FV^3

Namelist options in the GFDL Finite-Volume Cubed-Sphere Dynamical Core

Shian-Jiann Lin and Lucas Harris

Initial Version 12 Aug 2013
Version 2.0a Revised 7 Jan 2016

Entries in fv_core_nml

Required options:

layout Integer(2): Processor layout on each tile. The number of PEs assigned to a domain must equal layout(1)*layout(2)*ntiles. Must be set.

npx Integer: Number of grid *corners* in the x-direction on one tile of the domain; so one more than the number of grid cells across a tile. On the cubed sphere this is *one more than* the number of cells across a cube face. Must be set.

npy Integer: Number of grid *corners* in the y-direction on one tile of the domain. This value should be identical to npx on a cubed-sphere grid; doubly periodic or nested grids do not have this restriction. Must be set.

npz Integer: Number of vertical levels. Each choice of npz comes with a pre-defined set of hybrid sigma-pressure levels and model top (see atmos_cubed_sphere/tools/fv_eta.F90). Must be set.

ntiles Integer: Number of tiles on the domain. For the cubed sphere, this should be 6, one tile for each face of the cubed sphere; normally for most other domains (including nested grids) this should be set to 1. Must be set.

Initialization options:

add_noise Real: amplitude of random thermal noise (in K) to add upon startup. Useful for perturbing initial conditions. -1 by default; disabled if 0 or negative.

adjust_dry_mass Logical: whether to adjust the global dry-air mass to the value set by “dry_mass”. This is only done in an initialization step, particularly when using an initial condition from an external dataset, interpolated from another resolution (either horizontal or vertical), or when changing the topography, so that the global mass of the atmosphere matches some estimate of observed value. False by default. It is recommended to only set this to True when initializing the model.

breed_vortex_inline Logical: whether to bogus tropical cyclones into the model, which are specified by an external file. Options are set in fv_nwp_nudge_nml. False by default.

dry_mass Real: if adjust_dry_mass is true, sets the global dry air mass, measured in the globally-averaged surface pressure (Pascals) by adding or removing mass from the lowest layer of the atmosphere as needed. 98290. (Pa) by default.

external_ic Logical: Whether to initialize the model’s state using the data in an externally specified file, given in res_latlon_dynamics. By default this file is assumed to be a legacy lat-lon FV core restart file; set either ncep_ic or fv_diag_ic to true override this behavior. .false. by default. Note that external_ic = true will cause the model to re-initialize the dynamical fields from the input dataset regardless of whether warm_start is set.

fv_diag_ic Logical: If external_ic = .true., this variable says whether the file in res_latlon_dynamics is a latitude-longitude grid file containing FV core history . .false. by default. The input file does not need all tracer fields, but any that the model cannot find will be set to zero. Requires ncep_ic = .false.

mountain Logical: takes topography into account when initializing the model. Set this to true to apply the terrain filter (if n_zs_filter = 2 or 4) upon startup; also set to True when cold starting so that the topography can be initialized. Only set this to false if you wish to cold-start without any topography; this value is ignored for the aquaplanet test_case = 14. True by default. It is highly recommended to not alter this value unless you know what you are doing.

na_init Integer: Number of forward-backward dynamics steps used to initialize adiabatic solver. 0 by default. Not recommended if not cold-starting the model.

ncep_ic Logical: If external_ic = .true., this variable says whether the file in res_latlon_dynamics is an NCEP analysis or reanalysis file. This option zeros out all tracer fields except specific humidity. .false. by default.

nggps_ic Logical: If external_ic = .true., reads in vapor, liquid, and ice phases of water and ozone from the NCEP analysis or reanalysis file. ncep_ic must be false. False by default.

nord_zs_filter Integer: order of the topography filter applied to n_zs_filter. Set to 2 to get a second-order filter, or 4 to get a fourth-order filter; other values do no filtering. 0 by default. This should not be set to a non-zero value on multiple successive simulations; the filter is applied every time the model restarts. This option is useful for testing terrain filter, and should not be used for regular runs.

