A Generalization of Prather's Method for Tracer Advection

Robert B. Lowrie

Los Alamos National Laboratory
Computational Physics and Methods Group (CCS-2)
lowrie@lanl.gov

CCSM Ocean Model Working Group Meeting
National Center for Atmospheric Research
Boulder, Colorado
10 - 11 December 2009

Collaborator
Todd Ringler (T-3 LANL)

Overview

- Generalize of Prather's moment method (JGR 1986) to unsplit advection on general mesh topologies
- Take advantage of existing Lagrange-remap algorithms (Lipscomb & Ringler, MWR 2005)
- Resulting method: Characteristic Discontinuous Galerkin (CDG), which is based on space-time discontinuous Galerkin
- Ultimate goal: Minimize spurious diapycnal mixing (e.g., Griffies et al 2000)
    - Here, our approach is to increase the order-of-accuracy

Outline

Outline

Tracer Advection

- Given $\vec{u}(\vec{x}, t)$, solve

$$\partial_t \rho + \nabla \cdot (\rho \vec{u}) = 0\,, \tag{1a}$$
$$\partial_t (\rho T) + \nabla \cdot (\rho T \vec{u}) = 0\,. \tag{1b}$$

- Implies

$$\frac{DT}{Dt} = 0\,, \qquad \frac{D}{Dt} \equiv \partial_t + \vec{u} \cdot \nabla\,.$$

- To ensure conservation, we discretize the system (1).

Overview of Prather's Moment Method

- Within each individual mesh cell, maintain a quadratic representation of $T(x, y)$:

$$T(x, y) = \sum_{p=0}^{2} \sum_{q=0}^{2-p} c_{p,q} x^p y^q.$$

- The 6 coefficients $c_{p,q}$ may be related to 6 moments of $T(x, y)$:

$$m_{p,q} = \iint_{\text{cell}} x^p y^q T(x, y)\, dxdy.$$

  Forms 6×6 linear system.

- The coefficients are updated in time as follows:
  1. Advect the polynomial representation.
  2. Compute the new moments of the advected solution.
  3. Back out the polynomial coefficients from the moments.

Some Details of Prather's Method

- Instead of "$x^p y^q$," other bases may be used. Prather used tensor-product Legendre polynomials.
- Prather uses dimensional splitting:
  - + Reduces geometric complexity
  - + Simplifies limiting
  - – Restricts time accuracy to at best $O(\Delta t^2)$.
  - – May give mesh imprinting.
  - – **Restricts the method to logically-rectangular meshes.**

Outline

Solution Representation for quadratic CDG the same as for Prather

- At each time-level n, within each mesh cell Ω_k, expand solution as

$$T(\vec{x}, t^n) = \sum_{j=1}^{N} c_{k,j}^n \beta_{k,j}(\vec{x}), \quad \vec{x} \in \Omega_k .$$

- Example choice for $\beta_{k,j}(\vec{x})$ and $N = 6$ (quadratic):

$$\beta_{k,j}(\vec{x}) \in \{1, x, y, x^2, xy, y^2\},$$

  or some linear combination thereof (such as Legendre polynomials).
- Need method for updating coefficients $c_{k,j}^n$ for each cell-k.

Some manipulations...

- Begin with

$$\partial_t(\rho T) + \nabla \cdot (\rho T \vec{u}) = 0\,.$$

- Multiply by a smooth function $\phi_{k,i}(\vec{x}, t)$ and rearrange:

$$\partial_t(\phi_{k,i}\rho T) + \nabla \cdot (\phi_{k,i}\rho T \vec{u}) = \rho T \frac{D\phi_{k,i}}{Dt}\,.$$

- Weak form over a control volume $\Omega_k \times [t^n, t^{n+1}]$:

$$\int\limits_{\Omega_k} \left[(\phi_{k,i}\rho T)^{n+1} - (\phi_{k,i}\rho T)^n \right] d\Omega + \int\limits_{t^n}^{t^{n+1}} \oint\limits_{\partial\Omega_k} \phi_{k,i}\rho T \vec{u} \cdot \vec{n}\, ds dt = \int\limits_{t^n}^{t^{n+1}} \int\limits_{\Omega_k} \rho T \frac{D\phi_{k,i}}{Dt}\, d\Omega dt\,.$$

- This form is used by space-time discontinuous Galerkin to update the $c^n_{k,j}$.

