Basics of Data Assimilation

Jake Liu (liuz@ucar.edu)

Prediction, Assimilation, and Risk Communication Section
Mesoscale & Microscale Meteorology Laboratory
National Center for Atmospheric Research

MPAS-JEDI Tutorial, INPE, 15-16 August, 2024

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

Outline

- Scalar case
- Case with two state variables
- General n-dimensional case

What is data assimilation?

- A probabilistic method to obtain the best-possible estimate of state variables of a dynamic/physical system

- In the atmospheric sciences, DA typically involves combining a short-term model forecast (i.e., Background or Prior) and observations, along with their respective errors characterization, to produce an *analysis (Posterior)* that can initialize a numerical weather prediction model (e.g., WRF or MPAS)

Scalar Case

- State variable to estimate "x", e.g., consider this morning's 2-meter temperature at INPE, at 9 am local time, i. e., 12 UTC,

- Now we have a "background" (or "prior") information x_b of x, which is from a 6-h MONAN-v1.0 forecast initiated from 06 UTC GFS analysis.

- We also have an observation y of x at a surface station at INPE

- What is the best estimate (analysis) x_a of x?

Scalar Case

- We can simply average xb and y: $x_a = \frac{1}{2}(x_b + y)$
  - This actually means we trust equally the background and observation, giving them equal weight

- But if xb and y's accuracy are different and we have some knowledge about their errors
  - e.g., for background, we have statistics (e.g., mean and variance) of $x_b - y$ from the past
  - For observation, we have instrument error information from manufacturer

Scalar Case

- Then we can do a weighted mean: $x_a = ax_b + by$ in a least square sense, i.e.,

Minimize $$J(x) = \frac{1}{2}\frac{(x-x_b)^2}{\sigma_b^2} + \frac{1}{2}\frac{(x-y)^2}{\sigma_o^2}$$

Requires $$\frac{dJ(x)}{dx} = \frac{(x-x_b)}{\sigma_b^2} + \frac{(x-y)}{\sigma_o^2} = 0$$

Then we can easily get $$x_a = \frac{\sigma_o^2}{\sigma_b^2+\sigma_o^2}x_b + \frac{\sigma_b^2}{\sigma_b^2+\sigma_o^2}y = \frac{1}{1+\sigma_b^2/\sigma_o^2}x_b + \frac{1}{1+\sigma_o^2/\sigma_b^2}y$$

Or we can write in the form of analysis increment

Called "Innovation" or O minus B, or OMB

$$x_a - x_b = \frac{\sigma_b^2}{\sigma_b^2+\sigma_o^2}(y-x_b) = \frac{1}{1+\sigma_o^2/\sigma_b^2}(y-x_b)$$

NCAR | UCAR

Scalar Case

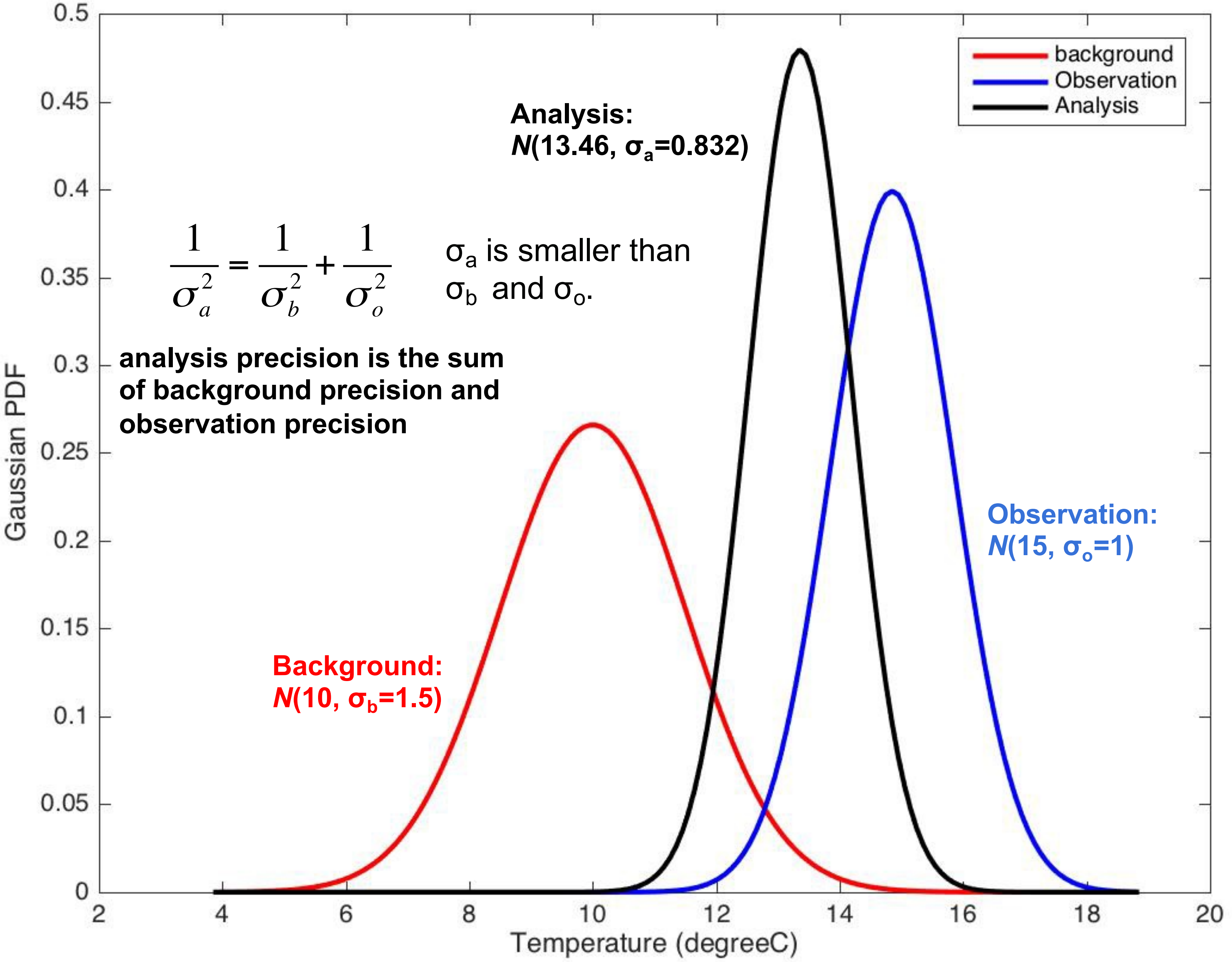
Minimize $$J(x) = \frac{1}{2}\frac{(x-x_b)^2}{\sigma_b^2} + \frac{1}{2}\frac{(x-y)^2}{\sigma_o^2}$$

is actually equivalent to maximize a Gaussian Probability Distribution Function (PDF)

$$ce^{-J(x)}$$

Assume errors of X_b and y are unbiased

A probabilistic view of scale case



Two state variables case

- Consider two state variables to estimate: INPE and USP's 2m temperatures x_1 and x_2 at 12 UTC today.

- Background from 6-h forecast: x_1^b and x_2^b and their error covariance with correlation *c*

$$\mathbf{B} = \begin{bmatrix} \sigma_1^2 & c\sigma_1\sigma_2 \\ c\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} = \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \begin{bmatrix} 1 & c \\ c & 1 \end{bmatrix} \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix}$$

- We only have an observation y_1 at the INPE station and its error variance σ_o^2

- Now we want to estimate T at 2 locations with obs at one location

Analysis increment for two variables

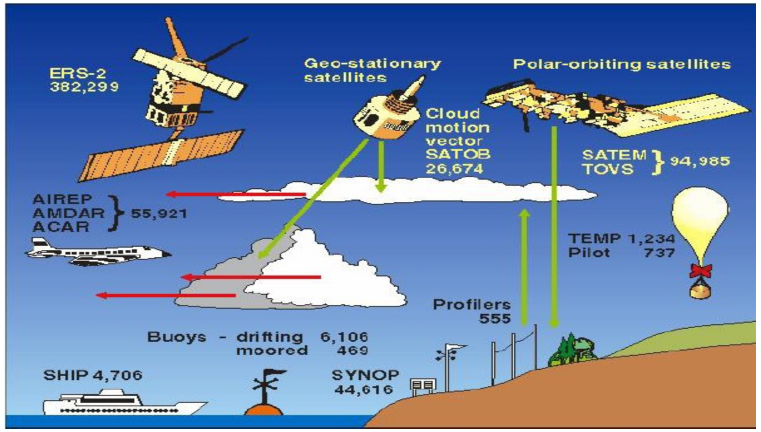
$$x_1^a - x_1^b = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_o^2}(y_1 - x_1^b) \longleftarrow \text{INPE}$$

$$x_2^a - x_2^b = \frac{c\sigma_1\sigma_2}{\sigma_1^2 + \sigma_o^2}(y_1 - x_1^b) \longleftarrow \text{USP}$$

Unobserved variable x_2 gets updated through the error correlation *c* in the background error covariance.

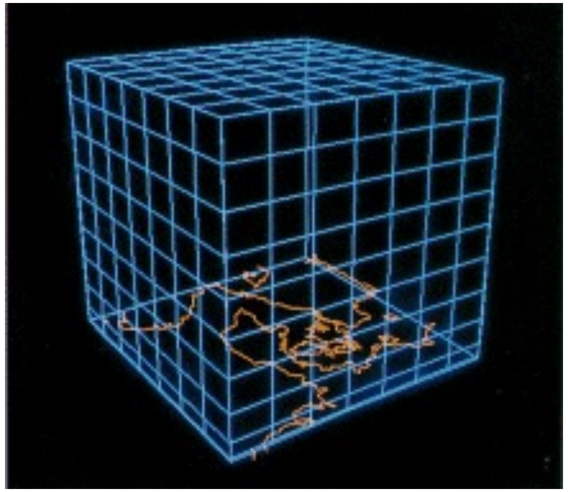
In general, this correlation can be correlation between two locations (spatial), two variables (multivariate), or two times (temporal).

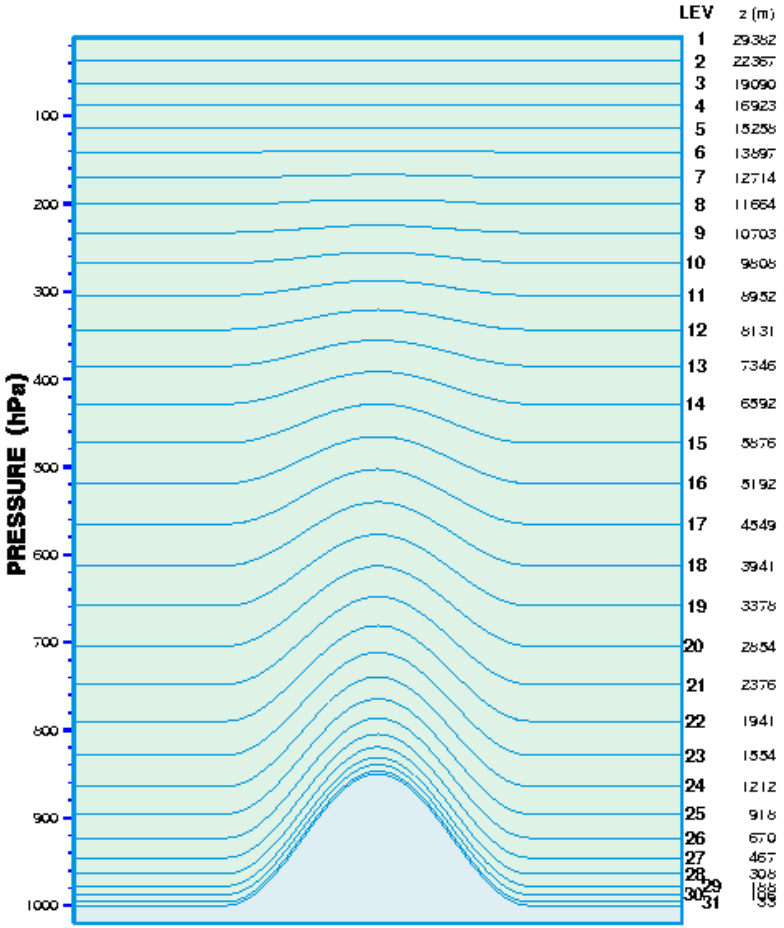
General Case



Observations
y^o, ~10^5-10^6

Model state
x, ~10^7





Vertical resolution of the DMI-HIRLAM system

General Case: vector and matrix notation

state vector

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

observation vector

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

background error covariance

$$\mathbf{B} = \begin{bmatrix} \sigma_1^2 & c_{12}\sigma_1\sigma_2 & \dots & \dots \\ c_{12}\sigma_1\sigma_2 & \sigma_2^2 & \dots & \dots \\ \dots & \dots & \ddots & \dots \\ \dots & \dots & \dots & \sigma_m^2 \end{bmatrix} = \sigma\ \mathrm{C}\ \sigma$$

Correlation matrix

m x m

Observation error covariance

$$\mathbf{R} = \begin{bmatrix} \sigma_{o1}^2 & 0 & \dots & 0 \\ 0 & \sigma_{o2}^2 & \dots & 0 \\ \vdots & \dots & \ddots & \vdots \\ 0 & \dots & \dots & \sigma_{on}^2 \end{bmatrix}$$

n x n

General Case: cost function

$$
\underbrace{J(x)}_{1\times 1}=\frac{1}{2}\underbrace{(x-x^{b})^{T}}_{1\times m}\underbrace{\mathbf{B}^{-1}}_{m\times m}\underbrace{(x-x^{b})}_{m\times 1}+\frac{1}{2}\underbrace{[\mathbf{H}x-y]^{T}}_{1\times n}\underbrace{\mathbf{R}^{-1}}_{n\times n}\underbrace{[\mathbf{H}x-y]}_{n\times 1}
$$

H maps x to y space, e. g., interpolation.
Terminology in DA: observation operator
Superscript ‘T’: transpose of a vector or matrix,
Superscript ‘-1’: inverse of a symmetric covariance matrix

Minimize J(x) is equivalent to maximize a multi-dimensional Gaussian PDF

Constant * $e^{-J(x)}$

General Case: analytical solution

Again, minimize J requires its gradient (a vector) with respect to x equal to zero:

$$\nabla J_{\mathbf{x}}(\mathbf{x}) = \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_{\mathrm{b}}) - \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}[\mathbf{y} - \mathbf{H}\mathbf{x}] = 0$$

m x 1

This leads to analytical solution for the analysis increment:

$$x^a - x^b = \underbrace{\boxed{\mathbf{BH}^{\mathrm{T}}(\mathbf{HBH}^{\mathrm{T}} + \mathbf{R})^{-1}}}_{\text{Kalman gain matrix}}\underbrace{\boxed{[y - \mathbf{H}x^b]}}_{\text{Innovation or OMB vector}}$$

