

# A logically Cartesian, adaptively refined two-patch sphere grid for modeling transport in the atmosphere

**Donna Calhoun** (Boise State University, USA)

**Carsten Burstedde** (University of Bonn, Germany)

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

PDEs on the Sphere

Cambridge, UK

Sept 24-28, 2012

# Start with some understanding of ...

- Finite volume schemes using the wave propagation for logically Cartesian grids (LeVeque),
- Grids and methods for solving PDEs on a sphere (Berger, DC, Helzel, LeVeque),
- Block-structured adaptive mesh refinement (Berger, Colella, DC, LeVeque), and
- Experience with several software packages for doing the above (Clawpack, AMRClaw, Chombo, ChomboClaw, Boxlib) and knowledge of many more (SAMRAI, Paramesh, Overture, ...).

## Add ...

- Belief in the importance of designing numerical algorithms that from the start can be made adaptive and will scale to large number of cores,
- Appreciation of the need for domain expertise in target applications,
- A desire to be more than just a user of other people's software, but to develop my own software, and
- A full awareness of the ridiculousness of doing the above by myself, at least while in a tenure track position, in a mathematics department.

# some fortuitous meetings ...

- An workshop on tracer transport on the sphere  
(NCAR, March 2011, Boulder CO)
- A meeting with USGS researchers from the Cascade Volcanic Observatory interested in using adaptivity to improve their forecasting of volcanic ash cloud transport (Seattle, October 2011)
- A hands-on workshop on high performance computing at the King Abdullah University of Science and Technology (KAUST) (HPC<sup>3</sup>, March 2012, Saudi Arabia)
- An opportunity to spend the semester at the Newton Institute at a program on Multiscale Numerics for the Ocean and Atmosphere (Fall, 2012)

and we get ...

## **ForestClaw = p4est + Clawpack**

- Adaptive, scalable, multi-block tree-code **p4est** (Carsten Burstedde, Univ. Bonn) to manage a quad- or octree of non-overlapping fixed sized grids,
- Wave propagation algorithms (R. J. LeVeque) on logically Cartesian mapped grids,

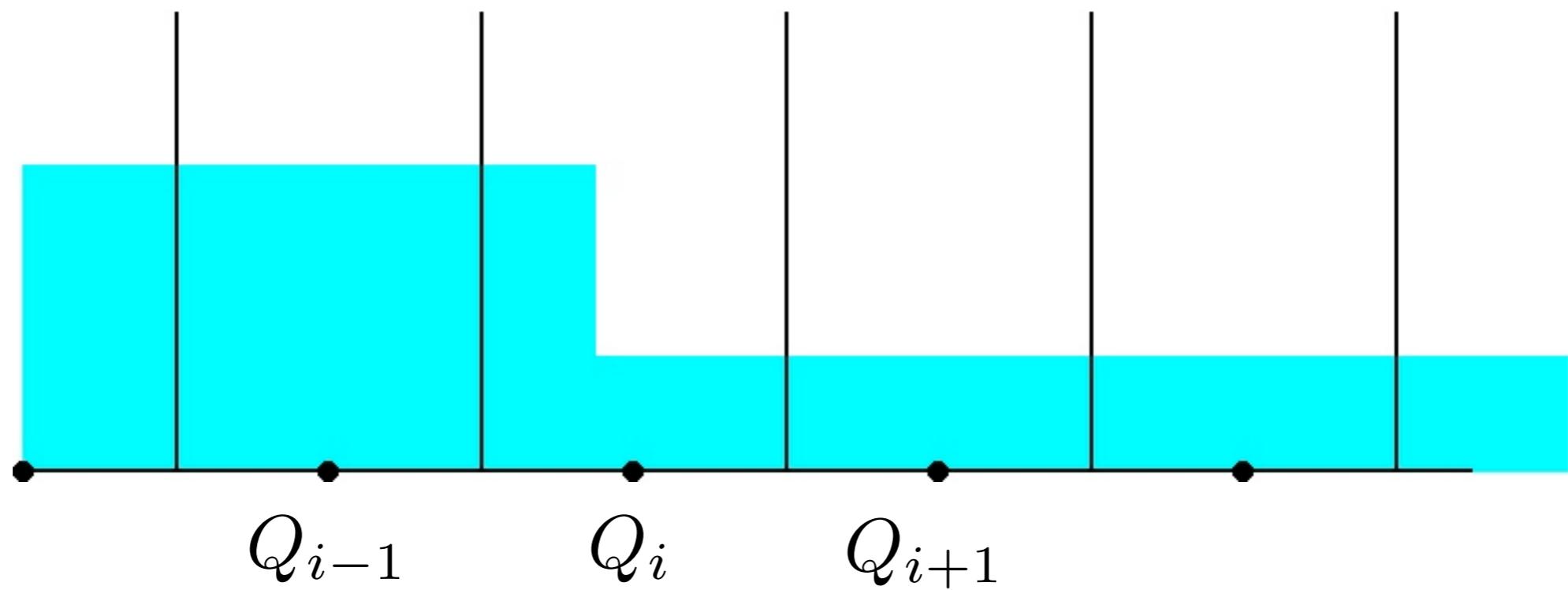
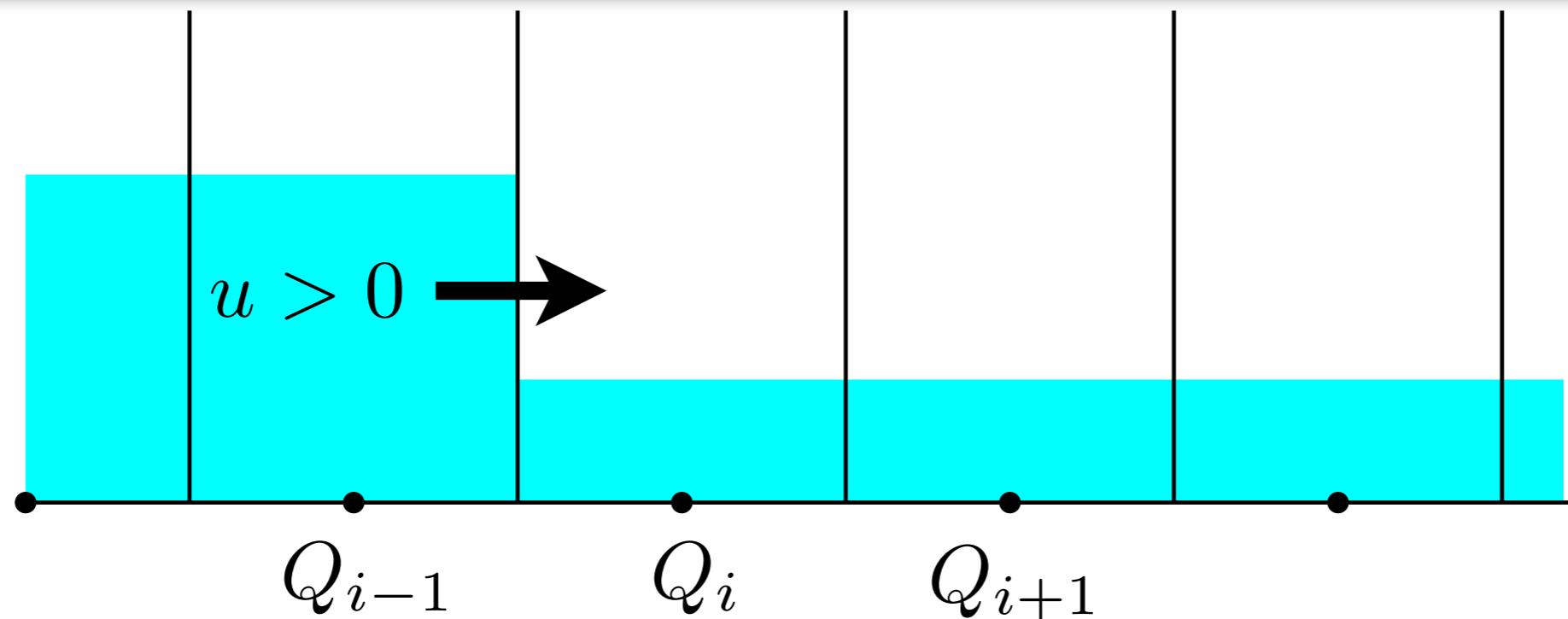
*Current aim is to use adaptivity to improve forecasting capabilities of **Ash3d**, a USGS code for modeling transport of volcanic ash in the atmosphere.*

