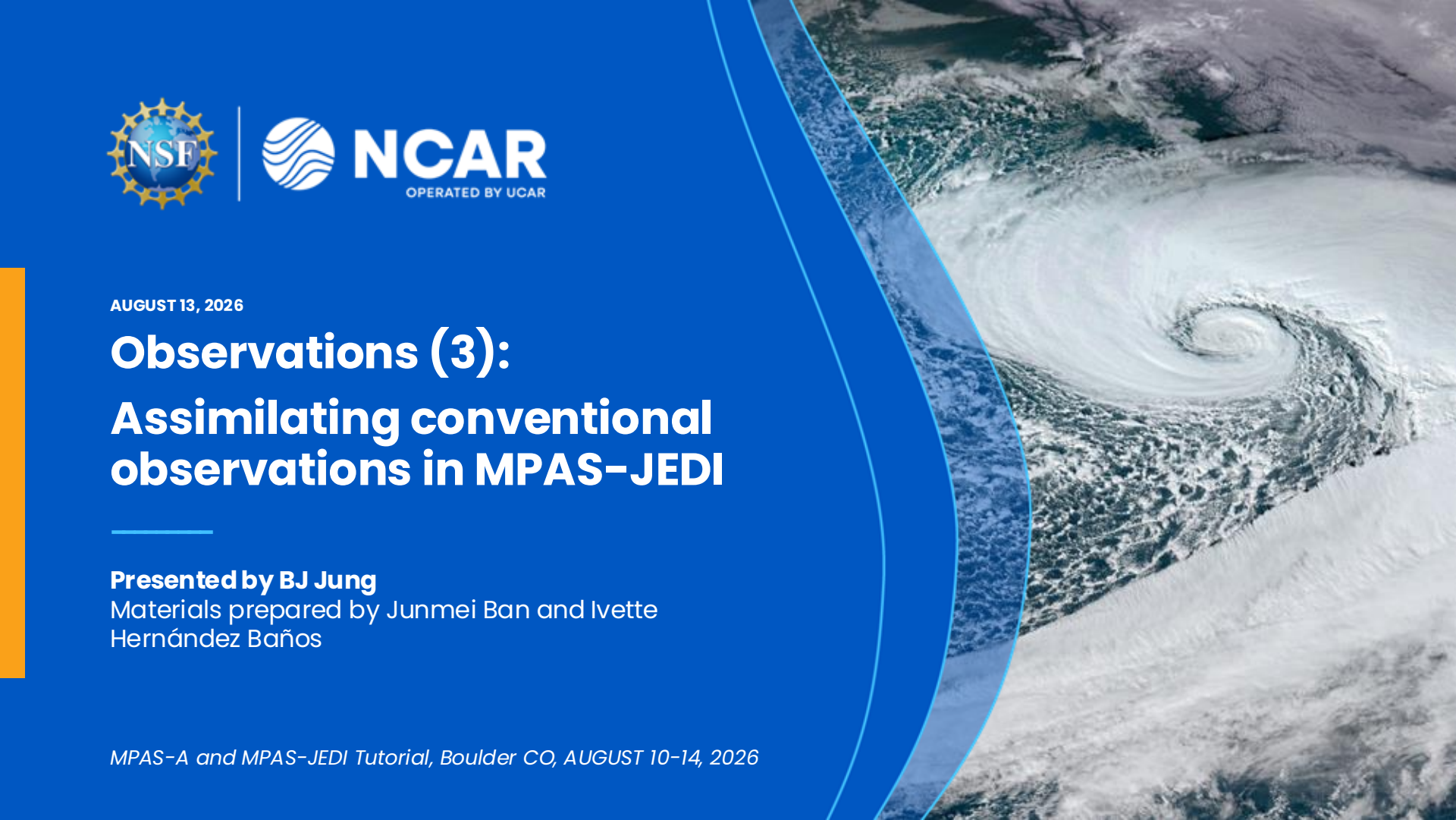


AUGUST 13, 2026

Observations (3): Assimilating conventional observations in MPAS-JEDI

Presented by BJ Jung
Materials prepared by Junmei Ban and Ivette Hernández Baños

MPAS-A and MPAS-JEDI Tutorial, Boulder CO, AUGUST 10-14, 2026

Outline

1. Assimilating non-radiance (conventional) observations
    a. Set up a variational run (hybrid 3DEnVar)
        i. Set up a yaml file
    b. Observation operators available in UFO
    c. Quality Control available in UFO
    d. Variational Application (3DEnVar)
2. Diagnostics

