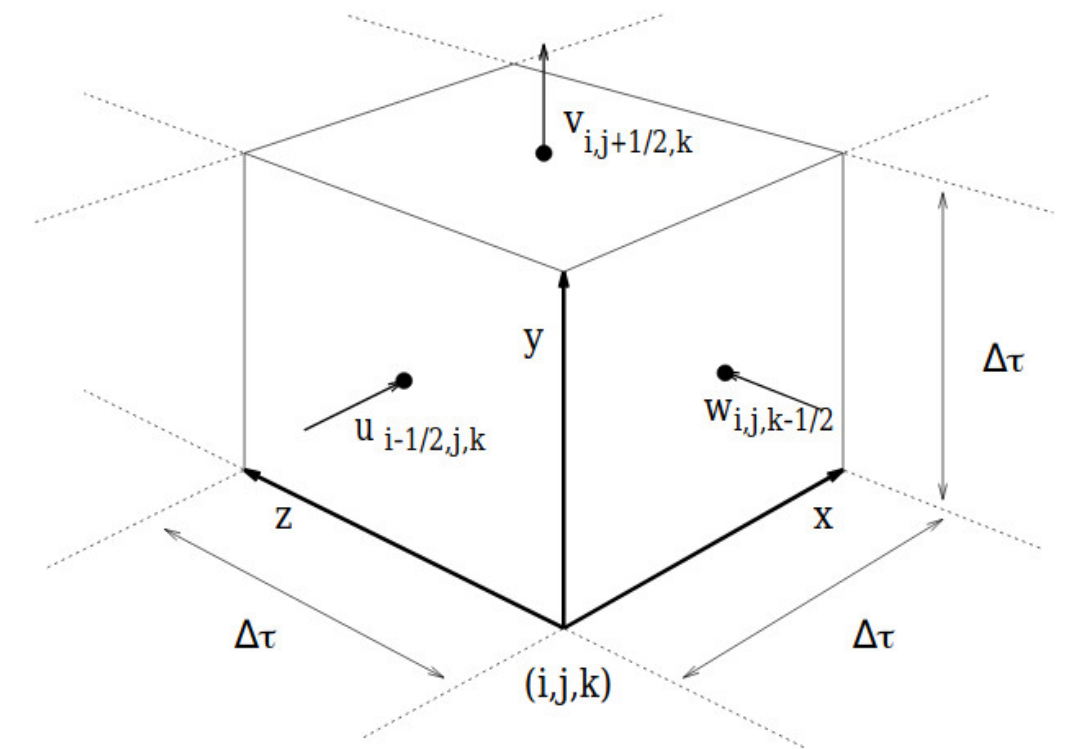
"Brute force" approach

- Rectangular grid filled with fluid
- Use finite differences on the grid for Navier-Stokes equation

$$\frac{\partial u}{\partial t} = -\frac{1}{\rho} \nabla p + f - (u \cdot \nabla)u + \nu \Delta u$$
$$\text{div}(u) = 0$$

(-) Stability conditions

(-) Loose advection details on the grid



[Modeling the Motion of a Hot, Turbulent Gas. N Foster and D. Metaxas. SIGGRAPH 1997]

Stable Fluids - Idea

Well known improvement: **Jos Stam, Stable Fluids, ACM SIGGRAPH 1999**

$$\frac{\partial u}{\partial t} = f - (u \cdot \nabla)u + \nu \Delta u - \frac{1}{\rho} \nabla p$$

- $1/\rho \nabla p$: Pressure term only used to ensure divergence free
- Similar to Lagrange multiplier for constraints

