FV3GFS Public Release v1

NOAA/NWS/NCEP/EMC FV3GFS prototype
March 2018

Contents of release

- Scientific contents:

  - Code: FV3 dynamical core with GFS physics parameterizations, plus GFDL MP
  - Scientific documentation for forecast model including physics parameterizations.

- Technical contents:

  - Tasks supported: forecast-only workflow
    - Pre-processing
      - pull initial conditions off NOAA R&D hierarchical storage HPSS or from staged inputs on disk
      - convert initial conditions to FV3 grids
    - Forecast
    - Post-processing (master and global half-degree pressure-level GRIB2 files)
  - Workflow system
    - CROW front-end with rocoto
  - Codes:
    - Pre-processing (change resolution)
    - Forecast model (FV3GFS)
    - Post-processing (UPP)
  - Machine support for running on NOAA R&D machines with pre-built system and utility software:
    - Theia, Gaea, and Jet
  - Resolutions supported:
    - Horizontal: 13 km (C768), 25 km (C384), 50 km (C192)
    - Vertical: 64 layers covering troposphere and stratosphere
  - Scripts pulling initial conditions from NOAA R&D hierarchical storage HPSS. Supported initial conditions from Q3FY17 GFS:
    - Pre-implementation parallels (2014 - 2017)
    - Production (2017 to present)
  - Baselines for test cases on all three platforms.
  - Technical documentation for running workflow.

Installed software dependencies

Installed software that the V1 public release depends on through modules on RDHPCS machines Theia, Gaea, and Jet

- NCEPLIBS
  - ip/2.0.0
  - ip/3.0.0
  - bacio/2.0.1
  - w3nco/2.0.6
  - w3emc/2.2.0
  - g2/2.5.0
  - g2/3.1.0
  - g2tmpl/1.4.0
  - g2tmpl/1.5.0
  - nemsio/2.2.2
  - nemsio/2.2.3
  - bufr/11.0.1
  - sfcio/1.0.0
  - sp/2.0.2
  - landsfcutil/2.0.0
  - landsfcutil/2.1.0
  - sigio/2.0.1
  - gfsio/1.1.0
  - nemsiogfs/2.0.1
  - CRTM/2.0.6
- ESMF/7.1.0r
- ESMF/7.1.0r/impi (jet)
- netCDF3/3.6.3
- netCDF4
- HDF5
- WRF-shared/1.1.0
- Doxygen/1.8.10
- wgrib2/2.0.6
- jasper/1.900.1
- png/1.2.44
- png/1.2.49

- zlib/1.2.7
- szip

System software not installed by NCEP

- MPICH
- IMPI

Software built in this release

- global_chgres
- nemsfv3gfs (fv3_gfs_nh.prod.32bit.x)
- UPP (gfs_ncep_post)
- nemsio_read
- nemsio_get
- nemsio_cvt
- mkgfsnemsioctl

Supported cases

The following single cycle cases have been set up for your use:

Case	Dates	Resolutions	Settings
Atmospheric river over NE Pacific during ENRR campaign 2016	Initial conditions: 2016 February 10th 00z Forecast to: 2016 February 26th 00z	C192 C384	FHMAX_GFS=384 FHOUT_GFS=12
Hurricane Harvey	Initial conditions: 2017 August 17th 12z Forecast to: 2017 August 24th 12z	C192 C384 C768	FHMAX_GFS=168 FHOUT_GFS=6 FHMAX_HF_GFS=48 FHOUT_HF_GFS=3
March Nor'easters	Initial conditions: 2018 February 28th 00z Forecast to: 2018 March 10th 00z	C192 C384 C768	FHMAX_GFS=240 FHOUT_GFS=6

Home of prepped ICs used by system and baseline output:

- Theia: /scratch4/NCEPDEV/global/noscrub/glopara/FV3GFS_V1_RELEASE
- Jet: /lfs3/projects/hfv3gfs/Samuel.Trahan/FV3GFS_V1_RELEASE
- Gaea: /lustre/f1/pdata/ncep_shared/FV3GFS_V1_RELEASE

Resource requirements

Case	Storage	Compute	
Atmospheric River C192	42 GB ICs + 60 GB input+comrot	fv3ic/getic forecast post	02:30:00 cpu, 00:14:00 wall 15:00:00 cpu, 00:40:00 wall 00:30:00 cpu, 00:01:20 wall
Harvey C768	42 GB ICs + 850 GB input+comrot	fv3ic/getic forecast post	00:30:00 cpu, 00:06:00 wall 35:00:00 cpu, 01:30:00 wall 00:45:00 cpu, 00:03:00 wall
March Nor'easters C384	42 GB ICs + 250 GB input+comrot	fv3ic/getic forecast post	00:06:00 cpu, 00:05:00 wall 35:00:00 cpu, 01:30:00 wall 00:30:00 cpu, 00:02:00 wall

Note: Sample resource information for Theia only.

