Wave propagation algorithms on logically Cartesian sphere grids

Donna Calhoun

Boise State University

with Marsha Berger (NYU), Christiane Helzel (RUB),
Randall J. LeVeque (UW)

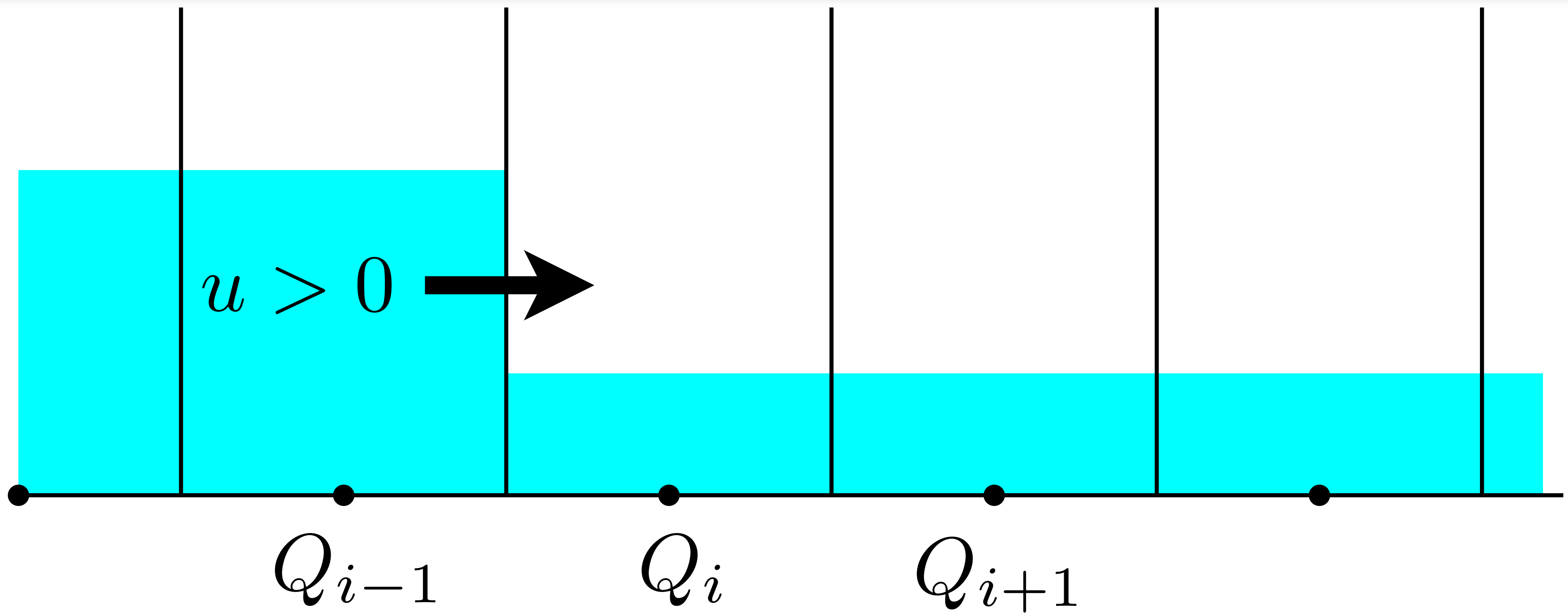
NCAR Transport Workshop
Boulder, CO
March 30-31, 2011

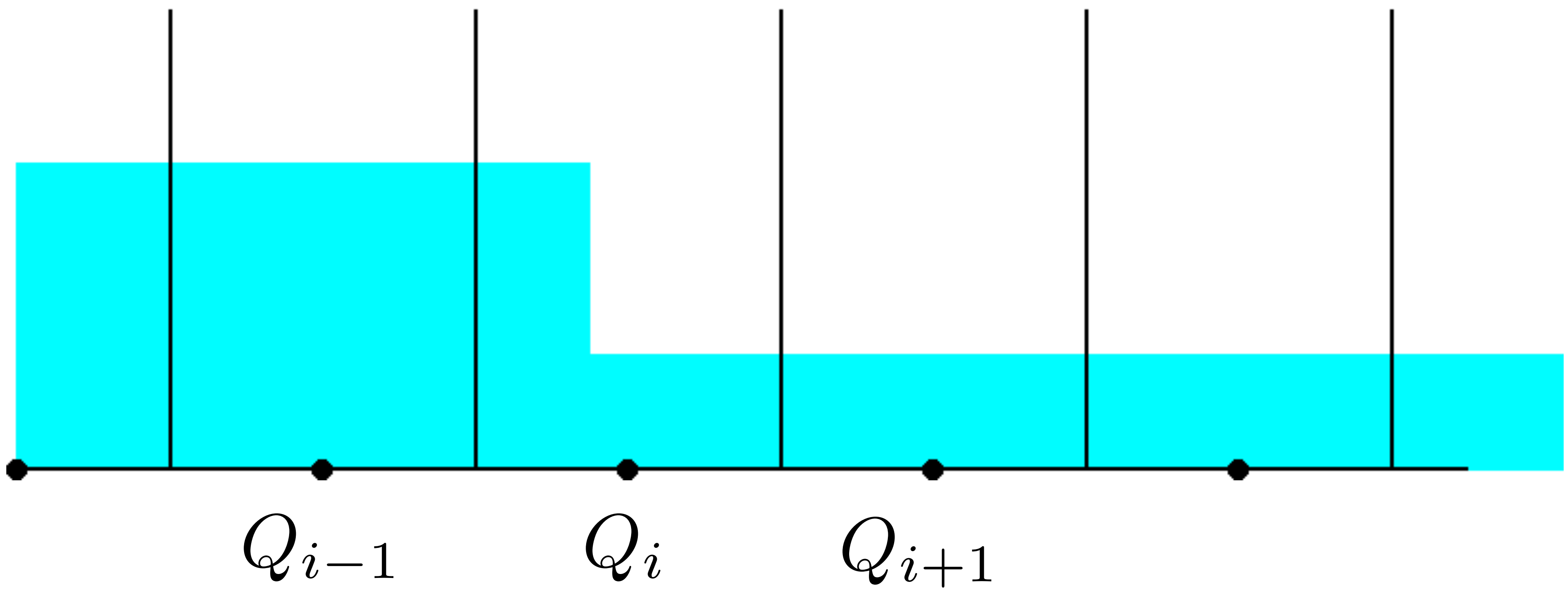
Wave propagation algorithm and Clawpack

Clawpack (R. J. LeVeque) general purpose code based on the wave propagation algorithm for solving hyperbolic problems.

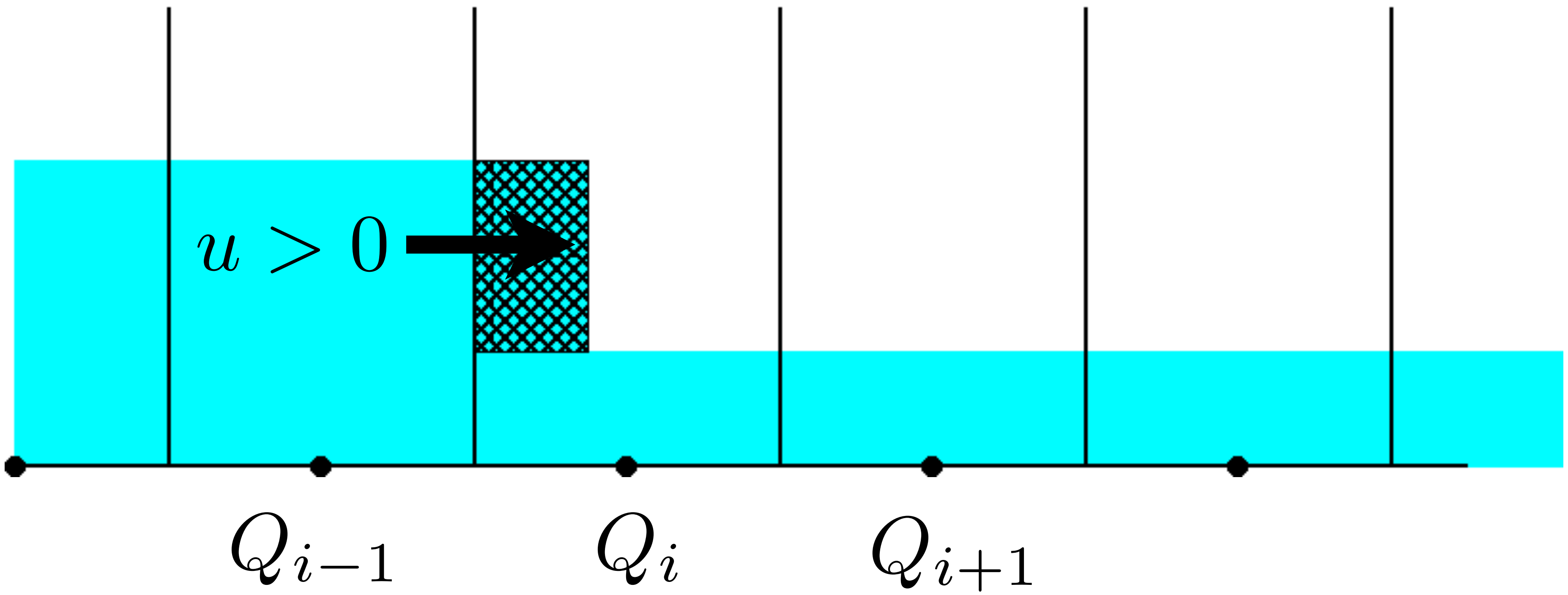
- Finite volume schemes on logically Cartesian grids
- First and second order versions, with or without limiters.
- Split and unsplit options available.
- Second order accuracy is achieved by propagating second order correction "waves" in normal and transverse directions.
- Version with adaptive mesh refinement available (AMRClaw)

Wave propagation algorithms





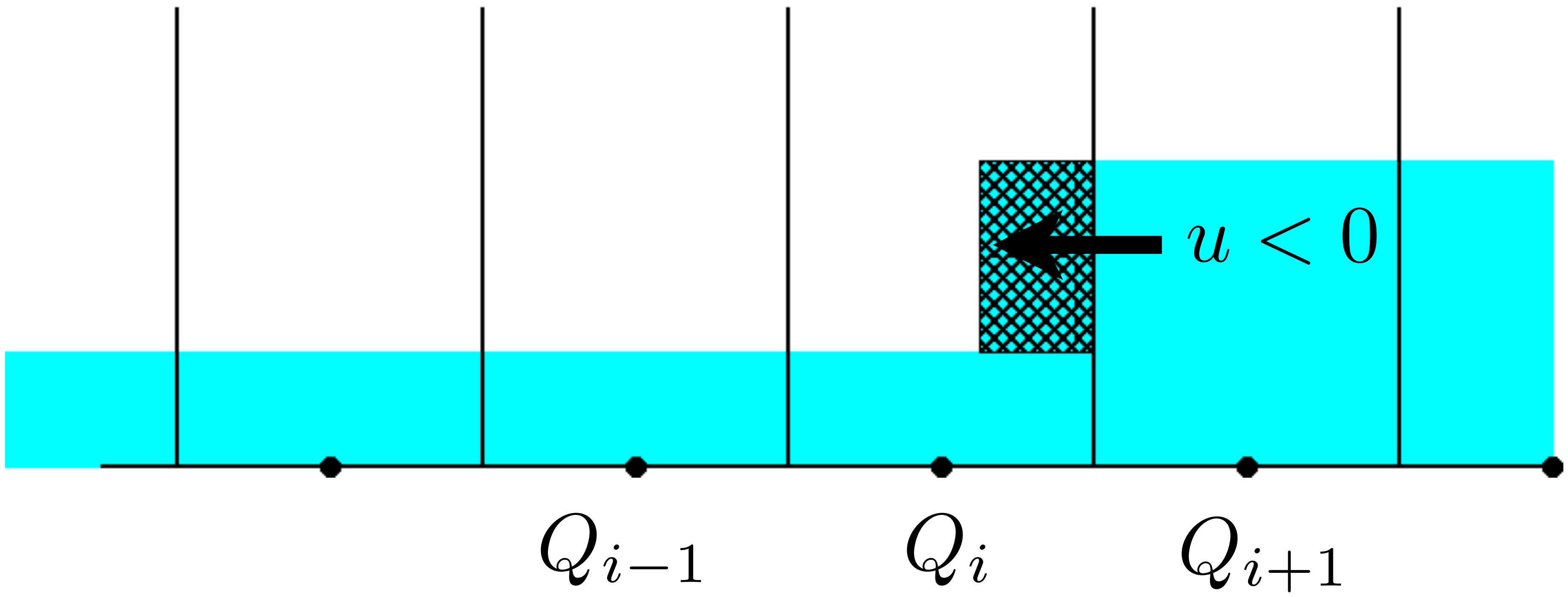
Wave propagation algorithms



In time Δt, cell average increases by shaded area :

$$\Delta x Q_i^{n+1} = \Delta x Q_i^n - u\,\Delta t\left(Q_i^n - Q_{i-1}^n\right)$$