1st Idea

Remove pressure term

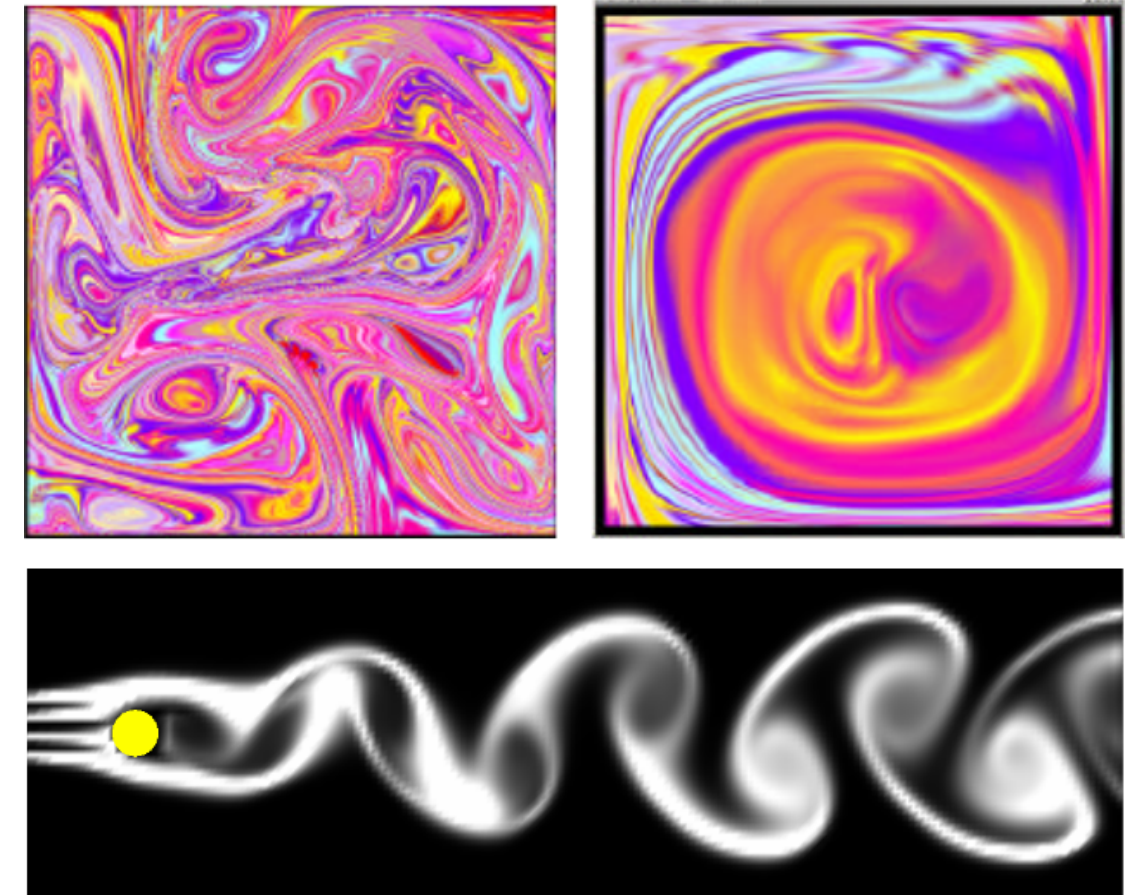
Replace by explicit projection on divergence free vector field P

$$\Rightarrow \frac{\partial u}{\partial t} = P(f - (u \cdot \nabla)u + \nu \Delta u)$$

2nd Idea

Compute each terms one after the other

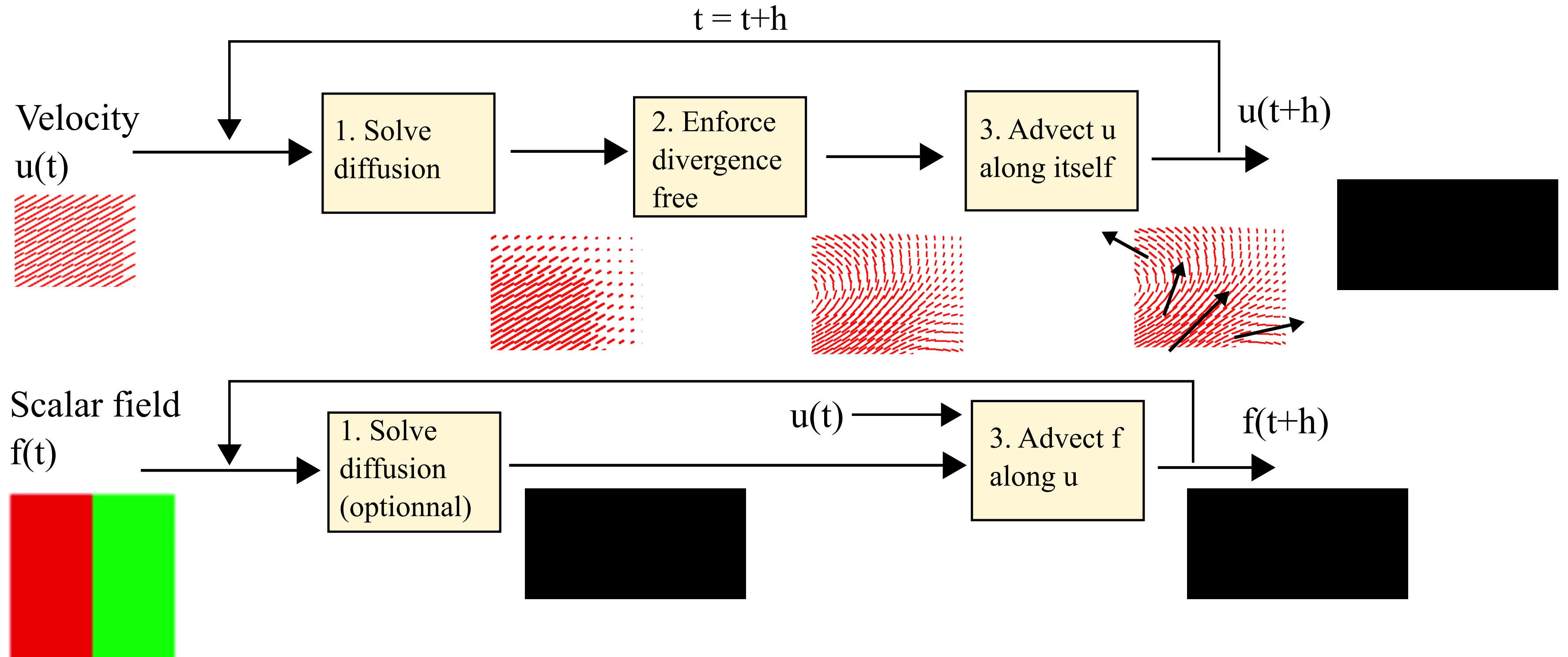
$$u^k \xrightarrow[\text{add forces } f]{\quad} u_1^k \xrightarrow[\text{diffuse } \nu \Delta u]{\quad} u_2^k \xrightarrow[\text{project } P]{\quad} u_3^k \xrightarrow[\text{advect } (u \cdot \nabla)u]{\quad} u^{k+1}$$