# Wave propagation algorithms on the sphere

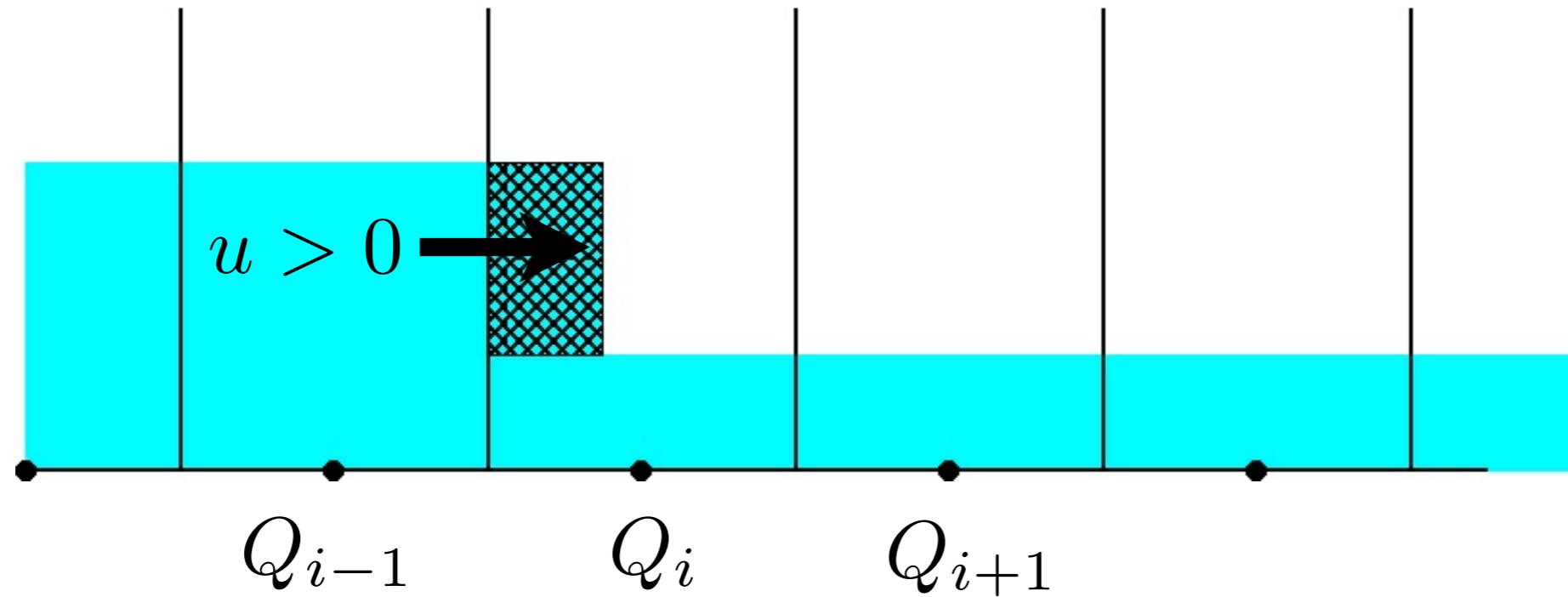
Clawpack (R. J. LeVeque) general purpose code based on the wave propagation algorithm for solving hyperbolic problems.

- 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.
- Explicit time stepping limited by  $CFL=1.0$

# Wave propagation algorithms



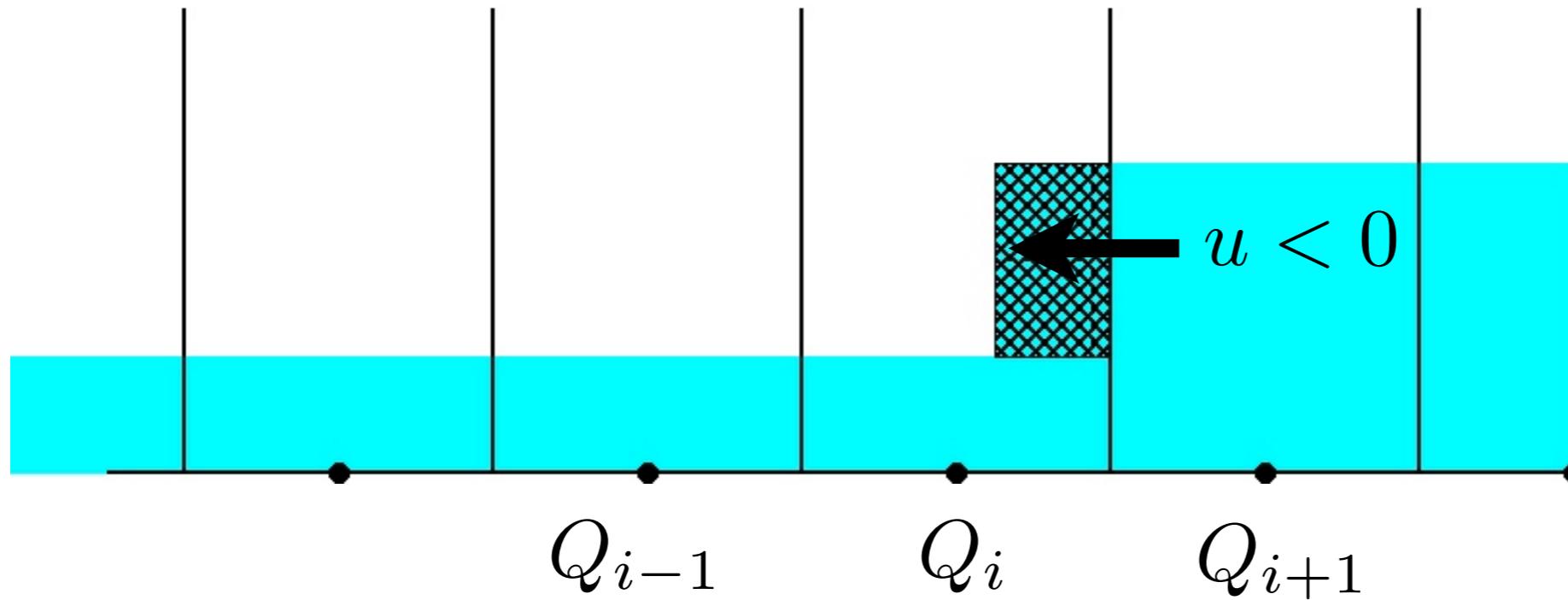
# Wave propagation algorithms



In time  $\Delta t$ , total mass in the cell increases by shaded area :

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

# Wave propagation algorithms



In time  $\Delta t$ , total mass in the cell increases by shaded area :

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

# Upwind scheme

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

where

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

We can define waves at each interface as :

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

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

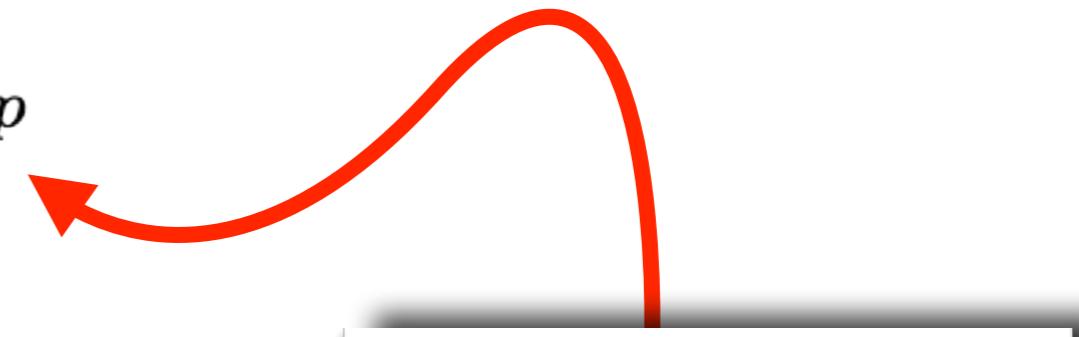
# Upwind scheme for systems

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

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

Waves :

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



Written in terms of *fluctuations* :

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

Solve Riemann problems

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

# Second order correction terms

High resolution terms gives better accuracy (2nd order for smooth solutions)

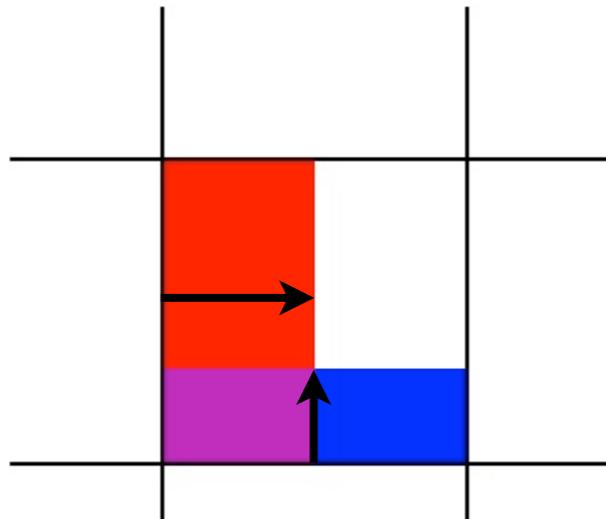
$$\begin{aligned} Q_i^{n+1} = Q_i^n & - \frac{\Delta t}{\Delta x} (\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i+1/2}) \\ & - \frac{\Delta t}{\Delta x} \left( \tilde{F}_{i+1/2} - \tilde{F}_{i-1/2} \right) \end{aligned}$$

where

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

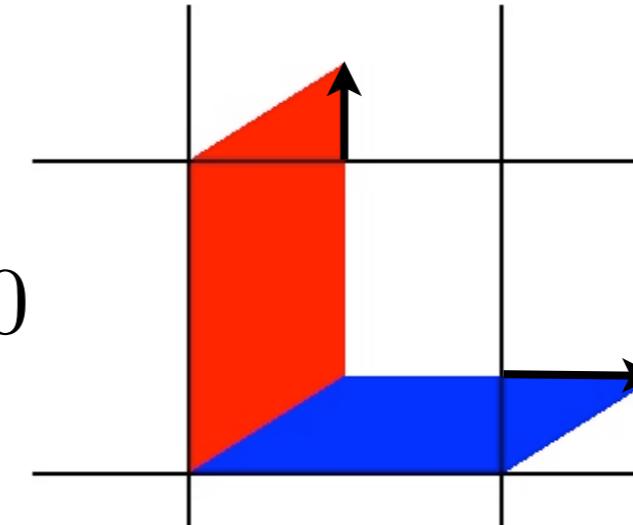
where  $\tilde{\mathcal{W}}_{i-1/2}^p$  are *limited waves*.