npz_rst Integer: If using a restart file with a different number of vertical levels, set npz_rst to be the number of levels in your restart file. The model will then remap the restart file data to the vertical coordinates specified by npz. 0 by default; if 0 or equal to npz no remapping is done.

nudge Logical: whether to use the nudging towards the state in some externally-supplied file (such as from reanalysis or another simulation). Further nudging options are set in fv_nwp_nudge_nml. False by default.

nudge_ic Logical: same as nudge, but works in adiabatic solo_core simulations to nudge the field to a single external analysis file. False by default.

n_zs_filter Integer: number of times to apply a diffusive filter to the topography upon startup, if mountain is True and the model is not being cold-started. This is applied every time the model is warm-started, so if you want to smooth the topography make sure this is set to 0 after the first simulation. 0 by default. If initializing the model from cold-start the topography is already being filtered by an amount appropriate for the model resolution.

reset_eta Logical: whether to reset the model's vertical levels on startup to the values hard-coded in fv_eta.F90, instead of using the values in the restart files. This option can be used to override the vertical coordinate in the model with either that of an older version of the model, or to user-specified values. False by default. (Not currently implemented.)

res_latlon_dynamics character(len=128) If external_ic = .true. gives the filename of the input IC file. INPUT/fv_rst.res.nc by default.

res_latlon_tracers character(len=128) If external_ic = .true. and both ncep_ic and fv_diag_ic are .false., this variable gives the filename of the initial conditions for the tracers, assumed to be a legacy lat-lon FV core restart file. INPUT/atmos_tracers.res.nc by default.

warm_start Logical; whether to start from restart files, instead of cold-starting the model. True by default; if this is set to true and restart files cannot be found the model will crash.

I/O and diagnostic options:

check_negative Logical: whether to print the most negative global value of microphysical tracers.

fv_debug Logical: whether to turn on additional diagnostics in fv_dynamics. .false. by default.

fv_land Logical: whether to create terrain deviation and land fraction for output to mg_drag restart files, for use in mg_drag and in the land model. .false. by default; .true. is recommended when, and only when, initializing the model, since the mg_drag files created provide a much more accurate terrain representation for the mountain gravity wave drag parameterization and for the land surface roughness than either computes internally. This has no effect on the representation of the terrain in the dynamics.

io_layout Integer(2): Layout of output files on each tile. 1,1 by default, which combines all restart and history files on a tile into one file. For 0,0, every process writes out its own restart and history files. If not equal to 1,1, you will have to use mppnccombine to combine these output files prior to post-processing, or if you want to change the number of PEs. Both entries must divide the respective value in layout.

nf_omega Integer: number of times to apply second-order smoothing to the diagnosed omega. When 0 the filter is disabled. 1 by default.

print_freq Integer: number of hours between print out of max/min and air/tracer mass diagnostics to standard output. 0 by default, which never prints out any output; set to -1 to see output after every dt_atmos. Computing these diagnostics requires some computational overhead.

range_warn Logical: checks whether the values of a number of variables are within a “reasonable” range at the end of a dynamics time step, and prints a warning if not. False by default; adds computational overhead, so we only recommend using this when debugging.

Options controlling tracers and interactions with physics

adiabatic Logical: whether to skip any physics. If true, the physics is not called at all and there is no virtual temperature effect. False by default; this option has no effect if not running solo_core.

do_Held_Suarez Logical: whether to use Held-Suarez forcing. Requires adiabatic to be false. False by default; this option has no effect if not running solo_core.

dnats Integer: The number of tracers which are not to be advected by the dynamical core, but still passed into the dynamical core; the last dnats+pnats tracers in field_table are not advected. 0 by default.

dwind_2d Logical: whether to use a simpler & faster algorithm for interpolating the A-grid (cell-centered) wind tendencies computed from the physics to the D-grid. Typically, the A-grid wind tendencies are first converted in 3D cartesian coordinates and then interpolated before converting back to 2D local coordinates. When this option enabled, a much simpler but less accurate 2D interpolation is used. False by default.

