# FV3: Finite-Volume Cubed-Sphere Dynamical Core

https://www.gfdl.noaa.gov/fv3/

The GFDL Finite­-Volume Cubed-Sphere Dynamical Core (FV3) is a scalable and flexible dynamical core capable of both hydrostatic and non-hydrostatic atmospheric simulations. The design of FV3 was guided by these tenets:

1. Discretization should be guided by physical principles as much as possible.
2. A fast model is a good model! Computational efficiency is crucial.

FV3 was “reverse engineered” to incorporate properties which have been used in engineering for decades, but only first adopted in atmospheric science by FV3.

FV3 has been chosen as the dynamical core for the Next Generation Global Prediction System project (NGGPS), designed to upgrade the current operational Global Forecast System (GFS) to run as a unified, fully-coupled system in NOAA’s Environmental Modeling System infrastructure. FV3 was successfully implemented within the GFS, and the FV3-based GFSv15 became operational on 12 June 2019. Other applications, such as regional high-resolution forecasting and coupled atmosphere-ocean modeling for seasonal prediction, are planned for later implementation at NCEP.

This website describes FV3, including the evolution of its development, basic algorithm, and its global variable resolution capabilities, in both nested and stretched grid configurations. The Performance page explains how efficient FV3 can be.

Dynamics isn’t the whole story. Coupling to physics and the ocean is necessary! Please see the Applications page for the family of models using FV3 and examples of how FV3 has been successfully implemented.

# Development History

## Finite-Volume Schemes

The FV core started its life at NASA/Goddard Space Flight Center (GSFC) during early and mid-90s as an offline transport model with emphasis on the conservation, accuracy, consistency (tracer to tracer correlation), and efficiency of the transport process. The development and applications of monotonicity-preserving Finite­-Volume schemes at GSFC were motivated in part by the need to have a “fix” for the noisy and unphysical negative water vapor and chemical species (Lin et al. 1994, and Lin and Rood 1996). It subsequently has been used by several high­-profile Chemistry Transport Models (CTMs), including the NASA­-community GMI model (Rotman et al., 2001), GOCART (Chin et al., 2000), and the Harvard University-­developed GEOS­-Chem model. This transport module has also been used by several climate models, including the ECHAM5 AGCM.

## Shallow-Water Model

Motivated by the success of monotonicity­-preserving FV schemes in CTM applications, a consistently formulated shallow-water model was developed. This solver was first presented at the 1994 PDE on the Sphere Workshop, and years later published by Lin and Rood (1997). The Lin­-Rood algorithm for shallow­-water equations maintains mass conservation and a key Mimetic property of “no false vorticity generation”, and for the first time in computational geophysical fluid dynamics, uses high­-order monotonic advection consistently for momentum and all other prognostic variables, instead of the inconsistent hybrid finite­-difference and finite­-volume approach used by practically all other “finite­-volume” models today.

## FV Hydrostatic Dynamical Core

The full 3D hydrostatic dynamical core, the FV core, was constructed based on the Lin­-Rood (1996) transport algorithm and the Lin­-Rood shallow­-water algorithm (1997). The pressure gradient force is evaluated by the Lin (1997) finite­-volume integration method, derived from Green’s integral theorem based directly on first principles, and demonstrated errors an order of magnitude smaller than other well­-known pressure­-gradient schemes. Finally, the vertical discretization is the “vertically Lagrangian” scheme described by Lin (2004).

