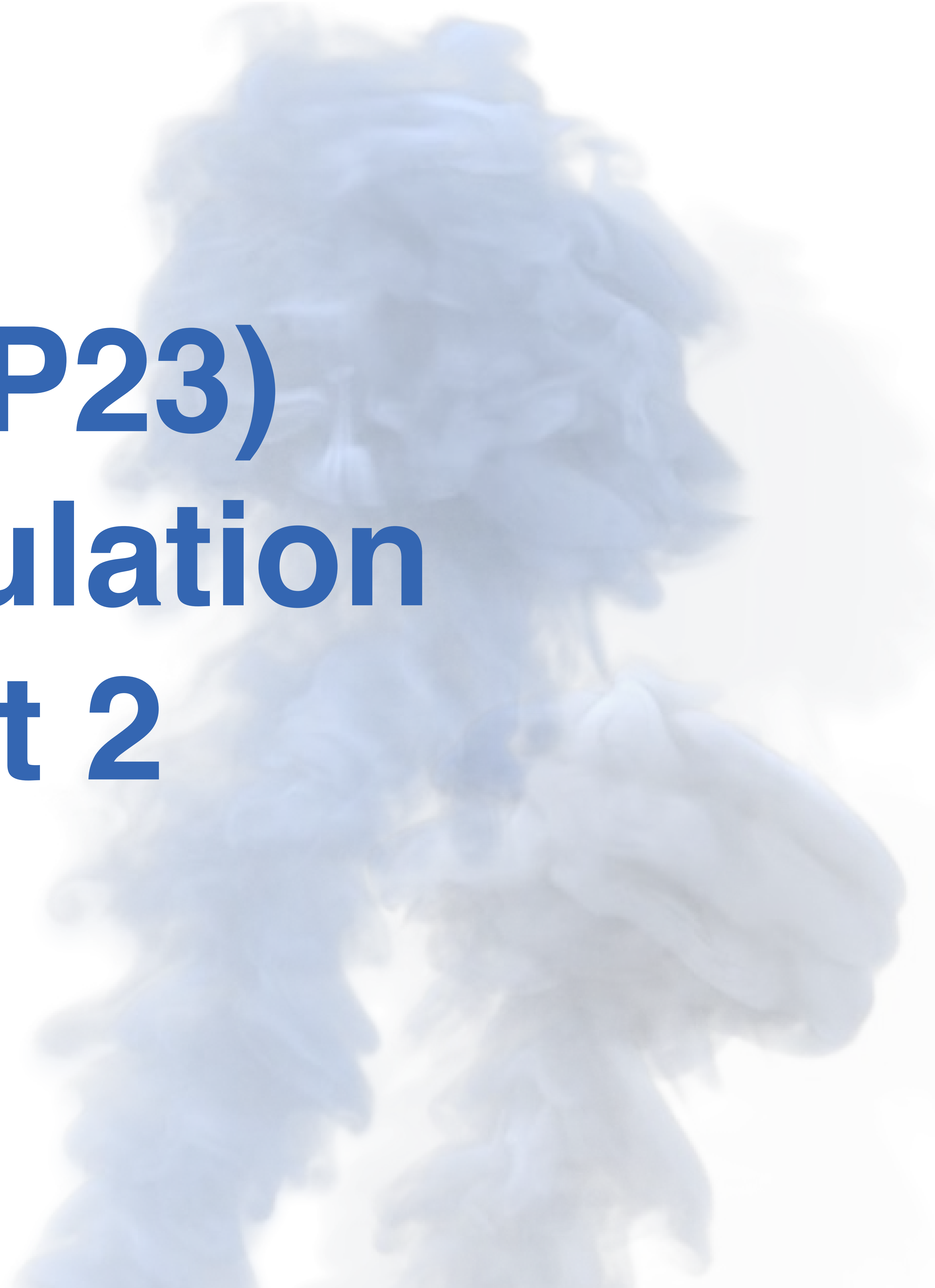
CSE 291 (SP23)
Physical Simulation
Fluid: Part 2

Albert Chern



UC San Diego
JACOBS SCHOOL OF ENGINEERING

Pressure Projection

- Pressure projection
- Time splitting
- Advection
- Particle in cell
- Vortex methods
- Lattice Boltzmann methods

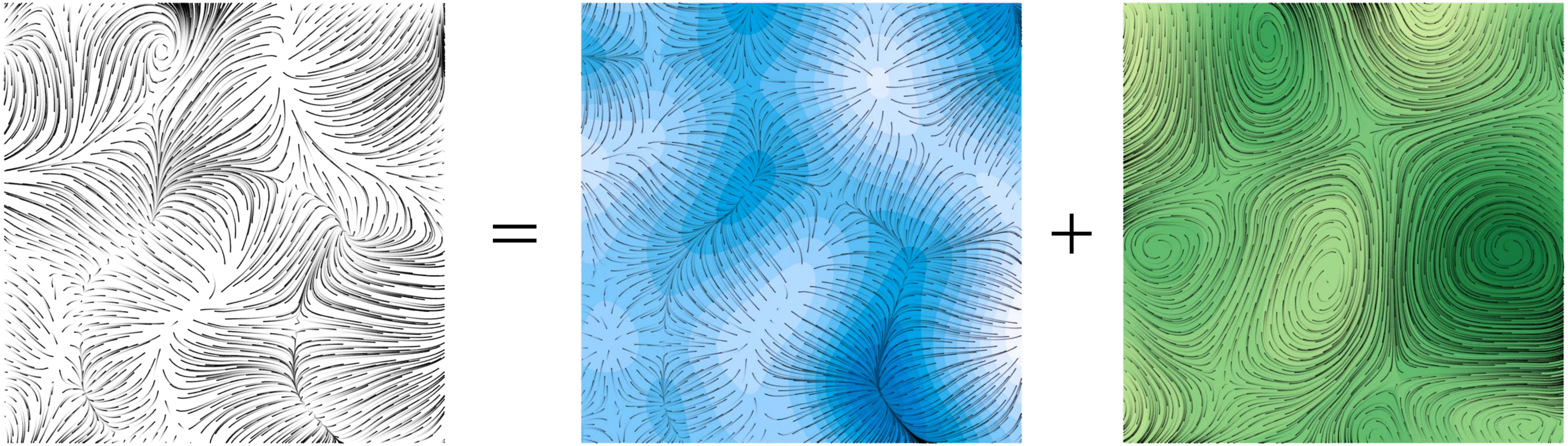
Helmholtz decomposition

- Helmholtz (1858) discovered the following vector field decomposition in $\mathbb{R}^2, \mathbb{R}^3$

$$\Gamma(T\mathbb{R}^n) = (\text{curl-free VF}) \oplus (\text{div-free VF})$$

$$\text{curl-free VF} = \text{im}(\text{grad})$$

$$\text{div-free VF} = \text{im}(\text{curl}) \text{ in 3D}, \quad \text{im}(R^{90^\circ}\,\text{grad}) \text{ in 2D}$$



Helmholtz decomposition

- Helmholtz (1858) discovered the following vector field decomposition in $\mathbb{R}^2, \mathbb{R}^3$

$$\Gamma(T\mathbb{R}^n) = (\text{curl-free VF}) \oplus (\text{div-free VF})$$

$$\begin{array}{cc} \| & \| \\ \mathrm{im}(\mathrm{grad}) & \mathrm{im}(\mathrm{curl}) \text{ in 3D} \\ & \mathrm{im}(R^{90^\circ}\,\mathrm{grad}) \text{ in 2D} \end{array}$$

- This decomposition is orthogonal.
- Given a vector field, we can remove the im(grad) part by the **pressure projection** (with a Poisson solve).
- One can also directly construct the divergence-free part using either a **Biot–Savart integral** or a **streamfunction Poisson solve**.

Pressure projection

- Let $\tilde{\mathbf{u}}$ be any vector field on a domain W.
- There is a scalar field p, unique up to additive constant, so that

$$\mathbf{u} = \tilde{\mathbf{u}} - \nabla p$$

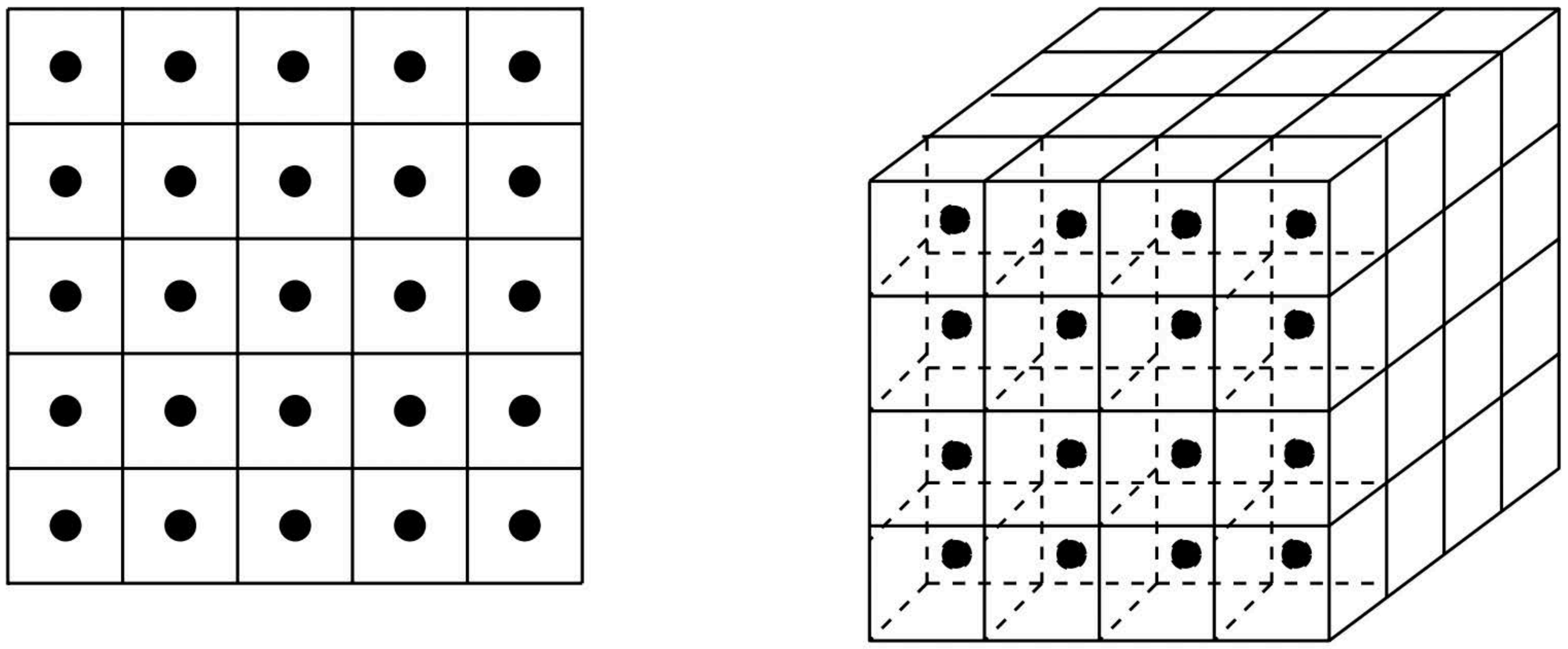
satisfies

$$\begin{cases} \nabla \cdot \mathbf{u} = 0 & \text{in } W \\ \mathbf{n} \cdot \mathbf{u} = 0 & \text{on } \partial W \end{cases}$$

- Solve the Poisson problem

$$\begin{cases} \Delta p = \nabla \cdot \tilde{\mathbf{u}} & \text{in } W \\ \dfrac{\partial p}{\partial \mathbf{n}} = \mathbf{n} \cdot \tilde{\mathbf{u}} & \text{on } \partial W \end{cases}$$

Marker and Cell (MAC grid)



- Define pressure and “label for obstacle” in cells, and velocity (as flux) on cell walls.
- Grad *p* and divergence are both naturally defined.
- Build gradient and divergence matrix: they are transpose of each other.

