Initial Report on Advanced Physics Test

Global Model Test Bed (GMTB)

February 17 2019 - DRAFT

Points of Contacts: Ligia Bernardet (ligia.Bernardet@noaa.gov) and Grant Firl (grantf@ucar.edu)

Additional GMTB staff that contributed to this activity: Jimy Dudhia, Mike Ek, Michelle Harrold, Judy Henderson, Tracy Hertneky, Weiwei Li, Linlin Pan, Jamie Wolff, and Man Zhang.

Table of Contents

Introduction

With the goal of supporting EMC in selecting an advanced physics suite for the GFS v16, due to be implemented operationally in 2021, the GMTB tested four configurations of NOAA's Unified Forecast System (UFS). The runs were started on 04 December 2018 and were completed on 09 February 2019. This test is being conducted collaboratively between the various groups as described below:

- GMTB: Conduct model runs. Produce verification and diagnostics to complement EMC's
- NOAA Environmental Modeling Center (EMC): Provide three model configurations (one baseline and two advanced physics configurations), the workflow, conduct verification and diagnostics, prepare the test plan.
- NOAA NOAA Earth System Research Laboratory (ESRL) Global Systems Division (GSD): Provide a model configuration using an advanced physics configuration.
- Navy Research Laboratory (NRL), NOAA ESRL Physical Sciences Division (PSD), and NCAR: Participate in an independent expert panel to evaluate test results.

This report details how the runs were conducted and what has been delivered by GMTB as of 15 February 2019. GMTB will issue a second report in the future with the results of the verification and diagnostics it conducted for this test.

Model configurations

The experiment consisted of four configurations of the GFS, all of which employed the Finite-Volume Cubed-Sphere dynamical core (FV3GFS). One of these was derived from the operational Rapid Refresh and High-Resolution Rapid Refresh (RAP/HRRR) modeling systems and was assembled and developed by GSD from years of community contributions through the WRF community modeling system for mesoscale applications (Suite 4 in Table 1), while the second candidate (Suite 3 Table 1) has components that were developed at multiple research centers and universities, including Colorado State University, University of Utah, NASA, NCAR, and EMC. Its individual parameterizations have been applied primarily to medium-range and longer prediction scales. The performance of these two suites was compared to the soon-to-be-operational GFS v15 (Suite 1 in Table 1) suite and a very similar, but updated, version of the GFS v15 suite (Suite 2 in Table 1). Note that, while both Suites 3 and 4 are aerosol aware (AA), Suite 3 uses a constant and horizontally homogeneous aerosol distribution, while Suite 4 uses a time-evolving aerosol distribution that is initialized from a monthly climatology. Table 1 also contains information about the suite configurations beyond the choice of physics. Note that, for physical processes not listed in Table 1, e.g., atmospheric radiation, gravity wave drag, ozone and stratospheric water photochemistry, GFS v15 parameterizations were used in all suites. Furthermore, note that Suite 4 was running using the Common Community Physics Package (CCPP), while the other suites did not employ this framework.

	Suite 1 (GFS v15)	Suite 2	Suite 3	Suite 4
Deep convection	SA-SAS	SA-SAS	CS-AW	SA/AA-GF
Shallow convection	SA-MF	SA-MF	SA-MF	MYNN-EDMF and SA GF
Microphysics	GFDL	GFDL	AA-MG3	AA-Thompson
Saturation adjustment in dycore	True	True	True	False
PBL/Turbulence	K-EDMF	SA-TKE-EDMF	K-EDMF	MYNN-EDMF
Land Surface Model	Noah	Noah	Noah	RUC
Physics-Dynamics coupling	non-CCPP	non-CCPP	non-CCPP	CCPP
nord	2	2	2	3
dddmp	0.1	0.1	0.1	0.2
d4_bg	0.12	0.12	0.12	0.15
vtdm4	0.02	0.02	0.02	0.06
sponge	10	10	26	10
tau	10	10	5	10
hord_mt	5	5	6	5
hord_ct	5	5	6	5
hord_tm	5	5	6	5
hord_dp	-5	-5	-6	-5
Platform	xjet	xjet	xjet	vjet
Intel compiler	v15	v15	v15	v18
Nodes/PPN	72/12	72/12	72/12	108/16
Layout	8x16	8x16	8x16	16x16
Threads	2	2	2	72
Code base	Oct 2018	Nov 2018	Nov 2018	Nov 2018

