Optimization-Based Mesh Correction

M. D'Elia, D. Ridzal, K. Peterson, P. Bochev

Sandia National Laboratories

M. Shashkov

Los Alamos National Laboratory

MultiMat

September 9, 2015

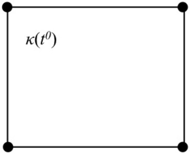
SAND 2015-7334C

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Motivation

Divergence Free Lagrangian Motion

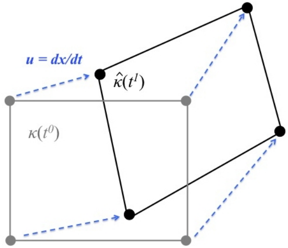
- Given cell κ at initial time t^0



Motivation

Divergence Free Lagrangian Motion

- Given cell κ at initial time t^0
- Compute nodal displacement from velocity field $\mathbf{u}$



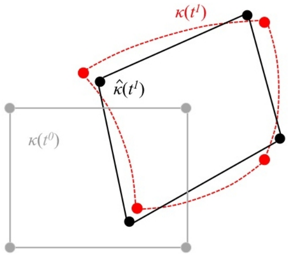
Motivation

Divergence Free Lagrangian Motion

- Given cell κ at initial time t^0
- Compute nodal displacement from velocity field $\mathbf{u}$
- Updated cell $\hat{\kappa}(t^1)$ has both temporal and spatial errors

Violation of volume preservation

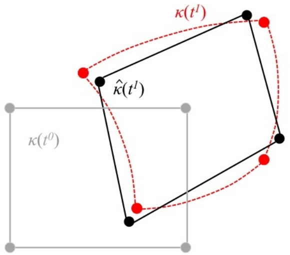
$$\frac{d}{dt}\int_{\kappa(t)} dV \neq 0$$



Motivation

Divergence Free Lagrangian Motion

- Given cell κ at initial time t^0
- Compute nodal displacement from velocity field $\mathbf{u}$
- Updated cell $\hat{\kappa}(t^1)$ has both temporal and spatial errors



Violation of volume preservation

$$\frac{d}{dt}\int_{\kappa(t)} dV \neq 0$$

Consider $\rho = const$

Let cell mass $M_\kappa(t) = \displaystyle\int_{\kappa(t)} \rho dV$ and cell density $\rho_\kappa = \dfrac{M_\kappa(t)}{|\kappa(t)|}$,
where $|\kappa(t)| = \int_{\kappa(t)} dV$

$$\rho_\kappa(t^1) = \frac{M_\kappa(t^1)}{|\hat{\kappa}(t^1)|} \neq \frac{M_\kappa(t^0)}{|\kappa(t^0)|} = \rho_\kappa(t^0)$$

Cannot maintain a constant density!

Geometric Conservation Law (GCL)

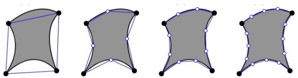
$$\frac{d}{dt}\int_{\kappa_i(t)} dV = \int_{\partial\kappa_i(t)} \mathbf{u}\cdot\mathbf{n}\, ds$$

Some recent work:

Use more Lagrangian points

- Enforces GCL approximately

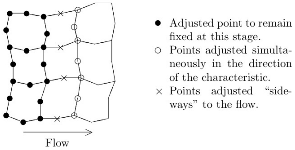
Lauritzen, Nair, Ullrich (2010), A conservative semi-Lagrangian multi-tracer transport scheme on the cubed-sphere grid, *JCP*.



Heuristic mesh adjustment procedure

- No theoretical assurance of completion

Arbogast and Huang (2006), A fully mass and volume conserving implementation of a characteristic method for transport problems, *SISC*.



Monge-Ampére trajectory correction

- Accuracy of the MAE scheme determines accuracy of GCL approximation

Cossette, Smolarkiewicz, Charbonneau (2014), The Monge-Ampere trajectory correction for semi-Lagrangian schemes, *JCP*.

$$\tilde{\mathbf{p}}_{ij}^{corr} = \tilde{\mathbf{p}}_{ij} + (t - t_n)\nabla\phi;$$

$$\det\frac{\partial\mathbf{p}_{ij}}{\partial x} = 1$$

Optimization-Based Solution

Given a *source mesh* $\widetilde{K}(\Omega)$, and *desired cell volumes* $\boldsymbol{c}_0 \in \mathbb{R}^m$ such that

$$\sum_{i=1}^{m} c_{0,i} = |\Omega| \quad \text{and} \quad c_{o,i} > 0 \forall i = 1, ..., m$$