Wave propagation algorithms



In time Δt, cell average increases by shaded area :

$$\Delta x Q_i^{n+1} = \Delta x Q_i^n - u\,\Delta t\left(Q_{i+1}^n - Q_i^n\right)$$

Upwind scheme

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left(u^+(Q_i^n - Q_{i-1}^n) + u^-(Q_{i+1}^n - Q_i^n)\right)$$

where

$$u^+ = \max(u, 0), \qquad u^- = \min(u, 0)$$

We can define waves at each interface as :

Waves : $\mathcal{W}_{i-1/2} \equiv Q_i - Q_{i-1}$

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left(u^+\mathcal{W}_{i-1/2} + u^-\mathcal{W}_{i+1/2}\right)$$

Upwind scheme for systems

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left(\sum_{p=1}^{m}(\lambda^p)^+\mathcal{W}^p_{i-1/2} + \sum_{p=1}^{m}(\lambda^p)^-\mathcal{W}^p_{i+1/2}\right)$$

where the waves are now defined as from an eigenvalue decomposition of the jump in value at each interface

Waves : $$\widetilde{\mathcal{W}}^p_{i-1/2} = \alpha^p r^p$$

Written in terms of *fluctuations* :

$$\begin{aligned}\mathcal{A}^+\Delta Q_{i-1/2} &\equiv \sum_{p=1}^{m}(\lambda^p)^+\mathcal{W}_{i-1/2}\\ \mathcal{A}^-\Delta Q_{i-1/2} &\equiv \sum_{p=1}^{m}(\lambda^p)^-\mathcal{W}_{i-1/2}\end{aligned}$$

First order update

Update in "wave propagation form" (Clawpack, R. J. LeVeque)

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left(\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i+1/2}\right)$$

Unified approach treats equations in both conservation and non-conservative form :

Conservative : $q_t + f(q)_x = 0$

Non-conservative : $q_t + A\, q_x = 0$

Second order correction terms

"High resolution terms" give better accuracy

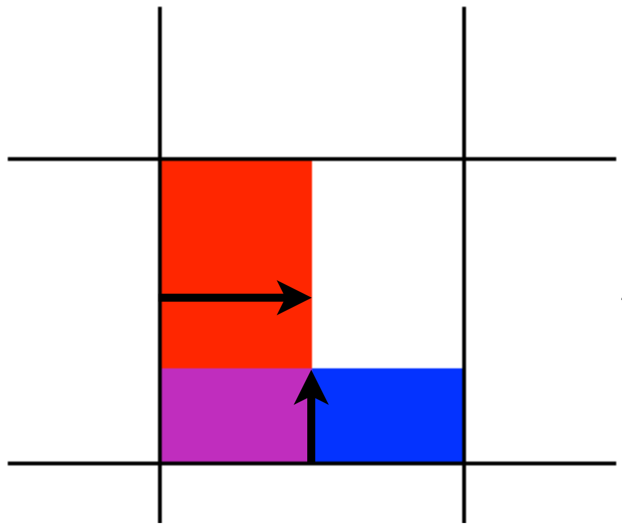
$$
\begin{aligned}
Q_i^{n+1} = Q_i^n \quad &- \quad \frac{\Delta t}{\Delta x}\left(\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i+1/2}\right) \\
&- \quad \frac{\Delta t}{\Delta x}\left(\widetilde{F}_{i+1/2} - \widetilde{F}_{i-1/2}\right)
\end{aligned}
$$

where

$$
\widetilde{F}_{i-1/2} = \frac{1}{2}\sum_{p=1}^{m} |\lambda^p| \left(1 - \frac{\Delta t}{\Delta x}|\lambda^p|\right) \widetilde{\mathcal{W}}^p_{i-1/2}
$$

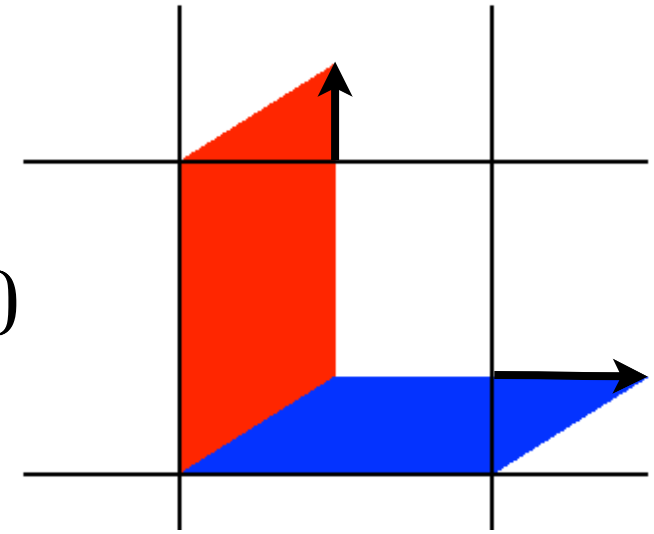
where $\widetilde{\mathcal{W}}^p_{i-1/2}$ are *limited* waves.

Wave-propagation method in two dimensions



Normal waves

$$u > 0, \quad v > 0$$



Transverse waves

Wave propagation algorithm

$$
\begin{aligned}
Q_{ij}^{n+1} = Q_{ij}^{n} &- \frac{u\Delta t}{\Delta x}\left(Q_{ij}^{n} - Q_{i-1,j}^{n}\right) - \frac{v\Delta t}{\Delta y}\left(Q_{ij}^{n} - Q_{i,j-1}^{n}\right) \\
&+ \frac{1}{2}(\Delta t)^2 \left\{ \frac{u}{\Delta x}\left[\frac{v}{\Delta y}(Q_{ij}^{n} - Q_{i,j-1}^{n}) - \frac{v}{\Delta y}(Q_{i-1,j}^{n} - Q_{i-1,j-1})\right]\right. \\
&\left. + \frac{v}{\Delta y}\left[\frac{u}{\Delta x}(Q_{ij}^{n} - Q_{i-1,j}^{n}) - \frac{u}{\Delta x}(Q_{i,j-1}^{n} - Q_{i-1,j-1})\right]\right\}
\end{aligned}
$$

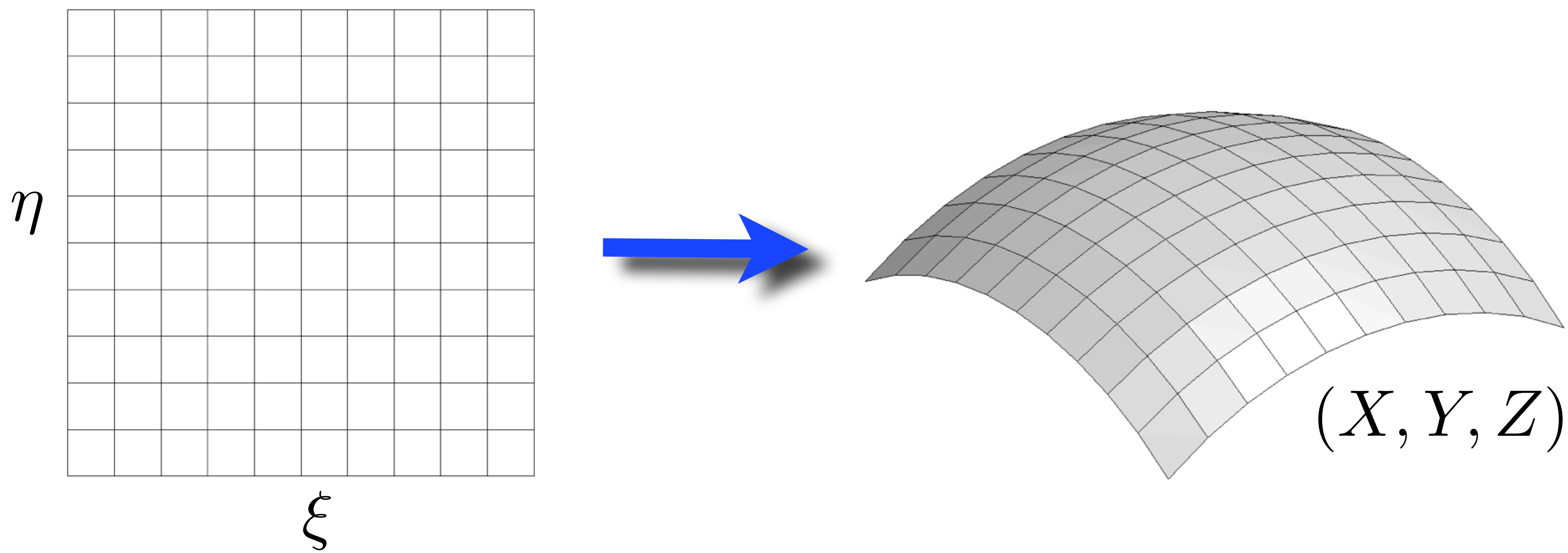
Forms of equations handled by Clawpack

Conservative form	$q_t + f(q)_x = 0$
Quasi-linear form	$q_t + f'(q)\, q_x = 0$
Transport equation	$q_t + u(x) q_x = 0$
Spatially varying flux functions	$q_t + f(q, x)_x = 0$
Source terms	$q_t + f(q)_x = \Psi(q, \ldots)$

Finite volume schemes on curvilinear grids

Assume an underlying logically Cartesian computational grid and a smooth mapping given by

$$\mathbf{T}(\xi,\eta) = (X(\xi,\eta), Y(\xi,\eta), Z(\xi,\eta))^T$$



Computational space

Physical space

Computational domain and metric terms

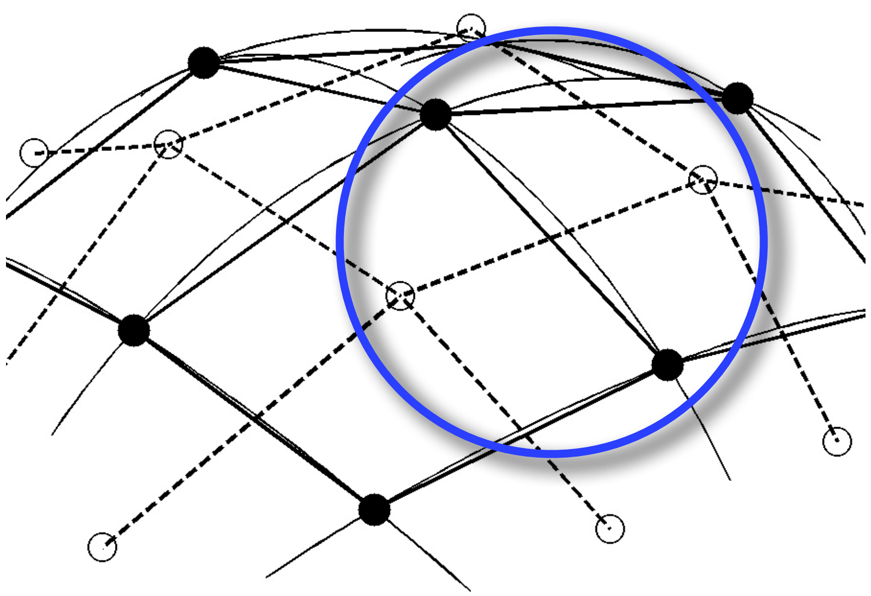
Solve

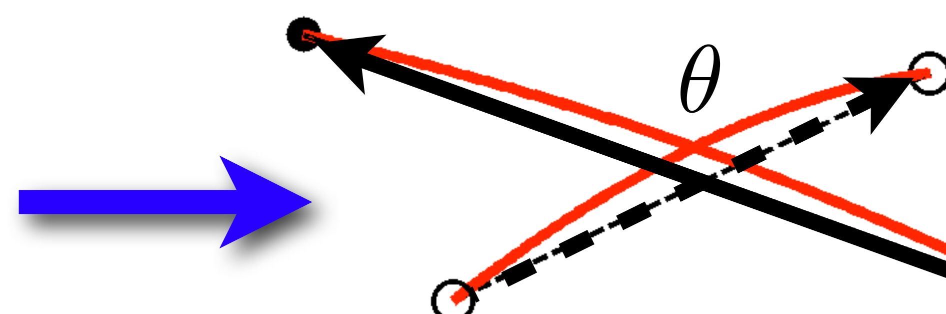
$$\phi_t + \mathbf{u} \cdot \nabla \phi = 0$$

on a computational domain with a velocity field and cell areas modified by metric terms for the surface grid.

