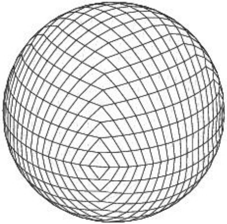
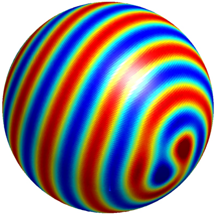
Quadrilateral Grids and Finite Volume Methods on the Sphere





Randall J. LeVeque, University of Washington

Donna Calhoun, Commissariat à l'Energie Atomique

Christiane Helzel, University of Bochum

Supported in part by NSF, DOE, German Science Foundation

Links

The animations shown in this talk are collected at:

http://www.clawpack.org/links/birs08

along with a pointer to the paper

Logically Rectangular Grids and Finite Volume Methods for PDEs in Circular and Spherical Domains, by D. Calhoun, C. Helzel, and RJL, to appear in *SIAM Review*

which contains more examples and codes in Fortran, Matlab, and Python.

Goals

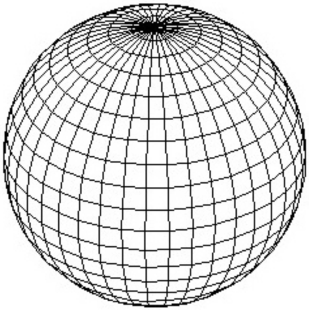
- Logically rectangular grids:
  - In 2d: quadrilaterals with (i, j) indexing
  - In 3d: hexahedra with (i, j, k) indexing
- Mapping to domains with smooth boundary:
  - Circle in 2d
  - Surface of sphere
  - Interior of ball in 3d
  - Smooth mappings of above domains
- Mapping of single rectangle (not multiblock)
- Nearly uniform cell sizes

Why logically rectangular?

- Efficient algorithms — sweeping over grid
- Adaptive mesh refinement on rectangular patches

  Berger–Colella–Oliger style AMR
- Can apply AMR software such as
  - AMRCLAW — incorporated in Clawpack
  - CHOMBO-CLAW — Donna Calhoun
  - BEARCLAW — Sorin Mitran
  - AMROC — Ralf Deiterding

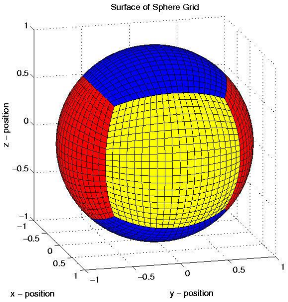
Latitude-Longitude grid on sphere

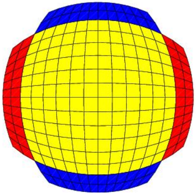


Logically rectangular, but suffers from “pole problem”

- Grid lines coalesce at poles, tiny cells
- Small time steps needed for explicit methods

Cubed Sphere Grid: another popular approach





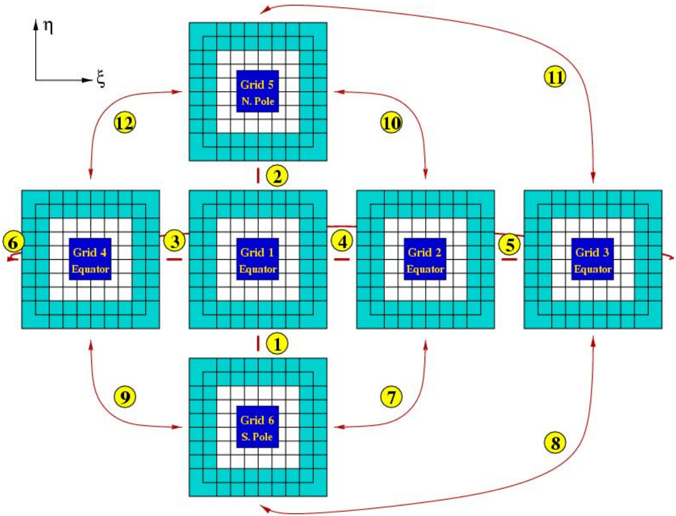
Six logically rectangular grids are patched together.

