# The GFDL Finite-Volume Cubed-Sphere Dynamical Core

## Structure and Usage

Lucas Harris, Xi Chen, Linjiong Zhou, and Jan-Huey Chen
for the GFDL FV3 Team

UFS MRW Application Training
5 November 2020

## Lucas Harris & Xi Chen

- FV3 design & algorithm
- Essential namelist items

Arbor Chen, born 3 November 2020

## Linjiong Zhou

- Physics-dynamics coupling in FV3
- In-line GFDL Microphysics and FV3 Integrated Physics

## Jan-Huey Chen

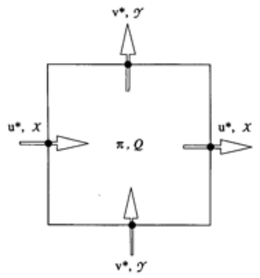
- The FV3 Community
- FV3 Applications

# FV3: The GFDL Finite-Volume Cubed-Sphere Dynamical Core

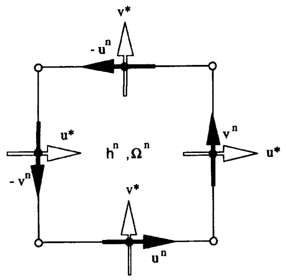
## The FV3 Way

- Physical consistency
- Fully-FV numerics
- Component coupling
- Computational efficiency

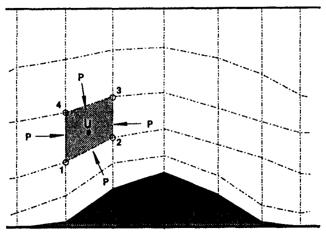
**Lin & Rood 1996**
Efficient 2D high-order conservative FV transport



**Lin & Rood 1997**
FV horizontal solver focusing on nonlinear vorticity dynamics

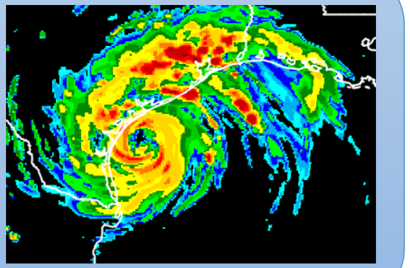


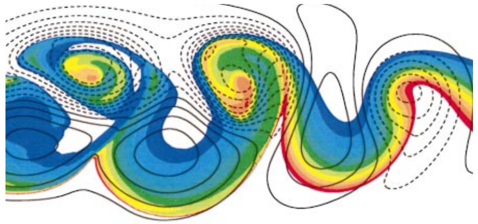
**Lin 1997** Efficient, mimetic FV PGF



## FV3 for the 2020s

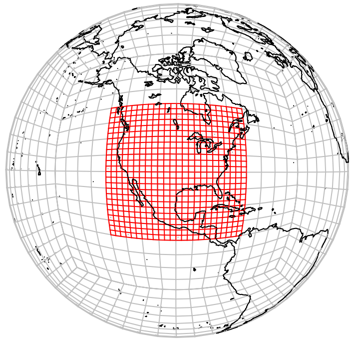
Rigorous Thermodynamics
Flexible dynamics
Adaptable physics interface
Variable-resolution techniques
Regional & periodic domains
Powerful initialization, DA,
and nudging functions

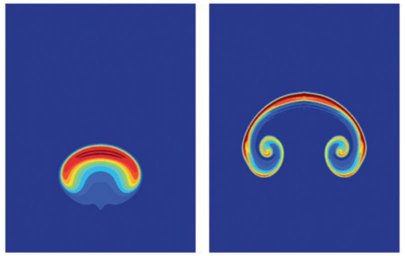




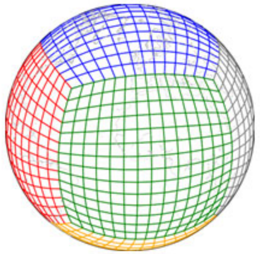
**Lin 1998–2004** FV core with “floating” Lagrangian vertical coordinate

