

Cycling DA with MPAS-JEDI and overview on graphics package

Ivette Hernández Baños

*Mesoscale & Microscale Meteorology Laboratory
National Center for Atmospheric Research*

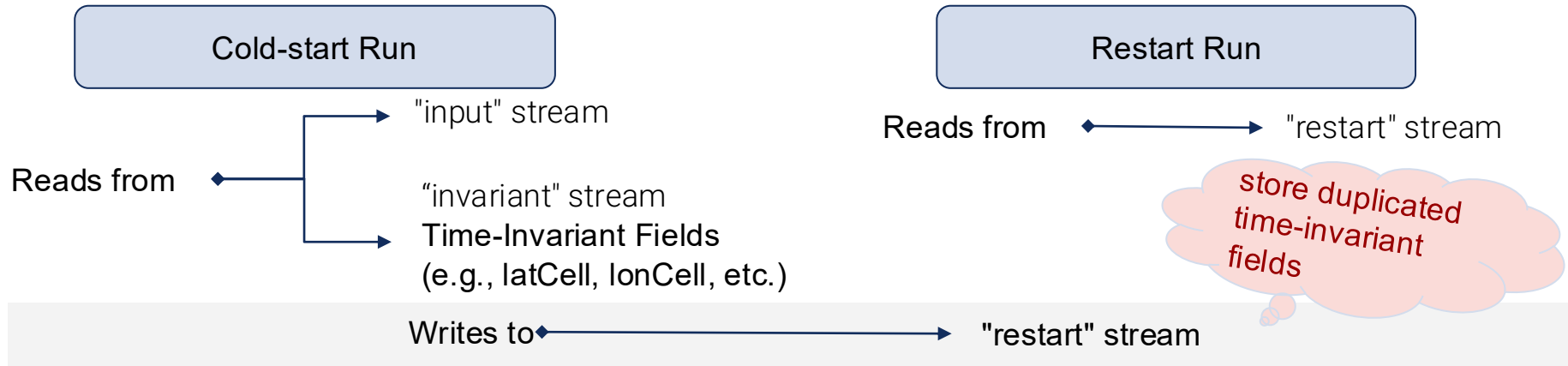


NCAR MPAS-JEDI Tutorial, Boulder, CO
13–14 August 2026

Outline

- ❖ Overview on the two-stream I/O (input/output) approach
 - 2-stream I/O in MPAS-JEDI
- ❖ Cycling DA with MPAS-JEDI
- ❖ Graphics package
 - Functionalities
 - Examples

2-stream I/O approach



In Large Ensembles:

- Each member writes its own restart file
- Time-invariant fields are **replicated**
- This leads to **redundant storage!**



Optimization approach:

- Store time-invariant fields **once**
- Reference shared fields across ensemble members

Running cycling DA more efficiently!!!

2-stream I/O approach

Real-data initial conditions

1. Processing static fields

In namelist.init_atmosphere:

```
&preproc_stages
  config_static_interp = true
  config_native_gwd_static = true
  config_vertical_grid = false
  config_met_interp = false
  config_input_sst = false
  config_frac_seaice = false
```

In streams.init_atmosphere:

```
<immutable_stream name="input"
  type="input"
  filename_template="x1.40962.grid.nc"
  input_interval="initial_only" />

<immutable_stream name="output"
  type="output"
  filename_template="x1.40962.static.nc"
  packages="initial_conds"
  output_interval="initial_only" />
```

2. Interpolating meteorological fields

In namelist.init_atmosphere:

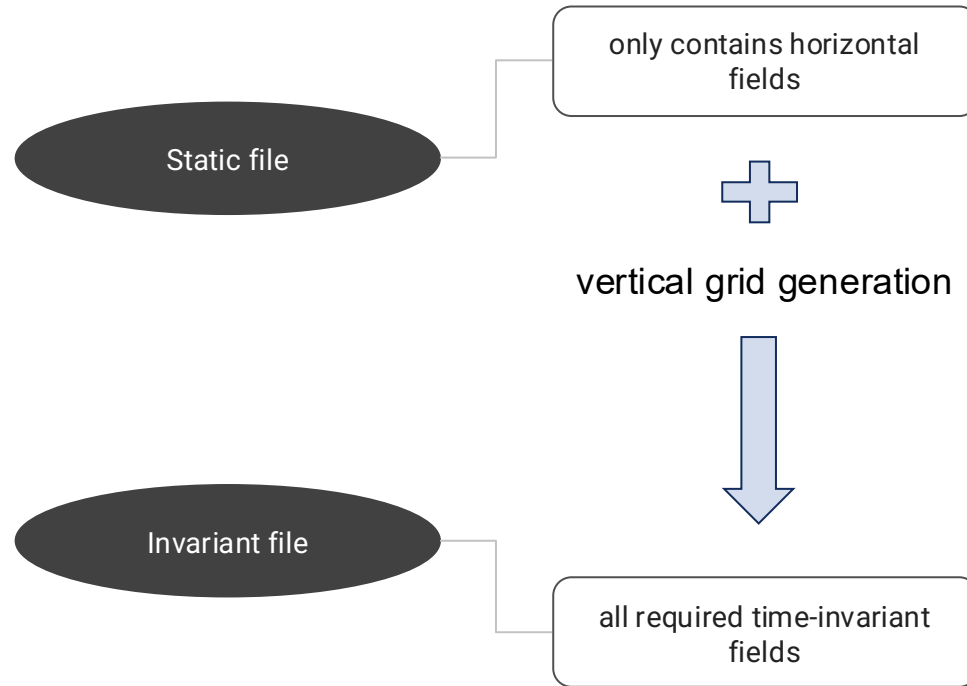
```
&preproc_stages
  config_static_interp = false
  config_native_gwd_static = false
  config_vertical_grid = true
  config_met_interp = true
  config_input_sst = false
  config_frac_seaice = true
```