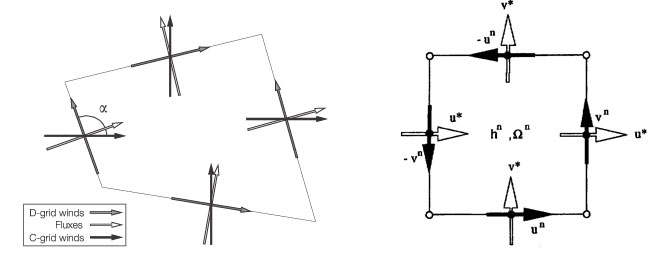
## From FV to FV3

The most unique aspect of the FV3 is its Lagrangian vertical coordinate, which is computationally efficient as well as more accurate given the same vertical resolution. Recently, a more computationally efficient non­hydrostatic solver is implemented using a traditional semi-implicit approach for treating the vertically propagating sound waves. This faster solver is the default. The Riemann solver option is more efficient for resolution finer than 1­km, and also more accurate, because sound waves are treated nearly exactly. A description of the non-hydrostatic extension can be found on the Key Components page.

# Key Components

## Horizontal Discretization

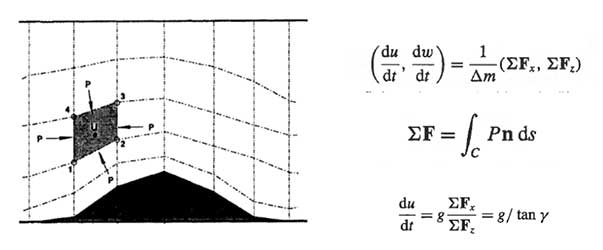
The horizontal discretization of FV3 is essentially the same as the original FV core (Lin and Rood, 1997; Lin 2004), except that all spatial averaging and pressure gradient operators have been upgraded from the 2nd to formally 4th­order accurate in the FV3. The FV shallow-water solver has discretization on D-grid, with C-grid winds which are interpolated from D-grid winds to compute fluxes. For the nonlinear vorticity flux term in the momentum equation, using D-grid allows exact computation of absolute vorticity with no averaging.



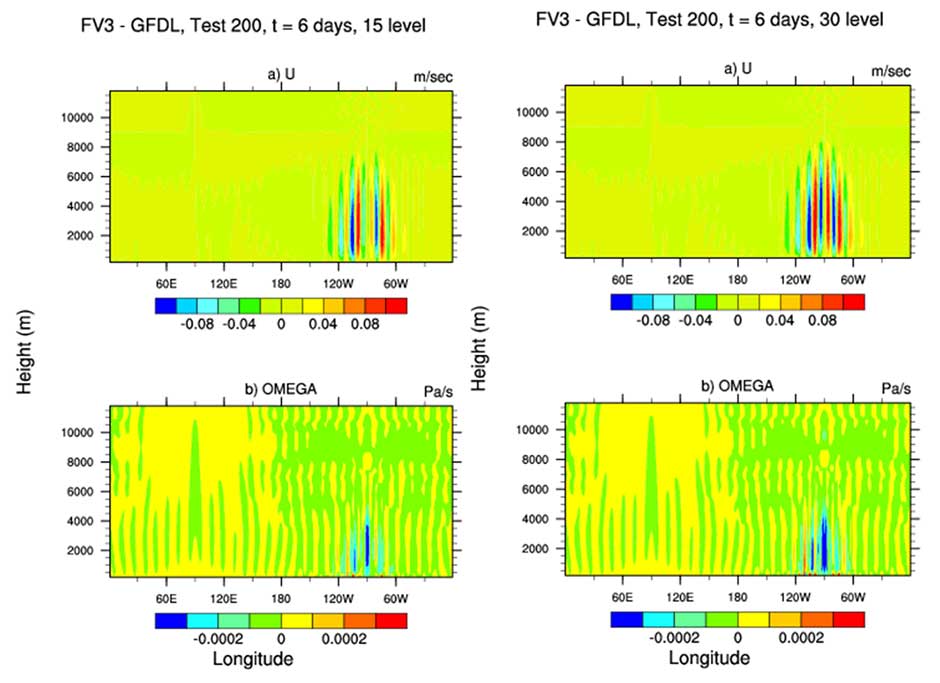
More details on the horizontal discretization in FV3 is given in Harris and Lin (2013) and Harris (2016 JCLI, accepted).