**Harris & Lin 2013, 2016**
Variable resolution with two-way nesting and Schmidt grid stretching





**Lin 2006, X Chen & Lin et al 2013**
Consistent Lagrangian nonhydrostatic dynamics



**Putman & Lin 2007**
Scalable cubed-sphere grid, doubly-periodic domain

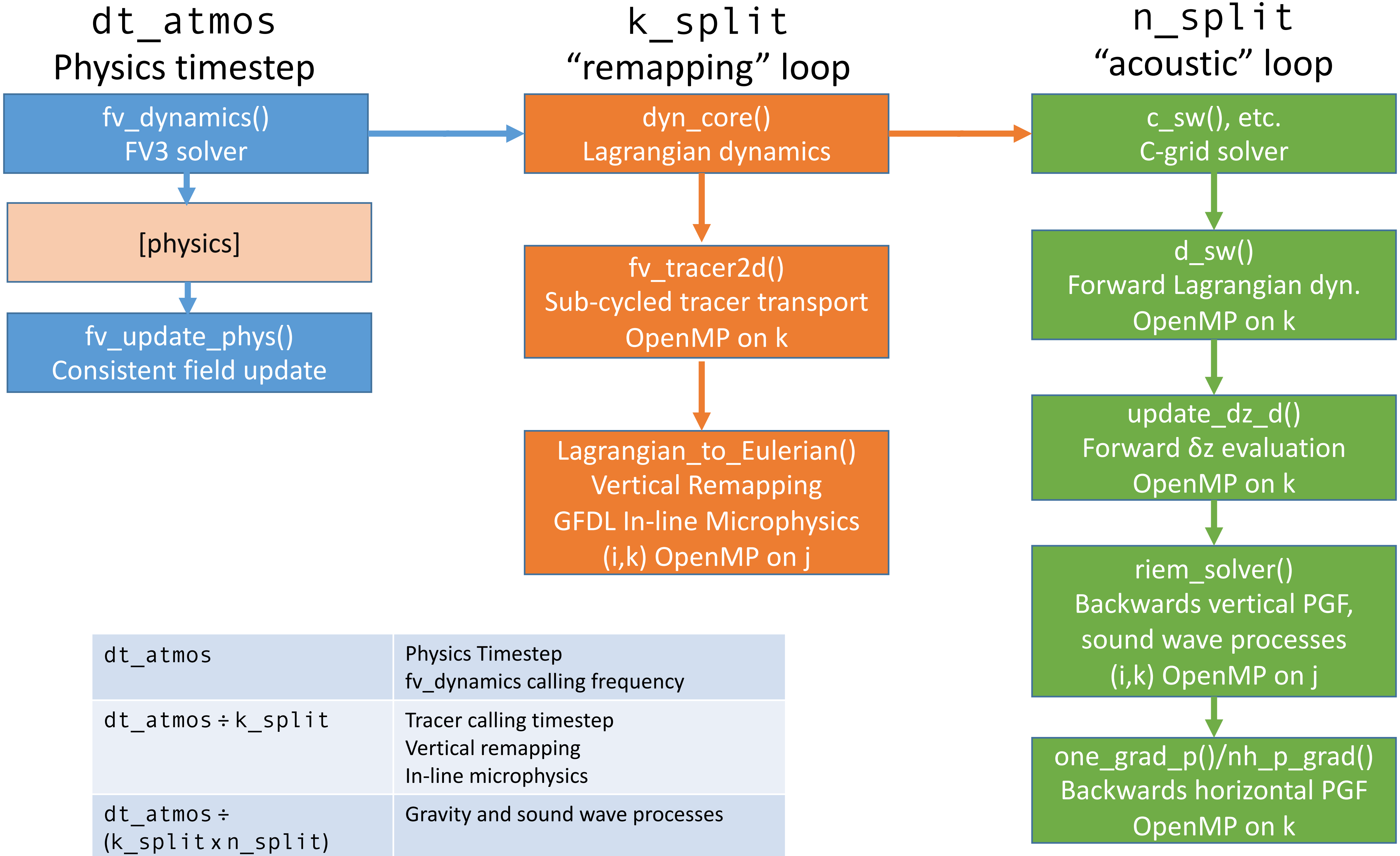
# Usage Guide

FV3 is a dynamical core and not a model.

- Correct: “FV3 is the dynamical core of the GFDL Modeling Suite and other UFS Configurations”
- Correct: “FV3 uses a Lagrangian vertical coordinate and the Putman and Lin (2007) advection scheme”
- Incorrect: “The convection scheme and land surface in FV3 have been updated.”
- ????: “FV-3 [sic]…an inferior model [sic] which will lead to decades of isolation.”

# Finite-volume methodology

- In FV3, all variables are 3D cell- or face-means…not gridpoint values
- We solve not the differential Euler equations but their cell-integrated forms using integral theorems
    - Everything is a flux, including the momentum equation
    - Mass conservation is ensured, to rounding error
    - C-D grid: Vorticity computed *exactly*; accurate divergence computation
    - Mimetic: Physical properties recovered by discretization, particularly Newton’s 3rd law
    - Fully compressible: calculation is horizontally local
    - Flow-following Lagrangian vertical coordinate ensures preservation of vertical structures and up/down drafts while greatly reducing computational cost
- FV3 is a fully forward-in-time solver with backwards PGF and acoustic terms



`dt_atmos`
Physics timestep

fv_dynamics()
FV3 solver

[physics]

fv_update_phys()
Consistent field update

`k_split`
“remapping” loop

dyn_core()
Lagrangian dynamics

fv_tracer2d()
Sub-cycled tracer transport
OpenMP on k

Lagrangian_to_Eulerian()
Vertical Remapping
GFDL In-line Microphysics
(i,k) OpenMP on j

`n_split`
“acoustic” loop

c_sw(), etc.
C-grid solver

d_sw()
Forward Lagrangian dyn.
OpenMP on k

update_dz_d()
Forward δz evaluation
OpenMP on k

riem_solver()
Backwards vertical PGF,
sound wave processes
(i,k) OpenMP on j

one_grad_p()/nh_p_grad()
Backwards horizontal PGF
OpenMP on k

| | |
|---|---|
| `dt_atmos` | Physics Timestep<br>fv_dynamics calling frequency |
| `dt_atmos ÷ k_split` | Tracer calling timestep<br>Vertical remapping<br>In-line microphysics |
| `dt_atmos ÷ (k_split x n_split)` | Gravity and sound wave processes |

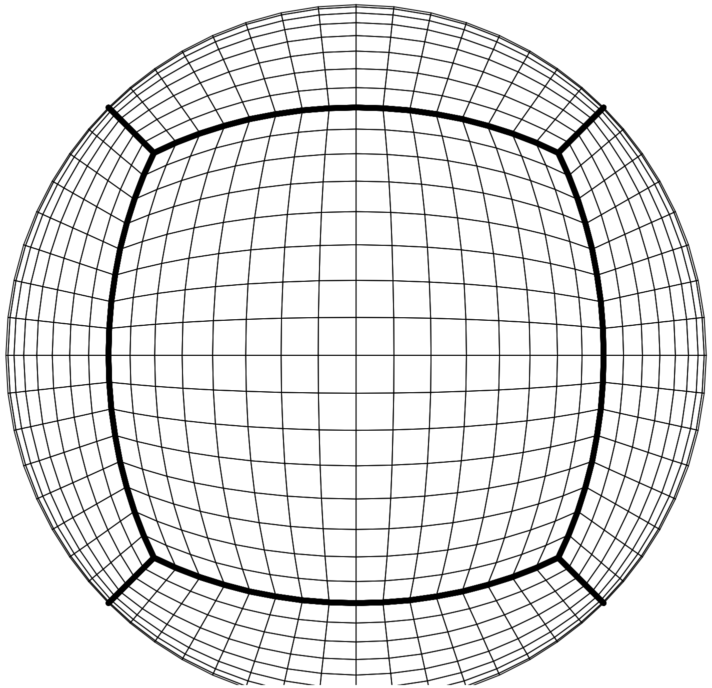
# Time integration: Namelist Options

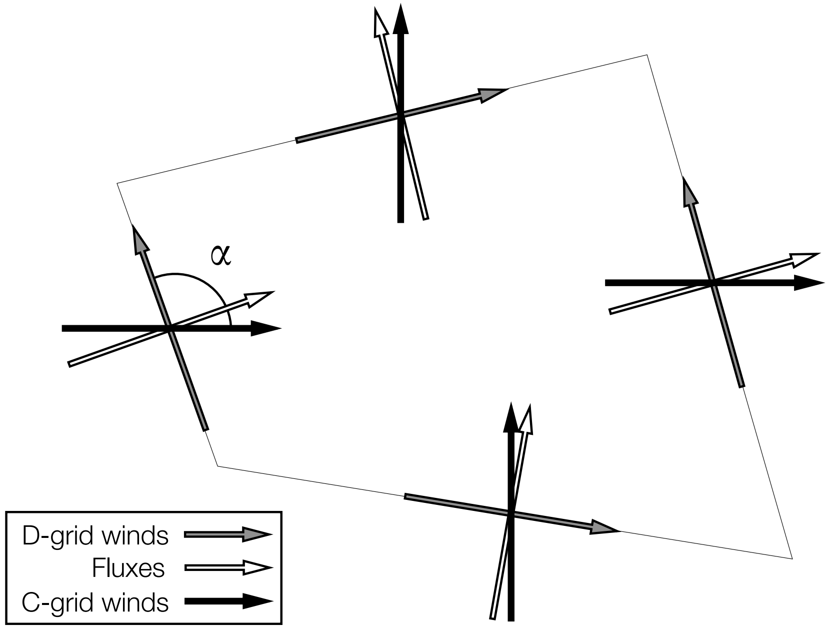
- `dt_atmos` (in atmos_model_nml): Timestep for the full FV3 solver and physics.
  - Should be motivated by physics design: for GFS Physics recommend 150–225
- `k_split`: Number of vertical remappings per long timestep.
  - More `k_splits` tend to improve stability but slow down model
- `n_split`: Number of acoustic timesteps per remapping timestep.
  - Recommend values between 5–10.
  - The acoustic timestep is equal to `dt_atmos` / (`k_split` x `n_split`)
- `hydrostatic`: whether to use the (much faster) hydrostatic solver.