# Wave-propagation method in two dimensions



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

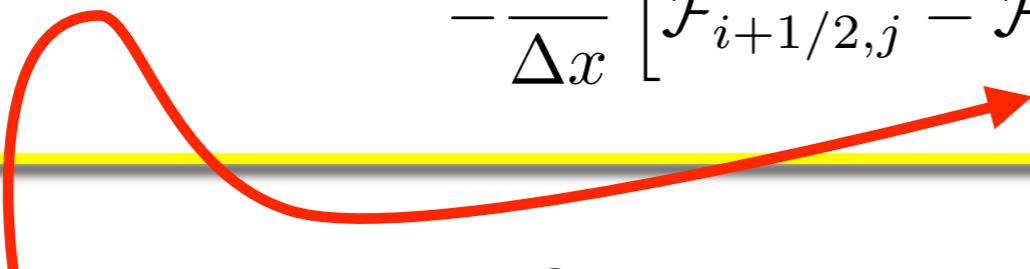
Normal waves



Transverse waves

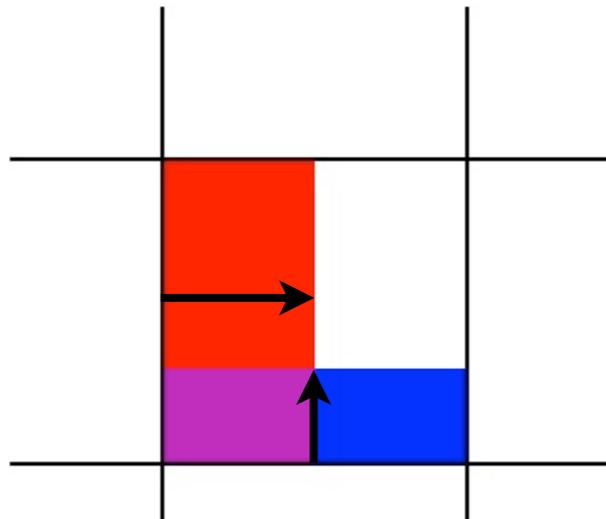
## Wave propagation algorithm

$$\begin{aligned} Q_{ij}^{n+1} = Q_{ij}^n - \frac{\Delta t}{\Delta x} [u^+ \mathcal{W}_{i-1/2,j} + u^- \mathcal{W}_{i+1/2,j}] - \frac{\Delta t}{\Delta y} [v^+ \mathcal{W}_{i,j-1/2} + v^- \mathcal{W}_{i,j+1/2}] \\ - \frac{\Delta t}{\Delta x} [\tilde{\mathcal{F}}_{i+1/2,j} - \tilde{\mathcal{F}}_{i-1/2,j}] - \frac{\Delta t}{\Delta y} [\tilde{\mathcal{G}}_{i,j+1/2} - \tilde{\mathcal{G}}_{i,j-1/2}] \end{aligned}$$



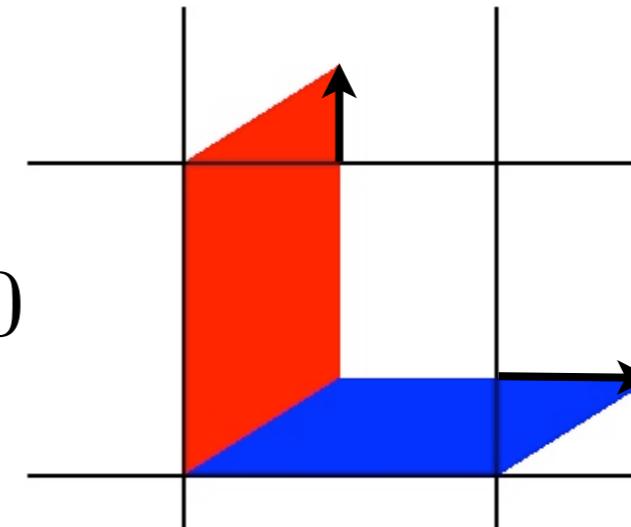
Correction terms come from linear terms and transverse propagation

# Wave-propagation method in two dimensions



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

Normal waves



Transverse waves

## Wave propagation algorithm

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

# 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

$$q_t + f(q, x)_x = 0$$

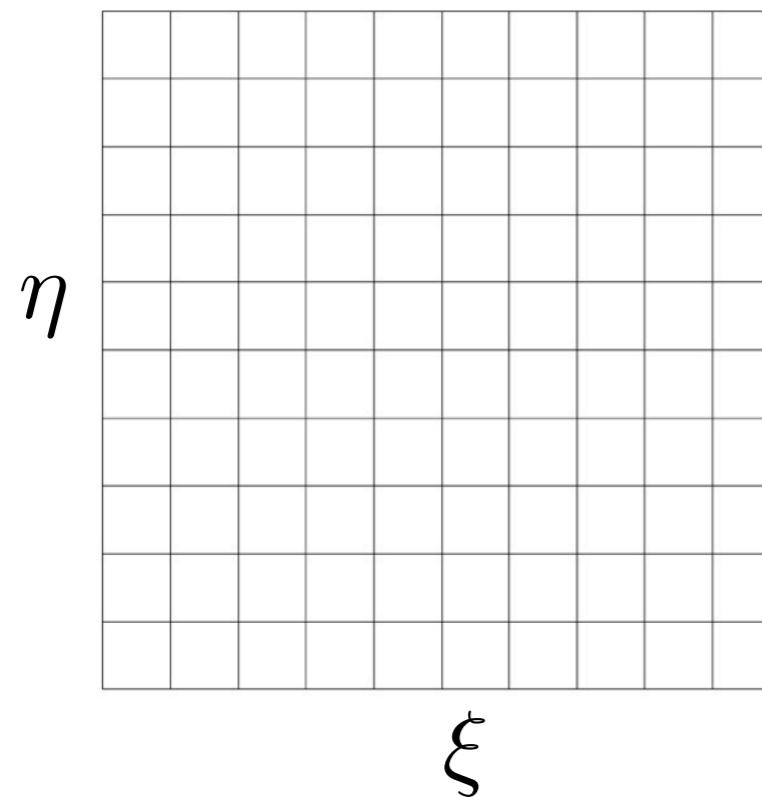
Source terms

$$q_t + f(q)_x = \Psi(q, \dots)$$

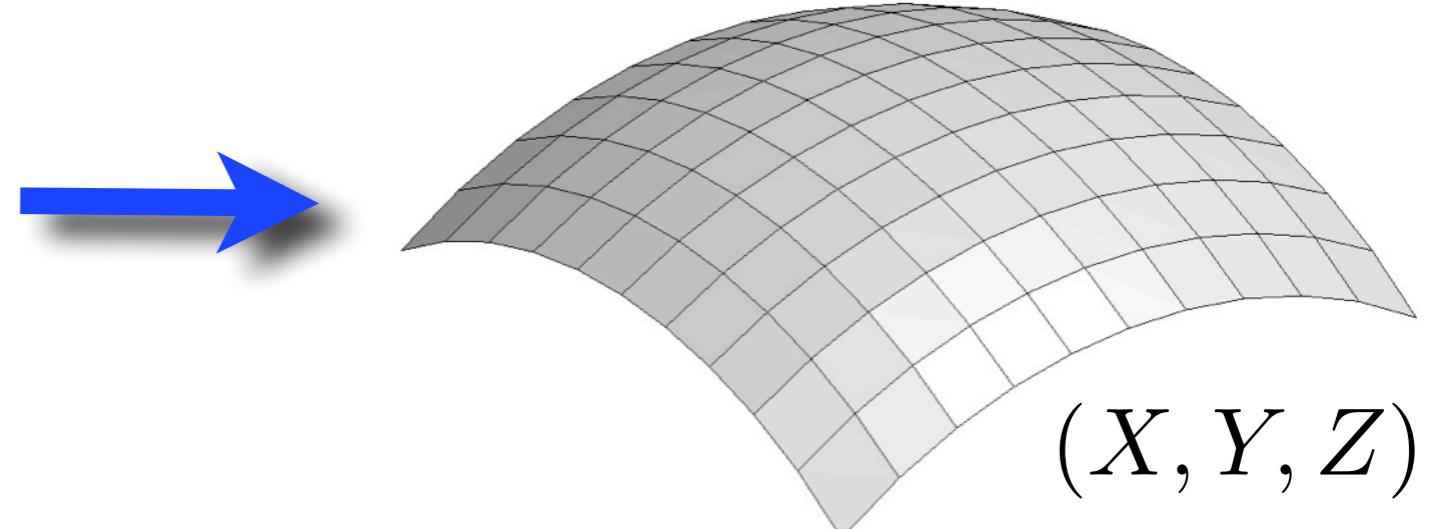
# Finite volume schemes on curvilinear grids