In streams.init_atmosphere:

```
<immutable_stream name="input"
  type="input"
  filename_template="x1.40962.static.nc"
  input_interval="initial_only" />

<immutable_stream name="output"
  type="output"
  filename_template="x1.40962.init.nc"
  packages="initial_conds"
  output_interval="initial_only" />
```

2-stream I/O approach



namelist.atmosphere:

```
&preproc_stages
  config_static_interp = false
  config_native_qwd_static = false
  config_vertical_grid = true
  config_met_interp = false
  config_input_sst = false
  config_frac_seaice = false
/
```

streams.init_atmosphere:

```
<immutable_stream name="input"
  type="input"
  filename_template="x20.835586.static.nc"
  input_interval="initial_only" />

<immutable_stream name="output"
  type="output"
  filename_template="x20.835586.invariant.nc"
  packages="initial_conds"
  output_interval="initial_only" />
```

2-stream I/O in MPAS-JEDI

→ We use “mpasout” file instead of “restart” file for background and analysis files

New stream

```
<immutable_stream name="da_state"  
  type="output"  
  precision="single"  
  clobber_mode="truncate"  
  filename_template="mpasout.$Y-$M-$D_$h.$m.$s.nc"  
  packages="jedi_da"  
  io_type="pnetcdf,cdf5"  
  output_interval="6:00:00" />
```

*time invariant fields in a separate file
*mpasout file **excludes** time invariant fields in the invariant file and also physical tendency fields

Dual role in DA cycling—input and output—updated through the analysis process

Ha et al. (2024): <https://doi.org/10.5194/gmd-17-4199-2024>

2-stream I/O in MPAS-JEDI

MPAS-JEDI reads in two streams (two files):

- ❑ “**invariant**” stream: mesh info, sfc input variables (landmask, soilcomp, ivgtyp, albedo12m, etc) and parameters for gravity wave drag over orography, vertical coordinate etc.
- ❑ “**da_state**” stream (i.e., ‘mpasout’ file): fields needed for DA purposes (either analysis variables or fixed input needed for CRTM or other obs operators).

2-stream I/O in MPAS-JEDI

Cold start (1st bkg generation)

```
<immutable_stream name="invariant"  
  type="input"  
  precision="single"  
  filename_template="x1.655362.invariant.nc"  
  io_type="pnetcdf,cdf5"  
  input_interval="initial_only" />
```

VS.

Cycling DA

```
<immutable_stream name="invariant"  
  type="input"  
  precision="single"  
  filename_template="invariant.655362.nc"  
  io_type="pnetcdf,cdf5"  
  input_interval="initial_only" />
```

```
<immutable_stream name="input"  
  type="input"  
  precision="single"  
  filename_template="x1.655362.init.nc"  
  io_type="pnetcdf,cdf5"  
  input_interval="initial_only" />
```

```
<immutable_stream name="input"  
  type="input"  
  precision="single"  
  filename_template="mpasin.$Y-$M-$D_$h.$m.$s.nc"  
  io_type="pnetcdf,cdf5"  
  input_interval="initial_only" />
```

```
<immutable_stream name="da_state"  
  type="output"  
  precision="single"  
  clobber_mode="truncate"  
  filename_template="mpasout.$Y-$M-$D_$h.$m.$s.nc"  
  packages="jedi_da"  
  io_type="pnetcdf,cdf5"  
  output_interval="06:00:00" />
```

```
<immutable_stream name="da_state"  
  type="output"  
  precision="single"  
  clobber_mode="truncate"  
  filename_template="mpasout.$Y-$M-$D_$h.$m.$s.nc"  
  packages="jedi_da"  
  io_type="pnetcdf,cdf5"  
  output_interval="6:00:00" />
```