# The Cubed-Sphere Grid

## The 3 in FV3

- Gnomonic cubed-sphere grid: coordinates are great circles but non-orthogonal
  - Solution winds are covariant, advection is by contravariant winds
- Winds u and v are defined in the local coordinate: rotation needed to get zonal and meridional components
  - Diagnostic winds *always* rotated into earth coordinates
- Special handling at edges and corners





# Cubed-Sphere Grid: Namelist Options

- `npx`, `npy`: Number of grid *corners* in each direction.
  - A global cubed sphere must use the same in both directions.
  - Nested, regional, or doubly-periodic domains do not.
- `ntiles`: Number of tiles on a domain.
  - For the cubed-sphere this must be 6
- `layout`: 2-element array for the number of MPI domain decompositions in each direction on each tile. These values should be divisors of (`npx` - 1) and (`npy` - 1)
  - Total number of cores is `layout`(1) x `layout`(2) x 6.

# Lagrangian Dynamics in FV3

- The Euler equations can be written in Lagrangian or Eulerian forms…
  or Eulerian in the horizontal, and Lagrangian in the vertical
- This constrains the flow along quasi-horizontal surfaces
- Lagrangian surfaces deform during the integration. Vertical motion and advection is “free”
- Requires layer thickness δp (and δz for nonhydro) to be a prognostic variable

$$\begin{aligned} D_L \delta p^* + \nabla \cdot (\mathbf{V} \delta p^*) &= 0 \\ D_L \delta p^* \Theta_v + \nabla \cdot (\mathbf{V} \delta p^* \Theta_v) &= 0 \\ D_L \delta p^* w + \nabla \cdot (\mathbf{V} \delta p^* w) &= -g \delta z \frac{\partial p'}{\partial z} \end{aligned}$$

$$\begin{aligned} D_L u &= \Omega v - \frac{\partial}{\partial x} \kappa - \frac{1}{\rho} \frac{\partial p^*}{\partial x} \Big|_z \\ D_L v &= -\Omega u - \frac{\partial}{\partial y} \kappa - \frac{1}{\rho} \frac{\partial p^*}{\partial x} \Big|_z \end{aligned}$$

$$\boxed{D_L \phi = \frac{\partial \phi}{\partial t} + \frac{\partial}{\partial z} \left( w \phi \right)}$$

# Prognostic Variables

| | |
|---|---|
| $\delta p$ | Total air mass (including vapor and condensates)<br>Equal to *hydrostatic* pressure depth of layer |
| $\theta_v$ | Virtual potential temperature |
| u, v | Horizontal D-grid winds in local coordinate<br>(defined on cell faces) |
| w | Vertical winds (nonhydrostatic) |
| $\delta z$ | Geometric layer depth (nonhydrostatic) |
| $q_i$ | Passive tracers |

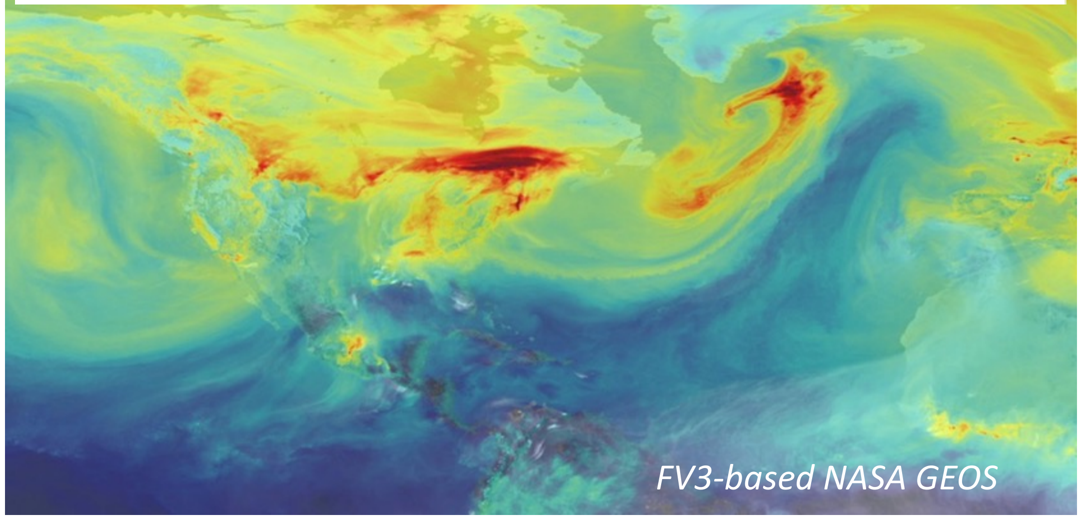
Cell-mean pressure, density, divergence, and specific heat are all *diagnostic* quantities
All variables are layer-means in the vertical: **No vertical staggering**