Find a *volume compliant mesh* $K(\Omega)$ such that

1. $K(\Omega)$ has the same connectivity as the source mesh
2. The volumes of its cells match the volumes prescribed in $\boldsymbol{c}_0$
3. Every cell $\kappa_i \in K(\Omega)$ is valid or convex
4. Boundary points in $K(\Omega)$ correspond to boundary points in $\widetilde{K}(\Omega)$

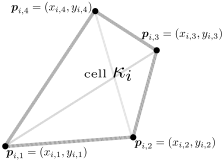
Requirements for Quadrilateral Cells

Oriented volume of quad cell:

$$\boldsymbol{c}_i(K(\Omega)) = \frac{1}{2}\left((x_{i,1}-x_{i,3})(y_{i,2}-y_{i,4})+(x_{i,2}-x_{i,4})(y_{i,3}-y_{i,1})\right)$$

Partitioning of quad into triangles:

$$(a_r, b_r, c_r) = \begin{cases} (1,2,4) & r = 1 \\ (2,3,4) & r = 2 \\ (1,3,4) & r = 3 \\ (1,2,3) & r = 4. \end{cases}$$



Oriented volume of triangle cell:

$$\boldsymbol{t}_i^r(K(\Omega)) = \frac{1}{2}\left(x_{i,a_r}(y_{i,c_r}-y_{i,b_r})-x_{i,b_r}(y_{i,a_r}-y_{i,c_r})-x_{i,c_r}(y_{i,b_r}-y_{i,a_r})\right).$$

Convexity indicator for a quad cell:

$$\mathcal{I}_i(K(\Omega)) = \begin{cases} 1 & \text{if } \forall \boldsymbol{t}_i^r \in \kappa_i, |\boldsymbol{t}_i^r| > 0 \\ 0 & \text{otherwise} \end{cases}$$

Optimization Problem

Objective:

Mesh distance $J_0(\boldsymbol{p}) = \frac{1}{2} d\big(K(\Omega), \tilde{K}(\Omega)\big)^2 = |\boldsymbol{p} - \tilde{\boldsymbol{p}}|_{l^2}^2$

Constraints:

(1) Volume equality $\forall \kappa_i \in K(\Omega), |\kappa_i| = c_{0,i}$

(2) Cell convexity $\forall \kappa_i \in K(\Omega)\,, \forall t_i^r \in \kappa_i\,, |t_i^r| > 0$

(3) Boundary compliance $\forall p_i \in \partial\Omega\,, \gamma(p_i) = 0$

Nonlinear programming problem (NLP)

$\boldsymbol{p}* =$ arg min$\big\{ J_0(\boldsymbol{p})$ subject to (1),(2), and (3)$\big\}$

Simplified Formulation

For polygonal domains

- boundary compliance can be subsumed in the volume constraint
- convexity can be enforced weakly by logarithmic barrier functions

Objective:

Mesh distance - log barrier

$$J(\boldsymbol{p}) = J_0(\boldsymbol{p}) - \beta \sum_{i=1}^{m} \sum_{r=1}^{4} \log \boldsymbol{t}_i^r(\boldsymbol{p})$$

$$J_0(\boldsymbol{p}) = |\boldsymbol{p} - \tilde{\boldsymbol{p}}|_{l^2}^2$$

Constraints:

(1) Volume equality $\forall \kappa_i \in K(\Omega), |\kappa_i| = c_{0,i}$

Simplified NLP

$$\boldsymbol{p}^* = \text{arg min}\{J(\boldsymbol{p}) \text{ subject to (1)}\}$$

Scalable Optimization Algorithm

Based on the inexact trust region sequential programming (SQP) method with key properties:

- Fast local convergence
- Use of very coarse iterative solvers
- Handles rank-deficient constraints

Linear systems for an optimization iterate $\boldsymbol{p}^k$ are of the form

$$\begin{pmatrix} I & \nabla \boldsymbol{C}(\boldsymbol{p}^k)^T \\ \nabla \boldsymbol{C}(\boldsymbol{p}^k) & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{v}^1 \\ \boldsymbol{v}^2 \end{pmatrix} = \begin{pmatrix} \boldsymbol{b}^1 \\ \boldsymbol{b}^2 \end{pmatrix},$$

where $\boldsymbol{C}(\boldsymbol{p}^k)$ is a nonlinear vector function of coordinates.