2-stream I/O in MPAS-JEDI

Changes in the `namelist.atmosphere`:

Cold start forecast:

- invariant stream file should be set to the **invariant.nc** file, generated by MPAS *init_atmosphere* executable

```
&restart  
  config_do_restart = false  
  config_do_DAcycling = false
```



used for cycling DA experiments
that analyze uncoupled fields in
restart files

```
&assimilation  
  config_jedi_da = true
```



run within the JEDI data
assimilation framework

[MPAS-Atmosphere Model User's Guide Version 8.4.0](#)

2-stream I/O in MPAS-JEDI

Changes in the `namelist.atmosphere`:

Forecast in cycling DA:

- input stream should point the file generated from da_state stream in `streams.atmosphere`

```
&restart  
  config_do_restart = false  
  config_do_DAcycling = true
```



used for cycling DA experiments
that analyze uncoupled fields in
restart files

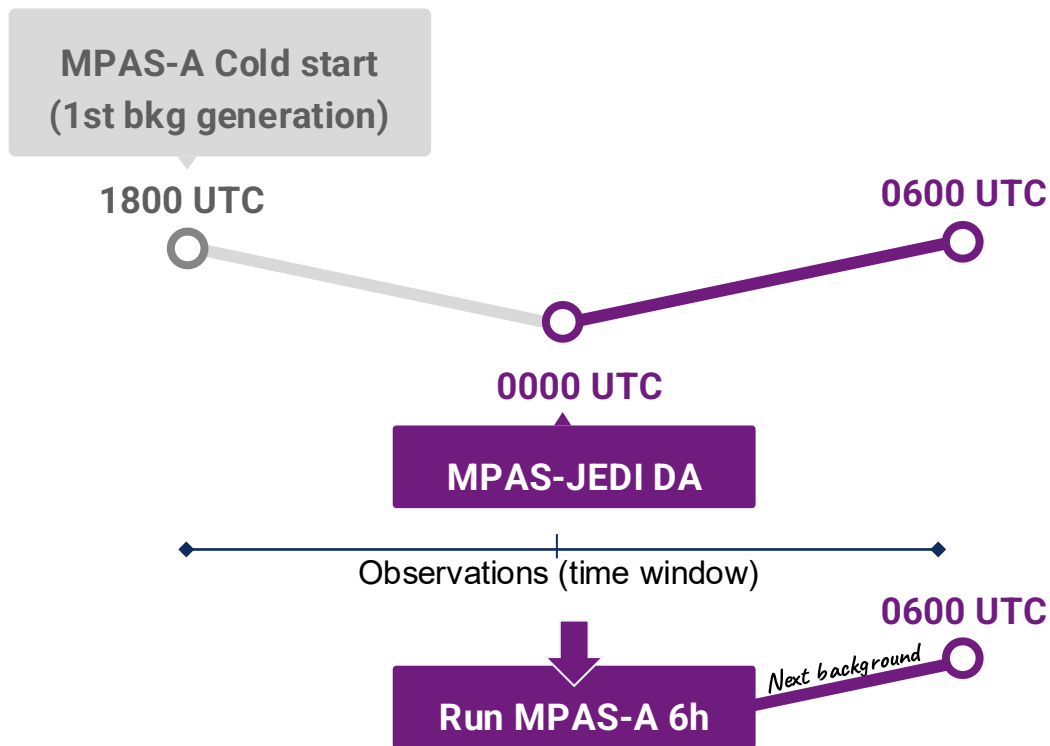
```
&assimilation  
  config_jedi_da = true
```



run within the JEDI data
assimilation framework

[MPAS-Atmosphere Model User's Guide Version 8.4.0](#)