# FV Advection

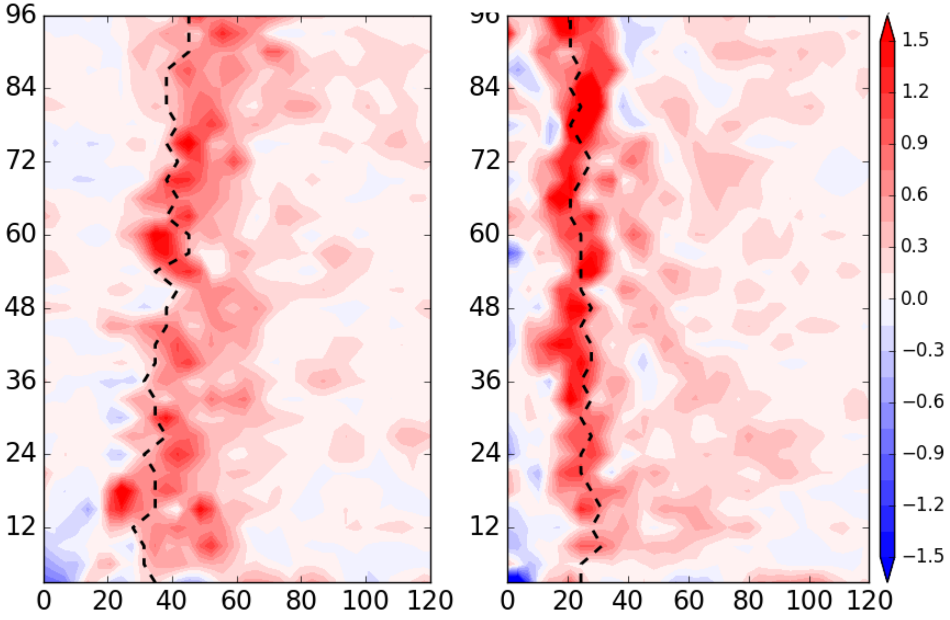
## Lin and Rood Advection

- “Reverse-engineered” forward-in-time 2D scheme constructed from 1D Piecewise-Parabolic Method (PPM) operators
  - Mass-conservative
  - Correlation-preserving for monotonic limiter
  - Cancels splitting error
  - Separate Courant number limit in x and y
  - Upwinding preserves hyperbolicity and causality
- Tracers are advected with an adaptive timestep using the accumulated mass fluxes
- *All* quasi-horizontal processes, except PGF, can be represented as advection
- Highly adaptable: Positive-definite tracer advection greatly improves hurricane structure

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



*FV3-based NASA GEOS*



Monotonic Advection

Positive-Definite Advection

*Axisymmetric 5-km W in Hurricane Irma*

*Harris et al. (2020) JAMES ; K Gao et al., in prep.*

# Advection Schemes: Namelist Options

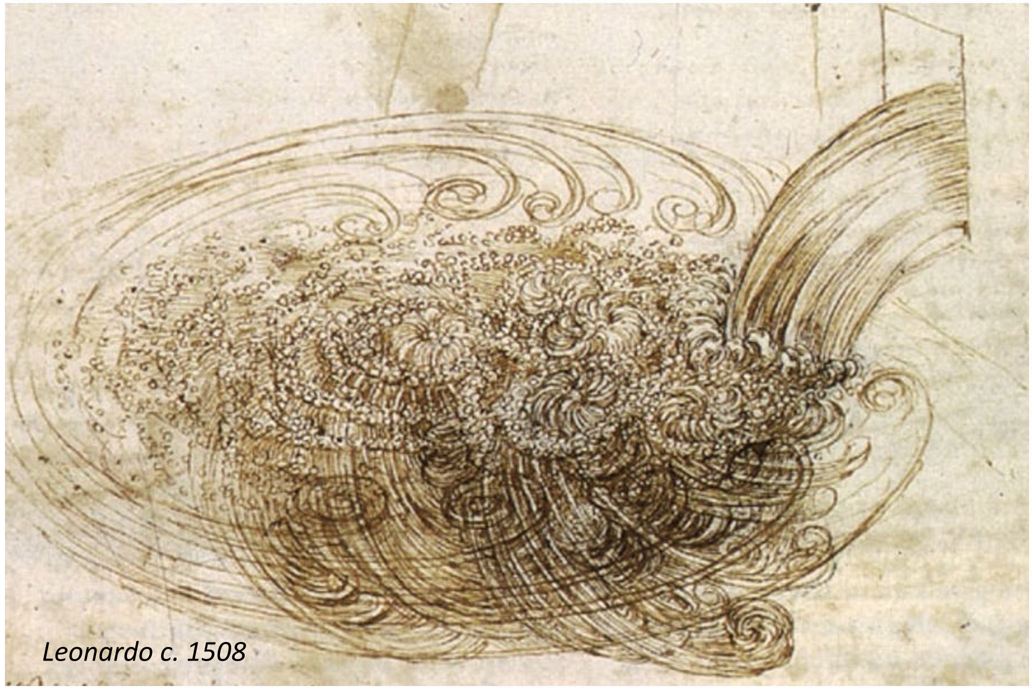
| | |
|---|---|
| `hord_mt` | KE gradient term |
| `hord_vt` | Vorticity and w fluxes |
| `hord_tm` | Potential temperature |
| `hord_dp` | δp, δz |
| `hord_tr` | Tracers |

- Strongly recommend `hord_mt`, `hord_vt`, and `hord_tm` use the same scheme
- `hord_tr` must use a monotone or positive-definite scheme

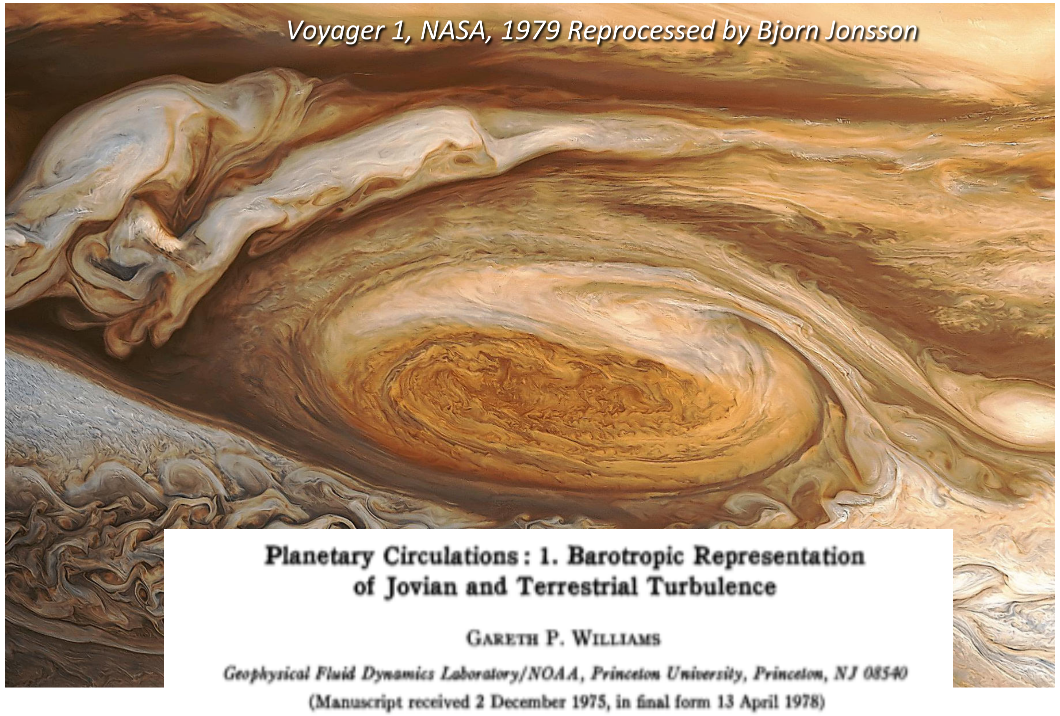
| `hord` | |
|---|---|
| 5 | Unlimited “fifth-order” scheme with weak 2Δx filter; fastest and least diffusive (“inviscid”) |
| 6 | Intermediate-strength 2Δx filter. Gives best ACC and storm structure but weaker TCs (“minimally-diffusive”) |
| 8 | Lin 2004 monotone PPM constraint (“monotonic”) |
| 9 | Hunyh constraint: more expensive but less diffusive than #8 |
| -5 | #5 with a positive-definite constraint |