[*Stable Fluids*. J. Stam. SIGGRAPH 1999]

[*Real Time Fluid Dynamics for Games*. J. Stam. Game Dev. Conf. 2003]

Stable Fluids - General Algorithm



1 - Diffusion

Use finite difference on $\frac{\partial f}{\partial t} = \nu \Delta f$

Notation: $f_{x,y}^t = f(k_x \Delta x, k_y \Delta y, k_t \Delta t)$

Explicit schemes may oscillates/diverge for large time steps

$\Rightarrow$ Use implicit scheme for unconditional stability

$$\frac{f_{x,y}^{k+1} - f_{x,y}^k}{\Delta t} = \nu \left(\frac{f_{x+1,y}^{k+1} - 2 f_{x,y}^{k+1} + f_{x-1,y}^{k+1}}{(\Delta x)^2} + \frac{f_{x,y+1}^{k+1} - 2 f_{x,y}^{k+1} + f_{x,y-1}^{k+1}}{(\Delta y)^2} \right)$$

Assuming $\Delta x = \Delta y = 1$

$$(1 + 4 \nu \Delta t) f_{x,y}^{k+1} - \nu \Delta t (f_{x+1,y}^{k+1} + f_{x-1,y}^{k+1} + f_{x,y+1}^{k+1} + f_{x,y-1}^{k+1}) = f_{x,y}^k$$

Use Gauss-Seidel iterative method to solve the sparse linear system

Initialize $f^{k+1} = f^k$

for $i = 1..N_{\max}$

$$f_{x,y}^{k+1} = \frac{1}{1+4a} (f_{x,y}^k + a(f_{x-1,y}^{k+1} + f_{x+1,y}^{k+1} + f_{x,y-1}^{k+1} + f_{x,y+1}^{k+1})) , \quad a = \nu \Delta t$$

2 - Advection

Advection = move some function along given velocity u .

