

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](#)



Pre-publication GFDL Technical Memorandum

What is the purpose of this technical note?

This technical note explains updates to the GFDL Finite-Volume Cubed-Sphere Dynamical Core, abbreviated FV3 or FV³, and the Split GFDL Microphysics. It does not repeat the contents of earlier documentation, especially publications. [A list of publications and prior technical notes describing FV3 is available on the GFDL website.](#)

Where is the FV3 Code?

FV3 and the GFDL Microphysics is made publicly available through GitHub at https://github.com/NOAA-GFDL/GFDL_atmos_cubed_sphere. This code contains the initialization, solver, diagnostics, miscellaneous utilities, and interfaces for two physics drivers, the GFDL AM4 physics and the GFS Physics driver, used for prototypes of GFSv15 and in GFDL SHIELD (formerly fvGFS).

What is required to use FV3?

FV3 is a dynamical core and not a model. It requires compilation into another model compatible with the interfaces provided within FV3, and does *not* compile into a stand-alone executable. This code base may be compiled into a variety of models, including GFDL AM4, SPEAR, and SHIELD; GFSv15 and other UFS models; NASA GEOS and Harvard GEOS-Chem; CWBGFS; and LASG FGOALS, amongst others. Interfaces are provided to compile into both the GFDL AMx Climate Model Physics and into the Interoperable Physics Driver (IPD).

FV3 also requires the [Flexible Modeling System \(FMS\)](#) libraries to run. The FMS infrastructure may serve as either the complete model or may interface within another framework such as NEMS, MAPL, CESM, etc.

Attribution of codes and science

Publications should cite Putman and Lin (2007, JCP) and Harris and Lin (2013, MWR) when describing a model using the FV3 dynamical core. Cite Chen et al (2013, JCLim) and Zhou et al (2019, BAMS) if using the GFDL Microphysics. If references are made to specific algorithms or functionality within FV3 the respective authors should be credited and their paper(s) cited; see [FV3 Documentation and References](#) for a list of relevant papers describing the FV3 algorithms. GFDL reserves all publication rights to new and as-of-yet undocumented science within FV3 and the GFDL Microphysics.

Pre-publication GFDL Technical Memorandum

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. Contact GFDL FV3 support at oar.gfdl.fv3_dycore_support@noaa.gov for assistance and more information. Please note we can only offer limited support for software and configuration issues.

Changes from previous 201707 Release

- FV3 Initialization Updates:
 - Incorporated changes (*provided by Tom Black and Jim Abeles at EMC*) to `nggps_ic` to read GFSv15 analyses from `global_chgres`; in particular it determines whether the inputs are nonhydrostatic or not, and whether or not the definitions of tracers need to be changed to be compatible with FV3's definitions. The subroutine `get_nggps_ic` reads attributes from the input netcdf file to determine whether it is from GFSv15 or an earlier model.
 - Added capability to apply analysis increments to native-grid GFSv15 restart files
 - Model initialization (`fv_control.F90`, `fv_restart.F90`) has been completely re-written to be much simpler and easier to maintain. Nested-grid initialization in particular is now much cleaner and faster, with reproducibility issues easier to fix.
 - `fv_eta.F90` has been completely re-written to make level selection easier and less reliant upon compiler directives. Hard-coded level coefficients have been split off into a new file, `fv_eta.h`. There is now a namelist option, `fv_core_nml::npz_type`, to select between different configurations of the same number of model levels.
 - On-line topography filter has been updated to be feature-compatible with the preprocessor's topography generation and filtering steps. This is enabled only when both `fv_core_nml::external_ic` and `fv_core_nml::full_zs_filter` are true. *The topography generation and filtering will be described in a forthcoming paper.* The old filtering mechanism is still present.
- FV3 Grids Updates:
 - Nesting code has been completely re-written to ease maintenance and to allow new improvements to be more easily incorporated.
 - Incorporates regional domain capability (*provided by Tom Black and Jim Abeles at EMC*), including initialization and application of boundary conditions, and initialization of a regional domain by reading from disk. This version is current as of 201807.