# Vorticity Dynamics

- *Fluids are strongly vortical at all scales.* Vortical motions are especially critical in geophysical flows
- FV3’s discretization emphasizes vorticity dynamics:
  - Vector-invariant equations
  - C-D Grid Discretization
  - Consistent advection of derived vorticial quantities
- FV3’s preservation of vorticity is superior to other solvers without these properties
  - S-J Lin et al, 2017, JAMES



*Leonardo c. 1508*



*Voyager 1, NASA, 1979 Reprocessed by Bjorn Jonsson*

**Planetary Circulations: 1. Barotropic Representation of Jovian and Terrestrial Turbulence**

Gareth P. Williams

*Geophysical Fluid Dynamics Laboratory/NOAA, Princeton University, Princeton, NJ 08540*

(Manuscript received 2 December 1975, in final form 13 April 1978)

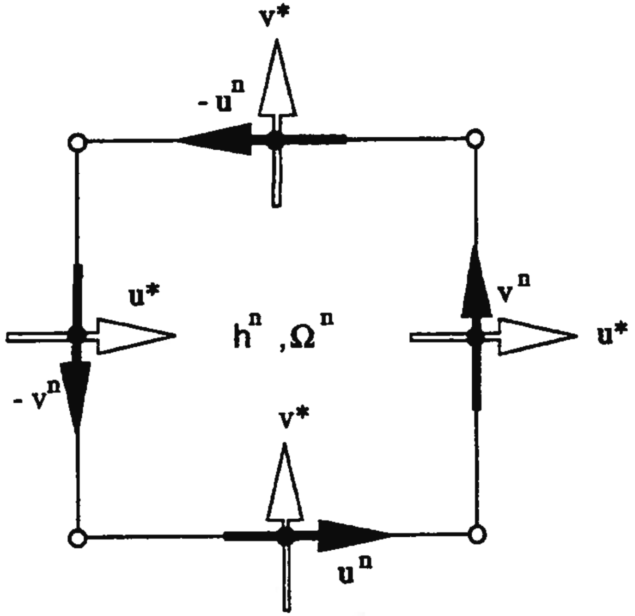
# Momentum equation

$$\frac{\partial \mathbf{V}}{\partial t} = -\Omega \hat{k} \times \mathbf{V} - \nabla \left( \kappa + \nu \nabla^2 D \right) - \frac{1}{\rho} \nabla p \Big|_z$$

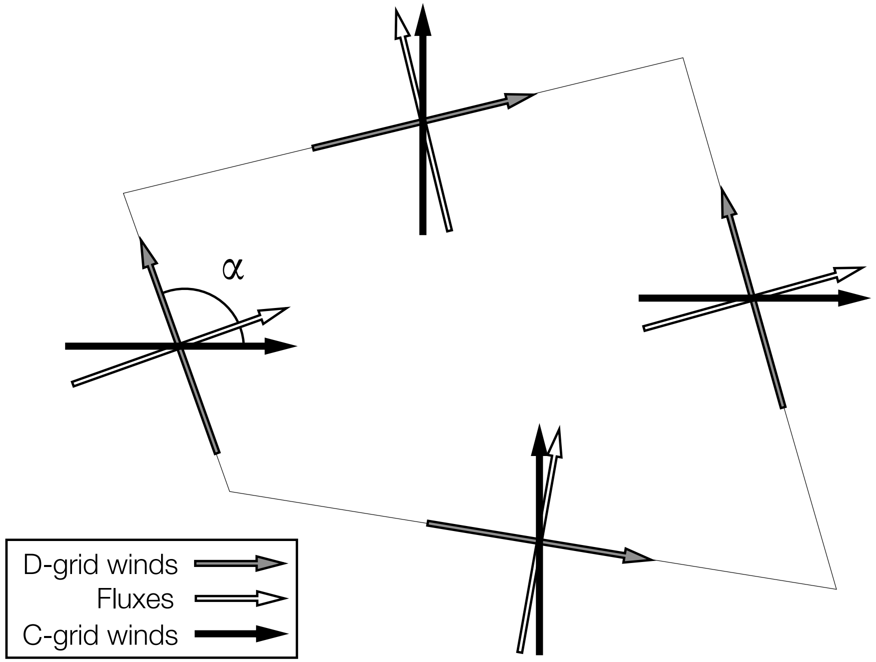
- FV3 solves the nonlinear flux-form vector invariant equations. One of the terms is the absolute vorticity flux.

$$u^{n+1} = u^n + \Delta\tau[Y(\widetilde{v^*}, \Delta\tau, \Omega^x) - \delta_x(\kappa^* - \nu\nabla^2 D) + \widehat{P}_x]$$
$$v^{n+1} = v^n + \Delta\tau[X(\widetilde{u^*}, \Delta\tau, \Omega^y) - \delta_y(\kappa^* - \nu\nabla^2 D) + \widehat{P}_y]$$

- D-grid allows *exact* computation of absolute vorticity using Stokes' theorem—no averaging!
- The cell-integrated vorticity is advected as a scalar, using the same fluxes as other variables:
  - Same flux as *h* → improves geostrophic balance → SW potential vorticity advected as a scalar
  - Same flux as *w* → improves nonlinear balance → updraft helicity advected as a scalar