- Advecting a scalar field f along u

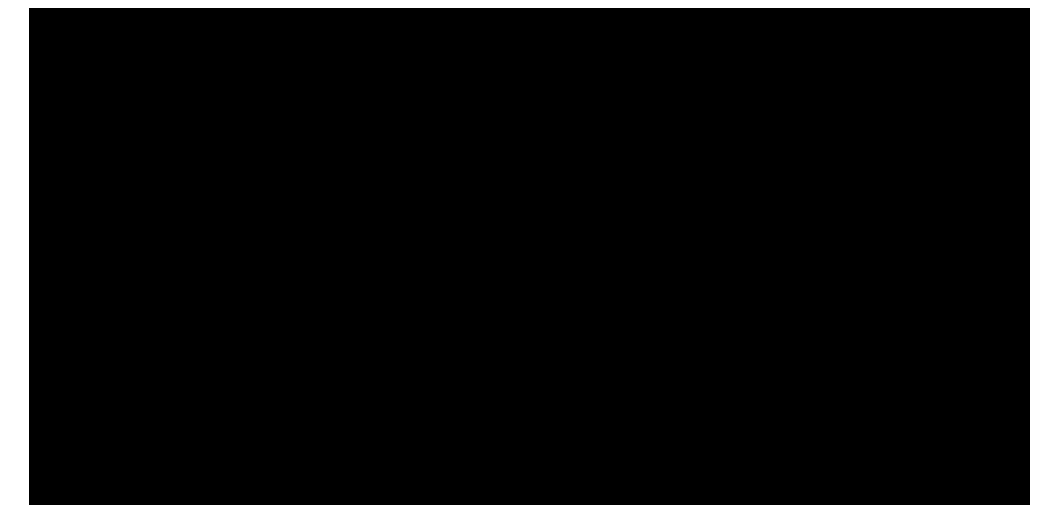
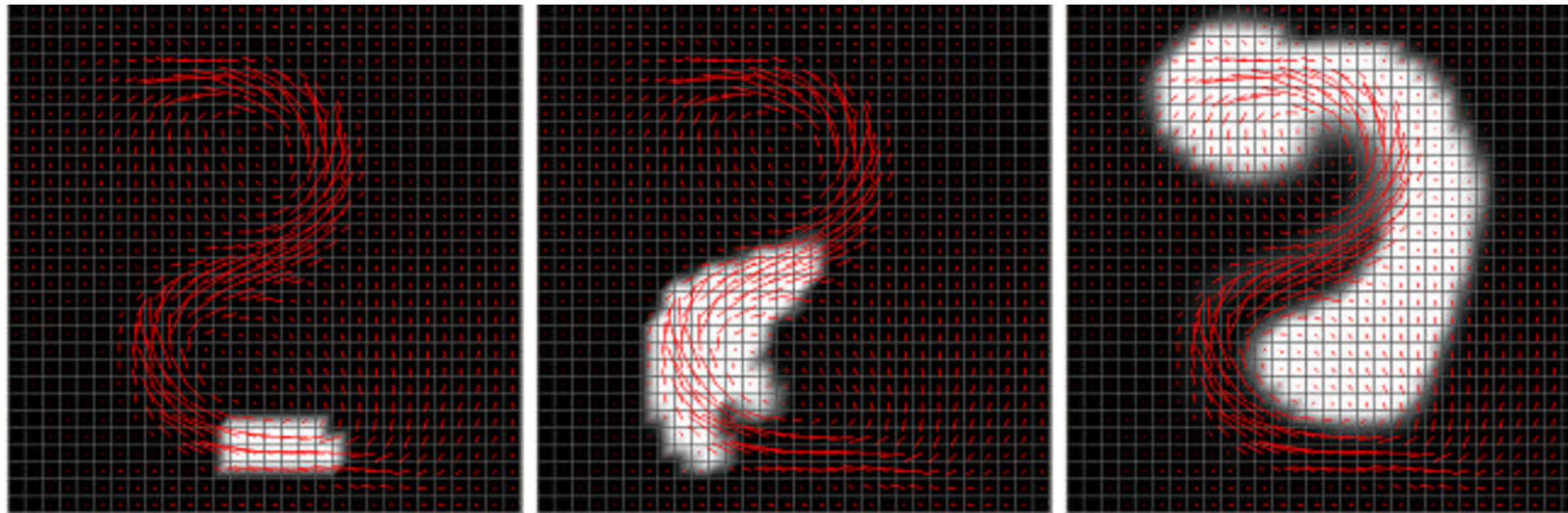
$$\frac{\partial f}{\partial t}(p, t) + u(p, t) \cdot \nabla f = 0$$

- Advecting a vector field f along u

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

- In Navier-Stokes advect the velocity itself $f = u$

- Can also advect density, color, texture coordinates, etc. to visualize the motion.



2 - Computing advection

Advecting generic value f along u

Idea Compute value of f at time t at fixed position grid p in moving back at $t - \Delta t$.