Table 1. Physics suite, dynamics namelist options (latter indicated by blue color fill), computational options (indicated by green color), and code base (indicated by yellow color) for preliminary advanced physics testing. Acronyms are defined in Appendix A. The definition of the dynamics namelist options can be found at https://www.gfdl.noaa.gov/wp-content/uploads/2017/09/fv3_namelist_Feb2017.pdf.

Model initialization

All model integrations used atmospheric initial conditions (ICs) based on the ECMWF model to eliminate model-climate biases that might be introduced if ICs from the Global Data Assimilation System (GDAS), which uses GFS physics, were applied. ECMWF ICs were obtained with the help of NOAA Physical Sciences Division and interpolated to the FV3GFS grid by EMC prior to the start of the test.

Soil ICs were taken from the retrospective runs of FV3GFS done during the testing geared to the GFS v15 implementation. The initialization for the RUC LSM was done internally in the FV3GFS model by interpolating the information from the levels supplied (the four levels of Noah LSM) to the RUC levels.

Aerosol ICs were suite-specific: suite 3 was initialized with a horizontally-homogeneous and time-invariant distribution set in the model code itself, while Suite 4 was initialized from a space- and time-dependent climatology of water- and ice-friendly aerosols (Thompson and Eidhammer, 2014; https://journals.ametsoc.org/doi/full/10.1175/JAS-D-13-0305.1). The GSD group provided the aerosol climatology file QNWFA_QNIFA_SIGMA_MONTHLY.dat.nc, which is also distributed with the Weather Research and Forecast model. GSD also provided a conversion utility to interpolate the aerosol climatology horizontally and vertically to the FV3GFS grid.

Before the start of the test, EMC created one IC file for each case, containing the atmospheric, soil, and aerosol fields. All suites were initialized from the same files, with aerosol information ignored by Suites 1, 2, and 3.

Forecast Cases

Runs were initialized every five days between 1 January 2016 and 31 December 2017, alternating between 00 and 12 UTC. In addition, 16 cases studies were added under the recommendation of the EMC Model Evaluation Group. The list of cases and a brief description can be found below.

- Tropical Cyclone (TC) cases
  - 10/1/15 00z TC Joaquin and flooding in SC
  - 10/2/16 00z TC Matthew
  - 8/26/17 00z TC Harvey
  - 9/7/17 00z TC Irma
  - 10/4/17 00z TC Nate
  - 8/19/18 00z TC Lane
  - 9/11/18 12z TC Florence
  - 7/31/17 00z TC Noru
- Other cases
  - 1/18/16 12z Blizzard of 2016 - progressive
  - 4/22/16 00z Plains severe weather - progressive, also a chance to examine drylines
  - 3/10/17 00z "Pi Day" Blizzard - Precipitation type
  - 4/20/17 00Z Valley flooding in MS

- 7/29/17 00z Too hot in FV3GFS in CA
- 10/16/17 12z Inversions and 2-m temperature
- 1/1/18 00z "Bomb" cyclone
- 3/15/17 00z Atmosphere river - progressive

Workflow and computational platform

The runs were computed on the NOAA research and development platform *Jet* using project *hfv3gfs*. A combination of individual user accounts and the *Role.fv3physics* role account was used to execute the runs. As described in Table 1, the *xjet* partition of *Jet* was used to run Suites 1-2-3, while the *vjet* partition was used to run Suite 4. Two partitions of *Jet* were employed to expedite the execution of the runs.