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

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

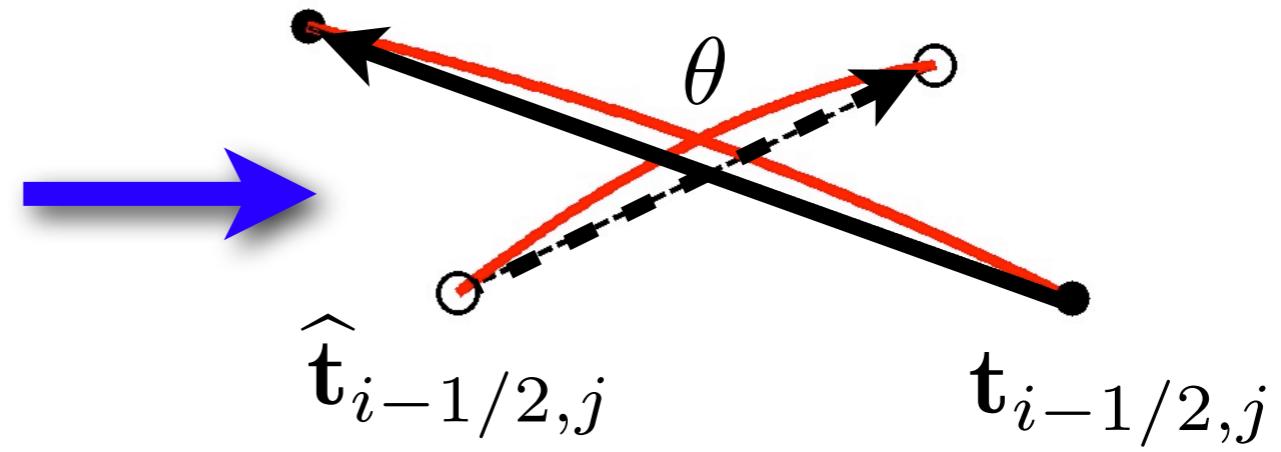
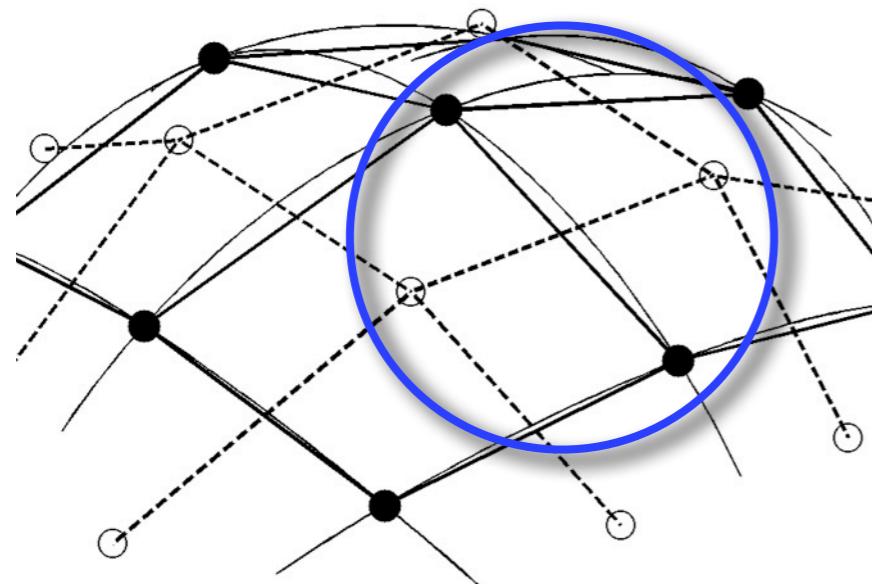


Computational space



Physical space

# Discrete metric terms



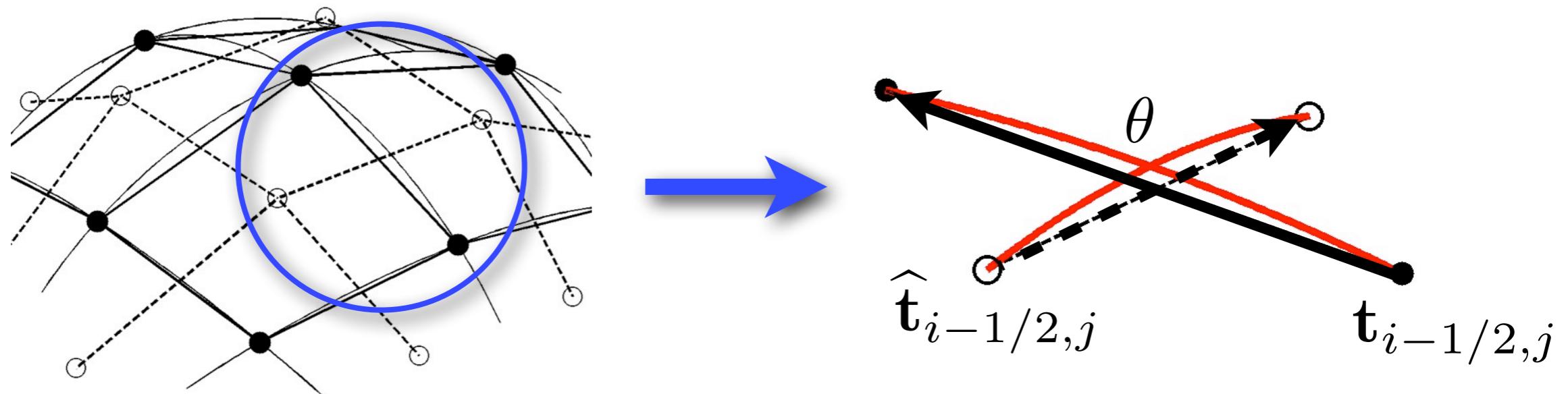
$$a_{11} = \mathbf{T}_\xi \cdot \mathbf{T}_\xi \approx \mathbf{t} \cdot \mathbf{t} = \|\mathbf{t}\|^2$$

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

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

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

# 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

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

# 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 :

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

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

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

# FV update formula for curvilinear grids

$$Q_{ij}^{n+1} = Q_{ij}^n - \frac{\Delta t}{\kappa_{ij} \Delta \xi} (u^+ \Delta Q_{i-1/2,j} + u^- \Delta Q_{i+1,j}) - \frac{\Delta t}{\kappa_{ij} \Delta \eta} (v^+ \Delta Q_{i,j-1/2} + v^- \Delta Q_{i,j+1}) - \frac{\Delta t}{\kappa_{ij} \Delta \xi} (\tilde{F}_{i+1/2,j} - \tilde{F}_{i-1/2,j}) - \frac{\Delta t}{\kappa_{ij} \Delta \eta} (\tilde{G}_{i,j+1/2} - \tilde{G}_{i,j-1/2})$$