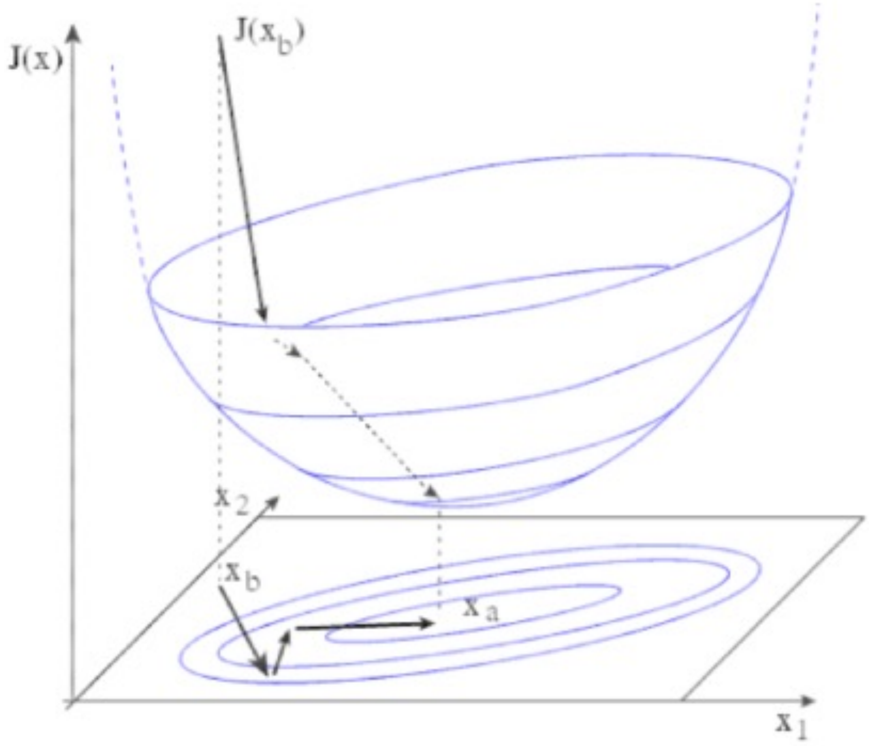
HBH^T : background error covariance projected into observation space

BH^T : background error covariance projected into cross background-observation space

Iterative algorithm to find minimum of cost function

- **Descending algorithms**
  - **Descending direction: γ_n (N-dimensional vector)**
  - **Descending step: μ_n**

$$\boxed{x_{n+1} = x_n + \mu_n \gamma_n}$$



from Bouttier and Courtier 1999

Precision of Analysis with optimal B and R

$$\mathbf{A}^{-1} = \mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}$$

Generalization of scalar case $\frac{1}{\sigma_a^2} = \frac{1}{\sigma_b^2} + \frac{1}{\sigma_o^2}$

Or in another form: $\mathbf{A} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{B}$

With

$$\mathbf{K} = \mathbf{B}\mathbf{H}^{\mathrm{T}}(\mathbf{H}\mathbf{B}\mathbf{H}^{\mathrm{T}} + \mathbf{R})^{-1}$$

called Kalman gain matrix

Precision of analysis: more general formulation

$$\mathbf{A} = (\mathbf{I} - \mathbf{KH})\mathbf{B}_t(\mathbf{I} - \mathbf{KH})^{\mathrm{T}} + \mathbf{K}\mathbf{R}_t\mathbf{K}^{\mathrm{T}}$$

where B_t and R_t are "true" background and observation error covariances.

This formulation is valid for any given gain matrix K, which could be suboptimal (e.g., due to incorrect estimation/specification of B and R).

Analysis increment with a single humidity observation

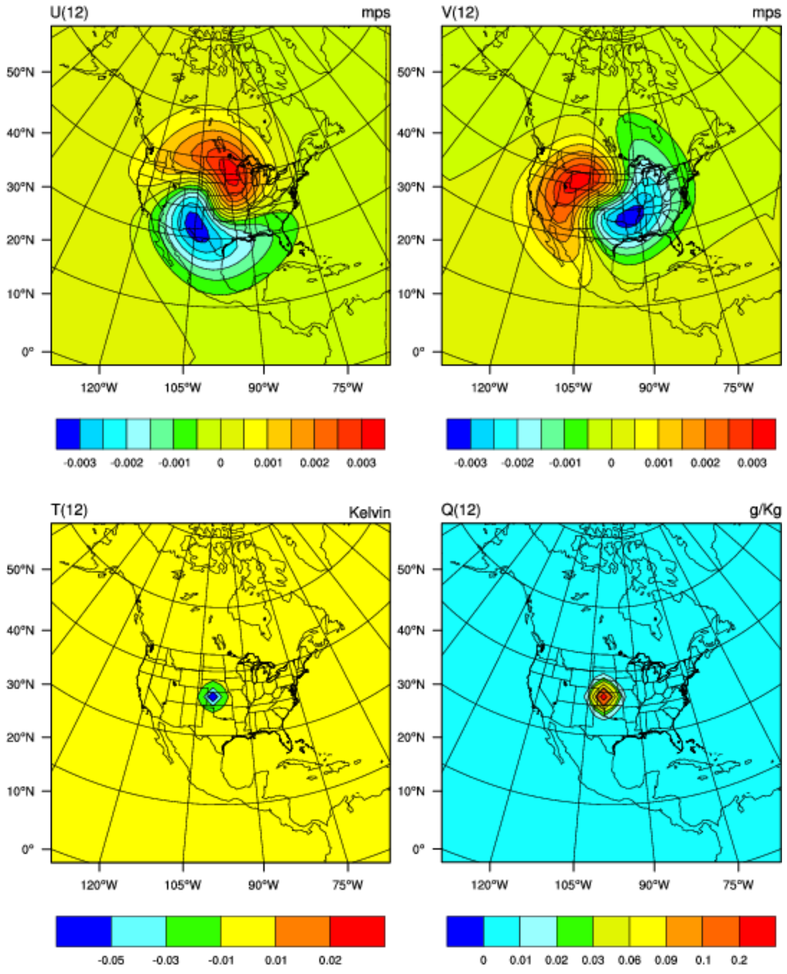
$$x^a - x^b = \mathbf{BH}^{\mathrm{T}}(\mathbf{HBH}^{\mathrm{T}} + \mathbf{R})^{-1}[y - \mathbf{H}x^b]$$

$$x_l^a - x_l^b = \frac{c_{lk}\sigma_l\sigma_k}{\sigma_k^2 + \sigma_{ok}^2}(y_k - x_k^b)$$

It is generalization of previous two variables case:

$$x_1^a - x_1^b = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_o^2}(y_1 - x_1^b)$$

$$x_2^a - x_2^b = \frac{c\sigma_1\sigma_2}{\sigma_1^2 + \sigma_o^2}(y_1 - x_1^b)$$

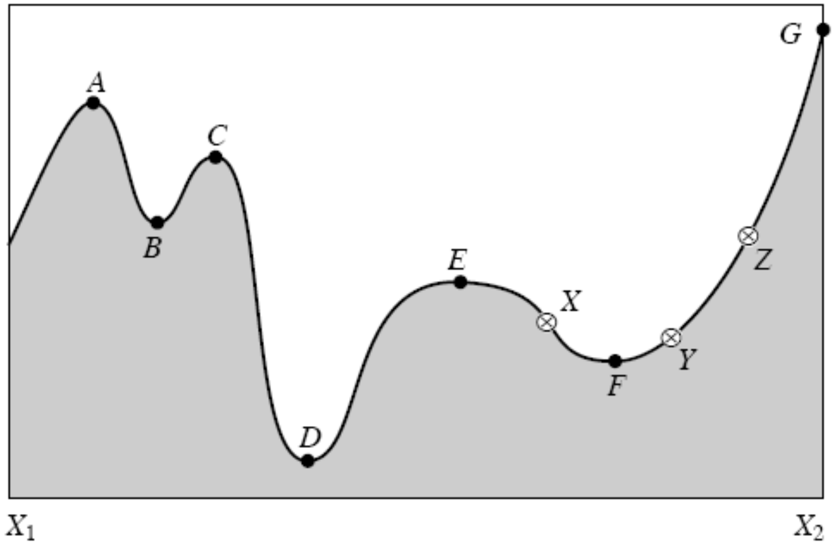


cv_options=6 in WRFDA

Other Remarks

- Observation operator H() can be non-linear and thus analysis error PDF is not necessarily Gaussian

- J(x) can have multiple local minima. Final solution of least square depends on starting point of iteration, e.g., choose the background x_b as the first guess.



.

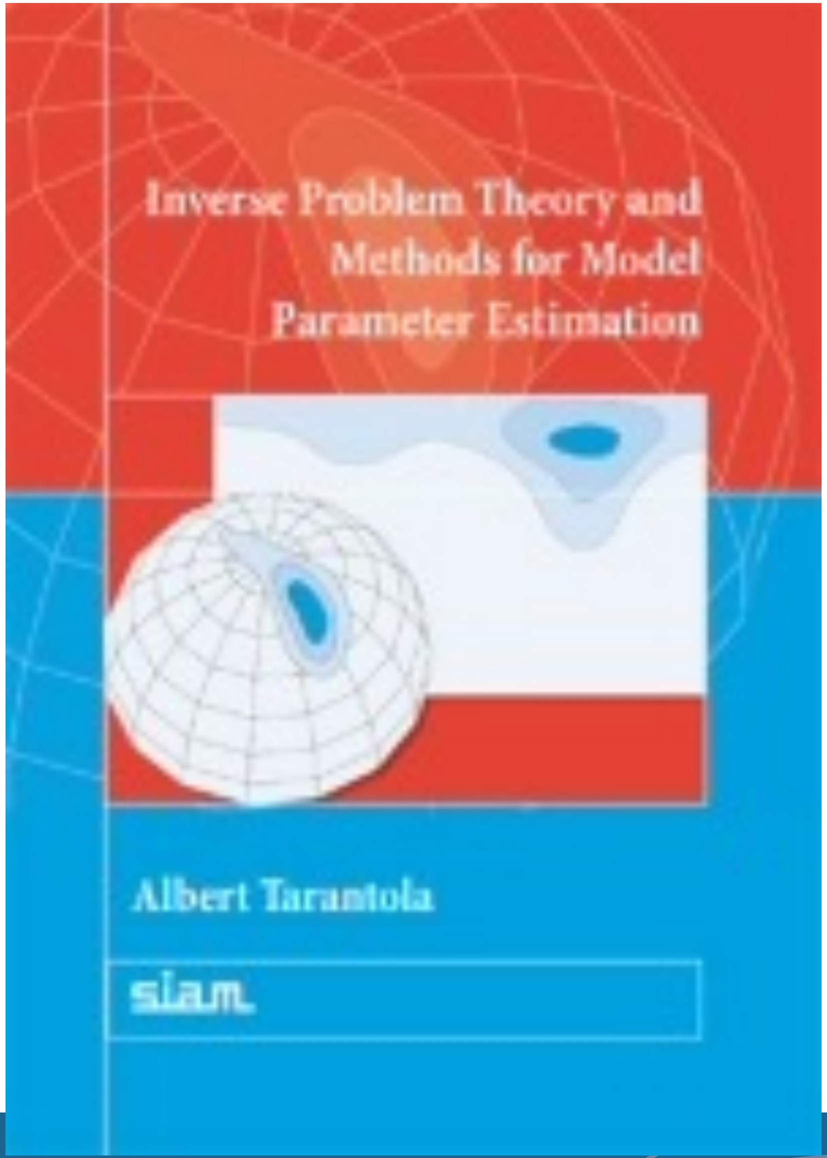
Other Remarks

- **B** matrix is of very large dimension, explicit inverse of **B** is impossible, substantial efforts in data assimilation were given to the estimation and modeling of **B**.
- **B** shall be spatially-varied and time-evolving according to weather regime.
- Analysis can be sub-optimal if using inaccurate estimate of **B** and **R**.
- Could use non-Gaussian PDF
  - Thus not a least square cost function
  - Difficult (usually slow) to solve; could transform into Gaussian problem via variable transform

Variational vs. Ensemble DA

- They are solving the same cost function, by using different techniques
- These days, combining both techniques are common at operational centers
  - NOAA/NCEP: hybrid-4DEnVar + LETKF
  - ECMWF: ensemble of 4DVar
  - UKMO: hybrid-4DVar + LETKF

Further reading



MPAS-JEDI Overview

Jake Liu

Mesoscale & Microscale Meteorology Laboratory
National Center for Atmospheric Research

NCAR

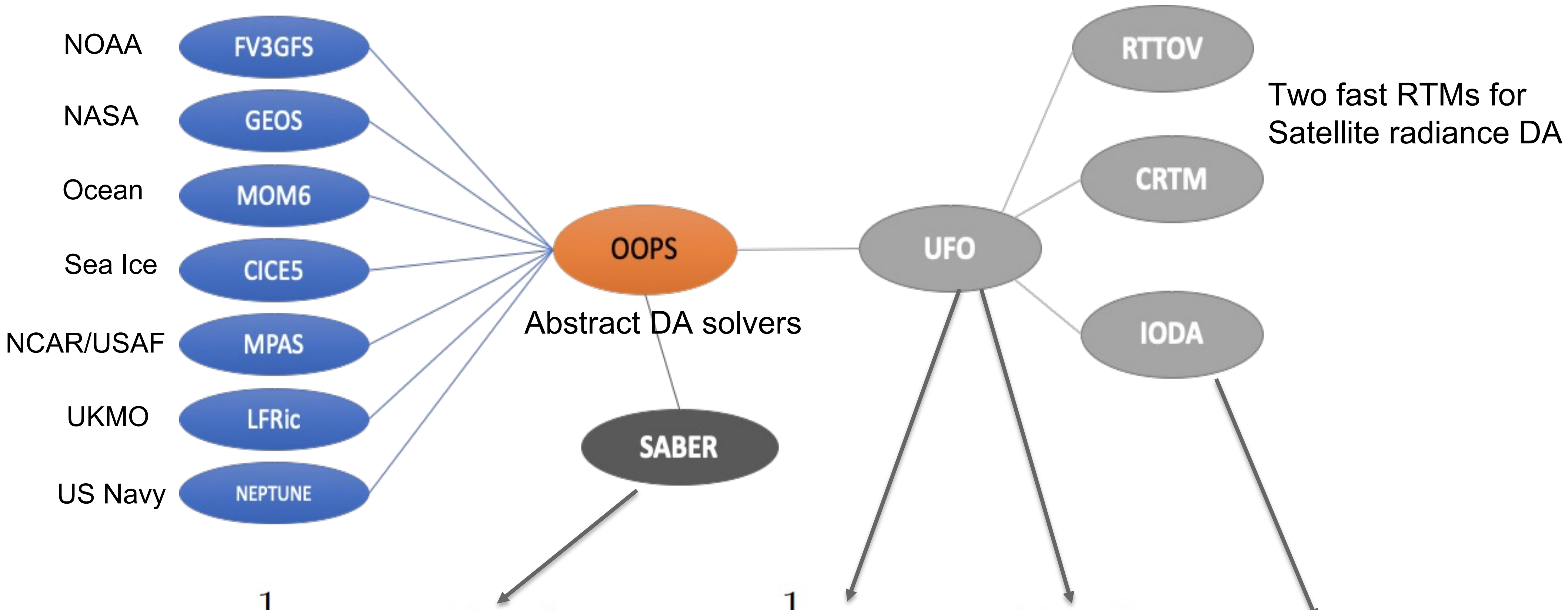
MPAS-JEDI Tutorial, INPE, 15-16 August, 2024

NSF

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

Joint Effort for Data assimilation Integration (JEDI)

led by Joint Center for Satellite Data Assimilation (JCSDA)



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

JCSDA and all partner groups contributing to JEDI's development

https://github.com/JCSDA/mpas-jedi/blob/release/2.0.0/src/mains/mpasVariational.cc

MPAS-JEDI
C++ main program
for Variational DA

```cpp
#include <oops/runs/Run.h>
#include <oops/runs/Variational.h>

#include <saber/oops/instantiateCovarFactory.h>
#include <saber/oops/instantiateLocalizationFactory.h>

#include <ufo/instantiateObsFilterFactory.h>
#include <ufo/ObsTraits.h>

#include "mpasjedi/Traits.h"

int main(int argc,  char ** argv) {
  oops::Run run(argc, argv);
  saber::instantiateCovarFactory<mpas::Traits>();
  saber::instantiateLocalizationFactory<mpas::Traits>();
  ufo::instantiateObsFilterFactory();
  oops::Variational<mpas::Traits, ufo::ObsTraits> var;
  return run.execute(var);
}
```