A simplified workflow was supplied by EMC to execute the runs. The workflow consisted of four distinct tasks: 1) establishing the directory structure and submission scripts for each case, 2) running the model, 3) running the unified post-processor and archiving the output, and 4) creating sounding output in BUFR format and archiving the output. In order to more efficiently run all of the cases, each suite was split into eight streams such that eight cases could be run concurrently for each suite. The workflow consisted of a set of *bash* scripts, with one script submitting the next. Once a case had been successfully run through all components of the workflow, file *case_list.txt* would be updated to remove the top date in the list and continue on to the next date.

The code repository for the workflow is at https://github.com/NCAR/emc-physics-workflow. Note that only directories 00_run, 02_mode, 03_post, and 04_bufr were used for conducting the runs. Directory 01_IC was set to generate the aerosol ICs, but later it was decided to generate the aerosol ICs ahead of the test. Directory 05_tracker was used for TC tracking, which was run after the test was complete.

Source codes and compilation

The executables were provided by GSD (Suite 4 NEMSfv3gfs) and EMC (Suites 1-2-3 NEMSfv3gfs, postprocessor, and BUFR soundings). They have provided GMTB with the information below regarding the source code and compilation.

Model

Suite 1:

VLab repository: NEMSfv3gfs; tag nemsfv3gfs_beta_v1.0.12 (implementation version) and submodules therein.

compile.sh *directory* jet " HYDRO=N 32BIT=Y" 1

Suites 2, 3:

VLab repository: NEMSfv3gfs; tag NEMSfv3gfs_suite23
VLab repository FV3; tag NEMSfv3gfs_suite23
VLab repository NEMS: tag NEMSfv3gfs_suite23

```
compile.sh directory jet " HYDRO=N 32BIT=Y" 1
```

Suite 4:

https://github.com/NCAR/NEMSfv3gfs/tree/gsd_suite4_physics_test_tag_20181210
https://github.com/NCAR/NEMS/tree/gsd_suite4_physics_test_tag_20181210
https://github.com/NCAR/FV3/tree/gsd_suite4_physics_test_tag_20181210
https://github.com/NCAR/ccpp-physics/tree/gsd_suite4_physics_test_tag_20181210
https://github.com/NCAR/ccpp-framework/tree/gsd_suite4_physics_test_tag_20181210
https://github.com/NCAR/FMS/tree/gsd_suite4_physics_test_tag_20181210

```
./compile.sh directory 'OPENMP=N INTEL18=Y 32BIT=Y SION=Y CCPP=Y HYBRID=N STATIC=Y SUITE=suite_FV3_GSD.xml'
```

Unified post-processor

VLab repository: EMC_post; tag ncep_post.v8.0.27
Two files were changed to be able to post-process MG3: GFIP3.f and INITPOST_GFS_NEMS_MPIIO.f. The differences are recorded in Appendix B.

Aerosol tools and climatology

https://github.com/NCAR/aeroconv; hash # 6d3aa

Archives

The following archives have been created:
Initial conditions: /NCEPDEV/emc-meso/5year/Ratko.Vasic/FV3_IFC_GMTB
Post-processed files: /BMC/gmtb/5year/Phys_Test_FV3GFSv2/POST_ARCH
Soundings in BUFR format: /BMC/gmtb/5year/Phys_Test_FV3GFSv2/BUFR_ARCH

Computation Cost

The computational cost for each suite is shown in Table 2. Note that Suites 1-2-3 were run on xjet (72 notes with 12 processors per node, two threads) and Suite 4 used vjet (108 nodes with 16 processors per node, no threading). The jet partitions xjet and vjet differ in a number of ways (https://jetdocs.rdhpcs.noaa.gov/wikis/jetdocs/doku.php?id=system_information) so the comparison among Suites 1-2-3 is valid, but there are inconsistencies when comparing against Suite 4.

SUITE	Average Time	Minimum Time	Maximum Time
1	2h22m	2h19m	2h31m
2	2h18m	2h16m	2h23m
3	4h25m	4h20m	4h49m
4	3h38m	3h35m	3h50m

Table 2. Compute time for each suite, including he average, minimum, and maximum time. Note that Suite 4 used a different computational platform, so the timing cannot be compared against the other suites.