How to run the FV3GFS Public Release v1

Clone and build FV3GFS Public Release tag

> git clone https://github.com/NOAA-EMC/fv3gfs.git
    Alternative commands:
- If you have a GitHub account:
> git clone https://username@github.com/NOAA-EMC/fv3gfs.git
- if you add your ssh key to your GitHub account:
> git clone git@github.com:NOAA-EMC/fv3gfs.git

> cd fv3gfs
> git checkout public_release_v1   ← you now have a copy of the public_release_v1 tag
> cd sorc
> sh build_all.sh
> sh link_fv3gfs.sh

Cases Made Available

- Prepped cases:
  - AtmRiv_2016021000_GFS@C192
  - AtmRiv_2016021000_GFS@C384
  - Harvey_2017081712_GFS@C192
  - Harvey_2017081712_GFS@C384
  - Harvey_2017081712_GFS@C768
  - March_Noreasters_2018022800_GFS@C192
  - March_Noreasters_2018022800_GFS@C384
  - March_Noreasters_2018022800_GFS@C768
- Non-prepped cases:
  - Run-From-HPSS-ICs

Case types:

- Prepped cases: initial conditions are staged on disk ready for your use.
- Non-prepped cases: the "Run-From-HPSS-ICs" case runs additional pre-processing step to pull initial conditions from HPSS and converts them to the FV3 grid for use by the model. See information later in document for how to use the Run-From-HPSS-ICs case.

Set up experiment

> cd ../ecf/ecfutils

1. Choose the case you'd like to run. See /cases folder or case table for options.
2. Copy 'user.yaml.default' to 'user.yaml'
3. Edit user.yaml and make sure to set PROJECT_DIR, cpu_project, and hpss_project. Add paths for LONG_TERM_TEMP and SHORT_TERM_TEMP as desired. Descriptions of variables below sample user.yaml.

Sample user.yaml on Theia:

```
user_places: &user_places
  PROJECT_DIR: /scratch4/NCEPDEV/global
   # ie. /scratch4/NCEPDEV/ocean

  # Override scrub areas here.  Mandatory on Jet.
  LONG_TERM_TEMP: /scratch3/NCEPDEV/stmp1/Joe.Schmo
  # SHORT_TERM_TEMP: /lfs3/projects/hfv3gfs/Samuel.Trahan/scrub

  # Override non-scrubbed areas if desired:
  # HOMEDIR:  /lfs3/projects/hfv3gfs/Samuel.Trahan/homedir
  NOSCRUB: /scratch4/NCEPDEV/global/noscrub/Joe.Schmo

accounting: &accounting
  # Project for CPU accounting.
  cpu_project: fv3-cpu
  hpss_project: emc-global
  ecflow_machine: xc40-dev   # only relevant on WCOSS using ecFlow
```

Where…

- **PROJECT_DIR** is your top level folder on the machine for your non-scrubbed or save space.
- **LONG_TERM_TEMP** is your scratch space to use for your comrot output.
- **SHORT_TERM_TEMP** is your scratch space to use for your directories used at runtime. If left commented out the system will determine the best area to use, typically the one with the most available space.
- **HOMEDIR** is your personal space where you may have test codes outside of the clone and is optional. You can set HOMEDIR and then use $HOMEDIR in configs to point to your files.
- **NOSCRUB** is your non-scrubbed space where your expdir folder will be created. Can use save space here too, it's up to you.
- **cpu_project** is your accounting code on the platform for submitting jobs.

- **hpss_project** is your project used to access HPSS, necessary if running full test that pulls ICs from HPSS.
- **ecflow_machine** is the machine type information for ecFlow. Not relevant outside of WCOSS and can be left as is on other platforms.

4. Run setup_case.sh script under ecf/ecfutils:

> ./setup_case.sh (case-you-chose) (experiment-name)

a. The case-you-chose must be something from the /cases directory. The basename is sufficient (e.g. Harvey_2017081712_GFS@C192)
b. Your experiment directory (expdir) will be created and information will be printed to screen.
c. Modify configs in expdir as needed. User beware: Only default settings are currently supported in public release.

Example:

```
[Joe.Schmo@tfe03]$ ./setup_case.sh Harvey_2017081712_GFS@C768 pr_harvey768
Input conditions from model: GFS
Copy input conditions from: /scratch4/NCEPDEV/global/noscrub/glopara/FV3GFS_V1_RELEASE/ICs
Copy input conditions to: /scratch3/NCEPDEV/stmp1/Joe.Schmo/comrot/pr_harvey768

Case "Harvey_2017081712_GFS@C768" is set up under experiment name "pr_harvey768" with:

 YAML files:    /scratch4/NCEPDEV/global/noscrub/Joe.Schmo/expdir/pr_harvey768
 Config files:   /scratch4/NCEPDEV/global/noscrub/Joe.Schmo/expdir/pr_harvey768
 COM directory:  /scratch3/NCEPDEV/stmp1/Joe.Schmo/comrot/pr_harvey768

Now you should make a workflow:

 Rocoto: ./make_rocoto_xml_for.sh
/scratch4/NCEPDEV/global/noscrub/Joe.Schmo/expdir/pr_harvey768
 ecFlow: ./load_ecflow_workflow.sh
/scratch4/NCEPDEV/global/noscrub/Joe.Schmo/expdir/pr_harvey768
```

What the output from that command means:

- Initial conditions have been linked to from your comrot to staged files on disk. Nothing else is needed to set up your initial conditions.
- An experiment directory (expdir) has been created and populated with config/yaml files.
- Your next step is to now create the workflow with the provided command (rocoto for this release). See next section...