Model-agnostic components of JEDI

- OOPS: Object Oriented Prediction System, https://github.com/JCSDA/oops
  - Originally from ECMWF, JCSDA's OOPS version is diverged from ECMWF
  - Abstract definition of data assimilation elements, e.g., x, B, y, R, H etc.
  - Multiple minimization algorithms for variational DA
  - DA solvers for ensemble of DA and LETKF
  - Actual DA implementation for toy models like Lorenz95 and QG model
  - Mostly written in C++ with some Fortran

- SABER: System-Agnostic Background-Error Representation, https://github.com/JCSDA/saber
  - Implementation of static B models (currently 4) and localization of ensemble covariance
  - **BUMP: Background error on Unstructured Mesh Package, used by MPAS-JEDI**
  - Under development: GSI's grid-point B model, UKMO's spectral B model, diffusion operator
  - BUMP mostly written in Fortran

Model-agnostic components of JEDI

- UFO: Unified Forward Operator, https://github.com/JCSDA/ufo
  - Implementation of observation operators (including Tangent Linear/Adjoint/Jacobian) or interface to observation operators (e.g., CRTM/RTTOV for satellite radiance, ROPP for GNSSRO, radar)
  - Quality control of observations
  - Thinning of observations
  - Observation error modelling
  - Bias correction, e.g., variational bias correction for radiance data
  - C++ and Fortran

- IODA: Interface for Observation Data Access, https://github.com/JCSDA/ioda
  - In-memory observational data structure
  - In-disk file I/O: HDF5 (used by mpas-jedi now) and ODB
  - C++ and Fortran

MPAS-specific interface to JEDI

- https://github.com/JCSDA/mpas-jedi
  – Horizontal and vertical model grids
  – Prognostic variables to/from analysis variables
  – Adoption of static B model
  – Supply input variables of observation operators in UFO
  – State variable data structure, parallelism, I/O follows that of MPAS-A model, **so need MPAS-A model code in the compilation of MPAS-JEDI**
  – Mostly written in Fortran
- https://github.com/JCSDA-internal/MPAS-Model
  – A modified version of the MPAS-A model, currently used by MPAS-JEDI
  – Will be merged back to the official MPAS repository
  – Note: we use ‘mpasout’ (instead of ‘restart’) file for DA background and analysis

MPAS-JEDI 2.0.0, code as of early June 2023

Begin development from early 2018

- MPAS-JEDI: a collection (bundle) of github code repositories with
  - **Model-agnostic components**, led by JCSDA and contributed by all partners
  - **MPAS-specific interfaces**, led/developed by NCAR/MMM
- MPAS-JEDI 2.0 code accessible from
  - https://github.com/JCSDA/mpas-bundle/tree/release/2.0.0

Model-agnostic components:
https://github.com/JCSDA/oops
https://github.com/JCSDA/saber
https://github.com/JCSDA/ufo
https://github.com/JCSDA/ioda

MPAS-A model and model-specific interfaces:
https://github.com/JCSDA-internal/MPAS-Model
https://github.com/JCSDA/mpas-jedi

Python-based Diagnostic/Verification package included in:
https://github.com/JCSDA/mpas-jedi/tree/release/2.0.0/graphics

Observation processing, format conversion:
https://github.com/NCAR/obs2ioda

Data assimilation cycling Workflow based on cylc:
https://github.com/NCAR/MPAS-Workflow

MPAS
Model for Prediction Across Scales

https://www2.mmm.ucar.edu/projects/mpas-jedi/tutorial/202309NCAR/

Welcome to the MPAS-JEDI tutorial practice guide

This web page is intended to serve as a guide through the practice exercises of this tutorial. Exercises are split into seven main sections, each of which focuses on a particular aspect of using the MPAS-JEDI data assimilation system.

In case you would like to refer to any of the lecture slides from previous days, you can open the Tutorial Agenda in another window. The test dataset can be downloaded from Here.

You can proceed through the sections of this practical guide at your own pace. It is highly recommended to go through the exercises in order, since later exercises may require the output of earlier ones. Clicking the grey headers will expand each section or subsection.

0. Prerequisites and environment setup

1. Compiling/Testing MPAS-JEDI

2. Converting NCEP BUFR obs into IODA-HDF5 format

3. Running MPAS-JEDI's HofX application

4. Generating localization files and running 3D/4DEnVar with "conventional" obs

5. Running 3DVar and hybrid-3DEnVar

6. Running EDA and LETKF

7. Plotting OMB/OMA from two experiments

8. Running regional MPAS-JEDI

The first MPAS-JEDI tutorial

In September 2023

Plan to make MPAS-JEDI 3.0.0 release, this summer

We use 3.0.0-beta for this tutorial

- Previous releases (1.0.0 and 2.0.0) of MPAS-JEDI use
  - https://github.com/JCSDA-internal/MPAS-Model, 7.x-based
- From MPAS-JEDI 3.0.0, will use the official MPAS model repository
  - https://github.com/MPAS-Dev/MPAS-Model, 8.2.1-based
- MPAS-A/MPAS-JEDI tutorials: Sept. 30 to Oct. 4, 2024, Howard University

Main features of current MPAS-JEDI

- Deterministic analysis:
  - **3DVar, 3D/4DEnVar**, and **hybrid-3D/4DEnVar with dual-resolution capability**
  - Multivariate static B modeling follows WRFDA/GSI, but via BUMP
- Ensemble analysis:
  - Ensemble of EnVar (**EDA**) with perturbed observations
  - **LETKF (newly enabled in MPAS-JEDI 2.0.0, recently began cycling experiments)**
- Analysis directly done on **MPAS unstructured grid** for uniform or **variable-resolution mesh, global** or **regional mesh**
- Analysis variables: (T, Q, U, V, Ps) at cell center, + hydrometeors (optional)
- Appy linear hydrostatic balance constrain to the analysis increment

Satellite Radiance DA capability with MPAS-JEDI

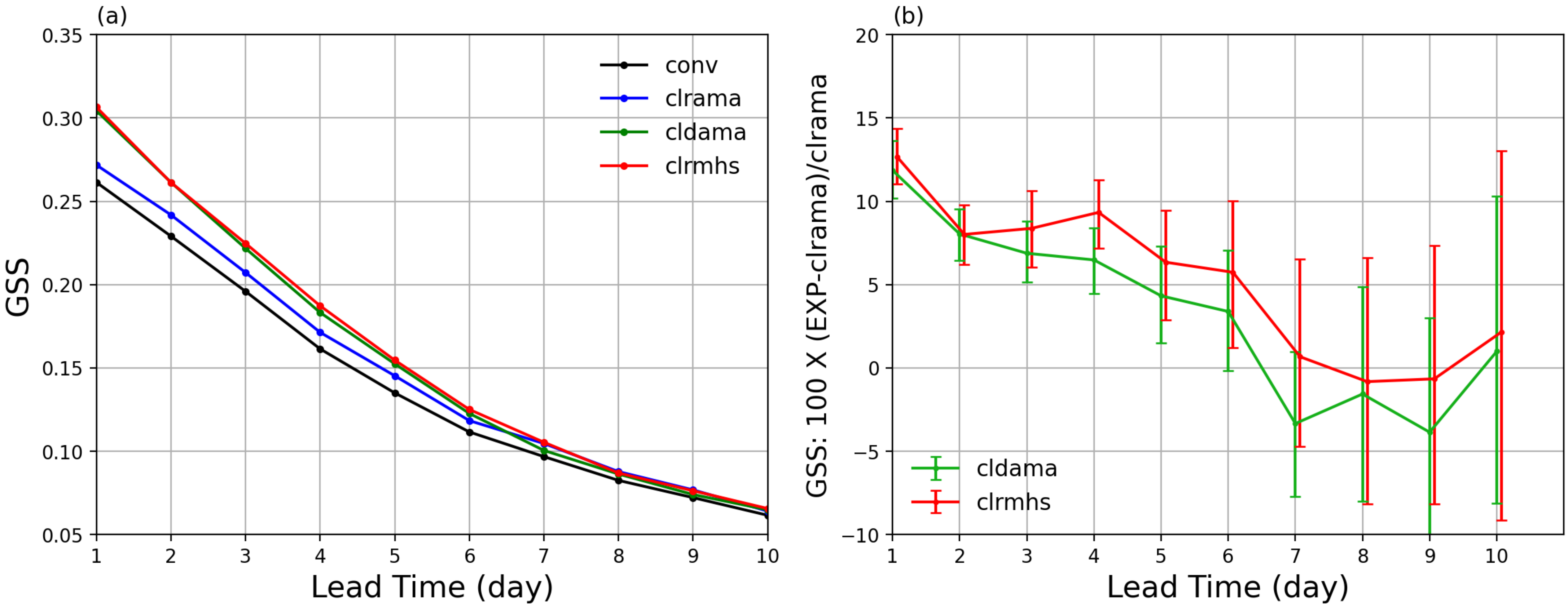
- So far MMM's MPAS-JEDI team mostly uses CRTM for radiance DA, though RTTOV could also be used

- Leverage comprehensive satellite radiance DA capability contributed by multiple groups

- Allow all-sky radiance DA with mixing ratios of hydrometeors as part of analysis variable

- So far MMM's MPAS-JEDI team have experimented several MW and IR sensors
  - Microwave: AMSU-A, MHS, ATMS
  - Infrared: ABI, AHI, IASI

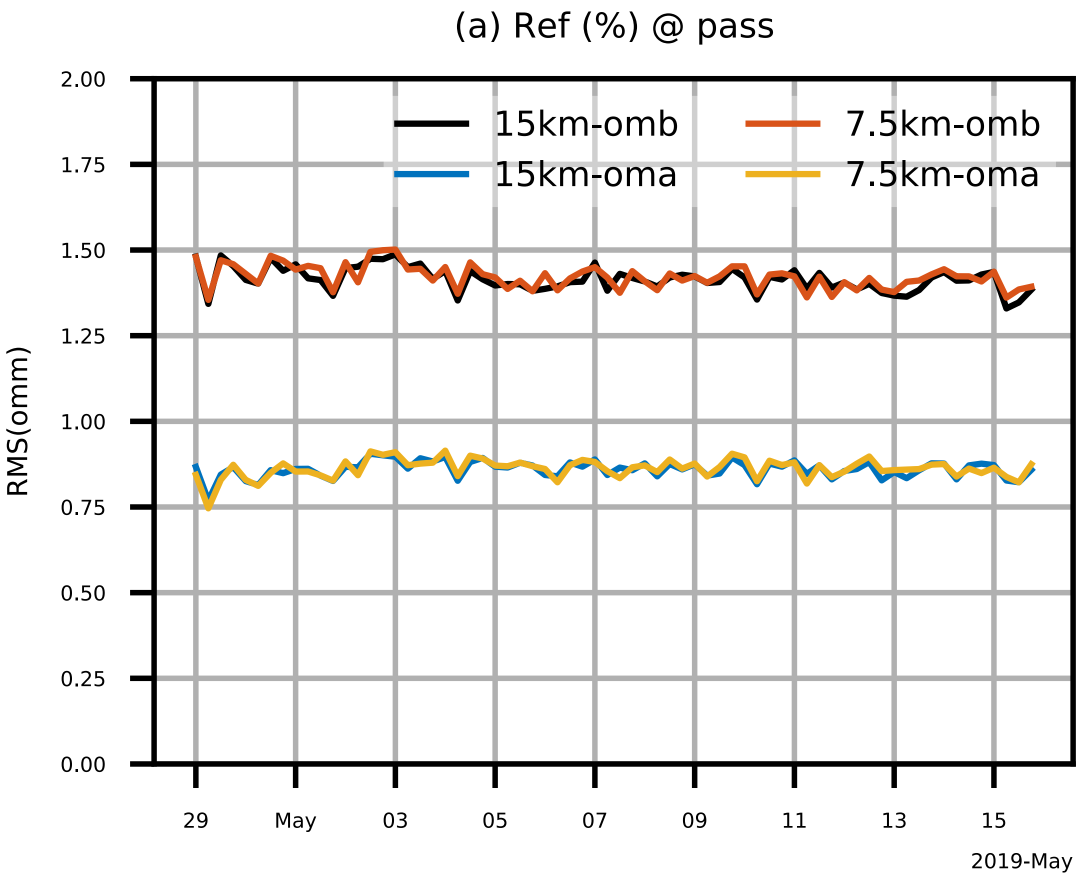
ETS Score for 1-10-day rainfall forecast w.r.t. CMORPH obs



Liu et al., 2022

2.0 code allows high-resolution global DA at 7.5km (>10M cells):

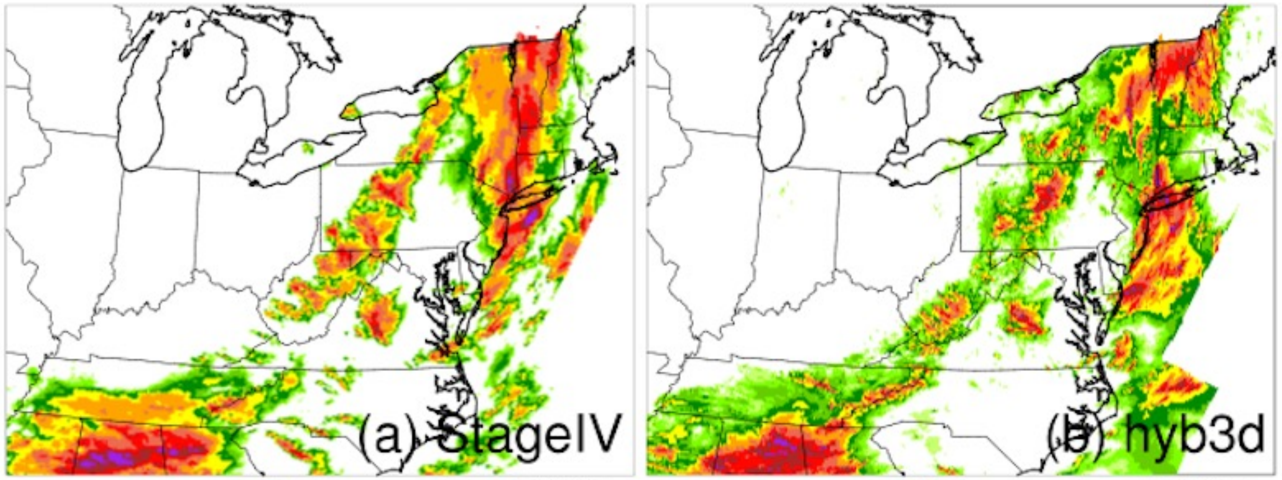
7.5km-15km dual-res. 3DEnVar with 80-member 15km ensemble input