Assimilating non-radiance observations

Set up a variational run (hybrid 3DEnVar)

3DVar cost function

$$J(\boldsymbol{x}) = \frac{1}{2}(\boldsymbol{x}-\boldsymbol{x}_b)^{\mathrm{T}}\mathbf{B}^{-1}(\boldsymbol{x}-\boldsymbol{x}_b) + \frac{1}{2}[H(\boldsymbol{x})-\boldsymbol{y}]^{\mathrm{T}}\mathbf{R}^{-1}[H(\boldsymbol{x})-\boldsymbol{y}]$$

Hybrid **B**

$$\mathbf{B} = \beta_{\mathrm{s}}\mathbf{B}_{\mathrm{s}} + \beta_{\mathrm{e}}\mathbf{L} \circ \mathbf{B}_{\mathrm{e}}$$

B$_s$: static B; **B**$_e$: ensemble B; **L**: localization matrix; β_s and β_s: static and ensemble weights

Inputs:

$\boldsymbol{x}_b$: background fields ----from forecast

$\boldsymbol{y}$: observations ----in **IODA** format; from **ioda-converter**

$\mathbf{R}$: observation error covariance matrix ----from observations file (ObsError group), or defined in YAML as a filter

$\mathbf{B}$: background error covariance matrix ----for pure 3DEnVar, determined from ensemble forecasts; needs localization input (**L**)

Output:

$\boldsymbol{x}$: *analysis*

Assimilating non-radiance observations

Set up a yaml file

(focus on observations)

```
observations:
  observers:
  - obs space:
      name: Aircraft
    obs error:
    obs operator:
    obs filters:
```

```
- obs space:
    name: GnssroRefNCEP
  obs error:
  obs operator:
  obs filters:
```

```
- obs space:
    name: Satwind
  obs error:
  obs operator:
  obs filters:
...
```

Assimilating non-radiance observations

Set up a yaml file

(focus on observations)

observations:
  observers:
  - obs space:
      name: Aircraft
    obs error:
    obs operator:
    obs filters:

```
79  observations:
80    observers:
81    - obs space:
82        name: Aircraft
83        obsdatain:
84          engine:
85            type: H5File
86            obsfile: ./aircraft_obs_2018041500.h5
87        obsdataout:
88          engine:
89            type: H5File
90            obsfile: ./obsout_da_aircraft.h5
91        simulated variables: [airTemperature, windEastward, windNorthward,
92                              specificHumidity]
93      obs error:
94        covariance model: diagonal
95      obs operator:
96        name: VertInterp
97      obs filters:
98      - filter: PreQC
99        maxvalue: 3
100     - filter: Background Check
101       threshold: 3.0
```

PreQC: Quality markers are assigned by various data pre-processing software

PreQC is assigned from **ioda-converter** in subroutine filter_obs_conv (as in GSI's read_prepbufr.f90)

Assimilating non-radiance observations

Set up a yaml file

(focus on observations)

