A Fully Lagrangian Advection Scheme

M. Ali Yassaei, John C. Bowman, and Anup Basu
University of Alberta

May 19, 2006

www.math.ualberta.ca/∼bowman/talks

Outline

- 2D Advection–Diffusion
- Passive Advection
- Casimir Invariants
- Lagrangian Rearrangement
  - Weighted Bresenham Algorithm
- Average Complexity
- Operator Splitting
  - Diffusion
  - Self-advection
- Energy Decay Rate
- Conclusions

Introduction

- 2D advection–diffusion:

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{v}\cdot\boldsymbol{\nabla}\boldsymbol{U} = \boldsymbol{D}\nabla^2\boldsymbol{U}.$$

- $\boldsymbol{U} = (\omega, C)$ represents
  - Scalar vorticity $\omega = \hat{\boldsymbol{z}}\cdot\boldsymbol{\nabla}\times\boldsymbol{v}$,
  - Concentration field C.
- The velocity $\boldsymbol{v}$ is incompressible: $\boldsymbol{\nabla}\cdot\boldsymbol{v} = 0$.
- Diffusion matrix $\boldsymbol{D} = \mathrm{diag}(\nu, D)$:

  $\nu =$ fluid viscosity,

  $D =$ diffusion constant for concentration field.

Eulerian vs. Lagrangian

- Passive advection without diffusion:

$$\frac{\partial C}{\partial t} + \boldsymbol{v}\cdot\boldsymbol{\nabla} C = 0.$$

- Finite difference:

$$\frac{C_i^{n+1} - C_i^n}{\tau} = -v\frac{C_{i+1}^n - C_{i-1}^n}{2h}.$$

- Problems with Eulerian methods:
  - Instability;
  - Upwinding and Lax schemes: numerical diffusion.

Method of Characteristics

- Solution to passive advection problem without diffusion:

$$C(\boldsymbol{x}, t) = C\left(\boldsymbol{\xi}_0(\boldsymbol{x}, t), 0\right).$$

- Introduce Lagrangian position

$$\boldsymbol{\xi}(t) = \boldsymbol{\xi}_0 + \int_0^t \boldsymbol{v}(\boldsymbol{\xi}(\tau), \tau)\, d\tau,$$

where $\boldsymbol{\xi}(t) = \boldsymbol{x}$ and $\boldsymbol{\xi}_0 = \boldsymbol{\xi}_0(\boldsymbol{x}, t)$ is the initial parcel position.

- Problem of viewing solution on grid: new Lagrangian positions may not lie on grid points.

- Solutions:

  – interpolate (semi-Lagrangian): numerical diffusion;

  – Lagrangian rearrangement: project advected parcel centroids onto rearrangment manifold.

Casimir Invariants

- Conservation equation:

$$\frac{dC(\boldsymbol{x}(t),t)}{dt} = \frac{d\boldsymbol{x}}{dt}\boldsymbol{\cdot}\boldsymbol{\nabla} C + \frac{\partial C}{\partial t} = \boldsymbol{v}\boldsymbol{\cdot}\boldsymbol{\nabla} C + \frac{\partial C}{\partial t} = 0,$$

where $\boldsymbol{v} = d\boldsymbol{x}/dt$.

- For any C^1 function f of concentration (or vorticity) field:

$$\frac{d}{dt}\int f(C)\,d\boldsymbol{x} = \int f'(C)\frac{\partial C}{\partial t}\,d\boldsymbol{x} = -\int f'(C)\boldsymbol{v}\boldsymbol{\cdot}\boldsymbol{\nabla} C\,d\boldsymbol{x}$$
$$= -\int \boldsymbol{v}\boldsymbol{\cdot}\boldsymbol{\nabla} f(C)\,d\boldsymbol{x} = \int f(C)\boldsymbol{\nabla}\boldsymbol{\cdot}\boldsymbol{v}\,d\boldsymbol{x} = 0.$$

- Proposition: enforce a discrete analog of this exact infinitesimal property:

$$\frac{d}{dt}\sum_{i,j} f(C_{i,j}) = 0.$$

Parcel Centroid

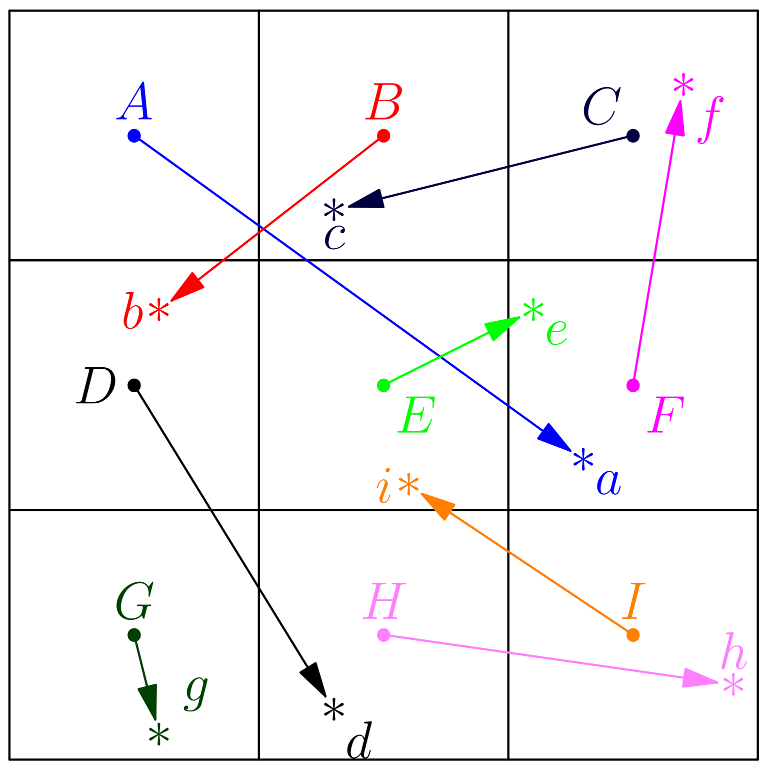
- Advection map is continuous and area-preserving $\Rightarrow$ rearrangement into distinct nonoverlapping parcels.
- Represent solution as finite union of piecewise constant functions.
- The resulting discrete constraint