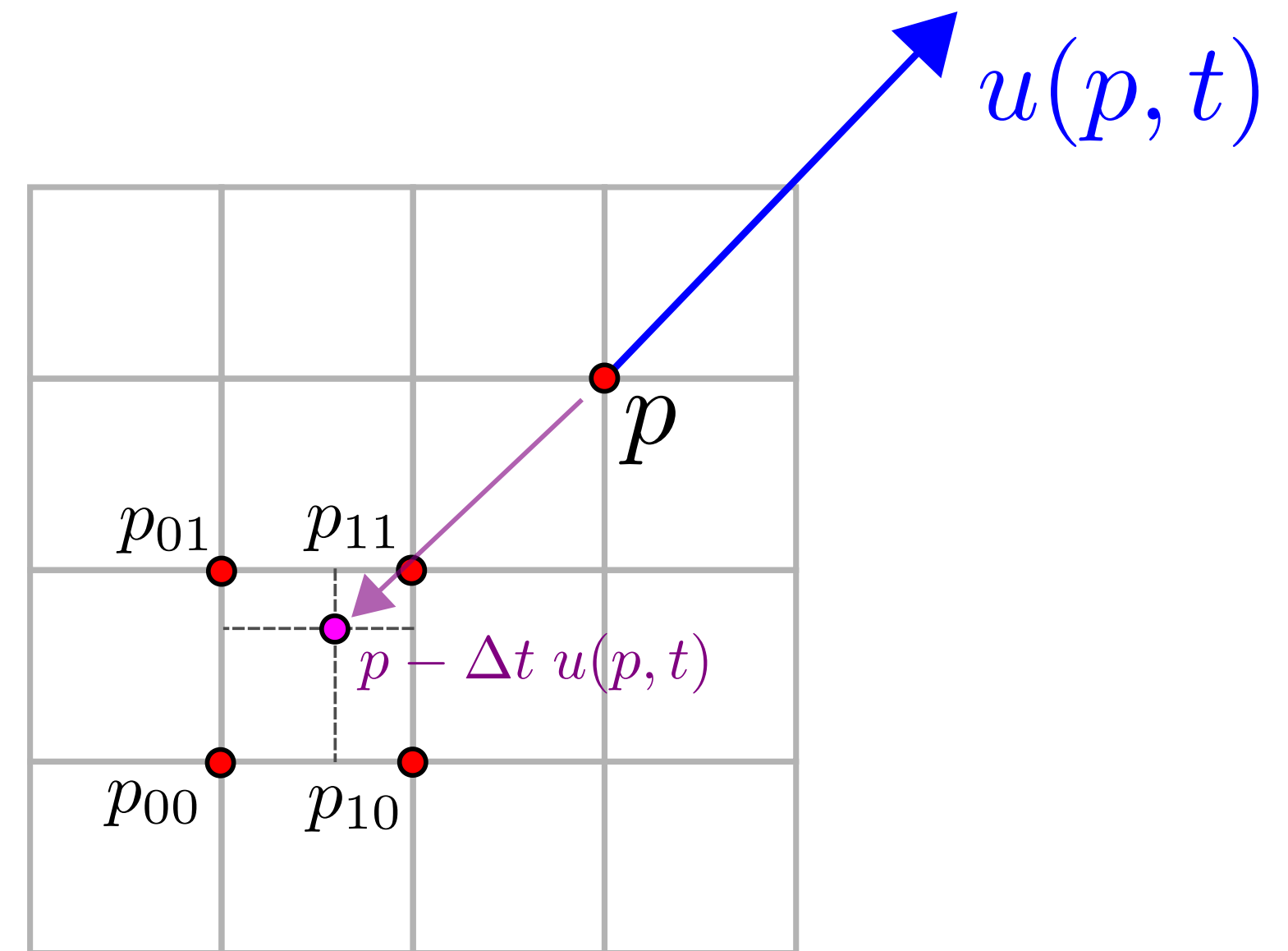
Value of f advected at point p at time t was at position $p_{prev} = p - \Delta t v(p, t)$ at time $t - \Delta t$.

$$\Rightarrow f(p, t) = f(p_{prev}, t - \Delta t)$$

p_{prev} is not a grid point coordinates: Use interpolation

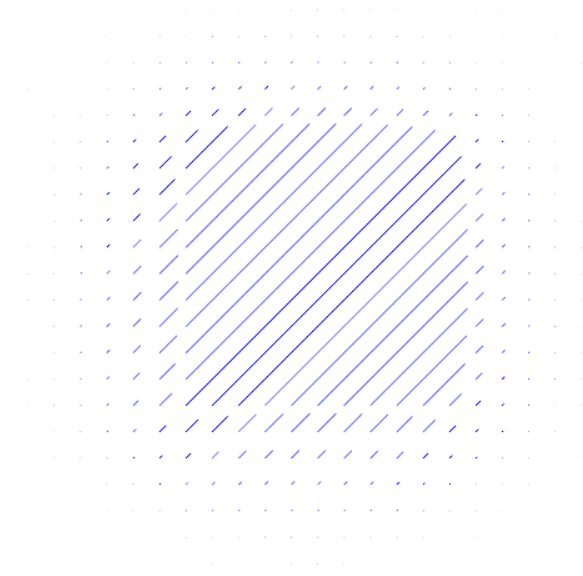
Can use Bilinear interpolation

$$f(p_{prev}) = (1 - \alpha)(1 - \beta)p_{00} + (1 - \alpha)\beta p_{01} + \alpha(1 - \beta)p_{10} + \alpha\beta p_{11}$$