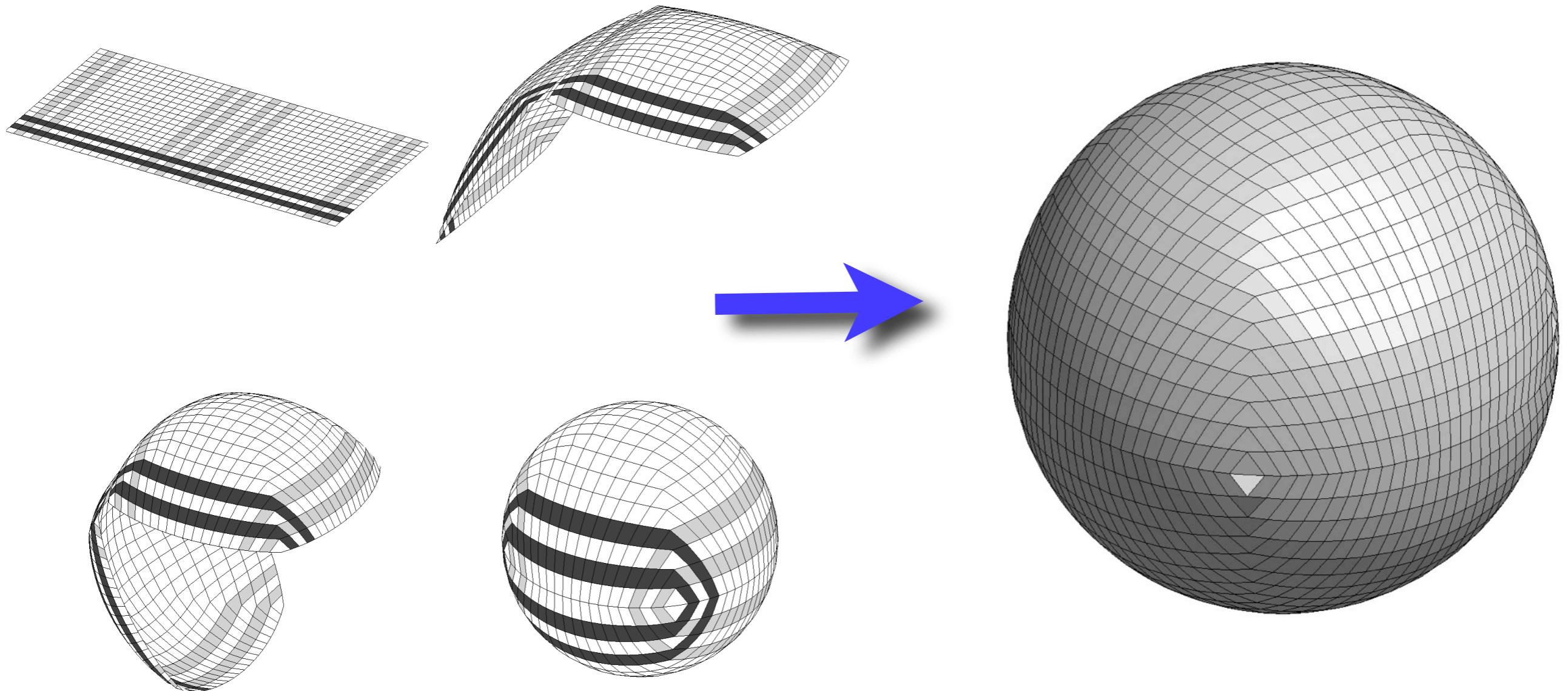
where

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

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

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

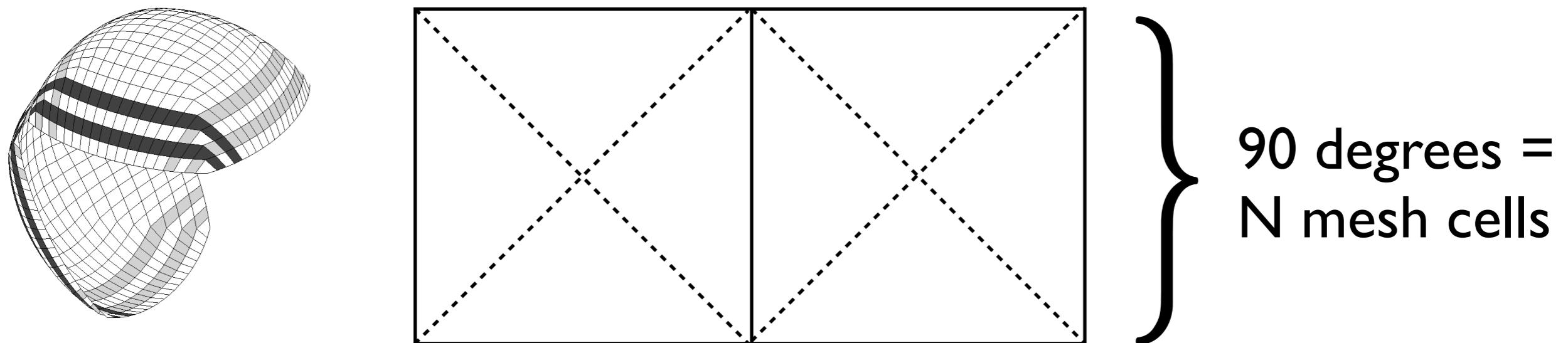
# Single logically Cartesian sphere grid



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

# A two-patch sphere grid?

- 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.

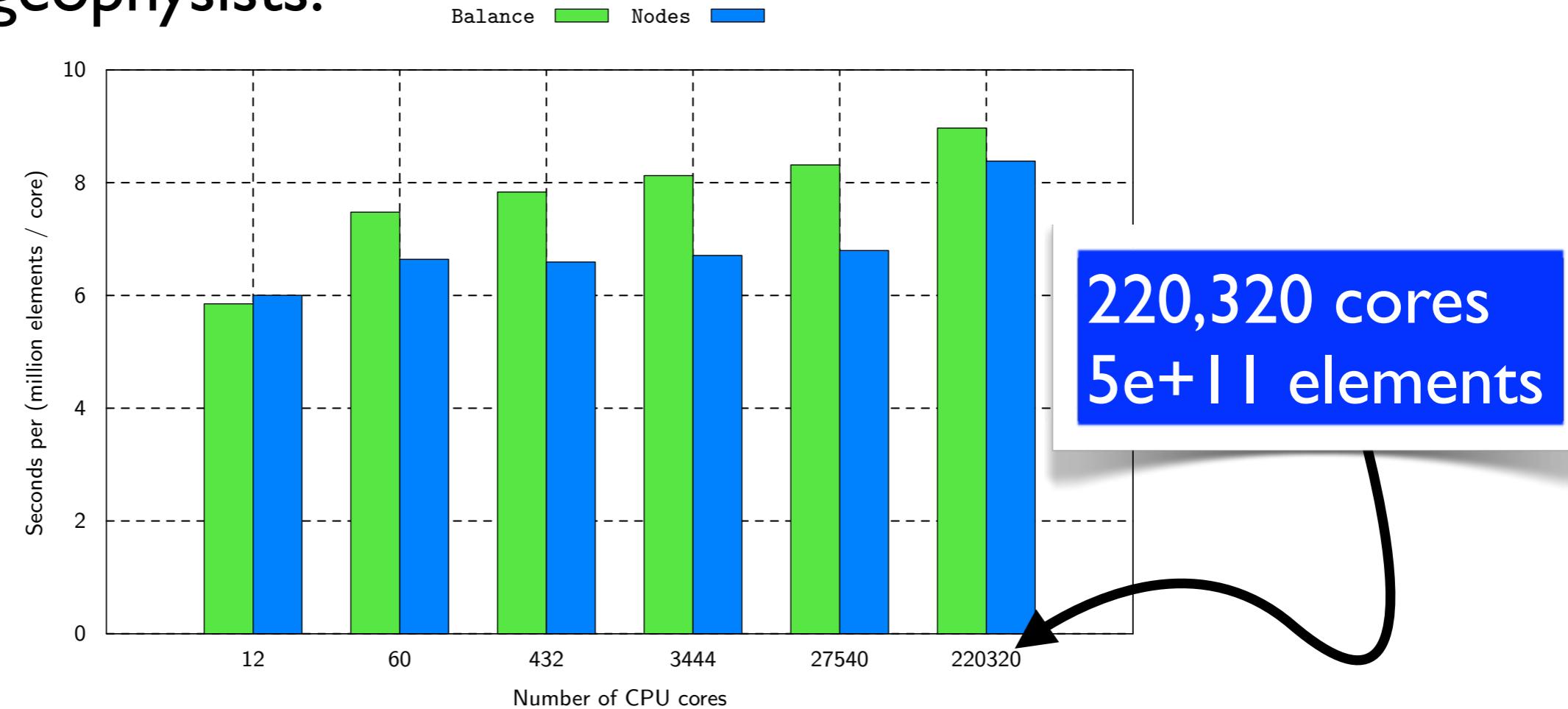


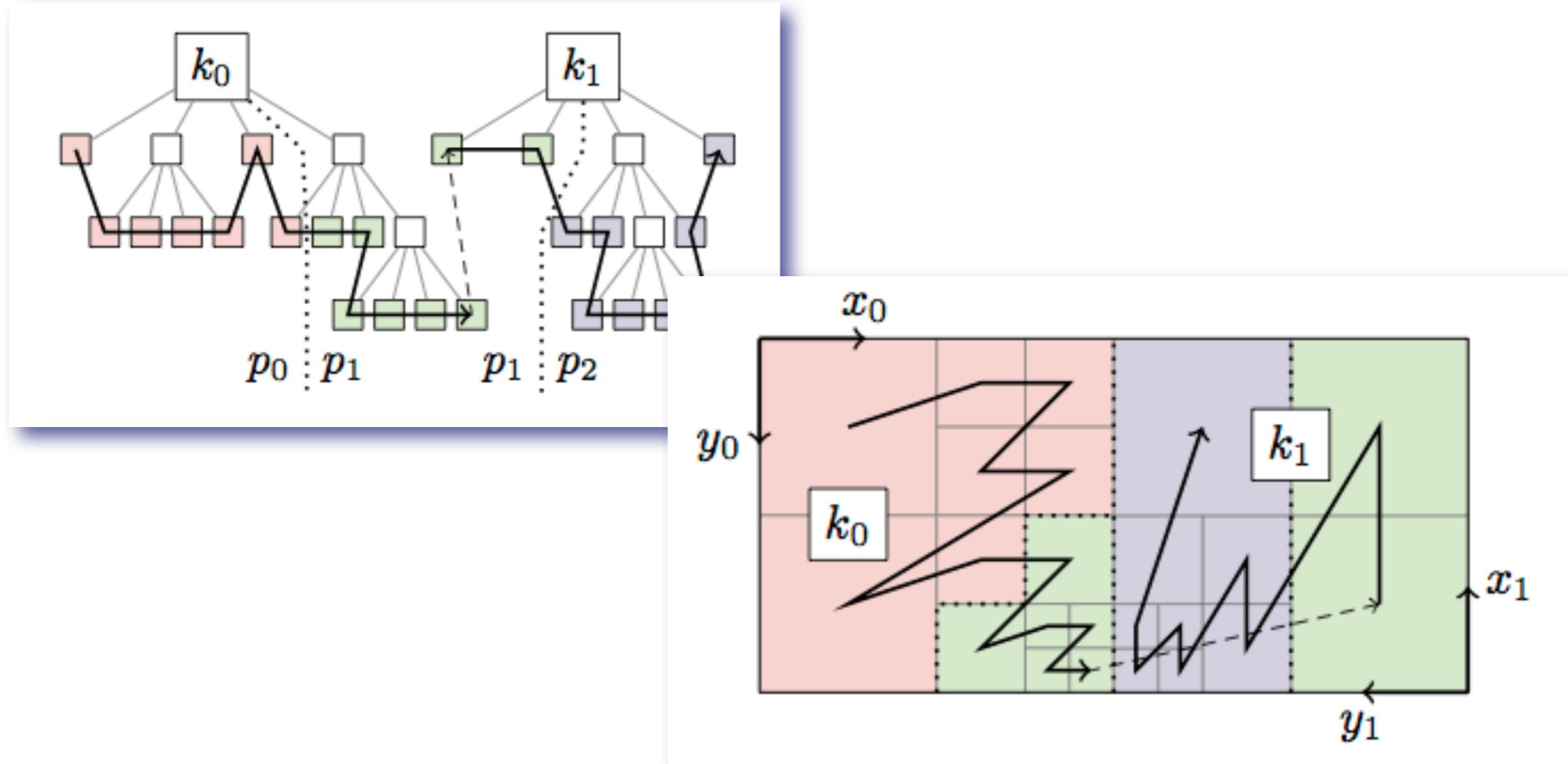
*Dashed and solid lines are discontinuities in the mapping*