$$\frac{d}{dt}\sum_{i,j} f(C_{i,j}) = 0$$

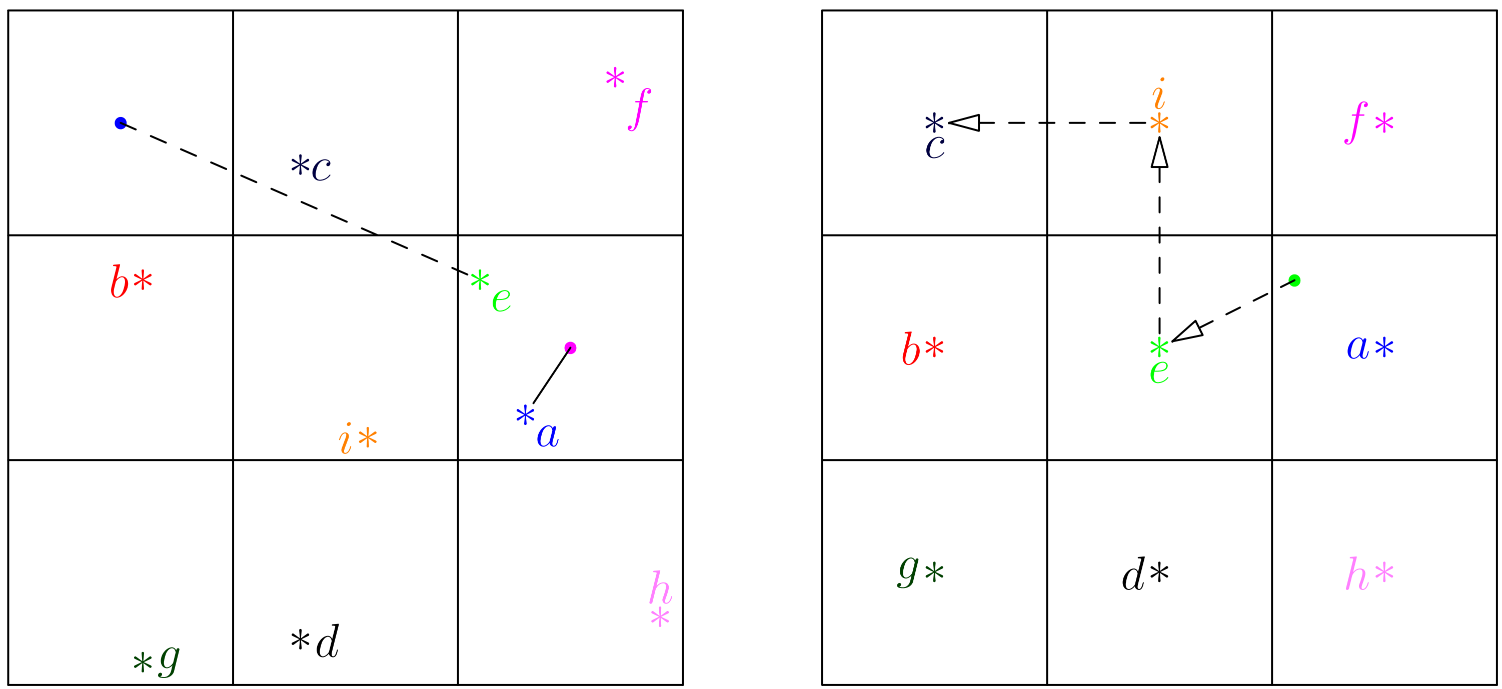
  is equivalent to imposing parcel rearrangement.
- Use RK4 to advect the parcel centroids.
- Under this linear map, centroid of parcel maps to centroid of advected parcel.
- For passive advection without diffusion: only evolve parcel centroids (no need to actually evolve the quadrilateral vertices).

Lagrangian→Eulerian Projection

- Advection in Lagrangian frame ⇒ piles and holes.
- New state must be a rearrangement of initial state to conserve Casimir invariants.
- How to map excess parcels to holes?

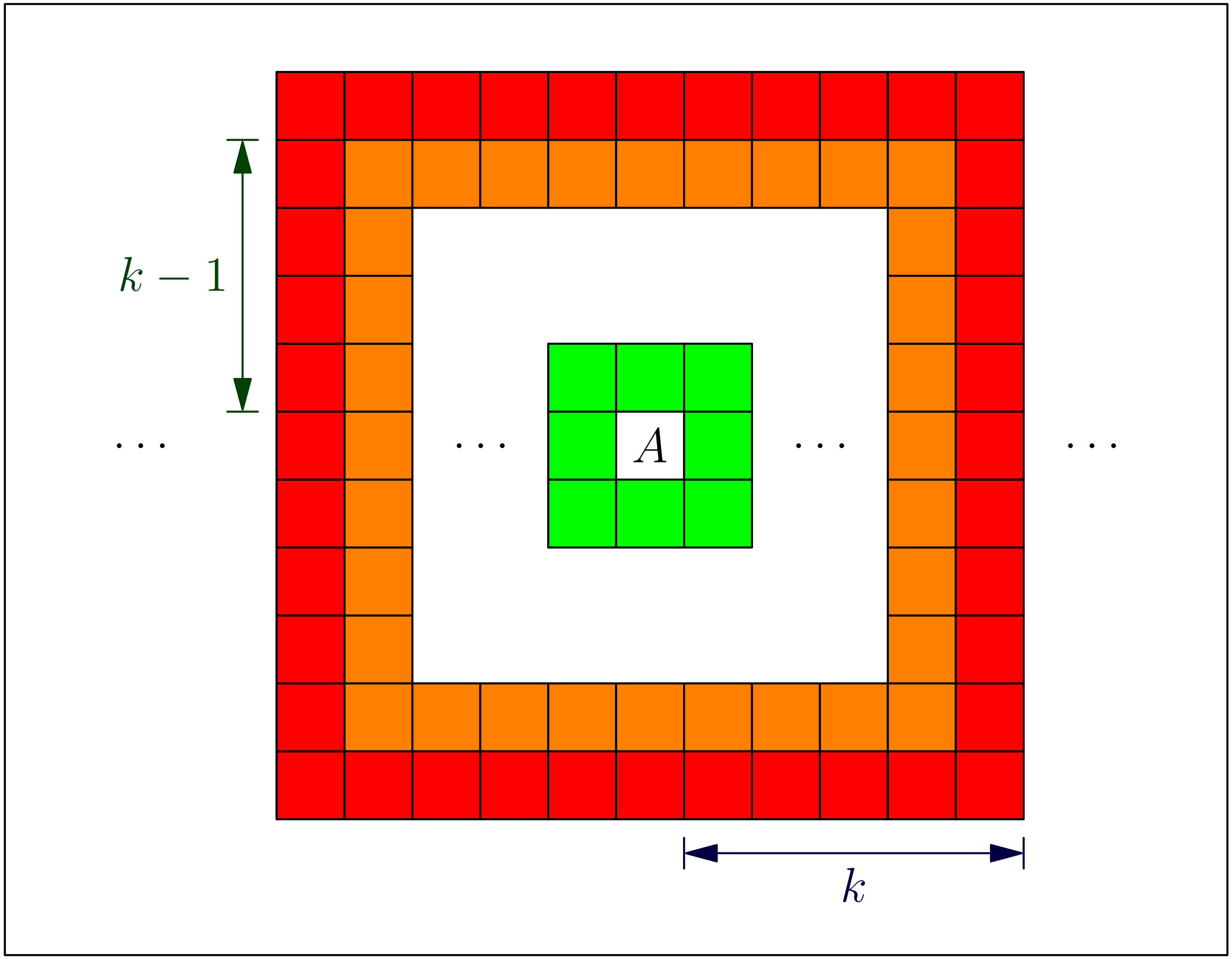


Lagrangian Rearrangement



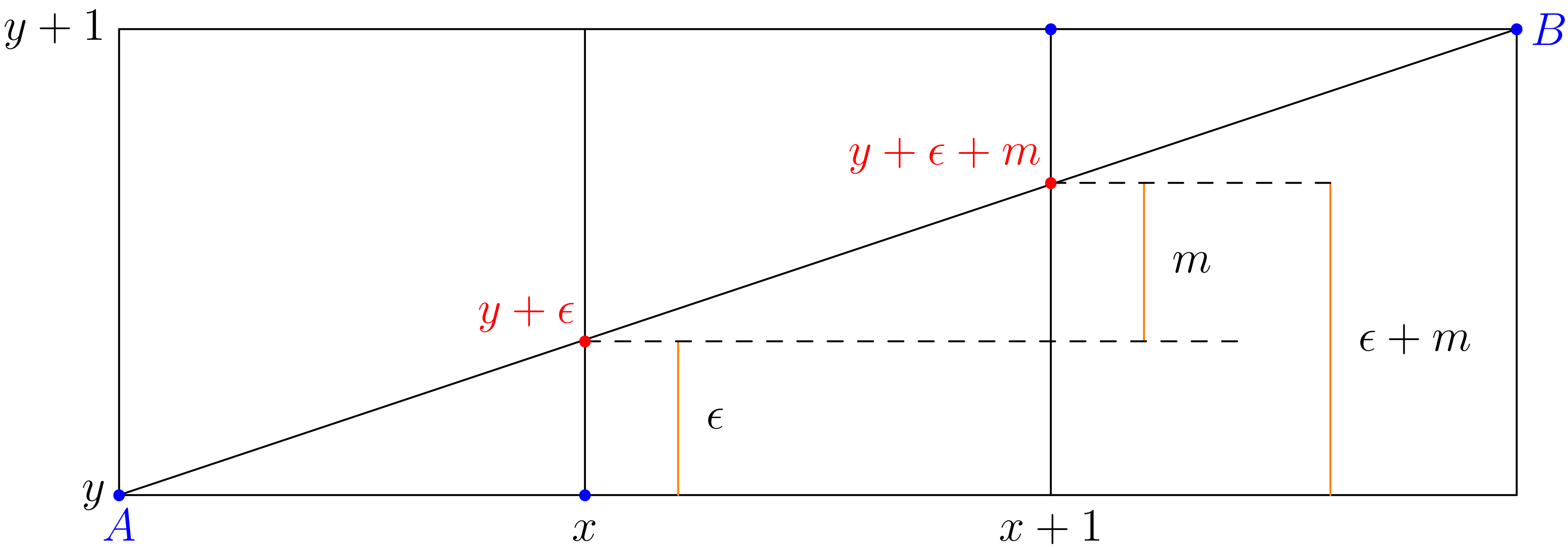
- Start with cells with most parcels.
- Find nearest hole (search in rectangular shells about pile).
- Discretize path from pile to hole.
- Push chain of parcels toward hole.