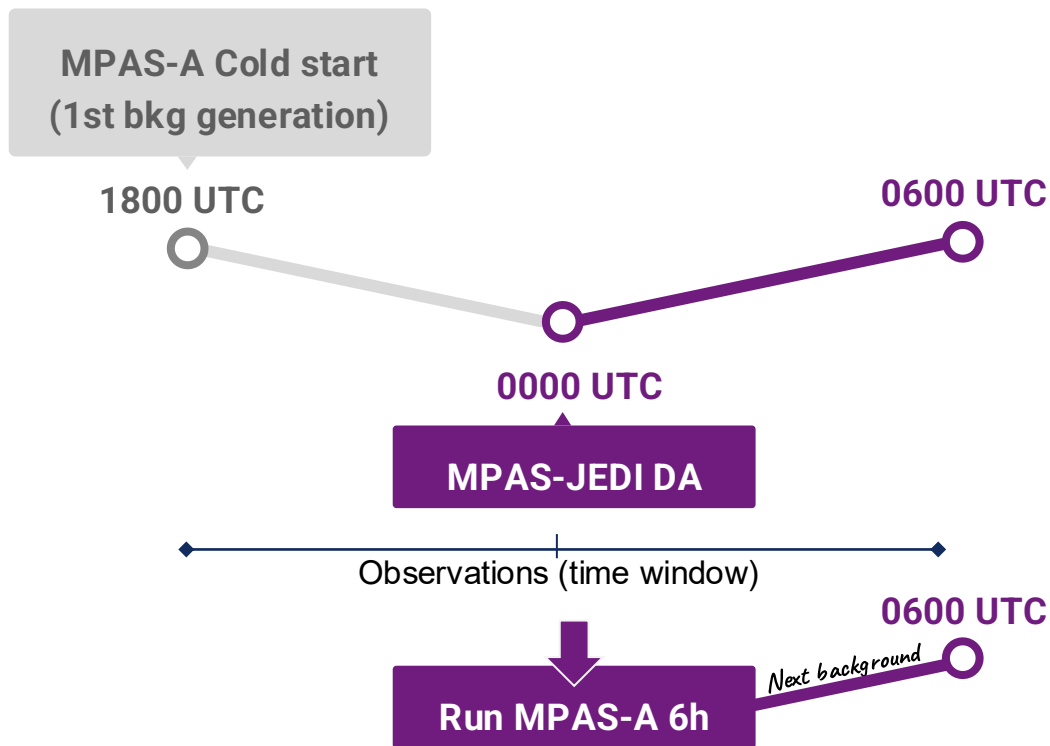
Cycling DA with MPAS-JEDI



Model-specific:

- Invariant file
- MPAS-A lookup tables
- Mesh partition file
- stream_list. files
- streams.atmosphere
- namelist.atmosphere

Cycling DA with MPAS-JEDI



MPAS-JEDI-specific:

- YAML configuration file
- geovars.yaml
- keptvars.yaml
- obsop_name_map.yaml
- templateFields. file (link to background file)
- Copy of mpasout into **mpasin (analysis or IC for MPAS-A) → updated after DA**
- Observations
- mpasjedi executable

Cycling DA with MPAS-JEDI

geovars.yaml



Model variables on the MPAS mesh to
**Geophysical Variables at
Locations (GeoVals)** for UFO

keptvars.yaml



MPAS-A variables that we want to keep in
memory

obsop_name_map.yaml



Mapping between observation variable
names and MPAS-JEDI specific variable
names

templateFields. file



Copy of background files with modified
"XTIME" variable (converted date strings
into NetCDF-compatible character arrays)
to get geometry information from MPAS-A

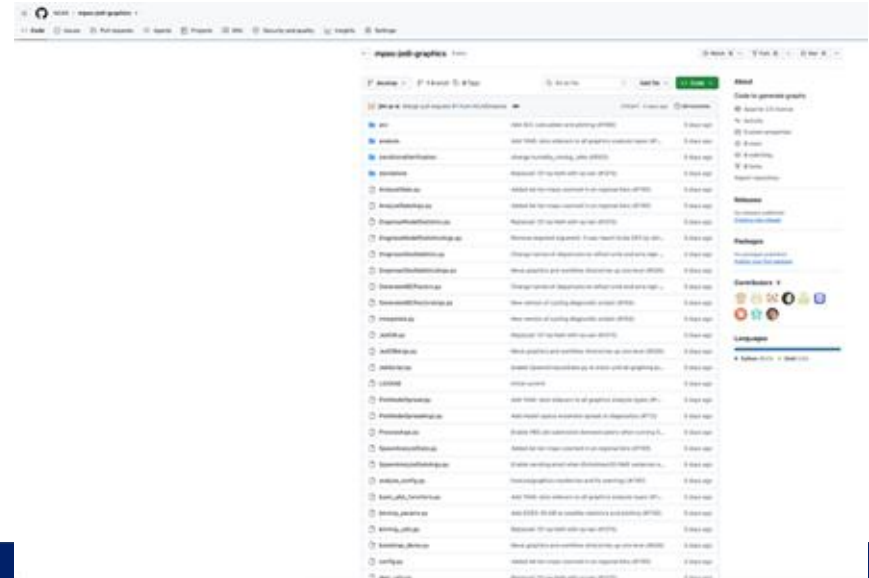
Graphics package

- ❑ Developed at NSF NCAR/MMM to aid in diagnosing results with MPAS and MPAS-JEDI
 - ❑ Observation space verification can be used for any JEDI model interface
- ❑ Python scripts
- ❑ Currently, only operates on NSF NCAR's Derecho HPC