## The FV Pressure Gradient Computation

The evaluation of the pressure­-gradient force in FV3 remains the same as in the FV core (Lin 1997), upgraded to 4th-order accuracy. A lesser­-known aspect of the Lin (1997) algorithm is its consistency with Newton’s 3rd law of motion, achieved by finite­-volume integration about a grid cell: the pressure force exerted upon a cell by its neighbor is equal and opposite to that exerted by the cell upon the neighbor. This form satisfies Newton’s third law of motion in the same way flux­form transport schemes satisfy mass conservation. Other algorithms for evaluating the pressure-­gradient force do not meet this requirement, with consequences revealed by a simple hydrostatic equilibrium test; see Lin (1997) for details.



A similar test was performed as part of the Dynamical Core Intercomparison Project (Test 2­0­0, Ullrich et al 2012): initially resting, hydrostatic atmosphere is imposed upon topography, and the resulting spurious accelerations are then computed. It is found that FV3 produces little spurious oscillation compared to many other schemes.



Results at 6 days from DCMIP test case 2-0-0 for two different vertical resolutions.

## Vertically Lagrangian Discretization and Non-Hydrostatic Extensions

The Lagrangian vertical coordinate used in FV3 is fully described by Lin (2004). It is a terrain-following pressure coordinate.

Pk=ak+bk\*Ps

Equations of motion are vertically integrated to yield a series of layers, and each layer is like a shallow-water system. The 2D horizontal-to-Lagrangian-surface transport and dynamical processes are discretized using the genuinely conservative flux-form semi-Lagrangian algorithm. Time marching is split-explicit, with large time steps for scalar transport, and small fractional steps for the Lagrangian dynamics, which permits the accurate propagation of fast waves.

A mass, momentum, and total energy conserving algorithm is developed for remapping the state variables periodically to an Eulerian terrain-following coordinate to perform vertical transport, and to avoid layers from becoming infinitesimally thin. As long as the layer thickness is positive, the model retains stability. Therefore, there is no vertical courant number limitation in FV3. This is critically important in non-hydrostatic simulations.

FV3 contains two non-hydrostatic solvers for vertical velocity and pressure perturbation. The first, a Riemann solver, was developed at GFDL during 2003-2006 based on a conservative Riemann invariants approach. This algorithm is particularly suitable for ultra-high resolution cloud-resolving simulations, with grid-cell widths of 1 km or less. A variation of this approach, within a simplified 2D vertically-Lagrangian framework, has been published by Chen et al. (2013).

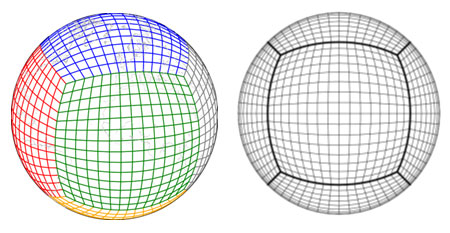
The second non-hydrostatic solver, developed recently for lower horizontal resolutions, is a more traditional semi-implicit time integration scheme for vertically propagating sound waves. This solver is more suitable and more efficient for lower horizontal resolution simulations, in which the extra damping provided by the semi-implicit time-integration scheme can act to filter out the poorly resolved sound waves, and therefore provides a less-noisy simulation. Both solvers use the same governing equations. All FV3 simulations for NGGPS use the semi-implicit solver.

More information about FV3 design and implementation is available at [FV3 Documentation and References.](https://www.gfdl.noaa.gov/fv3/fv3-documentation-and-references/)

# Grids

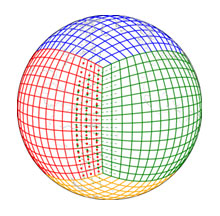
## FV3 on Cubed­-Sphere Grid

The transport scheme used by FV3 is documented in three main publications (Lin et al. 1994, Lin and Rood 1996, and Putman and Lin 2007). The Putman­-Lin scheme is a refinement of the Lin and Rood (1996) scheme for the various Cubed­-Sphere grids. Currently, the Gnomonic grid is the grid of our choice, due to its best grid uniformity.