Searching kth Rectangular Shell



Bresenham Algorithm

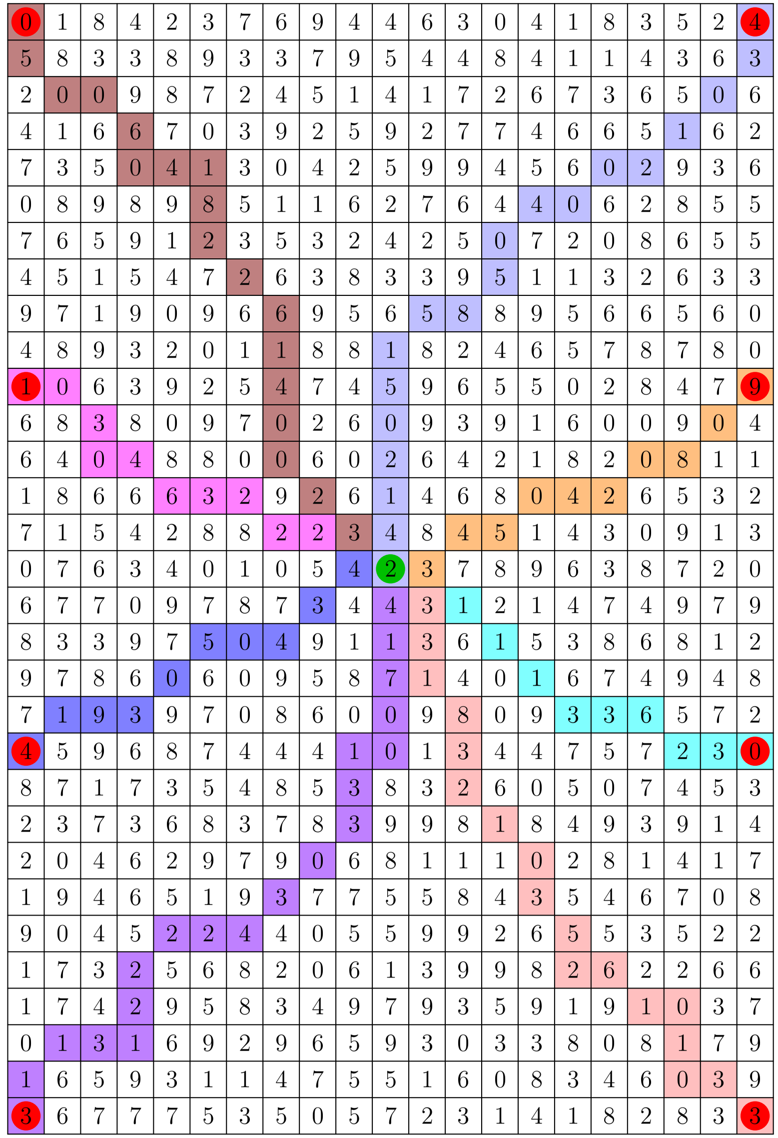
- Discretize path from pile to hole:
  - Reduce to case $0 \leq m \leq 1$.
  - Choose $(x+1, y)$ or $(x+1, y+1)$.



- Problem: multiple pushing of parcels $\Rightarrow$ visible streaks.

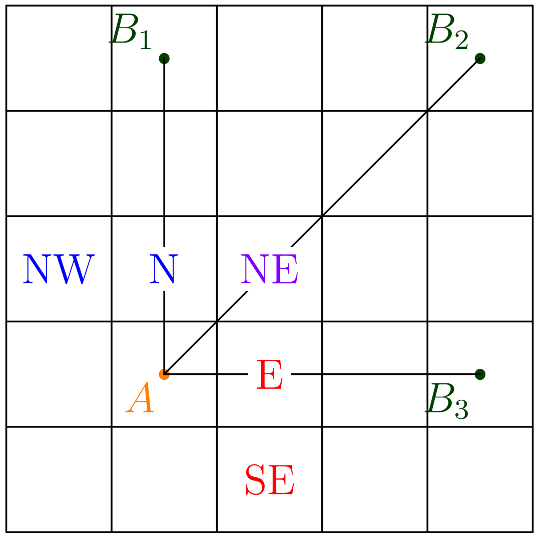
Weighted Bresenham Algorithm

- Randomize path:

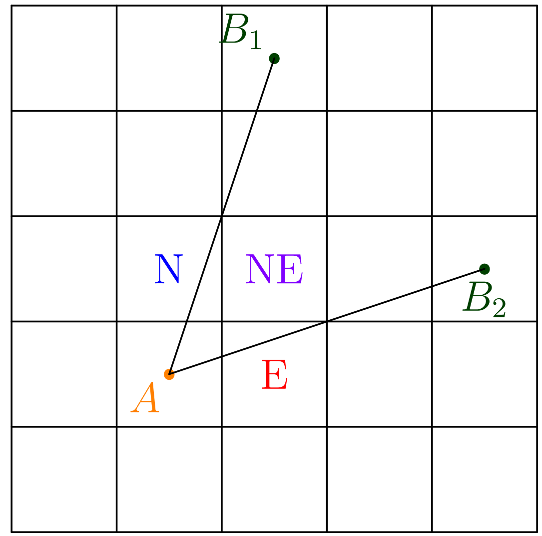


- Find quasi-optimal local path based on Lagrangian position.

- **Theorem 1:** *The weighted Bresenham algorithm produces a finite path between any two points on a regular lattice. For a unit square lattice, at most* $\lceil 1.82x \rceil$ *steps are needed to connect two points a distance* x *apart.*



(a)



(b)

- Parcel chains: select parcels with minimal weight.
- Multiple holes in same shell: minimize the error.

Average Complexity

- Probability analysis:
  - There are n cells, each with area $1/n$.
  - Probability that a cell contains a given parcel is $1/n$.
  - Binomial probability of a cell containing k parcels:

$$P(k) = \binom{n}{k} \left(\frac{1}{n}\right)^k \left(1 - \frac{1}{n}\right)^{n-k}.$$

  - Probability of having a hole:

$$P(0) = \left(1 - \frac{1}{n}\right)^n \to e^{-1} \text{ as } n \to \infty.$$

- Level k reached if no holes are found in first $k - 1$ shells.
- Multiple holes in a shell $\Rightarrow$ multiple Bresenham search.