Data is transferred between patches using ghost cells

Refs: Sadourny (1972), Ronchi, Iacono, Paolucci, Rancic, Purser, Messinger,...

Rossmanith implemented with CLAWPACK

Boundary conditions for cubed sphere

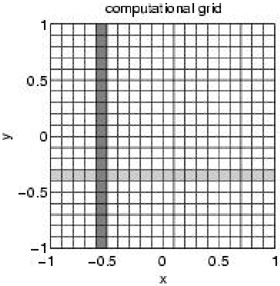


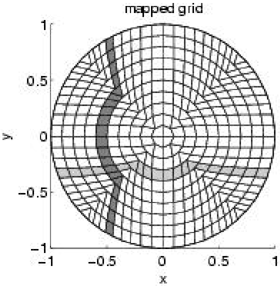
Our approach for circles

Radial projection grid:

Computational domain is square $[-1,1] \times [-1,1]$.

Map each point on concentric square of "radius" $d \leq 0$ radially inward to circle of radius d.





Movie of radial projection

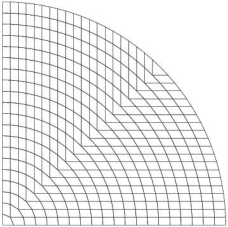
Our approach for circles

Smoother grid:

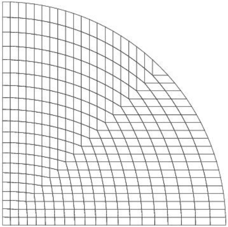
Map line segment $(-d, d)$ to (d, d) to circular arc of radius $R(d)$ passing through the points $(-D(d), D(d))$ and $(D(d), D(d))$.

Similarly in other three quadrants.

$D(d) = d, \;\; R(d) = 1:$



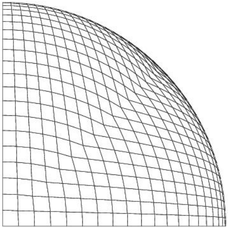
$D(d) = d, \;\; R(d) = 1:$



Our approach for circles and sphere

Redistribute points near boundary:

$$D(d) = d(2-d), \quad R(d) = 1:$$



Gives good mapping to upper hemisphere
(think of looking down on sphere)

Our approach for sphere

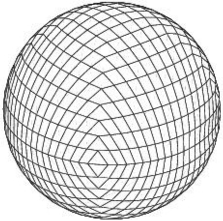
Map $[-1,1] \times [-1,1]$ to unit circle by this approach.

At each point set $z = \sqrt{1-(x^2+y^2)}$.

This defines mapping of $[-1,1] \times [-1,1]$ to upper hemisphere.

Map points in $[-3,-1] \times [-1,1]$ to lower hemisphere by similar mapping.

This defines mapping of rectangle $[-3,1] \times [-1,1]$ to sphere.



Ratio of largest to smallest cell is < 2.

Grid is highly non-orthogonal at a few points near equator.

Movie of mapping

Numerical results on the sphere

Direct application of CLAWPACK — wave-propagation finite volume method

Movie — advection on the sphere

Movie — in computational rectangle

Movie — shallow water on the sphere

Movie — depth vs. “latitude” compared to 1d solution

Movie — with AMR

Movie — with AMR in computational plane

Wave propagation algorithms

CLAWPACK requires:

- Normal Riemann solver `rpn2.f`
  Solves 1d Riemann problem $q_t + Aq_x = 0$
  Decomposes $\Delta Q = Q_{ij} - Q_{i-1,j}$ into $\mathcal{A}^+\Delta Q$ and $\mathcal{A}^-\Delta Q$.
  For $q_t + Aq_x + Bq_y = 0$, split using eigenvalues, vectors:

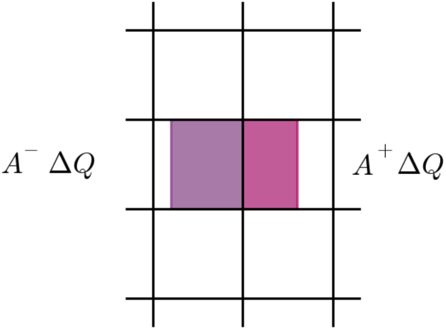
$$A = R\Lambda R^{-1} \implies A^- = R\Lambda^- R^{-1}, A^+ = R\Lambda^+ R^{-1}$$

- Transverse Riemann solver `rpt2.f`
  Decomposes $\mathcal{A}^+\Delta Q$ into $\mathcal{B}^-\mathcal{A}^+\Delta Q$ and $\mathcal{B}^+\mathcal{A}^+\Delta Q$ by splitting this vector into eigenvectors of B.