Spectral method

- On a periodic domain (no boundary) of size (L_1, L_2, L_3)

Solve Poisson problem $\Delta u = f$ $\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = f$

- Discretize both *u* and *f* on regular grid with resolution (N_1, N_2, N_3)
- Apply Fast Fourier Transform (FFT)

$$u(j_1, j_2, j_3) = \frac{1}{N_1 N_2 N_3} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} \sum_{k_3=0}^{N_3-1} \hat{u}(k_1, k_2, k_3)\, e^{2\pi i \left(\frac{j_1 k_1}{N_1} + \frac{j_2 k_2}{N_2} + \frac{j_3 k_3}{N_3}\right)}$$

$$\hat{u}(k_1, k_2, k_3) = \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} \sum_{j_3=0}^{N_3-1} u(j_1, j_2, j_3)\, e^{-2\pi i \left(\frac{j_1 k_1}{N_1} + \frac{j_2 k_2}{N_2} + \frac{j_3 k_3}{N_3}\right)}$$

Spectral method

- Differential operators become componentwise multiplication:

$$\widehat{\frac{\partial}{\partial x} u}(\mathbf{k}) = p_1 \hat{u}(\mathbf{k}) \qquad p_1 = \begin{cases} \frac{2\pi \mathbb{i} k_1}{L_1} & 0 \leq k_1 < N_1/2 \\ 0 & k_1 = N_1/2 \\ \frac{2\pi \mathbb{i} (k_1 - N_1)}{L_1} & k_1 > N_1/2 \end{cases}$$

- Poisson solve also becomes componentwise multiplication.

Spectral method: obstacle treatment

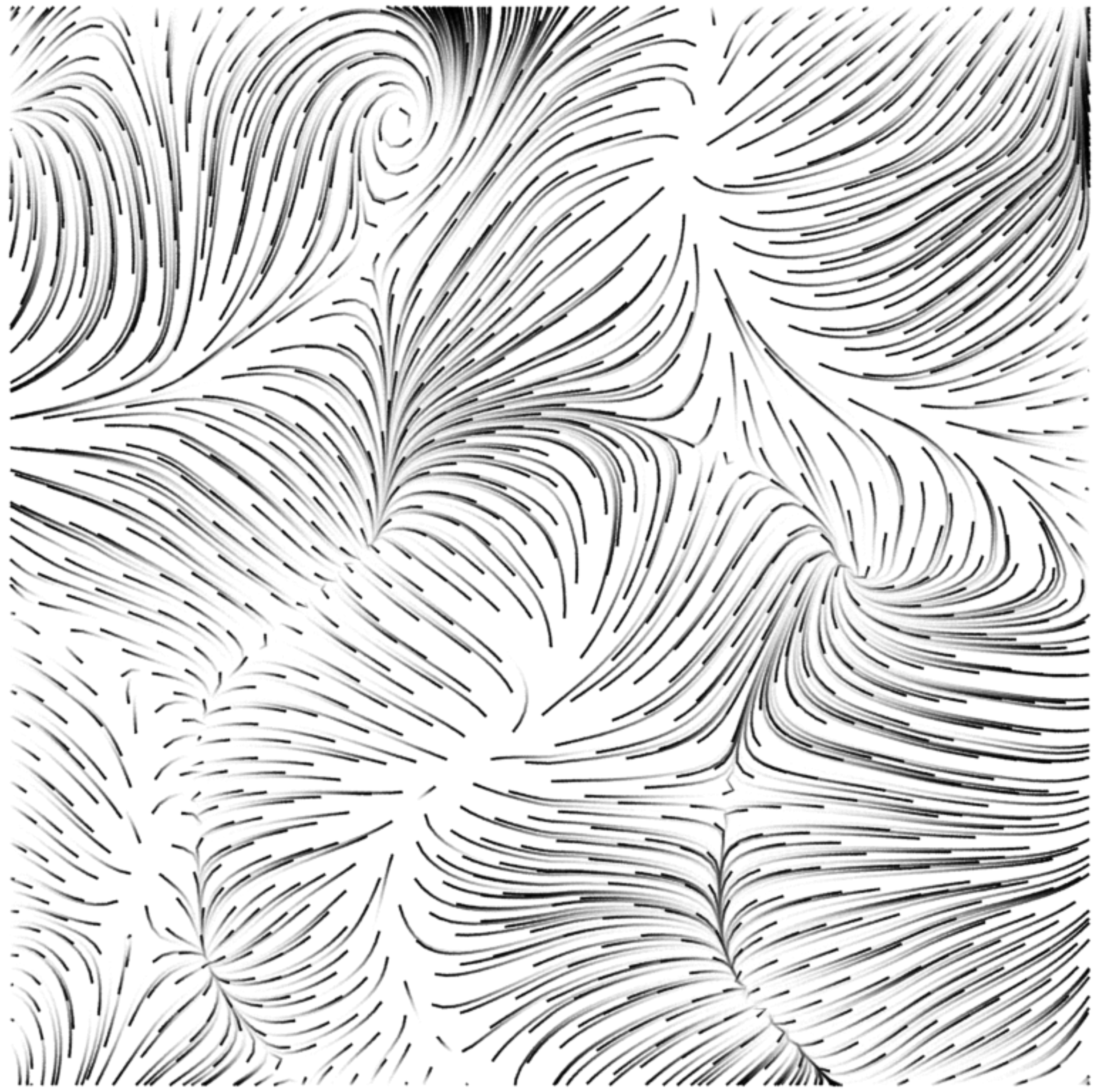
- FFT only works on periodic domain.
- To add obstacles, one can repeat the following
  - Pressure project
  - Set velocity in obstacle to zero
- It is an alternating orthogonal projection to two linear subspaces: the div-free subspace, and the zero-in-obstacle subspace.
- The alternating projection will converge to their intersection.

Biot–Savart integral

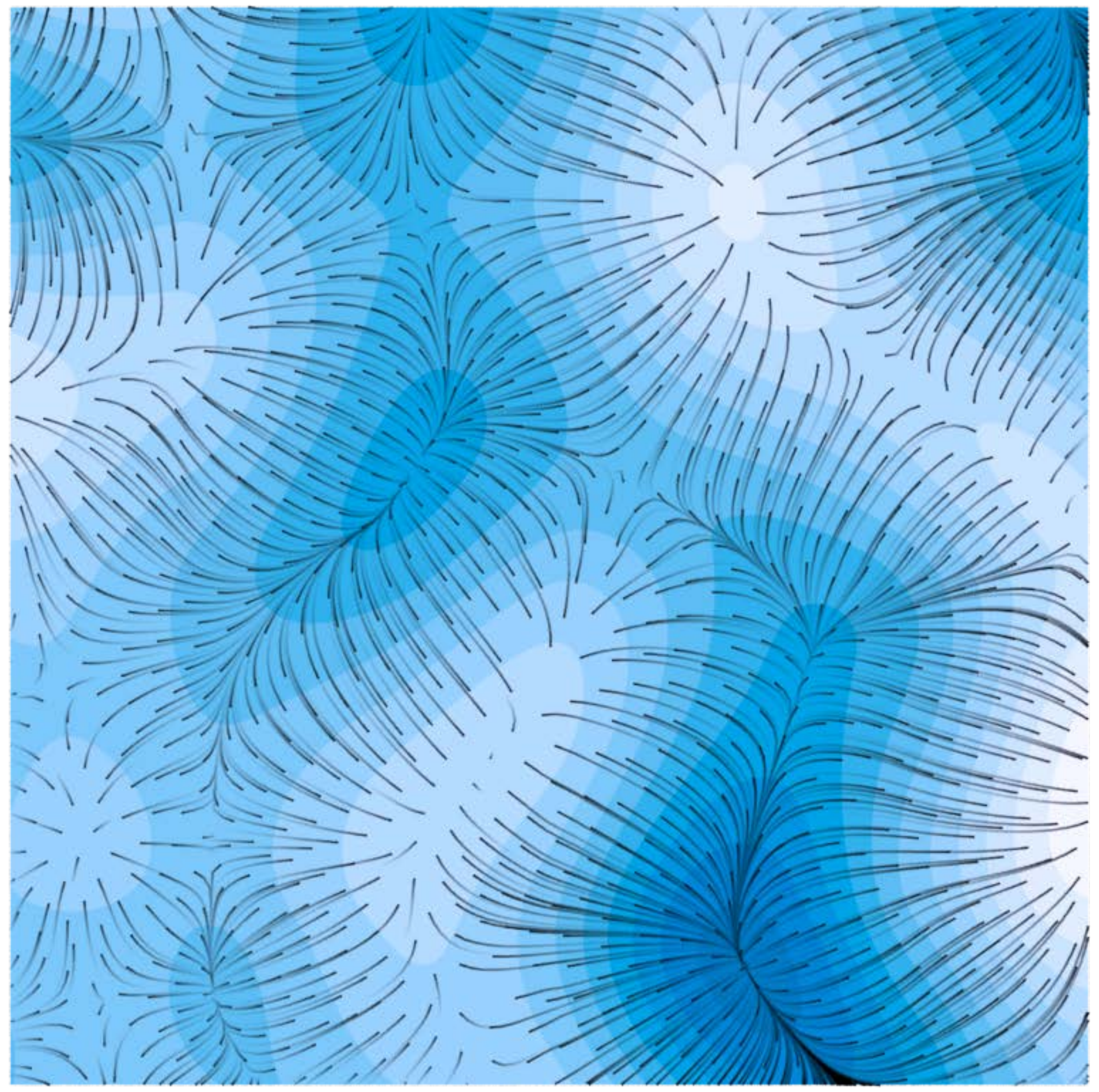
$$\Gamma(T\mathbb{R}^n) = (\text{curl-free VF}) \oplus (\text{div-free VF})$$

$$\text{(curl-free VF)} = \mathrm{im}(\mathrm{grad})$$

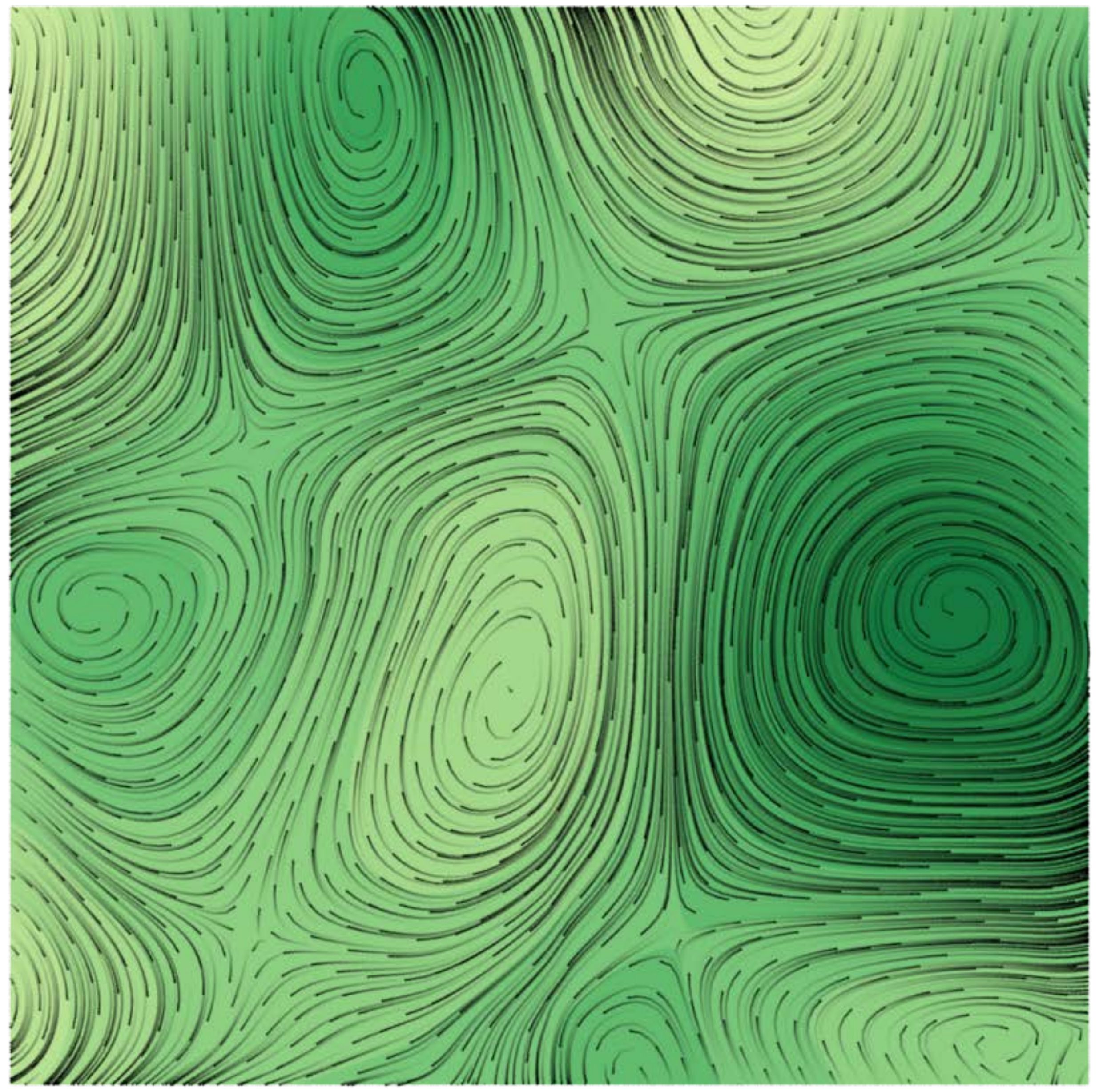
$$\text{(div-free VF)} = \mathrm{im}(\mathrm{curl}) \text{ in 3D}, \quad \mathrm{im}(R^{90^\circ}\,\mathrm{grad}) \text{ in 2D}$$



