Solving the Global Shallow Water Equations in Vorticity-Divergence Form with Element-Based Galerkin Methods

Michael Levy

University of Colorado at Boulder
Department of Applied Mathematics

NCAR Theme of the Year (2009)
Aug 18, 2009

Outline

Purpose of this Project

Focus of this thesis

Model the shallow water equations (SWE) on a sphere.

Why?

1. Physics
   - Model global horizontal dynamics of the atmosphere
   - Provide testbed for fully 3D models (time stepping, etc)
2. Computational Expense
   - Cheaper to test atmospheric core on 2D model such as SWE

Measures of success

1. 1992: Williamson *et al.* compiled a collection of 7 tests from literature (look at SWTC2 and SWTC5)
2. 2004: Galewsky *et al.* introduced additional test

Three Big Choices

1. What formulation of the SWE should be used?
2. What numerical method should be used?
3. How should the sphere be discretized?

(1) What formulation of SWE to use

Vorticity - Divergence form

Pros Small adjustment to Potential Vorticity (PV) - Divergence form, can ensure conservation of PV

Cons Requires solving three auxiliary equations, can add computational expense

Other options:

- flux form
- advective form
- stream function - velocity potential form

Three Big Choices

1. What formulation of the SWE should be used?
2. What numerical method should be used?
3. How should the sphere be discretized?

(2) What method to use

Element-based Galerkin (EBG) Method

Pros Mostly local, so great for parallelization; many EBG methods are inherently conservative; high-order accurate

Cons Expensive to implement serially

Other options:

- spectral methods
- finite difference methods
- finite volume methods

Three Big Choices

1. What formulation of the SWE should be used?
2. What numerical method should be used?
3. How should the sphere be discretized?

(3) How to discretize the sphere

Cubed Sphere

Pros No issues at poles; roughly uniform-sized rectangular elements

Cons Numerical noise along cube edges

Other options

- regular latitude-longitude grid
- geodesic grid
- icosahedral grid

Why is conservation so important?

Conservation in general

- On small time scales (e.g. weather forecasting model), a slight loss in mass or PV is not awful
- Full climate models simulate decades or centuries
  - small changes add up
  - forecast that says "atmosphere won't have mass in 300 years" is not useful

PV as a prognostic variable

- In 1939, Rossby pointed out that absolute vorticity is conserved in 2D flows; the next year, he realized that PV is as well
- In 2D models, PV can be inverted to find wind field
  - 3D models, PV contains information about wind, pressure, and temperature fields
- More info in Hoskins *et al.* (1985)

Why DG and SEM?

Discontinuous Galerkin

- Locally conservative, great for physically conserved quantities (mass, PV, etc)
- Mostly local method, good candidate for parallelization
- High-order accurate

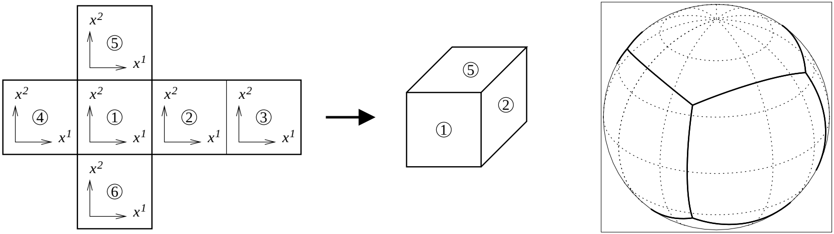
Spectral Element

- Same computational domain as DG
- Similar finite-dimensional function space as DG (see below)
- Easier to implement than DG

Note

In this implementation, the only difference between DG and SEM is in the treatment of the edges of each element - DG allows functions to be discontinuous at element edges, SEM enforces continuity.

Why the Cubed Sphere?