# A Pillow grid?

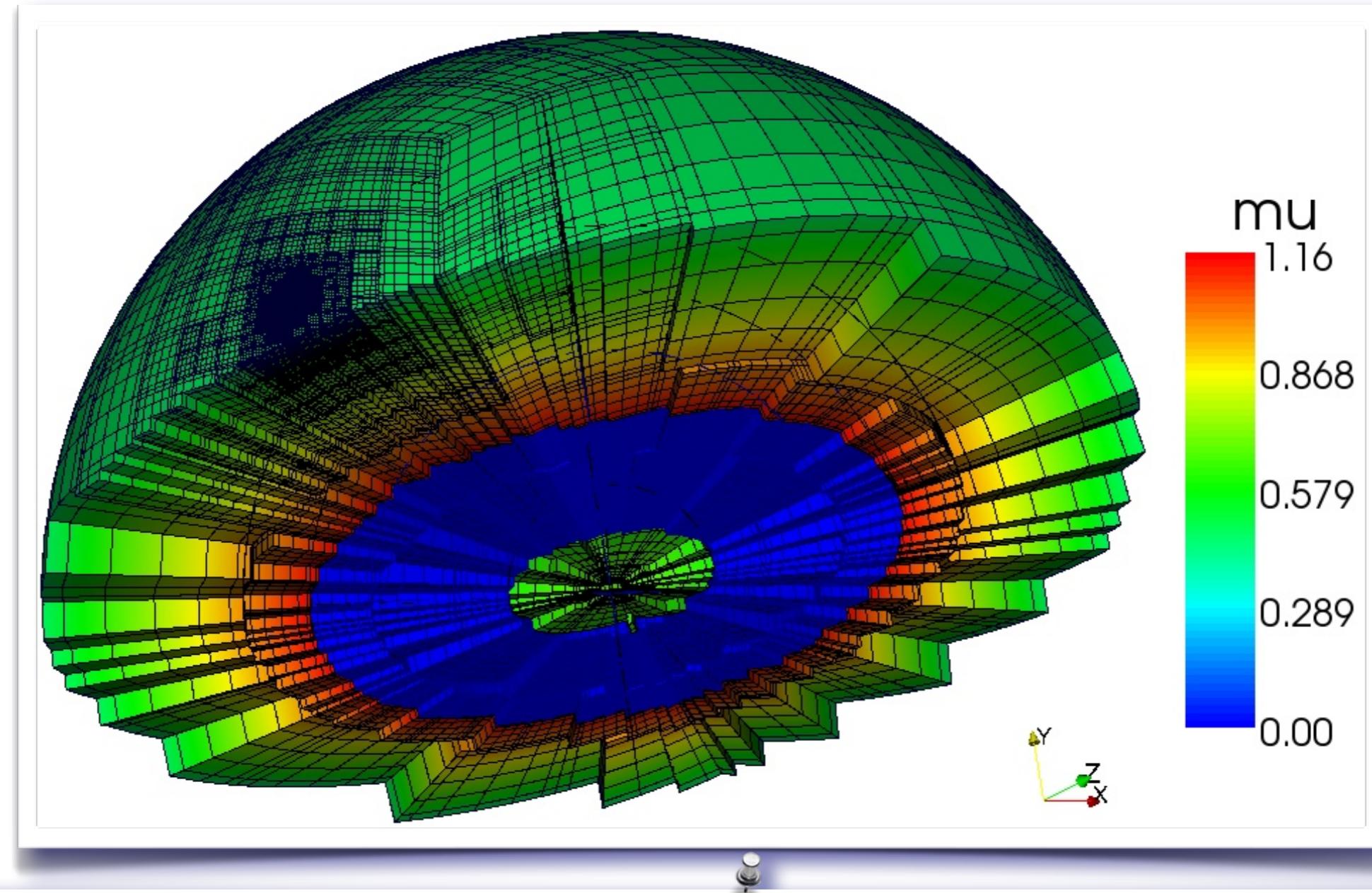


- Parallel, multiblock code for managing a forest of adaptive quad- or octrees.
- Developed by Carsten Burstedde (Univ. of Bonn), with Wilcox, Ghattas and others
- Highly scalable on realistic applications of interest to geophysists.

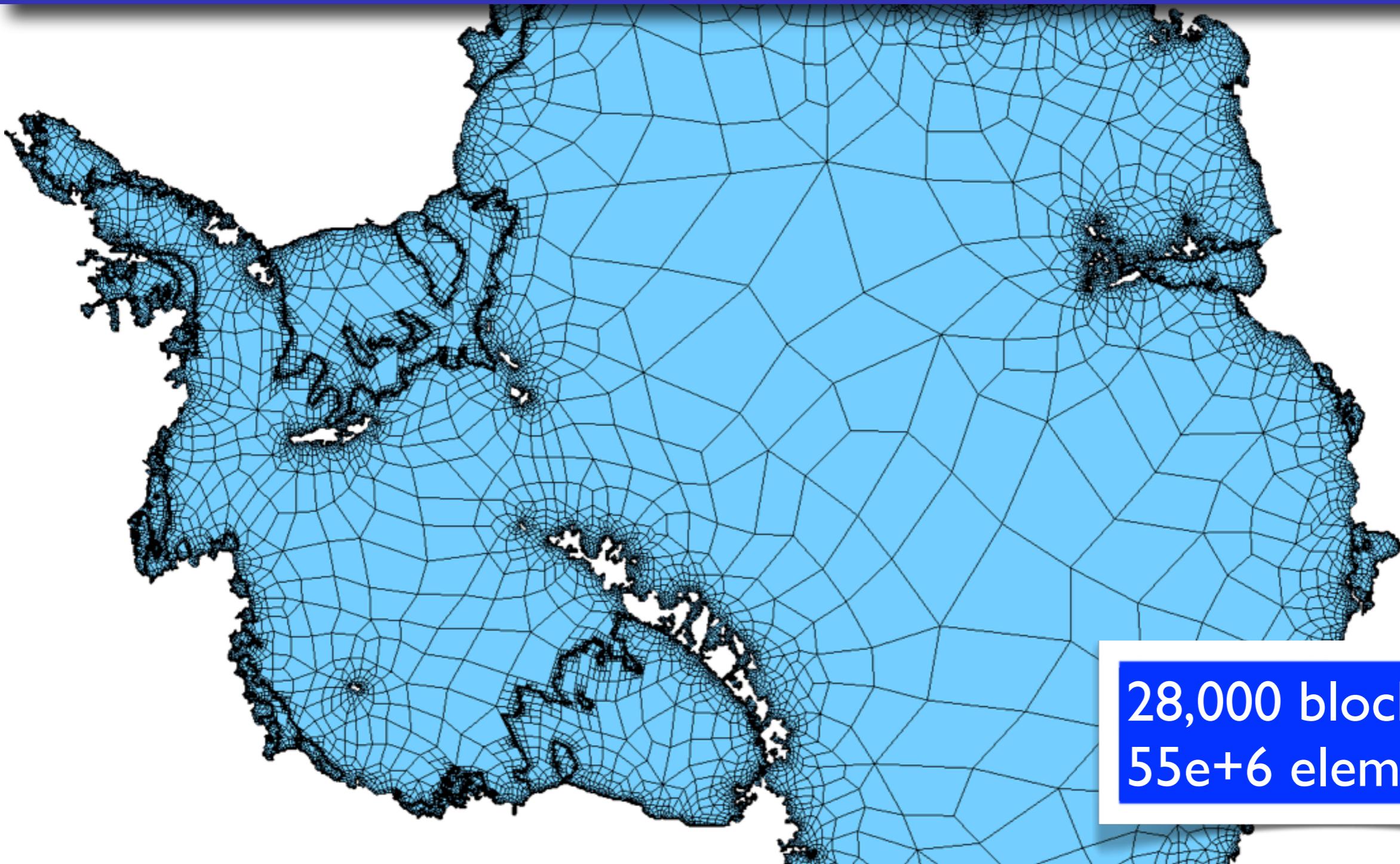




High scalability is achieved while preserving data locality



*Earthquake simulation using p4est on a cubed sphere mesh. (Burstedde, Wilcox, Ghattas, SISC, 2011)*



Antarctic ice sheet modeling (Tobin Isaac, C Burstedde)

# ForestClaw

- Use p4est to manage the parallel multiblock of quad- or octree whose leaves are non-overlapping, fixed size grids.
- Implement basic block-structured AMR algorithm (Berger, Colella, LeVeque and others)
- Reuse existing patch based solvers for single grids
- Do subcycling in time on finer grids to improve scalability and accuracy,
- Ensure mass conservation at coarse/fine interfaces
- Load balancing achieved by ensuring blocks are the same size.
- p4est stores information about face orientations.