Pre-publication GFDL Technical Memorandum

- *Experimental* support for FMS's multiple and telescoping nesting capability. (Note that this release does not yet incorporate developments from AOML in this regard.)
- A new cubed-sphere rotation, controlled by `fv_core_nml::do_cube_transform`, which rotates from the north pole instead of the south pole. This results in tile 6, in which nests are usually placed, being oriented with north at the "top", instead of at the bottom as in the current rotation.
- FV3 Science Updates:
 - More accurate energy calculation and restoration in total energy fixer.
 - Updated thermodynamic constants (heat capacity, latent heat coefficient).
 - Unnecessary halo for `delz` has been removed.
 - Minor stability improvements to sponge layers and `fv_sg`.
 - Miscellaneous performance enhancements.
- Split GFDL Microphysics Updates:
 - Corrected interface to `gfdl_cloud_microphys_driver()`, the driver of the Split GFDL Microphysics (called from the GFS Physics Driver).
 - Updated `gfdl_cloud_microphys_init()` and its interface to allow it to use the `INTERNAL_FILE_NML` internal namelist file reads.
- IPD/GFS Physics interface updates:
 - Renamed driver directories to mirror model development at GFDL.
 - Renamed 'mytile' to 'mygrid' in `atmosphere.F90` to make clear the distinction between "tiles" and "grids".
 - Added support for nudging to NCEP Analyses. This is only in the atmosphere; nudging of land and ocean to analyses must be done separately. This is enabled by the option `fv_core_nml::nudge` and controlled through the namelist `fv_nudge_nml`. (The namelist items `fv_core_nml::nudge_dz` and `fv_core_nml::nudge_qv` do *not* control this nudging and are for the adiabatic initialization instead.)
 - Removed unnecessary `fv_current_grid.F90`.
- FMS Infrastructure/Interface Updates:
 - Support for 2019 FMS Release; requires separate update of FMS code
 - Support for new FMS nesting API
 - Nest-to-parent communication (two-way updating) significantly optimized
- Diagnostics Updates:
 - New diagnostics for tendencies from the physics (`T_dt_phys`, `u_dt_phys`, `v_dt_phys`, `qv_dt_phys`, `ql_dt_phys`, `qi_dt_phys`), 3D layer-mean heights (`hght`), integrated microphysical tracer fields (`intqv`, `intql`, `intqi`, `intqr`, `intqs`, `intqg`), and vertical flux correlation terms (`uw`, `vw`, `hw`, `qvw`, `qlw`, `qiw`, `o3w`).
 - Improved range checking.
 - A `range_warn` violation is now a warning and no longer a fatal error.
 - Support for point sounding outputs, controlled by the new namelist `fv_diag_column_nml`. See example usage below.

Pre-publication GFDL Technical Memorandum

Please note that this release does not include the new positive-definite advection scheme or updated cloud-radiation interactions in the GFDL Microphysics.

Interface changes

- This codebase requires the 201912 update to FMS.
- The 'fvGFS' subdirectory under drivers/ has been renamed SHiELD to mirror GFDL model development. A convenience symlink called 'gfs_physics' is provided to this directory. Please update your build scripts as necessary.
- The external interfaces to the GFDL Microphysics (gfdl_cloud_microphys_driver and gfdl_cloud_microphys_init) have both been updated.
- The namelist nest_nml has been replaced by fv_nest_nml. The presence of a non-empty nest_nml will trigger a fatal error.
- The namelist nggps_diag_nml has been eliminated. The physics diagnostic output frequency (previously controlled by the namelist variable nggps_diag_nml::fdiag) should be specified outside of FV3.

The interface to the IPD (GFS Physics) is unchanged.

Namelist updates

fv_core_nml

- bc_update_interval (integer, hours): Interval between updates for external boundary conditions for a regional domain. 3 by default.
- do_cube_transform (logical): Applies same transform as when do_schmidt = True but rotates from the north pole instead of the south pole. If both do_cube_transform and do_schmidt are True the model throws a fatal error. False by default.
- full_zs_filter (logical): Whether to enable full topography filtering, compatible with off-line topography filter. This is only used when initializing the model with external_ic. False by default.
- n_sponge (integer): If non-negative controls the number of levels, counting from the top of the domain, on which the 2dz filter in fv_sg is applied. Note that this is different from the "sponge layer" applied in the uppermost two or three levels of the domain in FV3, which applies additional horizontal diffusion but not any explicit vertical diffusion. This value is overridden by a non-negative value of sg_cutoff. 0 by default.
- npz_type (character(24)): Controls which level setup to use when several vertical level configurations use the same number of levels. These are defined in fv_eta.F90; check there for defaults (ie. empty npz_type) and alternates. Empty by default.
- regional (logical): Controls whether the domain is a regional domain or not. If enabled, the model reads the regional domain and boundary conditions from files on disk and applies boundary conditions on the haloes of the domain. Note that the model does not

Pre-publication GFDL Technical Memorandum

yet have the capability to cold-start a regional domain, and so the grid files must exist or else a fatal error is raised. False by default.

- `sg_cutoff` (real, Pa): Controls the pressure above which the 2dz filter in `fv_sg` is applied, similar to the behavior of `rf_cutoff`. If this value is set to a non-negative value it overrides the value in `n_sponge`. -1 by default, which disables this option and uses `n_sponge` instead.