Issues with typical lat-lon grid

- Polar singularities
  - Mesh converges at the poles!
  - Leads to stability issues in polar region
  - Avoided with polar filtering, which is non-local (bad for parallel computation)

Solver Details

About the DG and SEM discretizations

- GLL grid used for quadrature on each element
- Basis functions: tensor products of Lagrange polynomials
- SEM requires continuity over element boundaries, DG uses flux formula

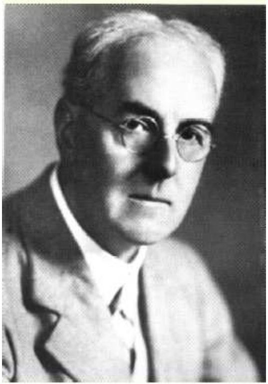
Time Stepping (for advection)

- Explicit 3rd-order TVD Runge-Kutta scheme used for BVE
- Reduced to 2nd-order RK for SWE

Solving linear system resulting from SEM

- Conjugate Gradient used initially
- Now that physical model works, need to replace with something more efficient (multigrid)

History of Numerical Weather Prediction



Weather Prediction by Numerical Process, 1922

"Perhaps some day in the dim future it will be possible to advance the computations faster than the weather advances.... But that is a dream." – Lewis Fry Richardson

First Step Towards Richardson's Dream

$$\frac{\partial \eta}{\partial t} + \nabla_H \cdot [\mathbf{v}\eta] = 0$$

- 1950: Charney, Fjörtoft, and von Neumann numerically solved Barotropic Vorticity Equation (BVE)
  - First successful numerical weather prediction algorithm
  - Took 24 hours of "compute time" on ENIAC to simulate 24 hours of atmospheric behavior

More on BVE

- Simple model for vortex dynamics
- Can be thought of as simplification of SWE in vorticity-divergence form
- Tested method described in this talk on BVE, will show results from one case

History of the Methods

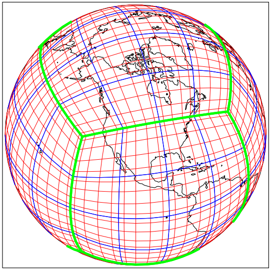
Discontinuous Galerkin

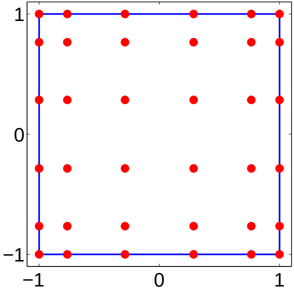
- Hybrid method, combines pieces of finite volume and finite element methods
- Introduced in technical paper out of Los Alamos: Reed and Hill, 1973 (neutron transport)
- Analysis of method done by Lesaint and Raviart, 1974
- Late '80s to Mid '90s: Series of papers by Cockburn and Shu combined DG with TVD Runge-Kutta time stepping

Spectral Element

- Introduced in Patera, 1984
  - Reviewed by Maday and Patera, 1989 (Incompressible Navier-Stokes)
- Similar to finite element methods available at the time, except high-order accurate

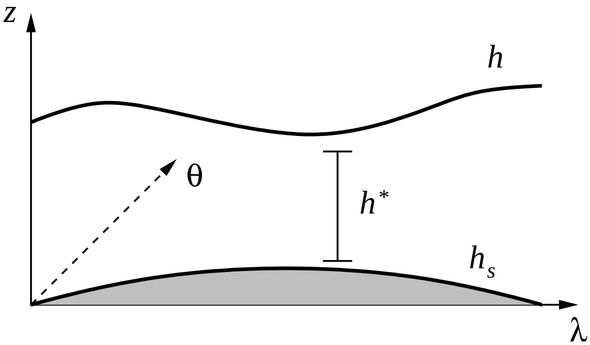
The Cubed Sphere Geometry





- Cube is inscribed in a sphere
  - Mapping between cube and sphere via gnomonic projection (ray from center of sphere to surface of sphere intersects cube)
  - No issue at poles, care needs to be taken at corners / edges
- Introduced by Sadourny in 1972
- Sat mostly dormant until 1996
  - Re-introduced independently by Rančić *et al.* and Ronchi *et al.*

The Shallow Water Equations



Variables

- (λ, θ) are coordinates on the sphere (lon-lat)
- $h(\lambda, \theta, t)$ is the height of the top of the fluid layer
- $h_S(\lambda, \theta)$ is the surface topography
- $h^* = h - h_S$ is the thickness of the fluid layer