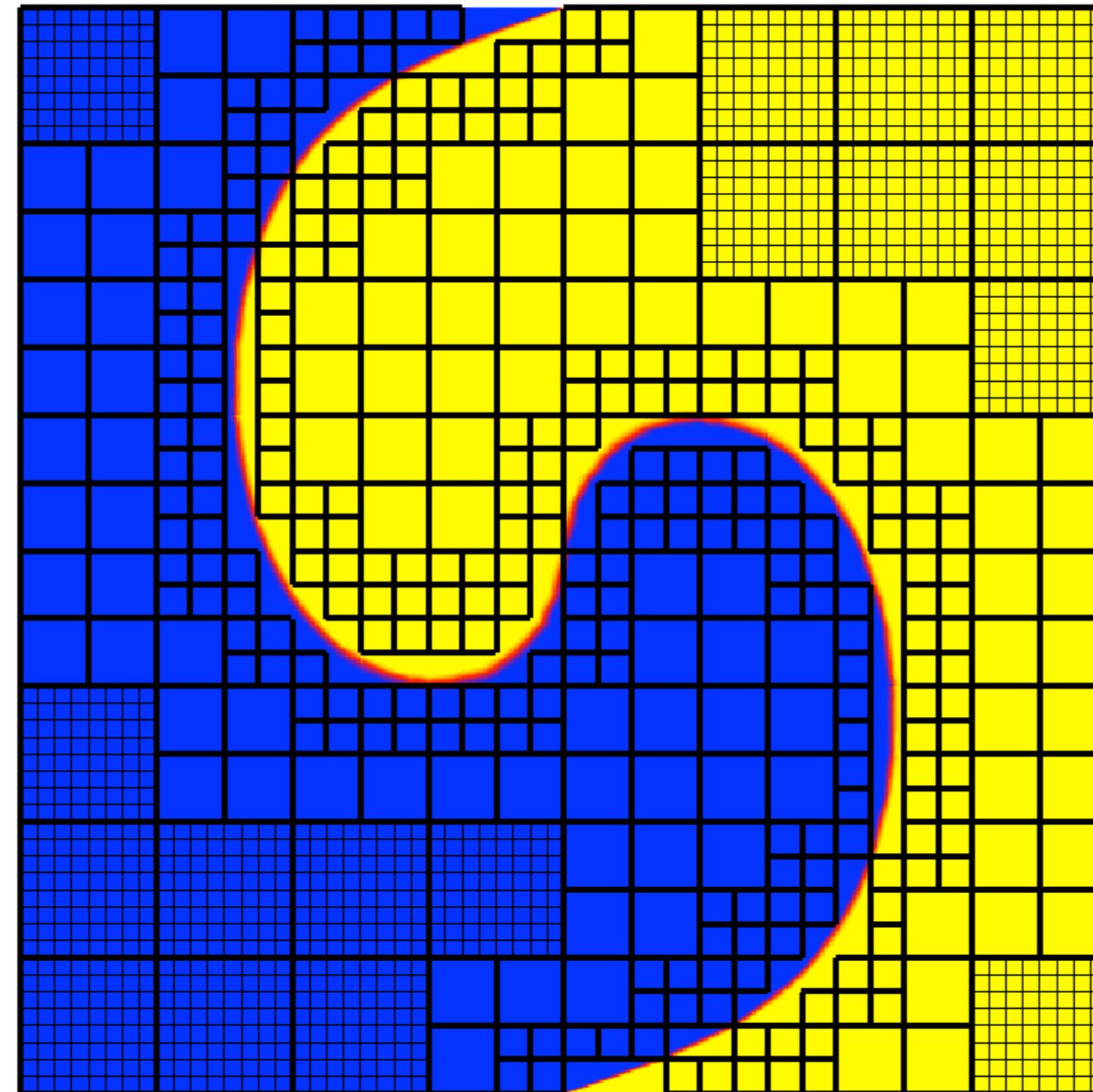
# Similar approaches

- “Building Cubes Method” (Sasaki, Akahito, Yamazaki, et al.)
- Parallel adaptive methods for weather prediction C. Jablonowski, Oehmke, Stout and others
- Gerris Flow solver - cell-based octree approach (S. Popinet)
- NIRVANA (U. Ziegler)
- Block-structured AMR codes (Chombo, Boxlib, AMRClaw, SAMRAI, AMROC, Agrif, ....)