The options `parent_grid`, `parent_tile`, `ioffset`, `joffset`, and `refinement` have been eliminated, and replaced by similar functionality in `fv_nest_nml`

`fv_nest_nml`

- `grid_pes` (integer(:)): Number of processor cores (or MPI ranks) assigned to each grid. The sum of the assigned cores in this array must sum to the number of cores allocated to the model. Up to one of the first `ngrids` entries may be 0, in which case all remaining cores are assigned to it. 0 by default.
- `Grid_coarse` (integer(:)): Grid number of parent grid, if any. The first element is ignored; positive values in any successive element indicates that a new nested grid is to be created; the model continues to create grids until it finds a non-positive element. The total number of grids, `ngrids`, is determined from this array. -1 by default.
- `tile_coarse` (integer(:)): Absolute index of the tile (sub-component of a grid) a given grid is nested within. The first element is ignored. This is only useful when the parent grid has multiple tiles, such as for the six-tile cubed-sphere grid, or if the parent is a multi-tile nest. If `grid_coarse(n)` is a single-tile domain the coarse-grid tile can be determined automatically by setting it to a non-positive value. 0 by default.
- `nest_refine` (integer(:)): Refinement ratio, relative to the parent, for each nest. The first element is ignored. FV3 supports any integer refinement ratio for nesting, although values larger than 6 may cause stability issues. 3 by default.
- `nest_ioffsets` (integer(:)): Index within a tile or grid of the coarse grid cell closest to the local left-hand boundary with a nested grid cell within it. For example, if you have a coarse grid which is 12 grid cells long, and `nest_ioffset` for its child is 3, then the nested grid will have its lower-left corner located in the `i=3` grid cell.
- `nest_joffsets` (integer(:)): as for `nest_ioffsets` but in the local `y`-direction.

Note that `fv_nest_nml` need only appear in `input_nml` and not in the input namelist file for any of the child grids. The `p_split` functionality, present in earlier versions, has also been removed.

`fv_diag_column_nml`

- `do_diag_sonde` (logical): Enables sounding output specified by the namelist variables `diag_sonde*`. The output is intended to match the format of text files produced by the [University of Wyoming's text soundings](#), except that output is on uninterpolated model levels. See below for example usage of this functionality. False by default.

Pre-publication GFDL Technical Memorandum

- `sound_freq` (integer): Frequency, in hours, of `diag_sonde` column output. 3 by default.
- `runname` (character(100)): Name of the simulation. This is only to add the runname to the output to enable a user to easily determine the simulation that produced a certain sounding file.
- `diag_sonde_lon_in` (real, dimension(MAX_DIAG_COLUMN)): List of longitudes (in degrees) for the desired sounding points. Longitudes may be used as [0, 360] (GFDL style) or [-180,180] (NCAR style).
- `diag_sonde_lat_in` (real, dimension(MAX_DIAG_COLUMN)): As for `diag_sonde_lon_in` except for latitudes.
- `diag_sonde_names` (character(16), dimension(MAX_DIAG_COLUMN)): List of names for each sounding point.
- `do_diag_debug`, `diag_debug_lon_in`, `diag_debug_lat_in`, and `diag_debug_names`: analogous to the functionality of `do_diag_sonde`, etc. but outputs different diagnostics at every `dt_atmos` more appropriate for debugging problems that are known to occur at a specific point in the model. This functionality is only implemented for the nonhydrostatic solver.

Disabled/eliminated namelists

- `nggps_diag_nml` has been eliminated.
- `nest_nml` has been disabled in favor of the new `fv_nest_nml` namelist. Use of any options in this namelist will cause a fatal error.

Example usage of point-sounding output

This update of FV3 includes functionality to easily write out text soundings or column diagnostic data from the model. Note that `diagnostics_manager` also has the capability to write out any diagnostic (either single level or a full column) at a single point, specified through the `diag_table`, albeit as NetCDF data and not text files.

When enabled, the model will write out data from the nearest grid centroid to the specified latitudes and longitudes. Each point will receive a different file. Note that if a point lies near the edges of a decomposed domain (ie. near the edge of a domain assigned to a single processor) the model will write out two files, one for each adjacent column nearest to the specified point.