- Now at: <https://github.com/NCAR/mpas-jedi-graphics>

but **NOT** supported



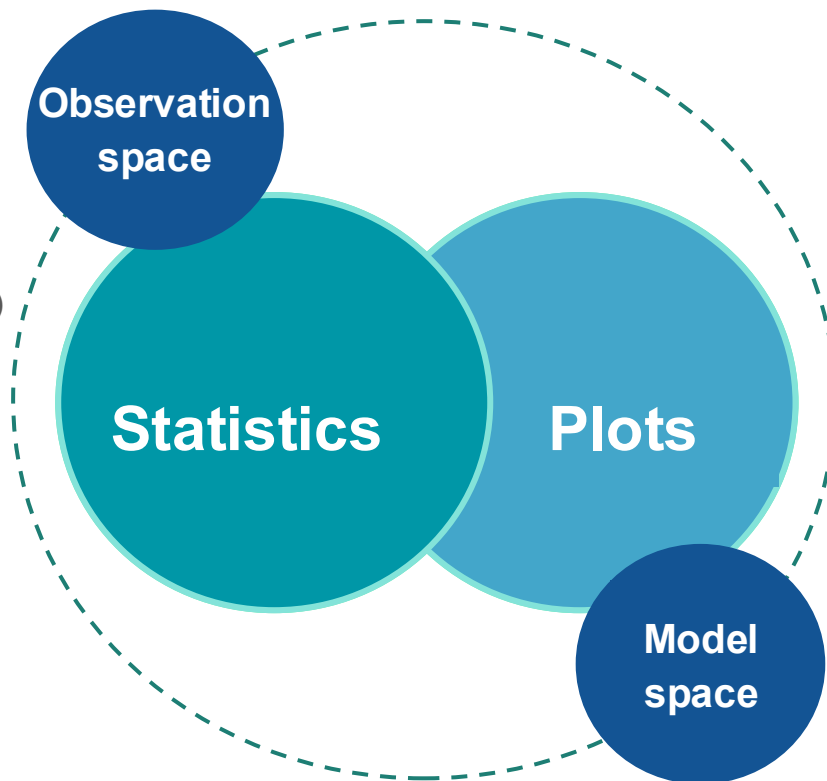
Graphics package: functionalities

- ☐ Produces statistics for selected diagnostics using the `DiagSpaces` selection.
- ☐ Distributed generation of information results in a database of processed statistics, stored in HDF5 files
- ☐ Distributed diagnostic files across multiple experiments, multiple cycle initial times, and multiple forecast lengths
- ☐ Enables portable reading of user-selected variables from multiples types of UFO feedback files (ObsSpace, GeoVaLs, ObsDiagnostics)
- ☐ Supports PBS script to submit verification jobs on Casper and Cheyenne
- ☐ IODA observation convention updates
- ☐ Updated QC flag numbers based on recent changes in UFO
- ☒ Users can select specific observation types, channels and variables to plot

Graphics package: functionalities

DiagSpaces:

Sondes, aircraft, AMV winds,
GNSSRO, surface pressure
AMSU-A (NOAA-15, NOAA-18,
NOAA-19, METOP-A, METOP-B)
MHS (NOAA-18, NOAA-19,
METOP-A, METOP-B)
IASI (METOP-A, METOP-B,
METOP-C)
ABI (GOES-16)
AHI (Himawari-8)



Analyzed variables:

2m T
2m Q
10m U and V
Ps
T
Theta
rho
W
Ps
U and V
Qv
Qv 1 to 10 model level
Qv 11 to 20 model level
Qv 21 to 30 model level
Qv 31 to 40 model level
QV 41 to 55 model level

Graphics package: functionalities

☐ Binning methods:

- ☐ global
- ☐ by latitude bands: Tro (-30.0, 30.0),
NXTro (30.0, 90.0), SXTro (-90.0, -30.0), NMid
(30.0, 60.0), SMid (-60.0, -30.0),
NPol (60.0, 90.0), SPol (-90.0, -60.0)
- ☐ by tropical latitude bands: ITZC (-5.0, 5.0),
STro (-30.0, -5.0), NTro (5.0, 30.0))
- ☐ by cloudiness: clear, mixed-pixels, cloudy, all-
sky
- ☐ Latitude vs Pressure 2D
- ☐ Longitude vs Latitude 2D
- ☐ Brightness temperature as a function of cloud
fraction 2D

☐ Types of plots:

- ☐ Time series plots with or without
confidence intervals calculated using
bootstrap resampling
- ☐ profile plots of binned data (e.g., over
pressure or latitude on the y-axis) with
and without confidence intervals
- ☐ maps of 2D-binned statistics
- ☐ score-card
- ☐ standalone: OmA/OmB diagnostics,
observations locations, analysis
increments, cost function

Count, Mean, STD, RMS, RMS relative difference

Graphics package: functionalities

How to run it?