fill Logical: Fills in negative tracer values by taking positive tracers from the cells above and below. This option is useful when the physical parameterizations produced negatives. False by default.

inline_q Logical: whether to compute tracer transport in-line with the rest of the dynamics instead of sub-cycling, so that tracer transport is done at the same time and on the same timestep as is δp and potential temperature. False by default; if true, q_split and z_tracer are ignored.

ncnst Integer: Number of tracer species advected by fv_tracer in the dynamical core. Typically this is set automatically by reading in values from field_table, but ncnst can be set to a smaller value so only the first ncnst tracers listed in field_table are not advected. 0 by default, which will use the value from field_table.

nwat Integer: Number of water species to be included in condensate and water vapor loading. The masses of the first nwat tracer species will be added to the dry air mass, so that δ*p* is the mass of dry air, water vapor, and the included condensate species. The value used depends on the microphysics in the physics package you are using. For GFS physics set to 2. For standard AM2/AM3/AM4 Rotstayn-Klein three-phase microphysics set to 3. For warm-rain (Kessler) microphysics set to 4 (with an inactive ice tracer), which only handles three species but uses 4 to avoid interference with the RK physics. For six-category Lin microphysics set to 6. A value of 0 turns off condensate loading. 3 by default.

phys_hydrostatic Logical: Option to enable "hydrostatic" application of heating from the physics in a nonhydrostatic simulation: heating is applied in hydrostatic balance, causing the entire atmospheric column to expand instantaneously. If false, heating from the physics is applied simply as a temperature tendency. True by default; ignored if hydrostatic = .true.

pnats Integer: The number of tracers not to advect by the dynamical core. Unlike dnats, these tracers are not seen by the dynamical core. The last pnats entries in field_table are not advected. 0 by default.

tau_h2o Real: time-scale (days) for simple methane chemistry to act as a source of water in the stratosphere. Can be useful if your stratosphere dries out too quickly; consider a value between 60 and 120 days if this is the case. 0. by default, which disables the methane chemistry. Values less than zero apply the chemistry above 100 mb; else applied above 30 mb. Requires adiabatic to be false.

use_hydro_pressure Logical: whether to compute hydrostatic pressure for input to the physics. Currently only enabled for the FV-GFS model. Ignored in hydrostatic simulations. False by default.

z_tracer Logical: whether to transport sub-cycled tracers layer-by-layer, each with its own computed sub-cycling timestep (if q_split = 0). This may improve efficiency for very large numbers of tracers. False by default; currently not implemented.

Timestep options

k_split Integer: number of vertical remappings per dt_atmos (physics time step). 1 by default.

n_split Integer: number of small dynamics ('acoustic') timesteps between vertical remapping. 0 by default, in which case the model produces a good "first guess" by examining the resolution, dt_atmos, and k_split.

umax Real: for the doubly-periodic grid (grid_type = 4) an estimate of the maximum wave speed (m/s), used to determine the value of n_split when n_split = 0. 350 by default.

q_split Integer: number of timesteps for sub-cycled tracer advection. 0 by default (recommended), in which case the model determines the number of timesteps from the global maximum wind speed at each call to the tracer advection.

Grid options

deglat Real: Latitude (in degrees) used to compute the uniform f-plane Coriolis parameter for doubly-periodic simulations (grid_type = 4). 15. by default.

do_schmidt Logical: Whether to enable grid stretching and rotation using stretch_fac, target_lat, and target_lon. .false. by default.

dx_const Real: on a doubly-periodic grid (grid_type = 4) specifies the (uniform) grid-cell-width in the x-direction, in meters. 1000 by default.

dy_const Real: on a doubly-periodic grid (grid_type = 4) specifies the (uniform) grid-cell-width in the y-direction, in meters. 1000 by default.

grid_type Integer: which type of grid to use. If 0, the equidistant gnomonic cubed-sphere will be used. If 4, a doubly-periodic f-plane cartesian grid will be used. If -1, the grid is read from INPUT/grid_spec.nc . Values 2, 3, 5, 6, and 7 are not supported and will likely not run. 0 by default.