Failures

The model run for Suite 3, case, 2016051000, crashed with a segmentation fault. The developer was contacted and diagnosed a shock resulting from the ICs. The developer requested two changes in namelist for this case: fhouri=2 and fac_n_spl=2, which result in a decrease of the dynamics time step by half for the first two hours of simulation. With these changes the model ran to completion.

During the execution of the test, numerous crashes occurred with the model, post processor, and BUFR executables (Table 3). The crashes occurred for all suites, but it should be noted that only post processor and BUFR soundings crashes occurred with Suite 4. Upon rerunning without any changes, sometimes repeatedly, all crashes were resolved (with the exception of Suite 3, case 2016051000 as described above). In other words, the crashes were not reproducible. A test indicated that, when the model ran to completion, its results were reproducible when it was run a second time.

Suite 1	Suite 2	Suite 3	Suite 4
38.7%	47.2%	36.7%	30.1%

Table 3. Number of cases with at least one failure reported in the process.

The large number of crashes that occurred in this test speak of an instability in the codes, in the *Jet* computational platform, or in a combination of the two. The *Jet* helpdesk was contacted and informed that other users, running other modeling systems, were not experiencing problems. These codes run operationally (on *wcoss*) and in research mode (in other NOAA R&D platforms, such as *Theia*) without problems. This instability caused a waste of computational resources. Moreover, this instability was a burden to the staff running the test, because the failed runs had to be diagnosed and re-started manually.

The workflow utilized in this test was simply a series of shell scripts that were initiated upon successful completion of the prior step. In principle this is sufficient; however, in practice additional checks and dependencies could be put in place to more efficiently manage the runs. With the simplified approach used, if any particular step in the process failed, the runs would be halted until manual intervention was applied. This required frequent checks of the state of the runs to ensure progress. During normal working hours this was acceptable, though, time consuming; however, during evenings and weekends it was an onerous task to ensure the runs continued. Even if checks were initiated every few hours it was possible that a task crashed right after a check and nothing ran for several hours, slowing the overall progress. Thought should be given to a more sophisticated workflow management system that could monitor status and task dependencies in a more automated fashion.

Appendix A: Acronyms

AA: Aerosol Aware
BUFR: Binary Universal Form for Representation of Meteorological Data
CCPP: Common Community Physics Package
ECMWF: European Centre for Medium-Range Weather Forecasts
EMC: Environmental Modeling Center
FV3: Finite-Volume Cubed-Sphere dynamical core
FV3GFS: Version of the GFS that employs the FV3 dynamical core
IC: Initial Conditions
K-EDMF: Hybrid Eddy-Diffusivity Mass-Flux PBL parameterization that employs K-theory
ESRL: NOAA Earth System Research Laboratory
GDAS: Global Data Assimilation System
GF: Grell-Freitas cumulus parameterization
GFDL: NOAA Geophysical Fluid Dynamics Laboratory
GFS: Global Forecast System
GMTB: Global Model Test Bed
GSD: Global Systems Division
HPSS: High Performance Storage System
MF - Mass Flux
MG3 - Morrison-Gettelman microphysics parameterization version 3
MYNN - Mellor-Yamada-Nakanishi-Niino PBL parameterization
NCAR: National Center for Atmospheric Research
NEMS: NOAA Environmental Modeling System
NRL: Navy Research Laboratory
NOAA: National Oceanic and Atmospheric Administration
PBL: Planetary Boundary Layer
PPN: Processor Per Node
RRTM: Rapid Radiative Transfer Model
RRTMG: RRTM for General Circulation Models
SA: Scale Aware
SAS: Simplified Arakawa-Schubert cumulus parameterization
TKE: Turbulent Kinetic Energy
TKE-EDMF: EDMF PBL parameterization based on TKE
TC: Tropical Cyclone
UPP: Unified Post Processor
UFS: Unified Forecast System
UTC: Coordinated Universal Time
VLab: NOAA's Virtual Laboratory