The Shallow Water Equations

The shallow water system is based on conservation of mass and momentum – it combines the mass equation and the horizontal momentum equation:

$$\begin{aligned} \frac{\partial h^*}{\partial t} + \nabla_H \cdot (h^* \mathbf{v}) &= 0 \\ \frac{\partial \mathbf{v}}{\partial t} &= -\nabla_H (gh + \mathbf{v} \cdot \mathbf{v}/2) - ((\nabla_H \times \mathbf{v}) + f)\mathbf{v}^\perp \end{aligned}$$

Variables and Operators

- ∇_H is the horizontal gradient operator
- $\mathbf{v}(\lambda, \theta, t) = (u, v)^T$ is the horizontal wind
- $\mathbf{v}^\perp = (-v, u)^T$ is orthogonal to $\mathbf{v}$
- f is Coriolis parameter, g is gravitational constant

Vorticity - Divergence Form of Momentum Equation

Momentum Equation

$$\frac{\partial \mathbf{v}}{\partial t} = -\nabla_H (gh + \mathbf{v}\cdot\mathbf{v}/2) - \left((\nabla_H \times \mathbf{v}) + f\right)\mathbf{v}^{\perp}$$

- Taking the curl and divergence of the momentum equation yields

$$\begin{aligned} \frac{\partial \eta}{\partial t} + \nabla_H \cdot [\mathbf{v}\eta] &= 0 \\ \frac{\partial \delta}{\partial t} + \nabla_H^2 (gh + \mathbf{v}\cdot\mathbf{v}/2) - \nabla_H \times [\mathbf{v}\eta] &= 0 \end{aligned}$$

Variables

- $\eta = \nabla_H \times \mathbf{v} + f$ is absolute vorticity
- $\delta = \nabla_H \cdot \mathbf{v}$ is divergence

Full Shallow Water System

Vorticity-Divergence formulation of SWE consists of

- The mass equation,
- the momentum equations in η - δ form, and
- three auxilary equations to recover $\mathbf{v}$.

$$
\begin{aligned}
\frac{\partial h^*}{\partial t} + \nabla_H \cdot \left[\mathbf{v} h^*\right] &= 0 \\
\frac{\partial \eta}{\partial t} + \nabla_H \cdot \left[\mathbf{v} \eta\right] &= 0 \\
\frac{\partial \delta}{\partial t} + \nabla_H^2 \left[gh + \mathbf{v} \cdot \mathbf{v}/2\right] - \nabla_H \times \left[\mathbf{v} \eta\right] &= 0 \\
-\nabla_H^2 \psi &= -\zeta \\
-\nabla_H^2 \chi &= -\delta \\
\mathbf{v} &= \nabla_H^{\perp} \psi + \nabla_H \chi
\end{aligned}
$$

Steps Necessary to Solve Shallow Water Equations

Given initial $\mathbf{v}$, η and δ can be calculated (analytically) by

$$\eta = \nabla_H \times \mathbf{v} + f \qquad \delta = \nabla_H \cdot \mathbf{v}$$

1. Keep $\mathbf{v}$ constant, advance through one stage of the RK time step in advection solver* to find new values of h^*, η, and δ:

$$\frac{\partial h^*}{\partial t} + \nabla_H \cdot (h^* \mathbf{v}) = 0$$
$$\frac{\partial \eta}{\partial t} + \nabla_H \cdot (\eta \mathbf{v}) = 0$$
$$\frac{\partial \delta}{\partial t} + \nabla_H^2 [gh + \mathbf{v} \cdot \mathbf{v}/2] - \nabla_H \times (\eta \mathbf{v}) = 0$$

** DG for h^* and η, SEM for δ*

Steps Necessary to Solve Shallow Water Equations

2. Given the new η and δ values, the Poisson solver is used to update the stream function and velocity potential

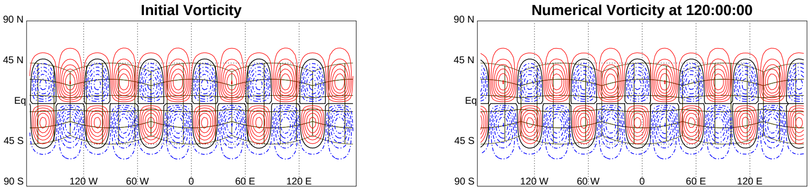
$$-\nabla_H^2\psi = -(\eta - f) \qquad -\nabla_H^2\chi = -\delta$$

3. With ψ and χ known at the new RK stage, **v** is updated from the relation

$$\mathbf{v} = \nabla_H^{\perp}\psi + \nabla_H\chi$$

4. **v** is again held constant, and steps 1-3 are repeated until the desired time has passed (filtering as needed).