hybrid_z Logical: whether to use a hybrid-height coordinate, instead of the usual sigma-p coordinate. False by default. (Not currently maintained.)

make_hybrid_z Logical: Converts the vertical coordinate to a hybrid-height coordinate, instead of the usual sigma-p coordinate. Requires hybrid_z = True. False by default.

p_ref Real: surface pressure used to construct a horizontally-uniform reference vertical pressure profile, used in some simple physics packages in the solo_core and in the Rayleigh damping. *This should not be confused with the actual, horizontally-varying pressure levels used for all other dynamical calculations.* 1.e5 by default. *Changing this value is strongly discouraged.*

shift_fac Real: westward zonal rotation (or shift) of cubed-sphere grid from its "natural" orientation with cube face centers at 0, 90, 180, and 270 degrees longitude. The shift, in degrees, is 180/shift_fac. This shift does not move the poles. By default this is set to 18, shifting the grid westward 180/18=10 degrees, so that the edges of the cube do not run through the mountains of Japan; all "standard" CM2.x, AM3, CM3, and HiRAM simulations use this orientation of the grid. Requires do_schmidt = .false.

stretch_fac Real: stretching factor for the Schmidt transformation. This is the factor by which tile 6 of the cubed sphere will be shrunk, with the grid size shrinking accordingly. 1 by default, which performs no grid stretching. Requires do_schmidt = .true. The model will crash if stretch_fac is set to zero. Values of up to 40 have been found useful and stable for short-term cloud-scale integrations.

target_lat Real: latitude (in degrees) to which the center of tile 6 will be rotated; if stretching is done with stretch_fac the center of the high-resolution part of the grid will be at this latitude. -90 by default, which does no grid rotation (the Schmidt transformation rotates the south pole to the appropriate target). Requires do_schmidt = .true.

target_lon Real: longitude to which the center of tile 6 will be rotated. 0 by default. Requires do_schmidt = .true.

nested Logical: whether this is a nested grid. False by default.

twowaynest Logical: whether to use two-way nesting, the process by which the nested-grid solution is allowed to feed back onto the coarse-grid solution. False by default.

parent_grid_num Integer: Number of the parent to this nested grid. The coarsest grid in a simulation is numbered 1; further nested grids are numbered sequentially. Required to be a positive value if nested = True. Unless you are nesting inside of nested grids or running multiple (coarse) grids that themselves do not interact, this should be set to 1. -1 by default, indicating that this grid does not have a parent grid.

parent_tile Integer: number of the tile (ie. face) in which this nested grid is found in its parent. Required to be a positive value if nested = true. If the parent grid is not a cubed sphere, or itself is a nested grid, this should be set to 1. If the parent grid has been rotated (using do_schmidt) with the intent of centering the nested grid at target_lat and target_lon, then parent_tile should be set to 6. 1 by default.

refinement Integer: refinement ratio of the nested grid. This is the number of times that each coarse-grid cell face will be divided into smaller segments on the nested grid. Required to be a positive integer if nested = true. Nested grids are aligned with the coarse grid, so non-integer refinements are not permitted. 3 by default.

nestupdate Integer: type of nested-grid update to use; details are given in model/fv_nesting.F90. 0 by default.

Solver options

a2b_ord Integer: order of interpolation used by the pressure gradient force to interpolate cell-centered (A-grid) values to the grid corners. 4 by default (recommended), which uses fourth-order interpolation; otherwise second-order interpolation is used.

beta Real: Parameter specifying fraction of time-off-centering for backwards evaluation of the pressure gradient force. 0.0 by default, which produces a fully backwards evaluation—the pressure gradient force is entirely evaluated using the updated (time n+1) dynamical fields. A value of 0.5 will equally weight the PGF determined at times n and n+1, but may not be stable; values larger than 0.45 are not recommended. A value of 0.4 is recommended for most hydrostatic simulations, which allows an improved representation of inertia-gravity waves in the tropics. In non-hydrostatic simulations using the semi-implicit solver (a_imp > 0.5) the values of a_imp and beta should add to 1, so that the time-centering is consistent between the PGF and the nonhydrostatic solver. Proper range is 0 to 0.45.