Appendix B. Modification in the post code

File GFIP3.f (< is the new code; > is the original code)

```
<       topIdx = -1
<       baseIdx = -1
---
>       topIdx = nz
>       baseIdx = nz
403c403
<          if (totalWater(m) > 0.001) then
---
>          if (totalWater(k) > 0.001) then
410,411c410,411
<       lp300: do m = k, nz
<          if (totalWater(m) > 0.001) then
---
>       lp300: do m = k, nz-1, -1
>          if (totalWater(k) > 0.001) then
419d418
<       if(topIdx == -1 .or. baseIdx == -1) cycle
462c461
<    else if(imp_physics == 11 .or. imp_physics == 8) then
---
>    else if(imp_physics == 11 .or. imp_physics == 8.or. imp_physics == 10) then
2218c2217
<   else if(imp_physics == 11 .or. imp_physics == 8) then
---
>   else if(imp_physics == 11 .or. imp_physics == 8.or. imp_physics == 10) then
```

File INITPOST_GFS_NEMS_MPIIO.f (< is the new code; > is the original code)

```
<             cnvctzgdrag, sconvmois, cnvctmgdrag, cnvctdetmflx, duwt, duem, dusd, dudp
---
>             cnvctzgdrag, sconvmois, cnvctmgdrag, cnvctdetmflx, duwt, duem, dusd, dudp,   &
>             ref_10cm
146a148
>      real dtp !physics time step
794c796
<       else if(imp_physics==11 .or. imp_physics==8.or. imp_physics==10)then ! GFDL or Thompson MP scheme
---
>       else if(imp_physics==11 .or. imp_physics==8)then ! GFDL or Thompson MP scheme
954a957,1003
> ! cloud fraction
>        VarName='cld_amt'
>        call getrecn(recname,reclevtyp,reclev,nrec,varname,VcoordName,l,recn)
>        if(recn /= 0) then
>          fldst = (recn-1)*fldsize
> !$omp parallel do private(i,j,js)
>          do j=jsta,jend
>            js = fldst + (j-jsta)*im
>            do i=1,im
>              cfr(i,j,ll)=tmp(i+js)
>            enddo
>          enddo
> !          if(debugprint)print*,'sample l ',VarName,' = ',ll, &
> !             cfr(isa,jsa,ll)
>        endif
```

```
>
>        if(imp_physics == 99)then
>         allocate(p2d(im,lm),t2d(im,lm),q2d(im,lm),cw2d(im,lm), &
>         qs2d(im,lm),cfr2d(im,lm))
>         do j=jsta,jend
>          do k=1,lm
>           do i=1,im
>            p2d(i,k)  = pmid(i,j,k)*0.01
>            t2d(i,k)  = t(i,j,k)
>            q2d(i,k)  = q(i,j,k)
>            cw2d(i,k) = cwm(i,j,k)
>            es = min(fpvsnew(t(i,j,k)),pmid(i,j,k))
>            qs2d(i,k) = eps*es/(pmid(i,j,k)+epsm1*es)!saturation q for GFS
>           enddo
>          enddo
>          call progcld1 &
> !...................................
> !  ---  inputs:
>             ( p2d,t2d,q2d,qs2d,cw2d,im,lm,0,                        &
> !  ---  outputs:
>               cfr2d                                                 &
>              )
> !$omp parallel do private(i,k)
>          do k=1,lm
>           do i=1,im
>            cfr(i,j,k) = cfr2d(i,k)
>           enddo
>          end do
>         end do
>         deallocate(p2d,t2d,q2d,qs2d,cw2d,cfr2d)
>        end if
>
978a1028,1050
> ! Read model derived radar ref.
>       VarName='ref3D'
>        recn = 0
>        call getrecn(recname,reclevtyp,reclev,nrec,varname,VcoordName,l,recn)
>        if(recn /=0 ) then
> !$omp parallel do private(i,j,js)
>          do j=jsta,jend
>            js = fldst + (j-jsta)*im
>            do i=1,im