To enable the point-sounding output in this update, you can add the following namelist to `input.nml`:

```
&fv_diag_column_nml
  do_diag_debug = .T.
  do_diag_sonde = .T.
  diag_debug_names = 'ORD', 'Princeton'
  diag_debug_lon_in = 272., 285.33
  diag_debug_lat_in = 42., 40.36
```

Pre-publication GFDL Technical Memorandum

```
sound_freq = 1
diag_sonde_names = 'OUN', 'Amarillo', 'DelRio', 'Jackson', 'ILX', 'AtlanticCity', 'DodgeCity',
diag_sonde_lon_in = -97.47, -101.70, -100.92, -90.08, -89.34, -74.56, -99.97,
diag_sonde_lat_in = 35.22, 35.22, 29.37, 32.32, 40.15, 39.45, 37.77,
runname = C48.L79z11a.test
```

/

Which produces these files in the run directory:

```
DEBUG_point1.out.0018    DEBUG_point2.out.0010    DEBUG_point2.out.0018
Sounding_point1.out.0018 Sounding_point2.out.0016 Sounding_point4.out.0018
Sounding_point5.out.0018 Sounding_point6.out.0018
```

The DEBUG files contain output at every timestep, and print out the lowest third of the levels in the column. This behavior is hard-coded but can be modified if necessary. Every prognostic variable is printed out, including the staggered native-grid winds (so u and v are grid-relative and not earth-relative).

=====

PRINTING Princeton

DIAGNOSTICS

```
time stamp: 2016 August 1 0 7 30
DIAGNOSTIC POINT COORDINATES, point # 2
```

```
longitude = 287.027 latitude = 44.062
on processor # 10 : processor i = 12 , processor j = 47
```

k	T	delp	delz	u	v	w	sphum	cond
	K	mb	m	m/s	m/s	m/s	g/kg	g/kg
52	273.02	17.963	226	-2.010	-4.198	0.007	4.888	0.00001
53	275.16	17.848	221	-1.872	-3.935	0.007	5.140	0.00000
54	275.90	17.719	214	-1.711	-3.596	0.006	5.420	0.00000
55	277.35	17.555	207	-1.529	-3.133	0.006	5.791	0.00000
56	278.33	17.355	201	-1.372	-2.465	0.005	6.350	0.00000
57	279.03	17.117	194	-1.196	-1.838	0.005	6.982	0.00000
58	280.19	16.841	187	-0.897	-1.454	0.005	7.560	0.00000
59	280.78	16.527	180	-0.361	-1.220	0.005	7.990	0.00000
60	281.63	16.175	173	0.233	-1.140	0.006	8.283	0.00000
61	282.48	15.786	166	0.575	-1.199	0.006	8.525	0.00000
62	283.15	15.359	159	0.828	-1.202	0.006	8.758	0.00000
63	284.01	14.895	152	1.125	-0.980	0.007	8.983	0.00000
64	284.88	14.395	144	1.483	-0.557	0.007	9.189	0.00000
65	285.64	13.858	137	1.833	-0.025	0.007	9.339	0.00000
66	286.51	13.288	130	2.126	0.490	0.006	9.460	0.00000
67	287.52	12.683	122	2.387	0.922	0.006	9.609	0.00000
68	288.26	12.046	115	2.686	1.338	0.005	9.702	0.00000
69	288.96	11.378	107	2.929	1.685	0.004	9.817	0.00000
70	289.57	10.679	100	3.123	1.949	0.003	9.957	0.00000

Pre-publication GFDL Technical Memorandum

71	290.11	9.953	92	3.284	2.124	0.002	10.089	0.00000
72	290.62	9.200	84	3.382	2.204	0.002	10.216	0.00000
73	291.03	8.423	76	3.404	2.246	0.001	10.337	0.00000
74	291.35	7.622	69	3.329	2.184	0.001	10.499	0.00000
75	291.62	6.802	61	3.165	2.004	0.001	10.677	0.00000
76	291.85	5.962	53	2.960	1.793	0.000	10.855	0.00000
77	292.04	5.107	45	2.734	1.587	-0.000	11.044	0.00000
78	292.12	4.237	37	2.471	1.335	-0.000	11.238	0.00000
79	291.07	3.378	29	1.551	0.757	-0.000	11.418	0.00000

The sounding output in the files Sounding_point* is intended to match the format of text files produced by the [University of Wyoming's text soundings](#). They are printed out at the frequency specified by the variable sound_frequency (in hours). Output from the surface level, including 2-m temperature and 10-m winds, is not yet implemented.

```
OUN.v2016080101.i2016080100.C48.L79z11a.test.dat#####
OUN Valid 20160801.01Z          Init 20160801.00Z \nC48.L79z11a.test 263.115 35.946
```

