FV³

The GFDL Finite-Volume Cubed-sphere Dynamical Core

Lucas Harris and Shian-Jiann Lin
NOAA/Geophysical Fluid Dynamics Laboratory

27 January 2016
FV$^3$

- Scalable, flexible dynamical core capable of both hydrostatic and nonhydrostatic simulation
- Successor to latitude-longitude FV core in NASA GEOS, GFDL AM2.1, and CAM-FV

- GFDL models
  - AM4/CM4
  - HiRAM
  - CM2.5/2.6/3

- CAM-FV$^3$
  - LASG
  - GEOS-CHEM (coming soon!)
  - GISS ModelE
FV$^3$ Design Philosophy

- Discretization should be guided by physical principles as much as possible
  - Finite-volume, integrated form of conservation laws
  - Upstream-biased fluxes

- Operators “reverse engineered” to achieve desired properties

- Computational efficiency is crucial. A fast model is a good model!
  - Solver should be built with vectorization and parallelism in mind

- Dynamics isn’t the whole story! Coupling to physics and the ocean is important.
Development of the FV$^3$ core

- Lin and Rood (1997, QJ): FV shallow-water solver
- Lin (1997, QJ): FV Pressure Gradient Force
- Putman and Lin (2007, JCP): Cubed-sphere advection
- Harris and Lin (2013, MWR): Describes FV$^3$ and grid nesting
- Lin (in prep): Nonhydrostatic dynamics
Development of the FV$^3$ core

- Lin and Rood (1997, QJ): FV shallow-water solver
- Lin (1997, QJ): FV Pressure Gradient Force
- Putman and Lin (2007, JCP): Cubed-sphere advection
- Harris and Lin (2013, MWR): Describes FV$^3$ and grid nesting
- Lin (in prep): Nonhydrostatic dynamics
Flux-form advection scheme

\[ q^{n+1} = \frac{1}{\pi^{n+1}} \left\{ \pi^n q^n + F \left[ q^n + \frac{1}{2} g(q^n) \right] + G \left[ q^n + \frac{1}{2} f(q^n) \right] \right\} . \]

- 2D scheme derived from 1D PPM operators
- Advective form inner operators f, g, allow elimination of leading-order deformation error
  - Allows preservation of constant tracer field under nondivergent flow
- Flux-form outer operators F, G ensure mass conservation
Lin and Rood (1996, MWR)
Flux-form advection scheme

- PPM operators are upwind biased
  - More physical, but also more diffusive

- Monotonicity/positivity constraint: important (implicit) source of model diffusion and noise control
  - Also available: linear advection schemes with a selective filter to suppress $2\Delta x$ noise. These can be useful in very high-res nonhydrostatic runs

- Scheme maintains linear correlations between tracers when unlimited or when monotonicity constraint applied (not necessarily so for positivity)
1D Advection Test

- 4th order centered
- 3rd order SL
- FV Monotone
- FV Positive

Lin and Rood 1996, MWR
Development of the FV$^3$ core

- **Lin and Rood (1997, QJ): FV shallow-water solver**
- Lin (1997, QJ): FV Pressure Gradient Force
- Putman and Lin (2007, JCP): Cubed-sphere advection
- Harris and Lin (2013, MWR): Describes FV$^3$ and grid nesting
- Lin (in prep): Nonhydrostatic dynamics
Lin and Rood (1997, QJ)

FV shallow-water solver

- Solves layer-averaged vector-invariant equations
- \( \delta p \) is proportional to layer mass
- \( \theta \): not in SW solver but is in full 3D Solver
- Forward-backward timestepping
  - PGF evaluated backward with updated pressure and height

\[
\begin{align*}
\frac{\partial \delta p}{\partial t} + \nabla \cdot (\mathbf{V} \delta p) &= 0 \\
\frac{\partial \delta p \theta}{\partial t} + \nabla \cdot (\mathbf{V} \delta p \theta) &= 0
\end{align*}
\]

\[
\frac{\partial \mathbf{V}}{\partial t} = -\Omega \hat{k} \times \mathbf{V} - \nabla (\kappa + \nu \nabla^2 D) - \frac{1}{\rho} \nabla p|_z
\]
Lin and Rood (1997, QJ)
FV shallow-water solver