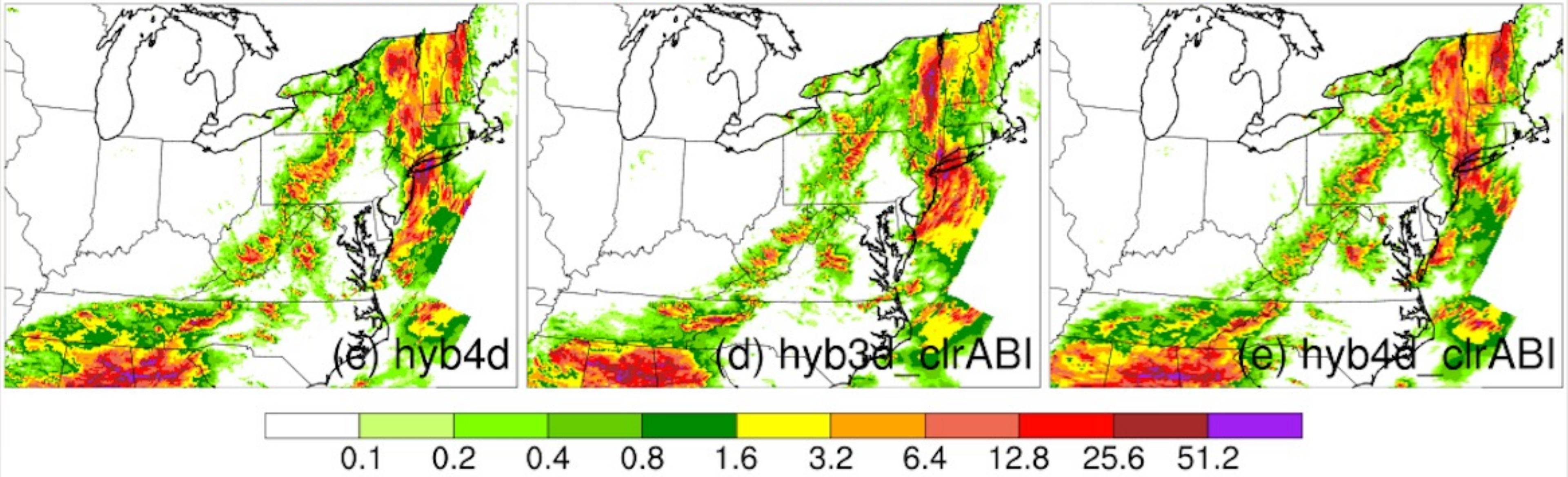


OMB/OMA of GNSSRO Refractivity

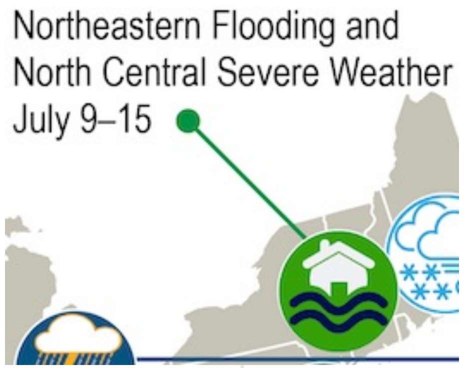
0-6-h accumulated rainfall from 00 UTC, 10 July 2023

Stage-IV obs **Hybrid-3DEnVar**





This is a billion $ disaster event

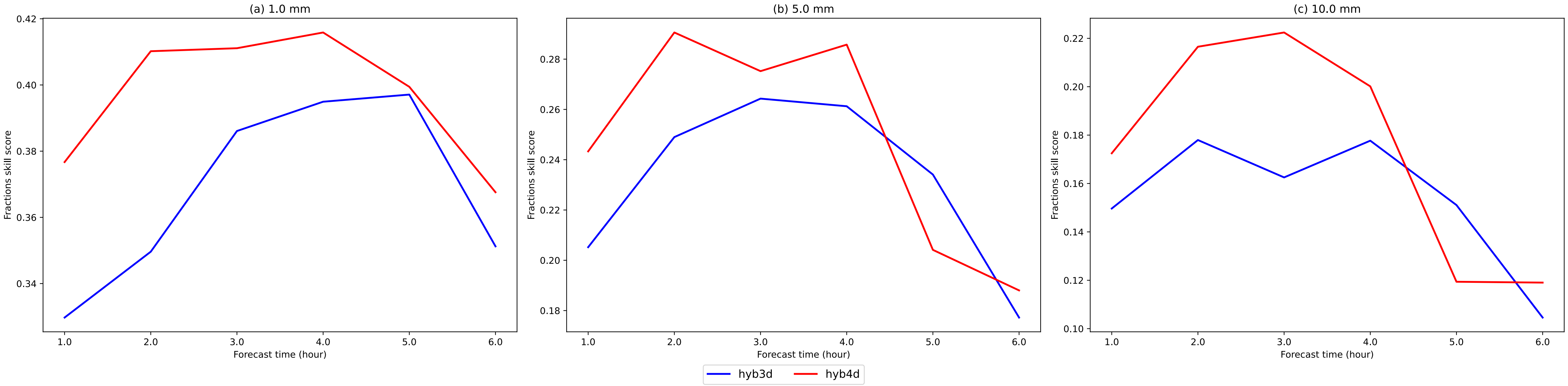


Regional MPAS-JEDI at 3.75km grid spacing over US

Hybrid-4DEnVar **Hybrid-3DEnVar w/ ABI** **Hybrid-4DEnVar w/ hourly ABI**

1-h accumulated rainfall forecast FSS scores: 1h - 6h lead time

Hybrid-3DEnVar vs. Hybrid-4DEnVar (without ABI)



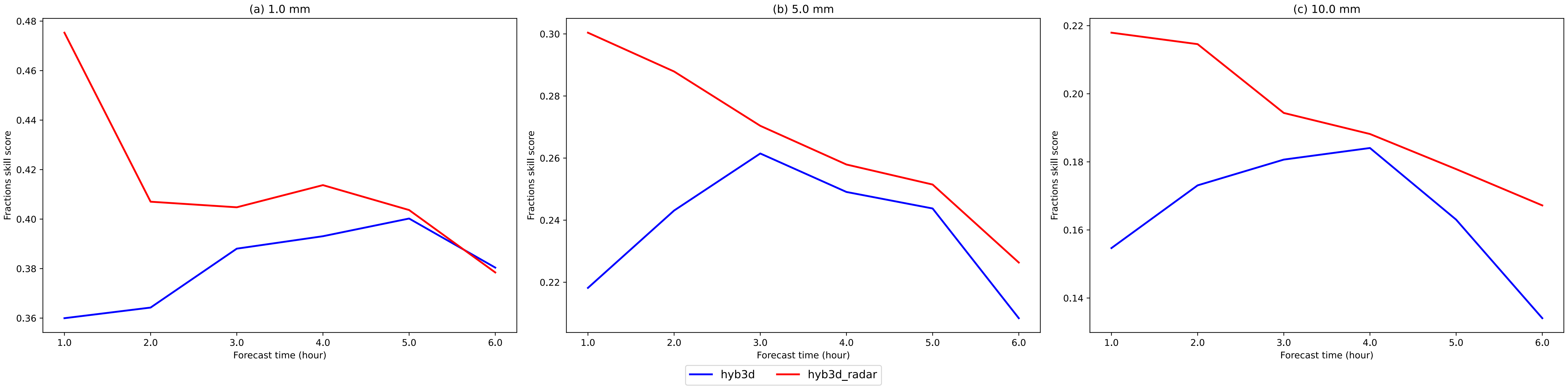
Clear improvement for the first several hours from hybrid-4DEnVar

Fraction Skill Scores (FSS) computed against Stage-IV obs with a radius of 25km, from 21 forecasts from 00 UTC 8 to 00 UTC 13 July.

NCAR | UCAR

Preliminary Radar DA

Hybrid-3DEnVar: without vs. with radar (radial wind + reflectivity)



33 forecasts from 00 UTC 9 to 18 UTC 17, July

MPAS-JEDI publications

EnVar and all-sky AMSU-A DA

Liu Z et al., 2022: Data Assimilation for the Model for Prediction Across Scales - Atmosphere with the Joint Effort for Data assimilation Integration (JEDI-MPAS 1.0.0): EnVar implementation and evaluation, Geosci. Model Dev., *15,* 7859–7878.

EDA

Guerrette, J. J. et al., 2023: Data assimilation for the Model for Prediction Across Scales – Atmosphere with the Joint Effort for Data assimilation Integration (JEDI-MPAS 2.0.0-beta): ensemble of 3D ensemble-variational (En-3DEnVar) assimilations, Geosci. Model Dev., 16, 7123–7142.

3DVar and multivariate background error covariance

Jung et al.. 2024: Three-dimensional variational assimilation with a multivariate background error covariance for the Model for Prediction Across Scales–Atmosphere with the Joint Effort for data Assimilation Integration (JEDI-MPAS 2.0.0-beta), Geosci. Model Dev., 17, 3879–3895.

This material is based upon work supported by the NSF National Center for Atmospheric Research, which is a major facility operated by the U.S. National Science Foundation under Cooperative Agreement No. 1852977.

Introduction to Practical Sessions

Presented by Ivette Hernández Baños based on materials prepared by I-Han Chen

Prediction, Assimilation, and Risk Communication Section
Mesoscale & Microscale Meteorology Laboratory
National Center for Atmospheric Research

MONAN: INPE MPAS-JEDI Training 2024, Cachoeira Paulista, São Paulo, Brazil
August 15-16, 2024

Instructions for MPAS-JEDI practice exercises

https://www2.mmm.ucar.edu/projects/mpas-jedi/tutorial/202408INPE

EGEON cluster computer system

ssh -X username@egeon-login.cptec.inpe.br

Check what shell you are using
- echo $SHELL

*** use bash**

Submitting Jobs with Slurm

sbatch	Submit a job script
squeue -u $USER	Check the status of your pending and running jobs
scancel	Delete a queued or running job

It is recommended that all code be **built** on the "**egeon-login**.cptec.inpe.br" host rather than a **head node**

the **"head node" can be very slow!!!*

Obtain the mpas_jedi_tutorial folder

cd /mnt/beegfs/${USER}

cp -r /mnt/beegfs/professor/mpasjedi_tutorial/mpasjedi_tutorial2024_testdata ./mpas_jedi_tutorial

(this will create your own working directory that contains all data needed for the tutorial: /mnt/beegfs/${USER}/mpas_jedi_tutorial)

ls -l mpas_jedi_tutorial, *you will see:*

```
total 7
drwxr-sr-x 3 professor professor   1 Aug  7 19:40 background
drwxr-sr-x 3 professor professor   1 Aug  7 19:36 background_120km
drwxr-sr-x 5 professor professor   3 Aug  7 19:39 B_Matrix
drwxr-sr-x 4 professor professor  38 Aug  9 06:18 conus15km
drwxr-sr-x 2 professor professor  27 Aug  7 19:36 crtm_coeffs_v3
drwxr-sr-x 3 professor professor   1 Aug  7 19:36 ensemble
drwxr-sr-x 2 professor professor 168 Aug  7 19:36 localization_pregenerated
drwxr-sr-x 2 professor professor  35 Aug 10 00:59 MPAS_JEDI_yamls_scripts
drwxr-sr-x 2 professor professor  30 Aug  8 04:13 MPAS_namelist_stream_physics_files
drwxr-sr-x 2 professor professor   6 Aug  7 19:40 ncl_scripts
drwxrwsr-x 5 professor professor   7 Aug  9 21:58 obs2ioda_prebuild
drwxr-sr-x 3 professor professor   7 Aug  8 19:11 obs_bufr
drwxr-sr-x 3 professor professor   1 Aug  7 19:35 obs_ioda_pregenerated
drwxr-sr-x 4 professor professor   2 Aug  9 00:41 omboma_from2experiments
```

Build the MPAS-JEDI and its dependencies

1. Generate build files *(cmake, CMakeLists.txt)*
2. Compile MPAS-JEDI executables *(make)*
3. Test if the code was compiled properly *(ctest)*

Required spack-stack build environment

This tutorial does not cover the installation of spack-stack, which was pre-installed on EGEON

- source ../code/env-setup/gnu-<machine>.sh
- module list (100 modules!!!!!)

```
Currently Loaded Modules:
  1) ecflow/5.8.4
  2) mysql/8.0.33
  3) ncarenv/23.09          (S)
  4) gcc/12.2.0
  5) stack-gcc/12.2.0
  6) craype/2.7.20
  7) cray-mpich/8.1.25
  8) libfabric/1.15.2.0
  9) cray-pals/1.2.11
 10) stack-cray-mpich/8.1.25
 11) tar/1.34
 12) gettext/0.21.1
 13) libxcrypt/4.4.35
 14) zlib/1.2.13
 15) sqlite/3.43.2
 16) util-linux-uuid/2.38.1
 17) python/3.10.13
 18) stack-python/3.10.13
 19) nghttp2/1.57.0
 20) curl/8.4.0
 21) cmake/3.23.1
 22) git/2.41.0
 23) pkg-config/0.29.2
 24) hdf5/1.14.0
 25) snappy/1.1.10
 26) zstd/1.5.2
 27) c-blosc/1.21.5
 28) netcdf-c/4.9.2
 29) nccmp/1.9.0.1
 30) netcdf-fortran/4.6.1
 31) parallel-netcdf/1.12.2
 32) parallelio/2.5.10
 33) py-pip/23.1.2
 34) wget/1.20.3
 35) base-env/1.0.0
 36) boost/1.83.0
 37) openblas/0.3.24
 38) py-setuptools/63.4.3
 39) py-numpy/1.22.3
 40) bufr/12.0.1
 41) ecbuild/3.7.2
 42) libpng/1.6.37
 43) openjpeg/2.3.1
 44) eccodes/2.32.0
 45) eigen/3.4.0
 46) eckit/1.24.5
 47) fftw/3.3.10
 48) fckit/0.11.0
 49) fiat/1.2.0
 50) ectrans/1.2.0
 51) qhull/2020.2
 52) atlas/0.35.1
 53) git-lfs/3.3.0
 54) gsibec/1.1.3
 55) gsl-lite/0.37.0
 56) libjpeg/2.1.0
 57) krb5/1.19.2
 58) libtirpc/1.3.3
 59) hdf/4.2.15
 60) jedi-cmake/1.4.0
 61) libxt/1.1.5
 62) libxmu/1.1.4
 63) libxpm/3.5.12
 64) libxaw/1.0.13
 65) udunits/2.2.28
 66) ncview/2.1.9
 67) netcdf-cxx4/4.3.1
 68) json/3.10.5
 69) json-schema-validator/2.1.0
 70) odc/1.4.6
 71) py-attrs/21.4.0
 72) py-pycparser/2.21
 73) py-cffi/1.15.1
 74) py-findlibs/0.0.2
 75) py-eccodes/1.5.0
 76) py-f90nml/1.4.3
 77) py-h5py/3.7.0
 78) py-cftime/1.0.3.4
 79) py-netcdf4/1.5.8
 80) py-bottleneck/1.3.7
 81) py-numexpr/2.8.4
 82) py-et-xmlfile/1.0.1
 83) py-openpyxl/3.1.2
 84) py-six/1.16.0
 85) py-python-dateutil/2.8.2
 86) py-pytz/2023.3
 87) py-pyxlsb/1.0.10
 88) py-xlrd/2.0.1
 89) py-xlsxwriter/3.1.7
 90) py-xlwt/1.3.0
 91) py-pandas/1.5.3
 92) py-pybind11/2.11.0
 93) py-pycodestyle/2.11.0
 94) py-pyhdf/0.10.4
 95) libyaml/0.2.5
 96) py-pyyaml/6.0
 97) py-scipy/1.11.3
 98) py-packaging/23.1
 99) py-xarray/2023.7.0
100) sp/2.5.0
101) jedi-base-env/1.0.0
102) jedi-mpas-env/1.0.0

  Where:
   S:  Module is Sticky, requires --force to unload or purge
```

More information (https://github.com/JCSDA/spack-stack)

Required spack-stack build environment

This tutorial does not cover the installation of spack-stack, which was pre-installed on EGEON

- **source ../code/env-setup/gnu-egeon.sh**
- **module list (here reduced to 31 modules!!!)**

```
Currently Loaded Modules:
  1) stack-gcc/12.2.0       7) libfabric/1.21.0  13) gsl-lite/0.37.0          19) parallelio/2.6.2
  2) binutils/2.42          8) atlas/0.36.0      14) hdf5/1.14.3              20) boost/1.84.0
  3) gcc/12.2.0             9) eckit/1.24.5      15) netcdf-c/4.9.2           21) cmake/3.27.9
  4) openssh/9.7p1         10) fckit/0.11.0      16) netcdf-cxx4/4.3.1        22) jedi-cmake/1.4.0
  5) openmpi/5.0.3         11) fftw/3.3.10       17) netcdf-fortran/4.6.1     23) ecbuild/3.7.2
  6) stack-openmpi/5.0.3   12) gptl/8.1.1        18) parallel-netcdf/1.12.3   24) eigen/3.4.0

 25) python-venv/1.0         31) nccmp/1.9.0.1
 26) py-setuptools/63.4.3
 27) py-pycodestyle/2.11.0
 28) sqlite/3.43.2
 29) openblas/0.3.24
 30) udunits/2.2.28
```

More information (https://github.com/JCSDA/spack-stack)

1. Generate build files (cmake, CMakeLists.txt)

Clone mpas-bundle repository and checkout the 'release/3.0.0-beta' branch

- **cd /mnt/beegfs/${USER}/mpas_jedi_tutorial**
- **mkdir mpas_bundle_v3**
- **cd mpas_bundle_v3**
- **git clone -b release/3.0.0-beta https://github.com/liujake/mpas-bundle code**

```
Cloning into 'code'...
remote: Enumerating objects: 485, done.
remote: Counting objects: 100% (106/106), done.
remote: Compressing objects: 100% (60/60), done.
remote: Total 485 (delta 60), reused 86 (delta 46), pack-reused 379
Receiving objects: 100% (485/485), 155.47 KiB | 1.49 MiB/s, done.
Resolving deltas: 100% (289/289), done.
```

The mpas-bundle repository does not contain actual source code. Instead, you will obtain the CMakeLists.txt file under code.