Set up workflow

5. Run make_rocoto_xml_for.sh script under ecf/ecfutils:

> ./make_rocoto_xml_for.sh /path/to/your/expdir

Example:

```
[Joe.Schmo@tfe03]$ ./make_rocoto_xml_for.sh
/scratch4/NCEPDEV/global/noscrub/Joe.Schmo/expdir/pr_harvey768
/scratch4/NCEPDEV/global/noscrub/Joe.Schmo/expdir/pr_harvey768/workflow.xml: Rocoto XML
document created here.
[Joe.Schmo@tfe03]
```

You now have a workflow xml file in your expdir that will run your chosen case.

Start run

> cd /path/to/your/expdir

Check rocoto module is loaded in your environment (module load rocoto). Version 1.2.4 is preferred rocoto version.

6. Run rocotorun to kick off experiment:

> rocotorun -d workflow.db -w workflow.xml
      (workflow.db is new and will be created here)

7. Add rocotorun to cron to keep experiment progressing forward. Example crontab entry on Theia:

```
*/5 * * * * /apps/rocoto/1.2.4/bin/rocotorun -d
/scratch4/NCEPDEV/global/noscrub/Joe.Schmo/expdir/pr_harvey768/workflow.db -w
/scratch4/NCEPDEV/global/noscrub/Joe.Schmo/expdir/pr_harvey768/workflow.xml
```

Monitor run

Monitor progress of run using rocotostat command from expdir:

> rocotostat -d workflow.db -w workflow.xml [-c CYCLES]

...where CYCLES can be "all" to show all cycles or specific cycle in format: YYYYMMDDCC00

Documentation on rocoto commands can be viewed here:
https://github.com/christopherwharrop/rocoto/wiki/Documentation

Manually Updating a Rocoto Workflow

Sometimes you may want to change the configuration of the workflow. There are two ways to do this:

1. Edit the yaml files and config files in the expdir
2. Edit the yaml files in the ecf/ecfutils directory and re-run setup_case.sh

After the yaml files have been edited, you must update the Rocoto XML file:

1. Stop your rocotorun cron or daemon
2. cd /path/to/ecf/ecfutils
3. ./make_rocoto_xml_for.sh $expdir
4. cd $expdir
5. continue running rocotorun as before

Additional information

- Throttles - The workflow has two throttle variables that control the forward progress:
    - Task throttle: Default = 5 tasks. This means no more than 5 tasks/jobs from the experiment will be queued/running at any given time. This value can be modified by manually changing it in the workflow.xml file.
    - Cycle throttle: Default = 2 cycles. This means no more than 2 cycles from the experiment will be active at any given time. Relevant if you modify the EDATE in your case yaml file to run more than just a single cycle. Currently the cases are set to run only one cycle by default. This value can be modified by manually changing it in the workflow.xml file.
- Run-From-HPSS-ICs case
    - Open ecf/ecfutils/cases/Run-From-HPSS-ICs.yaml

- Modify CASE, FHMAX_GFS, FHOUT_GFS, FHMAX_HF_GFS, FHOUT_HF_GFS, and SDATE to the date and settings you'd like to run. See documentation in yaml file for information on variables.
- Modify EDATE to use a similar convention as SDATE if you'd like to extend your run beyond the initial cycle.
- Follow the setup directions from the prepped cases and make sure to use "Run-From-HPSS-ICs" as your case name.
- Note: You can only run Run-From-HPSS-ICs for dates covered by the Q3FY17 pre-implementation parallels or operations since the last upgrade (mid-2014 to present). Be aware, there are some holes in that time period for the dates not covered by the pre-implementation parallels.

Troubleshooting - common issues encountered

Workflow

- Yaml file indentation (e.g. user.yaml) - do not indent with tabs and maintain the space indent as shown in user.yaml. Will get yaml syntax error if indentation is incorrect.

Jet

- The default stack size on Jet is very small. You need to ulimit or limit to a larger size or many things will break. Suggest to add "unlimit" to your .*rc file.

Gaea

- Use the gaea-support-1.2.4 branch for rocoto, which is installed here:
  module use /lustre/f1/unswept/ncep/Samuel.Trahan/rocoto/modulefiles/
  module load rocoto/gaea-support-1.2.4