The CDG Approach

- Replace

$$\partial_t(\phi_{k,i}\rho T) + \nabla\cdot(\phi_{k,i}\rho T\vec{u}) = \rho T\frac{D\phi_{k,i}}{Dt}\,,$$

with the system

$$\partial_t(\phi_{k,i}\rho T) + \nabla\cdot(\phi_{k,i}\rho T\vec{u}) = 0\,, \tag{2a}$$

$$\frac{D\phi_{k,i}}{Dt} = 0\,. \tag{2b}$$

- Because we seek $\frac{DT}{Dt} = 0$, eq. (2b) might seem redundant. However,
  - (2a) maintains conservation
  - (2b) is local to each element and can be solved once for all tracers

Solving $\partial_t(\phi_{k,i}\rho T) + \nabla \cdot (\phi_{k,i}\rho T\vec{u}) = 0$

- For a polygon Ω_k with faces $\partial\Omega_{k,f}$, CDG solves the integral form

$$\int_{\Omega_k} \left[(\phi_{k,i}\rho T)^{n+1} - (\phi_{k,i}\rho T)^n \right] d\Omega + \sum_f \int_{\Omega'_{k,f}} \left(\phi_{k,i}\rho T\right)^n d\Omega = 0\,,$$

  where $(\Omega'_{k,f}, t^n)$ is the Lagrangian pre-image of the face $\partial\Omega_{k,f} \times [t^n, t^{n+1}]$.
- $\Omega'_{k,f}$ geometry computed using Lagrange-remap (*e.g.*, Lipscomb & Ringler, MWR 2005)
- Integration approximated using quadrature
- First term represents change in the $\phi_{k,i}$-moment, analogous with Prather
- Still need to define $\phi_{k,i}(\vec{x}, t)$

Solving $D\phi_{k,i}/Dt = 0$

- Use a "semi-Lagrangian" approach
- Recall that our solution in each cell is given by

$$T(\vec{x}, t^n) = \sum_{j=1}^{N} c_{k,j}^n \beta_{k,j}(\vec{x}), \quad \vec{x} \in \Omega_k .$$

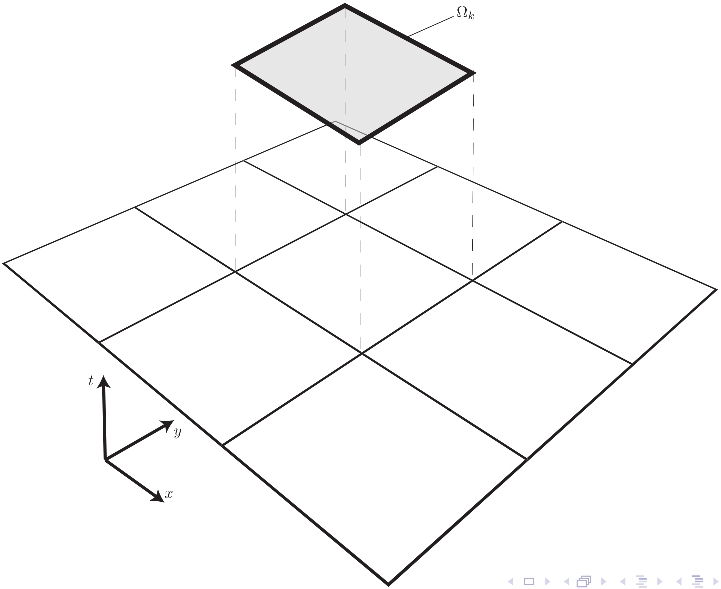
- For a given time interval $t^n \leq t \leq t^{n+1}$, a solution to $D\phi_{k,i}/Dt = 0$ is $\phi_{k,i}(\vec{x}, t) = \beta_{k,i}(\vec{\Gamma}(\vec{x}, t))$, where