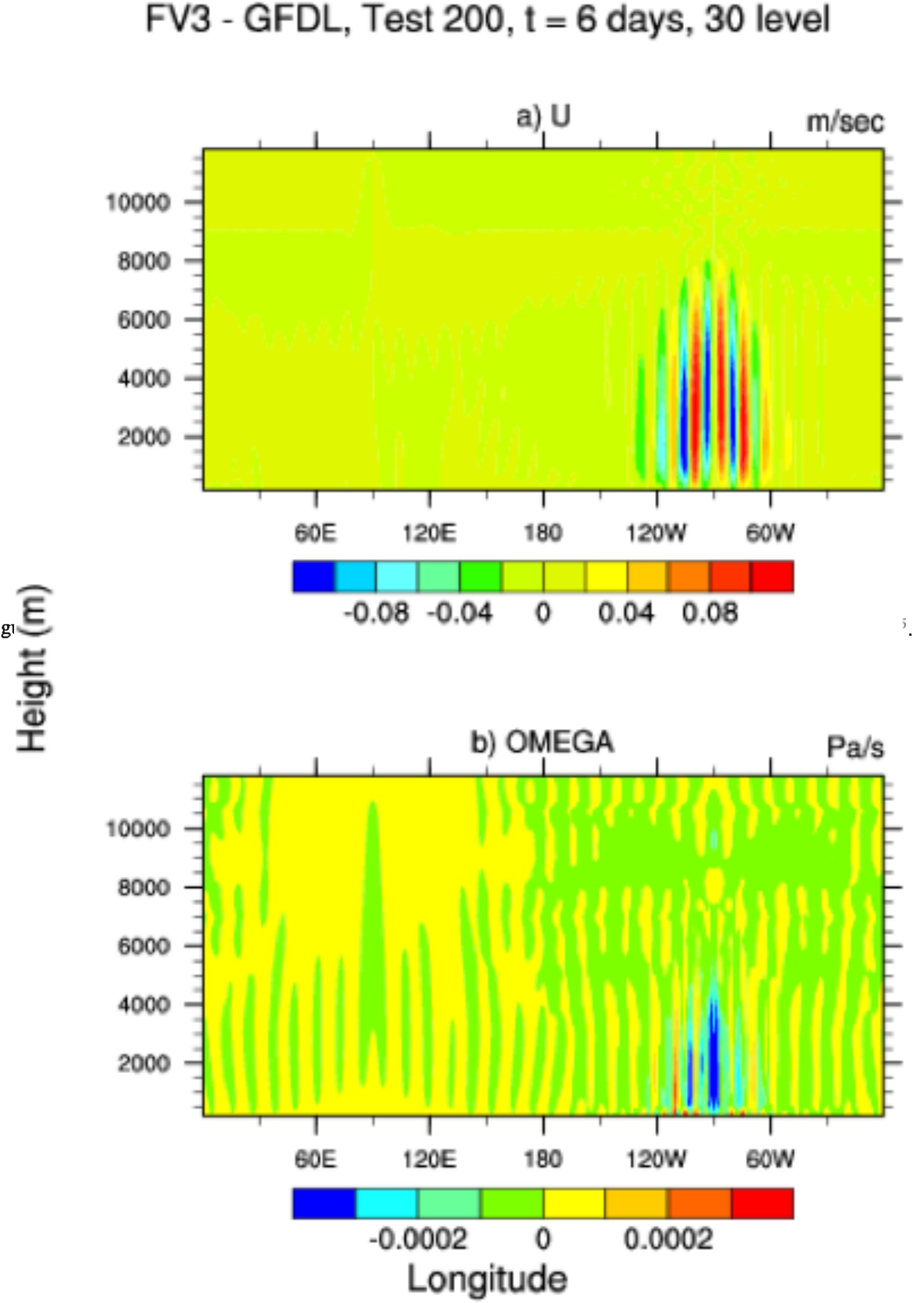
# The C-D grid solver



- FV3 solves for the (purely horizontal) D-grid staggered winds. But solver requires face-normal and time-mean fluxes.
- For time step-mean fluxes, the C-grid winds are interpolated and then advanced a half-timestep.
    - A sort of simplified *Riemann solver*
    - The C-grid solver is the same as the D-grid, but uses lower-order fluxes for efficiency
- Upstream flux also allows consistent computation of the KE gradient term, preventing the Hollingsworth-Kallberg instability
- Two-grid discretization and time-centered upwind fluxes avoid computational modes.
    - See the excellent discussion in Lin & Rood (1997, QJ).

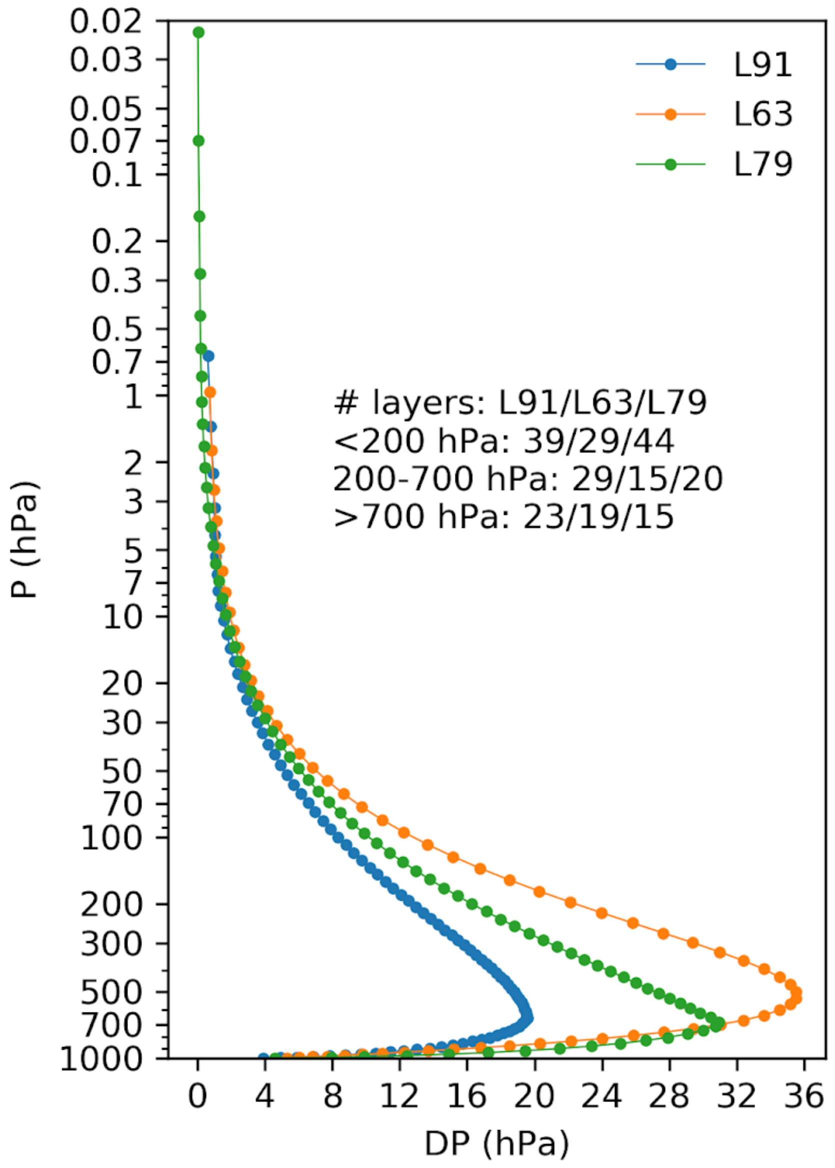
# Backward horizontal pressure gradient force