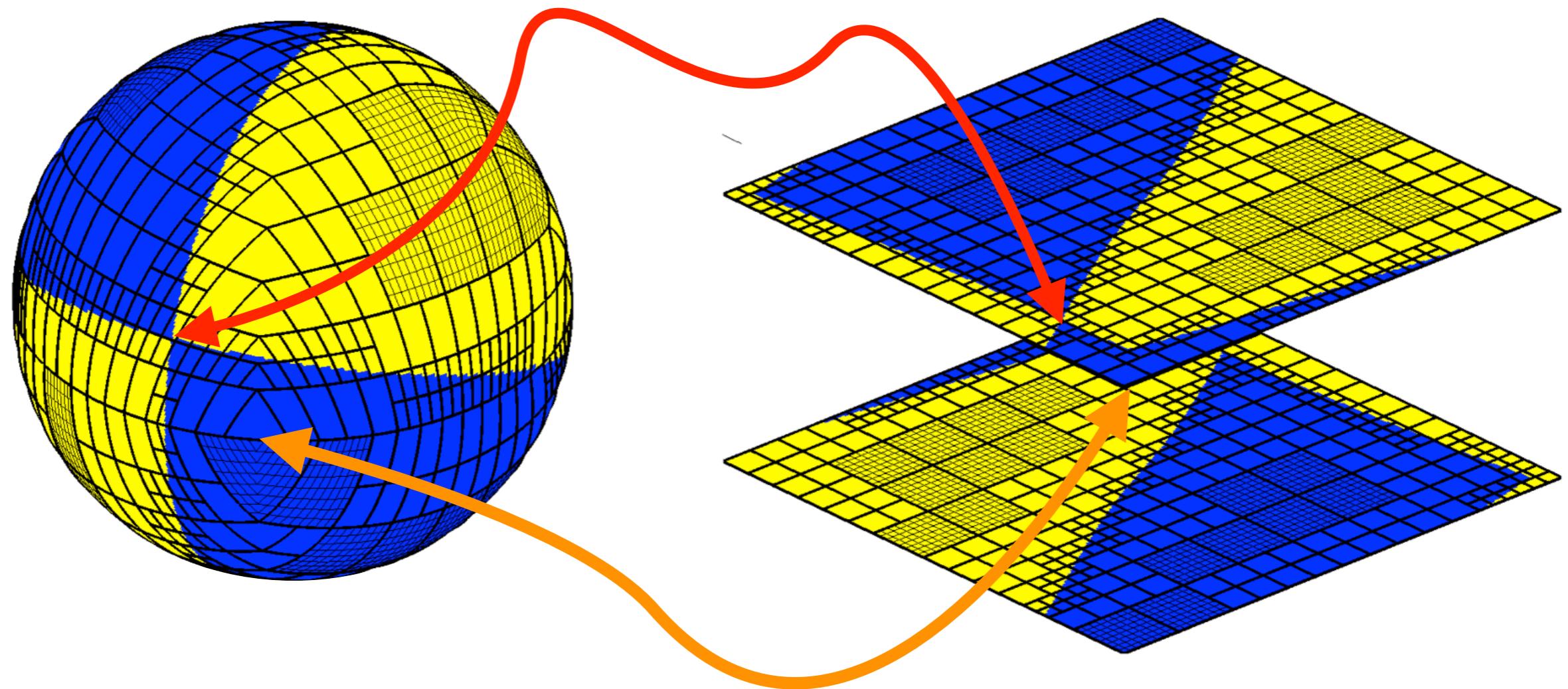
# Results

# Results

$q(1)$  at time 2.5009

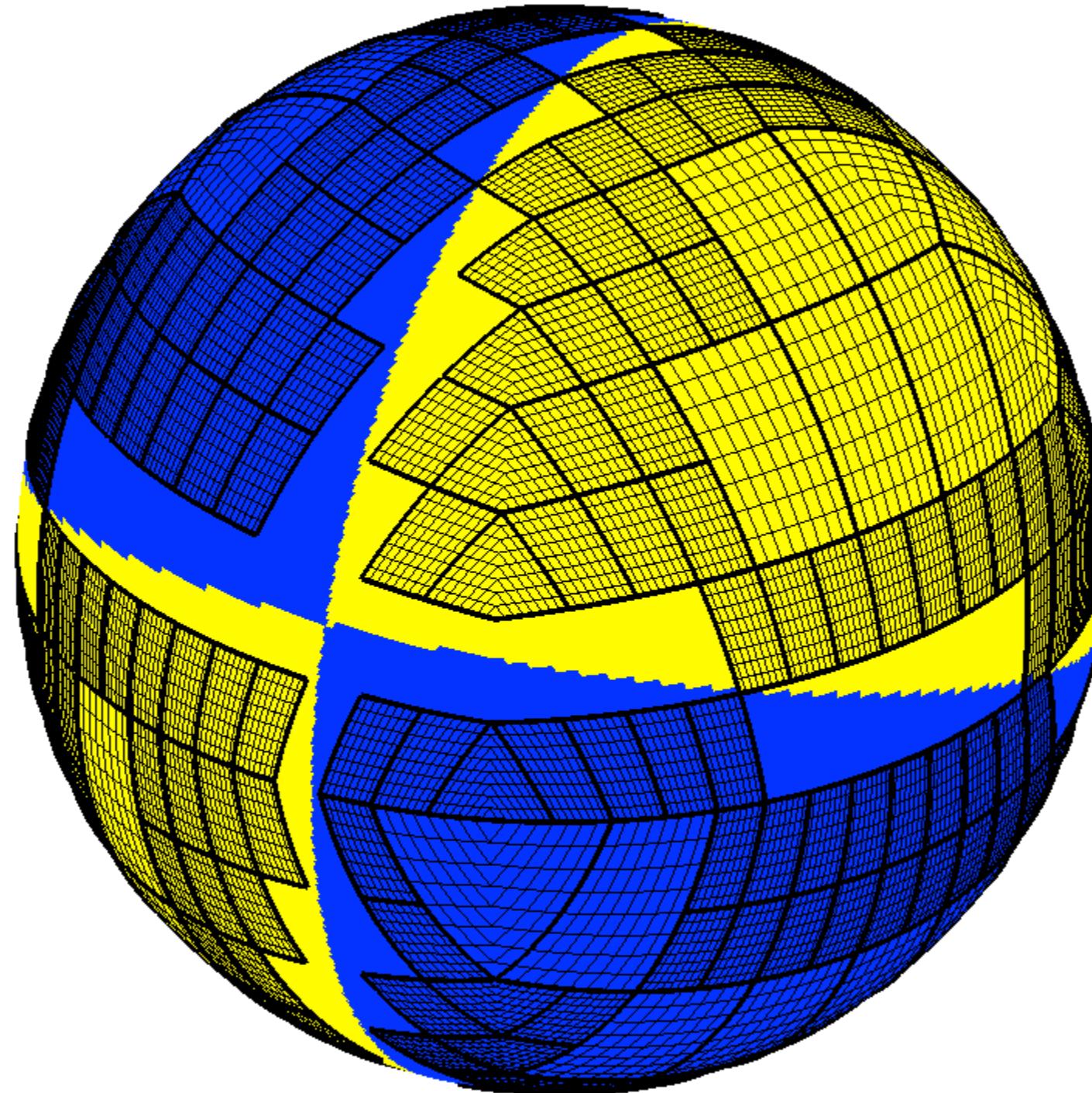


# Scalar transport on the sphere



# Scalar advection

# Scalar advection



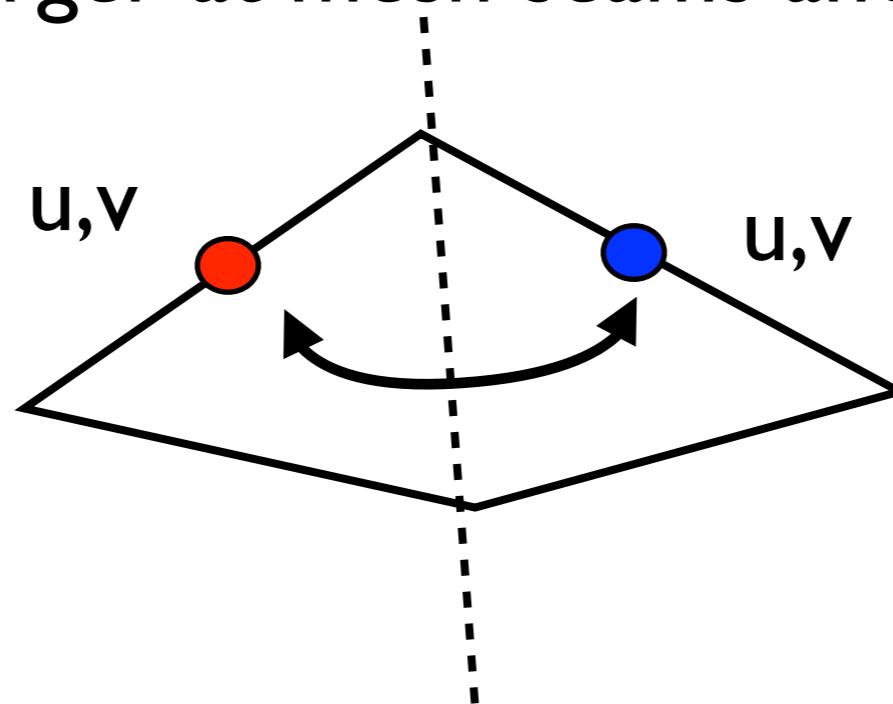
# Refinement criteria

- Feature-based refinement - refine near boundaries, where a tracer is highly concentrated, large gradients
- Extrapolate from coarser grid solutions to estimate truncation error
- A posteriori error estimation
- Adjoint methods

*A code should be flexible enough to allow the user to experiment with different choices.*

# Current issues

- Convergence is around 1.6 - 1.8, depending on where error is measured,
- Accuracy of our methods depends on the smoothness of the mesh.
- Errors are larger at mesh seams and along the diagonals



Transverse propagation uses velocities on two sides of a discontinuity, introducing  $O(1)$  errors in the truncation error.

# Future work

(From NCAR Tracer Transport 2011 meeting)

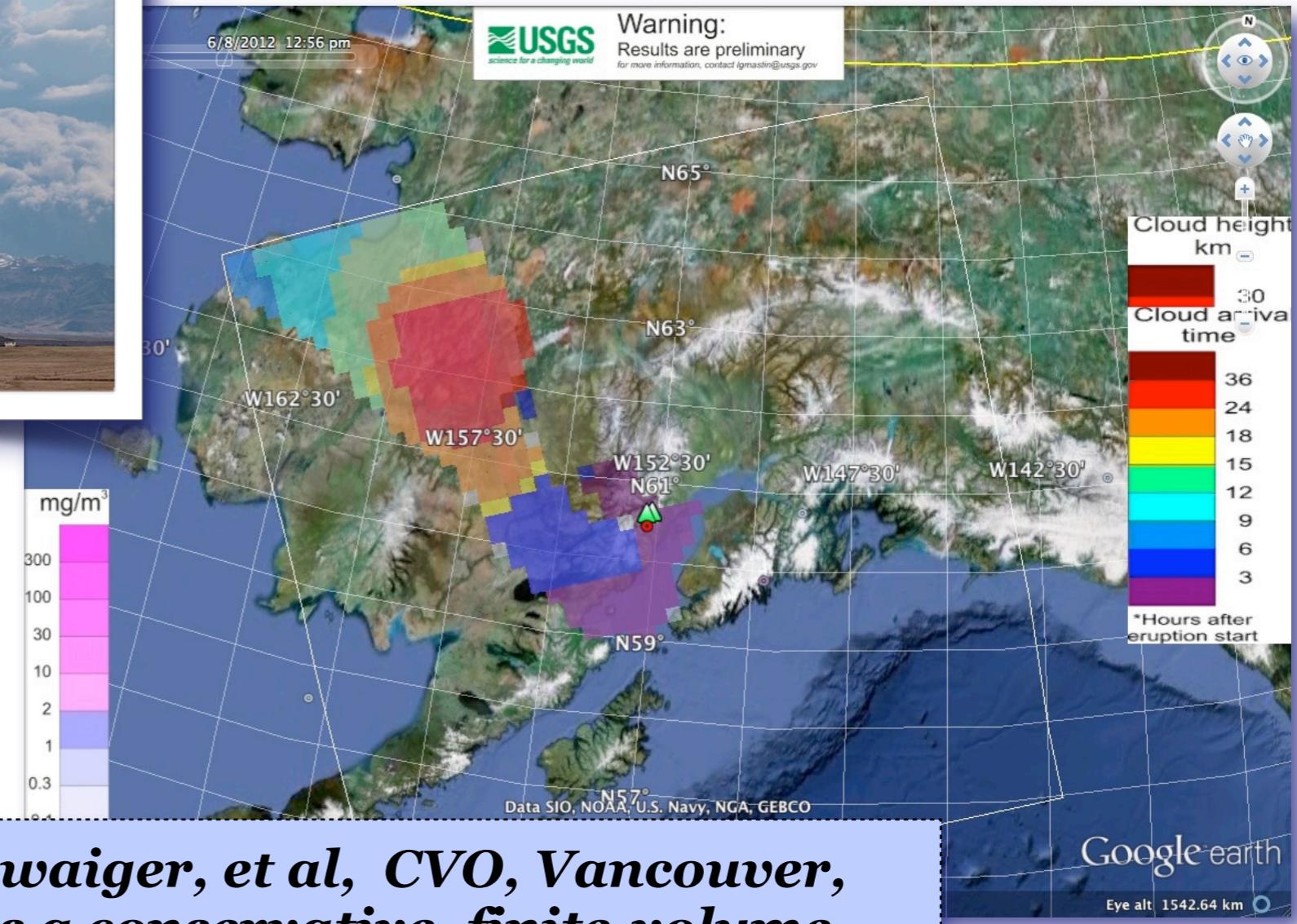
- Adaptive Mesh Refinement
- Implementation in ~~AMRClaw or ChomboClaw~~

- Develop general way to handle possible cases for face orientations in the multiblock setting.
- More general approach to handling user input, including specification of block layout and orientations,
- Scaling tests,
- Accuracy tests
- and ...

# Ash cloud modeling



Ash3d



*Ash3d (Schwaiger, et al, CVO, Vancouver, WA, USA) is a conservative, finite-volume code developed to forecast potential hazards associated with volcanic ash dispersal.*