$$\begin{aligned} \vec{\Gamma}(\vec{x}, t) &= \vec{x} + \int_{t}^{t^{n+1}} \vec{u}(\vec{\Gamma}(\vec{x}, \xi), \xi)\, d\xi \\ &= \vec{x} + (t^{n+1} - t)\vec{u}, \qquad \text{for } \vec{u} = \text{const.} \end{aligned}$$

- Integration of characteristics needed once for ALL tracers.

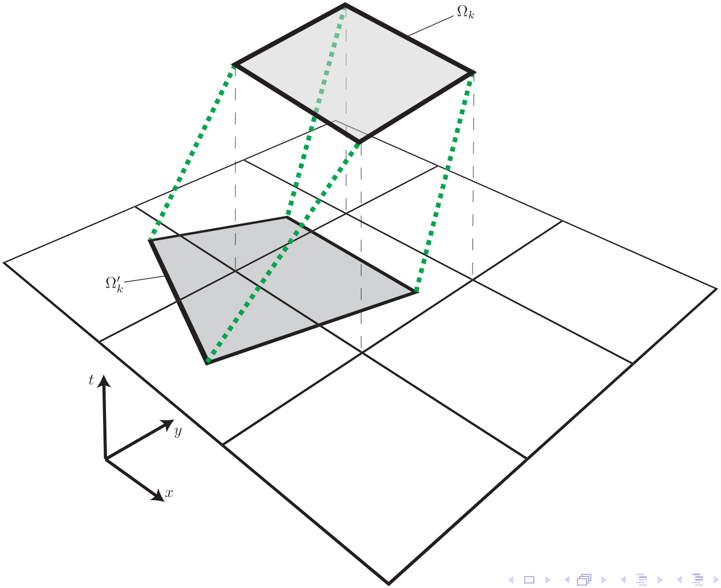
CDG on a Cartesian Mesh

Quest: Find polynomial representation of solution in center cell at new time level.



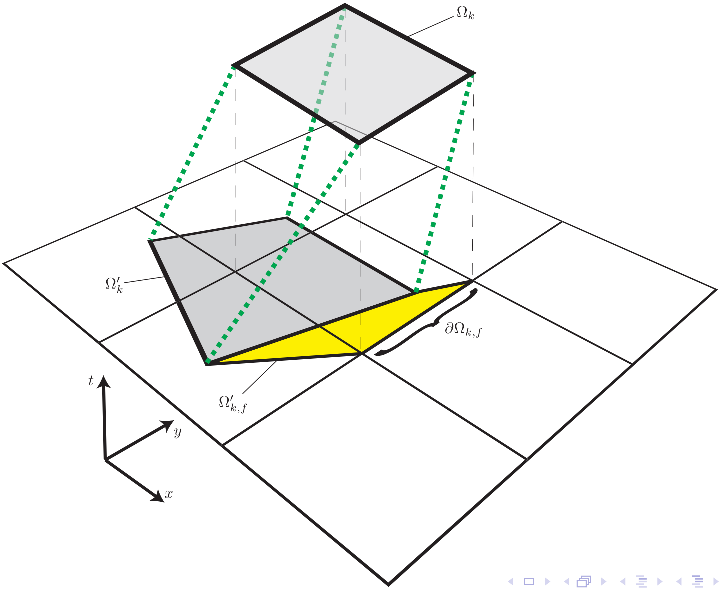
"Semi-Lagrangian" Step

Trace characteristics at each node from t^{n+1} to t^n (use RK4)