$=$



$+$



- Directly compute the divergence-free part

Biot–Savart integral

- Let $\mathbf{w}$ be the vorticity field.
- If the fluid domain is the entire $\mathbb{R}^3$

  then there is a unique divergence-free velocity, decay to zero at infinity:

$$\mathbf{u}(\mathbf{x}) = \frac{1}{4\pi}\int_{\mathrm{supp}(\mathbf{w})} \frac{\mathbf{w}(\mathbf{y}) \times (\mathbf{x}-\mathbf{y})}{|\mathbf{x}-\mathbf{y}|^3}\, dV_{\mathbf{y}}$$

- For 2D

$$\mathbf{u}(\mathbf{x}) = \frac{1}{2\pi}\int_{\mathrm{supp}(\mathbf{w})} \frac{w(\mathbf{y})R^{90^\circ}(\mathbf{x}-\mathbf{y})}{|\mathbf{x}-\mathbf{y}|^2}\, dA_{\mathbf{y}}$$

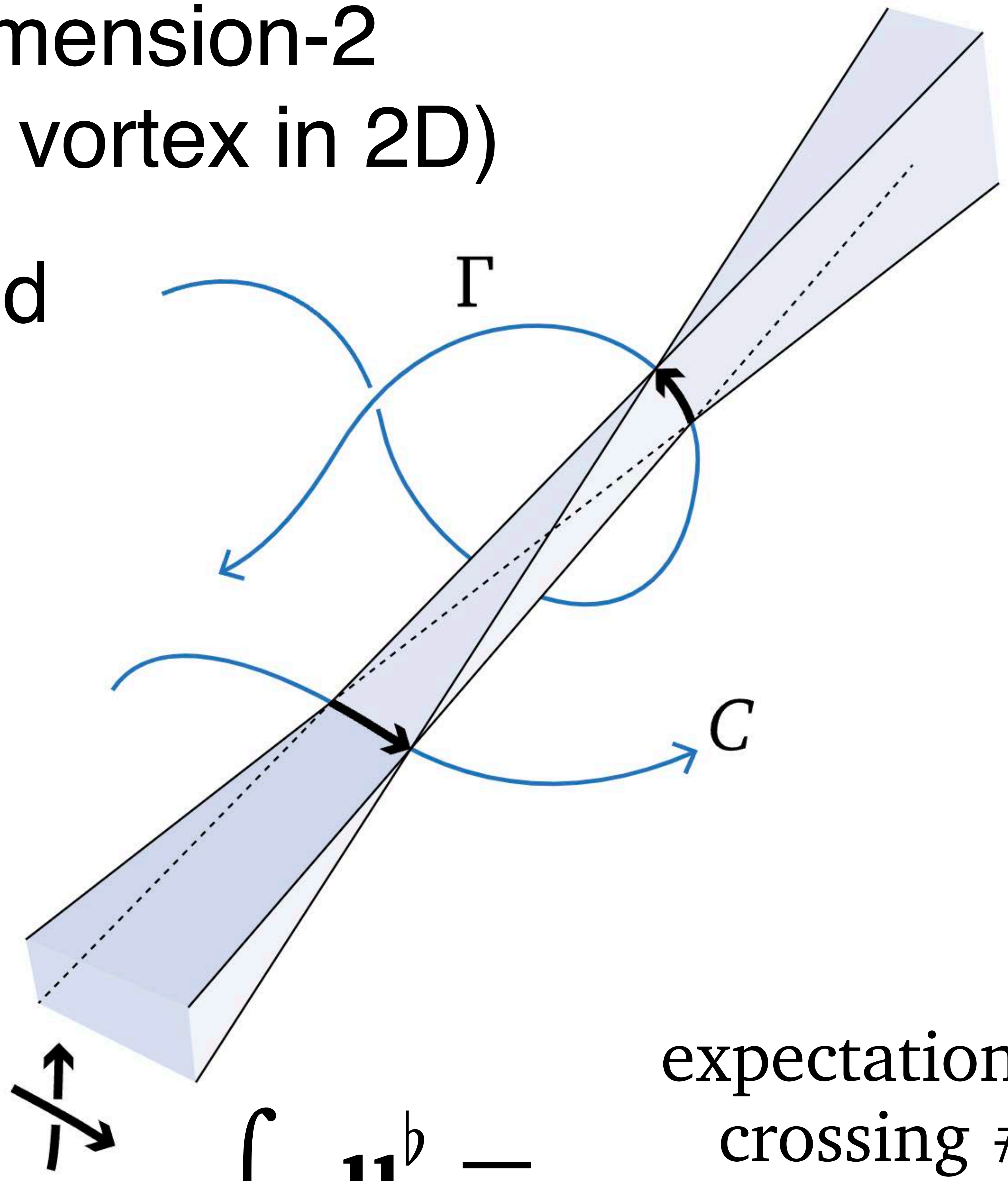
Biot–Savart integral

- When vorticity is concentrated into codimension-2 geometry (vortex filament in 3D or point vortex in 2D)
  - Filament must be closed, with a fixed strength $\kappa = \oint_{\substack{\text{any loop}\\ \text{around } \Gamma}} \mathbf{u}^\flat$

$$\mathbf{u}(\mathbf{x}) = \frac{\kappa}{4\pi} \int_\Gamma \frac{\mathbf{T}(\mathbf{y}) \times (\mathbf{x}-\mathbf{y})}{|\mathbf{x}-\mathbf{y}|^3} \, ds_{\mathbf{y}}$$

  - In 2D:

$$\mathbf{u}(\mathbf{x}) = \sum_i \frac{\kappa_i}{2\pi} \frac{R^{90^\circ}(\mathbf{x}-\mathbf{y})}{|\mathbf{x}-\mathbf{y}|^2}$$



$\int_C \mathbf{u}^\flat =$ expectation of crossing # from a random orthogogrpahic view

Biot–Savart integral: obstacle treatment

- If there are obstacles, the Biot–Savart integral does not obey the no-through boundary condition.
- In that case modify the Biot–Savart field with a gradient field i.e. a pressure projection.
    - Solve the Laplace problem in the outer domain with Neumann boundary condition.
    - Classically this is done with **boundary element method**.
    - One can also turn this into an interior domain problem by a **Kelvin transformation**.

Wikipedia: Kelvin transform

[Nabizadeh, Ramamoorthi, Chern 2021 "Kelvin tranform for simulations in infinite domain"]

Stream function

- To reconstruct velocity from vorticity, make the ansatz

$$\mathbf{u} = \nabla \times \Psi \qquad \text{in 3D}$$

$$\mathbf{u} = -R^{90^\circ} \nabla \psi \qquad \text{in 2D}$$

- This stream-function (vector potential) satisfies

$$\nabla \times \nabla \times \Psi = \mathbf{w}$$

- To make the equation having unique solution:

$$(\nabla \times \nabla \times - \nabla \nabla \cdot)\Psi = \mathbf{w} \qquad -\Delta \Psi = \mathbf{w}$$

and Ψ is normal to the boundary

Stream function

- To reconstruct velocity from vorticity, make the ansatz

$$\mathbf{u} = \nabla \times \Psi \qquad \text{in 3D}$$
$$\mathbf{u} = -R^{90^\circ} \nabla \psi \qquad \text{in 2D}$$

- In 2D $$-\Delta \psi = w$$
$$\psi = 0 \quad \text{on the boundary}$$

- Beware the result could be incorrect if there are multiple boundary components, or that the domain is non-simply-connected.

[Yin, Nabizadeh, Wu, Wang, Chern 2023 "Fluid cohomology"]

Helmholtz–Hodge–Morrey–Friedrichs

- In general, we have the following orthogonal decomposition

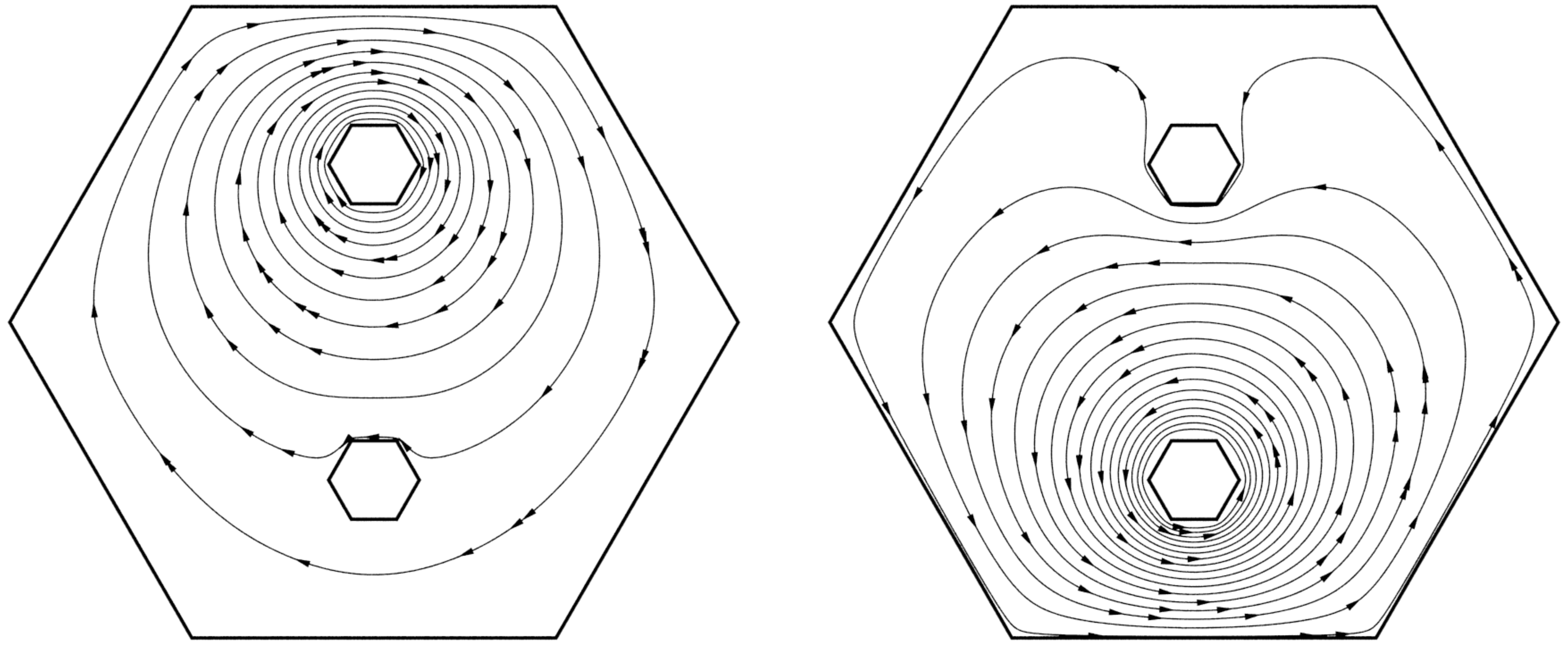
$$\tilde{\mathbf{u}} = \nabla p + \nabla \times \Psi + \mathbf{h} \qquad \text{in 3D}$$

$$\tilde{\mathbf{u}} = \nabla p - R^{90^\circ} \nabla \psi + \mathbf{h} \qquad \text{in 2D}$$

with Ψ normal to the boundary

$\psi = 0$ on the boundary

$$\begin{cases} \nabla \cdot \mathbf{h} = 0 \\ \nabla \times \mathbf{h} = 0 \\ \mathbf{n} \cdot \mathbf{h} = 0 \qquad \text{on boundary} \end{cases}$$



