A Lagrangian particle method for scalar transport on the sphere

Peter A. Bosler
pbosler@umich.edu
Acknowledgements

- Work supervised by Robert Krasny and Christiane Jablonowski
  - Advisors: Robert Krasny and John Boyd
- Thanks to NCAR and workshop organizers
Lagrangian particle method

- Particles move with the flow
- Able to use large time-steps
- Context: Poisson equation on the sphere: \( \nabla^2 \psi = -\zeta \)
  
  - 1. Finite difference, finite element, finite volume methods
  
  - 2. Spectral methods
  
  - 3. Green’s function \( \psi(x) = -\int_{S^2} G(x, \tilde{x}) \zeta(\tilde{x}) d\tilde{A} \)

- Chorin 1973: Vortex-blob method
- Cottet & Koumoutsakos 2000: Vortex methods
- Lindsay & Krasny 2001: Cartesian treecode \( \mathcal{O}(N^2) \rightarrow \mathcal{O}(N \log N) \)
- Dritschel 2004: Contour dynamics
Discretization of unit sphere

- Represent the sphere as a union of panels
  \[ S^2 = \bigcup P_k, \quad k = 1, \ldots, N \]
- Panel k: Area = \( A_k \), Scalar = \( \phi_k \), Center = \( x_k \)
  \[ A_k = A_{k1} + A_{k2} \]
- Midpoint rule quadrature (for example):
  \[ \int_{S^2} \phi \, dA = \sum_{k=1}^{N} \int_{P_k} \phi \, dA \approx \sum_{k=1}^{N} \phi_k A_k \]
Advection equations

- Given:
  \[ u = u(\lambda, \theta, t) \quad v = v(\lambda, \theta, t) \]

- Spherical coordinates:
  \[ \frac{d\lambda}{dt} = \frac{u}{\cos \theta}, \quad \frac{d\theta}{dt} = v \]

- Cartesian coordinates:
  \[ \frac{dx}{dt} = -u \frac{y}{\sqrt{x^2 + y^2}} - v \frac{xz}{\sqrt{x^2 + y^2}} \]
  \[ \frac{dy}{dt} = u \frac{x}{\sqrt{x^2 + y^2}} - v \frac{yz}{\sqrt{x^2 + y^2}} \]
  \[ \frac{dz}{dt} = v \sqrt{x^2 + y^2} \]

For BVE/SWE:
- No pole problem
- Cartesian treecode,
- Velocity from Biot-Savart Law
Gaussian hills: Low-resolution

$0 \leq t \leq T$

$\Delta x_0 = 5.64^\circ$

$N = 1536$

$\Delta t = T/80$

$CFL_0 = 2.07$

$\varepsilon = 0.104$

$MSR = 0.95$

$CPU = 17.74$ s
Distorted mesh $t = T/2$

$\Delta x_0 = 2.82^\circ$

$N = 6144$

$\Delta t = T/80$

$CFL_0 = 4.135$

$\varepsilon = 0.052$

$MSR = 0.95$

$CPU = 62.6$ s
Interpolation from Lagrangian particles to regular mesh

- **Local**: Piecewise polynomials
  - RBF
  - etc..

- **Global**: Spherical harmonics
  - Green’s function (Bogomolov 1977; Kimura and Okamoto 1987)

- **Regularized Green’s function**:
  - Removes singularity
  - Enhances numerical stability
  - Sub-grid filter

\[
\phi(x) = \int_{S^2} \delta(x - \tilde{x}) \phi(\tilde{x}) \, d\tilde{A} \approx \sum_{k=1}^{N} \delta_\epsilon(x - x_k) \phi_k A_k
\]

- **Two sources of error**:
  1. Smoothing error
  2. Discretization error

Data from Panel k
Grid size vs. Smoothing parameter

- We define the “mesh/smoothing ratio”, MSR as:

\[ MSR = \frac{\Delta x_0}{\epsilon} \]

- Given discretization parameter \( \Delta x_0 \):
  - Choose appropriate smoothing parameter \( \epsilon \)

- Given a smoothing parameter \( \epsilon \):
  - Find the resolution \( \Delta x_0 \) required to resolve that scale

Interpolation Tests:
Interpolate \( \phi \) (Gaussian hills) from cubed sphere to 1°x1° lat-long grid
Look for optimal MSR
Mesh/smoothing ratio vs. $l_2$ Interpolation Error

Interpolation Tests:
Plot convergence for $MSR_1 = 0.90$ $MSR_2 = 1.15$
Delta-convolution Interpolation Error
Cubed sphere to $1^\circ \times 1^\circ$ lat-long grid

\[ l_1 \]
\[ l_2 \]

\[ \Delta x_0 = 5.6^\circ \quad 2.8^\circ \quad 1.4^\circ \quad 0.71^\circ \quad 0.35^\circ \]

\[ N = 1536 \quad 6144 \quad 24576 \quad 98304 \quad 393216 \]

\[ O(N^{-1}) \]
\[ O(N^{-2}) \]
Delta-convolution Interpolation Error
Cubed sphere to $1^\circ \times 1^\circ$ lat-long grid

$$l_\infty$$

$\phi_{\text{max}}$ and $\phi_{\text{min}}$

$O(N^{-1})$

$O(N^{-2})$

$\Delta x_0 = 5.6^\circ, 2.8^\circ, 1.4^\circ, 0.71^\circ, 0.35^\circ$

$N = 1536, 6144, 24576, 98304, 393216$

$\phi_{\text{max}}$

$\phi_{\text{min}}$

$O(N^{-1})$
Gaussian hills: Low-resolution

$\Delta x_0 = 5.64^\circ$
$N = 1536$
$\Delta t = T/80$
$CFL_0 = 2.07$
$\varepsilon = 0.104$
$MSR = 0.95$
$CPU = 17.74 \text{ s}$
Gaussian Hills: Medium resolution