BVE Results



Test Case: Wave Propogation

- Wave 6 propagating eastward 20° per day
- Test taken from Gates and Riegel (1962)

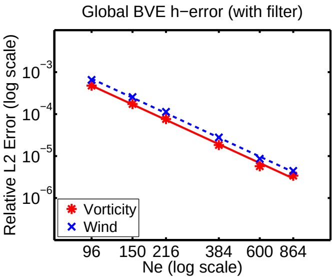
Solution given in terms of stream function. At time t,

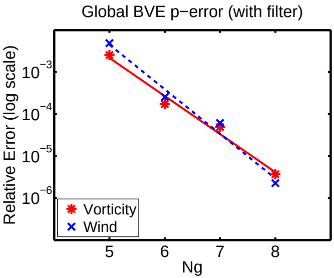
$$\psi(\lambda, \theta, t) = A \sin(m\lambda - \nu t) L_n^m(\sin(\theta)) - Ba^2 \sin\theta + CL_n(\sin\theta).$$

Note that η and $\mathbf{v}$ can be calculated directly from ψ

Vorticity animation

BVE Results





h-error

Measured by leaving the number of nodes per element constant but increasing the number of elements.

p-error

Measured by leaving the number of elements constant but increasing the number of nodes per element.

Shallow Water Results – SWTC2

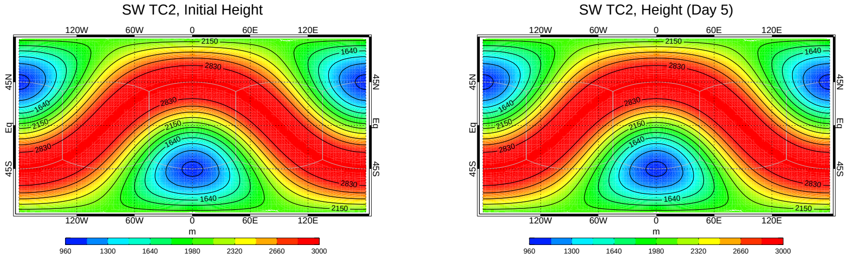
Test Case: Stable Steady State

- Test case #2 from Williamson *et al.* (1992)
- Vorticity and height fields are balanced, shouldn't change
- $h_S = 0$ (no specified topography)
- $\alpha = \pi/4$ is the angle between axis of rotation and polar axis
- Compare final state to initial condition for error analysis

Initial (steady) state given by

$$\begin{aligned} h^*(\lambda,\theta) &= h_0 - \frac{1}{g}\left(a\Omega u_0 + \frac{{u_0}^2}{2}\right)(\cos\lambda\cos\theta\sin\alpha + \sin\theta\cos\alpha) \\ \eta(\lambda,\theta) &= \left(\frac{2u_0}{a} + 2\Omega\right)(-\cos\lambda\cos\theta\sin\alpha + \sin\theta\cos\alpha) \end{aligned}$$

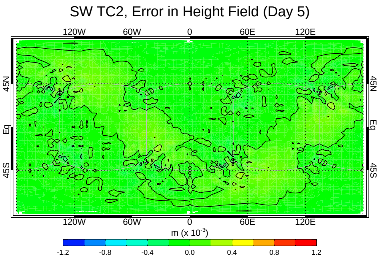
Shallow Water Results – SWTC2



Notes

- Contours Range from 960 to 3000 (units = m)
- Results using 96 elements and an 8 x 8 GLL grid.