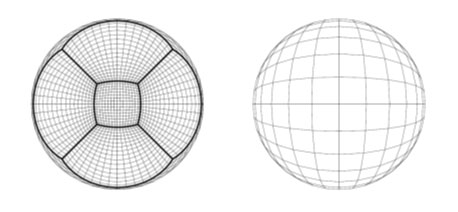
## Cubed-Sphere Edge Handling

At the edges of the cubed­-sphere tiles, two one-­sided 3rd order extrapolations were averaged to form a directionally symmetric scheme across the edges (see Putman and Lin 2007). This averaging, however, locally reduces the formal accuracy from 4th order in the interior to only 2nd order at the edges of the cubed-sphere, and it creates some grid imprinting due to the mild discontinuity of the great­-circle grid lines and some reduction in accuracy. Fortunately, the grid imprinting is greatly reduced at increasing resolutions, since the two­-sided extrapolation algorithm converges although more slowly than in the interior. At NOAA’s Global Forecast System’s current resolution of 13­km the grid imprinting is practically non­existent for weather predictions. Nonetheless, to improve lower resolution climate applications, we are still working on a revised edge handling algorithm, by using an extended-grid approach, so the algorithm remains 4 th order accurate at the edges. We believe improved edge handling will further reduce the already­-small grid imprinting.

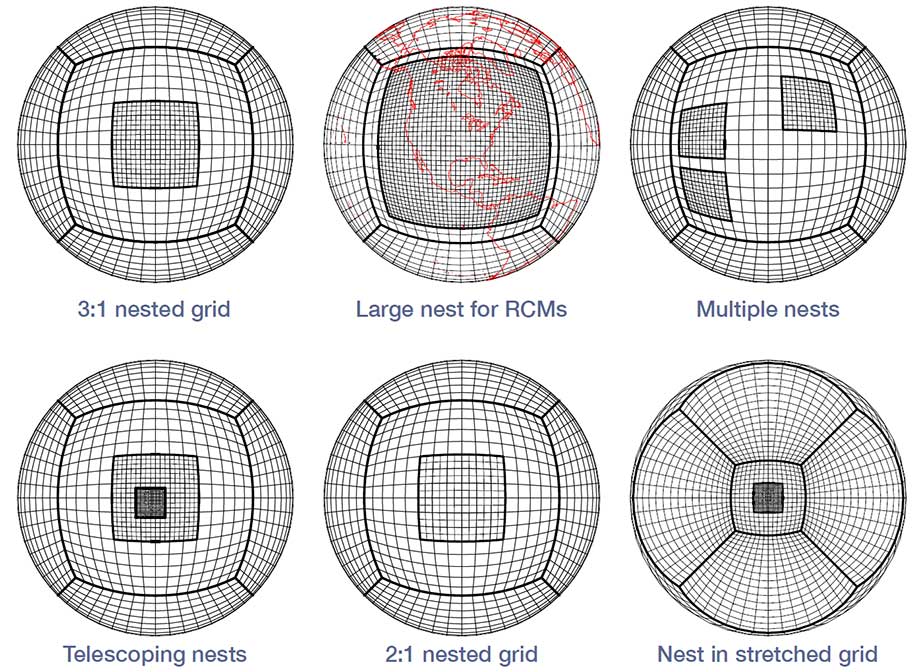


## Grid Stretching and Two-way Nested Grid

FV3 supports two means of grid refinement: a continuous grid stretching and two-way grid nesting. Grid stretching is described in Harris et al. (2016 CJLI, accepted). Stretching is achieved by applying an analytic Schmidt (1977) transformation to the cubed sphere grid to smoothly reduce the size of one cube face, and to center that enhanced-resolution cube face over the targeted region of interest. The resulting grid has a smoothly-varying resolution between the enhanced- and degraded-resolution regions (antipodal point of the target). The solver can be used un-modified on this stretched grid, although the time step is restricted by the smallest grid cell. The refinement ratio can be any floating point value; short-term storm-scale simulations have been performed using ratios as high as 80, so that it is operationally feasible within a GFS-like global model to have local grid-cell widths as high as 500 meters. This configuration has been used to explore tornado-producing supercell predictions.