Preconditioner

$$\pi^k = \begin{pmatrix} I & 0 \\ 0 & \left(\nabla \boldsymbol{C}(\boldsymbol{p}^k) \nabla \boldsymbol{C}(\boldsymbol{p}^k)^T + \varepsilon I\right)^{-1} \end{pmatrix}$$

- $\epsilon > 0$ small parameter $10^{-8}h$
- $\nabla \boldsymbol{C}(\boldsymbol{p}^k) \nabla \boldsymbol{C}(\boldsymbol{p}^k)^T$ formed explicitly
- Smoothed aggregation AMG used for inverse

Heinkenschloss, Ridzal (2014), A matrix-free trust-region SQP method for equality constrained optimization *SIOPT*.

Scalablility Test

To challenge the algorithm we test performance as follows:

- Start with uniform $n \times n$ mesh and advance to final time using velocity field
- Apply algorithm to the deformed mesh at the final time using initial mesh volumes

Analytic Hessian

n	SQP	CG	GMRES tot.	GMRES av.	CPU	% ML time
64	5	15	101	2.8	2.475	66
128	4	9	106	4.1	8.799	78
256	5	5	130	5.0	45.733	83
512	6	1	100	3.8	184.446	83

Gauss-Newton Hessian

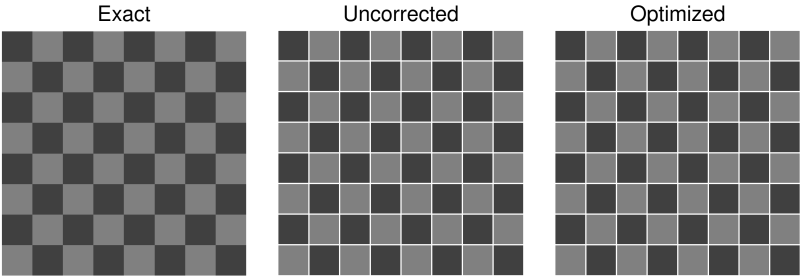
n	SQP	CG	GMRES tot.	GMRES av.	CPU	% ML time
64	5	5	64	2.5	1.666	63
128	4	4	79	3.8	6.466	82
256	5	5	126	4.8	43.241	86
512	6	6	100	3.8	183.697	86

- Almost constant GMRES iterations
- CPU per SQP iteration scales linearly with problem
- Computational time dominated by AMG (ML) preconditioner
- The algorithm inherits its scalability from the AMG solver

Lagrangian Motion

Swirling velocity field: $\mathbf{u}(\mathbf{x},t) = \begin{pmatrix} \cos\left(\frac{t\pi}{T}\right)\sin(\pi x)^2\sin(2\pi y) \\ -\cos\left(\frac{t\pi}{T}\right)\sin(\pi y)^2\sin(2\pi x) \end{pmatrix}$

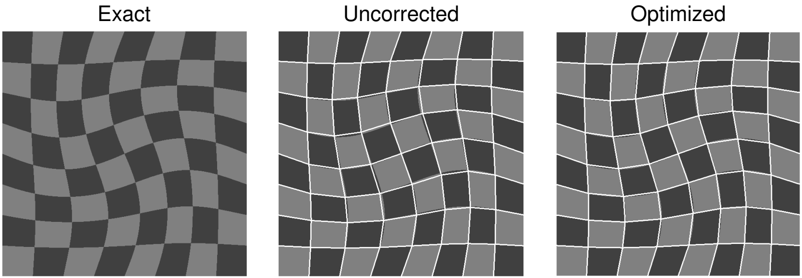
8x8 mesh, $T = 8$, $CFL = 2$, $\beta = 0$, forward Euler for trajectories.



Lagrangian Motion

Swirling velocity field: $\mathbf{u}(\mathbf{x}, t) = \begin{pmatrix} \cos\left(\frac{t\pi}{T}\right) \sin(\pi x)^2 \sin(2\pi y) \\ -\cos\left(\frac{t\pi}{T}\right) \sin(\pi y)^2 \sin(2\pi x) \end{pmatrix}$

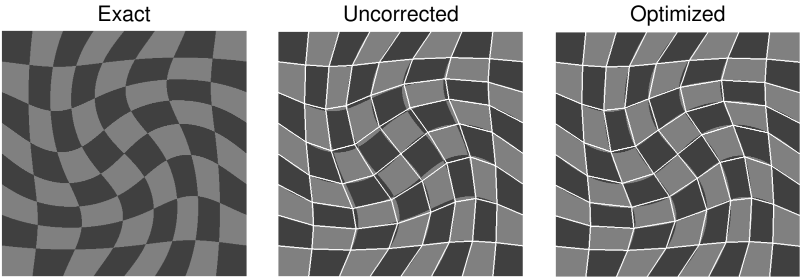
8x8 mesh, $T = 8$, $CFL = 2$, $\beta = 0$, forward Euler for trajectories.