Time Splitting

- Pressure projection
- Time splitting
- Advection
- Particle in cell
- Vortex methods
- Lattice Boltzmann methods

Time splitting

Incompressible Euler equation

$$\begin{cases} \dfrac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p \\ \nabla \cdot \mathbf{u} = 0 \end{cases}$$

- For each time step:
  - Solve advection equation $\dfrac{\partial \mathbf{u}}{\partial t} + \nabla_{\mathbf{u}} \mathbf{u} = 0$
  - Pressure project

Time splitting (Strang splitting)

Incompressible Euler equation

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p \\ \nabla \cdot \mathbf{u} = 0 \end{cases}$$

- For each time step:
  - Solve advection equation $\frac{\partial \mathbf{u}}{\partial t} + \nabla_{\mathbf{u}} \mathbf{u} = 0$ for half timestep
  - Pressure "reflect"
  - Solve advection equation $\frac{\partial \mathbf{u}}{\partial t} + \nabla_{\mathbf{u}} \mathbf{u} = 0$ for half timestep
  - Pressure project

Wikipedia: Strang splitting

[Zehnder, Narain, Thomaszewski 2018 "Advection-Reflection Solver"]

Advection

- Pressure projection
- Time splitting
- Advection
- Particle in cell
- Vortex methods
- Lattice Boltzmann methods

Advection

- Given a vector field $\mathbf{v}$ representing the velocity of a flow
- and a quantity α we want to advect
- We want to evolve α for a short time

$$\frac{\partial}{\partial t}\alpha + \underset{\langle \text{type} \rangle}{\mathscr{L}_{\mathbf{v}}}\, \alpha = 0$$

- **Eulerian method:** use pullback operator.
- **Lagrangian method:** set quantities on particles (or other geometry), and let particle move by $\mathbf{v}$.

  Lagrangian advection is essentially exact, but hard to do other things such as pressure solve

  Eulerian advection's pullback often requires interpolation, but easier for solving pressure

Eulerian advection (semi Lagrangian)

$$\frac{\partial}{\partial t}\alpha + \underset{\langle\text{type}\rangle}{\mathscr{L}_{\mathbf{v}}}\,\alpha = 0$$

- Suppose we are given α_t and want to find $\alpha_{t+\Delta t}$
- Use $\mathbf{v}$ to generate an inverse flow map, called "back trace"

$$\psi(\mathbf{x}) = \mathbf{x} - \Delta t\,\mathbf{v}|_{\mathbf{x}} \qquad \text{or} \qquad \psi(\mathbf{x}) = \mathrm{RK4}(-\mathbf{v}; \mathbf{x}; \Delta t)$$

(time span: Δt; ODE func: $-\mathbf{v}$; initial condition: $\mathbf{x}$)

- Then $\alpha_{t+\Delta t} = \underset{\langle\text{type}\rangle}{\psi^*}\,\alpha_t$

- This is also called the upwind advection scheme.

Eulerian advection (semi Lagrangian)

- Study the upwind scheme for 0-form in 1D (Blackboard)

$$\frac{\partial}{\partial t} q + v \frac{\partial}{\partial x} q = 0 \qquad v > 0 \quad \text{constant in } t \text{ and } x$$

  - Exact solution: translation $q(t, x) = q(0, x - tv)$
  - Derive the **modified equation**
  - Derive the CFL condition (Courant–Friedrichs–Lewy)

Eulerian advection (semi Lagrangian)

- Back-and-forth error correction and compensation (BFECC)
  - Suppose $\mathscr{A}^{\mathrm{sL}}$ is a semi-Lagrangian advection

$$\mathscr{A}^{\mathrm{sL}}_{\mathbf{v},\Delta t}\alpha \approx \mathscr{D}_{\Delta t}\psi^{*}_{\mathbf{v},\Delta t}\alpha \approx \psi^{*}_{\mathbf{v},\Delta t}\mathscr{D}_{\Delta t}\alpha$$

where $\mathscr{D} \approx e^{\Delta t \varepsilon L} \approx 1 + \Delta t \varepsilon L$ is some diffusion operator.

  - Then a forward-backward advection will extract twice the diffusion operator

$$\mathscr{A}^{\mathrm{sL}}_{-\mathbf{v},\Delta t} \circ \mathscr{A}^{\mathrm{sL}}_{\mathbf{v},\Delta t} \approx \mathscr{D}^2 \approx 1 + 2\Delta t \varepsilon L$$

  - The amount of diffusion is estimated as $\frac{1}{2}(\mathscr{A}^{\mathrm{sL}}_{-\mathbf{v},\Delta t} \circ \mathscr{A}^{\mathrm{sL}}_{\mathbf{v},\Delta t} - 1)$
  - Remove the diffusion

$$\mathscr{A}^{\mathrm{BFECC}}_{\mathbf{v},\Delta t} := \mathscr{A}^{\mathrm{sL}}_{\mathbf{v},\Delta t} - \boxed{(\mathscr{A}^{\mathrm{sL}}_{\mathbf{v},\Delta t})}\frac{1}{2}(\mathscr{A}^{\mathrm{sL}}_{-\mathbf{v},\Delta t} \circ \mathscr{A}^{\mathrm{sL}}_{\mathbf{v},\Delta t} - 1)$$

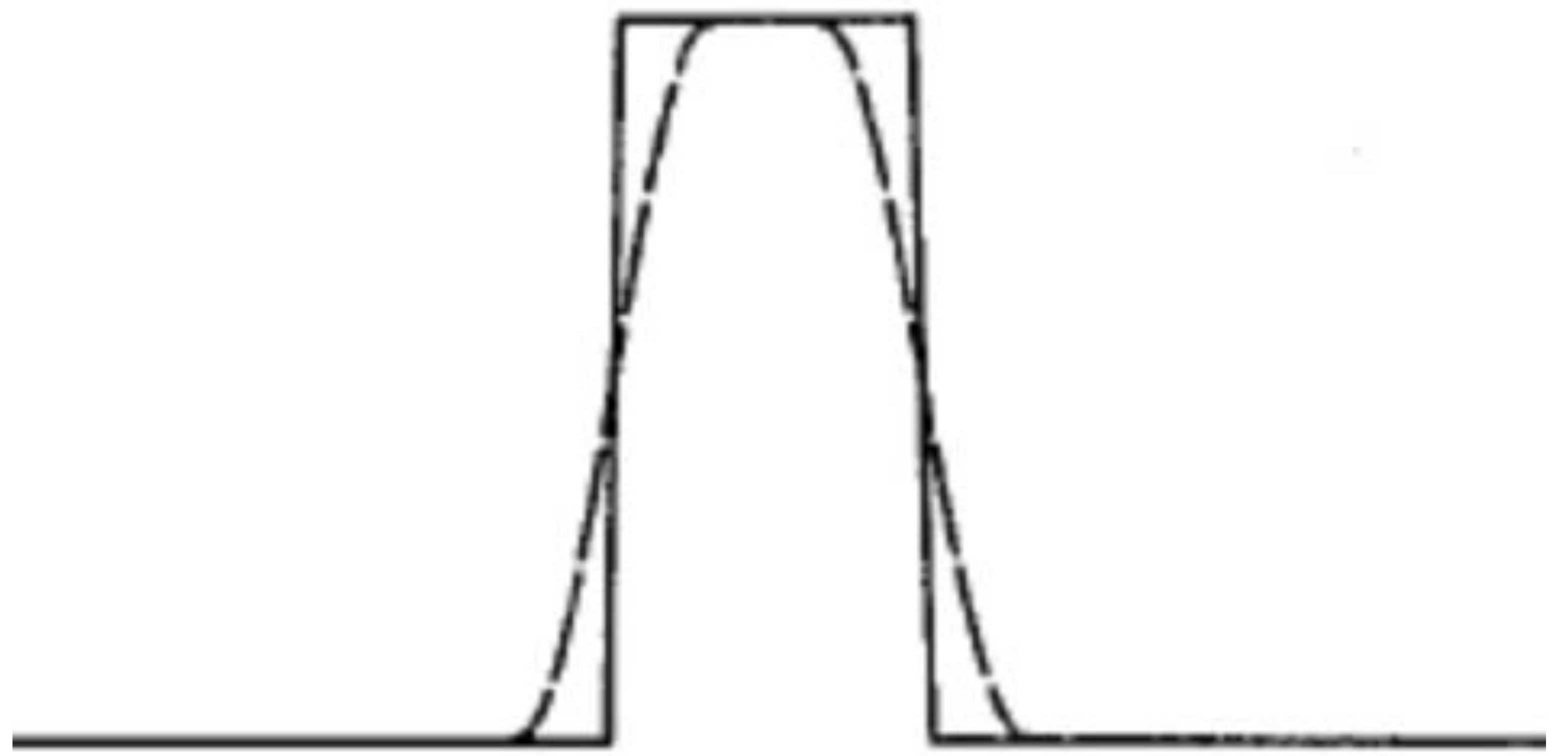
optional. Without it, it's called modified MacCormack.