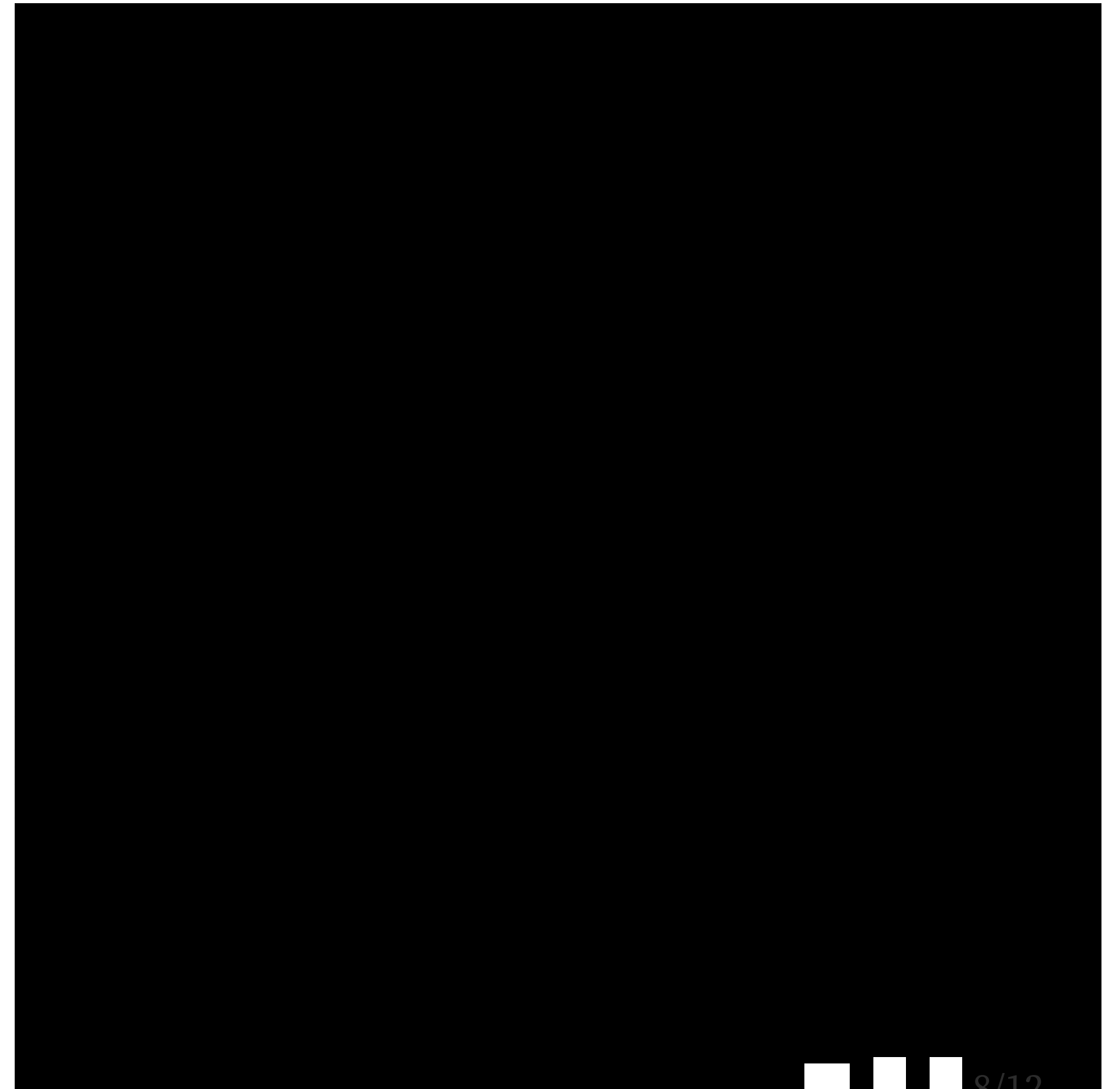
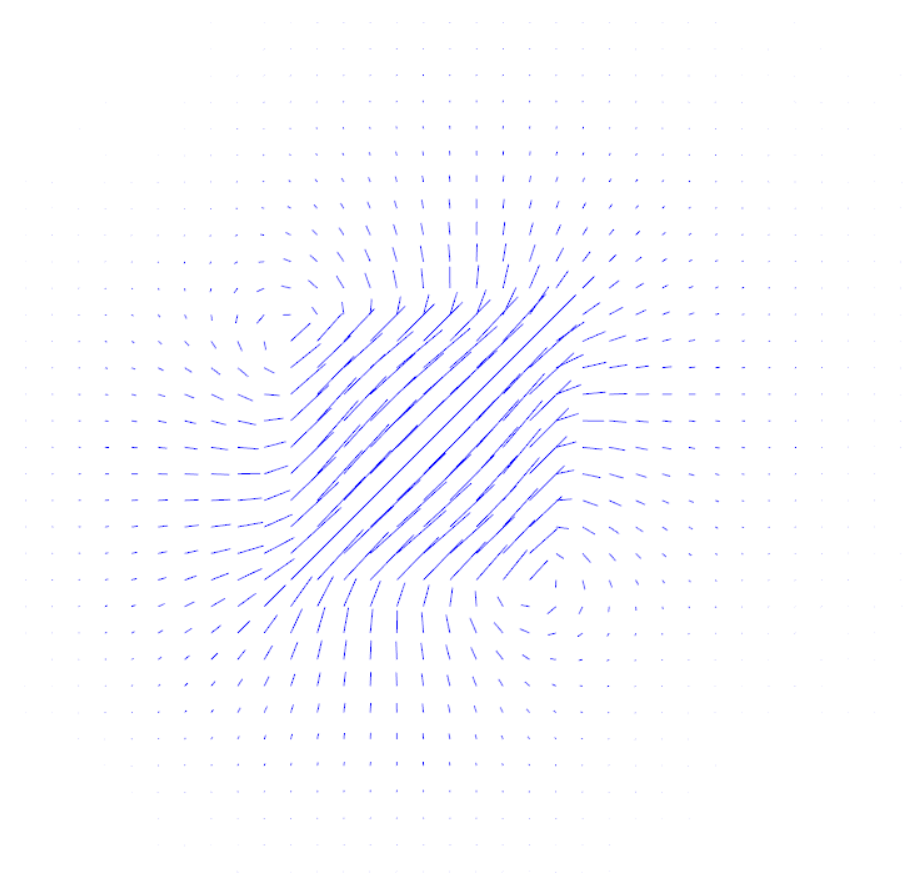