- ☑ Metric terms (cell areas, edge lengths and normals) are computed discretely,
- ☑ Only knowledge of location of mesh cell centers and nodes is required.
- ☑ Handles general smooth or piecewise smooth logically Cartesian surface meshes.

Discrete metric terms





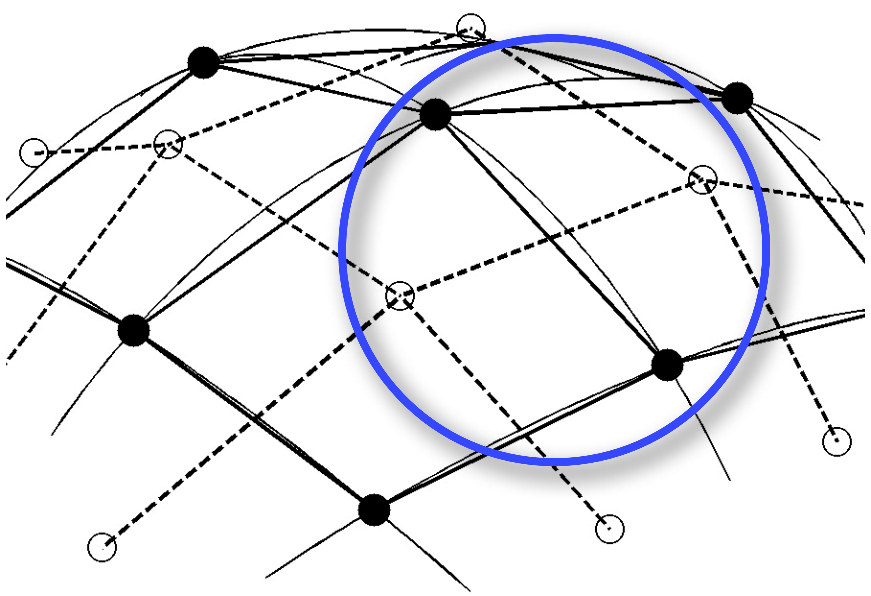
$$a_{11} = \mathbf{T}_\xi \cdot \mathbf{T}_\xi \approx \mathbf{t} \cdot \mathbf{t} = \|\mathbf{t}\|^2$$

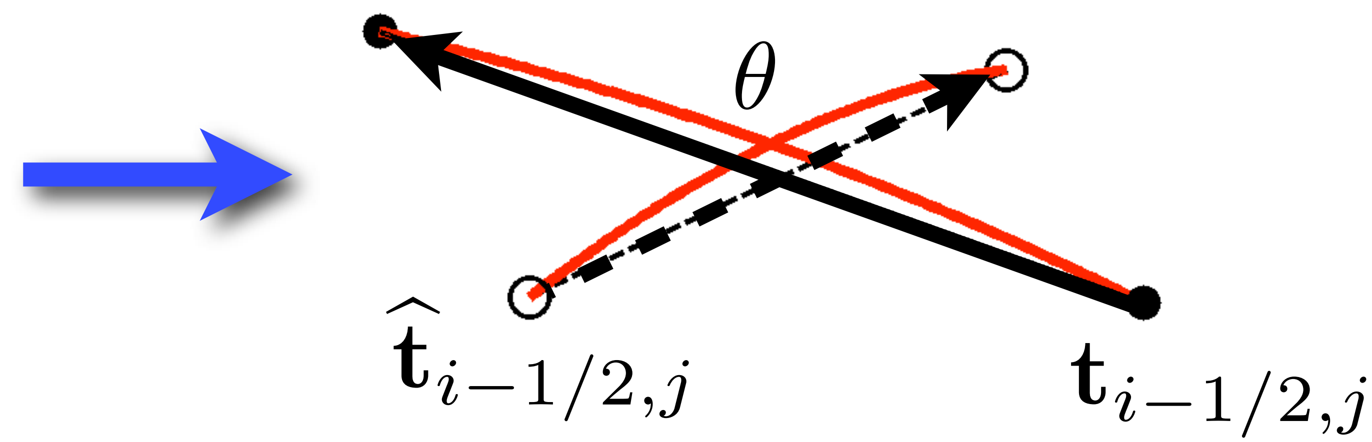
$$a_{12} = a_{21} = \mathbf{T}_\xi \cdot \mathbf{T}_\eta \approx \mathbf{t} \cdot \widehat{\mathbf{t}} = \|\mathbf{t}\|\|\widehat{\mathbf{t}}\| \cos(\theta)$$

$$a_{22} = \mathbf{T}_\eta \cdot \mathbf{T}_\eta \approx \widehat{\mathbf{t}} \cdot \widehat{\mathbf{t}} = \|\widehat{\mathbf{t}}\|^2$$

$$\sqrt{a} = \| \mathbf{T}_\xi \times \mathbf{T}_\eta \| \approx \| \mathbf{t} \times \widehat{\mathbf{t}} \| = \|\mathbf{t}\|\|\widehat{\mathbf{t}}\| \sin(\theta)$$

Normals and lengths





Lengths

$$\ell_{i-1/2,j} = \int_{\eta_{j-1/2}}^{\eta_{j+1/2}} \| \mathbf{T}_\eta \| \; d\eta \approx \| \mathbf{t}_{i-1/2,j} \|$$

Edge normals tangent to the surface

$$\mathbf{n}_{i-1/2,j} \approx \frac{\mathbf{t}^{(1)}}{\| \mathbf{t}^{(1)} \|} \approx \frac{1}{\| \mathbf{t} \|} \left(\frac{\| \mathbf{t} \|}{\| \widehat{\mathbf{t}} \|} \csc(\theta)\, \widehat{\mathbf{t}} - \cot(\theta) \mathbf{t} \right)$$

Discrete mesh cell areas

$$A_{ij} = \int_{C_{ij}} \| \mathbf{T}_\xi \times \mathbf{T}_\eta \| \; d\xi \, d\eta$$

Approximate mesh cell surface as a ruled surface :

$$\mathbf{S}(u, v) = \mathbf{a}_{00} + \mathbf{a}_{01} u + \mathbf{a}_{10} v + \mathbf{a}_{11} uv, \qquad 0 \le u, v \le 1$$

where the coefficients $\mathbf{a}_{\ell m} \in \mathcal{R}^3$ are computed from four corners of the mesh cell. Then

$$\begin{aligned} A_{ij} &\approx \int_{C_{ij}} \|\mathbf{S}_u \times \mathbf{S}_v\| \; du dv \\ &\approx \| \, (\mathbf{a}_{01} + \frac{1}{2}\mathbf{a}_{11}) \times (\mathbf{a}_{10} + \frac{1}{2}\mathbf{a}_{11}) \, \| \end{aligned}$$

Curvature and surface normals

$$\nabla \cdot \nabla \mathbf{T} = 2\mathbf{H}$$

where H is the mean curvature normal. Using the divergence theorem, we get

$$\int_C \nabla \cdot \nabla \mathbf{T}\, dS = \int_{\partial C} \nabla \mathbf{T} \cdot \mathbf{n}\, dL = \int_{\partial C} \mathbf{n}\, dL$$

so that the mean curvature vector can be computed as

$$\mathbf{H}_{ij} \approx \frac{1}{2A_{ij}} \sum_{k=1}^{4} \ell_{ij,k}\, \mathbf{n}_{ij,k}$$

Curvature and surface normals

A surface curvature can then be approximated as

$$\widehat{\kappa}_{ij} \approx \| \mathbf{H}_{ij} \|$$

and the surface normal can be approximated as

$$\mathbf{n}_{ij}^S \approx \frac{\mathbf{H}_{ij}}{\widehat{\kappa}_{ij}}$$

This is an intrinsic definition of the surface normal and does not require any knowledge of the shape of the surface, e.g. we do not need to know the surface is a sphere.

FV update formula for curvilinear grids

$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{\Delta t}{\kappa_{ij}\Delta\xi}\left(u^+ \Delta Q_{i-1/2,j} + u^- \Delta Q_{i+1,j}\right) -$$

$$\frac{\Delta t}{\kappa_{ij}\Delta\eta}\left(v^+ \Delta Q_{i,j-1/2} + v^- \Delta Q_{i,j+1}\right) -$$

$$\frac{\Delta t}{\kappa_{ij}\Delta\xi}\left(\widetilde{F}_{i+1/2,j} - \widetilde{F}_{i-1/2,j}\right) - \frac{\Delta t}{\kappa_{ij}\Delta\eta}\left(\widetilde{G}_{i,j+1/2} - \widetilde{G}\right.$$