Observation space:

OmA/OmB

```
python DiagnoseObsStatistics.py -n 36 -p ./dbOut -o obsout -g geoval -d ydiags -app variational -  
nout 2
```

Forecast vs observations (HofX)

```
python DiagnoseObsStatistics.py -n 36 -p ./dbOut -o obsout -g geoval -d ydiags -app hofx
```

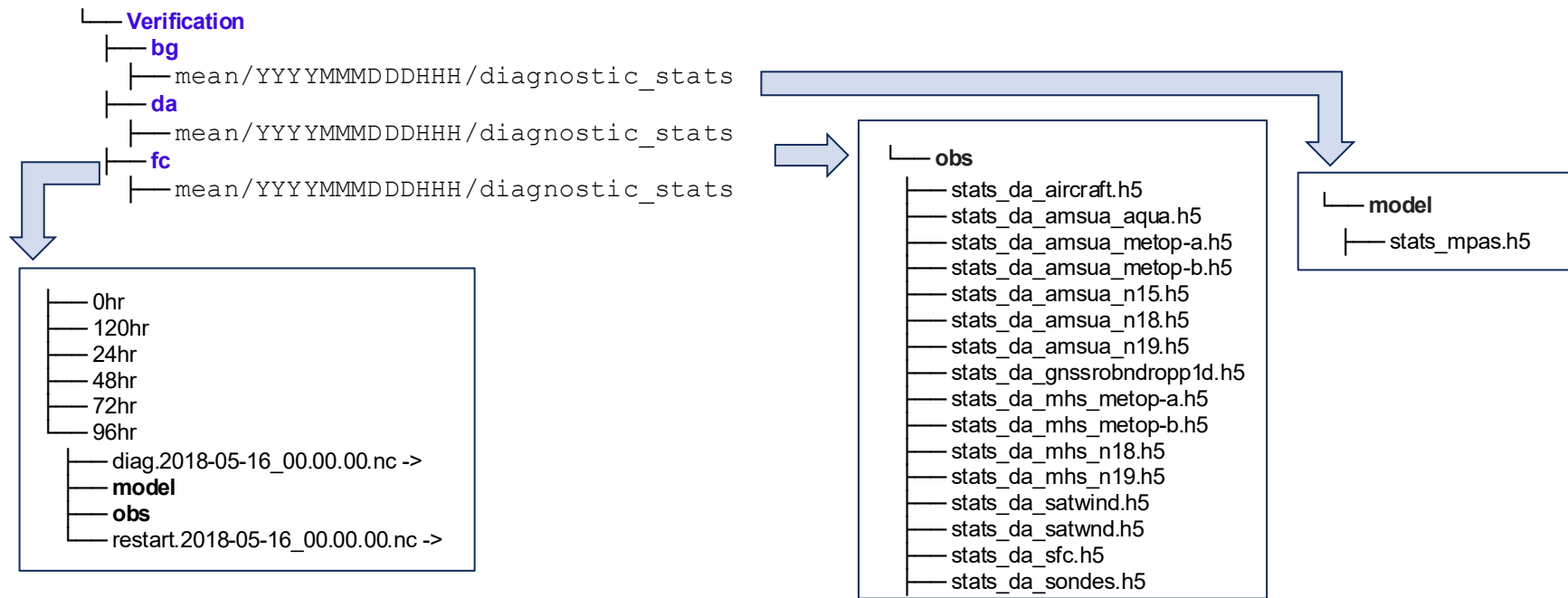
Model space (vs GFS analysis):

Forecast vs model

```
python DiagnoseModelStatistics.py YYYYMMMDDDDHHH -n 36 -r ./x1.655362.init  
30km
```

Graphics package: examples

Experiment folders structure: ivette_3dvar_OIE120km_WarmStart



Graphics package: functionalities

How to run it?

[analyze_config.py](#): top-level script that controls cycle times and forecast length, verification configuration, experiments and statistics to analyze, and analysis types to apply to the statistics

Observation space:

Carry out analyses for all DiagSpaces that contain "amsua"

```
python AnalyzeStats.py -d amsua
```

Job-submission examples:

```
./SpawnAnalyzeStats.py -nout 2 -d amsua_,sonde,airc,sfc,gnsro,satw
```

```
./SpawnAnalyzeStats.py -app hofx -d mhs,amsua,abi_,ahi_,sonde,airc,sfc,gnsro,satw
```

Model space (vs GFS analysis):

```
./SpawnAnalyzeStats.py -d mpas
```

Graphics package: functionalities

How to set it up?