The two-way nested algorithm is fully described by Harris and Lin (2013). The nested and coarse grids are run concurrently on separate sets of processors, to aid load balancing between the processors dedicated to the coarse grid and those dedicated to the nested grid, and to allow simultaneous, coupled regional and global solutions. Each grid may use different settings appropriate for its own resolution, including different time steps and different physics parameterizations. Only the winds and temperature are updated to the coarse grid in the two-way interaction; the coarse-grid air mass field is undisturbed, trivially conserving mass. The two-way nested algorithm may be extended to multiple telescoping nests, and supports any integer refinement ratio. Further resolution enhancement can be found by combining the stretched and nested grids.



Both the nested and stretched grids have been used for multi-decadal regional climate simulations, in which the resolved regional scales are able to directly interact with the global-scale circulation. Efficient climate simulations with 25-km grid-cell widths over North America and the Western Pacific were performed by Harris and Lin (2014); economical high-resolution climate simulations with 10-km grid-cell widths over the same regions were performed by Harris et al. (2016 CJLI, accepted). Robust improvements to orographic precipitation, the diurnal cycle of continental precipitation, and tropical cyclone intensity were found in the refined regions compared to the uniform-resolution base global grids, with little evidence of grid artifacts. In particular, any grid artifacts are much less severe than those found from one-way nesting (Harris and Lin 2014). The large-scale climate was not appreciably degraded by the use of either grid stretching or nesting; in some cases, particularly the 10-km stretched grid, the errors in large-scale climate statistics was found to decrease compared to the uniform-resolution base grid.

# Performance

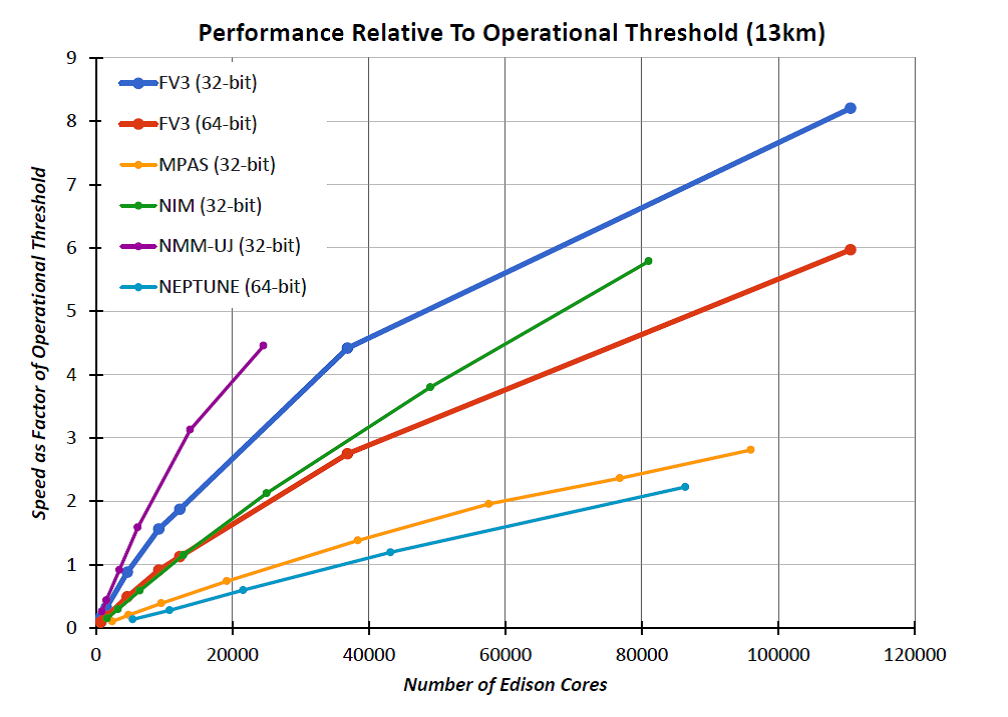
The National Weather Service has performed a technical evaluation of high-performance computing suitability and readiness of all candidate model dycores running on the NERSC Edison system, for the Next Generation Global Prediction System initiative.