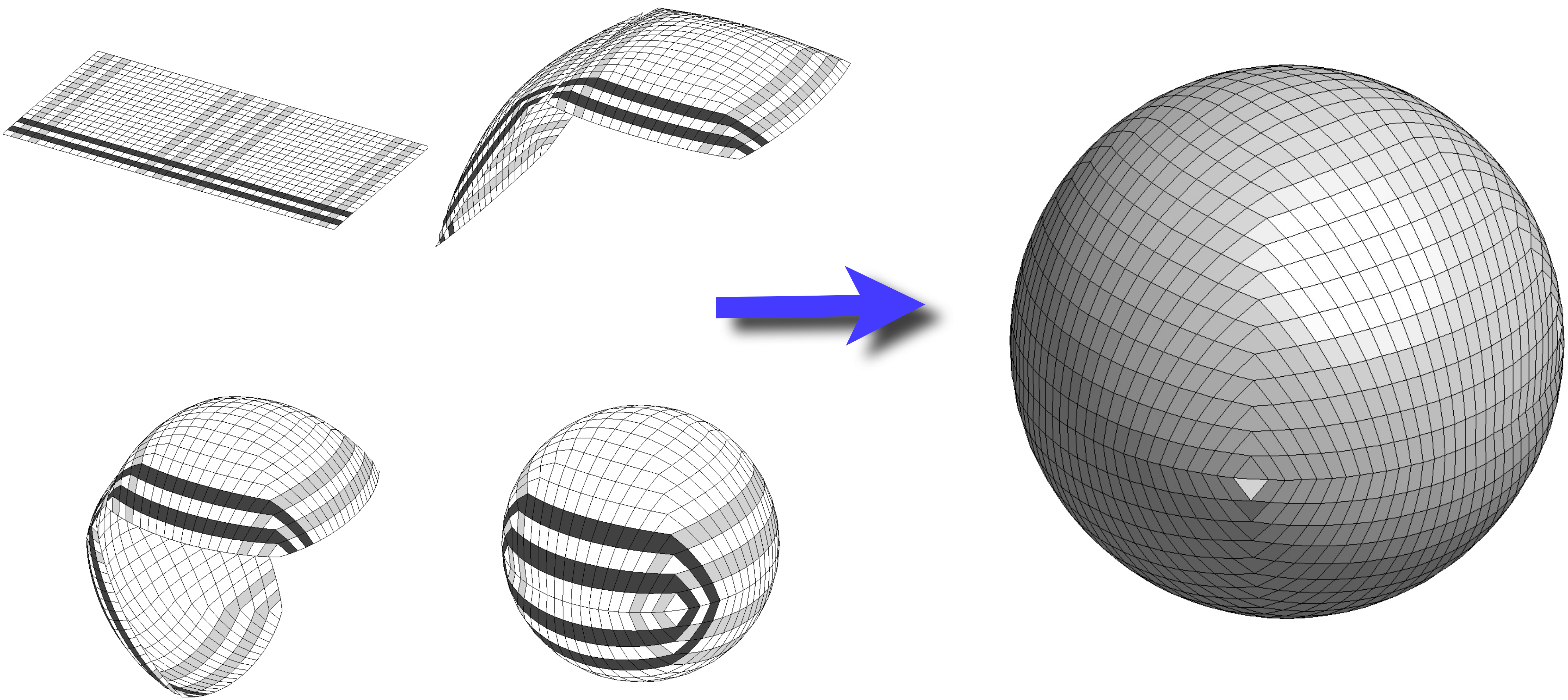
where

$$u_{i-1/2,j} = \frac{\ell_{i-1/2,j}}{\Delta\eta}\widetilde{u}_{i-1/2,j}$$

$$v_{i,j-1/2} = \frac{\ell_{i,j-1/2}}{\Delta\xi}\widetilde{v}_{i,j-1/2}$$

$$\kappa_{ij} = \frac{A_{ij}}{\Delta\xi\Delta\eta}$$

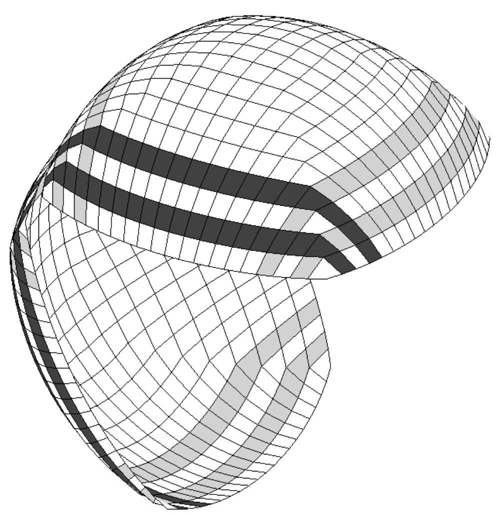
Single logically Cartesian sphere grid

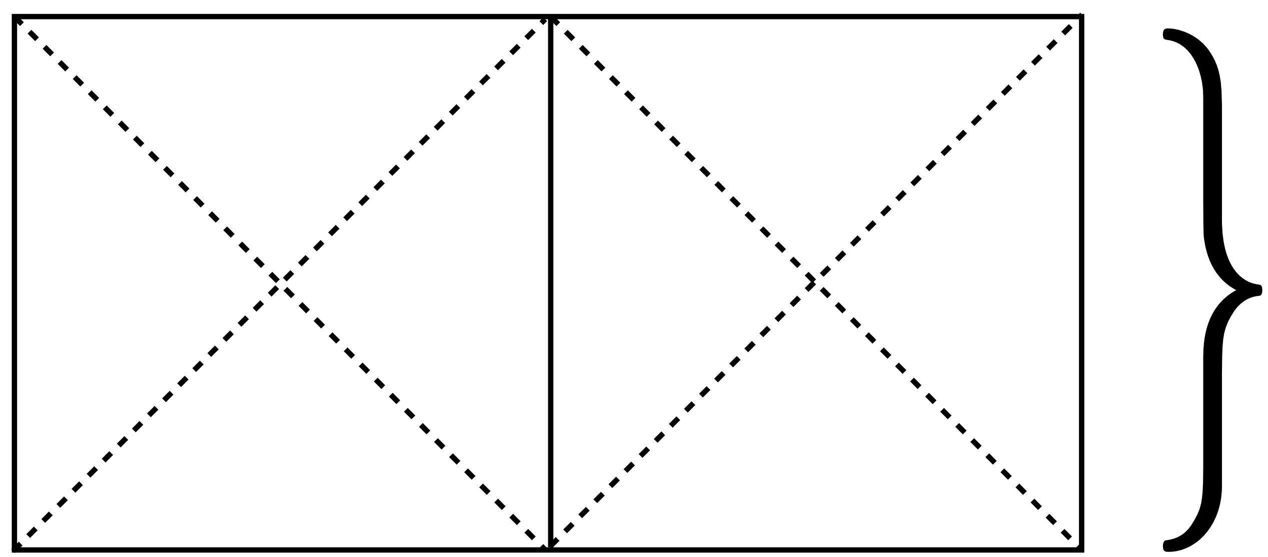


D. Calhoun, C. Helzel, and R. LeVeque, SIAM Review, 50 (2008)

Sphere grid with two patches

- Our sphere grid is like the cubed-sphere grid, but with two patches
- We will refer to the grid resolution by the number of grid cells on a patch edge, which is approximately 90 degrees.

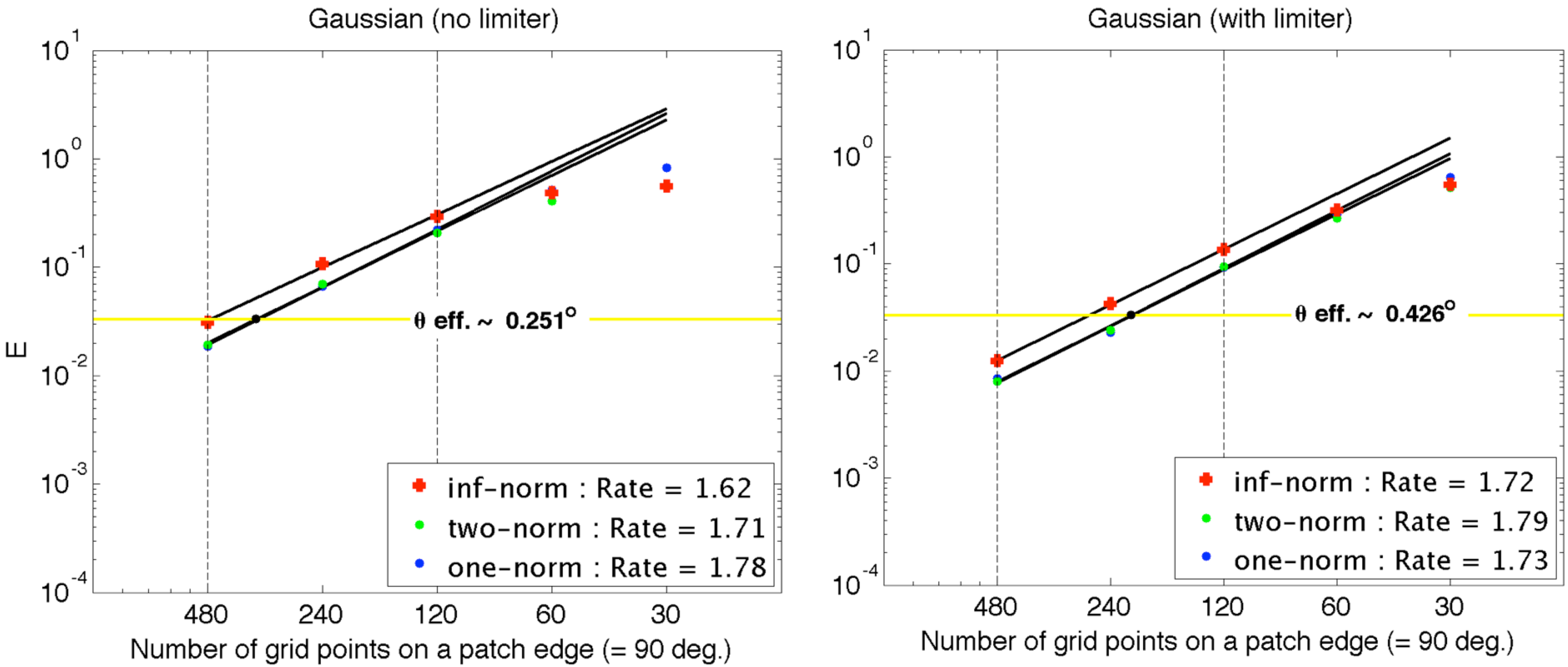




90 degrees = N mesh cells

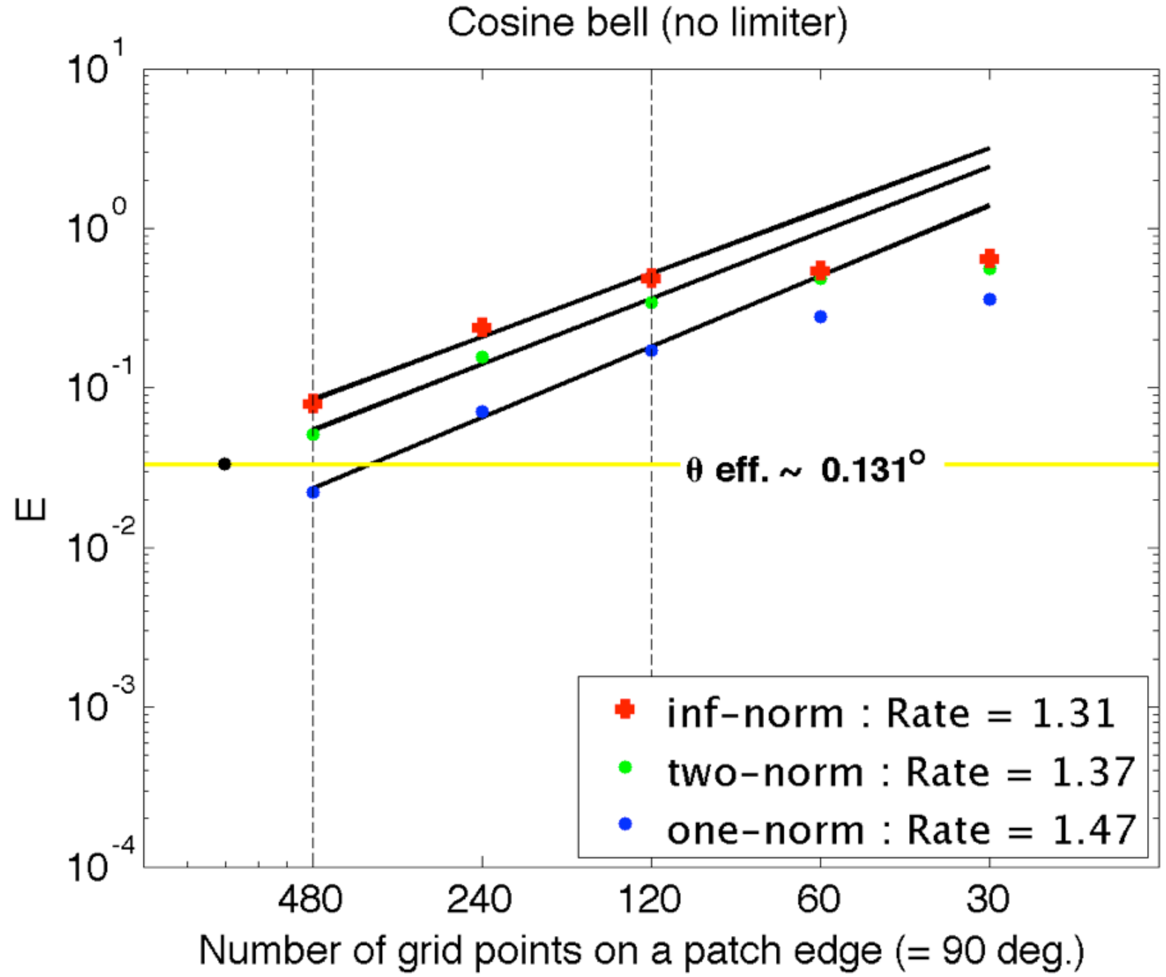
Dashed and solid lines are discontinuities in the mapping