$\Delta x_0 = 2.82^\circ$

$N = 6144$

$\Delta t = T/80$

$CFL_0 = 4.135$

$\varepsilon = 0.052$

$MSR = 0.95$

$CPU = 62.6 \text{ s}$
Gaussian Hills : High resolution

\[ \Delta x_0 = 0.71^\circ \]
N = 98304
\[ \Delta t = T/80 \]
CFL_0 = 16.54
\[ \varepsilon = 0.013 \]
MSR = 0.95
CPU = 17.3 min
Cosine bells: Medium resolution

\[ \Delta x_0 = 2.82° \]
\[ N = 6144 \]
\[ \Delta t = T/80 \]
\[ CFL_0 = 4.135 \]
\[ \varepsilon = 0.052 \]
\[ MSR = 0.95 \]
\[ CPU = 62.6 \text{ s} \]
Cosine bells: High resolution

\[ \Delta x_0 = 0.71^\circ \]

\[ N = 98304 \]

\[ \Delta t = T/80 \]

\[ \text{CFL}_0 = 16.54 \]

\[ \varepsilon = 0.013 \]

\[ \text{MSR} = 0.95 \]

\[ \text{CPU} = 17.3 \text{ min} \]

---

Latitude  Longitude  CFL = 16.5401
t = 2.5, 98304 panels, \( \varepsilon = 0.012967 \), mesh health = 100

Latitude  Longitude  CFL = 16.5401
t = 5, 98304 panels, \( \varepsilon = 0.012967 \), mesh health = 100
Δx₀ = 2.82°
N = 6144
Δt = T/80
CFL₀ = 4.135
ε = 0.052
MSR = 0.95
CPU = 62.6 s
Slotted cylinders: High resolution

\[ \Delta x_0 = 0.71^\circ \]

\[ N = 98304 \]

\[ \Delta t = T/80 \]

\[ \text{CFL}_0 = 16.54 \]

\[ \varepsilon = 0.013 \]

\[ \text{MSR} = 0.95 \]

\[ \text{CPU} = 17.3 \text{ min} \]
Mass Error

- Lagrangian dynamics: \( \phi(x(t)) = \phi(x(0)) \)

- Relative mass error:
  \[
  E = \frac{|\int_{S^2} \phi \, dA - \int_{S^2} \phi \, dA_0|}{\int_{S^2} \phi \, dA_0}
  \]
Gaussian hills  \textbf{Mass Error : } t = T/2  Cosine bells

- Midpoint rule in 2D : $\mathcal{O}(N^{-1})$ as $N \to \infty$
- Better than expected convergence down to $\approx 1.5^\circ$
- Linear (expected) convergence for resolutions better than $\approx 1.5^\circ$
Correlated cosine bells: Low / Medium resolution

Low Res. $t = T$

$\Delta x_0 = 5.64^\circ \quad N = 1536 \quad \varepsilon = 0.104 \quad \text{MSR} = 0.95$

Med. Res. $t = T$

$\Delta x_0 = 2.82^\circ \quad N = 6144 \quad \varepsilon = 0.052 \quad \text{MSR} = 0.95$
Correlated cosine bells: High resolution

$\Delta x_0 = 0.71^\circ \quad N = 98304 \quad \varepsilon = 0.013 \quad \text{MSR} = 0.95$
Current status & future work

• Interpolation errors are too large at $t = T/2$

• To improve interpolation, improve quadrature:
  
  • Adaptive mesh refinement

$$\epsilon_{\text{tol}} = c (\max \phi - \min \phi)$$

Refine panel $k$ if

$$\max_{i=1,\ldots,4} |\phi_k - \phi_{k_i}| > \epsilon_{\text{tol}}$$
\[ 1.41 \leq \Delta x_0 \leq 5.64 \]
\[ N = 3360 \]
\[ CFL_0 = 4.135 \]
\[ \varepsilon = 0.068 \]
\[ MSR = 0.725 \]
\[ CPU = 38.5\text{s} \]

Compare to:
\[ \Delta x_0 = 2.82 \]
\[ CFL_0 = 4.135 \]
\[ N = 6144 \]
\[ \varepsilon = 0.052 \]
\[ MSR = 0.95 \]
\[ CPU = 62.6\text{s} \]
Uniform mesh interpolation

\[ \Delta x_0 = 0.71^\circ \]
\[ N = 98,304 \]
\[ CFL_0 = 16.54 \]
\[ \varepsilon = 0.013 \]
\[ MSR = 0.95 \]
\[ CPU = 17.3 \text{ min} \]

Adaptive mesh interpolation

\[ 0.35^\circ \leq \Delta x_0 \leq 2.82^\circ \]
\[ N = 11,124 \]
\[ CFL_0 = 8.27 \]
\[ \varepsilon = 0.0274 \]
\[ MSR = 0.9 \]
\[ CPU = 2.0 \text{ min} \]
Summary

• Large time stepping: All calculations shown used $\Delta t = T/80$

• Local conservation of scalar is implicit
  • Global conservation depends on mesh quality

• Managing mesh quality over time is crucial
  • $\Rightarrow$ Remeshing is necessary

• Delta-convolution interpolation using Green’s function:
  • Requires good mesh quality
  • Allows differentiation without a stencil (SWE)
Future work

- Next step: Remeshing w/adaptive mesh refinement
- Adaptive smoothing parameter
- Increase order of quadrature method