c2l_ord Integer: order of interpolation from the solver's native D-grid winds to latitude-longitude A-grid winds, which are used as input to the physics routines and for writing to history files. 4 by default (recommended); fourth-order interpolation is used unless c2l_ord = 2.

consv_am Logical: whether to enable Angular momentum fixer. False by default.

consv_te Real: fraction of total energy lost during one full dynamic step to be added back globally as heat; essentially the strength of the "energy fixer" in the physics. 0 by default. Proper range is 0 to 1. 1 will restore the energy completely to its original value before entering the physics; a value of 0.7 *roughly* causes the energy fixer to compensate for the amount of energy changed by the physics in GFDL HiRAM or AM3. Note that this is a global energy fixer. The default algorithm used the potential temperature increment so as to maintain the pressure gradients.

convert_ke Logical: If true, adds energy dissipated through mechanical damping to heat throughout the *entire* depth of the domain; if false (default) this is only done in the sponge layer at the top of the domain. This option is only enabled if d_con > 1.e-5.

d_con Real: Fraction of kinetic energy lost to net damping (divergence and vorticity) to be converted to heat. Acts as a dissipative heating mechanism in the dynamical core. 0. by default. Proper range is 0 to 1. Note that this is a local, physically correct, energy fixer.

fill_dp Logical: like fill except for δp, the hydrostatic pressure thickness. When the filling occurs a diagnostic message is printed out, which is helpful for diagnosing where the problem may be occurring. Typically if the pressure filling is needed a crash is inevitable, and thus this option is often better for debugging than as a "safety valve". False by default.

fv_sg_adj Real: timescale (in seconds) at which to remove two-delta-z instability when the local (between two adjacent levels) Richardson number is less than 1. This is achieved by local mixing, which conserves not only mass, momentum, but also total energy. The mixing is instantaneous if fv_sg_adj is smaller than the dynamics timestep (does not work at this point). Values of 0 or smaller disable this feature. If n_sponge > 0 then the mixing is applied only to the top n_sponge layers of the domain. Set to -1 (inactive) by default. Proper range is 0 to 3600.

halo_update_type Integer: which scheme to use for halo updates in multiprocessor simulations. If set to 1 non-blocking updates are used, which can improve simulation efficiency on some systems. Otherwise, blocking halo updates are performed. 1 by default.

hord_mt Integer: horizontal advection scheme for momentum fluxes. A complete list of kord options is given in the table below. 9 by default, which uses the third-order piecewise-parabolic method with the monotonicity constraint of Huynh, which is less diffusive than other constraints (such as the original Colella and Woodward PPM monotonicity constraint, option 4, or the monotonicity constraint of Lin 2004, option 8). 10 is recommended for resolutions higher than AM2/AM3's standard two degree resolution. Other options are 1, which does first-order upwind advection (useful for debugging purposes); and 6, which does a completely un-limited (i.e., linear) PPM advection. It is recommended that hord_mt, hord_vt, hord_tm, and hord_dp use the same value, to ensure consistent transport of all dynamical fields, unless a positivity constraint on mass advection (hord_dp) is desired; see below. If using the -5 options for hord_vt and hord_tm, use 6 for hord_mt, -5 for hord_dp, -8 for hord_tr, and then enable vorticity damping (see below).

hord_vt Integer: horizontal advection scheme for absolute vorticity and for vertical velocity in nonhydrostatic simulations. 9 by default.

hord_tm Integer: horizontal advection scheme for potential temperature and layer thickness in nonhydrostatic simulations. 9 by default.

hord_dp Integer: horizontal advection scheme for mass. A positivity constraint may be warranted for hord_dp but not strictly necessary. 9 by default.

hord_tr Integer: horizontal advection scheme for tracers. 12 by default. This value can differ from the other hord options since tracers are sub-cycled (if inline_q == False) and often require positive-definite advection. 13 or -8 are typically recommended.