1. Generate build files (cmake, CMakeLists.txt)

- **vi code/CMakeLists.txt**

JEDI components

Repositories and branch/tag information

```
39    ecbuild_bundle( PROJECT crtm   GIT "https://github.com/JCSDA/crtm.git"    TAG bfede42 )
40    ecbuild_bundle( PROJECT oops   GIT "https://github.com/JCSDA/oops.git"    TAG 5fca331 )
41    ecbuild_bundle( PROJECT saber  GIT "https://github.com/JCSDA/saber.git"   TAG 1c35ddd )
42    ecbuild_bundle( PROJECT ioda   GIT "https://github.com/JCSDA/ioda.git"    TAG 26e8a8e )
43    ecbuild_bundle( PROJECT ufo    GIT "https://github.com/JCSDA/ufo.git"     TAG 5e3d981 )
76    set(MPAS_DOUBLE_PRECISION "ON" CACHE STRING "MPAS-Model: Use double precision 64-bit Floating point.")
77    set(MPAS_CORES init_atmosphere atmosphere CACHE STRING "MPAS-Model: cores to build.")
78    ecbuild_bundle( PROJECT MPAS GIT "https://github.com/JCSDA-internal/MPAS-Model.git" TAG jedi-2.0.0 )
79    ecbuild_bundle( PROJECT mpas-jedi GIT "https://github.com/JCSDA/mpas-jedi"  TAG bae33fb )
```

default build option

Note 1: MPAS-Model inside mpas-bundle is built in double precision by default, it is suggested to build it in single precision for production by changing Line76 from “ON” to “OFF”.

Note 2: We will use the pre-build single-precision mpas-bundle executable in practical sessions

1. Generate build files (cmake, CMakeLists.txt)

Use cmake to generate build files

- **mkdir build ; cd build**

  (We will compile the executables under build)

- **cmake ../code**
  - *git clone repos in CMakeLists.txt into ../code*
  - *generate makefiles under build*

2. Compile MPAS-JEDI executables (make)

- **make -j14**

(compile MPAS-JEDI using a login node with 14 cores)
(The compilation will take ~14 min to complete)

MPAS-JEDI related executables under ~build/bin

```
mpas_atmosphere                          MPAS-Atmosphere forecast mode
mpas_atmosphere_build_tables
mpas_data_checker.py
mpas_data_downloader.py
mpas_init_atmosphere                     MPAS-Atmosphere init core
mpasjedi_convertstate.x
mpasjedi_dirac.x
mpasjedi_eda.x                           for EDA
mpasjedi_enkf.x                          for LETKF
mpasjedi_enshofx.x
mpasjedi_error_covariance_training.x     for doing statistics of static B
mpasjedi_forecast.x
mpasjedi_gen_ens_pert_B.x
mpasjedi_hofx3d.x
mpasjedi_hofx.x
mpasjedi_rtpp.x
mpasjedi_staticbinit.x
mpasjedi_variational.x                   for 3DVar, 3D/4DEnVar, hybrid-3D/4DEnVAR
mpas_namelist_gen
mpas_parse_atmosphere
mpas_parse_init_atmosphere
mpas_streams_gen
```

How to obtain, compile, and test the MPAS-JEDI for this tutorial?

We will use a **'local'** version of the source code and test dataset:

cd /mnt/beegfs/${USER}/mpas_jedi_tutorial

cp /mnt/beegfs/professor/mpasjedi_tutorial/mpasbundlev3_local.tar.gz .

tar xzf mpasbundlev3_local.tar.gz; ls -l mpasbundlev3_local/*

```
mpasbundlev3_local/build:
total 1
drwxrwsr-x 3 professor professor 1 Aug 13 21:15 test_data

mpasbundlev3_local/code:
total 34
-rw-rw-r--  1 professor professor  5032 Aug 13 21:28 CMakeLists.txt
drwxrwsr-x 11 professor professor    21 Aug 13 21:08 crtm
drwxrwsr-x  2 professor professor     7 Aug 14 08:58 env-setup
drwxrwsr-x 10 professor professor    18 Aug 13 21:09 ioda
drwxrwsr-x  5 professor professor     5 Aug 13 21:12 ioda-data
-rw-rw-r--  1 professor professor 11334 Aug 13 21:09 LICENSE
drwxrwsr-x  8 professor professor    12 Aug 13 21:09 MPAS
drwxrwsr-x 11 professor professor    17 Aug 13 21:09 mpas-jedi
drwxrwsr-x  4 professor professor     8 Aug 13 23:35 mpas-jedi-data
drwxrwsr-x 13 professor professor    23 Aug 13 21:08 oops
-rw-rw-r--  1 professor professor 10146 Aug 13 21:09 README.md
drwxrwsr-x 12 professor professor    22 Aug 13 21:12 saber
drwxrwsr-x  2 professor professor     2 Aug 13 21:10 scripts
drwxrwsr-x  3 professor professor     2 Aug 13 21:12 test-data-release
drwxrwsr-x 11 professor professor    19 Aug 13 21:10 ufo
drwxrwsr-x  6 professor professor     7 Aug 13 23:58 ufo-data
drwxrwsr-x  9 professor professor    14 Aug 13 21:12 vader
```

How to obtain, compile, and test the MPAS-JEDI for this tutorial?

cd mpasbundlev3_local/build

source ../code/env-setup/gnu-egeon.sh

module list

```
Currently Loaded Modules:
  1) stack-gcc/12.2.0       7) libfabric/1.21.0  13) gsl-lite/0.37.0          19) parallelio/2.6.2
  2) binutils/2.42          8) atlas/0.36.0      14) hdf5/1.14.3              20) boost/1.84.0
  3) gcc/12.2.0             9) eckit/1.24.5      15) netcdf-c/4.9.2           21) cmake/3.27.9
  4) openssh/9.7p1         10) fckit/0.11.0      16) netcdf-cxx4/4.3.1        22) jedi-cmake/1.4.0
  5) openmpi/5.0.3         11) fftw/3.3.10       17) netcdf-fortran/4.6.1     23) ecbuild/3.7.2
  6) stack-openmpi/5.0.3   12) gptl/8.1.1        18) parallel-netcdf/1.12.3   24) eigen/3.4.0

 25) python-venv/1.0          31) nccmp/1.9.0.1
 26) py-setuptools/63.4.3
 27) py-pycodestyle/2.11.0
 28) sqlite/3.43.2
 29) openblas/0.3.24
 30) udunits/2.2.28
```

How to obtain, compile, and test the MPAS-JEDI for this tutorial?

CMakeLists.txt was modified for this tutorial so that cmake will **NOT** fetch code and test data from github, instead, it uses the pre-downloaded local **code/testdata** under "**../code**".

"Makefile" files to build executables are generated under the **build** directory

```
make -j18
```

```
$ ls bin/mpas*
bin/mpas_atmosphere
bin/mpas_atmosphere_build_tables
bin/mpas_init_atmosphere
bin/mpasjedi_convertstate.x
bin/mpasjedi_converttostructuredgrid.x
bin/mpasjedi_eda.x
bin/mpasjedi_enkf.x
bin/mpasjedi_enshofx.x
bin/mpasjedi_ens_mean_variance.x
bin/mpasjedi_error_covariance_toolbox.x
bin/mpasjedi_forecast.x
bin/mpasjedi_gen_ens_pert_B.x
bin/mpasjedi_hofx3d.x
bin/mpasjedi_hofx.x
bin/mpasjedi_rtpp.x
bin/mpasjedi_saca.x
bin/mpasjedi_variational.x
bin/mpas_namelist_gen
bin/mpas_parse_atmosphere
bin/mpas_parse_init_atmosphere
bin/mpas_streams_gen
```

3. Test if the code was compiled properly (ctest)

export
LD_LIBRARY_PATH=/mnt/beegfs/${USER}/mpas_jedi_tutorial/mpasbundlev3_local/build/lib:$LD_LIBRARY_PATH

cd mpas-jedi

ctest

```
......
82% tests passed, 10 tests failed out of 57

Label Time Summary:
executable    =  24.35 sec*proc (13 tests)
mpasjedi      = 265.92 sec*proc (57 tests)
mpi           = 261.83 sec*proc (56 tests)
script        = 241.57 sec*proc (44 tests)

Total Test time (real) = 266.22 sec

The following tests FAILED:
          8 - test_mpasjedi_unsinterp_4pe (Failed)
          9 - test_mpasjedi_geometry_iterator_2d_2pe (Failed)
         10 - test_mpasjedi_geometry_iterator_3d_2pe (Failed)
         50 - test_mpasjedi_letkf_3dloc_4pe (Failed)
         51 - test_mpasjedi_lgetkf_4pe (Failed)
         53 - test_mpasjedi_forecast_2pe (Failed)
         54 - test_mpasjedi_parameters_bumpcov_2pe (Failed)
         55 - test_mpasjedi_parameters_bumploc_2pe (Failed)
         56 - test_mpasjedi_3dvar_2pe (Failed)
         57 - test_mpasjedi_3dhybrid_bumpcov_bumploc_2pe (Failed)
```

Ideally: all 57 ctest cases pass!!!

However, 10 cases of using more than 1 core are failing on Egeon for some unknown reason. This will not affect our tutorial test cases.

3. Test if the code was compiled properly (ctest)

Use ctest to ensure that the code was compiled properly

- **cd mpas-jedi**
- **ctest**

(takes ~8 min to finish)

```
      Start 46: test_mpasjedi_3dvar_2pe
46/47 Test #46: test_mpasjedi_3dvar_2pe ..................... Passed 22.43 sec
Start 47: test_mpasjedi_3dhybrid_bumpcov_bumploc_2pe
47/47 Test #47: test_mpasjedi_3dhybrid_bumpcov_bumploc_2pe ... Passed 21.44 sec

100% tests passed, 0 tests failed out of 47

Label Time Summary:

executable = 27.56 secxproc (13 tests)
mpasjedi = 523.69 secxproc (47 tests)
mpi = 503.92 secxproc (43 tests)
script = 496.13 seckproc (34 tests)

Total Test time (real) = 523.74 sec
```

3. Test if the code was compiled properly (ctest)

What a ctest case 'Passed' means?

Each test run will produce text log files
(Under ~mpas_bundle/build/mpas-jedi/test/testoutput)

```
4denvar_bumploc.ref
4denvar_bumploc.run
4denvar_bumploc.run.ref
4denvar_ID.ref          → existing reference file
4denvar_ID.run          → full text log file for the present test
4denvar_ID.run.ref      → shortened reference file
convertstate_bumpi        (part of the 4denvar_ID.run)
convertstate_bumpinterp.run
convertstate_bumpinterp.run.ref
convertstate_unsinterp.ref
```

➔ **4denvar_ID.run.ref** is compared with the existing **4denvar_ID.ref**.

➔ The test is deemed as **"Passed"** if numerical values between the two files are identical or within a **tolerance**.

3. Test if the code was compiled properly (ctest)

‘ctest –N’ will list, but not run 57 test cases

```
Test  #1: mpasjedi_coding_norms
Test  #2: mpas_get_ufo_test_data
Test  #3: mpas_get_crtm_test_data
Test  #4: mpas_get_mpas-jedi_test_data
Test  #5: test_mpasjedi_geometry
Test  #6: test_mpasjedi_state
Test  #7: test_mpasjedi_model
Test  #8: test_mpasjedi_increment
Test  #9: test_mpasjedi_errorcovariance
Test #10: test_mpasjedi_linvarcha
Test #11: test_mpasjedi_unsinterp_4pe
Test #12: test_mpasjedi_geometry_iterator_2d_2pe
Test #13: test_mpasjedi_geometry_iterator_3d_2pe
Test #14: test_mpasjedi_getvalues
Test #15: test_mpasjedi_obslocalization
Test #16: test_mpasjedi_obslocalization_vertical
Test #17: test_mpasjedi_obslocalizations
Test #18: test_mpasjedi_forecast
Test #19: test_mpasjedi_hofx3d
Test #20: test_mpasjedi_hofx
Test #21: test_mpasjedi_convertstate_bumpinterp
Test #22: test_mpasjedi_convertstate_unsinterp
Test #23: test_mpasjedi_parameters_bumpcov
Test #24: test_mpasjedi_parameters_bumploc
```

```
Test #25: test_mpasjedi_dirac_bumpcov
Test #26: test_mpasjedi_dirac_bumploc
Test #27: test_mpasjedi_dirac_noloc
Test #28: test_mpasjedi_3dvar
Test #29: test_mpasjedi_3dvar_bumpcov
Test #30: test_mpasjedi_3denvar_bumploc
Test #31: test_mpasjedi_3denvar_dual_resolution
Test #32: test_mpasjedi_3denvar_2stream_bumploc
Test #33: test_mpasjedi_3denvar_amsua_allsky
Test #34: test_mpasjedi_3denvar_amsua_bc
Test #35: test_mpasjedi_3dhybrid_bumpcov_bumploc
Test #36: test_mpasjedi_3dfgat
Test #37: test_mpasjedi_4denvar_ID
Test #38: test_mpasjedi_4denvar_bumploc
Test #39: test_mpasjedi_eda_3dhybrid
Test #40: test_mpasjedi_rtpp
Test #41: test_mpasjedi_letkf_3dloc_4pe
Test #42: test_mpasjedi_lgetkf_4pe
Test #43: test_mpasjedi_forecast_2pe
Test #44: test_mpasjedi_parameters_bumpcov_2pe
Test #45: test_mpasjedi_parameters_bumploc_2pe
Test #46: test_mpasjedi_3dvar_2pe
Test #47: test_mpasjedi_3dhybrid_bumpcov_bumploc_2pe
```