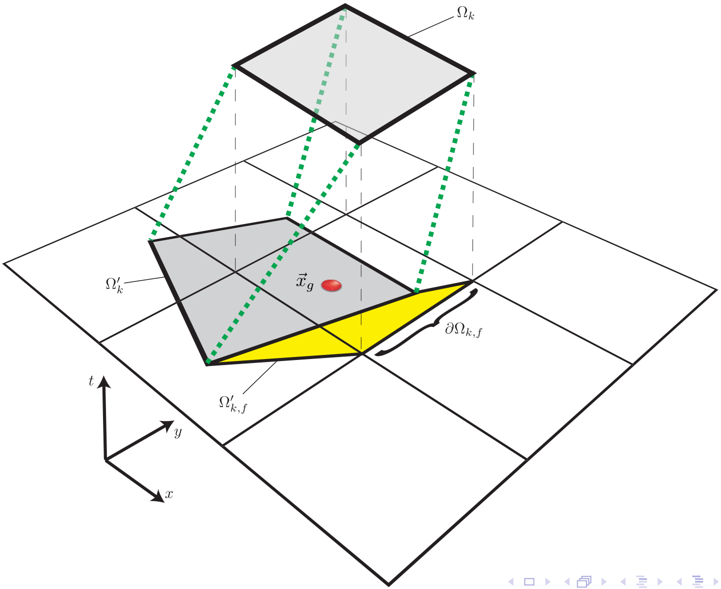
Find Lagrangian pre-image for each face...

...and break into triangles; see Lipscomb & Ringler (MWR 2005)



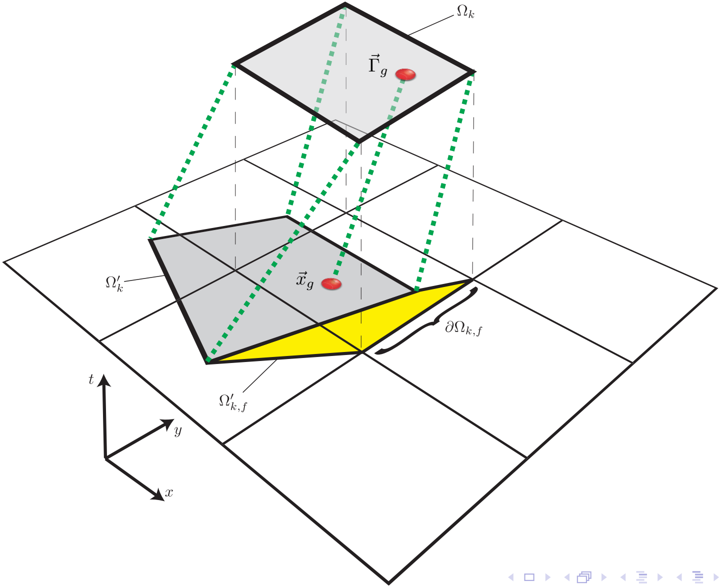
Evaluate each integral with quadrature

Below is an example quadrature point, $\vec{x}_g$



At each quadrature point, trace characteristics...

... from t^n to t^{n+1} to determine $\phi(\vec{x}_g, t^n) = \beta(\vec{\Gamma}_g)$



Properties of CDG(p)

- CDG(p) uses a polynomial basis of order-p, with $p \geq 0$.
- With incremental remap (Lipscomb & Ringler, MWR 2005), stable for CFL < 1. Larger time steps possible with general remap.
- Locally conservative
- At a fixed CFL, error is typically $O(\Delta x^{p+1})$ in space *and* time
  - But "quasi-accurate:" If pre-image is non-polygonal, then current remap limits overall accuracy to $O(\Delta x^2)$
- Parallelizes well with a *single* communication per Δt
- Our bounds preserving limiter maintains order-of-accuracy for smooth solutions
  - Enforces $T_{\text{min}} \leq T \leq T_{\text{max}}$