Gaussian Hills



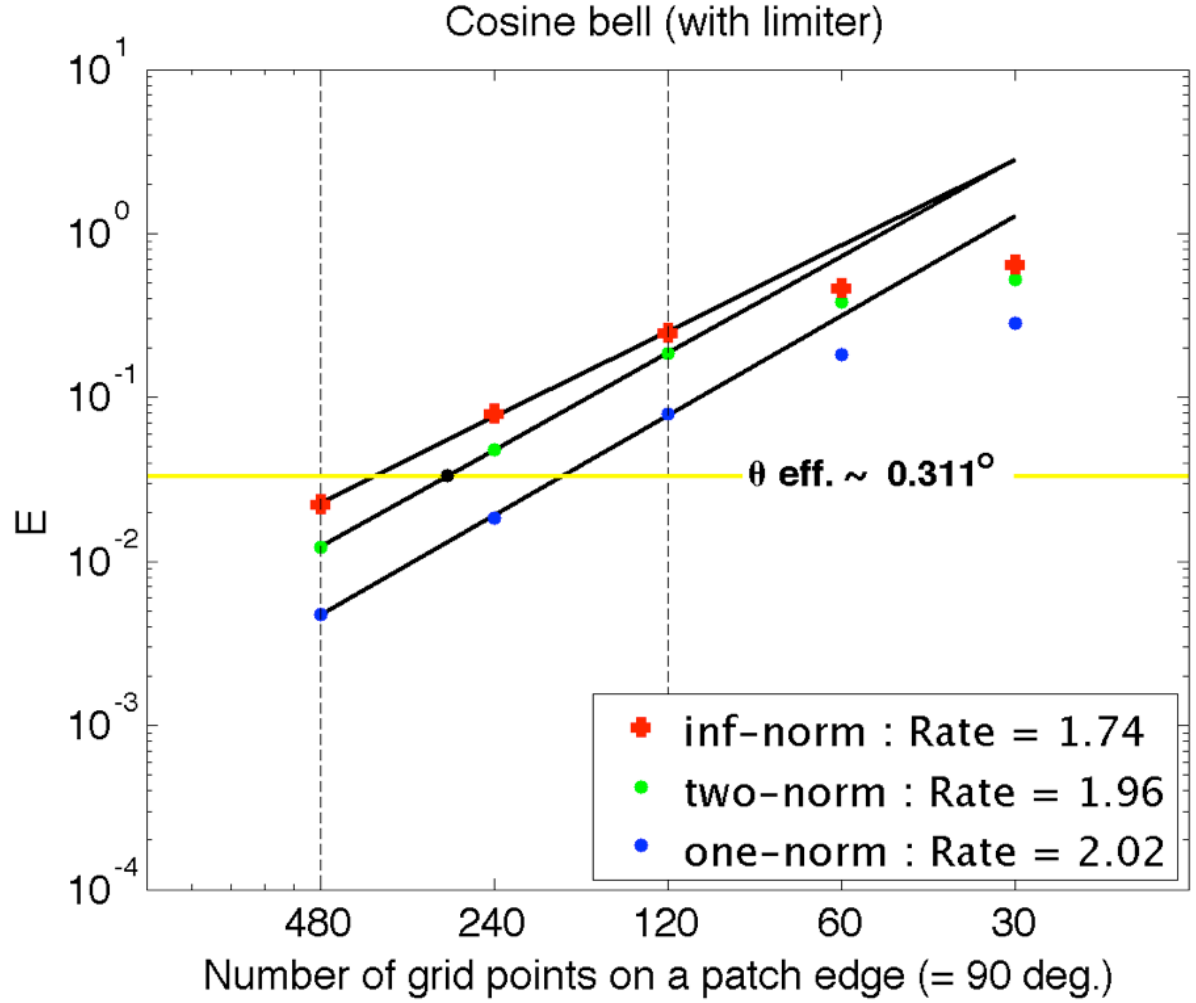
Cosine bell (no limiter)

N	1-norm	2-norm	inf-norm	phi_min	phi_max
30	3.57E-01	5.55E-01	6.47E-01	-5.84E-01	-1.46E-01
60	2.81E-01	4.83E-01	5.42E-01	-3.87E-01	-2.16E-01
120	1.72E-01	3.43E-01	4.83E-01	-1.77E-01	-2.63E-01
240	7.13E-02	1.56E-01	2.38E-01	-1.55E-02	-1.56E-01
480	2.24E-02	5.13E-02	7.89E-02	5.37E-03	-7.02E-02

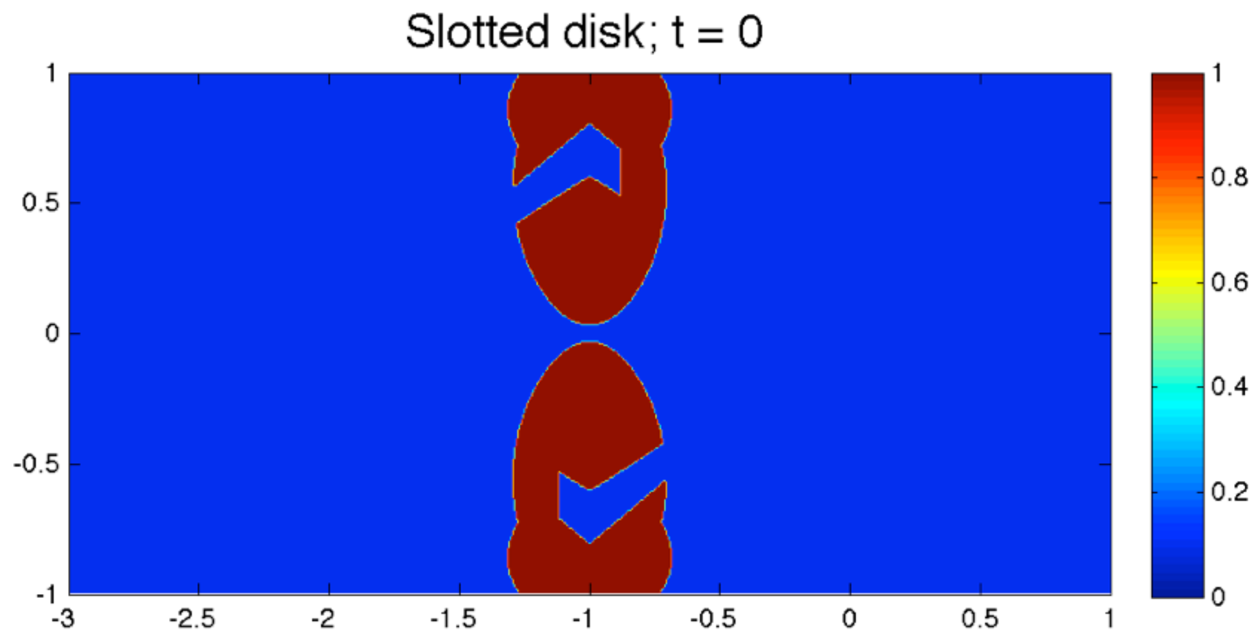


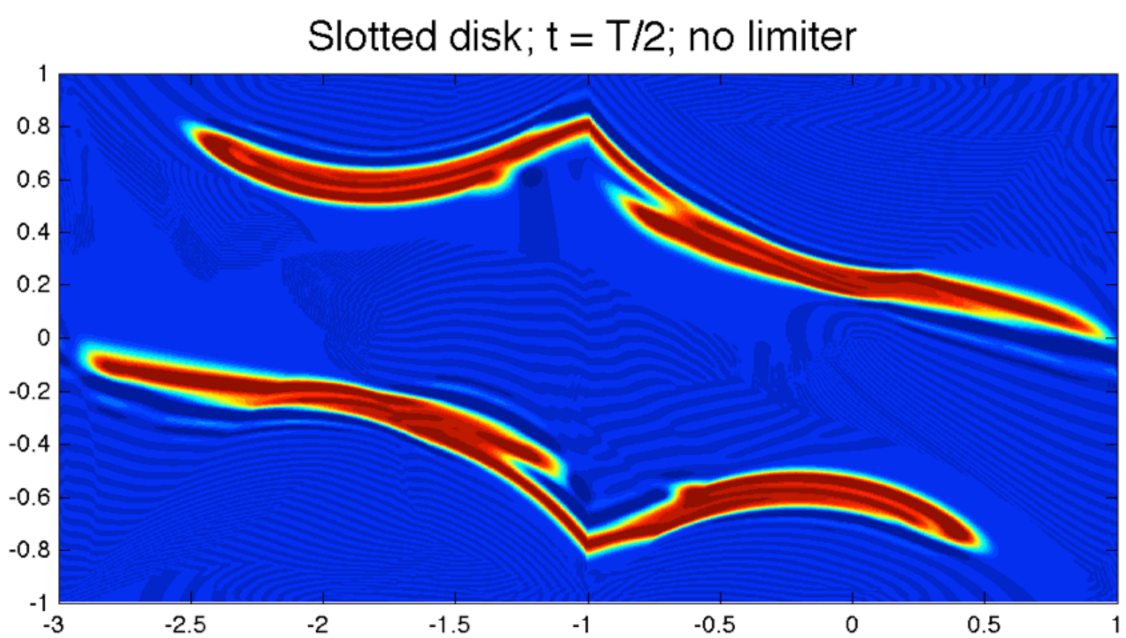
Cosine Bell (with limiter)

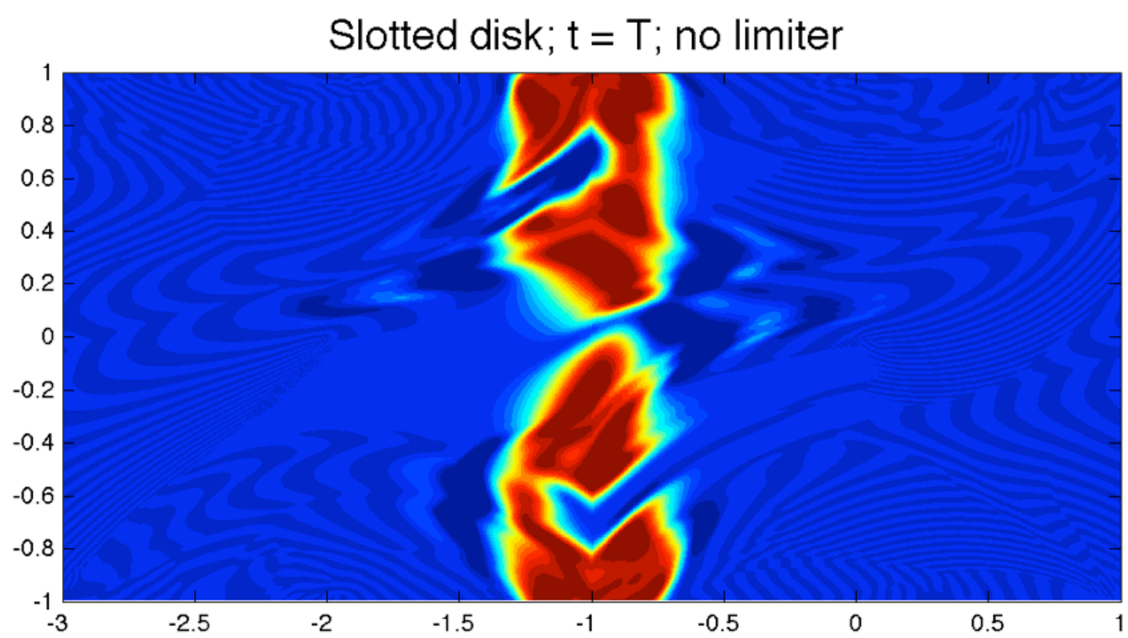
N	l-norm	2-norm	inf-norm	phi_min	phi_max
30	2.84E-01	5.25E-01	6.45E-01	-7.05E-01	-2.85E-04
60	1.84E-01	3.83E-01	4.63E-01	-4.92E-01	-6.92E-04
120	7.88E-02	1.85E-01	2.47E-01	-2.27E-01	-1.62E-03
240	1.83E-02	4.80E-02	7.96E-02	-6.83E-02	-1.42E-03
480	4.79E-03	1.22E-02	2.23E-02	-2.02E-02	-1.07E-03



Slotted cylinders (no limiters)