A detailed report can be found from:

<https://www.earthsystemcog.org/site_media/projects/dycore_test_group/20150602_AVEC_Level_1_Benchmarking_Report_08.pdf>

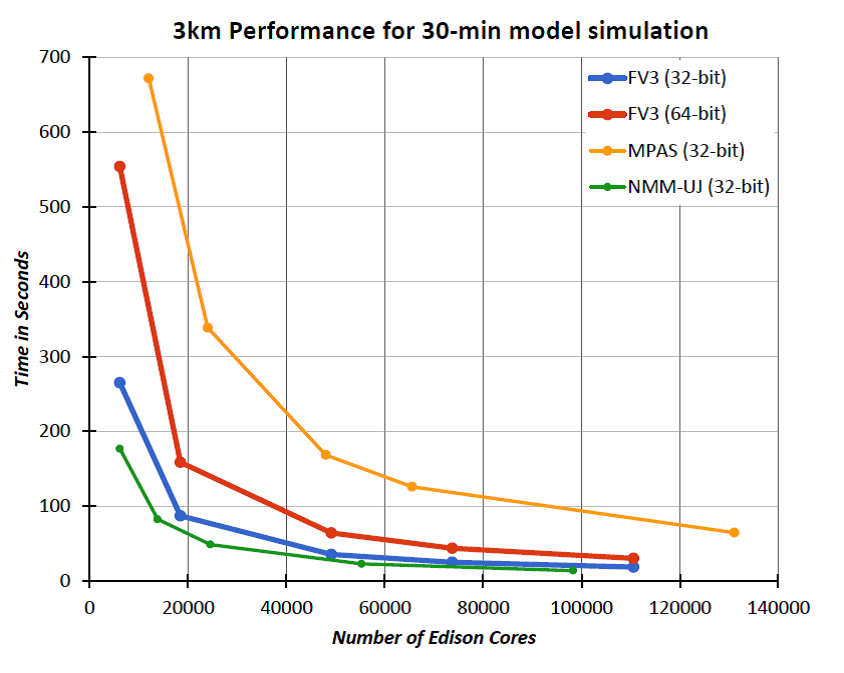
Here, we highlight the FV3 performance and scaling tests for 13-km and 3-km simulations shown in the report. Based on Tables 3 and 4 in the report, we replot the charts with better labels to indicate the numerical precision (32-bit or 64-bit) used by each model.

The NGGPS test plan specified 8.5 minutes per forecast day as the operational forecast speed requirement. Assuming dynamics would comprise half the model run time, the speed requirement of the dycore is to run a 2-hour simulation in 21.25 seconds. The figure below shows model speed as a function of number of cores, normalized to the speed requirement of 8.5 min/day, for 13-km simulations.



低分辨率情况下，MPAS很有优势！

For 3-km simulations, the performance is plotted as elapsed time to run a 30-minute simulation. FV3 with single (double) precision takes 265 (554) seconds when using 6411 processor cores, while scaling from 6144 to 110592 processor cores with 78 (88) percent efficiency.



高分辨率情况下，FV3有绝对优势！