Lagrangian Motion

Swirling velocity field: $\mathbf{u}(\mathbf{x},t) = \begin{pmatrix} \cos\left(\frac{t\pi}{T}\right)\sin(\pi x)^2\sin(2\pi y) \\ -\cos\left(\frac{t\pi}{T}\right)\sin(\pi y)^2\sin(2\pi x) \end{pmatrix}$

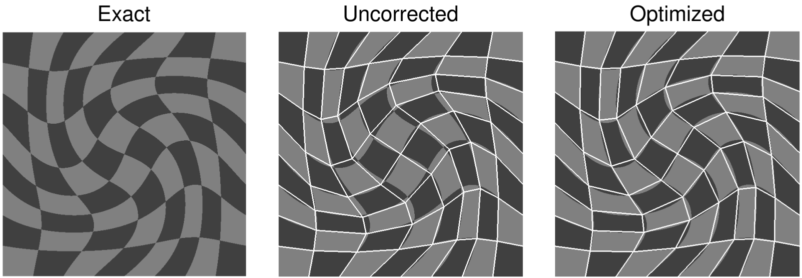
8x8 mesh, $T = 8$, $CFL = 2$, $\beta = 0$, forward Euler for trajectories.



Lagrangian Motion

Swirling velocity field: $\mathbf{u}(\mathbf{x},t) = \begin{pmatrix} \cos\left(\frac{t\pi}{T}\right)\sin(\pi x)^2\sin(2\pi y) \\ -\cos\left(\frac{t\pi}{T}\right)\sin(\pi y)^2\sin(2\pi x) \end{pmatrix}$

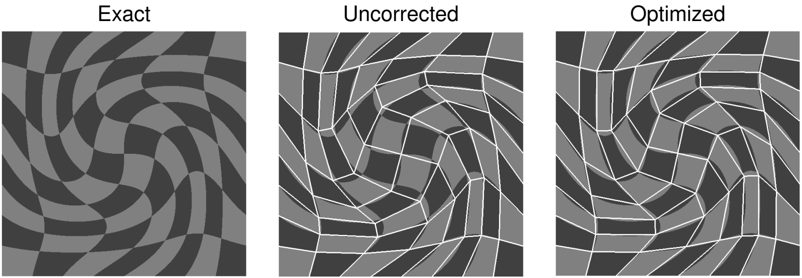
8x8 mesh, $T = 8$, $CFL = 2$, $\beta = 0$, forward Euler for trajectories.



Lagrangian Motion

Swirling velocity field: $\mathbf{u}(\mathbf{x}, t) = \begin{pmatrix} \cos\left(\frac{t\pi}{T}\right)\sin(\pi x)^2\sin(2\pi y) \\ -\cos\left(\frac{t\pi}{T}\right)\sin(\pi y)^2\sin(2\pi x) \end{pmatrix}$

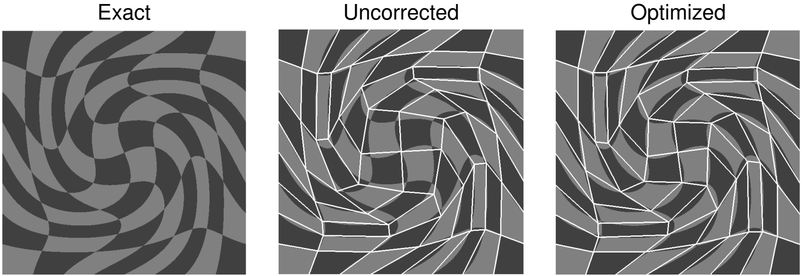
8x8 mesh, $T = 8$, $CFL = 2$, $\beta = 0$, forward Euler for trajectories.



Lagrangian Motion

Swirling velocity field: $\mathbf{u}(\mathbf{x},t) = \begin{pmatrix} \cos\left(\frac{t\pi}{T}\right)\sin(\pi x)^2\sin(2\pi y) \\ -\cos\left(\frac{t\pi}{T}\right)\sin(\pi y)^2\sin(2\pi x) \end{pmatrix}$

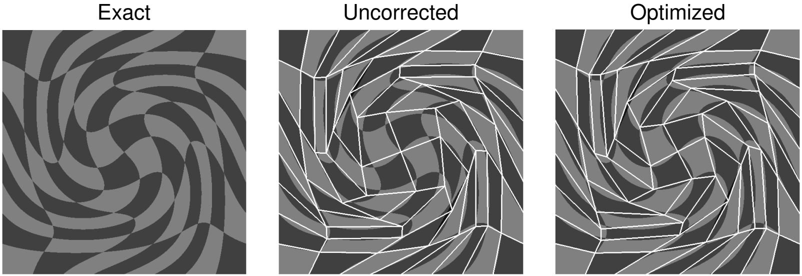
8x8 mesh, $T = 8$, $CFL = 2$, $\beta = 0$, forward Euler for trajectories.