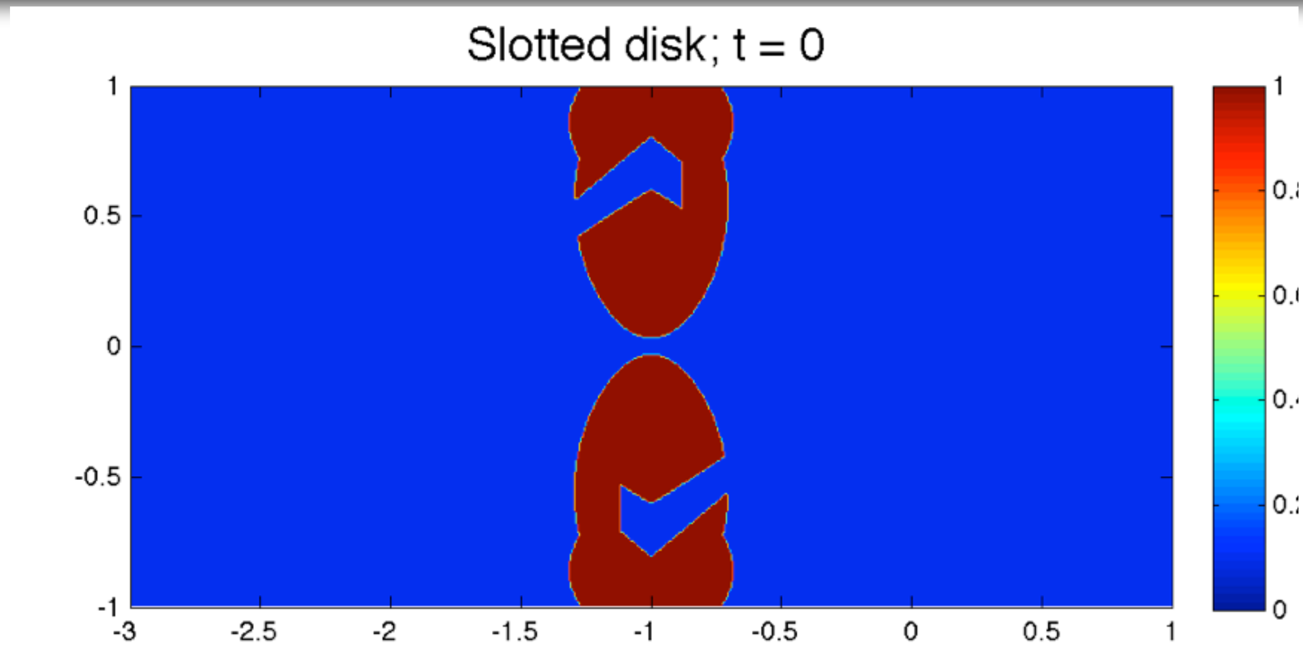


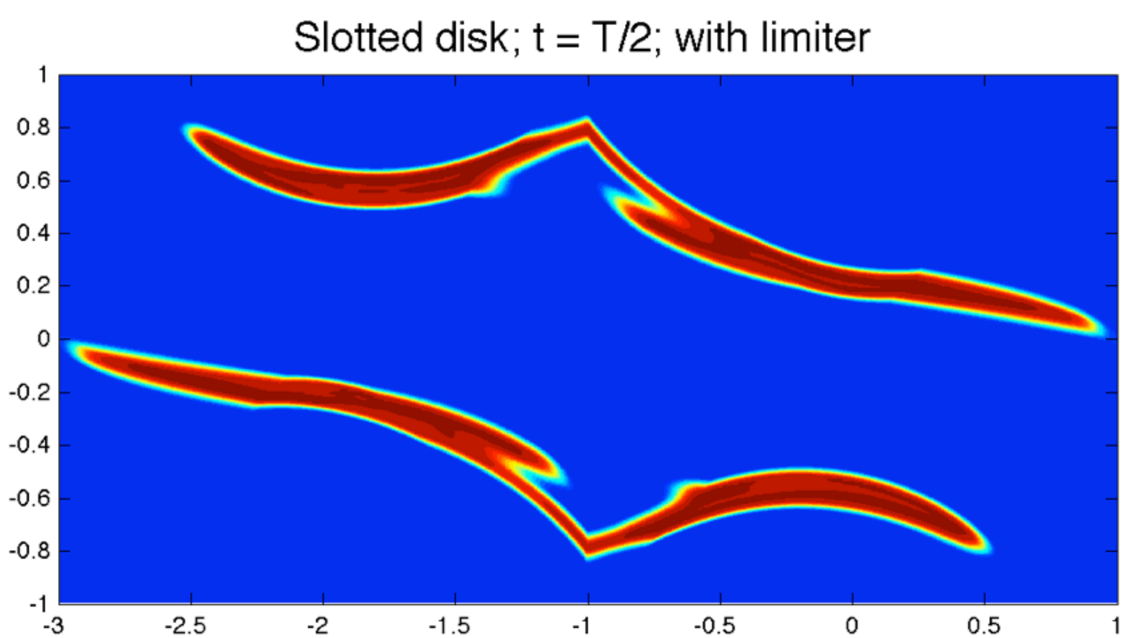


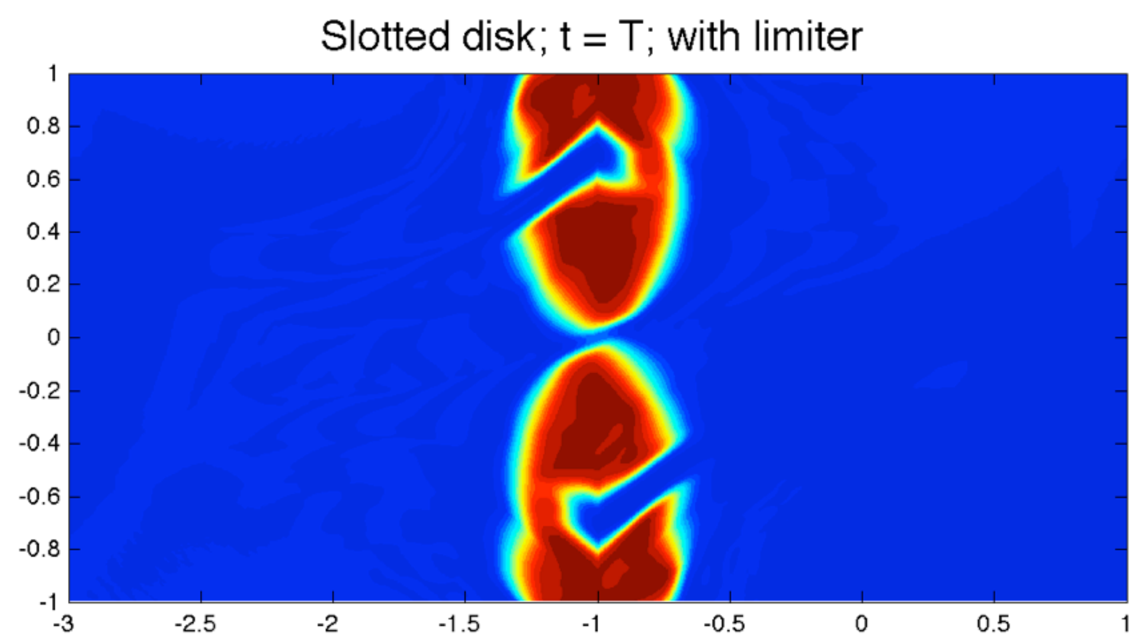


Mesh cells on patch edge : 300 $(\theta = 0.3^o)$

Slotted cylinders (with limiters)