observations:
  observers:
  - obs space:
      name: Satwind
    obs error:
    obs operator:
    obs filters:

```
152       obs error:
153         covariance model: diagonal
154       obs operator:
155         name: VertInterp
156         observation alias file: obsop_name_map.yaml
157       obs filters:
158       - filter: PreQC
159         maxvalue: 3
160       # Assign the initial observation error, based on height/pressure
161       - filter: Perform Action
162         filter variables:
163         - name: windEastward
164         - name: windNorthward
165         action:
166           name: assign error
167           error function:
168             name: ObsFunction/ObsErrorModelStepwiseLinear
169             options:
170               xvar:
171                 name: MetaData/pressure
172               xvals: [100000, 95000, 80000, 65000, 60000, 55000, 50000, 45000, 40000, 35000, 30000, 25000, 20000, 15000, 10000]
173               errors: [1.4, 1.5, 1.6, 1.8, 1.9, 2.0, 2.1, 2.3, 2.6, 2.8, 3.0, 3.2, 2.7, 2.4, 2.1]
174       - filter: Bounds Check
175         filter variables:
176         - name: windEastward
177         - name: windNorthward
178         test variables:
179         - name: ObsErrorData/windEastward
180         - name: ObsErrorData/windNorthward
181         minvalue: 0.0
182         maxvalue: 200.0
183       - filter: Gaussian Thinning
184         horizontal_mesh: 145.0
185       - filter: Background Check
186         threshold: 3.0
```

Filter: Perform Action; **action**: assign error ⇒ ufo/src/ufo/filters/actions/AssignError.cc

Error estimates of observations flagged by the filter are set to a specified value. This can be either a constant (specified using the error parameter option) or a variable (specified using the error function option)

Assimilating non-radiance observations

Set up a yaml file

(focus on observations)

observations:
  observers:
  - obs space:
      name: GnssroRefNCEP
    obs error:
    obs operator:
    obs filters:

```
115       obs operator:
116         name: GnssroRefNCEP
117         obs options:
118           use_compress: 0
119       obs filters:
120       - filter: Domain Check
121         where:
122         - variable:
123             name: MetaData/height
124           minvalue: 0.0
125           maxvalue: 30000.0
126         - variable:
127             name: MetaData/earthRadiusCurvature
128           minvalue: 6250000.0
129           maxvalue: 6450000.0
130         - variable:
131             name: MetaData/geoidUndulation
132           minvalue: -200.0
133           maxvalue: 200.0
134       - filter: RObserror
135         variable: atmosphericRefractivity
136         errmodel: NCEP
137         apply at iterations: 0,1,2
138       - filter: Background Check
139         threshold: 3.0
140         apply at iterations: 0,1,2
```

Domain Check:
ufo/src/ufo/filters/ObsDomainCheck.cc

Retains all observations selected by the **where** statement and rejects all others; here, the filter is used to control the maximum height one wants to assimilate RO observation.

RObserror (errmodel: NCEP): RO specific filter

Assimilating non-radiance observations

Set up a yaml file

(focus on observations)

observations:
  observers:
  - obs space:
      name: SfcCorrected
    obs error:
    obs operator:
    obs filters:

```
201       obs operator:
202         name: SfcCorrected
203         correction scheme to use: UKMO    # or WRFDA
204       linear obs operator:
205         name: Identity
206         observation alias file: obsop_name_map.yaml
207       obs filters:
208       - filter: PreQC
209         maxvalue: 3
210       - filter: Difference Check
211         reference: MetaData/stationElevation
212         value: GeoVaLs/height_above_mean_sea_level_at_surface
213         threshold: 200.0
214       - filter: Background Check
215         threshold: 3.0
216         apply at iterations: 0,1
```

SfcCorrected operator: corrects the computation of surface atmospheric **P** at a location for the discrepancy in model topography at the observation location.

Difference Check:
ufo/src/ufo/filters/DifferenceCheck.cc
Compares the difference between a reference variable and a second variable and assign a QC flag if the difference is outside of a prescribed range.

Assimilating non-radiance observations

Observation operators available in UFO

- Categorical
- Composite
- Density Reduction
- Vertical Interpolation
- Column Retrieval Operator
- Atmosphere Vertical Layer Interpolation (deprecated)
- Atmosphere Surface Interpolation (deprecated)

Radiance
- Community Radiative Transfer Model (CRTM)
- RTTOV

- Aerosol Optical Depth (AODCRTM)
- Aerosol Optical Depth (AODMassFraction)
- Aerosol Optical Depth (AOD) for dust (Met Office)