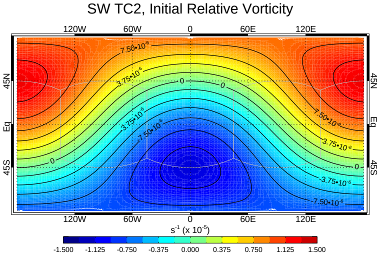
Shallow Water Results – SWTC2

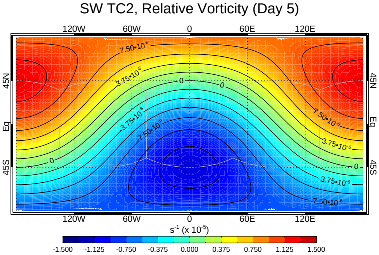


Notes

- Contours Range from $-1.2 \cdot 10^{-3}$ to $1.2 \cdot 10^{-3}$ (units $=$ m)
- Results using 96 elements and an 8 x 8 GLL grid.

Shallow Water Results – SWTC2

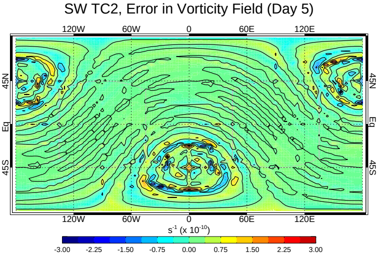




Notes

- Contours Range from $-1.5 \cdot 10^{-5}$ to $1.5 \cdot 10^{-5}$ (units $= s^{-1}$)
- Results using 96 elements and an 8 x 8 GLL grid.

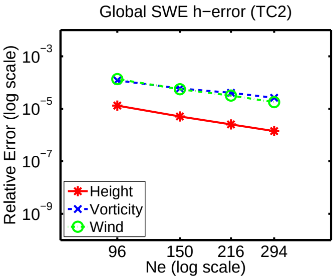
Shallow Water Results – SWTC2

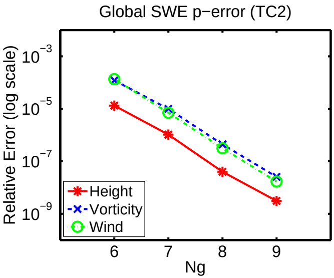


Notes

- Contours Range from $-3 \cdot 10^{-10}$ to $3 \cdot 10^{-10}$ (units $= s^{-1}$)
- Results using 96 elements and an 8 x 8 GLL grid.

Shallow Water Results – SWTC2





h-error

Measured by leaving the number of nodes per element constant but increasing the number of elements.

p-error

Measured by leaving the number of elements constant but increasing the number of nodes per element.

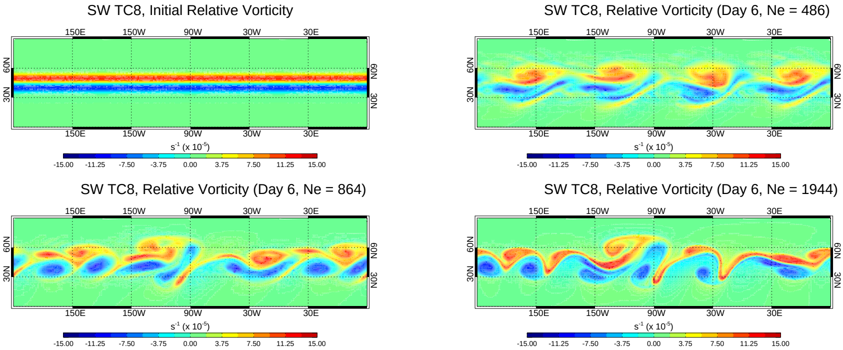
Shallow Water Results – SWTC8

Test Case: Barotropic Instability

- Test taken from Galewsky *et al.* (2004)
- Highlights issue with Williamson test suite
  - Mountain in TC #5 is not differentiable (leads to spectral ringing, small time step at beginning of test case, etc)
- Zonal jet between θ_0 and θ_1 (in N. hemisphere)
- Small perturbation to h^* in middle of jet
- Interesting on cubed sphere because of jet location

$$
\begin{aligned}
u(\lambda,\theta) &= \begin{cases} \frac{u_{\max}}{e_n}\exp\left[\frac{1}{(\theta-\theta_0)(\theta-\theta_1)}\right] & \theta_0<\theta<\theta_1 \\ 0 & \text{otherwise} \end{cases} \\
h^*(\lambda,\theta) &= h_0 - \frac{1}{g}\int_{-\pi/2}^{\theta}[afu(\phi)+\tan(\phi)u^2(\phi)]d\phi + h'(\lambda,\theta)
\end{aligned}
$$