hord	Horizontal Advection Method
1	First-order upwind (also called donor-cell or piecewise-constant)
2	Van Leer unlimited (piecewise-linear; may not work)
3	Unlimited Colella and Woodward (1984) Piecewise-Parabolic
4	3, with the monotonic limiter of Colella and Woodward; not available for hord_mt
5	3, using Hunyh's second constraint to enforce monotonicity; not available for hord_mt
-5	Fastest unlimited “fifth-order” scheme with built-in 2Δ filter; not available for hord_mt, not recommended for hord_tr. This is also the most accurate and least diffusive FV scheme available within FV3 if monotonicity preservation is not a high priority.
6	Unlimited "fifth-order" PPM. This option may be useful for 4D-DA.
7	6, applying a $2\Delta x$ filter and a positivity constraint; not available for hord_mt
8	PPM with the constraint of Lin 2004; better optimized for hord_mt
-8	Optimized monotone PPM scheme; not available for hord_mt
9	PPM with the Hunyh constraint; better optimized for hord_mt
10	9, with a $2\Delta x$ filter, and the Huynh constraint applied only if a certain condition is met; otherwise unlimited; better optimized for hord_mt
11	Same as 8, but better optimzed; not available for hord_mt
12	Same as 9, but better optimized and with a positivity constraint; not available for hord_mt
13	Same as 10, but better optimized and with a positivity constraint; not available for hord_mt

kord_mt Integer: vertical remapping scheme for the winds. 8 by default; 9 recommended. See table below for a complete list of kord options.

kord_tm Integer: vertical remapping scheme for temperature. If positive, then vertical remapping is performed on total energy instead of temperature (see remap_t below). -8 by default. -9 recommended.

kord_tr Integer: vertical remapping scheme for tracers. 8 by default. 9 or 11 recommended.

kord_wz Integer: vertical remapping scheme for vertical velocity in nonhydrostatic simulations. 8 by default; 9 recommended. It is also recommended to use the same value for kord_wz as for kord_mt.

kord	Vertical remapping reconstruction method
7	Monotonic PPM
8	Monotonic (and conservative) cubic spline
9	Monotonic cubic spline with 2dz oscillations removed
10	Selectively monotonic cubic spline, where local extrema are retained, with 2dz oscillations removed
12	Monotonic cubic spline with all local extrema flattened
13	Monotonic cubic spline with 2dz oscillations removed
14	Non-monotonic ("linear") cubic spline
11,15	Non-monotonic ("linear") cubic spline with 2dz oscillations removed; if an invalid value for kord is given, this scheme is used
16	Optimized non-monotonic ("linear") cubic spline

no_dycore Logical: disables execution of the dynamical core, only running the initialization, diagnostic, and I/O routines, and any physics that may be enabled. False by default.

remap_t Logical: whether the vertical remapping is performed on (virtual) temperature instead of (virtual) potential temperature. Since typically potential temperature increases exponentially from layer to layer near the top boundary, the cubic-spline interpolation in the vertical remapping will have difficulty with the exponential profile. Temperature does not have this problem and will often yield a more accurate result. True by default.

reproduce_sum Logical: uses an exactly-reproducible global sum operation performed when computing the global energy for consv_te. This is used because the FMS routine mpp_sum() is not bit-wise reproducible due to its handling of floating-point arithmetic, and so can return different answers for (say) different processor layouts. True by default.