3 - Divergence Free Vector Field

Before projection:



After projection:



3 - Projection to divergence free vector field

Consider a general vector field w

Helmoltz decomposition: $w = u + v$

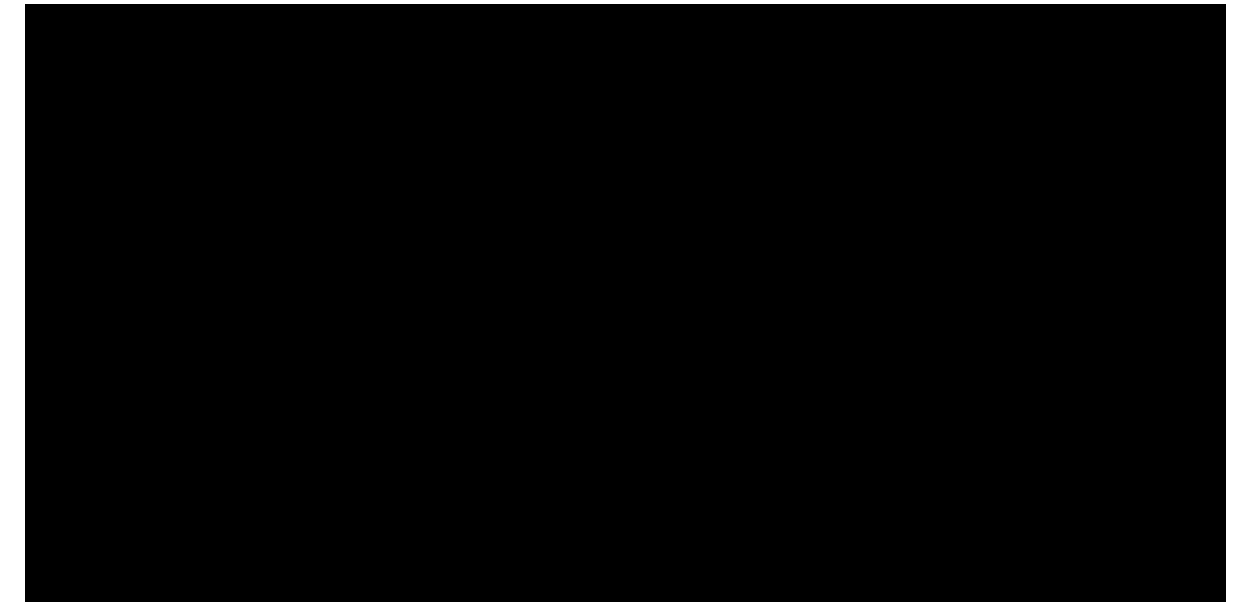
- u : Divergence free vector field such that $\text{div}(u) = 0$
- v : Gradient field $v = \nabla q$, q scalar field.

q satisfies a Poisson equation

$$\text{div}(w) = \underbrace{\text{div}(u)}_{=0} + \text{div}(v) \Rightarrow \text{div}(w) = \underbrace{\text{div}(\nabla q)}_{\Delta q}$$

Method- Given an input field w

1. Compute q as solution of $\Delta q = \text{div}(w)$
2. Compute $u = w - \nabla q$



3 - Projection to divergence free vector field (Algo)