CDG(p): Relationship to Other Methods

- In 1-D with mass coordinates (or ρ, $\vec{u}$ constant):
  - CDG(0) is equivalent to first-order upwind
  - CDG(1) is equivalent to:
    - ★ Van Leer's Scheme III (JCP 1977, "exact evolution with L^2-projection")
    - ★ Russell & Lerner's method (JAM 1981)
  - CDG(2) is equivalent to:
    - ★ Van Leer's Scheme IV (JCP 1977)
    - ★ Prather's method (JGR 1986)
    - ★ Piecewise-Parabolic Boltzmann (PPB) (Woodward 1986)
- Can be viewed as the following extensions to Prather's method:
  - Any $p \geq 0$ (Prather: $p = 2$)
  - General mesh topologies (Prather: Cartesian)
  - Dimensionally unsplit (Prather: split)
  - Triangle or diamond basis truncation (Prather: triangle)

Outline

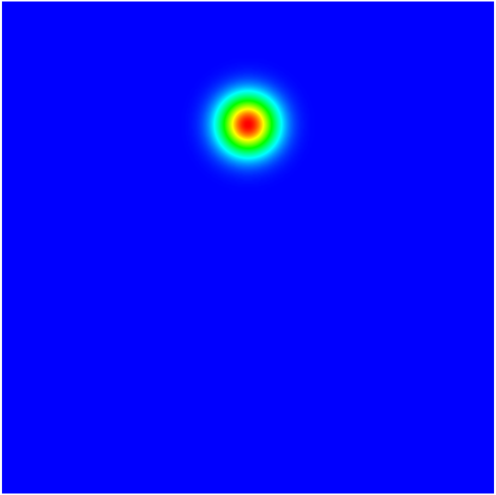
Common Properties of Test Cases

- ρ constant
- 2-D unit square, doubly periodic
- Cartesian mesh, $\Delta x = \Delta y$
- CFL $= 0.8$
- CDG(p) used tensor-product Legendre polynomials with triangle truncation

Solid-Body Rotation of a Gaussian Bump

Gaussian bump rotates about center of domain.

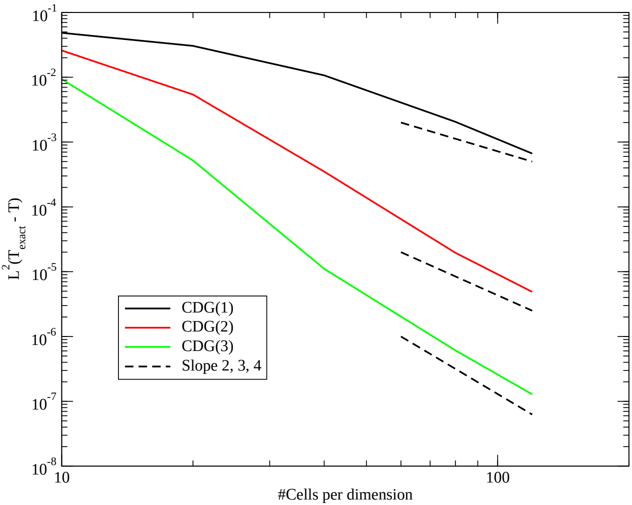
$t = 0$ and $t = 1$



Errors for Solid-Body Rotation of a Gaussian Bump

After 1 rotation

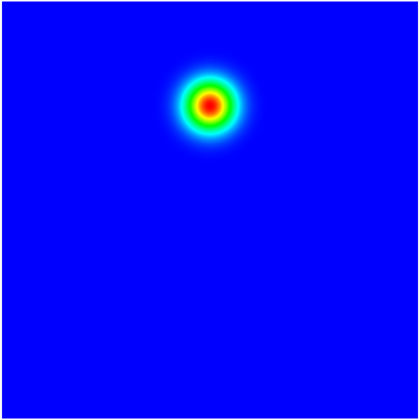
In this case, each cell's Lagrangian pre-image are nearly polygonal.