Wave propagation algorithm for $q_t + Aq_x + Bq_y = 0$

Decompose $A = A^+ + A^-$ and $B = B^+ + B^-$.

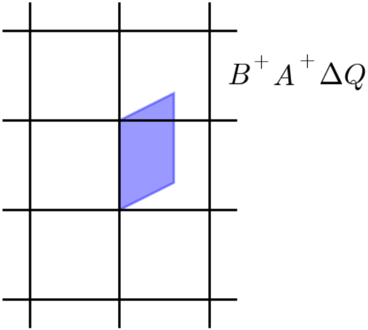
For $\Delta Q = Q_{ij} - Q_{i-1,j}$:



Wave propagation algorithm for $q_t + Aq_x + Bq_y = 0$

Decompose $A = A^+ + A^-$ and $B = B^+ + B^-$.

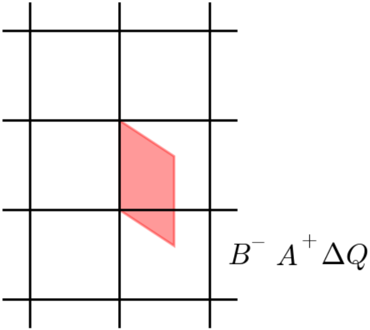
For $\Delta Q = Q_{ij} - Q_{i-1,j}$:



Wave propagation algorithm for $q_t + Aq_x + Bq_y = 0$

Decompose $A = A^+ + A^-$ and $B = B^+ + B^-$.

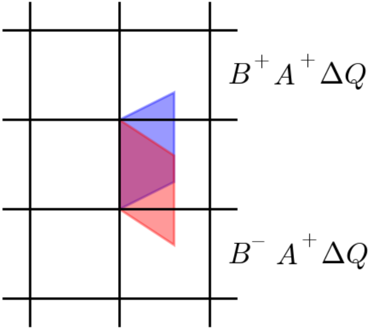
For $\Delta Q = Q_{ij} - Q_{i-1,j}$:



Wave propagation algorithm for $q_t + Aq_x + Bq_y = 0$

Decompose $A = A^+ + A^-$ and $B = B^+ + B^-$.

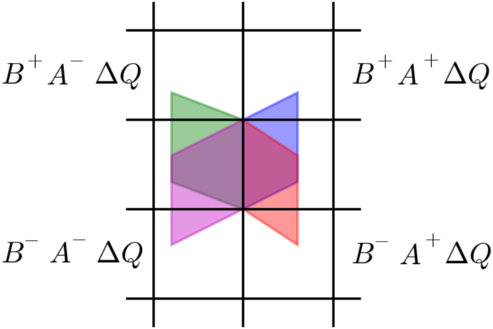
For $\Delta Q = Q_{ij} - Q_{i-1,j}$:



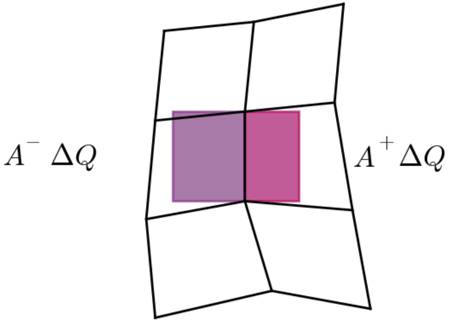
Wave propagation algorithm for $q_t + Aq_x + Bq_y = 0$

Decompose $A = A^+ + A^-$ and $B = B^+ + B^-$.