Nonhydrostatic options

a_imp Real: Controls behavior of the non-hydrostatic solver. Values > 0.5 enable the semi-implicit solver, in which the value of a_imp controls the time-off-centering: use a_imp = 1.0 for a fully backward time-stepping. For consistency, the sum of beta and a_imp should be 1 when the semi-implicit solver is used. To use the Riemann solver, set a_imp = 0. The semi-implicit algorithm is substantially more efficient except at very high (km-scale) resolutions with an acoustic timestep of a few seconds or less. 0.75 by default. Proper values are 0, or between 0.5 and 1 Only used if hydrostatic = .false.

hydrostatic Logical: whether to use the hydrostatic or nonhydrostatic solver. True by default.

m_split Integer: Number of time splits for the Riemann solver. Only used if hydrostatic = .false. and if a_imp < 0.5. 0 by default, in which case the model produces a good “first guess” by examining the resolution, dt_atmos, k_split, and n_split.

make_nh Logical: Whether to re-initialize the nonhydrostatic state, by re-computing dz from hydrostatic balance and setting w to 0. False by default.

p_fac Real: Safety factor for minimum nonhydrostatic pressures, which will be limited so the full pressure is no less than p_fac times the hydrostatic pressure. This is only of concern in mid-top or high-top models with very low pressures near the model top, and has no effect in most simulations. The pressure limiting activates only when model is in danger of blowup due to unphysical negative total pressures. 0.05 by default. Only used if hydrostatic = .false. and the semi-implicit solver is used. Proper range is 0 to 0.25.

scale_z Real: scaling to reduce the effect of hydrostatic pressure in semi-implicit solver. 0 (disabled) by default. This option is no longer supported.

use_logp Logical: Enables a variant of the Lin pressure-gradient force algorithm, which uses the logarithm of pressure instead of the Exner function (as in Lin 1997). This yields more accurate results for regions that is near isothermal. Ignored if hydrostatic = true. False by default.

w_max Real: Not used. 75 by default.

z_min Real: Not used. 0.05 by default.

Damping options

d2_bg Real: coefficient for background second-order divergence damping. This option remains active even if nord is nonzero. 0.0 by default. Proper range is 0 to 0.02.

d2_bg_k1 Real: strength of second-order diffusion in the top sponge layer. 0.16 by default. This value, and d2_bg_k2, will be changed appropriately in the model (depending on the height of model top), so the actual damping may be very reduced. See atmos_cubed_sphere/model/dyn_core.F90 for details. Recommended range is 0. to 0.2.

d2_bg_k2 Real: strength of second-order diffusion in the second sponge layer from the model top. 0.02 by default. This value should be lower than d2_bg_k1.

d4_bg Real: Dimensionless coefficient for background higher-order divergence damping. 0.0 by default. If no second-order divergence damping is used, then values between 0.1 and 0.16 are recommended. Requires nord > 0. Note that the scaling for d4_bg differs from that of d2_bg; nord >= 1 and d4_bg = 0.16 will be less diffusive than nord = 0 and d2_bg = 0.02.

dddmp Real: Dimensionless coefficient for the second-order Smagorinsky-type divergence damping. 0.0 by default. 0.2 (the Smagorinsky constant) is recommended if ICs are noisy.

d_ext Real: coefficient for external (barotropic) mode damping. 0.02 by default. Proper range is 0 to 0.02. A value of 0.01 or 0.02 may help improve the model's maximum stable timestep in low-resolution (2-degree or poorer) simulations; otherwise a value of 0 is recommended.

do_vort_damp Logical: whether to apply damping (of strength governed by vtdm4) to the other dynamical variables: vorticity, and nonhydrostatic vertical velocity, but not to the tracers. The form is the same as is used for the divergence damping, including the same order (from nord) damping, unless nord = 0, in which case this damping is fourth-order. We recommend enabling this damping when the "linear" or non-monotonic horizontal advection schemes are enabled, but is unnecessary when using monotonic advection. False by default.

n_sponge Integer: controls behavior of sponge layers at the upper boundary. If 0 (recommended for climate simulations of 1 degree or better) a resolution-aware "best choice" of settings for the sponge layer. For positive values, the filter fv_sg_adj is applied only to the top n_sponge layers. 0 by default.

nord Integer: order of divergence damping: 0 for second-order; 1 for fourth-order (default); 2 for sixth-order; 3 for eighth-order. Sixth-order may yield a better solution for low resolutions (one degree or coarser) by virtue of it being more scale-selective and will not damp moderately-well-resolved disturbances as much as does lower-order damping.