Shallow Water Results – SWTC8



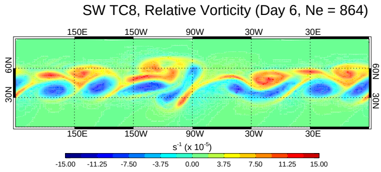
Results using varying number of elements and a 6 x 6 GLL grid

What's happening?

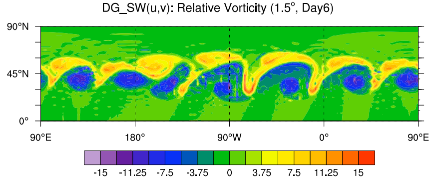
Numerical error from grid dominates the low-resolution solution

Animation (1944 Elements, 6 $\times$ 6 GLL)

Shallow Water Results – Comparing to a Different Model



864 elements, 6×6 GLL grid using the method discussed here

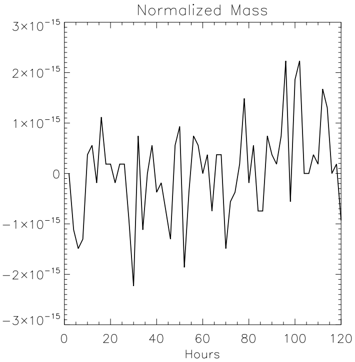


864 elements, 6×6 GLL grid using the advective form of the SWE (on cubed sphere)

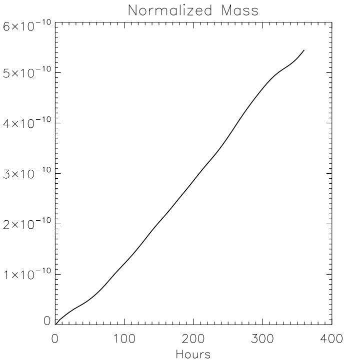
About these plots

As previously mentioned, numerical error from the grid pollutes low resolution tests. Here a grid resolution of $\sim 1.5^\circ$ captures the instability in the η-δ formulation but not in the advective formulation.

Conservation Results – η-δ Formulation



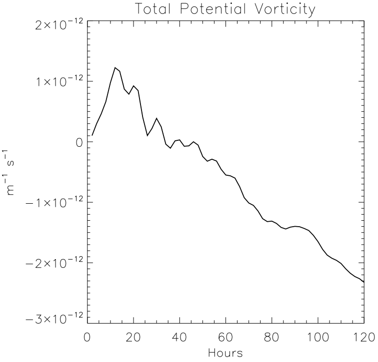
Change in mass, SWTC2



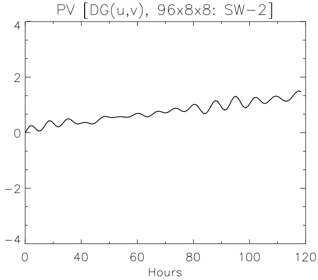
Change in mass, SWTC5

Mass is conserved to machine precision in the steady state test, but not in either of the other tests – indicates need to move to fully DG method.