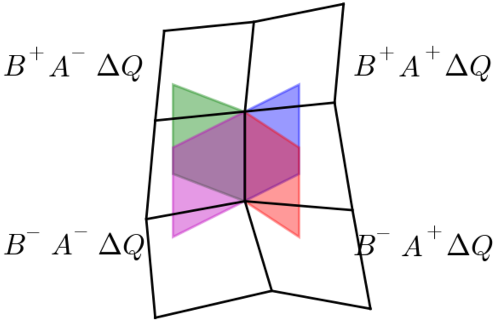
For $\Delta Q = Q_{ij} - Q_{i-1,j}$:



Wave propagation algorithm on a quadrilateral grid



Wave propagation algorithm on a quadrilateral grid



Shallow water algorithm on the sphere

Need to store depth*velocity values in some coordinate system.

Coordinate mapping is not smooth and inconvenient to use.

Shallow water algorithm on the sphere

Need to store depth*velocity values in some coordinate system.

Coordinate mapping is not smooth and inconvenient to use.

Instead we store 3 components of momentum in x–y–z Cartesian coordinates.

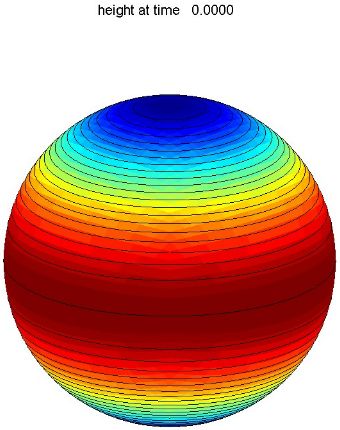
Velocity vector should be tangent to sphere.

Compute velocities normal to each cell interface, solve 1d normal and transverse Riemann problems.

Source term appears in equations to keep flow on the sphere.
Ref: J. Coté, F. Giraldo, T. Warburton, ...