- Computed from Newton's second and third laws, and Green's Theorem
- Errors lower, with much less noise, compared to traditional evaluations
  - Purely horizontal
    **no** along-coordinate projection
  - PGF equal and opposite—3rd law!
    Momentum is conserved
  - Curl-free in the absence of density gradients
- Nonhydrostatic and hydrostatic components can be computed separately
  - $\log(p_{hyd})$ PGF more accurate



# Vertical Levels in FV3

- Reference interface pressure: $p_k = a_k + b_k * p_s$
  - Top level is k=1
- Vertical level setups are pre-defined in fv_eta.F90 with differing `npz`, ptop, and level positions.
  - These are carefully built to avoid instability and place levels where they are needed. *Use extreme caution when creating new sets of vertical levels.*
- `npz`: number of vertical levels
- `npz_type`: specific choice of level, depends on `npz`



GFS-like vertical level setups
Courtesy Linjiong Zhou

# The Lagrangian Vertical Coordinate

- Vertical advection is *implicit* through the deformation of quasi-horizontal layers. Computing δp and δz is sufficient for vertical advection.
  - **No** Courant number restriction or time-splitting
- Periodically, a high-order conservative remapping back to the reference coordinate is done to avoid δp → 0
  - Longer remapping interval yields less artificial diffusion
  - This is the **only** way cross-layer diffusion is introduced!!
- Remapping is from deformed layers back to the “Eulerian” reference coordinate

# Vertical Remapping Schemes: Namelist Options

| | |
|---|---|
| `kord_mt` | u and v |
| `kord_wz` | w |
| `kord_tm` | Temperature (< 0, recommended) or potential temperature (> 0); and density |
| `kord_tr` | Tracers |

- Strongly recommend all options use the same remapping scheme
- Remapping temperature is *geopotential conserving*

| `kord` | |
|---|---|
| 4 | Monotone PPM |
| 6 | Vanilla PPM |
| 7 | PPM with Hyunh’s monotonicity constraint (more expensive but less diffusive) |
| 9 | Monotonic Cubic Spline |
| 10 | Selectively (local extrema retained) monotonic Cubic Spline with 2Δz oscillations removed |
| 11 | Non-monotonic cubic spline with 2Δz oscillations removed |

# Semi-implicit solver

- Vertical pressure gradient and non-advective changes to layer depth δz are solved by semi-implicit solver
  - Vertically-propagating sound waves weakly damped

- This is all that is needed to make the classic FV hydrostatic algorithm nonhydrostatic
  - Fully compressible and nonhydrostatic! Full Euler equations solved
  - w, δz advected as other variables—consistent!
  - Nonhydrostatic horizontal PGF evaluated same way as hydrostatic

# Numerical Diffusion and Physical Dissipation

- *All useful numerical models have grid-scale motions removed by numerical diffusion—whether they know it or not.*
- Energy cascades to grid scales and **must** be removed since dissipative scales (~1 cm) are not explicitly resolved
- Grid-scale noise also arises from initial and boundary adjustment, physics interactions, and other imperfections, and must be removed
    - C-grids produce particularly prodigious noise at discontinuities
      X Chen et al. 2018, JAMES
- Diffusion is also a powerful tool to **improve** simulations
    - S2S and climate models: Zhao, Held, and Lin 2012, JAS
    - Convective scale and LES: Tompkins and Semie 2017; Pressel et al. 2017; see also *Implicit LES*

# Damping in FV3

- FV3's physical consistency produces very few computational modes and thus can be minimally-diffusive.
  But well-configured diffusion can give *improved* results
- FV3 applies **no** direct implicit diffusion to divergent modes which cascade to grid scale unimpeded.
  Scale-selective divergence damping represents their physical dissipation.
- Rotational modes can be damped implicitly by monotonic advection or explicitly by vorticity damping.
  - For consistency also damps $\delta p$, $\delta z$, $\theta_v$, w.
    **No** explicit damping for tracers.
- Note that **all** implicit (except vertical remapping) and explicit diffusion is **along** Lagrangian surfaces.

# Numerical Damping: Namelists

| `nord` | Damping order |  |
|---|---|---|
|  | Divg. | Vort. |
| 1 | 4th | 4th |
| 2 | 6th | 6th |
| 3 | 8th | 6th |

- `nord`: Controls order of damping.
  - Higher values mean more scale-selective damping
- `d4_bg`: Nondimensional divergence damping coefficient.
  - Values between 0.1 and 0.15 recommended.
- `do_vort_damp`: Logical flag for enabling vorticity damping.
- `vtdm4`: Nondimensional coefficient for vorticity damping. Should be much smaller than `d4_bg`.
  - Values between 0.02 and 0.06 are recommended.
- `d_con`: Fraction of damped KE restored as **heat**, conserving energy.
  - Set to 0.0 to disable this conversion; 1.0 restores all energy.
- `delt_max`: Limit on heating from damped KE (K/s).
  - Values between 0.002 and 0.008 are recommended.

# Upper Boundary: Namelist Options

- `d2_bg_k1`: Strength of second-order damping in top layer (k=1).
  - Values between 0.15 and 0.2 recommended.
- `d2_bg_k2`: …in second layer (k=2).
  - Recommend values between 0.02 and 0.1.
- `tau`: Timescale (days, smaller is stronger) of Rayleigh damping.
  - Recommend 5 for 13-km, 3 or 1.5 for 3-km.
- `rf_cutoff`: Level [Pa] above which Rayleigh damping is applied to u, v, w
- `n_sponge` (misleading artifact name): Number of layers from the top on which 2Δz energy-momentum-mass conserving filter is applied.
  - Recommend applying to layers above 100 mb.
- `fv_sg_adj`: Timescale (s, smaller is stronger) of 2Δz filter.
  - Use values larger than `dt_atmos` to avoid interfering with the PBL scheme.

# A few debugging and diagnostic options

- `print_freq`: frequency (in hours if > 0; in timesteps if < 0) of diagnostic outputs (max/min/ave, global integrals, etc.)
- `range_warn`: whether to check the ranges of values at different places in the core, and print out location of bad values
- `fv_debug`: print great volumes of solver information
- `no_dycore`: turn OFF the dynamics, enabling the column physics mode.
    - Good for debugging or testing “single column”.