Approximate Cost/Chain

- Searching for hole:

$$\sum_{k=1}^{\infty} 8k \left(1 - \frac{1}{e}\right)^{4k(k-1)} \approx 8.4.$$

- Identifying the path:

$$1.82 \sum_{k=1}^{\infty} k\sqrt{2} \left(1 - \frac{1}{e}\right)^{4k(k-1)} \left(\frac{1}{e}\right) \frac{8k}{1 - \left(1 - \frac{1}{e}\right)^{8k}} \approx 8.6.$$

- Pushing a chain of parcels:

$$1.82 \sum_{k=1}^{\infty} k\sqrt{2} \left(1 - \frac{1}{e}\right)^{4k(k-1)} \approx 2.7.$$

Diffusion

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{v}\boldsymbol{\cdot}\boldsymbol{\nabla}\boldsymbol{U} = \boldsymbol{D}\nabla^2\boldsymbol{U}.$$

- Use operator splitting to include diffusion:

$\boldsymbol{U}(t) = \boldsymbol{U}(t_1, t_2)$

$$\frac{\partial \boldsymbol{U}}{\partial t_1} = -\boldsymbol{v}\boldsymbol{\cdot}\boldsymbol{\nabla}\boldsymbol{U}, \quad \frac{\partial \boldsymbol{U}}{\partial t_2} = \boldsymbol{D}\nabla^2\boldsymbol{U}$$

$$\Rightarrow \Delta\boldsymbol{U} = -\boldsymbol{v}\boldsymbol{\cdot}\boldsymbol{\nabla}\boldsymbol{U}\,\Delta t_1 + \boldsymbol{D}\nabla^2\boldsymbol{U}\,\Delta t_2.$$

- Crank–Nicholson scheme solves for diffusive part:

$$\frac{\boldsymbol{U}(t+\tau) - \boldsymbol{U}(t)}{\tau} = \boldsymbol{D}\frac{\nabla^2\boldsymbol{U}(t+\tau) + \nabla^2\boldsymbol{U}(t)}{2}.$$

- In the advection equation $\partial\widetilde{\boldsymbol{U}}/\partial t = -\boldsymbol{v}\boldsymbol{\cdot}\boldsymbol{\nabla}\widetilde{\boldsymbol{U}}$:

- Calculate $\widetilde{\boldsymbol{U}}$, interpolate to Eulerian grid.

- Finite difference:

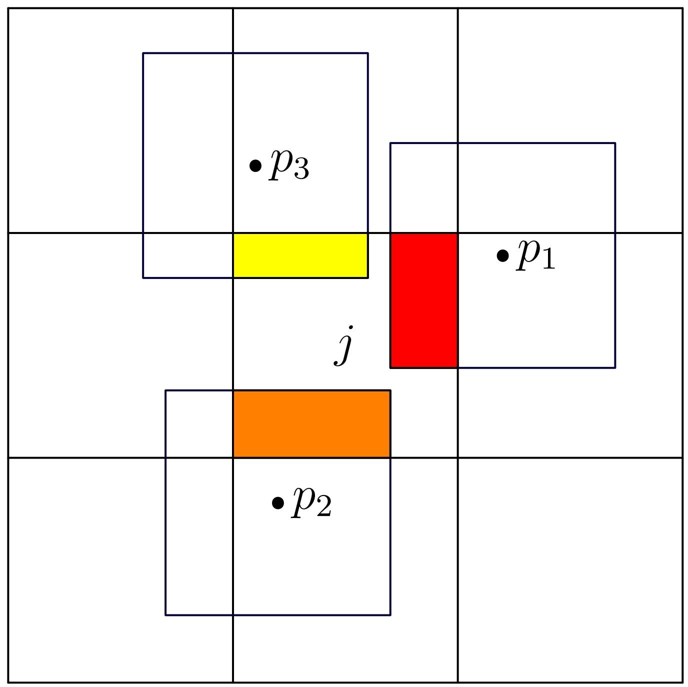
$$\frac{\boldsymbol{U} - \widetilde{\boldsymbol{U}}}{\tau} = \boldsymbol{D}\nabla^2 \left(\frac{\boldsymbol{U} + \widetilde{\boldsymbol{U}}}{2} \right).$$

- Multigrid:
  - Let $\boldsymbol{\mathcal{L}} = \mathbf{1} + \frac{\tau}{2}\boldsymbol{D}\nabla^2 \quad \Rightarrow \quad \boldsymbol{\mathcal{L}}(-\tau)\boldsymbol{U} = \boldsymbol{\mathcal{L}}(\tau)\widetilde{\boldsymbol{U}}$.

- Contribution of diffusion to the Lagrangian solution:
  - Calculate $\boldsymbol{U} - \widetilde{\boldsymbol{U}}$.
  - Project to Lagrangian frame.
  - Add to parcel values.