• Discretization on D-grid, with C-grid winds used to compute fluxes

• D-grid winds interpolated to get C-grid winds, which are stepped forward a half-step for an approx. to time-centered winds—a simplified Riemann solver

• Advantages of D-grid are preserved, and diffusion due to C-grid averaging is alleviated

• Two-grid discretization and time-centered fluxes avoid computational modes

\[ \frac{\partial \delta p}{\partial t} + \nabla \cdot (\mathbf{V} \delta p) = 0 \]
\[ \frac{\partial \delta p \Theta}{\partial t} + \nabla \cdot (\mathbf{V} \delta p \Theta) = 0 \]
\[ \frac{\partial \mathbf{V}}{\partial t} = -\Omega \hat{k} \times \mathbf{V} - \nabla \left( \kappa + \nu \nabla^2 D \right) - \frac{1}{\rho} \nabla p \bigg|_z \]
FV shallow-water solver:
Time-stepping procedure

• Interpolate time $t^n$ D-grid winds to C-grid

• Advance C-grid winds by one-half timestep to time $t^{n+1/2}$
  • Used as approximation to time-averaged winds for time-averaged fluxes

• Use time-averaged air mass fluxes to update $\delta p$ and $\theta$ to time $t^{n+1}$

• Compute vorticity flux and KE gradient to update D-grid winds to time $t^{n+1}$

• Use time $t^{n+1}$ $\delta p$ and $\theta$ to compute PGF to complete D-grid wind update
FV shallow-water solver: Vorticity flux

- Nonlinear vorticity flux term in momentum equation
- D-grid allows exact computation of absolute vorticity—no averaging!
- Advantages to this form not apparent in linear analyses
FV shallow-water solver: Vorticity flux

- Vorticity uses same flux as \( \delta p \)

- Consistent flux of mass and vorticity improves preservation of geostrophic balance.

- Consistent flux also means PV is advected as a scalar!

- PV is thus **conserved** in adiabatic shallow-water flow.
FV shallow-water solver: Kinetic Energy Gradient

- Vector-invariant equations susceptible to Hollingsworth-Kallberg instability if KE gradient not consistent with vorticity flux

- Solution: use C-grid fluxes again to advect wind components, yielding an upstream-biased kinetic energy

\[ \kappa^* = \frac{1}{2} \left\{ \mathcal{X}(u^\theta, \Delta t; u^n) + \mathcal{Y}(v^\lambda, \Delta t; v^n) \right\} . \]

- Consistent advection again!
FV shallow-water: Polar vortex test

• Note how well strong PV gradients are maintained

• Vorticity isn’t just important for large-scale flow. Many mesoscale flows are also governed by vorticity too!

Figure 10. Polar stereographic projection (from the equator to the north pole) of the potential vorticity contours at DAY-24 in the ‘stratospheric vortex erosion’ test case at three different resolutions.
Development of the FV$^3$ core

- Lin and Rood (1997, QJ): FV shallow-water solver
- Lin (1997, QJ): FV Pressure Gradient Force
- Putman and Lin (2007, JCP): Cubed-sphere advection
- Harris and Lin (2013, MWR): Describes FV$^3$ and grid nesting
- Lin (in prep): Nonhydrostatic dynamics
Lin (1997, QJ)

Finite-Volume Pressure Gradient Force

- Computed from Newton’s laws and Green’s Theorem

- The pressure force on one cell from its neighbor is equal and opposite to that exerted by the cell on its neighbor

- Momentum is conserved the same way finite-volume algorithms conserve mass
Lin (1997, QJ)
Finite-Volume Pressure Gradient Force

- Errors lower, with much less noise, compared to a finite-difference pressure gradient evaluation
- Linear line-integral evaluation used in example yields larger errors near model top
- Now using fourth-order scheme to evaluate line integrals

Figure 6. As in Fig. 5, but for the finite-volume method.
DCMIP 2012

Resting atmosphere test

- DCMIP Test 2-0-0

- 15 years later: same great results!