3. Test if the code was compiled properly (ctest)

Sample yaml files of ctest cases under ~mpas-jedi/test/testinput/

```
3denvar_2stream_bumploc.yaml      eda_3dhybrid_2.yaml            hofx3d.yaml
3denvar_amsua_allsky.yaml         eda_3dhybrid_3.yaml            hofx.yaml
3denvar_amsua_bc.yaml             eda_3dhybrid_4.yaml            increment.yaml
3denvar_bumploc.yaml              eda_3dhybrid.yaml              letkf_2dloc.yaml
3denvar_dual_resolution.yaml      enshofx_1.yaml                 letkf_3dloc.yaml
3dfgat.yaml                       enshofx_2.yaml                 lgetkf.yaml
3dhybrid_bumpcov_bumploc.yaml     enshofx_3.yaml                 linvarcha.yaml
3dvar_bumpcov_ropp.yaml           enshofx_4.yaml                 model.yaml
3dvar_bumpcov_rttovcpp.yaml       enshofx_5.yaml                 namelists
3dvar_bumpcov.yaml                enshofx.yaml                   obslocalizations.yaml
3dvar.yaml                        errorcovariance.yaml           obslocalization_vertical.yaml
4denvar_bumploc.yaml              forecast.yaml                  obslocalization.yaml
4denvar_ID.yaml                   gen_ens_pert_B.yaml            obsop_name_map.yaml
convertstate_bumpinterp.yaml      geometry_iterator_2d.yaml      parameters_bumpcov.yaml
convertstate_unsinterp.yaml       geometry_iterator_3d.yaml      parameters_bumploc.yaml
dirac_bumpcov.yaml                geometry.yaml                  rtpp.yaml
dirac_bumploc.yaml                getvalues.yaml                 state.yaml
dirac_noloc.yaml                  hofx3d_ropp.yaml               unsinterp.yaml
eda_3dhybrid_1.yaml               hofx3d_rttovcpp.yaml
```

JEDI yaml file is like Fortran's namelist

3. Test if the code was compiled properly (ctest)

Further reading about JEDI Testing

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

- JEDI Testing
  - Running ctest
  - Manual Execution
  - The JEDI test suite
  - Tests as Applications
  - Initialization and Execution of Unit Tests
  - Anatomy of a Unit Test
  - Integration and System (Application) Testing
  - JEDI Testing Framework
- Adding a New Test
  - Step 1: Create a File for your Test Application
  - Step 2: Define A Test Fixture
  - Step 3: Define Your Unit Tests
  - Step 4: Register your Unit Tests with eckit
  - Step 6: Create an Executable
  - Step 7: Create a Configuration File
  - Step 8: Register all files with CMake and CTest
  - Adding an Application Test

Example of a JEDI yaml file

```yaml
test:
  float relative tolerance: 0.00000001
  integer tolerance: 0
  reference filename: testoutput/3dvar.ref
  log output filename: testoutput/3dvar.run
  test output filename: testoutput/3dvar.run.ref
cost function:
  cost type: 3D-Var
  window begin: '2018-04-14T21:00:00Z'
  window length: PT6H
  geometry:
    nml_file: "./Data/480km/namelist.atmosphere_2018041500"
    streams_file: "./Data/480km/streams.atmosphere"
  analysis variables: &incvars
  - temperature
  - spechum
  - uReconstructZonal
  - uReconstructMeridional
  - surface_pressure
  - qc
  - qi
  - qr
  - qs
  - qg
  background:
    state variables: [temperature, spechum, uReconstructZonal, uReconstructMeridional, surface_pressure,
                      qc, qi, qr, qs, qg, theta, rho, u, qv, pressure, landmask, xice, snowc, skintemp,
                      ivgtyp, isltyp, snowh, vegfra, u10, v10, lai, smois, tslb]
    filename: "./Data/480km/bg/restart.2018-04-15_00.00.00.nc"
    date: &analysisdate '2018-04-15T00:00:00Z'
  background error:
    covariance model: MPASstatic
    date: *analysisdate
```

Parameters for ctest

Analysis type and time window

Analysis variables

Parameters related to first guess/background

Parameters related to observations

```yaml
observations:
  observers:
  - obs space:
      name: Radiosonde
      obsdatain:
        engine:
          type: H5File
          obsfile: Data/ufo/testinput_tier_1/sondes_obs_2018041500_m.nc4
      obsdataout:
        engine:
          type: H5File
          obsfile: Data/os/obsout_3dvar_sondes.nc4
      simulated variables: [airTemperature, windEastward, windNorthward, specificHumidity]
    obs operator:
      name: VertInterp
      observation alias file: testinput/obsop_name_map.yaml
    obs error:
      covariance model: diagonal
    obs filters:
    - filter: PreQC
      maxvalue: 3
    - filter: Background Check
      threshold: 3
      apply at iterations: 0,1
  - obs space:
      name: Aircraft
```

More details on the JEDI YAML file configuration will be provided in the upcoming talks

This material is based upon work supported by the NSF National Center for Atmospheric Research, which is a major facility operated by the U.S. National Science Foundation under Cooperative Agreement No. 1852977.

Observations (1):
Converting observations to IODA format & HofX Application

Presented by Ivette Hernández Baños based on materials prepared by Junmei Ban

Prediction, Assimilation, and Risk Communication Section
Mesoscale & Microscale Meteorology Laboratory
National Center for Atmospheric Research

MONAN: INPE MPAS-JEDI Training 2024, Cachoeira Paulista, São Paulo, Brazil
August 15-16, 2024

Outline

IODA: Interface for Observational Data Access

1. Observation types in MPAS-JEDI

2. Converting observations to IODA format

3. HofX Application

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

This talk focus on:

- ***y*** → Observations
- ***H*(x)** → calculate model equivalents of the observations; computed through the forward operator

Observation types in MPAS-JEDI

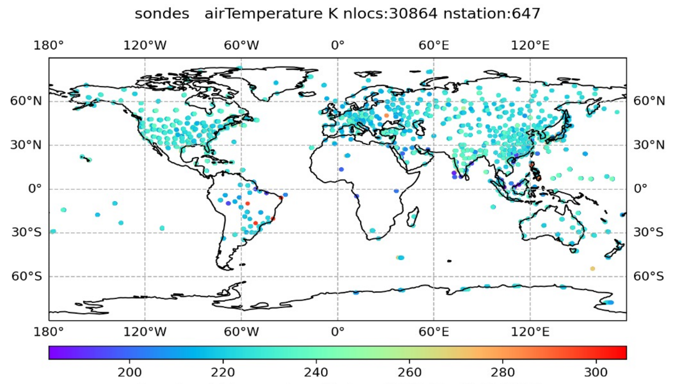
- **Non-Radiances**:
  - Aircraft (U, V, T, spechum)
  - Sondes (U, V, T, spechum)
  - Surface pressure (surface synoptic observations (SYNOP), METAR, ships, drifting buoys and CMAN station reports)
  - atmospheric motion vectors (AMVs) (NCEP prepBURF and BURF files)
  - GNSS radio occultation
    - bending angle
    - atmosphere refractivity

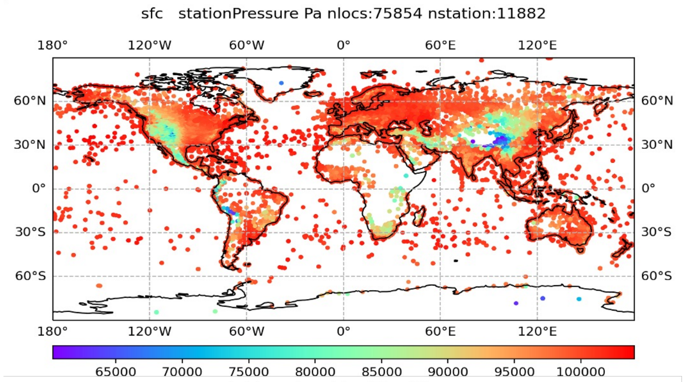
- **Radiances** (using CRTM or RTTOV):
  - AMSU-A (NOAA-15–16, NOAA-18–19, EOS-Aqua, MetOp-A–B)
  - MHS (NOAA-18–19, MetOp-A–B)

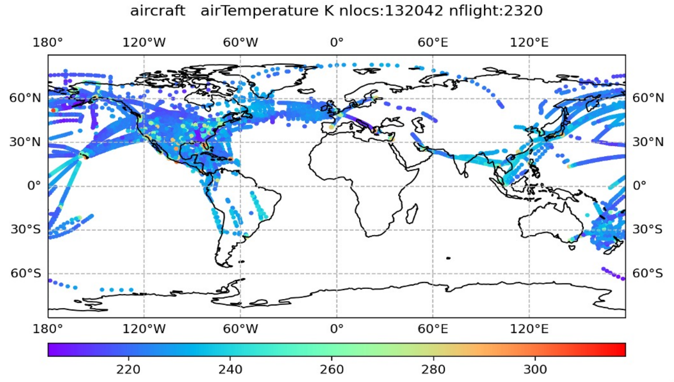
  Testing mode:
    - ATMS (Suomi NPP, NOAA-20–21)
    - IASI (MetOp-A–B)
    - CrIS (Suomi NPP, NOAA-20, JPSS-2)
    - GMI (GPM)

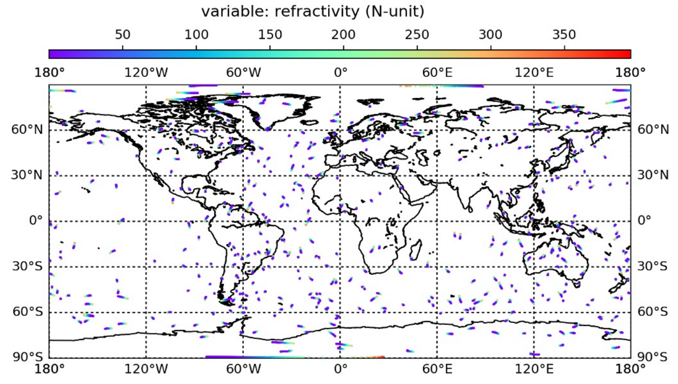
Observation types in MPAS-JEDI

Obs coverage
00Z 15 April 2018







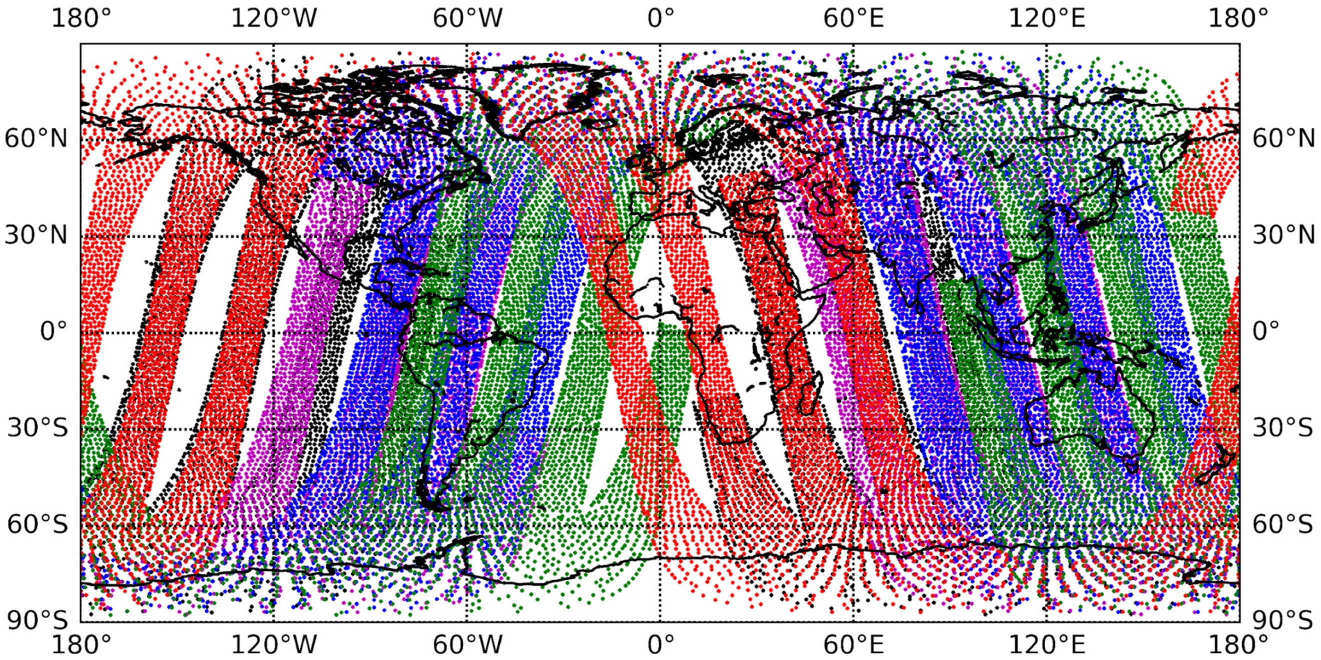


Observation types in MPAS-JEDI

AMSU-A
Obs coverage
12Z 18 April 2018

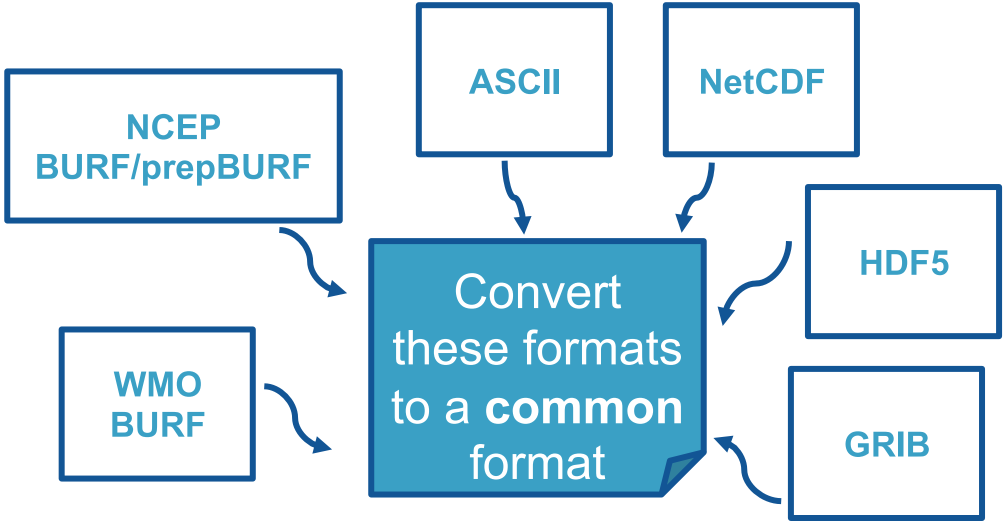
Thinning : 145km

amsua_n19 amsua_n15 amsua_n18 amsua_metop-a amsua_aqua



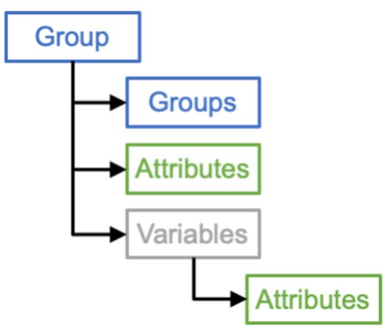
Converting observations to IODA format

What is the IODA format?