Eulerian advection (semi Lagrangian)

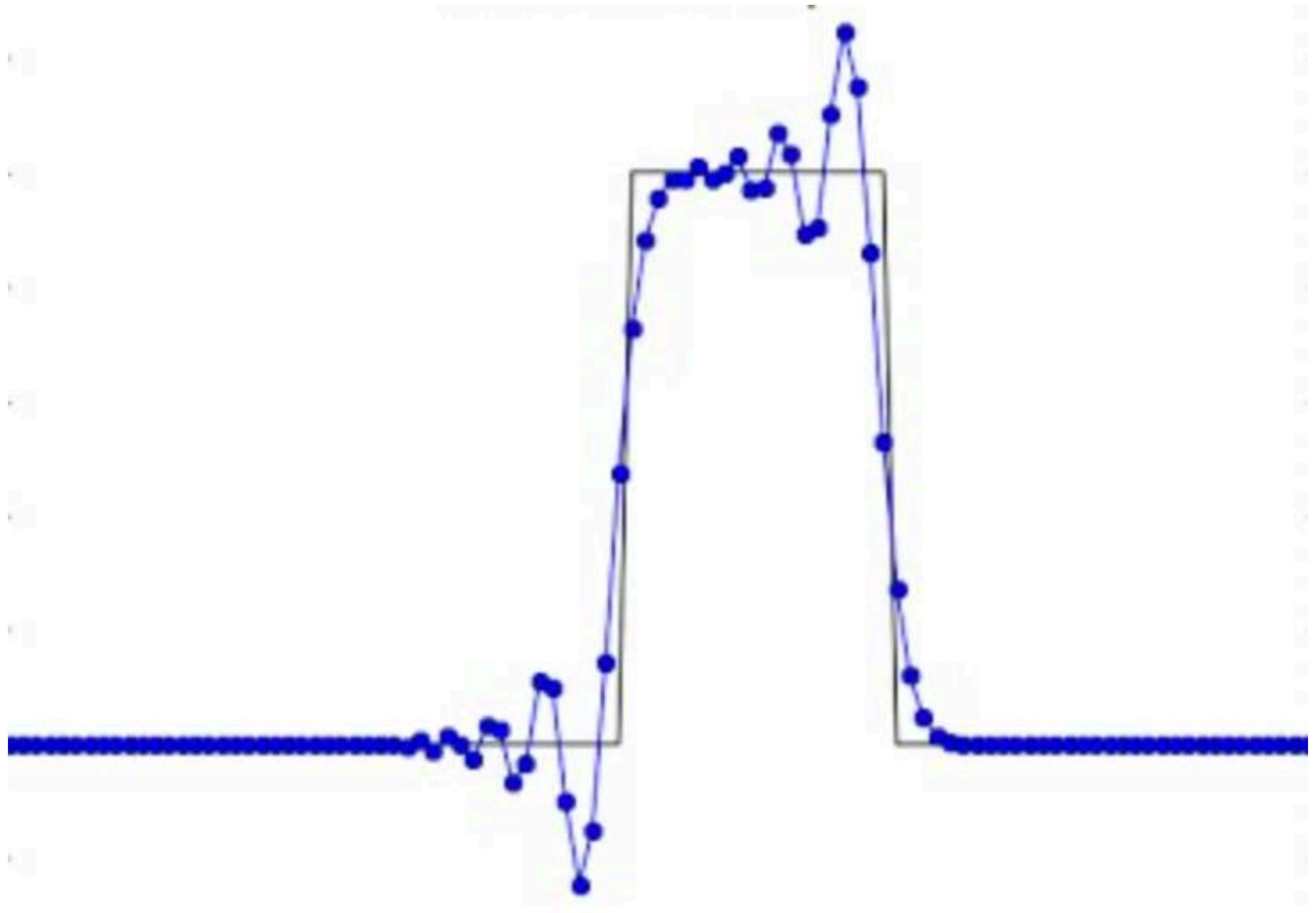
- Back-and-forth error correction and compensation (BFECC)

$$\mathscr{A}^{\mathrm{BFECC}}_{\mathbf{v},\Delta t} := \mathscr{A}^{\mathrm{sL}}_{\mathbf{v},\Delta t} - (\mathscr{A}^{\mathrm{sL}}_{\mathbf{v},\Delta t})\frac{1}{2}(\mathscr{A}^{\mathrm{sL}}_{-\mathbf{v},\Delta t} \circ \mathscr{A}^{\mathrm{sL}}_{\mathbf{v},\Delta t} - 1)$$

  - Sometimes this modification will create spatial oscillation (dispersion error)



numerical dissipation



numerical dispersion

Eulerian advection (semi Lagrangian)

- Back-and-forth error correction and compensation (BFECC)

$$\mathscr{A}^{\mathrm{BFECC}}_{\mathbf{v},\Delta t} := \mathscr{A}^{\mathrm{sL}}_{\mathbf{v},\Delta t} - \left(\mathscr{A}^{\mathrm{sL}}_{\mathbf{v},\Delta t}\right)\frac{1}{2}\left(\mathscr{A}^{\mathrm{sL}}_{-\mathbf{v},\Delta t} \circ \mathscr{A}^{\mathrm{sL}}_{\mathbf{v},\Delta t} - 1\right)$$

  - Sometimes this modification will create spatial oscillation (dispersion error)
  - You can detect and cutoff the overshoots using ***limiter***.
  - Limiter: If the value of the field after BFECC is lying outside of some convex hull of the neighborhood values of stable semi-Lagrangian, blend with or use the stable semi-Lagrangian result

Diffusion can still occur in time splitting

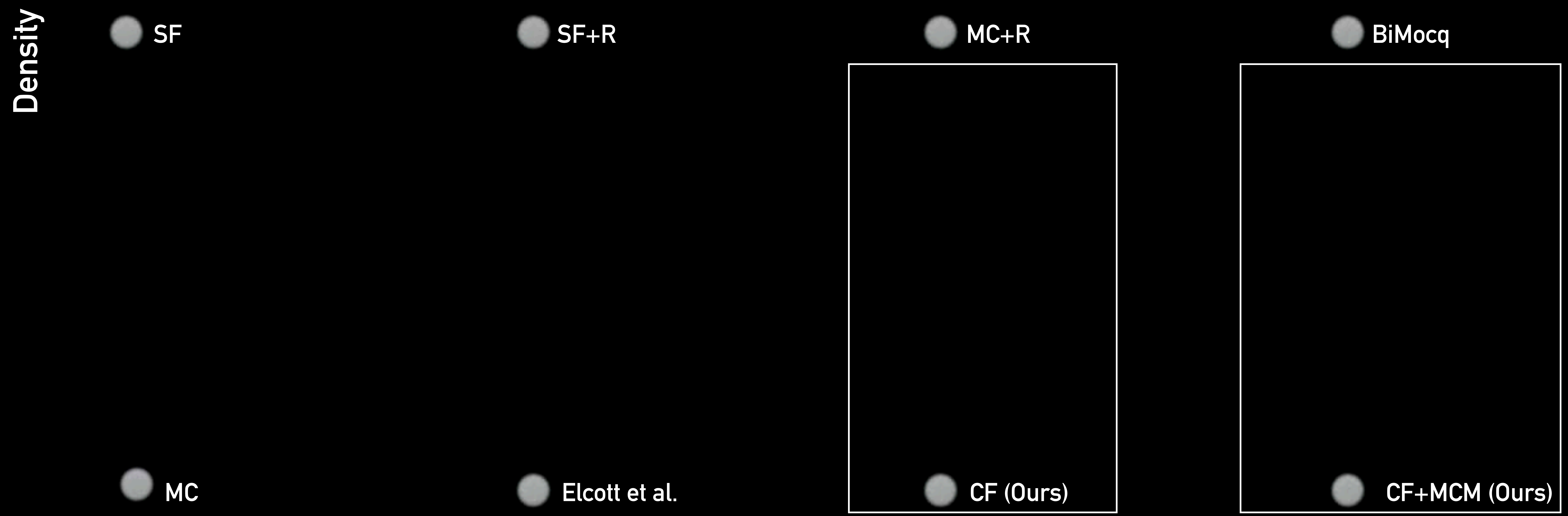
- Even if the advection solver is perfect, the advection–projection time-splitting can still give rise to numerical diffusion

  (blackboard)

- Solution: use **reflection method**, or **covector formulation**

  [Zehnder, Narain, Thomaszewski 2018 "Advection-Reflection Solver"]

  [Nabizadeh, Wang, Ramamoorthi, Chern 2022 "Covector Fluids"]


Density
SF
SF+R
MC+R
BiMocq
MC
Elcott et al.
CF (Ours)
CF+MCM (Ours)

Vorticity

SF

SF+R

MC+R

BiMocq

MC

Elcott et al.

CF (Ours)

CF+MCM (Ours)

Particle in Cell

- Pressure projection
- Time splitting
- Advection
- Particle in cell
- Vortex methods
- Lattice Boltzmann methods

Hybrid of Lagrangian and Eulerian

- Discretize material space as particles $M = \{1, \ldots, \#\text{particles}\}$
  Each particle $p \in M$ has a mass m_p that is fixed over time.
- The particle location is our flow map $\phi : M \to W = \mathbb{R}^3$
- Computing pressure projection is difficult using this particle method (require near-neighbor search)
- Put a grid for the Eulerian coordinate.
- Device an interpolation scheme transferring functions on particle and functions on grid (pullback of zero-form).

Advection

- Denote Eulerian function and its Lagrangian twin

$$f : W \to \mathbb{R}$$
$$\hat{f} : M \to \mathbb{R}$$

$$\hat{f} = \phi^* f = f \circ \phi$$

0-form

- Advection step: Do it in Lagrangian coordinate

$$\frac{\partial}{\partial t}\hat{f} = 0$$
$$\frac{\partial}{\partial t}\phi = \mathbf{u} \circ \phi$$

$$\frac{\partial}{\partial t} f + \mathscr{L}_{\mathbf{u}} f = 0$$

0-form

Much easier
No diffusion

Numerical diffusion

Particle in cell (material point method)

- Beginning of each iteration:
  Each particle has a velocity $\mathbf{v}_p$
  - Advection step
    $$\frac{\partial}{\partial t}\phi = \mathbf{v}$$
    or
    $$\mathbf{u} = \text{P2G}(\mathbf{v})$$
    $$\frac{\partial}{\partial t}\phi = \mathbf{u} \circ \phi$$
  - Transfer to grid
    $$\mathbf{u} = \text{P2G}(\mathbf{v})$$
  - Pressure projection
    $$\mathbf{u} = \text{PressureSolve}(\mathbf{u})$$
  - Transfer back to particle
    $$\mathbf{v} = \text{G2P}(\mathbf{u})$$
- End of iteration

(transfer has lots of diffusion)

FLIP (fluid implicit particle) solver

- Beginning of each iteration:
  Each particle has a velocity $\mathbf{v}_p$
  - Advection step

    $$\frac{\partial}{\partial t}\phi = \mathbf{v}$$

    or

    u = P2G(**v**)

    $$\frac{\partial}{\partial t}\phi = \mathbf{u} \circ \phi$$

  - Transfer to grid

    u = P2G(**v**)

  - Pressure projection

    u’ = PressureSolve(**u**)

  - Transfer the incremental effect of pressure back to particle

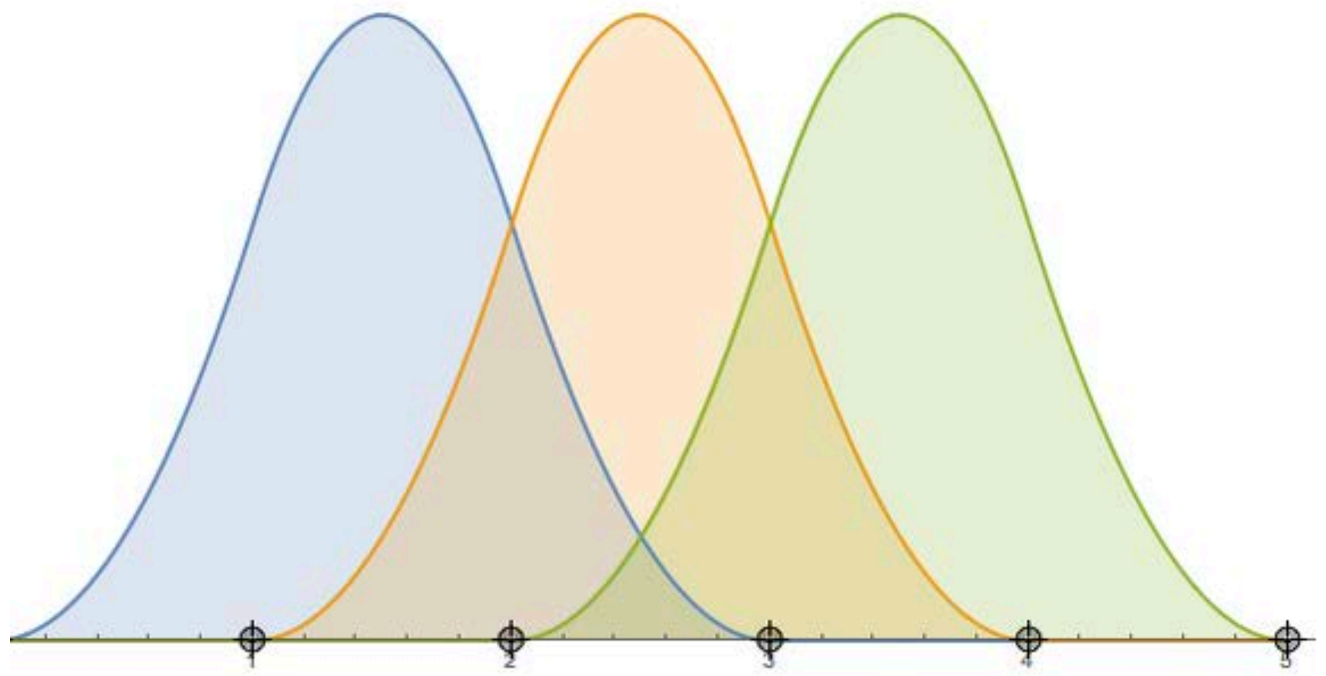
    v += G2P(**u’**-**u**)