- Compare to other DCMIP participants (links to DCMIP website):
  - CAM-FV (lat-lon FV core)
  - CAM-SE
  - UKMO ENDGAME
  - ICON IAP
  - ICON MPI DWD
  - MPAS
Development of the $FV^3$ core

- Lin and Rood (1997, QJ): FV shallow-water solver
- Lin (1997, QJ): FV Pressure Gradient Force
- Putman and Lin (2007, JCP): Cubed-sphere advection
- Harris and Lin (2013, MWR): Describes $FV^3$ and grid nesting
- Lin (in prep): Nonhydrostatic dynamics
Lin (2004, MWR)

Vertically-Lagrangian Discretization

- Equations of motion are vertically integrated to yield a series of layers
- Each layer like shallow-water, except $\theta$ is active
- Layers deform freely while horizontal equations integrated
  - Only cross-layer interaction here is through pressure force
- To perform vertical transport, and to avoid layers from becoming infinitesimally thin, we periodically remap to an Eulerian vertical coordinate
Vertical remapping

- Reconstructions by a cubic spline for remapping accuracy
  - Implicit in vertical, so no message passing
- Remapping conserves mass, momentum, and geopotential
  - Option to apply an energy fixer
- Vertical remapping is computationally expensive, but only needs to be done a few times an hour, not every time step
- As long as $\delta p > 0$, we retain stability. No vertical courant number limitation! This becomes critically important in nonhydrostatic simulations.
FV³ and the GFDL models

• Terrain following pressure coordinate: \( p_k = a_k + b_k p_s \)

  • Other coordinates possible: hybrid-z, hybrid-isentropic

• Divergence damping: fourth-order damping now standard, with a sixth-order option

  • Hyperdiffusion on vorticity also available. Useful when using non-monotonic schemes in very high-resolution nonhydrostatic simulations

• Physics coupling is time-split
Development of the FV$^3$ core

- Lin and Rood (1997, QJ): FV shallow-water solver
- Lin (1997, QJ): FV Pressure Gradient Force
- Putman and Lin (2007, JCP): Cubed-sphere advection
- Harris and Lin (2013, MWR): Describes FV$^3$ and grid nesting
- Lin (in prep): Nonhydrostatic dynamics
Putman and Lin (2007, JCP)
Cubed-sphere advection

• Gnomonic cubed-sphere grid
  • Coordinates are great circles
  • Widest cell only $\sqrt{2}$ wider than narrowest
  • More uniform than conformal, elliptic, or spring-dynamics cubed spheres

• Tradeoff: coordinate is non-orthogonal
Putman and Lin (2007, JCP)

Non-orthogonal coordinate

• Gnomonic cubed-sphere is non-orthogonal

• Instead of using numerous metric terms, use covariant and contravariant winds

  • Solution winds are covariant

  • Advection is by contravariant winds

  • KE is product of the two
Cubed-sphere edge handling

• Fluxes need to be the same across edges to preserve mass-conservation

• Gnomonic cubed sphere has ‘kink’ in coordinates at edge

• Currently getting edge values through two-sided linear extrapolation

• More sophisticated edge handling in progress
Development of the FV$^3$ core

- Lin and Rood (1997, QJ): FV shallow-water solver
- Lin (1997, QJ): FV Pressure Gradient Force
- Putman and Lin (2007, JCP): Cubed-sphere advection
- Harris and Lin (2013, MWR): Describes FV$^3$ and grid nesting
- Lin (in prep): Nonhydrostatic dynamics
Nonhydrostatic \( FV^3 \)

- \( FV^3 \) does a great job for hydrostatic flows. How can we maintain the hydrostatic performance and still do a good job with nonhydrostatic dynamics?

- Introduce new prognostic variables: \( w \) and \( \delta z \) (height thickness of a layer). The pressure thickness is still hydrostatic pressure, and thereby mass

- Density is then easily computed:

\[
\rho = \frac{M}{V} = \frac{W}{gV} = \frac{\delta p \Delta A}{g\delta z \Delta A} = \frac{\delta p}{g\delta z}
\]

- Nonhydrostatic pressure is then diagnosed as a deviation (not small-amplitude!) from the hydrostatic pressure.
Nonhydrostatic FV$^3$

• FV$^3$ does a great job for hydrostatic flows. How can we maintain the hydrostatic performance and still do a good job with nonhydrostatic dynamics??