Comparing PV and Energy Conservation



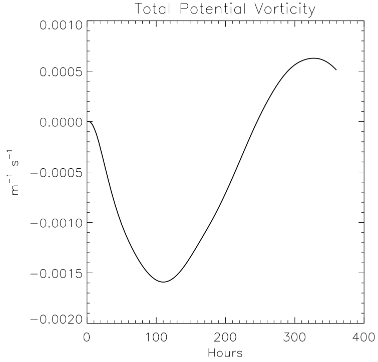
Vort-Div model



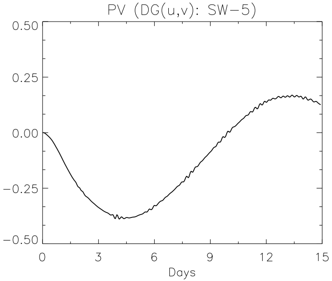
Advective Model

Potential Vorticity for SWTC2

Comparing PV and Energy Conservation



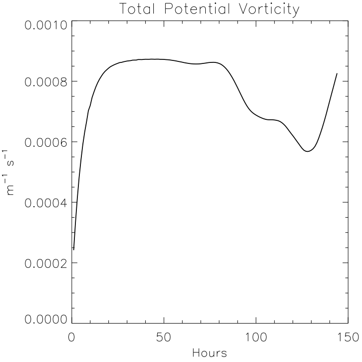
Vort-Div model



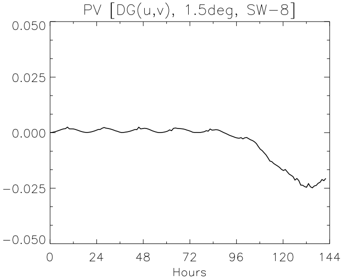
Advective Model

Potential Vorticity for SWTC5

Comparing PV and Energy Conservation



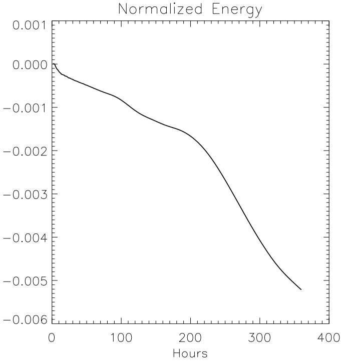
Vort-Div model



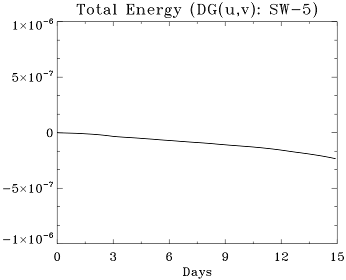
Advective Model

Potential Vorticity for SWTC8

Comparing PV and Energy Conservation



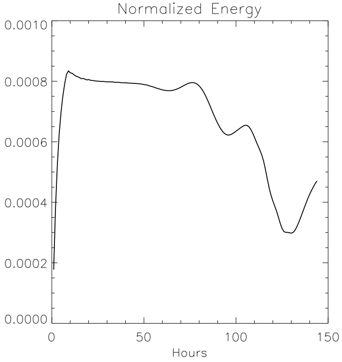
Vort-Div model



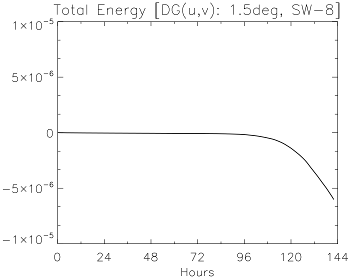
Advective Model

Energy for SWTC5

Comparing PV and Energy Conservation



Vort-Div model



Advective Model

Energy for SWTC8

The Elephant in the Room

Model portrays physical conditions accurately, but...

The choice of the conjugate gradient method to solve the discrete Poisson problem is highly in-efficient.

N_g	# of iterations	
	$N_e = 96$	$N_e = 384$
3	81	166
4	135	272
5	192	385
6	254	505

The number of CG iterations needed to reduce the residual in the global Poisson system by a factor of 10^{-14}.

- Note that the iteration count increases with both N_e and N_g, which is bad news for high resolution runs.

The Elephant in the Room

Effect of this cost on the Shallow Water model

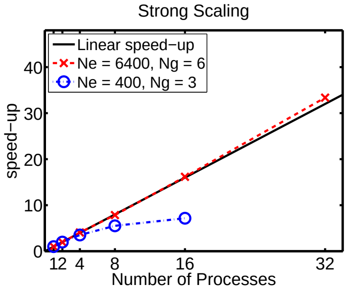
As the grid increases in size, a larger percentage of time is spent solving the Poisson equations.

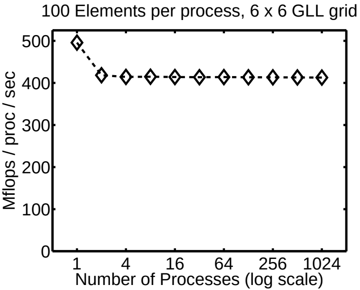
N_e	N_g	Advection	Poisson	Helmholtz	Filtering
96	6	0.34	98.82	0.10	0.75
216	6	0.25	99.19	0.07	0.49
96	8	0.13	99.40	0.03	0.44
216	8	0.07	99.71	0.02	0.21

Percentage of time spent in each of four phases of the solver for SWTC2.