PRES	HGHT	TEMP	DWPT	RELH	MIXR	DRCT	SKNT	THTA	THTE	THTV
hPa	m	C	C	%	g/kg	deg	knot	K	K	K
972.5	325	32.2	21.1	52	16.40	-105	11.62	307.8	357.6	310.8
969.2	360	32.6	20.6	49	15.91	-104	15.32	308.5	357.0	311.5
964.8	404	32.6	20.1	47	15.57	-102	17.55	308.9	356.5	311.8
959.5	456	32.4	19.8	47	15.28	-101	19.30	309.2	355.9	312.0
953.3	516	32.1	19.4	47	15.01	-99	20.78	309.4	355.4	312.2
946.0	585	31.6	19.0	47	14.80	-97	21.81	309.6	354.9	312.3
938.1	661	30.9	18.8	48	14.68	-95	22.35	309.7	354.7	312.4
929.3	746	30.2	18.5	49	14.62	-93	22.59	309.7	354.5	312.4
919.7	838	29.3	18.3	51	14.54	-90	22.71	309.8	354.3	312.5
909.5	939	28.4	18.0	53	14.45	-87	22.70	309.9	354.1	312.6
898.4	1047	27.5	17.7	55	14.34	-84	22.55	310.0	353.9	312.6
886.7	1164	26.4	17.4	57	14.23	-81	22.31	310.0	353.7	312.7
874.3	1288	25.3	17.1	60	14.13	-77	21.91	310.1	353.4	312.7
861.3	1419	24.1	16.7	63	13.98	-73	21.38	310.2	353.1	312.8
847.8	1558	22.9	16.1	65	13.74	-70	20.80	310.4	352.6	312.9
833.6	1705	21.7	15.6	68	13.46	-66	20.16	310.6	351.9	313.1
819.0	1859	20.3	14.9	71	13.13	-60	19.15	310.6	351.0	313.1
803.8	2020	19.0	14.1	72	12.67	-56	18.42	311.0	350.0	313.4
788.3	2188	17.8	13.1	73	12.07	-51	17.59	311.4	348.7	313.7
772.3	2363	16.3	11.8	74	11.32	-44	16.40	311.6	346.6	313.7
756.0	2545	15.3	10.3	71	10.44	-39	15.45	312.4	344.9	314.4
739.2	2734	14.3	8.4	67	9.41	-33	14.24	313.3	342.7	315.1
722.3	2930	12.7	6.0	63	8.15	-27	12.54	313.7	339.3	315.2
705.1	3133	12.3	2.7	51	6.59	-24	10.87	315.4	336.4	316.7
...										
93.4	17133	-66.1	-85.3	4	0.00	129	8.25	407.6	407.6	407.6
84.8	17721	-65.8	-83.9	5	0.00	157	11.37	419.7	419.7	419.7
76.7	18331	-65.1	-83.9	5	0.00	166	13.79	433.3	433.3	433.3

Pre-publication GFDL Technical Memorandum

69.1	18964	-64.1	-83.1	5	0.01	171	15.69	448.6	448.6	448.6
62.1	19624	-62.7	-82.3	5	0.01	175	17.38	465.5	465.5	465.5
55.6	20311	-61.3	-82.7	4	0.01	177	18.89	483.8	483.8	483.8
49.5	21026	-60.1	-82.7	3	0.01	176	20.87	502.8	502.8	502.8
44.0	21771	-58.7	-82.3	3	0.01	175	22.03	523.6	523.6	523.6
38.9	22548	-57.2	-82.1	2	0.01	173	23.27	546.3	546.4	546.3
34.1	23370	-55.6	-82.2	2	0.01	169	25.82	571.1	571.2	571.1
29.7	24269	-53.8	-85.2	0	0.01	167	29.23	599.2	599.2	599.2
25.3	25290	-52.0	-83.9	0	0.01	169	32.26	632.1	632.2	632.1
21.0	26502	-49.9	-82.5	0	0.01	175	34.22	672.9	673.0	672.9
16.7	28011	-47.4	-80.8	0	0.01	-178	34.87	726.4	726.5	726.4
12.5	30000	-43.8	-78.4	0	0.01	177	34.61	802.9	803.0	802.9
8.3	32789	-38.9	-75.0	0	0.01	172	37.02	921.1	921.2	921.1
4.5	37176	-29.3	-68.7	0	0.01	174	49.36	1140.9	1141.0	1140.9

Acknowledgements

We thank S-J Lin for his many years leading the FV3 Team and for his singular contributions to FV3 and to the science and practice of numerical modeling over the last 25 years. We also thank Rusty Benson and Kate Zhou (EMC) for their help in preparing this code release, and Whit Anderson for his support for developing the GFDL Technical Note and code release protocol. Tom Knutson and Alistair Adcroft provided useful reviews and edits of this document.