[analyze_config.py](#): Most common parameters to set up for 6hr verification

General settings

```
dbConf['firstCycleDTime'] = dt.datetime(2018,4,15,0,0,0)
dbConf['lastCycleDTime'] = dt.datetime(2018,5,14,18,0,0)

# time increment (TimeInc) between valid Cycle (cy) date-times
dbConf['cyTimeInc'] = dt.timedelta(hours=6)
```

Experiment names (cntrlExpName has to match!!)

```
## cntrlExpName is the experiments key of the control experiment, which is used for DiffCI analyses
dbConf['cntrlExpName'] = 'clrama'

## experiments - dictionary with key, value pairs as follows
# + the key is a short name for the experiment (see expNames below)
# + the value is the directory where the verification statistics files are located
# + if using MPAS-Workflow, users only need to add one new 'experiments' entry per experiment and
#   select their desired VerificationType and VerificationSpace above

experiments = OrderedDict()

experiments['clrama'] = \
    'guerrett_3dhybrid-60-60-iter_gnssrorefncep_030kmI60km_ensB-SE80+RTTP70_VarBC_RefNCEP_2ndDoaDob' + \
    deterministicVerifyDir
```

Verification type and Verification space

```
## VerificationType
# OPTIONS: 'omb/oma', 'forecast'
# 'omb/oma' - calculated from a da application, only available when
#             VerificationSpace=='obs'
# 'forecast' - single- or multi-duration forecasts either in observation or model space
VerificationType = 'forecast'

## VerificationSpace
# OPTIONS: 'obs', 'model'
# 'obs' - observation space
# 'model' - compare to analyses in model space, only available when VerificationType=='forecast'
VerificationSpace = 'obs'
```

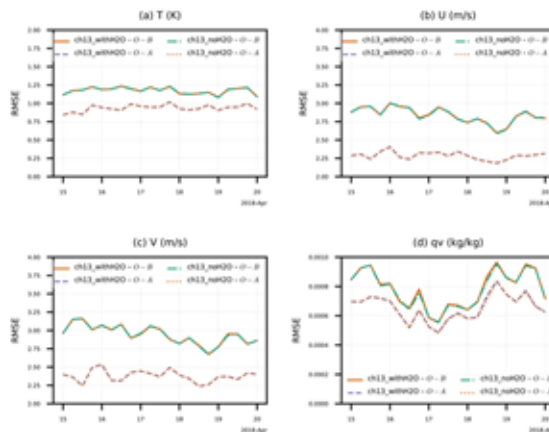
Graphics package: examples

Observation space

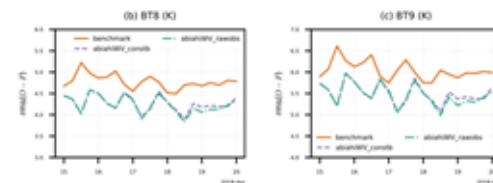
DiagSpace_analyses

- BinValAxes2D
- BinValAxisProfileDiffCI
- CYandBinValAxes2D
- CYAxisExpLines

aircraft: OmA/OmB



ABI: OmB (HofX)

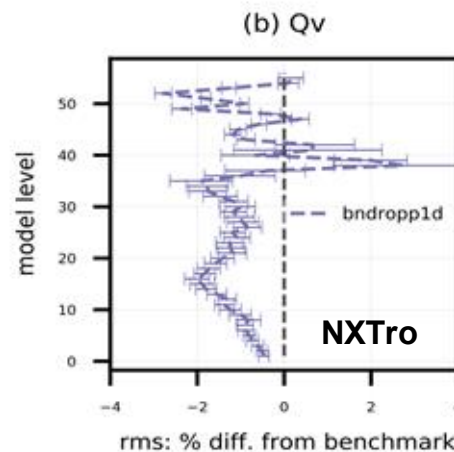
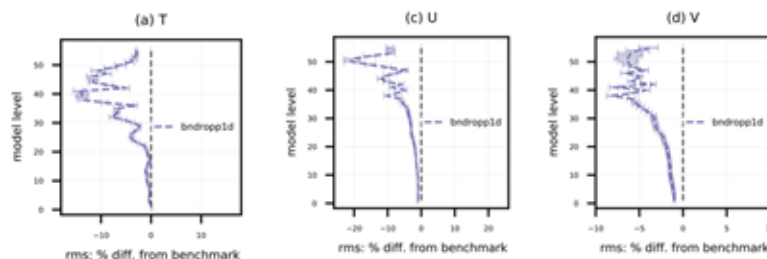