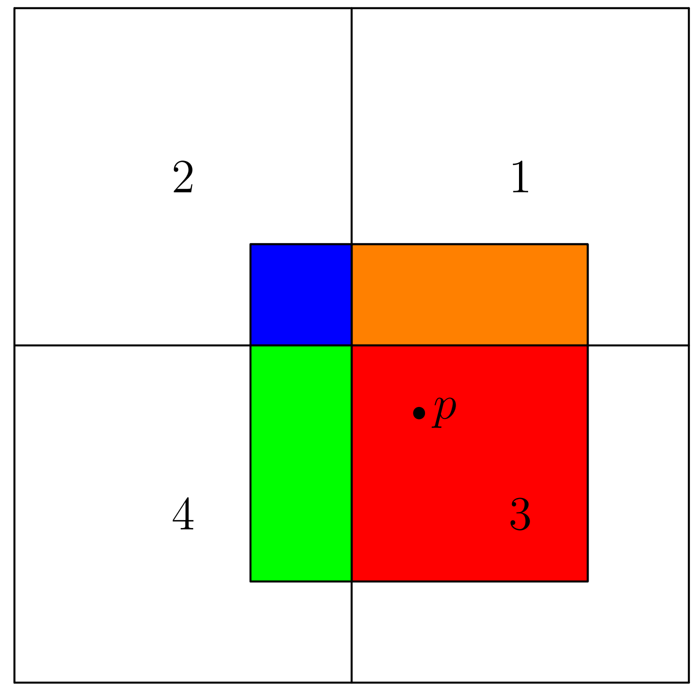
Area-Weighted Interpolation

Lagrangian → Eulerian



$$\boldsymbol{U}_j = \frac{\sum_i A_{ij}\boldsymbol{p}_i}{\sum_i A_{ij}}$$

Eulerian → Lagrangian



$$p = \sum_j A_j U_j$$

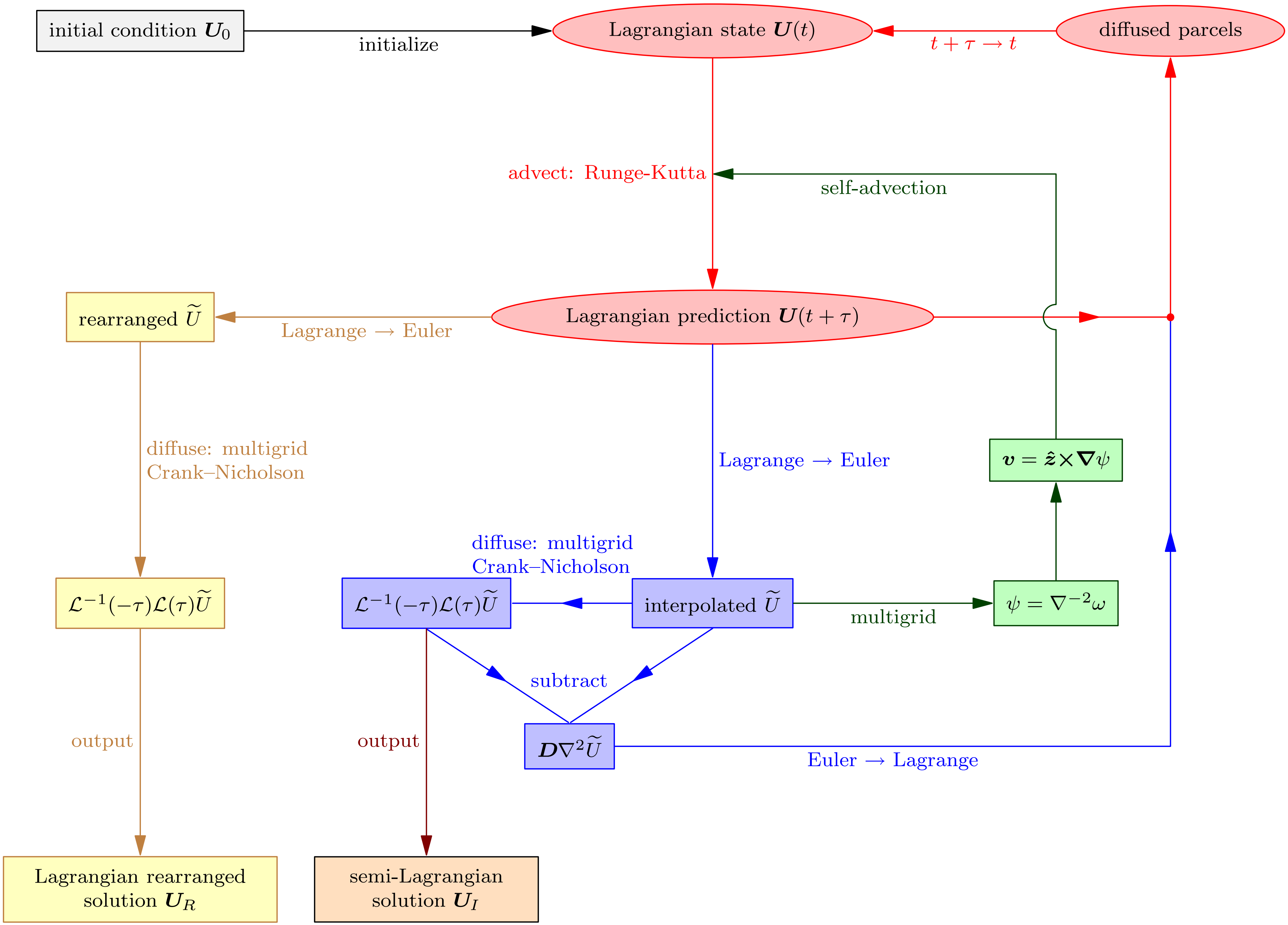
Self-Advection

- Velocity is now a functional of $\boldsymbol{U}$ determined by 2D vorticity equation:

$$\frac{\partial \omega}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} \omega = \boldsymbol{D} \nabla^2 \omega.$$

- Use multigrid solver: compute stream function $\psi = \nabla^{-2} \omega$.
- Calculate $\boldsymbol{v} = \hat{\boldsymbol{z}} \times \boldsymbol{\nabla} \psi$ from ψ.
- Problem: calculating $\boldsymbol{v}$ from rearranged $\omega \Rightarrow$ pushing errors accumulate:
  - Propagation of error *via* advection term $\boldsymbol{v} \cdot \boldsymbol{\nabla} \omega$.
  - Introduces large gradients in ω and $C \Rightarrow$ excessive diffusion.
- Solution: use interpolated rather than rearranged values: $\boldsymbol{v}_I \cdot \boldsymbol{\nabla} \omega$, $\nu \nabla^2 \omega_I$, and $D \nabla^2 C_I$.
- This interpolation does not destroy the conservation of Casimirs: velocity need not be a rearrangement.

Summary



initial condition $\boldsymbol{U}_0$
initialize
Lagrangian state $\boldsymbol{U}(t)$
$t+\tau \to t$
diffused parcels
advect: Runge-Kutta
self-advection
rearranged $\widetilde{U}$
Lagrange → Euler
Lagrangian prediction $\boldsymbol{U}(t+\tau)$
diffuse: multigrid
Crank–Nicholson
Lagrange → Euler
$\boldsymbol{v} = \hat{\boldsymbol{z}} \times \boldsymbol{\nabla}\psi$
diffuse: multigrid
Crank–Nicholson
$\mathcal{L}^{-1}(-\tau)\mathcal{L}(\tau)\widetilde{U}$
$\mathcal{L}^{-1}(-\tau)\mathcal{L}(\tau)\widetilde{U}$
interpolated $\widetilde{U}$
multigrid
$\psi = \nabla^{-2}\omega$
subtract
output
output
$\boldsymbol{D}\nabla^2\widetilde{U}$
Euler → Lagrange
Lagrangian rearranged
solution $\boldsymbol{U}_R$
semi-Lagrangian
solution $\boldsymbol{U}_I$

Simulations: 2 Test Cases

- Semi-Lagrangian solution vs. Lagrangian rearrangement:
- Grid scale $h = 1.95 \times 10^{-3}$, time step $\tau = 1.95 \times 10^{-2}$.
- Initial condition:

$$v_x = \sin(2\pi x)\cos(2\pi y), \quad v_y = -\cos(2\pi x)\sin(2\pi y).$$

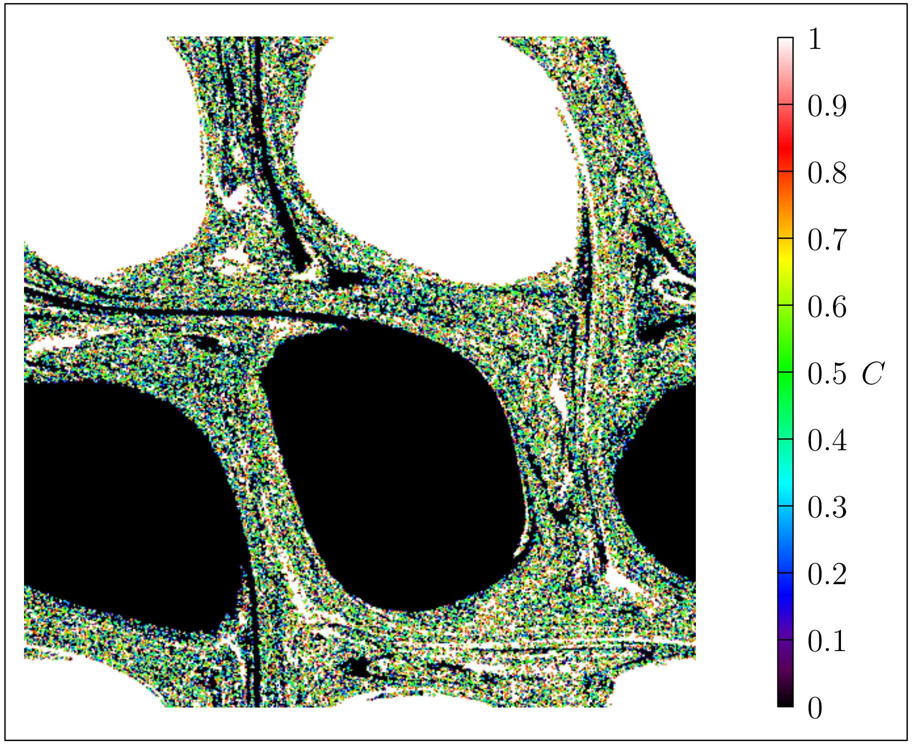
– Self-advection with no diffusion:

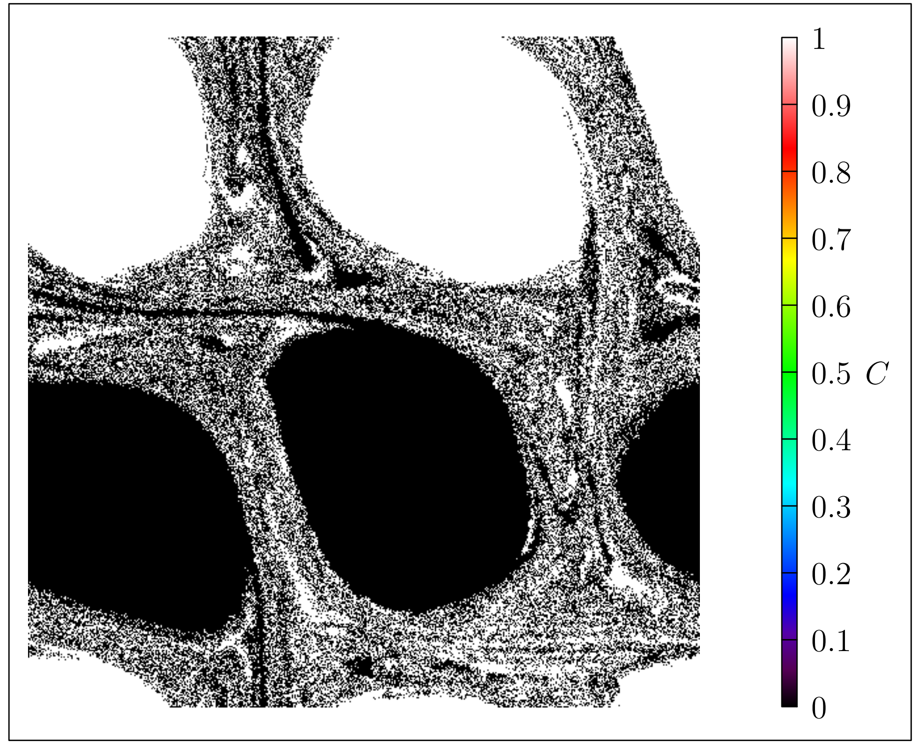
0 (black) and 1 (white) initial condition for C.

– Self-advection with diffusion:

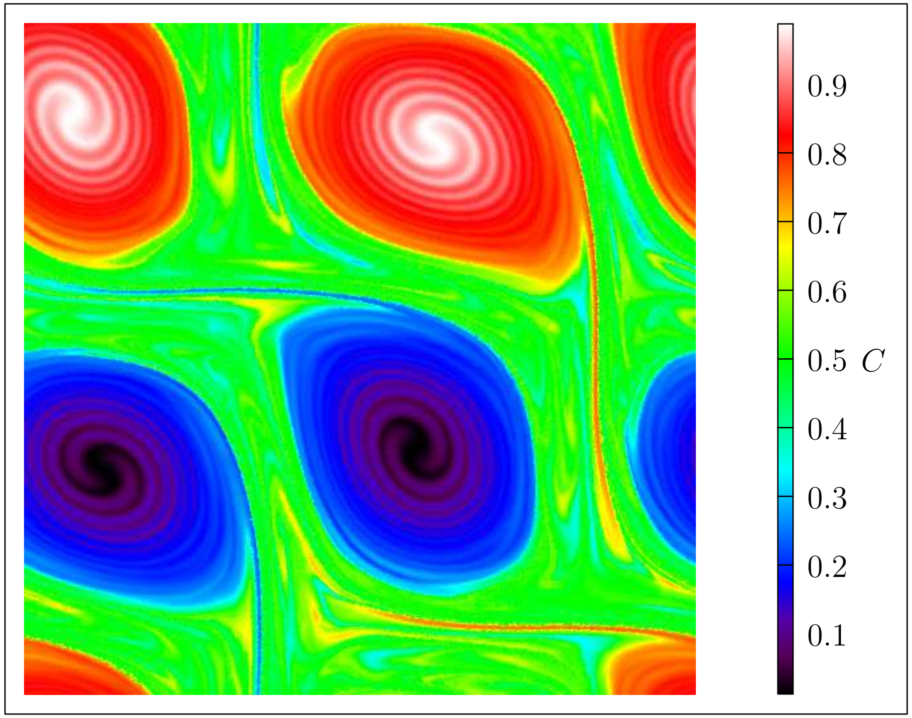
$D = \nu = 2 \times 10^{-6}$.

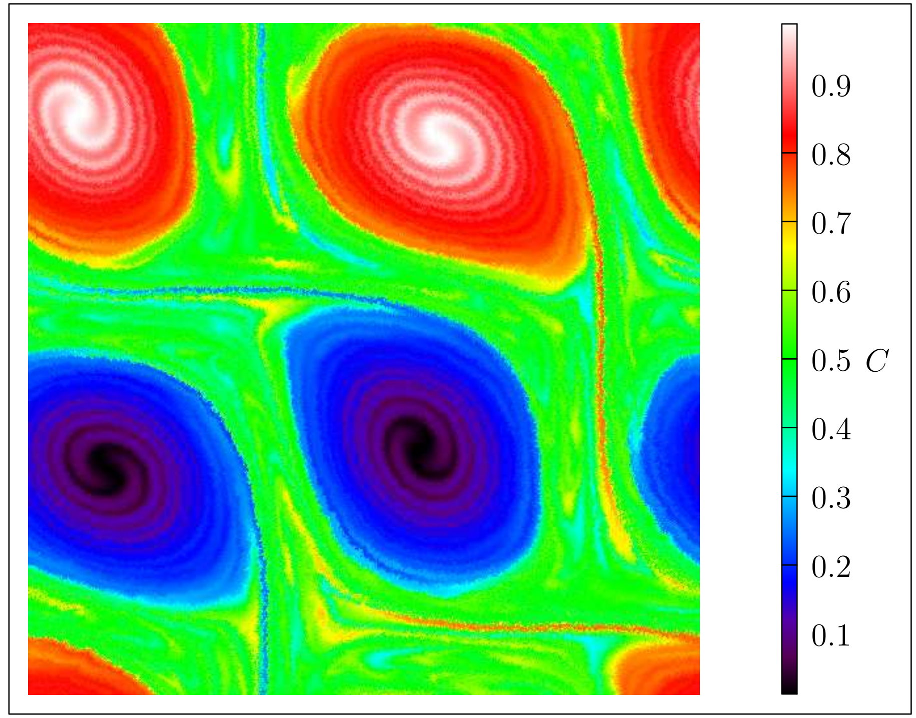
Semi-Lagrangian vs. Lagrangian Rearrangement After 750 Time Steps ($\boldsymbol{D} = \mathbf{0}$)





Semi-Lagrangian vs. Lagrangian Rearrangement After 750 Time Steps ($\boldsymbol{D} \neq \mathbf{0}$)