GNSS RO
- GNSS RO bending angle (NBAM)
- GNSS RO bending angle (ROPP 1D)
- GNSS RO bending angle (ROPP 2D)
- GNSS RO bending angle (MetOffice)
- GNSS RO refractivity (NCEP)
- GNSS RO refractivity (Met Office)

- Ground Based GNSS observation operator (Met Office)
- Identity observation operator
- Product observation operator
- Logarithm observation operator
- Generic particulate matter (PM) operator
- CMAQ particulate matter (PM) operator (**deprecated**)
- Radar Radial Velocity
- Radar Doppler wind
- Radar Reflectivity
- Scatterometer neutral wind (Met Office)
- SfcCorrected
- Background Error Vertical Interpolation
- Background Error Identity
- Total column water vapour
- Absolute dynamic topography
- Cool skin
- Insitu temperature
- Vertical Interpolation
- Sea ice thickness
- Sea ice fraction
- Profile Average operator
- Vertical/Slant-path summation/integration Operator

https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/inside/jedi-components/ufo/index.html

Assimilating non-radiance observations

Quality Control available in UFO

Generic filters

- Bounds Check Filter
- Background Check Filter
- Bayesian Background Check Filter
- Bayesian Background QC Flags filter
- Bayesian Whole Report Filter
- PreQC Filter
- Polygon Check Filter
- Domain Check Filter
- BlackList Filter
- RejectList Filter
- AcceptList Filter
- Perform Action Filter
- Thinning Filter
- Gaussian Thinning Filter
- Temporal Thinning Filter
- Poisson Disk Thinning Filter
- Duplicate Thinning Filter
- Stuck Check Filter
- Difference Check Filter
- Derivative Check Filter
- Spike and Step Check Filter
- Track Check Filter
- Ship Track Check Filter
- Met Office Buddy Check Filter
- History Check Filter
- Variable Assignment Filter
- Create Diagnostic Flags Filter
- RTTOV 1D-Var Check (RTTOVOneDVar) Filter
- ModelOb Threshold Filter
- Satwind Inversion Filter
- GNSS-RO 1D-Var Check (GNSSROOneDVar) Filter
- Model Best Fit Pressure Filter
- Process AMV QI
- Satname Filter
- Met Office Duplicate Check Filter
- SuperOb filter
- Parameter Substitution filter
- Ensemble Statistics
- Find Nearest Neighbors Filter
- Percentile Filter

Background

- Observation Filters
- QCmanager
- Order of Filter Application
- Derived Variables
- Observation Errors

Additional QC Filter Options

- Where Statement
- ObsFunction and ObsDiagnostic Suffixes
- Filter Actions
- Outer Loop Iterations
- Identifier

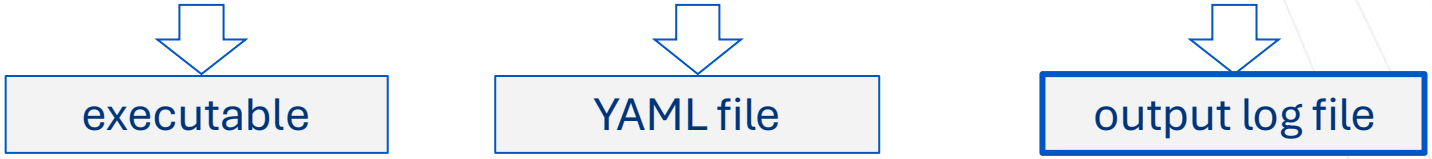
Profile Specific QC Filters

- Profile Background Check
- Profile Few Observations Check
- Profile Unflag Observations Check
- Impact Height Check
- Conventional Profile Processing
- Ocean Vertical Stability Check
- Average Observations to Model Levels
- Average Observations To GeoVals Model Levels

https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/inside/jedi-components/ufo/qcfilters/index.html

Variational Application

./mpasjedi_variational.x ./3denvar.yaml ./mpasjedi_3denvar.log



mpasjedi_3denvar.log:

```
OOPS_STATS Run end                      - Runtime:    126.83 sec,  Memory: total:    30.45 GB, per task: min
=   410.86 MB, max =   976.70 MB

Run: Finishing oops::Variational<MPAS, UFO and IODA observations> with status = 0
OOPS Ending   2026-08-06 11:32:52 (UTC-0600)
```