>              ref_10cm(i,j,ll) = tmp(i+js)
>            enddo
>          enddo
>        else
> !$omp parallel do private(i,j)
>          do j=jsta,jend
>            do i=1,im
>              ref_10cm(i,j,ll) = spval
>              end do
>            end do
>            if(me==0)print*,'fail to read ', varname,' at lev ',ll
>          endif
>         if(debugprint)print*,'sample l ',VarName,' = ',ll,ref_10cm(isa,jsa,ll)
>
1597a1670,1677
>       VarName='dtp'
>       call nemsio_getheadvar(ffile,trim(VarName),dtp,iret)
>       if (iret /= 0) then
>        print*,VarName," not found in file-Assigned 225. for dtp as default"
>        dtp=225.
>       end if
>       if (me == 0) print*,'dtp= ',dtp
>
1757a1838,1840
> !      NPHS=2.
> !      DT=80.
>       DTQ2 = DTP   !MEB need to get physics DT
1759,1760c1842
<       DT=80.
<       DTQ2 = DT * NPHS  !MEB need to get physics DT
---
>       DT=DTQ2/NPHS
1777c1859
<           cprate(i,j) = avgcprate(i,j)
---
> !wm          cprate(i,j) = avgcprate(i,j)
1845a1928,1945
> ! unit of prec and cprate in post is supposed to be m per physics time step
> ! it will be converted back to kg/m^2/s by multiplying by density in SURFCE
>       do j=jsta,jend
>         do i=1,im
>           if (prec(i,j) /= spval) prec(i,j) = prec(i,j) * (dtq2*0.001) &
>                  * 1000. / dtp
>         enddo
>       enddo
>
> ! convective precip rate in m per physics time step
>       VarName='cnvprcp'
> !     VcoordName='sfc'
> !     l=1
>       call assignnemsiovar(im,jsta,jend,jsta_2l,jend_2u                &
>                          ,l,nrec,fldsize,spval,tmp                  &
>                          ,recname,reclevtyp,reclev,VarName,VcoordName &
>                          ,cprate)
> !$omp parallel do private(i,j)
1848c1948,1949
<           if (prec(i,j) /= spval) prec(i,j) = prec(i,j) * (dtq2*0.001)
---
```

```
>         if (cprate(i,j) /= spval) cprate(i,j) = max(0.,cprate(i,j)) * (dtq2*0.001) &
>                 * 1000. / dtp
1850a1952
>       if(debugprint)print*,'sample ',VarName,' = ',cprate(isa,jsa)
2013,2042d2114
<       allocate(p2d(im,lm),t2d(im,lm),q2d(im,lm),cw2d(im,lm),          &
<               qs2d(im,lm),cfr2d(im,lm))
<       do j=jsta,jend
< !$omp parallel do private(i,k,es)
<         do k=1,lm
<           do i=1,im
<           p2d(i,k)  = pmid(i,j,k)*0.01
<           t2d(i,k)  = t(i,j,k)
<           q2d(i,k)  = q(i,j,k)
<           cw2d(i,k) = cwm(i,j,k)
<           es = min(fpvsnew(t(i,j,k)),pmid(i,j,k))
<           qs2d(i,k) = eps*es/(pmid(i,j,k)+epsm1*es)!saturation q for GFS
<           enddo
<         enddo
<         call progcld1                                           &
< !...................................
< !  ---  inputs:
<              ( p2d,t2d,q2d,qs2d,cw2d,im,lm,0,                           &
< !  ---  outputs:
<               cfr2d                                           &
<              )
< !$omp parallel do private(i,k)
<         do k=1,lm
<           do i=1,im
<             cfr(i,j,k) = cfr2d(i,k)
<           enddo
<         end do
<       end do
<       deallocate(p2d,t2d,q2d,qs2d,cw2d,cfr2d)
<
2700c2772,2774
<       VarName='vgtyp'
---
> !      VarName='vgtyp'
> !Use for fv3 model output
>       VarName='vtype'
```