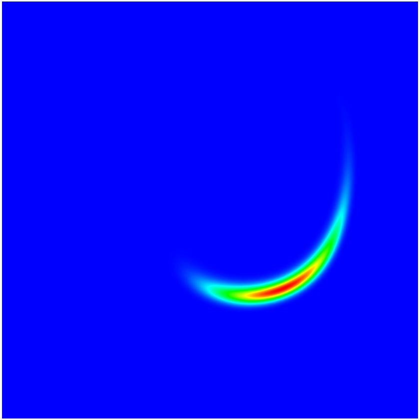
Deformation of a Gaussian Bump (period = 2)

Stream function: $\psi(x, y, t) = \cos(\pi t/2)\sin^2(\pi x)\sin^2(\pi y)/\pi$. Compute errors at $t = 2$.

$t = 0$ and $t = 2$



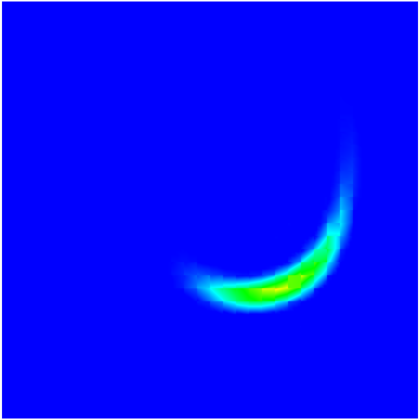
$t = 1$



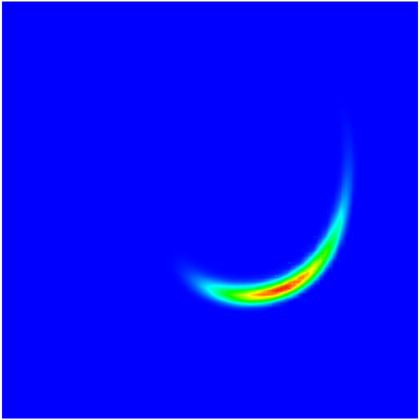
Sample Results at $t = 1$

32×32 Mesh, exact $T_{\max} = 1$. Both methods used the same Δt (CFL $= 0.8$)

CDG(1), $T_{\max} = 0.7997$



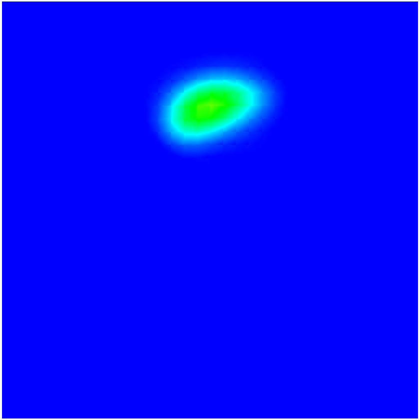
CDG(3), $T_{\max} = 1.0170$



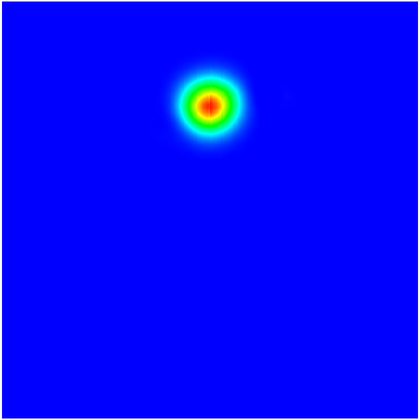
Sample Results at $t = 2$

32×32 Mesh, exact $T_{\max} = 1$. Approximately 4 cells across initial Gaussian.