HDF data model

Primary classes: Layout policy:



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

Converting observations to IODA format

NCAR obs2ioda converter

1. Source code: git clone https://github.com/NCAR/obs2ioda

   Dependencies:

   - ❑ NCEP BUFR library:

     git clone https://github.com/NOAA-EMC/NCEPLIBS-bufr
   - ❑ Fortran or GNU compilers
2. Set BUFR_LIB in obs2ioda-v2/src/Makefile
3. Make to compile the code

   make → Successful compilation produces the executable: obs2ioda.x

Converting observations to IODA format

BUFR and PREPBUFR format

NCEP operational observation files in BUFR and PREPBUFR format:

- ❑ NCEP real-time data

  http://www.ftp.ncep.noaa.gov/data/nccf/com/gfs/prod

- ❑ NSF NCAR CISL archive

  http://rda.ucar.edu/datasets/ds337.0

  http://rda.ucar.edu/datasets/ds735.0

**If you have an account on Derecho (or Casper) HPC:*

/glade/campaign/collections/rda/data/ds337.0 ⇒ prepBUFR

/glade/campaign/collections/rda/data/ds735.0 ⇒ BURF

Files to look for:

prep48h
└── ${yyyy}/prepbufr.gdas.${yyyy}${mm}${dd}.t${hh}z.nr.48h

satwnd
└── ${yyyy}/satwnd${yyyy }${mm}${dd}.tar.gz

gpsro
└── ${yyyy}/gpsro${yyyy}${mm}${dd}.tar.gz

1bamua
└── ${yyyy}/1bamua.${yyyy}${mm}${dd}.tar.gz

Converting observations to IODA format

BUFR and prepBUFR to IODA

Usage: obs2ioda.x [-i input_dir] [-o output_dir] [bufr_filename(s)_to_convert]

**input_dir and output_dir: optional augment*

Output: IODA v2 format

For aircraft/satwind/satwnd/sfc/sondes, need to run upgrade executable:

“char” to “string” for station_id and variable_name

Usage: ./ioda-upgrade-v1-to-v2.x inputFile outputFile

For all the observation types, need to run upgrade executable:

Usage: ./ioda-upgrade-v2-to-v3.x inputFile outputFile

Example output files:

aircraft_obs_YYYYMMDDHH.h5

satwind_obs_YYYYMMDDHH.h5

sfc_obs_YYYYMMDDHH.h5

sondes_obs_YYYYMMDDHH.h5

amsua_aqua_obs_2018041500.h5

gnssrobndropp1d_obs_2018041500.h5

…

Converting observations to IODA format

BUFR and prepBUFR to IODA

Usage: obs2ioda.x [-i input_dir] [-o output_dir] [bufr_filename(s)_to_convert]

**input_dir and output_dir: optional augment*

About observation errors:

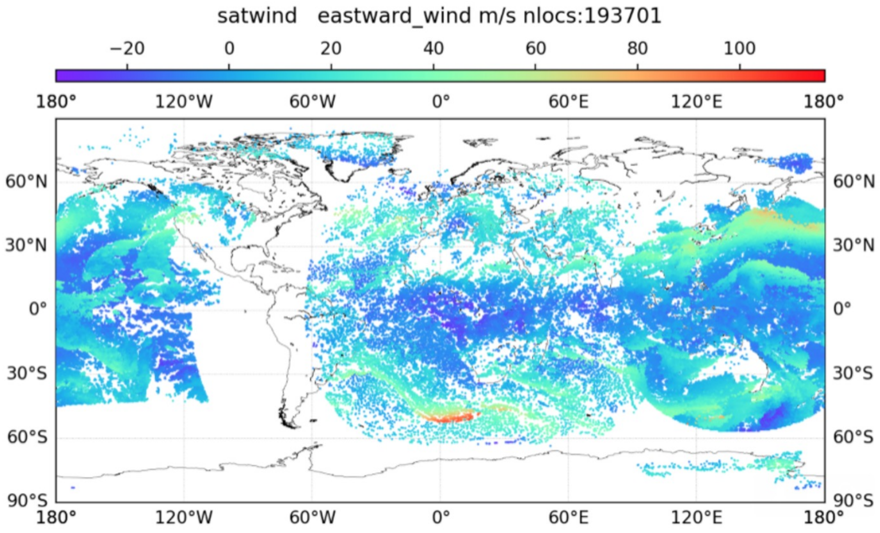
- Observation errors of conventional data are either extracted from the input prepBUFR or from an external error table (if obs_errtable exists in the working directory)
- Observation errors of AMSU-A/MHS radiances are coded in **define_mod.f90**
    - Observation errors of **satwnd-decoded AMVs** are from an external error table (**obs_errtable**)

About quality controls (QC):

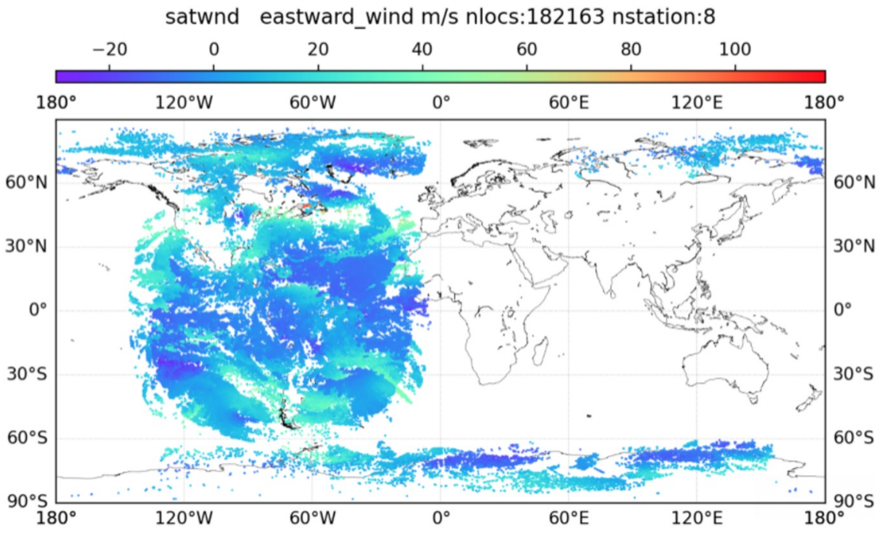
* Subroutine **filter_obs_conv** applies some additional QC for conventional observations as in **GSI's read_prepbufr.f90** for the global model and can be deactivated through ``-noqc`` command-line option

Converting observations to IODA format

Satellite wind converted from prepBUFR and BUFR are complementary and should be assimilated **together**



Satellite wind converted from prepBUFR file



Satellite wind converted from BUFR file

Other AMVs are from PREPBUFR files

Includes **GOES-16/GOES-17, AVHRR (METOP/NOAA) and VIIRS (NPP/NOAA) polar AMVs,** also LEOGEO AMVs

Converting observations to IODA format

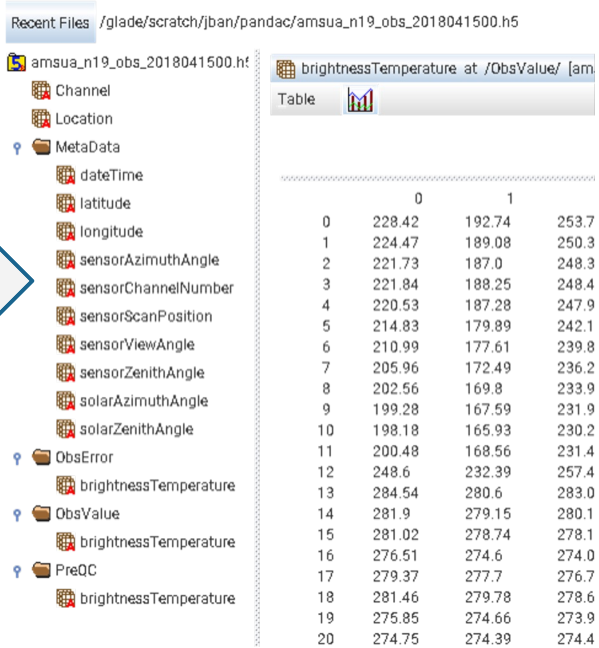
Tools to check IODA observations

- ncdump, h5dump, hdfview, Python h5py, …

h5dump –n filename

```
HDF5 "obs_iodav3/obsiodav3_221216/newobs_2018/raw_obs/20180
FILE_CONTENTS {
 group      /
 dataset    /Channel
 dataset    /Location
 group      /MetaData
 dataset    /MetaData/dateTime
 dataset    /MetaData/latitude
 dataset    /MetaData/longitude
 dataset    /MetaData/sensorAzimuthAngle
 dataset    /MetaData/sensorChannelNumber
 dataset    /MetaData/sensorScanPosition
 dataset    /MetaData/sensorViewAngle
 dataset    /MetaData/sensorZenithAngle
 dataset    /MetaData/solarAzimuthAngle
 dataset    /MetaData/solarZenithAngle
 group      /ObsError
 dataset    /ObsError/brightnessTemperature
 group      /ObsValue
 dataset    /ObsValue/brightnessTemperature
 group      /PreQC
 dataset    /PreQC/brightnessTemperature
 }
}
```

hdfview



	0	1	
0	228.42	192.74	253.7
1	224.47	189.08	250.3
2	221.73	187.0	248.3
3	221.84	188.25	248.4
4	220.53	187.28	247.9
5	214.83	179.89	242.1
6	210.99	177.61	239.8
7	205.96	172.49	236.2
8	202.56	169.8	233.9
9	199.28	167.59	231.9
10	198.18	165.93	230.2
11	200.48	168.56	231.4
12	248.6	232.39	257.4
13	284.54	280.6	283.0
14	281.9	279.15	280.1
15	281.02	278.74	278.1
16	276.51	274.6	274.0
17	279.37	277.7	276.7
18	281.46	279.78	278.6
19	275.85	274.66	273.9
20	274.75	274.39	274.4

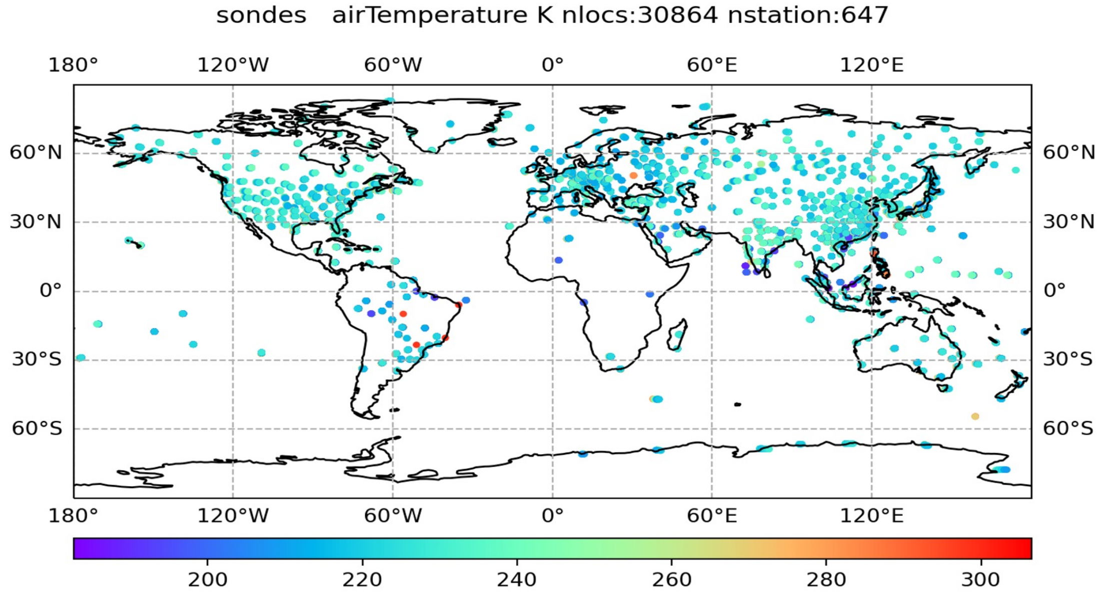
Converting observations to IODA format

Plotting observation locations

Under **graphics**:

├── standalone

  ├── plot_obs_loc_tut.py



HofX Application

mpasjedi_hofx3d.x ./hofx3d.yaml ./mpasjedi_hofx3d.log

executable	YAML file	output log file

```
window begin: 2018-04-14T21:00:00Z
window length: PT6H
geometry:
  nml_file: ./namelist.atmosphere_240km
  streams_file: ./streams.atmosphere_240km
  deallocate non-da fields: true
state:
  state variables: [spechum,surface_pressure,temperature,uReconstructMer
typ,isltyp,snowh,vegfra,u10,v10,lai,smois,tslb,pressure_p]
  filename: ./bg/bg.2018-04-15_00.00.00.nc
  date: 2018-04-15T00:00:00Z
observations:
```

window begin:

 # datetime in ISO format

window length:

 # duration in ISO format

geometry:

 # geometry of the model

state:

 # model state used for computing H(x)

HofX Application

mpasjedi_hofx3d.x ./hofx3d.yaml ./mpasjedi_hofx3d.log

executable

YAML file

output log file

```yaml
observations:
  observers:
  - obs space:
      name: Aircraft
      obsdatain:
        engine:
          type: H5File
          obsfile: ./aircraft_obs_2018041500.h5
      obsdataout:
        engine:
          type: H5File
          obsfile: ./obsout_hofx_aircraft.h5
      simulated variables: [airTemperature, windEastward, windNorthward, specificHumidity]
    obs operator:
      name: VertInterp
```

observation:

observers:

-obs space:

obs operator:

-obs space:

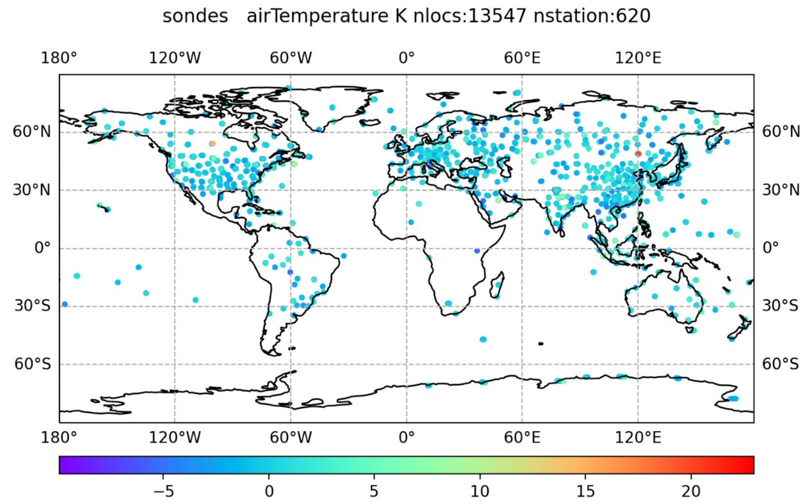
obs operator:

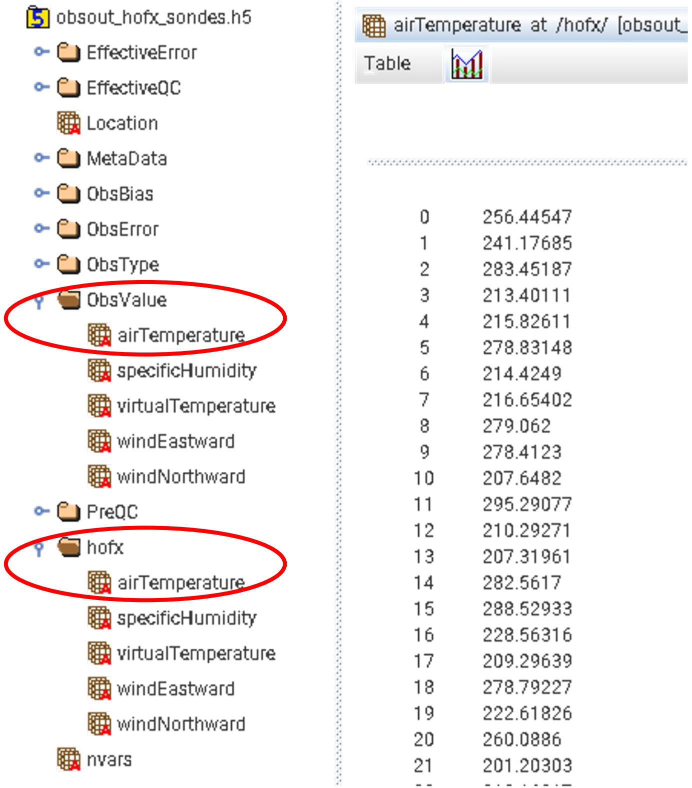
INDENTATION!!!