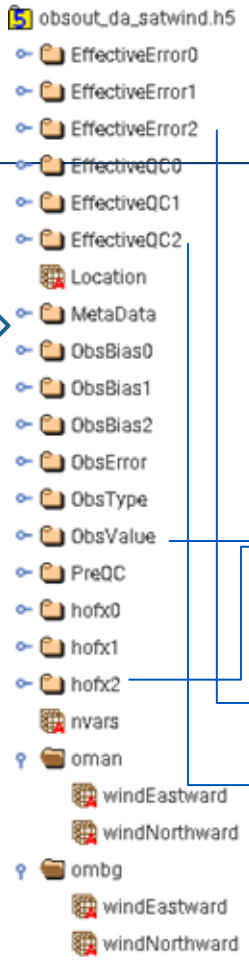
Output feedback files (per assimilated observation type):

obsout_da_aircraft.h5 obsout_da_gnssrorefncep.h5
obsout_da_satwind.h5 obsout_da_sondes.h5
obsout_da_sfc.h5 ...

Diagnostics

hdfview

satwind feedback file



Selection of common group names and meanings

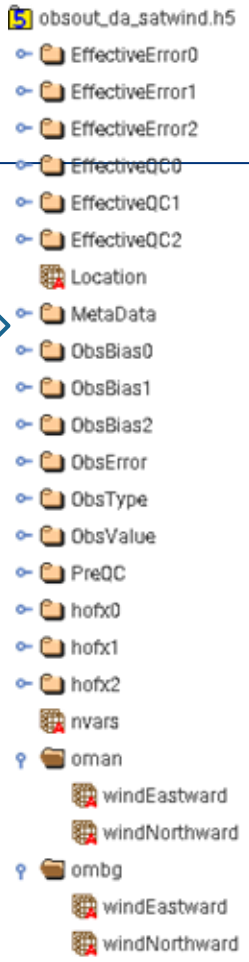
Group Name	Meaning
ObsValue	For when a specific variable is a direct observed/reported measurement, such as satellite radiance or surface weather observations of airTemperature and dewpointTemperature.
Metadata	Use this group name for ancillary data that provides added description to an ObsValue in general. Simple examples are stationElevation and airTemperature to provide the added information needed for the altitude for which a surface temperature observation was made. Similarly, the airPressure, altitude, and eastwardWind for radiosonde or satellite atmospheric motion vector winds.
HofX	This is the end product of the forward operator, known in DA as H(x) or HofX.
ObsError	This group name denotes Observation Errors that arrive from upstream data sources. The values are usually considered to be the standard deviation of observation errors.
EffectiveError	This group name is UFO's computed effective ObsError value after any number of QC steps that may "inflate" or alter the ObsError. In JEDI, this final value given to the DA means that ObsValues with large relative EffectiveError have less impact than relatively small EffectiveError values.
EffectiveQC	This group name is UFO's final QC value given by the QCflags.h enumeration of values associated with various QC rejection or other steps. Examples include Bounds Check, Domain Check, Background Check, etc.

https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/inside/conventions/objects_and_layouts.html#group-based-data-organization