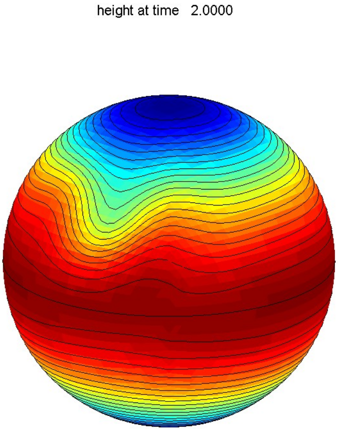
Rossby-Haurwitz test problem

Shallow water on rotating sphere with a mountain



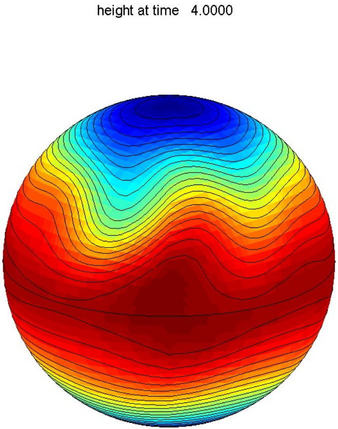
Rossby-Haurwitz test problem

Shallow water on rotating sphere with a mountain



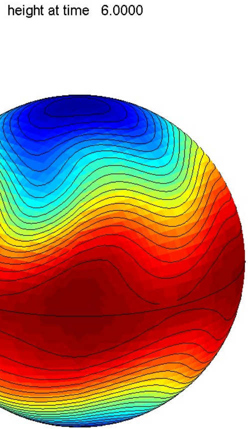
Rossby-Haurwitz test problem

Shallow water on rotating sphere with a mountain



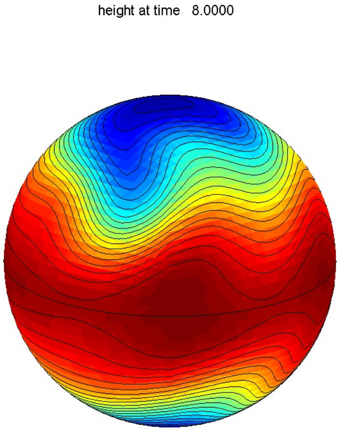
Rossby-Haurwitz test problem

Shallow water on rotating sphere with a mountain



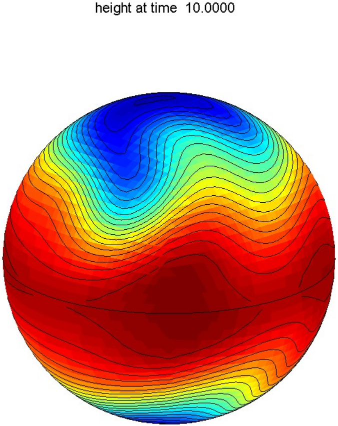
Rossby-Haurwitz test problem

Shallow water on rotating sphere with a mountain



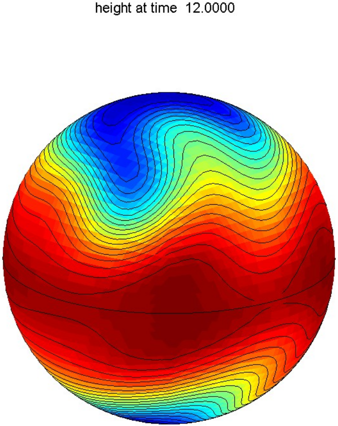
Rossby-Haurwitz test problem

Shallow water on rotating sphere with a mountain



Rossby-Haurwitz test problem

Shallow water on rotating sphere with a mountain



Rossby-Haurwitz test problem

Convergence study for Shallow water on rotating sphere
(flat topography)

Order of accuracy estimated using three grids with 100×50, 200×100 and 400×200 grid cells.

No limiters, Courant number ≈ 0.9.

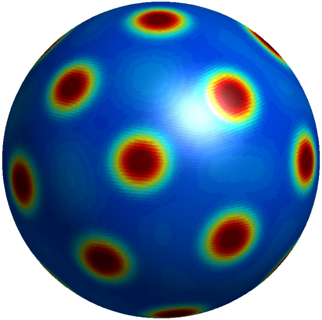
	1 day	2 days	3 days	4 days
h	1.64	1.73	1.89	2.04
hu	1.77	1.93	1.93	1.92
hv	1.77	1.93	1.95	1.90
hw	1.76	1.80	2.38	1.85

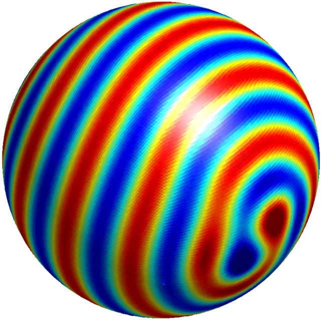
Pattern formation on the sphere

Turing model: Movie of pattern formation

$$\frac{\partial u}{\partial t} = D\delta\nabla^2_{\vec{n}}u + \alpha u\left(1 - r_1 v^2\right) + v(1 - r_2 u)$$

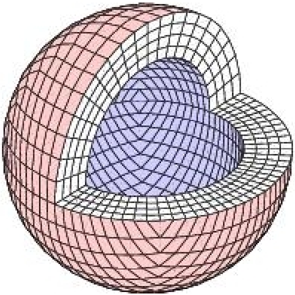
$$\frac{\partial v}{\partial t} = \delta\nabla^2_{\vec{n}}v + \beta v\left(1 + \frac{\alpha r_1}{\beta}uv\right) + u(\gamma + r_2 v).$$





Our approach for shells

Above approach can be used on sphere and then extended radially:

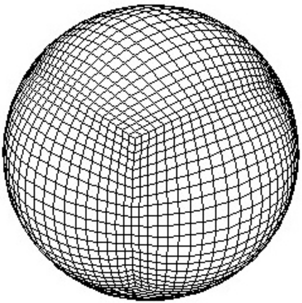


Useful for atmosphere, mantle convection, etc.

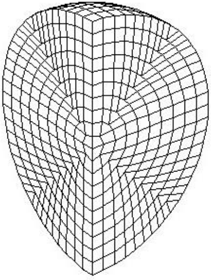
Wouldn't want to extend into origin for full ball —
radial lines meet at center and give small cells.