Lagrangian Motion

Swirling velocity field: $\mathbf{u}(\mathbf{x},t) = \begin{pmatrix} \cos\left(\frac{t\pi}{T}\right)\sin(\pi x)^2\sin(2\pi y) \\ -\cos\left(\frac{t\pi}{T}\right)\sin(\pi y)^2\sin(2\pi x) \end{pmatrix}$

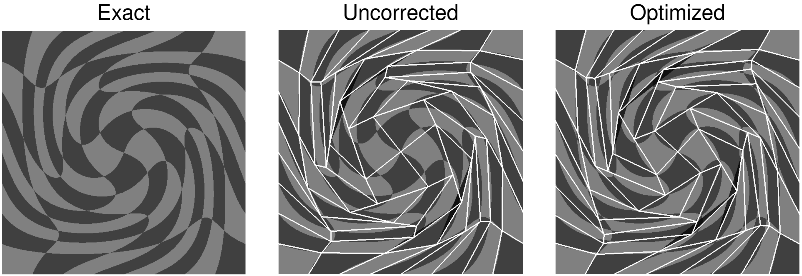
8x8 mesh, $T = 8$, $CFL = 2$, $\beta = 0$, forward Euler for trajectories.



Lagrangian Motion

Swirling velocity field: $\mathbf{u}(\mathbf{x}, t) = \begin{pmatrix} \cos\left(\frac{t\pi}{T}\right)\sin(\pi x)^2 \sin(2\pi y) \\ -\cos\left(\frac{t\pi}{T}\right)\sin(\pi y)^2 \sin(2\pi x) \end{pmatrix}$

8x8 mesh, $T = 8$, $CFL = 2$, $\beta = 0$, forward Euler for trajectories.



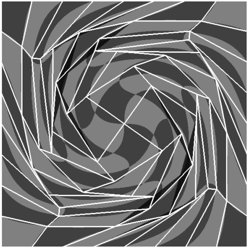
Lagrangian Motion

Swirling velocity field: $\mathbf{u}(\mathbf{x}, t) = \begin{pmatrix} \cos\left(\frac{t\pi}{T}\right) \sin(\pi x)^2 \sin(2\pi y) \\ -\cos\left(\frac{t\pi}{T}\right) \sin(\pi y)^2 \sin(2\pi x) \end{pmatrix}$

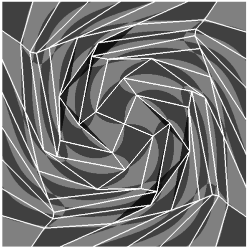
8x8 mesh, $T = 8$, $CFL = 2$, $\beta = 0$, forward Euler for trajectories.

Exact

Uncorrected



Optimized

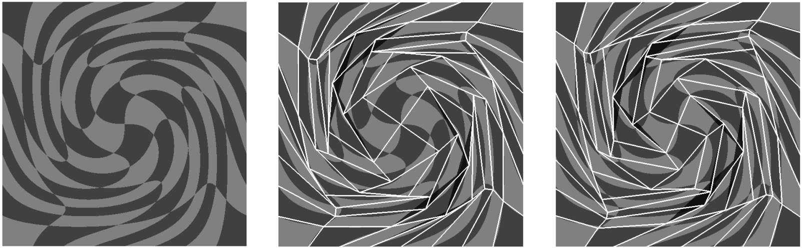


Improvements in Mesh Geometry

We observe significant improvements in the geometry of the corrected mesh:

- The shapes of the corrected cells are closer to the exact Lagrangian shapes

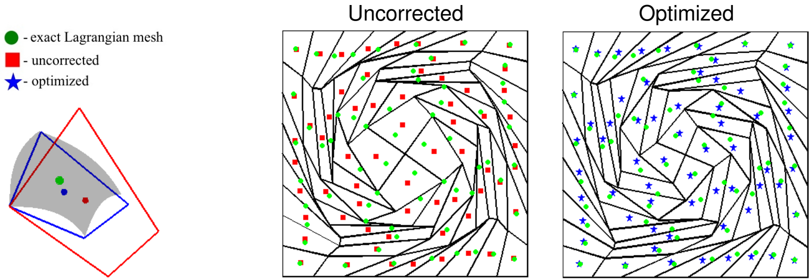
Exact | Uncorrected | Optimized



Improvements in Mesh Geometry

We observe significant improvements in the geometry of the corrected mesh:

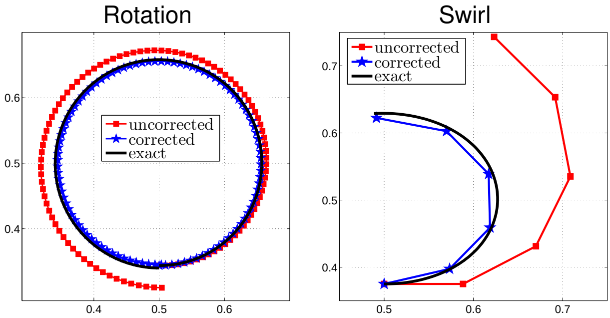
- The shapes of the corrected cells are closer to the exact Lagrangian shapes
- The barycenters of the corrected cells are closer to the exact barycenters



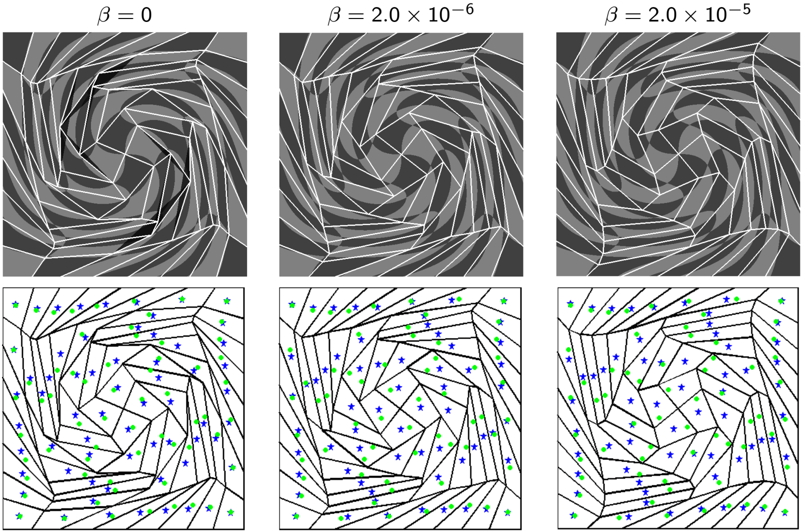
Improvements in Mesh Geometry

We observe significant improvements in the geometry of the corrected mesh:

- The shapes of the corrected cells are closer to the exact Lagrangian shapes
- The barycenters of the corrected cells are closer to the exact barycenters
- The trajectories of the corrected cells are closer to the exact Lagrangian trajectories