Diagnostics

hdfview

satwind feedback file



Quality Control Flags

```
  constexpr int pass    = 0;  // we like that one!
  constexpr int passive = 1;  // H(x) is computed (for monitoring, BC...) but obs not assimilated
// Single digit values reserved for DA use.
// For now only 0, 1 and >1 are used but keeping space for other potential use cases.

// Actual rejection flags
  constexpr int missing = 10;  // missing values prevent use of observation
  constexpr int preQC   = 11;  // observation rejected by pre-processing
  constexpr int bounds  = 12;  // observation value out of bounds
  constexpr int domain  = 13;  // observation not within domain of use
  constexpr int black   = 14;  // observation black listed
  constexpr int Hfailed = 15;  // H(x) computation failed
  constexpr int thinned = 16;  // observation removed due to thinning
  constexpr int diffref = 17;  // metadata too far from reference
  constexpr int clw     = 18;  // observation removed due to cloud field
  constexpr int fguess  = 19;  // observation too far from guess
  constexpr int seaice  = 20;  // observation based sea ice detection, also flags land points
  constexpr int track   = 21;  // observation removed as inconsistent with the rest of track
  constexpr int buddy   = 22;  // observation rejected by the buddy check
  constexpr int derivative = 23;  // observation removed due to metadata derivative value
  constexpr int profile = 24;  // observation rejected by at least one profile QC check
  constexpr int onedvar = 25;  // observation failed to converge in 1dvar check
  constexpr int bayesianQC = 26;  // observation failed due to Bayesian background check
  constexpr int modelobthresh = 27;  // observation failed modelob threshold check
  constexpr int history = 28;  // observation failed when compared with historical data
  constexpr int processed = 29;  // observation processed but deliberately H(x) not calculated
  constexpr int superrefraction = 30;  // observation rejected by GNSSRO super refraction QC
  constexpr int superob = 31;  // superob value not set at this location
  constexpr int percentile = 33;  // observation removed due to percentile filter
```

https://github.com/JCSDA/ufo/blob/develop/src/ufo/filters/QCflags.h

Diagnostics

Check output log file:

QC counts for surface pressure

```
QC SfcCorrected stationPressure: 66147 missing values.
QC SfcCorrected stationPressure: 3921 rejected by pre QC.
QC SfcCorrected stationPressure: 12662 rejected by first-guess check.
QC SfcCorrected stationPressure: 15503 rejected by difference check.
QC SfcCorrected stationPressure: 36351 passed out of 134584 observations.
```

QC counts for satwnd (U component)

```
QC Satwind windEastward: 84723 rejected by pre QC.
QC Satwind windEastward: 179267 removed by thinning.
QC Satwind windEastward: 1043 rejected by first-guess check.
QC Satwind windEastward: 13391 passed out of 278424 observations.
```

Cost function and norm reduction

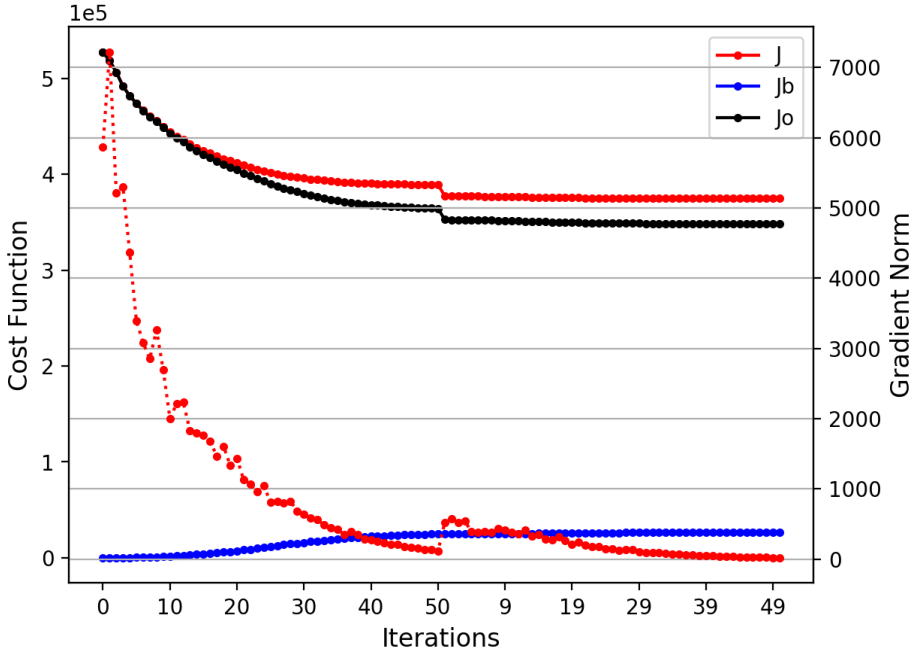
```
Quadratic cost function: J  ( 0) = 527463.0707663337
Quadratic cost function: Jb  ( 0) = 0
Quadratic cost function: JoJc( 0) = 527463.0707663337
Quadratic cost function: J  ( 1) = 518591.5723829174
Quadratic cost function: Jb  ( 1) = 4.577579695354762
Quadratic cost function: JoJc( 1) = 518586.994803222
Quadratic cost function: J  ( 2) = 506511.4946705618
Quadratic cost function: Jb  ( 2) = 31.13332883225612
Quadratic cost function: JoJc( 2) = 506480.3613417295
Quadratic cost function: J  ( 3) = 491953.1839359514
. . .

Norm reduction ( 0) = 1
Norm reduction ( 1) = 1.23090225813257
Norm reduction ( 2) = 0.888815108038685
Norm reduction ( 3) = 0.9037112461414154
Norm reduction ( 4) = 0.7446933413990573
Norm reduction ( 5) = 0.5788186291861125
. . .
```