Input vector field $w = (w^x, w^y)$

Note: we assume in the following $\Delta x = \Delta y = 1$

1 - Compute $d = \text{div}(w)$

$$d_{x,y} = (w_{x+1,y}^x - w_{x-1,y}^x + w_{x,y+1}^y - w_{x,y-1}^y) / 2$$

2 - Compute q in solving $\Delta q = b$

$$(q_{x+1,y} + q_{x-1,y} - 2q_{x,y}) + (q_{x,y+1} + q_{x,y-1} - 2q_{x,y}) = d_{x,y}$$
$$\Rightarrow 4q_{x,y} = q_{x+1,y} + q_{x-1,y} + q_{x,y+1} + q_{x,y-1} - d_{x,y}$$

ex. Numerical iterations using Gauss Seidel

Initialize $q = 0$

For $i = [1..N_{\max}]$

$$q_{x,y} = 1/4 (q_{x+1,y} + q_{x-1,y} + q_{x,y+1} + q_{x,y-1} - d_{x,y})$$

3 - Compute $u = w - \nabla q$

$$u_{x,y} = w_{x,y} - (q_{x+1,y} - q_{x-1,y}, q_{x,y+1} - q_{x,y-1}) / 2$$

Handling boundaries

Boundaries $x = 0, x = N_x - 1, y = 0, y = N_y - 1$
need special care

- For density

Assume value C^0 continuity on the boundary

Row/Column $f_{x,0} = f_{x,1}, f_{0,y} = f_{1,y}$ etc.

- For velocity: $f = (f^x, f^y)$

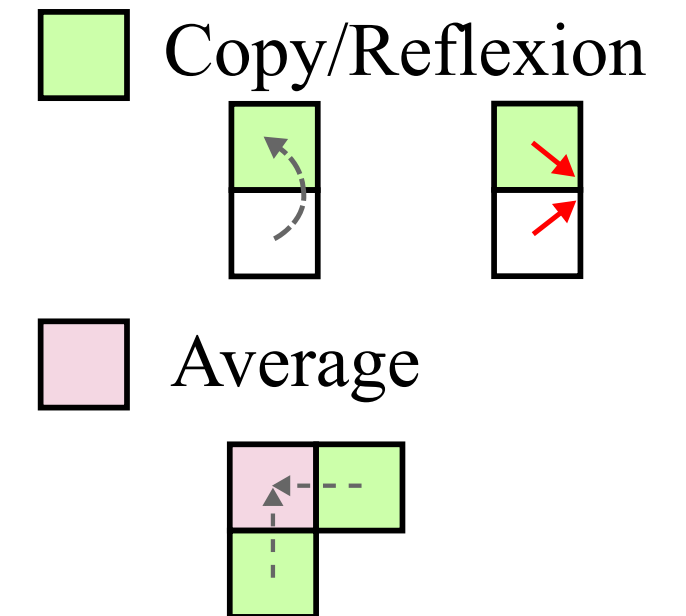
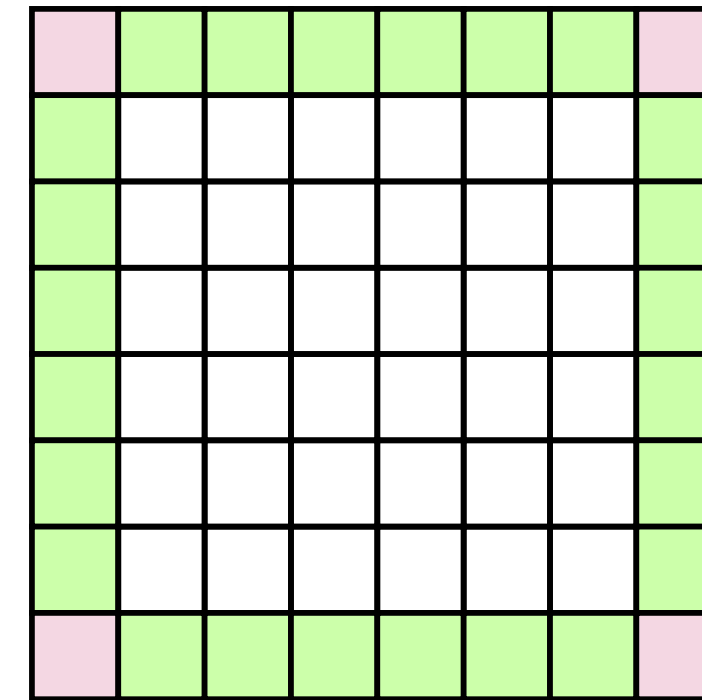
Assume reflexion on walls

Row: $f_{x,0} = (f_{x,1}^x, -f_{x,1}^y)$

Column: $f_{0,y} = (-f_{1,y}^x, f_{1,y}^y)$

- In all cases: Average value for corners

$f_{0,0} = (f_{1,0} + f_{0,1})/2$, etc.



Stable fluids example