- End of iteration

Particle–Grid transfer

- Consider a basis function for each grid point

$$N_i(\mathbf{x}) \qquad i \in \text{Grid} \quad \mathbf{x} \in \mathbb{R}^3$$



  - Local: $N_i(\mathbf{x}) = 0$ when $\mathbf{x}$ is far away from the grid point i.
  - Partition of unity (add up to 1)

$$\sum_{i \in \text{Grid}} N_i(\mathbf{x}) = \mathbb{1}(\mathbf{x}) = 1$$

  - (Nonnegative $N_i(\mathbf{x}) \geq 0$)

- Usually it's just the product of B-spline basis from each dimension
- Lagrange interpolation basis is also a candidate choice

Particle–Grid transfer

- Grid to Particle

$$\hat{f}_p \approx \sum_{i \in \text{Grid}} f_i N_i(\phi_p)$$

- Particle to Grid

$$f_i \approx \frac{\sum_{p \in M} \hat{f}_p m_p N_i(\phi_p)}{\sum_{p \in M} m_p N_i(\phi_p)}$$

Affine Particle–Grid transfer

- Affine Grid to Particle

$$\hat{f}_p \approx \sum_{i \in \text{Grid}} f_i N_i(\phi_p)$$

$$\widehat{\nabla f_p} \approx \underset{\mathbf{A}}{\operatorname{argmin}} \sum_{i \in \text{Grid}} \left| \frac{\sum_{p \in M} (\hat{f}_p + \mathbf{A}[\![\mathbf{x}_i - \phi_p]\!]) m_p N_i(\phi_p)}{\sum_{p \in M} m_p N_i(\phi_p)} - f_i \right|^2$$

- Affine Particle to Grid

$$f_i \approx \frac{\sum_{p \in M} (\hat{f}_p + \widehat{\nabla f_p}[\![\mathbf{x}_i - \phi_p]\!]) m_p N_i(\phi_p)}{\sum_{p \in M} m_p N_i(\phi_p)}$$

Affine Particle–Grid transfer

- Affine Grid to Particle

$$\hat{f}_p \approx \sum_{i \in \mathrm{Grid}} f_i N_i(\phi_p)$$

$$\widehat{\nabla f_p} \approx \mathbf{B}\mathbf{D}^{-1} \qquad \mathbf{B} = \sum_{i \in \mathrm{Grid}} N_i(\phi_p) f_i (\mathbf{x}_i - \phi_p)^\mathsf{T}$$

$$\mathbf{D} = \sum_{i \in \mathrm{Grid}} N_i(\phi_p)(\mathbf{x}_i - \phi_p)(\mathbf{x}_i - \phi_p)^\mathsf{T}$$

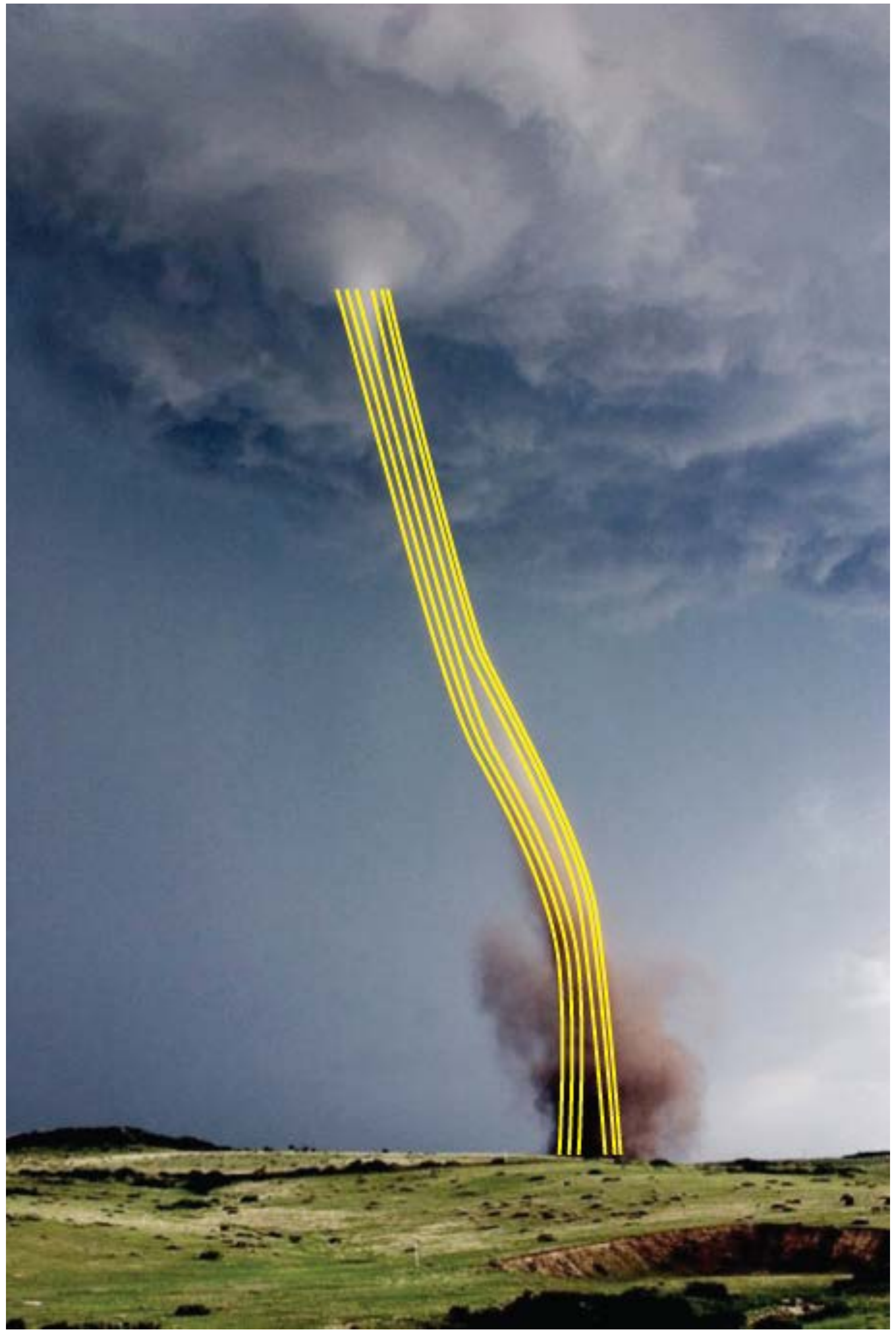
- Affine Particle to Grid

$$f_i \approx \frac{\sum_{p \in M} (\hat{f}_p + \widehat{\nabla f_p} [\![\mathbf{x}_i - \phi_p]\!]) m_p N_i(\phi_p)}{\sum_{p \in M} m_p N_i(\phi_p)}$$

Vortex methods

- Pressure projection
- Time splitting
- Advection
- Particle in cell
- Vortex methods
- Lattice Boltzmann methods

Vortex in Fluids



Vortex in Fluids

Vortex in Fluids

Vorticity equation

$$\frac{\partial \omega}{\partial t} + \mathscr{L}_{\mathbf{u}} \underset{\text{2-form}}{\omega} = 0$$

$$\mathbf{u} = \mathrm{BiotSavart}(\omega)$$

- Pullbacks by flow maps preserve singularities.
- Singularities of a k-form are generically $(n-k)$-dimensional submanifolds.
- Vorticity in 3D: ***vortex filaments***.
- Vorticity in 2D: ***point vortices***

- (Vortex particle method in 3D makes no sense)

2D Vortex particle system

- Each particle has a fixed vorticity strength κ_p
  - Strength can be positive or negative.
- The 2D position of each particle

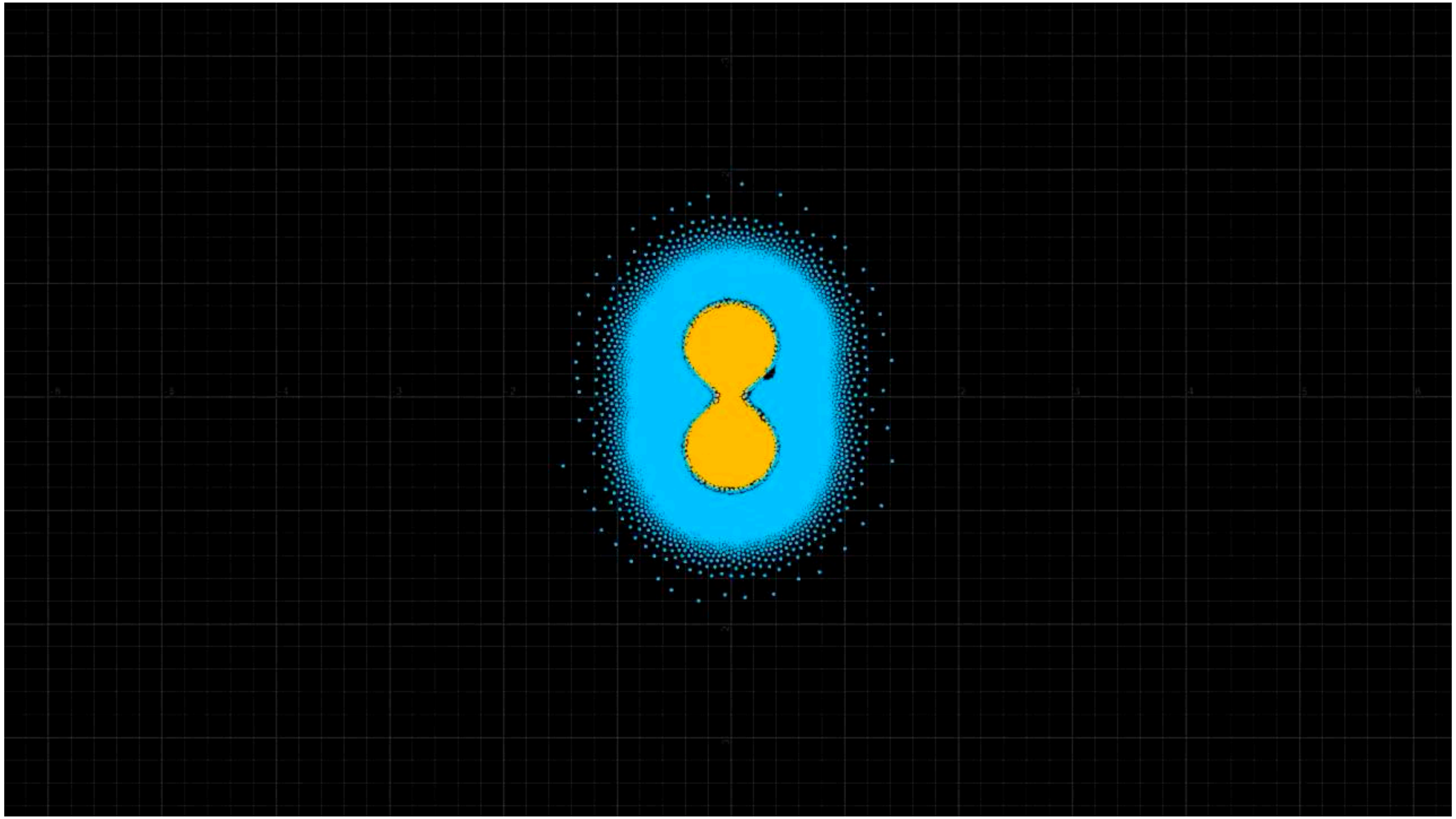
$$\frac{d}{dt}\boldsymbol{\phi}_p = \sum_{\substack{q \in \text{VortexParticles} \\ q \neq p}} \frac{\kappa_q}{2\pi} \frac{\text{Rotate}^{90^\circ}(\boldsymbol{\phi}_p - \boldsymbol{\phi}_q)}{|\boldsymbol{\phi}_p - \boldsymbol{\phi}_q|^2}$$