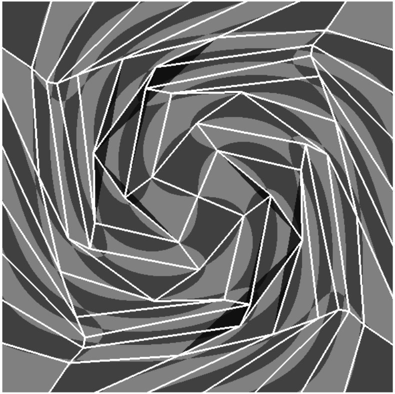


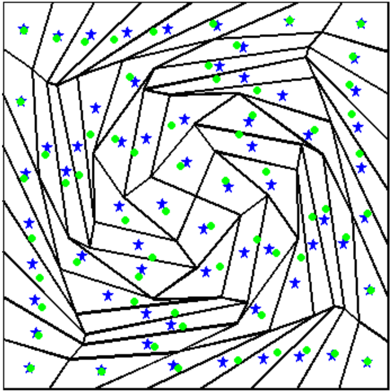
Enforcement of Convexity Constraints



Enforcement of Convexity Constraints

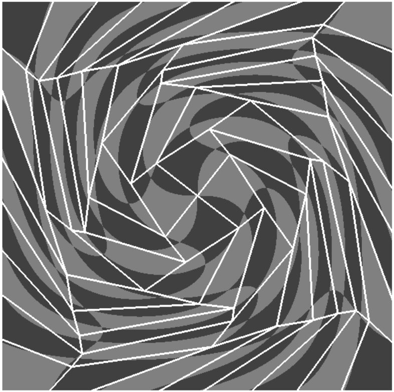
$\beta = 0$

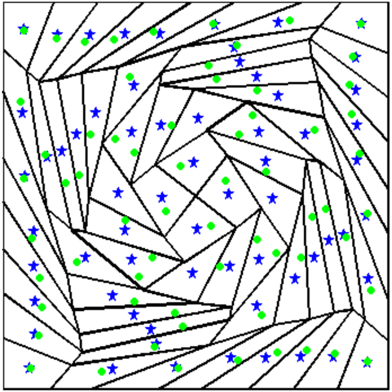




Enforcement of Convexity Constraints

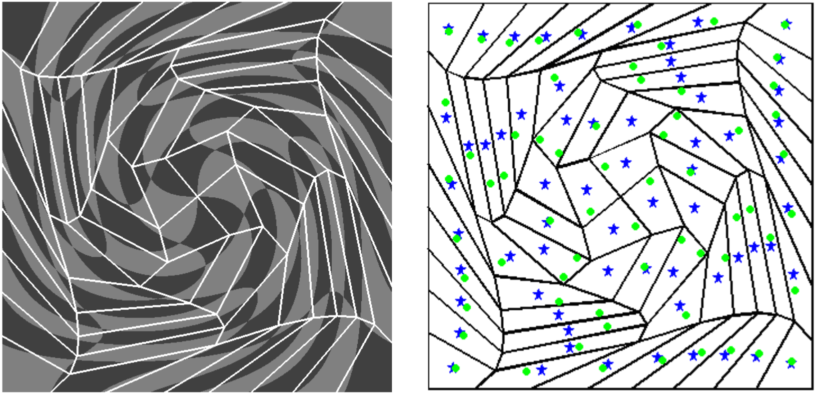
$\beta = 2.0 \times 10^{-6}$





Enforcement of Convexity Constraints

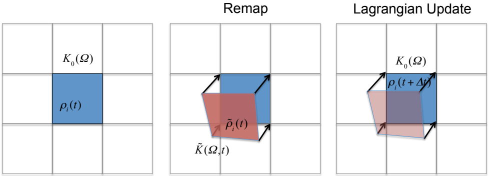
$\beta = 2.0 \times 10^{-5}$



Application to Semi-Lagrangian Transport

Want to solve: $\dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$

Given cell volume $c_i = \int_{\kappa_i} dV$, cell mass $m_i = \int_{\kappa_i} \rho(\mathbf{x}, t) dV$, and cell average density $\rho_i = m_i / c_i$ at time t

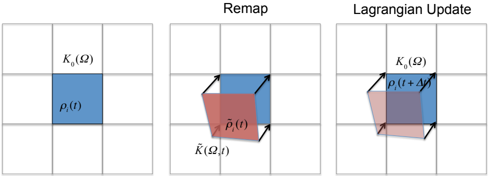


1. Define Lagrangian departure cells: $c_i \rightarrow \tilde{c}_i$
2. Remap from fixed grid to departure grid: $\rho_i \rightarrow \tilde{\rho}_i, \; \tilde{m}_i = \tilde{\rho}_i \tilde{c}_i$
3. Lagrangian update: $m_i(t + \Delta t) = \tilde{m}_i, \; \rho_i(t + \Delta t) = m_i(t + \Delta t)/c_i$

Application to Semi-Lagrangian Transport

Want to solve: $\dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$

Given cell volume $c_i = \int_{\kappa_i} dV$, cell mass $m_i = \int_{\kappa_i} \rho(\mathbf{x}, t)dV$, and cell average density $\rho_i = m_i/c_i$ at time t

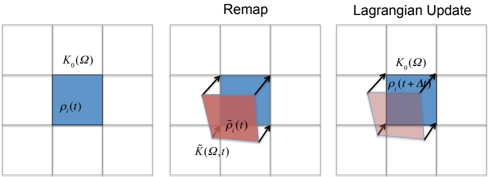


1. Define Lagrangian departure cells: $c_i \rightarrow \hat{c}_i$
2. Volume correction: $\hat{c}_i \rightarrow \tilde{c}_i$
3. Remap from fixed grid to departure grid: $\rho_i \rightarrow \tilde{\rho}_i,\ \tilde{m}_i = \tilde{\rho}_i \tilde{c}_i$
4. Lagrangian update: $m_i(t+\Delta t) = \tilde{m}_i,\ \rho_i(t+\Delta t) = m_i(t+\Delta t)/c_i$