Energy Decay Rate

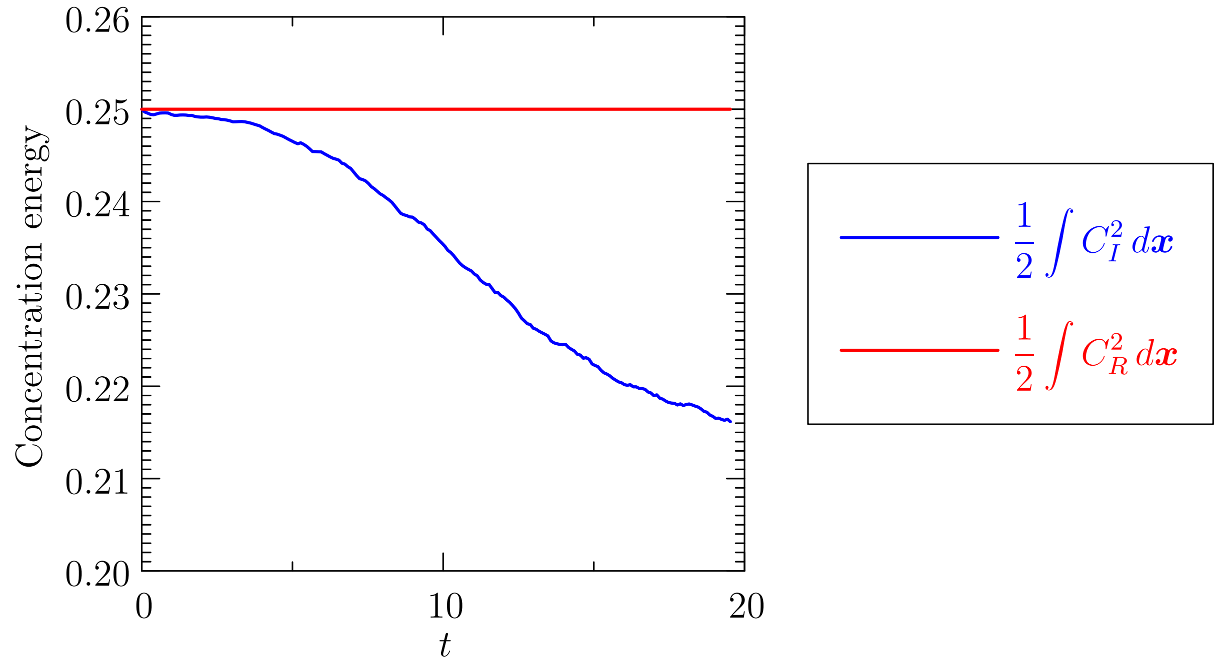
$$\frac{\partial C}{\partial t} + \boldsymbol{v}\boldsymbol{\cdot}\boldsymbol{\nabla} C = D\nabla^2 C.$$

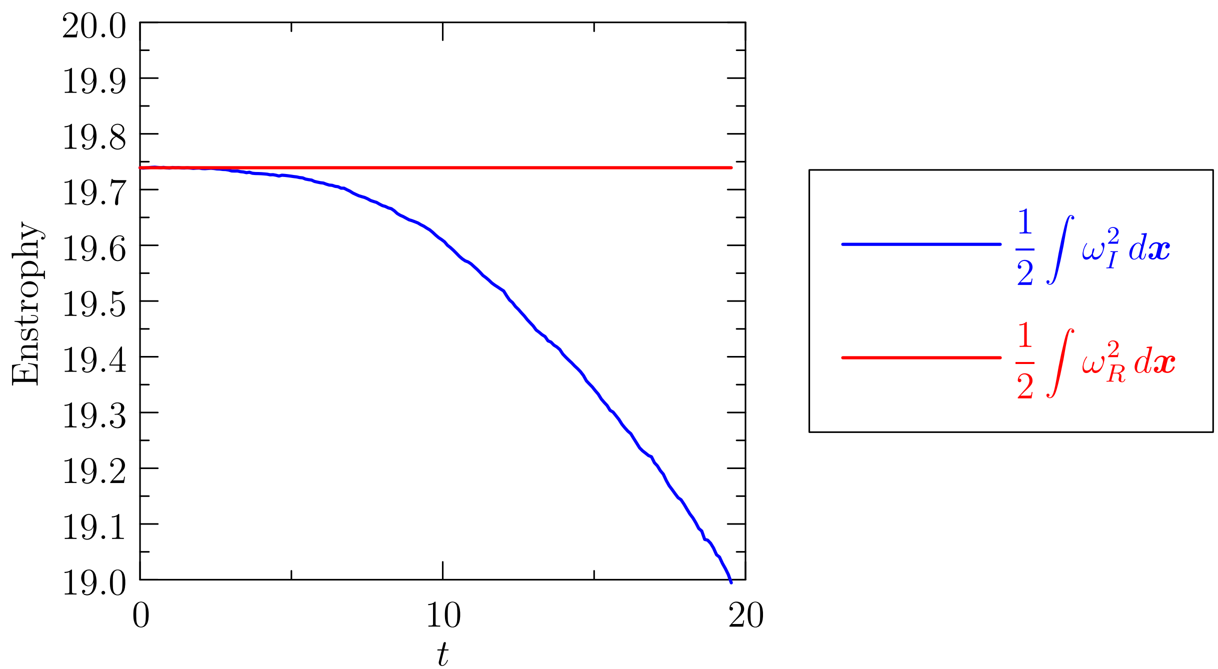
- Evolution of concentration energy:

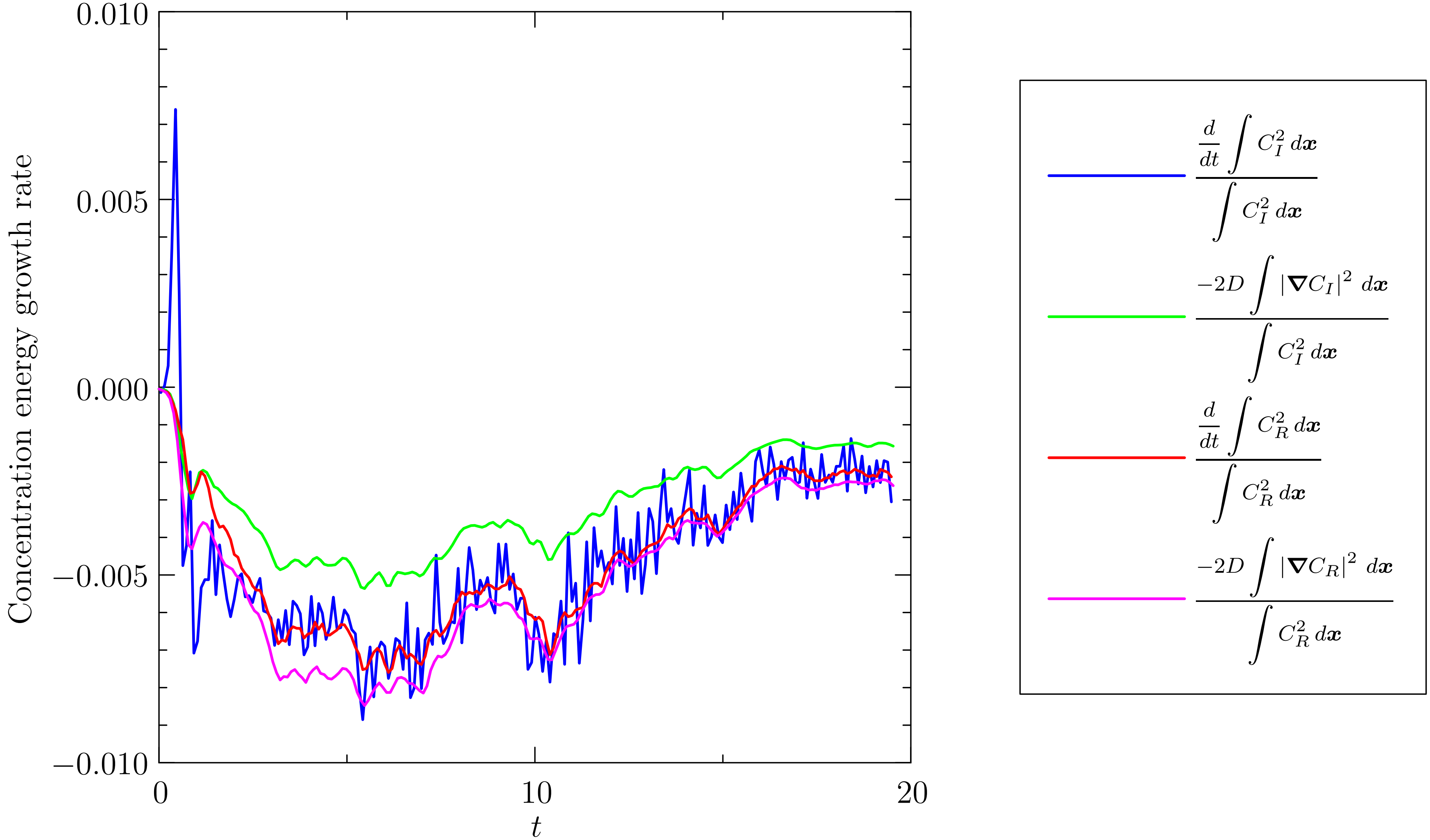
$$\frac{1}{2}\frac{\partial}{\partial t}\int C^2\, d\boldsymbol{x} = -D\int |\boldsymbol{\nabla} C|^2\, d\boldsymbol{x}.$$