- Also evaluate velocity for any passively advected smoke/dye by

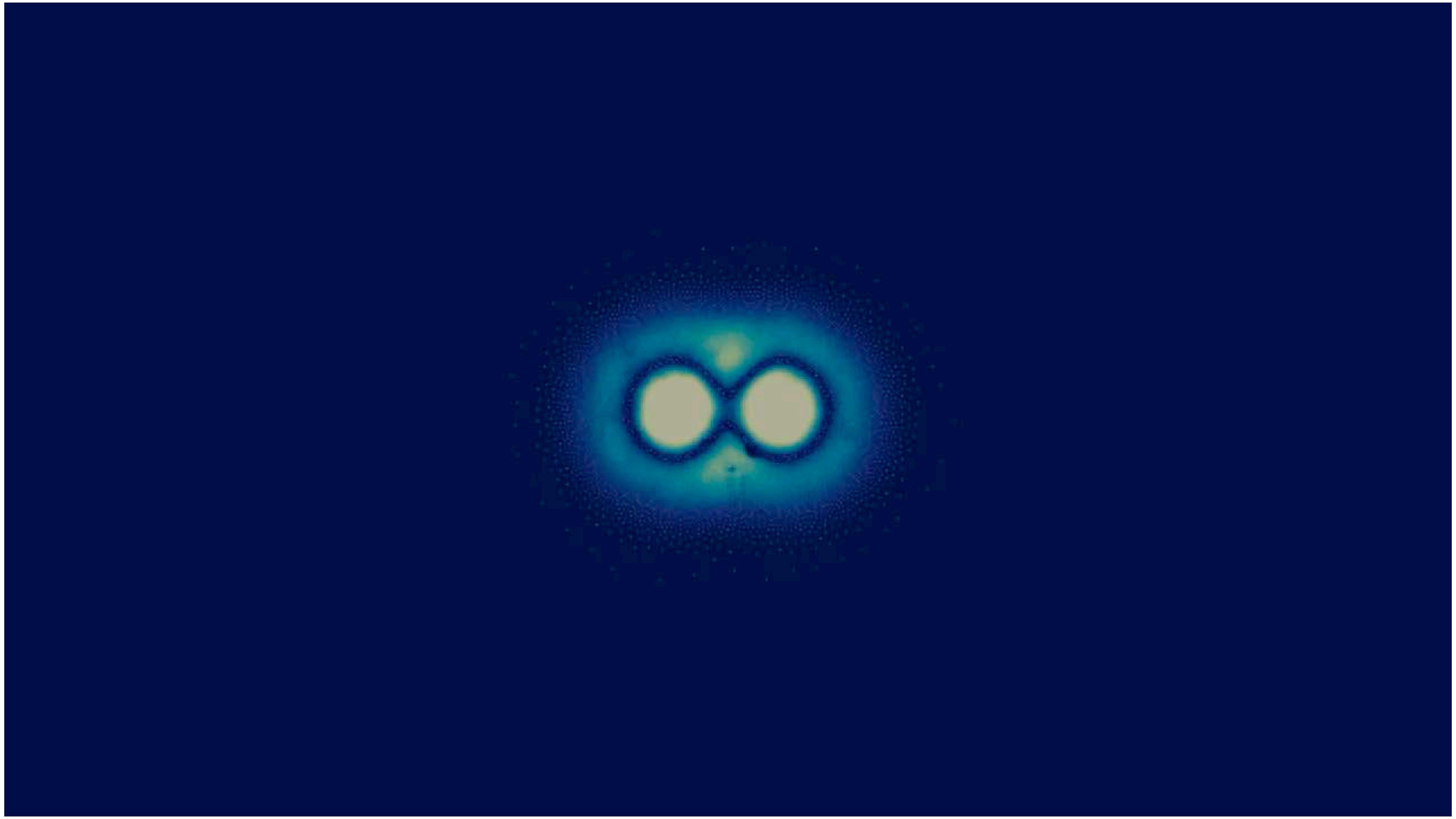
$$\mathbf{v}_{\mathbf{x}} = \sum_{q \in \text{VortexParticles}} \frac{\kappa_q}{2\pi} \frac{\text{Rotate}^{90^\circ}(\mathbf{x} - \boldsymbol{\phi}_q)}{|\mathbf{x} - \boldsymbol{\phi}_q|^2}$$

- Solve ODE by RK4

2D Vortex particle system



2D Vortex particle system

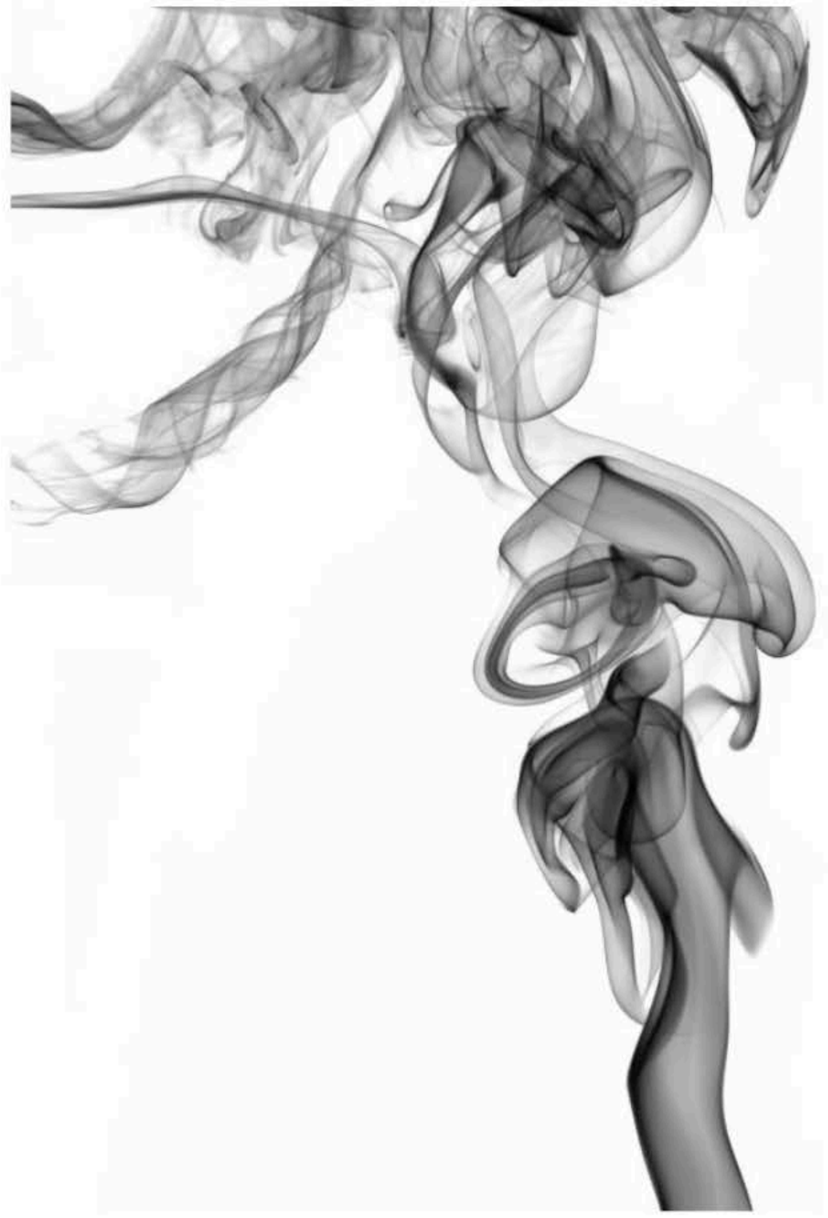


3D vortex filament

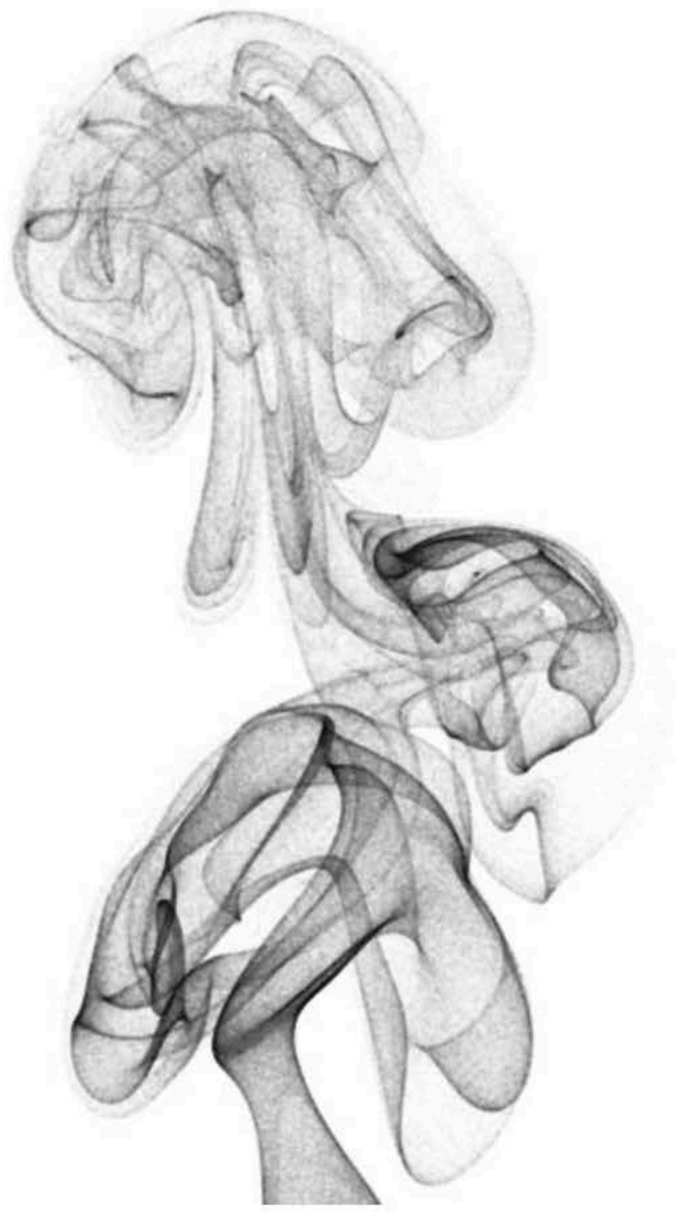
- Filament based smoke (see e.g. Weissmann 2010)