# Initialization: Namelist Options

- `external_ic`: enable module for reading ICs from external file
- `nggps_ic`: Read regridded GFS ICs. Does no horizontal interpolation.
- `ecmwf_ic`: Read lat-lon ECMWF ICs, including horizontal interpolation
- `read_increment`: whether to read a DA increment from an external file and apply it in an FV-consistent way
- `res_latlon_dynamics`: input file for `ecmwf_ic` or increments
- `na_init`: # of forwards-backwards initialization iterations.
  - Spins up nonhydrostatic state when init from hydrostatic ICs
  - Set to 0 for GFSv15 or later ICs

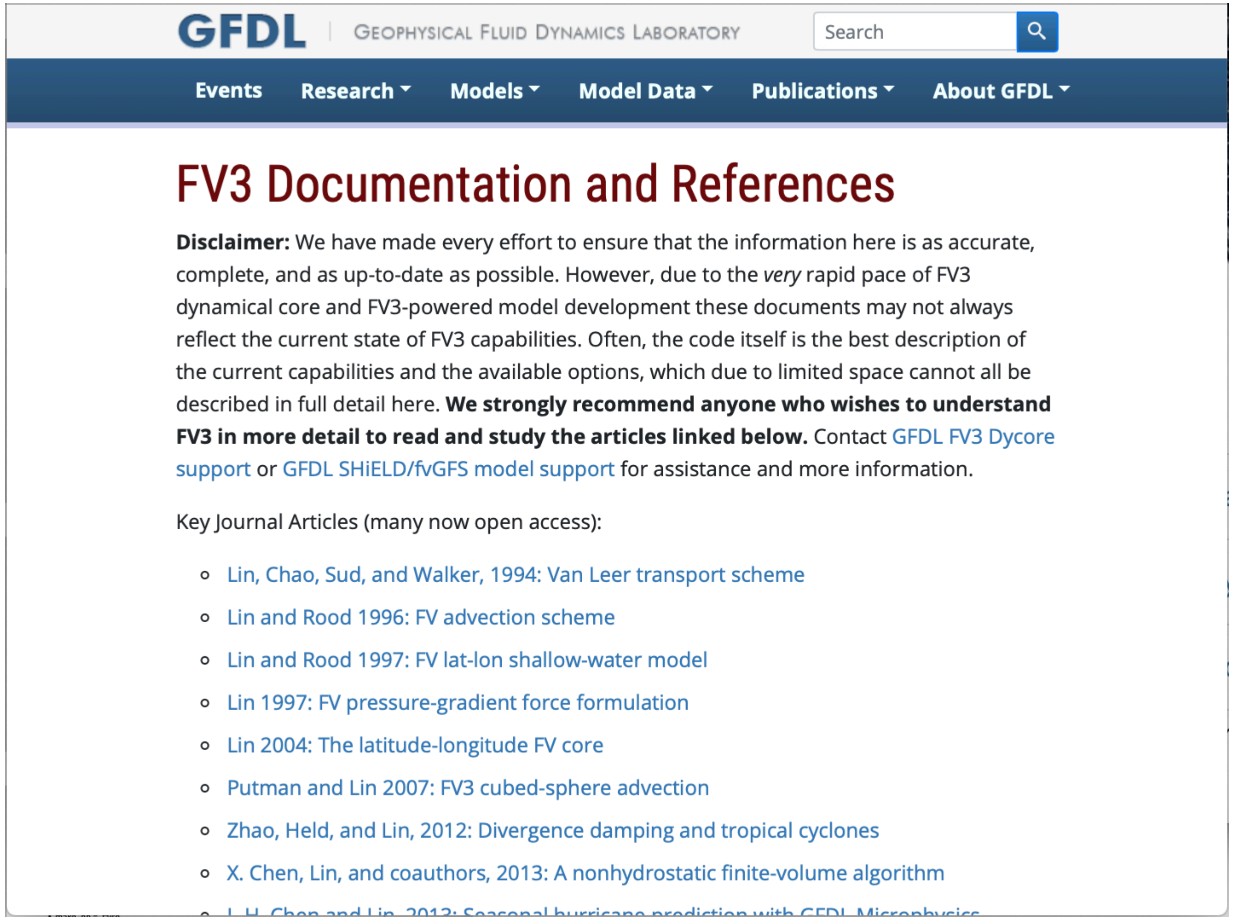
# Restarts: Namelist Options

- `external_eta`: read vertical level coefficients (ak, bk) from restarts instead of hard-coded values
- `agrid_vel_rst`: write out interpolated A-grid winds to restart files; very useful for DA cycling
- `npz_rst`: number of vertical levels in a restart file, if different from npz; FV3 will remap to the correct level spacing
- `make_nh`: Whether to re-generate nonhydrostatic fields from existing hydrostatic restarts. Not used for `nggps_ic`.

# When restarting

- Restarting FMS-based models is easy. Simply move the restart files from the RESTART/ directory to the INPUT/ directory
- Make sure to set several options to avoid solutions being reset:
    - `na_init = 0`
    - `external_ic = .false.`
    - `make_nh = .false.`
    - `mountain = .true.`
    - `n_zs_filter = 0`
    - `full_zs_filter = .false.`
    - `warm_start = .true.`

# For further reference



## FV3 Documentation and References

**Disclaimer:** We have made every effort to ensure that the information here is as accurate, complete, and as up-to-date as possible. However, due to the *very* rapid pace of FV3 dynamical core and FV3-powered model development these documents may not always reflect the current state of FV3 capabilities. Often, the code itself is the best description of the current capabilities and the available options, which due to limited space cannot all be described in full detail here. **We strongly recommend anyone who wishes to understand FV3 in more detail to read and study the articles linked below.** Contact GFDL FV3 Dycore support or GFDL SHiELD/fvGFS model support for assistance and more information.

Key Journal Articles (many now open access):

- Lin, Chao, Sud, and Walker, 1994: Van Leer transport scheme
- Lin and Rood 1996: FV advection scheme
- Lin and Rood 1997: FV lat-lon shallow-water model
- Lin 1997: FV pressure-gradient force formulation
- Lin 2004: The latitude-longitude FV core
- Putman and Lin 2007: FV3 cubed-sphere advection
- Zhao, Held, and Lin, 2012: Divergence damping and tropical cyclones
- X. Chen, Lin, and coauthors, 2013: A nonhydrostatic finite-volume algorithm
- J.-H. Chen and Lin, 2013: Seasonal hurricane prediction with GFDL Microphysics

www.gfdl.noaa.gov/fv3/fv3-documentation-and-references/

The GFDL Finite-Volume Cubed-Sphere
Dynamical Core: Release 201912

GFDL Weather and Climate Dynamics Division
Technical Memorandum GFDL2020001
10 Jan 2020

Document Prepared by Lucas Harris, Linjiong Zhou, Xi Chen, and
Jan-Huey Chen
For Shian-Jiann Lin and the GFDL FV3 Team

GFDL Tech Memos on the NOAA Institutional Repository
(more coming soon)