CDG(1), $T_{\max} = 0.6080$



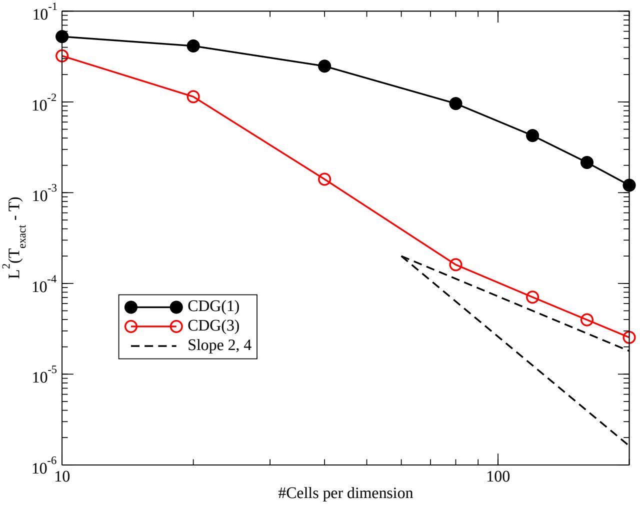
CDG(3), $T_{\max} = 0.9872$



CDG(2), $T_{\max} = 0.8685$

Errors for Deformation of a Gaussian Bump

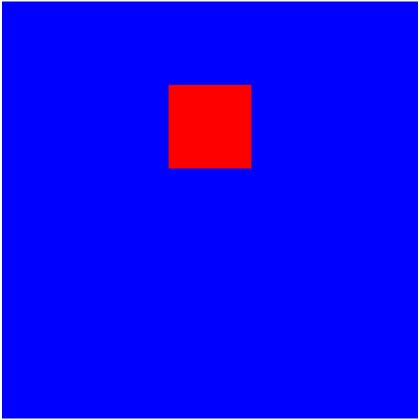
Lagrangian pre-image non-polygonal ⇒ CDG accuracy limited to 2nd-order



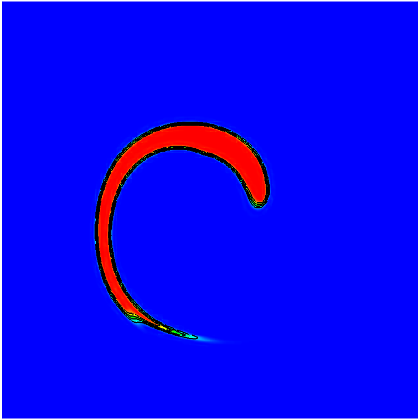
Deformation of a Square (period = 4)

$\psi(x, y, t) = \cos(\pi t/4)\sin^2(\pi x)\sin^2(\pi y)/\pi$, 80×80 mesh (square 16×16)

$t = 0$ and $t = 4$



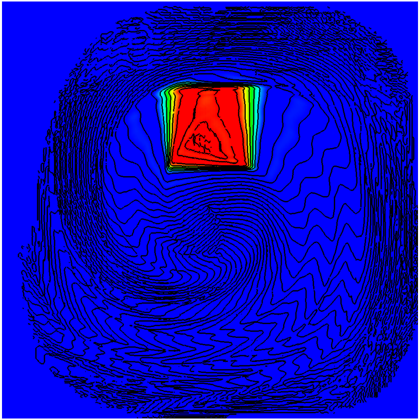
$t = 2$



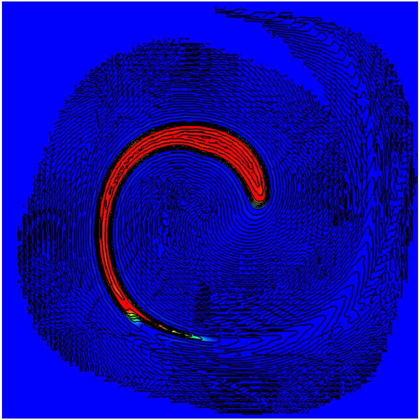
Deformation of a Square (period = 4): No Limiter