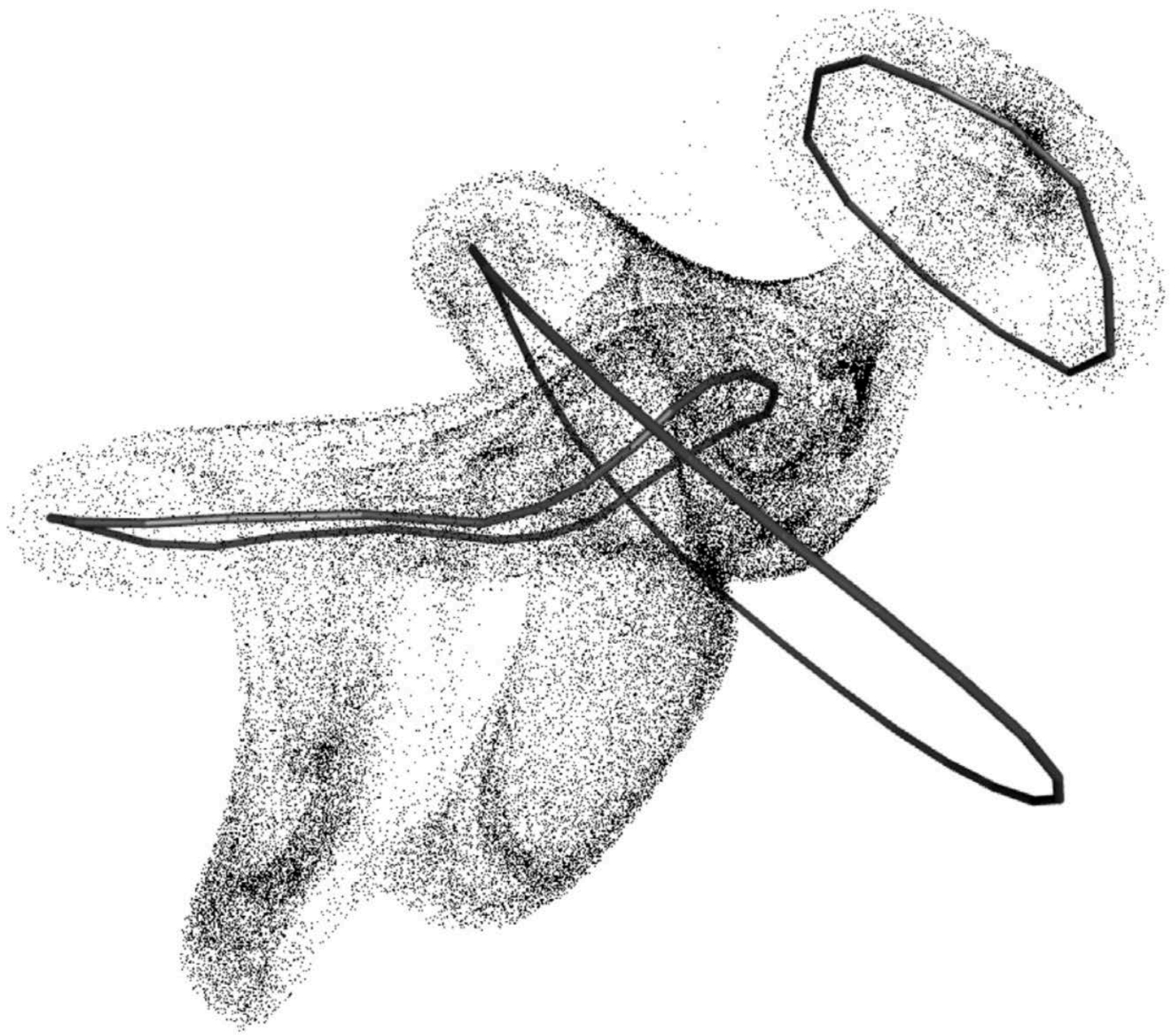
  http://diglib.eg.org/handle/10.2312/8252

photograph



simulation





3D vortex filament

- Limitation: Filament length can grow exponentially, need reconnection, which is hard
  - Variants: “Vortex segment clouds” [Xiong 2021]
  - Variants: “Implicit vortex filaments” [Ishida 2022]
  - (MPM + vortex segment?)

3D vorticity equation

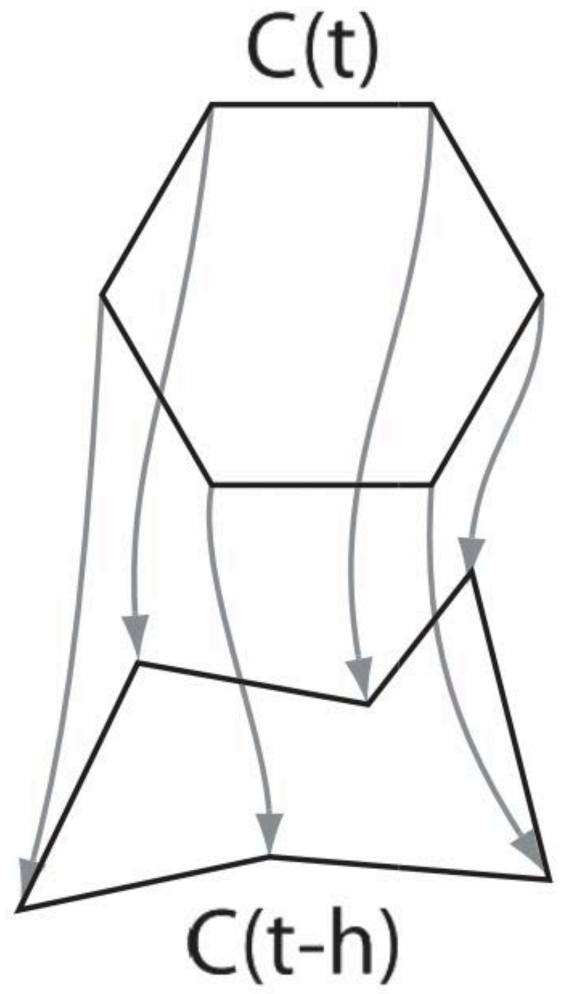
$$\frac{\partial \omega}{\partial t} + \underset{\text{2-form}}{\mathscr{L}_{\mathbf{u}}\ \omega} = 0$$

$$\mathbf{u} = \mathrm{BiotSavart}(\omega)$$

- Most papers solve it by splitting advection and stretching

$$\frac{\partial \mathbf{w}}{\partial t} + \nabla_{\mathbf{u}}\mathbf{w} = \nabla_{\mathbf{w}}\mathbf{u}$$

- Better way: Do a semi-Lagrangian pullback for 2-form [Elcott et al 2007 "Stable and circulation preserving simplicial fluids"]



Lattice Boltzmann Method

- Pressure projection
- Time splitting
- Advection
- Particle in cell
- Vortex methods
- Lattice Boltzmann methods

Statistical model

- Continuum mechanics is only a macroscopic model as a global deformable body
  - Each world-point has one averaged velocity
- Mesoscopic (kinetic, statistical) model
  - Each world-point has a probability distribution of velocities
- Microscopic: Individual molecules
  - Each of 10^23 of molecules has its velocity

Statistical model

- The state of the system is a probability measure $f(\mathbf{x}, \mathrm{v}, t)$ over the tangent bundle TW

- The mass density $\rho_{(\mathbf{x},t)} = \int_{\mathrm{v} \in T_\mathbf{x} W} f(\mathbf{x}, \mathrm{v}, t)$

- The momentum density $(\rho \mathbf{u})_{(\mathbf{x},t)} = \int_{\mathrm{v} \in T_\mathbf{x} W} \mathrm{v} f(\mathbf{x}, \mathrm{v}, t)$

- Internal energy $e_{(\mathbf{x},t)} = \int_{\mathrm{v} \in T_\mathbf{x} W} \frac{1}{2} |\mathrm{v} - \mathbf{u}|^2 f(\mathbf{x}, \mathrm{v}, t)$

$$= \frac{3kT_{(\mathbf{x},t)}}{2}$$

- Stress-energy tensor $\Sigma_{(\mathbf{x},t)} = \int_{\mathrm{v} \in T_\mathbf{x} W} (\mathrm{v} - \mathbf{u})(\mathrm{v} - \mathbf{u})^\mathsf{T} f(\mathbf{x}, \mathrm{v}, t)$

Boltzmann equation

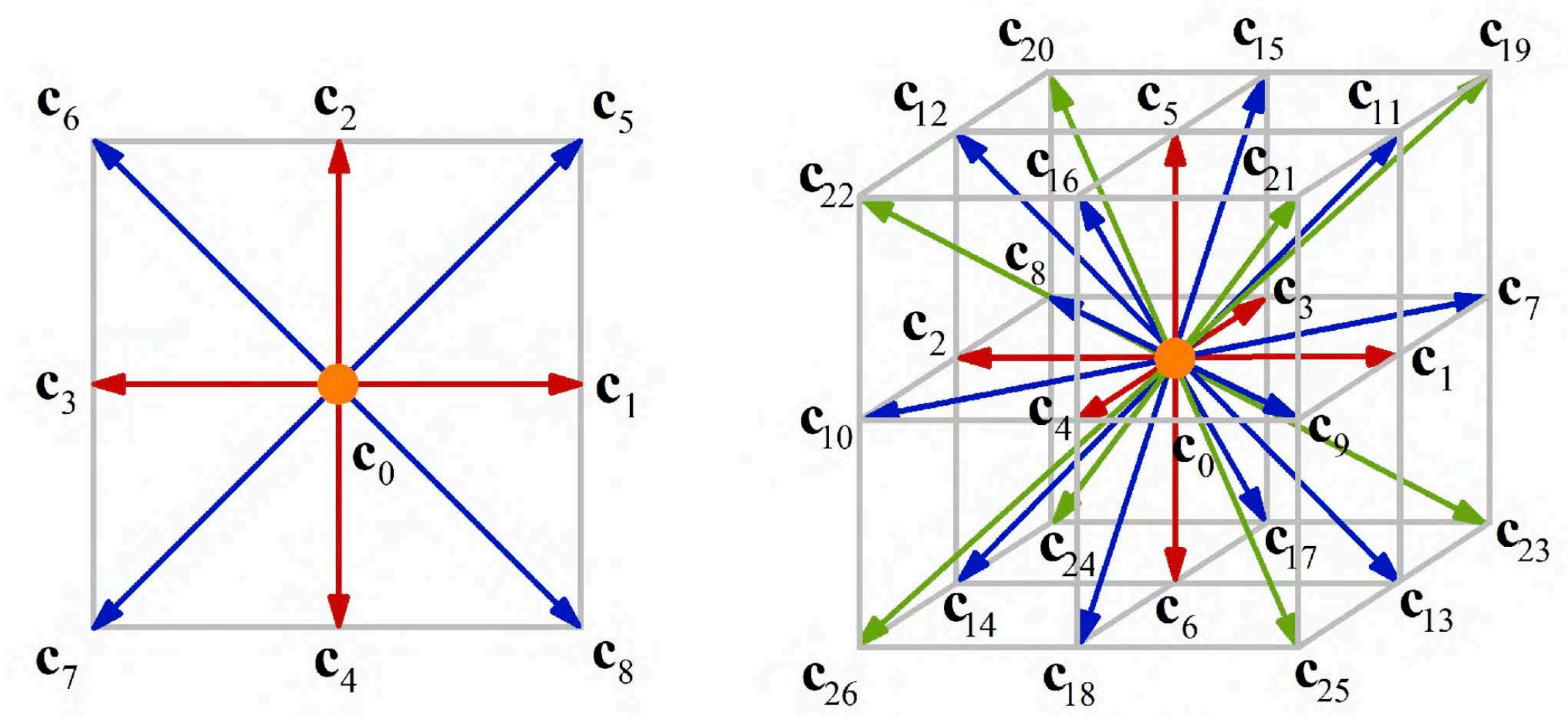
- Boltzmann equation

$$\frac{\partial}{\partial t} f(\mathbf{x}, \mathrm{v}, t) + \mathbf{v} \cdot \frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}, \mathrm{v}, t) + \mathbf{g}^{\mathrm{ext}} \cdot \frac{\partial}{\partial \mathrm{v}} f(\mathbf{x}, \mathrm{v}, t) = Q(f)_{(\mathbf{x}, \mathrm{v}, t)}$$

  - The probability distribution is advected by its own velocity
  - There can be pointwise operator Q modeling the internal collision that mutates the distribution (obeying some conservation law)
  - There can also be external force that accelerate velocity

Lattice Boltzmann method

- Discretize the tangent bundle as lattice with discrete velocity directions.



- Probability density are numbers assigned on each of these 27 x (#grid pts) stencils
- Advection: exact exchange of probability (because velocity points onto another lattice grid)
- Model collision operator, with physics and numerical consideration