- For the more complex tests, even more time is spent in at the Poisson step.

Towards Efficient Parallelization - 2D DG Advection





Strong scaling

Measured by increasing the number of processes running while keeping the problem size constant.

Weak scaling

Measured by scaling the problem along with the number of processes, so that work per process is constant.

Future Work

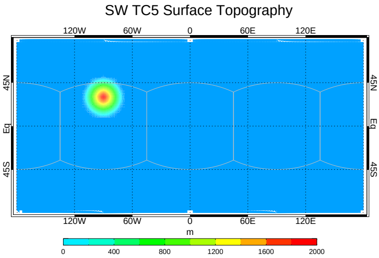
Four big goals for this project

1. Improve computational efficiency
    - First step: replace the CG iterations
    - Parallelization: EBG methods are great on distributed memory machines
2. Maintain conservation (get rid of SEM discretization)
3. Move from vorticity-divergence to PV-divergence
4. Expand to 3D model
    - Stratified atmosphere, each layer is shallow water model

Shallow Water Results – SWTC5

Test Case: Zonal Flow over a Mountain

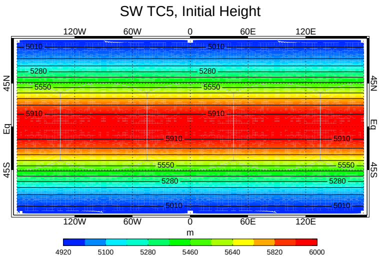
- Test case #5 from Williamson *et al.* (1992)
- Same initial conditions as TC 2 ($\alpha = 0$)
- Single mountain located in northern hemisphere

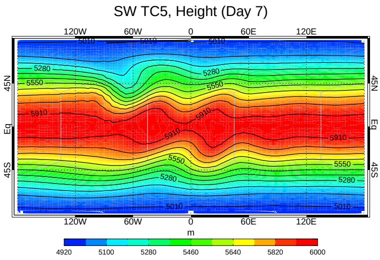


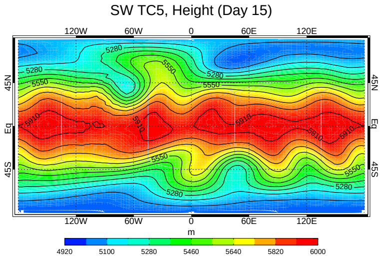
$$h_S(\lambda, \theta) = h_{S_0}(1 - r/R),$$

where $R = \pi/9$ and $r = \min(R, \sqrt{(\lambda - \lambda_c)^2 + (\theta - \theta_c)^2})$

Shallow Water Results – SWTC5

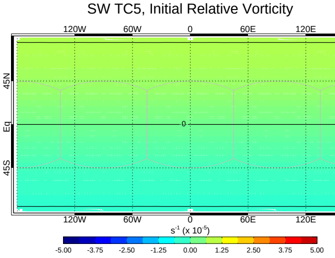


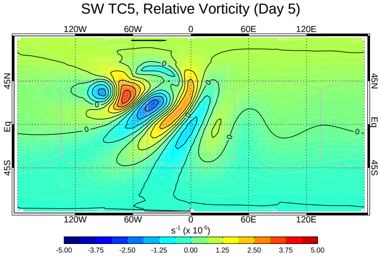


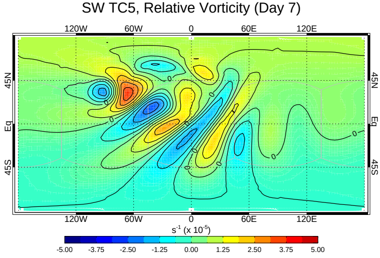


Results using 96 elements and an 8 x 8 GLL grid (Animation)

Shallow Water Results – SWTC5







Results using 96 elements and an 8 x 8 GLL grid (Animation)