Exceeds bounds: $-0.25 \lesssim T \lesssim 1.32$

$t = 4$



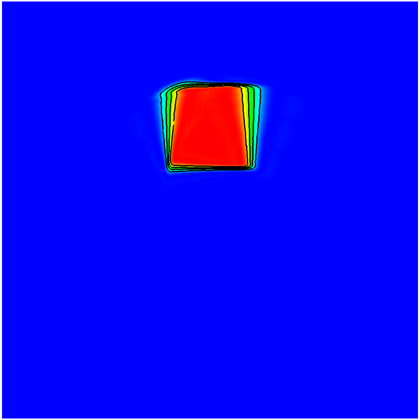
$t = 2$



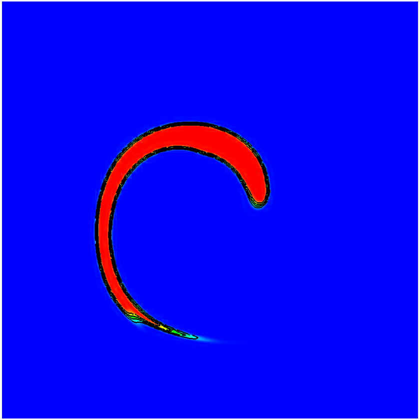
Deformation of a Square (period = 4): With Limiter

Enforces bounds: $0 \leq T \leq 1$

$t = 4$

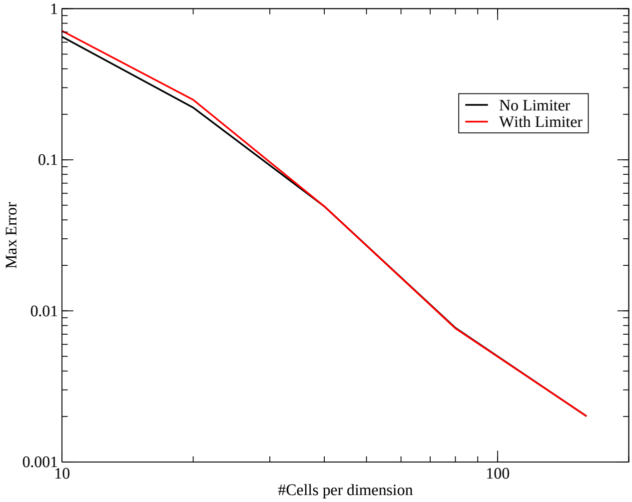


$t = 2$



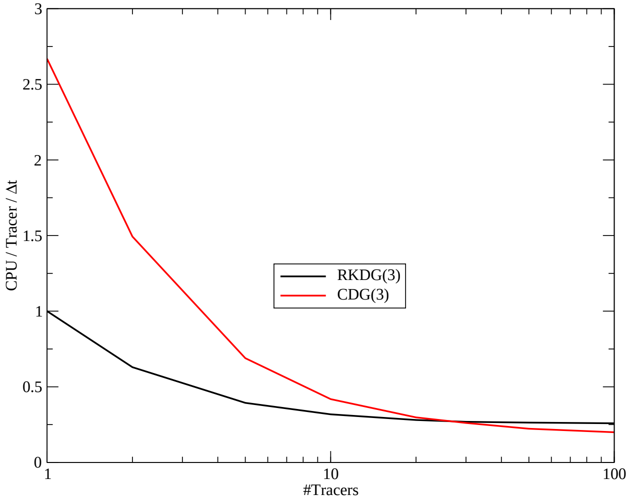
Limiter maintains accuracy for smooth problems

Deformation of Gaussian bump



Scaling of CPU Time with Number of Tracers

Results normalized by RKDG(3) time for 1 tracer



Summary and Future Work

Summary:

- CDG(p) with incremental remap is
  - stable for CFL < 1
  - $O(\Delta x^{p+1})$ accurate in space and time whenever pre-image is a polygon; otherwise, $O(\Delta x^2)$
- Computational cost increases roughly as 2^p
- Majority of computational work independent of number of tracers
- Van Leer IV, Prather, PPB, ... $\Rightarrow$ CDG(2)

Future work:

- Couple with fluid models
- Other meshes
  - Voronoi mesh: Matthew Buoni (LANL, T-5)
- Adaptive-p