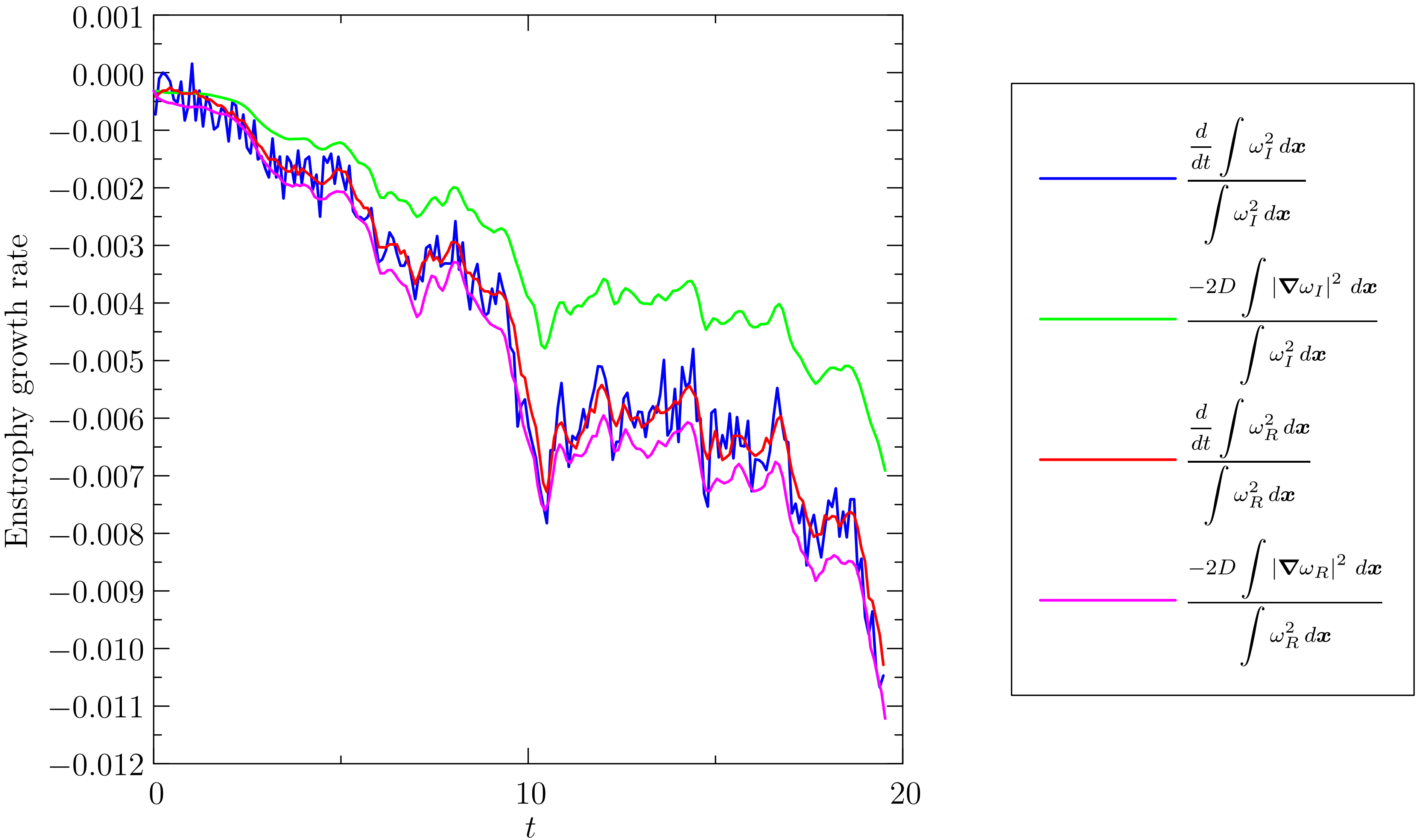
- Compare

$$\frac{\dfrac{\partial}{\partial t}\displaystyle\int C^2\, d\boldsymbol{x}}{\displaystyle\int C^2\, d\boldsymbol{x}} \quad \text{and} \quad \frac{-2D\displaystyle\int |\boldsymbol{\nabla} C|^2\, d\boldsymbol{x}}{\displaystyle\int C^2\, d\boldsymbol{x}}.$$


0.26
0.25
0.24
0.23
0.22
0.21
0.20
Concentration energy
0
10
20
t
1/2 ∫ C_I^2 dx
1/2 ∫ C_R^2 dx


20.0
19.9
19.8
19.7
19.6
19.5
19.4
19.3
19.2
19.1
19.0
Enstrophy
0
10
20
t
1/2 ∫ ω_I^2 dx
1/2 ∫ ω_R^2 dx


Concentration energy growth rate
0.010
0.005
0.000
−0.005
−0.010
0
10
20
t
$\dfrac{\frac{d}{dt}\int C_I^2\,d\boldsymbol{x}}{\int C_I^2\,d\boldsymbol{x}}$
$\dfrac{-2D\int |\boldsymbol{\nabla} C_I|^2\,d\boldsymbol{x}}{\int C_I^2\,d\boldsymbol{x}}$
$\dfrac{\frac{d}{dt}\int C_R^2\,d\boldsymbol{x}}{\int C_R^2\,d\boldsymbol{x}}$
$\dfrac{-2D\int |\boldsymbol{\nabla} C_R|^2\,d\boldsymbol{x}}{\int C_R^2\,d\boldsymbol{x}}$


Enstrophy growth rate
0.001
0.000
−0.001
−0.002
−0.003
−0.004
−0.005
−0.006
−0.007
−0.008
−0.009
−0.010
−0.011
−0.012
0
10
20
t
$\frac{\frac{d}{dt}\int \omega_I^2 \, d\boldsymbol{x}}{\int \omega_I^2 \, d\boldsymbol{x}}$
$\frac{-2D\int |\boldsymbol{\nabla}\omega_I|^2 \, d\boldsymbol{x}}{\int \omega_I^2 \, d\boldsymbol{x}}$
$\frac{\frac{d}{dt}\int \omega_R^2 \, d\boldsymbol{x}}{\int \omega_R^2 \, d\boldsymbol{x}}$
$\frac{-2D\int |\boldsymbol{\nabla}\omega_R|^2 \, d\boldsymbol{x}}{\int \omega_R^2 \, d\boldsymbol{x}}$

Conclusions

- New numerical method Lagrangian rearrangement respects Casimir invariants.
- Based on a weighted Bresenham Lagrangian-to-Eulerian projection algorithm.
- Fully Lagrangian:
  - Projected solution is used only for viewing;
  - Error does not propagate to future time steps.
- Can combine with:
  - Diffusion

    ($\Rightarrow$ more consistent energy behaviour than interpolation);
  - Self-advected flow.
- Complexity $\mathcal{O}(n)$.

Asymptote: The Vector Graphics Language

`http://asymptote.sf.net`

(freely available under the GNU public license)