3D hexahedral grid in the ball

Three-dimensional version of the circle mappings

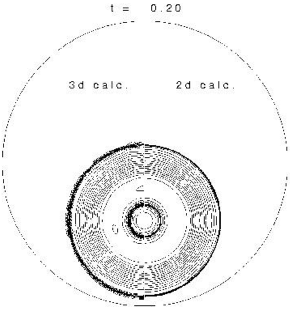


Full grid



Quarter Section

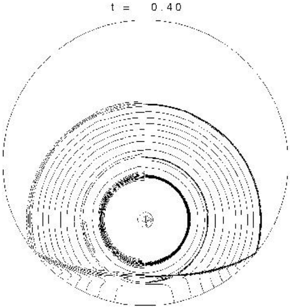
Euler equations in the ball



Left: cross section of $75 \times 75 \times 150$ grid in 3d

Right: 300×600 grid in 2d (axi-symmetric)

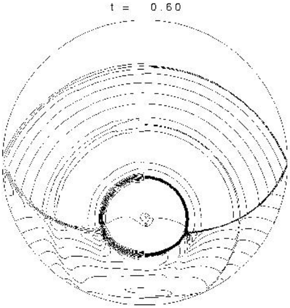
Euler equations in the ball



Left: cross section of $75 \times 75 \times 150$ grid in 3d

Right: 300×600 grid in 2d (axi-symmetric)

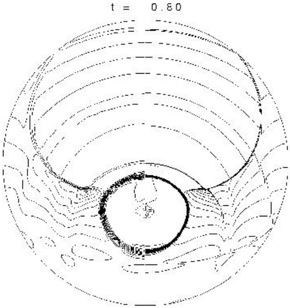
Euler equations in the ball



Left: cross section of $75 \times 75 \times 150$ grid in 3d

Right: 300×600 grid in 2d (axi-symmetric)

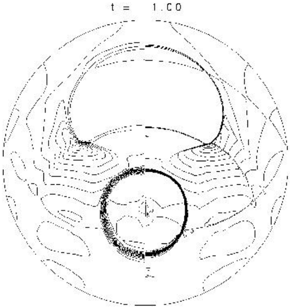
Euler equations in the ball



Left: cross section of $75 \times 75 \times 150$ grid in 3d

Right: 300×600 grid in 2d (axi-symmetric)

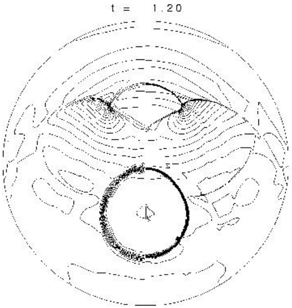
Euler equations in the ball



Left: cross section of $75 \times 75 \times 150$ grid in 3d

Right: 300×600 grid in 2d (axi-symmetric)

Euler equations in the ball

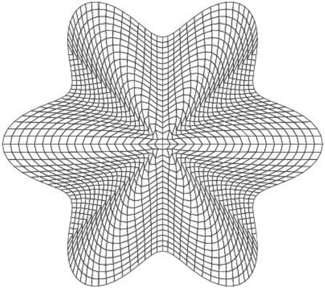


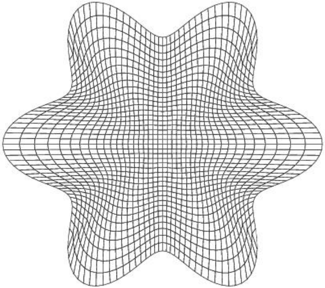
Left: cross section of $75 \times 75 \times 150$ grid in 3d

Right: 300×600 grid in 2d (axi-symmetric)

Mappings to circle or sphere can be composed with other smooth mappings.

Example: Shallow water equations in starlike domain:





Movie of radial dam break

References

Logically Rectangular Grids and Finite Volume Methods for PDEs in Circular and Spherical Domains,
by D. Calhoun, C. Helzel, and RJL, to appear in *SIAM Review*

http://www.amath.washington.edu/∼rjl/pubs/circles

Also contains

- m-files for mappings
- Clawpack codes for the examples

See also:

A finite-volume method for solving parabolic equations on logically Cartesian curved surface meshes,
by D. Calhoun and C. Helzel

http://www.amath.washington.edu/∼calhoun/Surfaces