nord_tr Integer: Order of tracer damping; values mean the same as for nord. 0 by default.

rf_cutoff Real: pressure below which no Rayleigh damping is applied if tau > 0.

tau Real: time scale (in days) for Rayleigh friction applied to horizontal and vertical winds; lost kinetic energy is converted to heat, except on nested grids. 0.0 by default, which disables damping. Larger values yield less damping. For models with tops at 1 mb or lower values between 10 and 30 are useful for preventing overly-strong polar night jets; for higher-top hydrostatic models values between 5 and 15 should be considered, with larger values producing less damping; and for non-hydrostatic models values of 10 or less should be considered.

trdm2 Real: coefficient for background tracer damping, the order of which is controlled by nord_tr. The appropriate values of trdm2 scale like d4_bg (for nord_tr > 0) or d2_bg (for nord_tr == 0). 0 by default. Disabled for values less than 1.e-4. Tracer damping is typically undesirable and should not usually be enabled. This option is added for some rare applications in which explicit diffusion to tracers may need to be specified.

vtdm4 Real: coefficient for background other-variable damping. The value of vtdm4 should be less than that of d4_bg. A good first guess for vtdm4 is about half the value of d4_bg. 0.0 by default. Disabled for values less than 1.e-3; requires do_vort_damp to be True. Other-variable damping should not be used if a monotonic horizontal advection scheme is used.

Legacy and obsolete options

non_ortho Logical: whether to use an orthogonal grid. This value is overridden for a cartesian grid (grid_type == 4). True by default; this value should not need to be set.

old_divg_damp Logical: reverts damping behavior back to that used by default in AM3, to reproduce results from earlier model versions. False by default.

use_old_omega Logical: whether to use the old calculation for omega, the pressure-coordinate vertical velocity. The “old” omega is computed directly from the definition of omega, , through the sum of the local, Eulerian change in pressure over a k_split timestep, and the advection of pressure. The “new” omega computation is to use the vertical integral of the continuity equation, to obtain the expression . The two forms will have slightly different results; the “new” omega is more consistent with the numerics in that it uses the same computed mass divergence used to update δp. True by default.

Entries in surf_map_nml

surf_file Character(len=128): File containing topography data. This file must be in NetCDF format. INPUT/topo1min.nc by default. (Previous versions of the model have used 5 minute USGS data, which is not recommended.)

nlon Integer: Size of the longitude dimension in topography data; not used.

nlat Integer: Size of the latitude dimension in topography data; not used.

zero_ocean Logical: whether to prevent the smoothing from extending topography out into the ocean. True by default.

zs_filter Logical: whether to apply smoothing to the topography. True by default.

Entries in fv_grid_nml

grid_name Character(len=80): Name of the grid either being read in (if grid_spec = -1) or being created. This is only used for writing out a binary file in the directory from which the model is run. 'Gnomonic' by default.

grid_file Character(len=120): If grid_type = -1 the name of the grid_spec file to read in. INPUT/grid_spec.nc by default; other values will not work.

Entries in test_case_nml

test_case Integer: number of the idealized test case to run. A number of nest cases are defined in tools/test_cases.F90, of which numbers 1–9 are intended for the shallow-water model. Requires warm_start = .false. 11 by default; this creates a resting atmosphere with a very basic thermodynamic profile, with topography. If you wish to initialize an Aquaplanet simulation (no topography) set to 14.

alpha Real: In certain shallow-water test cases specifies the angle (in fractions of a rotation, so 0.25 is a 45-degree rotation) through which the basic state is rotated. 0 by default.

Entries in nest_nml

ngrids Integer: number of grids in this simulation. 1 by default.

nest_pes Integer(100): array carrying number of PEs assigned to each grid, in order. Must be set if ngrids > 1.

p_split Integer: number of times to sub-cycle dynamics, performing nested-grid BC interpolation and (if twowaynest == .true.) two-way updating at the end of each set of dynamics calls. If p_split > 1 then it is recommended to appropriately decrease k_split so the remapping and dynamics timesteps remain the same. 1 by default.