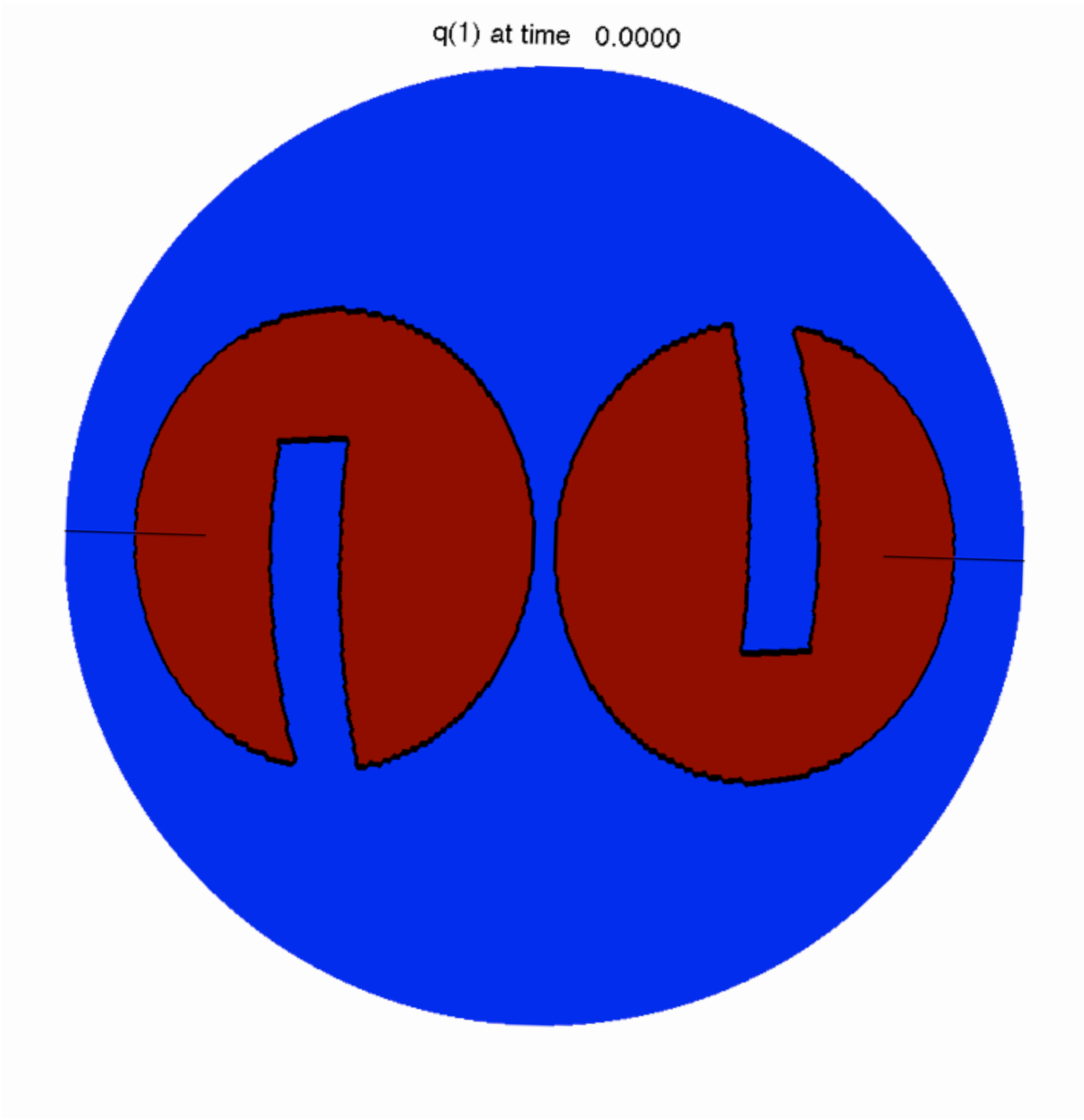


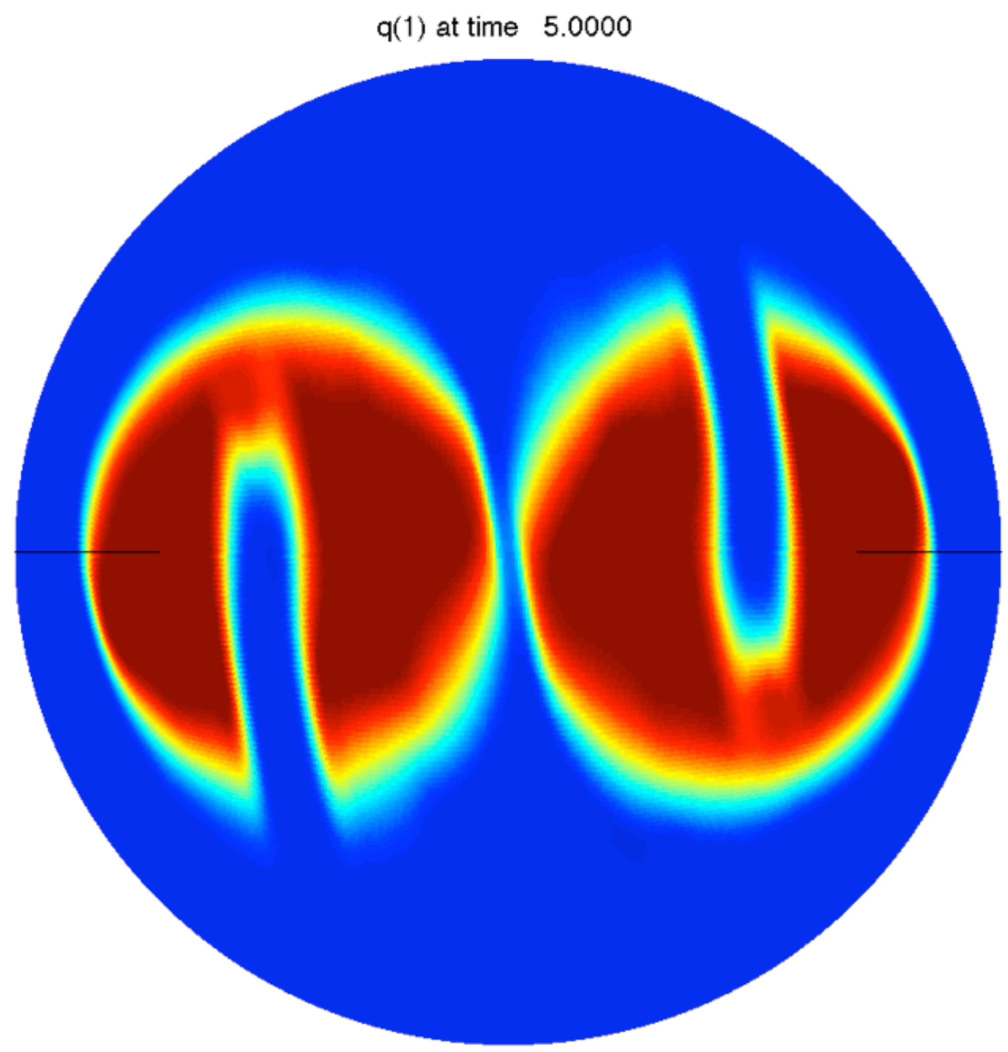




N	1-norm	2-norm	inf-norm
300	0.11	0.24	0.84

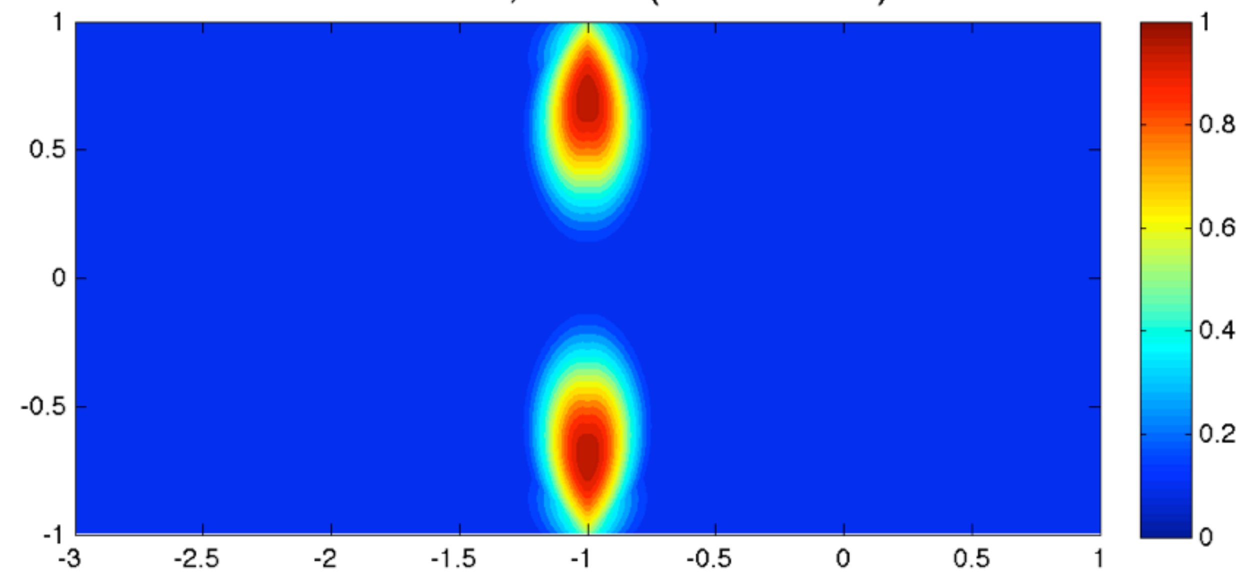
Slotted cylinders (with limiters)



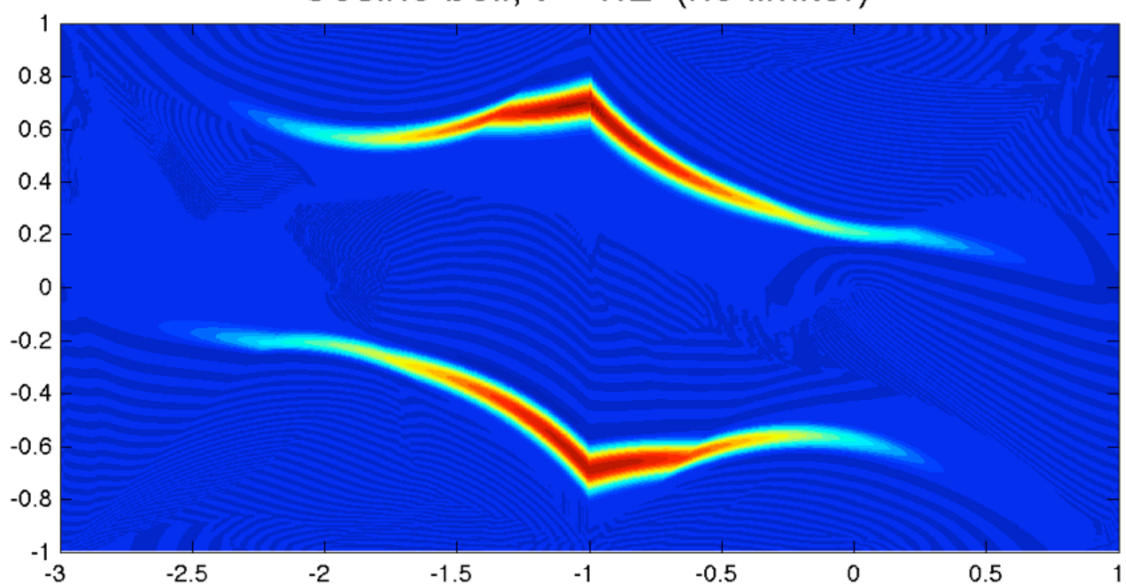


Cosine Bell (no limiter)

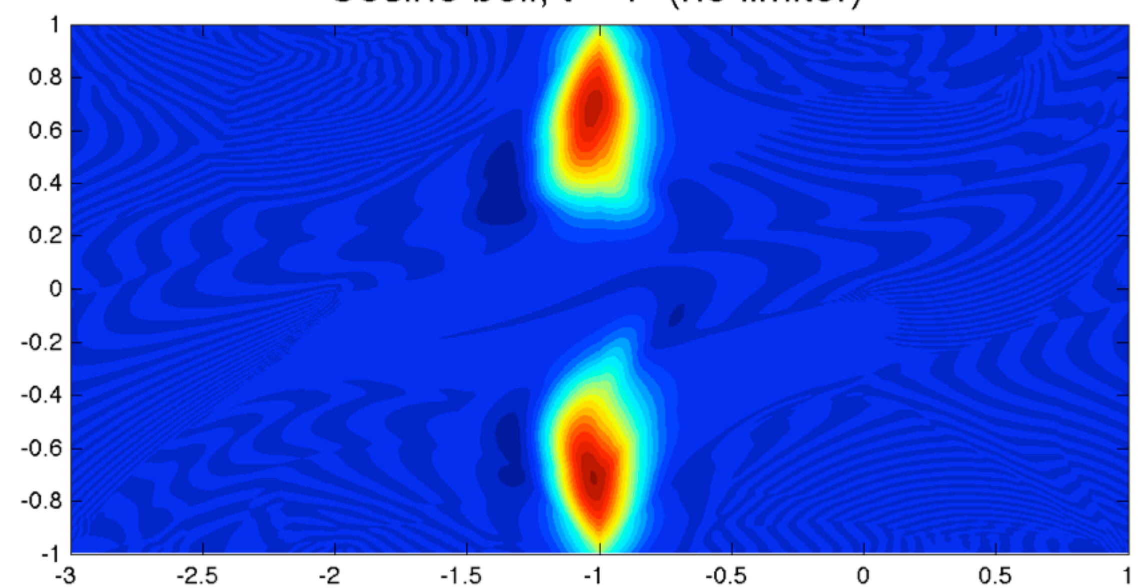
Cosine bell; t = 0 (with limiter)



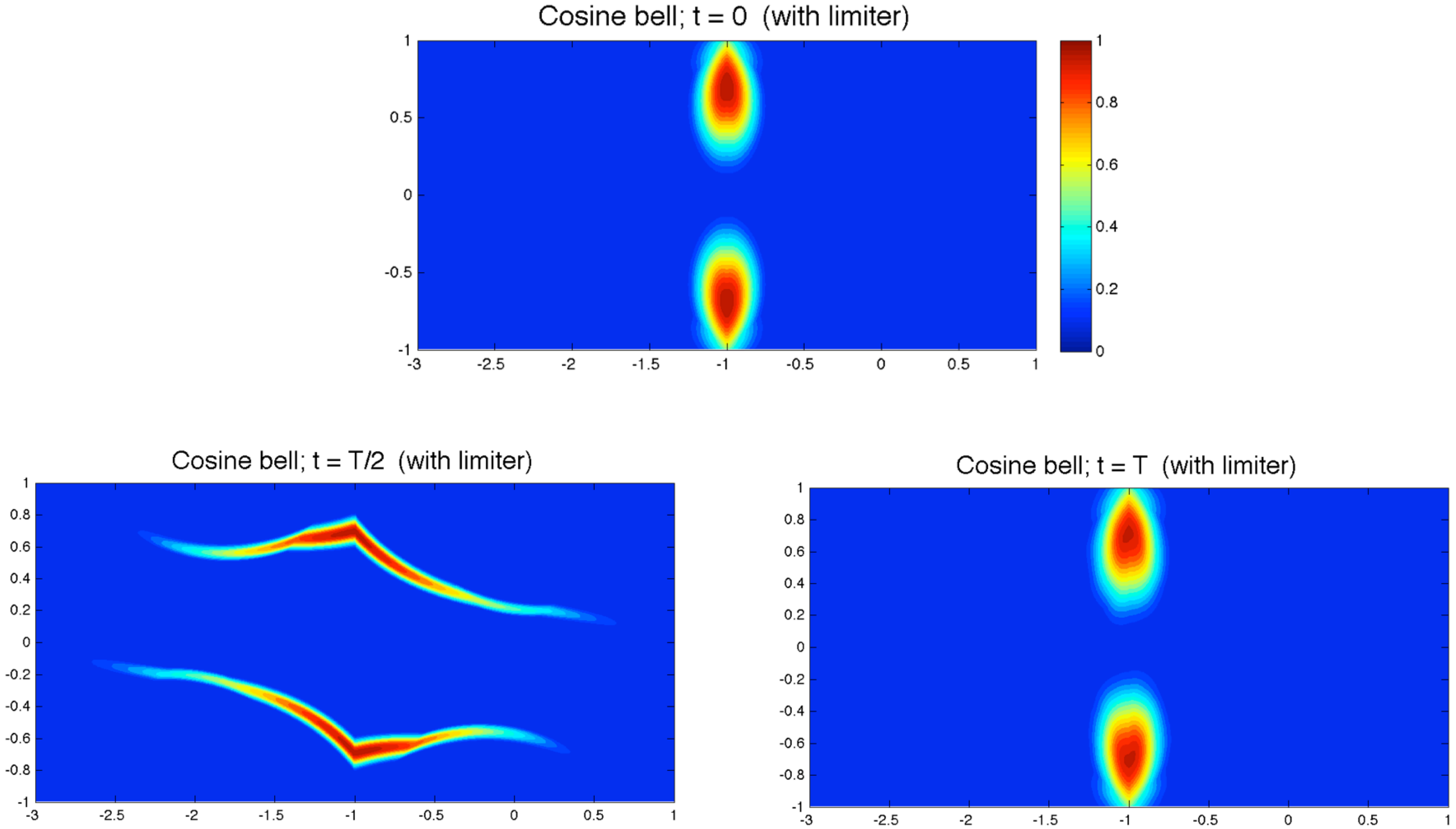
Cosine bell; t = T/2 (no limiter)



Cosine bell; t = T (no limiter)