• Introduce new prognostic variables: $w$ and $\delta z$ (height thickness of a layer). The pressure thickness is still hydrostatic pressure, and thereby mass

  • Vertical velocity $w$ is the cell-mean value. Remember that vorticity is also a cell-mean value.

  • Helicity can be computed without averaging!
**FV³ nonhydrostatic solver:**
Time-stepping procedure

- Interpolate time $t^n$ D-grid winds to C-grid

- Advance C-grid winds by one-half timestep to time $t^{n+1/2}$

- Use time-averaged air mass fluxes to update $\delta p$ and $\theta$, **and to advect** $w$ **and** $\delta z$, **to time** $t^{n+1}$

- Compute vorticity flux and KE gradient to update D-grid winds to time $t^{n+1}$

- **Solve nonhydrostatic terms for** $w$ **and nonhydrostatic pressure perturbation using either a Riemann solver or a semi-implicit solver**

- Use time $t^{n+1}$ $\delta p$, $\delta z$, **and** $\theta$ to compute PGF to complete D-grid wind update
Nonhydrostatic FV$^3$: nonhydrostatic solvers

• Instead of using a time-split solver for the fast vertical waves, FV$^3$ presents two solvers for the nonhydrostatic terms:

1. Exact Riemann solver: solves for the Riemann invariants along the gravity wave characteristic curves. Highly accurate!

2. Semi-implicit solver: solves a vertically-tridiagonal system for the sound waves. Diffusivity in semi-implicit solver works to damp sound waves.

• Most simulations do very well with the semi-implicit solver. For very high-resolution simulations ($\Delta x < 1$ km) where the vertical Courant number is < 1, the Riemann solver may be more appropriate (and possibly faster).

• All FV$^3$ simulations for NGGPS use the semi-implicit solver
FV$^3$: Model design and model performance

- Scientific accuracy is very important. But performance considerations cannot be ignored.

- FV3 originally designed for 90's vector supercomputers: lots of concurrency with a minimum of copies and transposes.

- Shared-memory threading and distributed-memory decomposition work together—not against one another!
Stretched grid

- Deforms a global grid so that one face has a higher resolution than the others
- Conceptually straightforward: requires no changes to solver!
Stretched grid

• Smoothly deformed! Even continental-scale flows may see little effect from the refinement.

• Capable of extreme refinement (80x!!) for storm-scale simulations
Stretched grid

- Opposing face can become very coarse

- Scale-aware parameterizations? (Are these really so important?)
Grid nesting

- **Simultaneous** coupled, consistent global and regional solution. No waiting for a regional prediction!

- Different grids permit different parameterizations; **doesn’t need a “compromise” or scale-aware physics** for high-resolution region

- **Very flexible!** Currently only uses static nesting but moving nests are also possible
3:1 nested grid
Large nest for RCMs
Multiple nests
Telescoping nests
2:1 nested grid
Nest in stretched grid
Nesting methodology: boundary conditions

• Simple interpolation BC
  • Linearly interpolate all variables in time and space to fill nested-grid halo
  • Traditionally, time interpolation requires the coarse grid to be advanced before the nested grid

• Concurrent nesting: integrate coarse and nested grids simultaneously by extrapolating coarse-grid data in time to create nested-grid BC
  • For scalars: extrapolation is limited so that the BCs are positive-definite

• Nonhydrostatic nesting: use same solver to produce BCs for the (diagnosed) nonhydrostatic pressure perturbation. Treat $w$ and $\delta z$ as if they were the other variables
Nesting methodology: two-way update

- Simple averaged update
  - Cell average on scalars
  - In-line average for winds, to conserve vorticity
  - Averaging is more consistent with FV discretization than pointwise interpolation
Mass conservation two-way nesting

- Usually quite complicated: requires flux BCs, conserving updates, and precisely-aligned grids

- Update only winds and temperature; not $\delta p$ or $\delta z$

  - Two-way nesting overspecifies solution anyway

- **Very simple**: works regardless of BC and grid alignment

  - $\delta p$ is the vertical coordinate: need to remap the nested-grid data to the coarse-grid’s vertical coordinate

- Option: a “renormalization-conserving” means of updating tracers to the coarse grid while conserving tracer mass
Nonhydrostatic HiRAM
2013 Moore Outbreak
72-hour forecast
1.3 km nest