Diagnostics

Under **graphics**:

- standalone
  - plot_cost_grad.py

Check cost function and norm reduction:



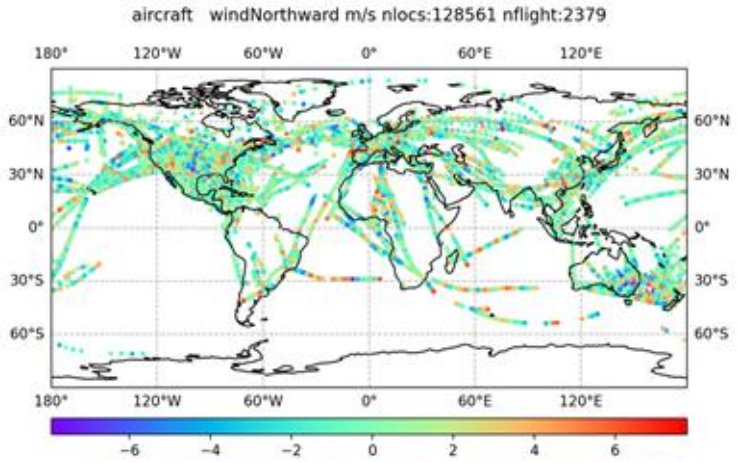
Diagnostics

Under **graphics**:

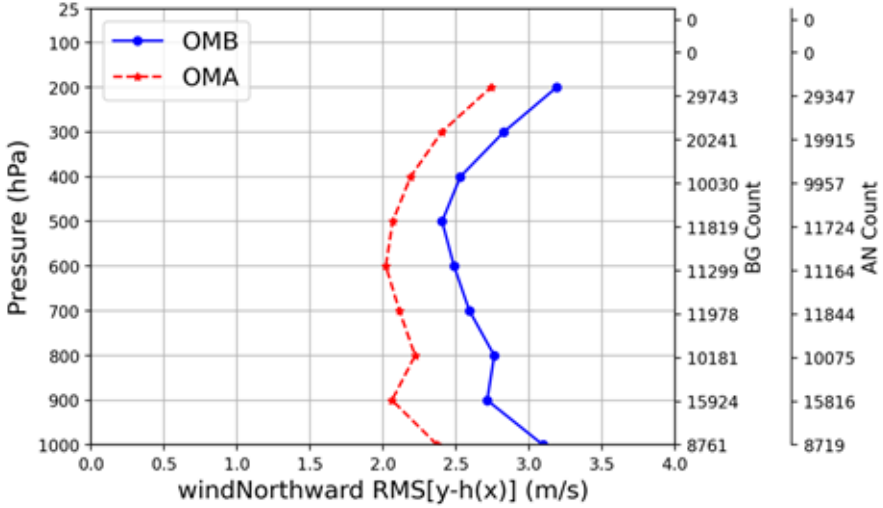
- standalone
  - plot_diag.py

Check observation departures figures:

OmA distribution



RMS of OmB/OmA profile



Diagnostics

For more scripts for diagnostics, check https://github.com/JCSDA/mpas-jedi/tree/develop/graphics/standalone