Model space

mpas_analyses

- BinValAxes2D
- BinValAxisProfileDiffCI
- CYandBinValAxes2D
- CYAxisExpLines

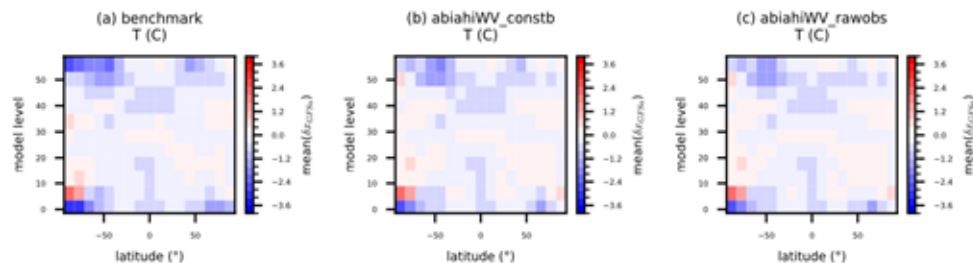
MPAS 6-h
verification vs
GFS analysis



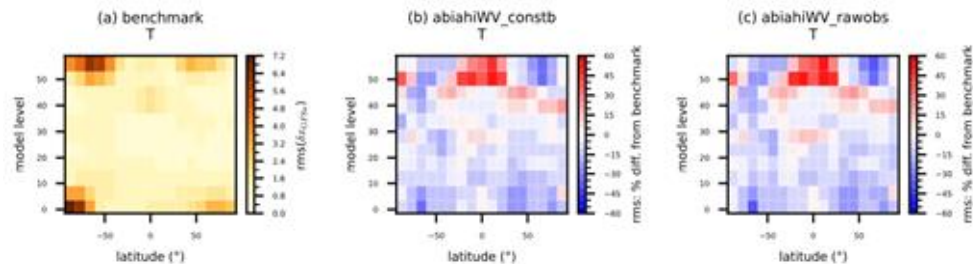
Graphics package: examples

MPAS 6-h verification vs GFS analysis

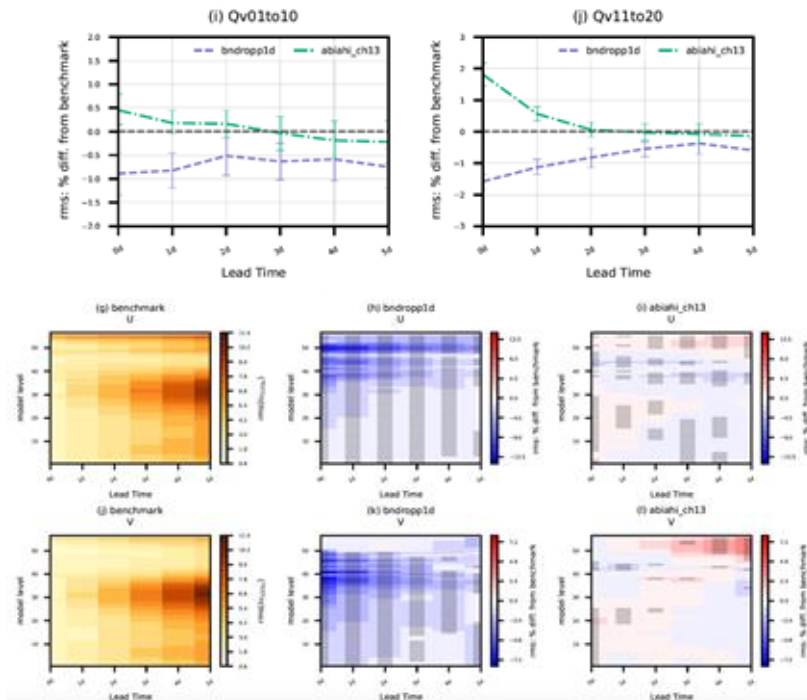
BIAS



RMSE



MPAS 5-days verification vs GFS analysis



**Contributions for new diagnostics/capabilities
are welcome!!!**

<https://github.com/NCAR/mpas-jedi-graphics>

References

Duda, M., Fowler, L., Skamarock, B., Roesch, C., Jacobsen, D., Ringler, T. & Gopal, A. (2026). *MPAS-Atmosphere Model User's Guide Version 8.4.0* (Last updated: August 14, 2026). UCAR. https://www2.mmm.ucar.edu/projects/mpas/mpas_atmosphere_users_guide_8.4.0.pdf

Ha, S., Guerrette, J. J., Hernández Baños, I., Skamarock, W. C., and Duda, M. G.: Incremental analysis update (IAU) in the Model for Prediction Across Scales coupled with the Joint Effort for Data assimilation Integration (MPAS–JEDI 2.0.0), *Geosci. Model Dev.*, 17, 4199–4211, <https://doi.org/10.5194/gmd-17-4199-2024>, 2024.