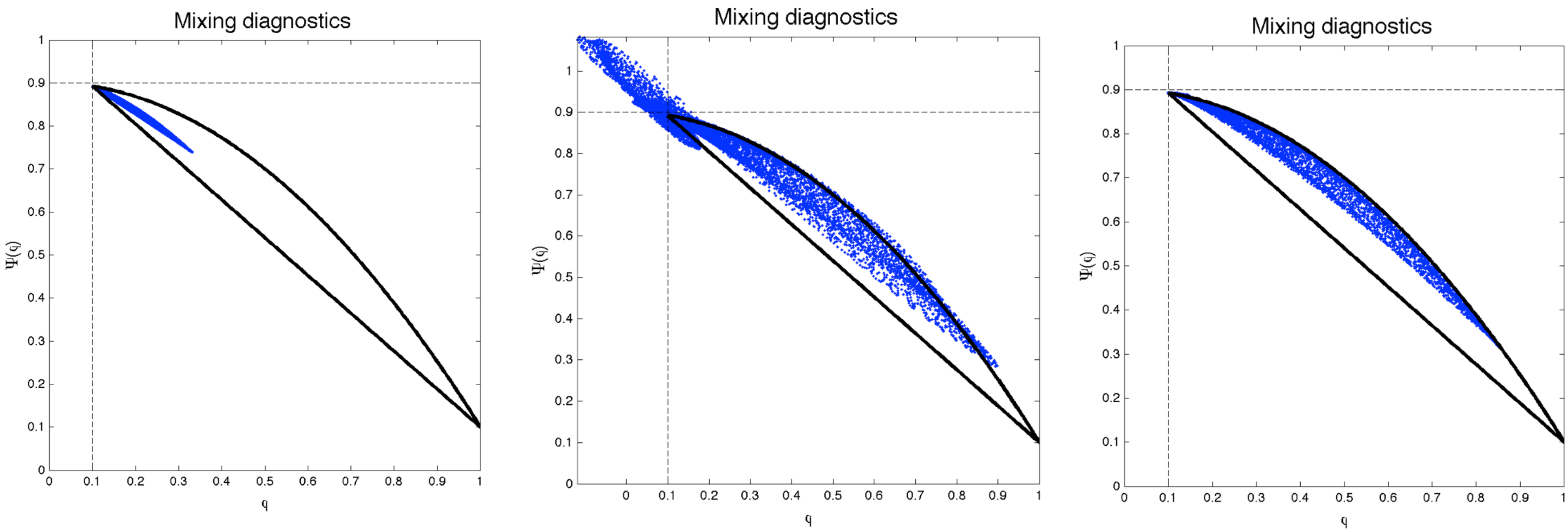


Cosine Bells (with limiters)



Mesh cells on patch edge : 300 $(\theta = 0.3^o)$

Mixing diagnostics



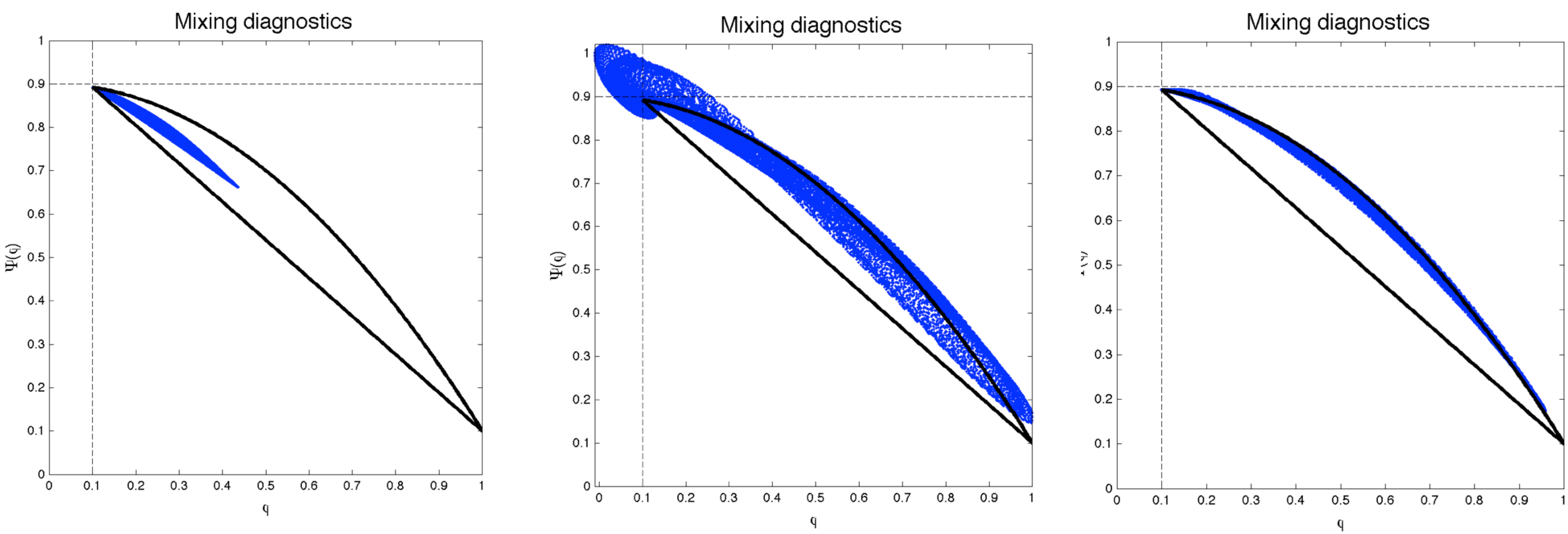
First order

Second order (no limiter)

Second order (with limiter)

150 mesh cells on a patch edge (45,000 cells total)

Mixing diagnostics



First order

Second order (no limiter)

Second order (with limiter)

300 mesh cells on a patch edge (180,000 cells total)

Diagnostics

	First order	No limiters	With limiters
Real mixing	1.24E-02	4.21E-03	2.30E-03
Range preserving unmixing	2.69E-11	1.13E-04	1.22E-05
Overshooting	0.00	5.49E-03	4.05E-05

N = 150 (45,000 mesh cells)

	First order	No limiters	With limiters
Real mixing	1.03E-02	1.99E-03	6.15E-04
Range preserving unmixing	3.51E-10	4.21E-04	1.57E-04
Overshooting	0.00	2.04E-03	3.98E-05

N = 300 (180,000 mesh cells)

Computational efficiency

- Large percentage (over 65%) of the time was spent in evaluating the wind fields, especially since we needed to convert values to polar coordinates. *This portion of the code could easily be optimized.*

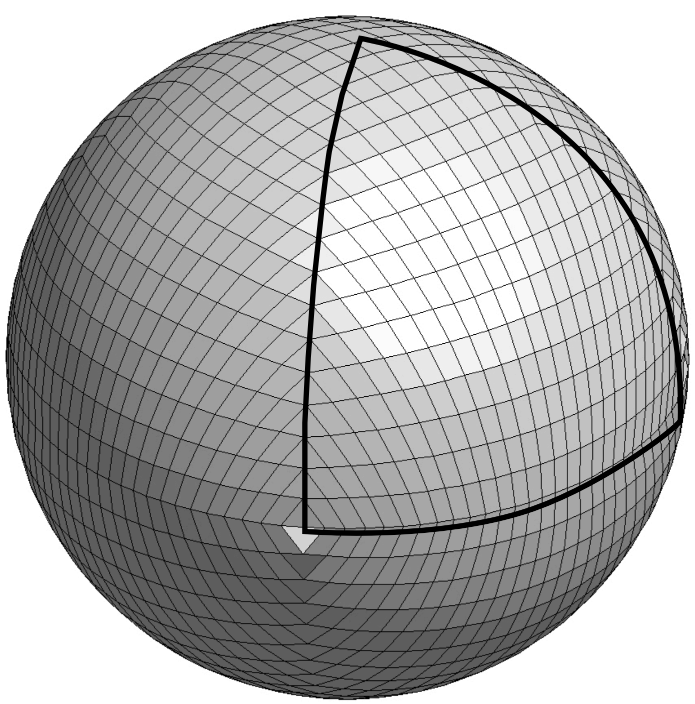
- Less significant portion of the time (< 20%) was used in sweeping over the grid, solving Riemann problems at each horizontal and vertical face.

- Less than 2% of the time was actually spent in the normal and transverse Riemann problems.

Future issues

Accuracy relies on *grid smoothness*

- Seams along the diagonals are handled by modifying the transverse solve so that velocities are all taken on the same side of the discontinuity
- Seams at coordinate lines are more challenging to handle because they affect the normal propagation.



Future Issues

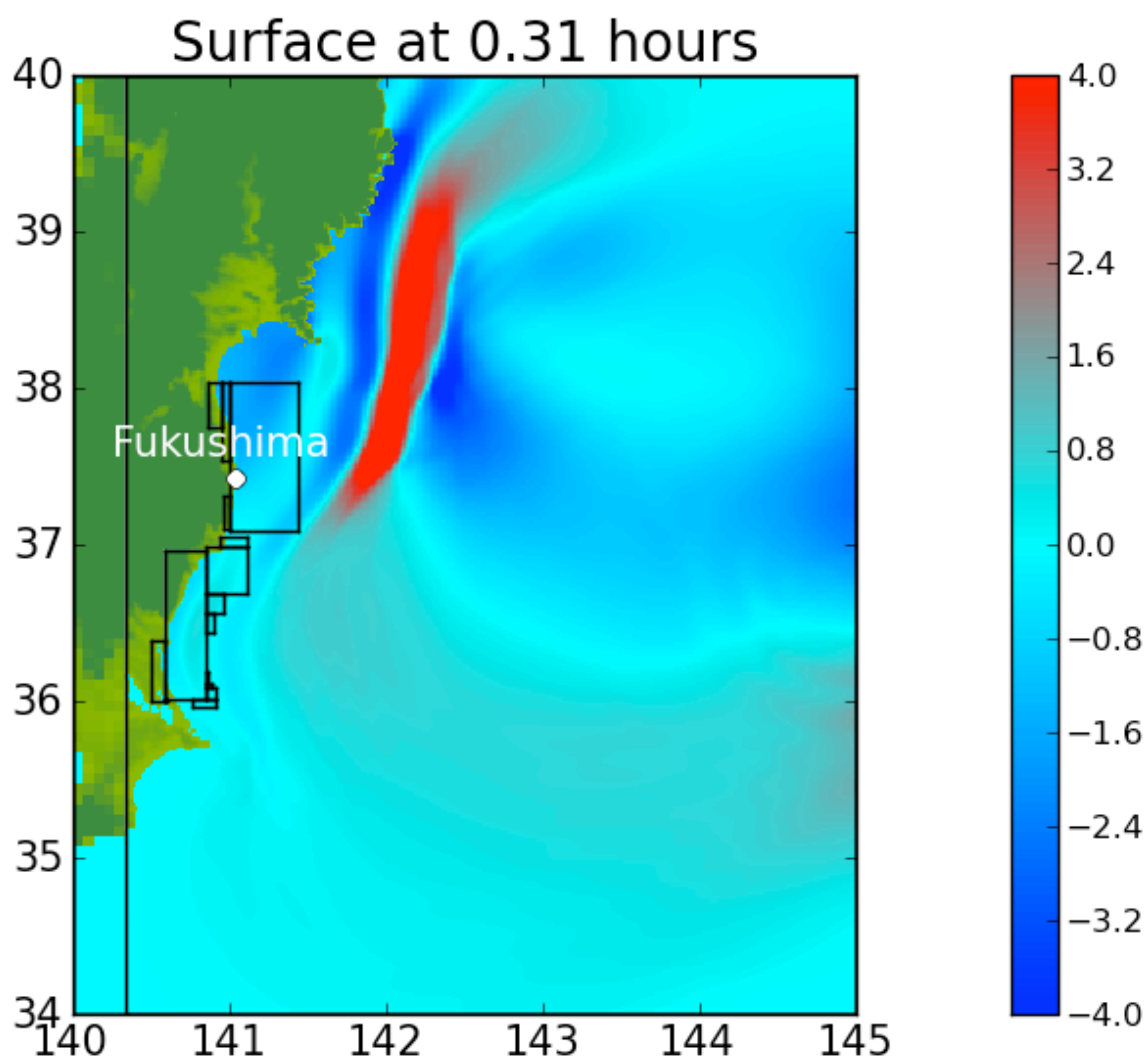
Adaptive Mesh Refinement

- Maintaining constant values (free-stream preservation)
- Implementation in AMRClaw or ChomboClaw

3d transport

- Proper definition of metric terms to maintain geometric divergence-free condition
- Handling discontinuities at mapping boundaries

GeoClaw and Tsunami modeling



R. J. LeVeque, D. George and M. Berger

http://kingkong.amath.washington.edu/clawpack/links/honshu2011/