Application to Semi-Lagrangian Transport

Want to solve: $\dfrac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$

Given cell volume $c_i = \int_{\kappa_i} dV$, cell mass $m_i = \int_{\kappa_i} \rho(\mathbf{x}, t) dV$, and cell average density $\rho_i = m_i / c_i$ at time t



1. Define Lagrangian departure cells: $c_i \rightarrow \hat{c}_i$
2. Volume correction: $\hat{c}_i \rightarrow \tilde{c}_i$
3. Remap from fixed grid to departure grid: $\rho_i \rightarrow \tilde{\rho}_i, \; \tilde{m}_i = \tilde{\rho}_i \tilde{c}_i$
4. Lagrangian update: $m_i(t + \Delta t) = \tilde{m}_i, \; \rho_i(t + \Delta t) = m_i(t + \Delta t)/c_i$

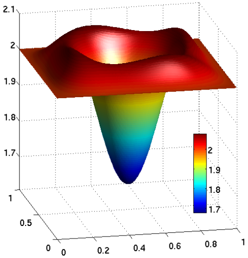
We use linear reconstruction of density with Van Leer limiting for remap.

Dukowicz and Baumgardner (2000), Incremental remapping as a transport/advection algorithm, *JCP*.

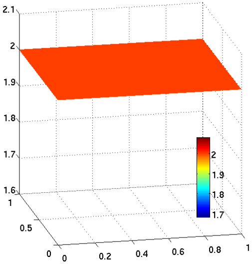
Semi-Lagrangian Transport Results

Constant density, rotational flow

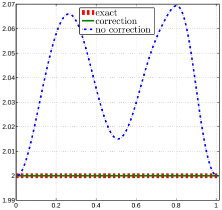
Uncorrected



Corrected



Comparison

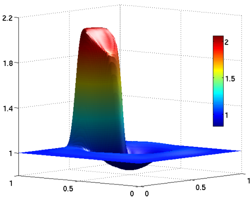


Forward Euler with $\Delta t = 0.006$.

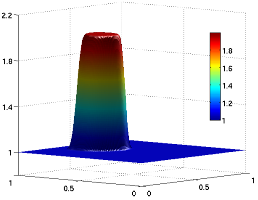
Semi-Lagrangian Transport Results

Cylindrical density, rotational flow

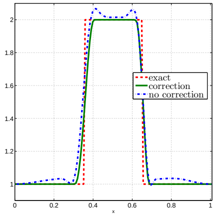
Uncorrected



Corrected



Comparison



Forward Euler with $\Delta t = 0.006$.

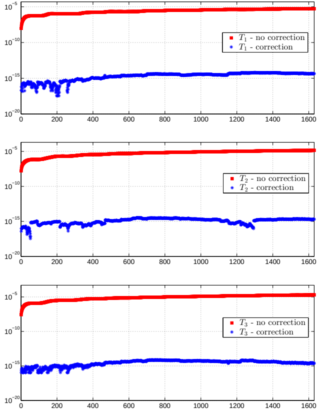
Multi-Material Semi-Lagrangian Transport

Consider transport of volume fraction of material s

$$\frac{\partial T_s}{\partial t} + \boldsymbol{u} \cdot \nabla T_s = 0, \quad s = 1, \ldots, 3,$$

$$T_{s,i}(t) = \frac{\int_{\kappa_i(t)} T_s \, dV}{\int_{\kappa_i(t)} dV} = \frac{|\kappa_{s,i}(t)|}{|\kappa_i(t)|}$$

$$|\Omega_s^n| = \sum_{i=1}^{m} T_{s,i}^n |\kappa_i|$$



Without volume correction:

$$|\Omega_s^{n+1}| = \sum_{i=1}^{m} T_{s,i}^{n+1} |\kappa_i| = \sum_{i=1}^{m} \widetilde{T}_{s,i}^{n} |\kappa_i| \neq \sum_{i=1}^{m} \widetilde{T}_{s,i}^{n} |\widetilde{\kappa}_i(t^n)| = |\widetilde{\Omega}_s^n| = |\Omega_s^n| \, .$$

Conclusions

Presented a new approach for improving the accuracy and physical fidelity of numerical schemes that rely on Lagrangian mesh motion

- Optimization-based volume correction
  - Is computationally efficient
  - Provides significant geometric improvements in corrected meshes
  - Enables semi-Lagrangian transport methods to preserve volumes and constant densities
- Future work
  - Further development of mesh quality constraints and rigorous enforcement of mesh validity
  - Investigate utility of algorithm for mesh quality improvement