HofX Application

Output: obsout_hofx_sondes.h5

Observations departure: O-B





To learn more

- IODA: Interface for Observation Data Access
  https://github.com/JCSDA/ioda
    - Other converters: https://github.com/JCSDA/ioda-bundle
- UFO: Unified Forward Operator
  https://github.com/JCSDA/ufo

MPAS-JEDI 3D/4DEnVar

Presented by Jake Liu

Based on the materials prepared by I-Han Chen

NCAR | MESOSCALE & MICROSCALE METEOROLOGY

Overview

1. **Variational Cost Function**
2. Ensemble Error Covariance Matrix
3. Overview of 3DEnVar
4. Setting up a .yaml file for 3DEnVar
5. Overview of 4DEnVar
6. Setting up a .yaml file for 4DEnVar

The Problem

We want to find the **analysis state** (x) that minimizing a cost function with an optimal fit to the background and observations.

$$J(x) = \boxed{\frac{1}{2}(x - x_b)^T \mathbf{B}^{-1}(x - x_b)} + \boxed{\frac{1}{2}(h(x) - y)^T \mathbf{R}^{-1}(h(x) - y)}$$

Distance to background **Distance to observations**

Incremental Cost Function in JEDI

Liu et al. (2022)

Full-form $J(x) = \frac{1}{2}(x - x_b)^T \boldsymbol{B}^{-1}(x - x_b) + \frac{1}{2}(h(x) - y)^T \boldsymbol{R}^{-1}(h(x) - y)$

Incremental-form $J(\delta x) = \frac{1}{2}(\delta x - \delta x_g)^T \boldsymbol{B}^{-1}(\delta x - \delta x_g) + \frac{1}{2}(\boldsymbol{H}\delta x - d)^T \boldsymbol{R}^{-1}(\boldsymbol{H}\delta x - d)$

$\boldsymbol{\delta x = x - x_g}$ $\delta x_g = x_b - x_g$ $\boldsymbol{d = y - h(x_g)}$

The minimization deals with increments to a known reference state

- Cost function minimizes δx=x− x_g instead of the full state (x)
- Start from $x_g = x_b$ and $\delta x_g = 0$
- **After minimization->** $x_a = x_g + \delta x$

Appropriately assign B and R is critical

We want to find the analysis state (x) that minimizing a cost function with **an optimal fit** to the background and observations.

Distance to background **Distance to observations**

$$J(\delta x) = \boxed{\frac{1}{2}\left(\delta x - \delta x_g\right)^T B^{-1}\left(\delta x - \delta x_g\right)} + \boxed{\frac{1}{2}(H\delta x - d)^T R^{-1}(H\delta x - d)}$$

The weighting between the two components is determined by B (background error) and R (observation error).

- A larger B means background is less accurate -> x will get closer to observation
- A larger R means observation is less accurate -> x will get closer to background

Two types of background error covariance (B)

$$J(\delta x) = \frac{1}{2}\left(\delta x - \delta x_g\right)^T B^{-1}\left(\delta x - \delta x_g\right) + \frac{1}{2}(H\delta x - d)^T R^{-1}(H\delta x - d)$$

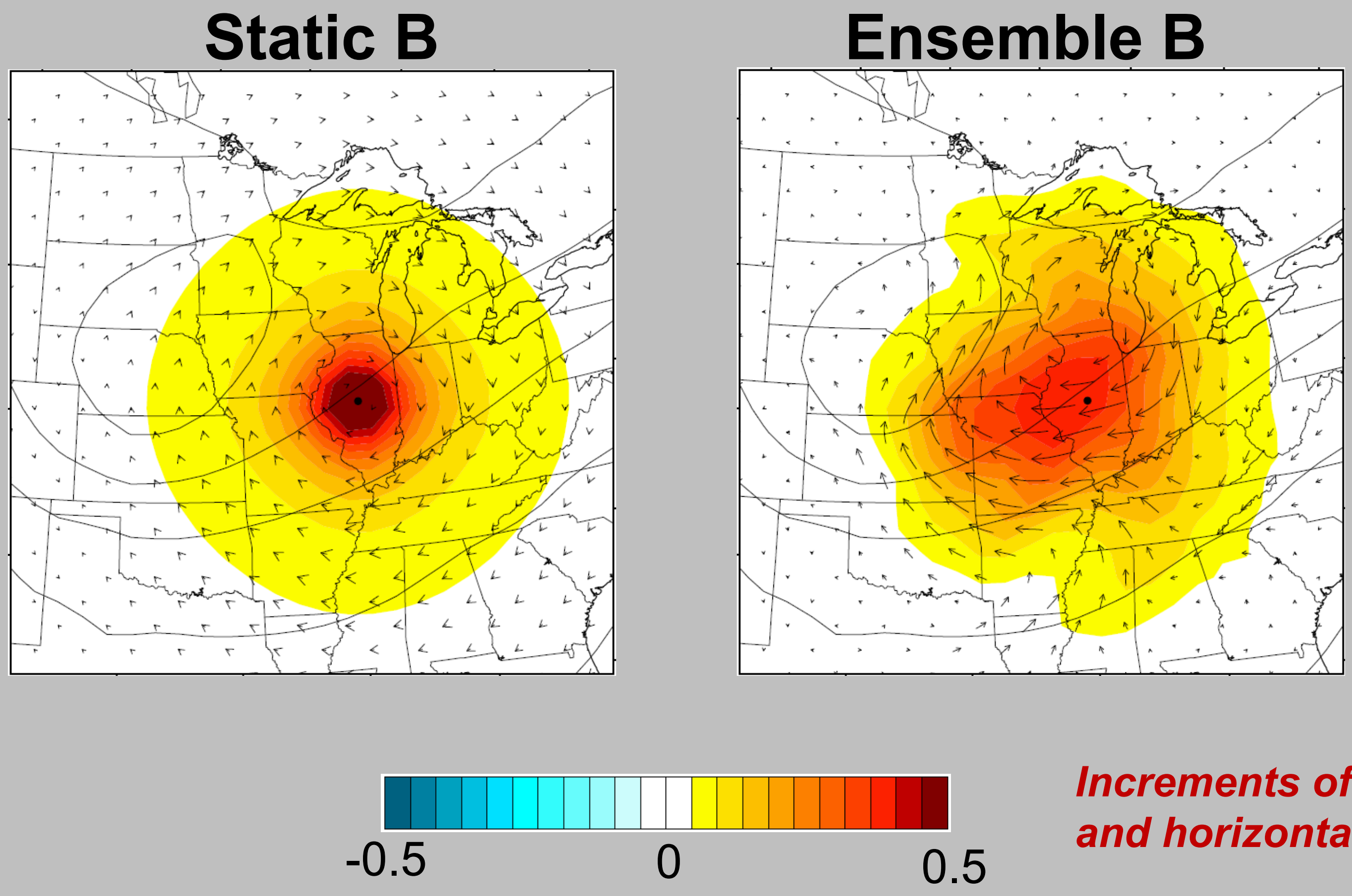
1. **Static B**

   -> from statistic, does not vary with time

2. **Ensemble B**

   -> flow-dependent, reflect the background error in different time

Example to show the B effect (Single observation tests)



Increments of temperature (shaded) and horizontal winds (vector)

Ensemble ***B***:

- Errors of the day are sampled
- flow-dependent update

Overview

1. Variational Cost Function
2. **Ensemble Error Covariance Matrix**
3. Overview of 3DEnVar
4. Setting up a .yaml file for 3DEnVar
5. Overview of 4DEnVar
6. Setting up a .yaml file for 4DEnVar

Derive B matrix from an ensemble of forecasts

ensemble size

State variable of ensemble mean

$$\boldsymbol{B_e} = \frac{1}{n-1}\sum_{i=1}^{\boldsymbol{n}}(\boldsymbol{x_i} - \overline{\boldsymbol{x}})(x_i - \bar{x})^T$$

State variable of each ensemble member

$$\boldsymbol{B_e} = \frac{1}{n-1}\sum_{i=1}^{\boldsymbol{n}}(\delta x_{\boldsymbol{i}})(\delta x_{\boldsymbol{i}})^T$$

ensemble perturbation

- The ensemble mean provides an estimation of the truth
- The perturbations from the mean estimate the uncertainty, which is used to model background-error covariance matrix.

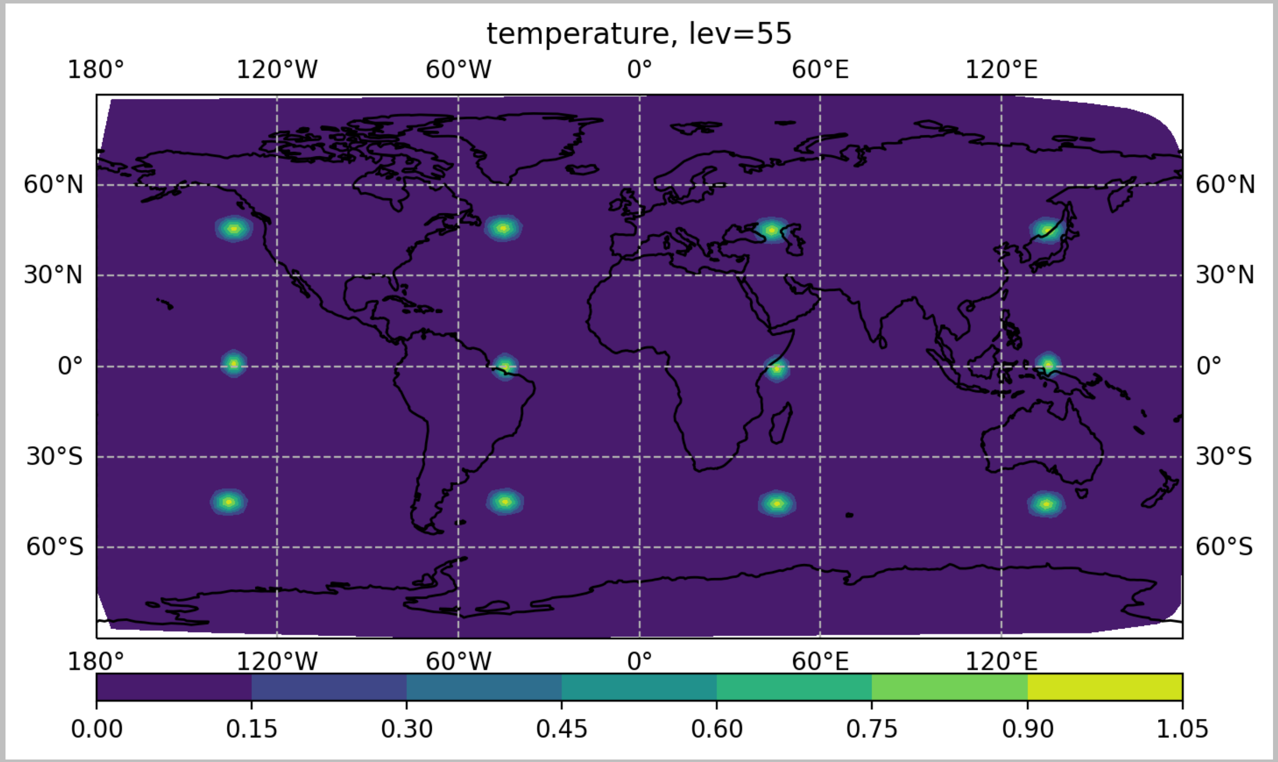
Localization of the B matix

Because we do not have a complete estimate of $\boldsymbol{B}$ (e.g., limited ensemble size) we need to use localization

Basic idea: observations should only influence an area nearby the observation

$$\boldsymbol{B} = \boldsymbol{L} \circ \boldsymbol{B_e}$$

Small localization



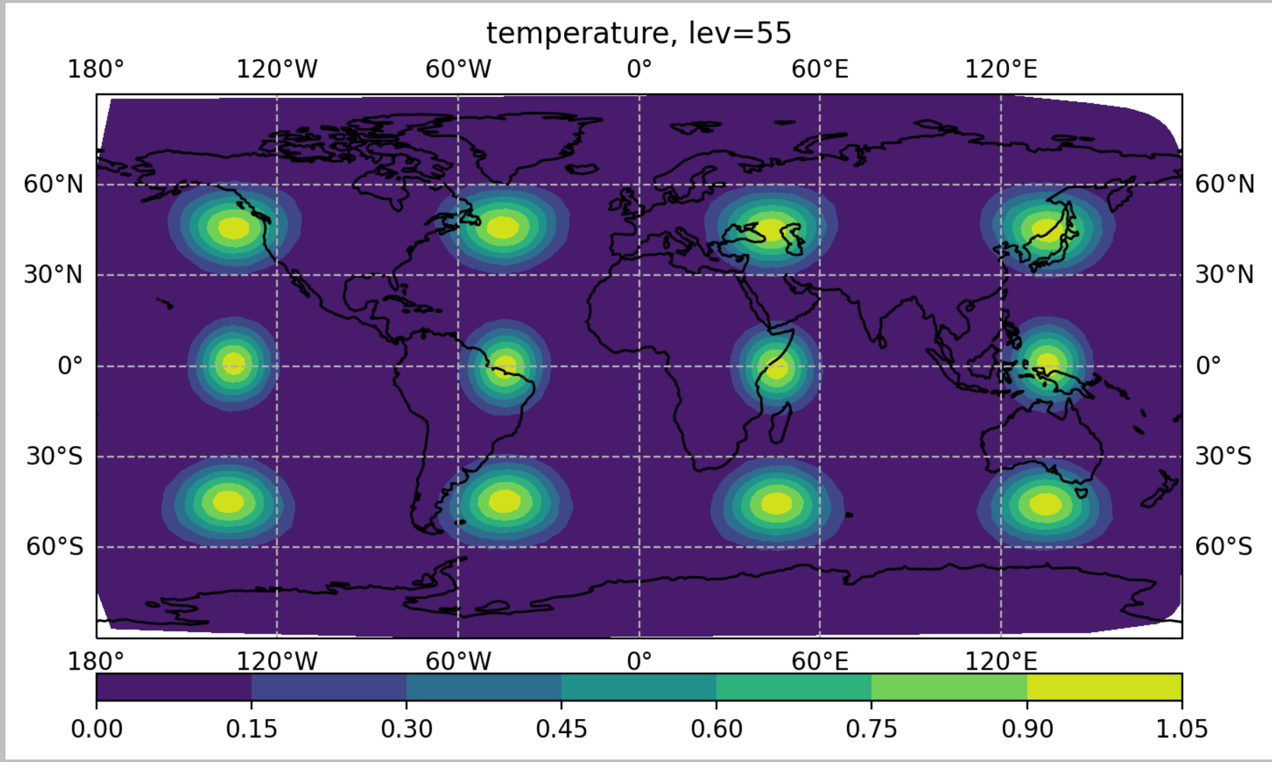
Localization of the B matix

Because we do not have a complete estimate of $\boldsymbol{B}$ (e.g., limited ensemble size) we need to use localization

Basic idea: observations should only influence an area nearby the observation

$$\boldsymbol{B} = \boldsymbol{L} \circ \boldsymbol{B}_e$$

large localization



Benefits of using an ensemble to estimate *B*

- Simple to implement

- Provides a flow-dependent estimate of the errors and uncertainties
  - Depends on the quality of the ensemble

- Incorporates ensemble estimate of background errors within the variational update
  - **Still updates a deterministic forecast**

EnVar uses a pure ensemble B to updates a deterministic forecast

In hybrid methods, B can be a weighting sum between static B ($\boldsymbol{B_s}$) and ensemble B ($\boldsymbol{B_e}$).

$$\boldsymbol{B} = \beta_s \boldsymbol{B_s} + \beta_e \boldsymbol{B_e}$$
$$\beta_s + \beta_e = 1$$

=1

pure ensemble B

Overview

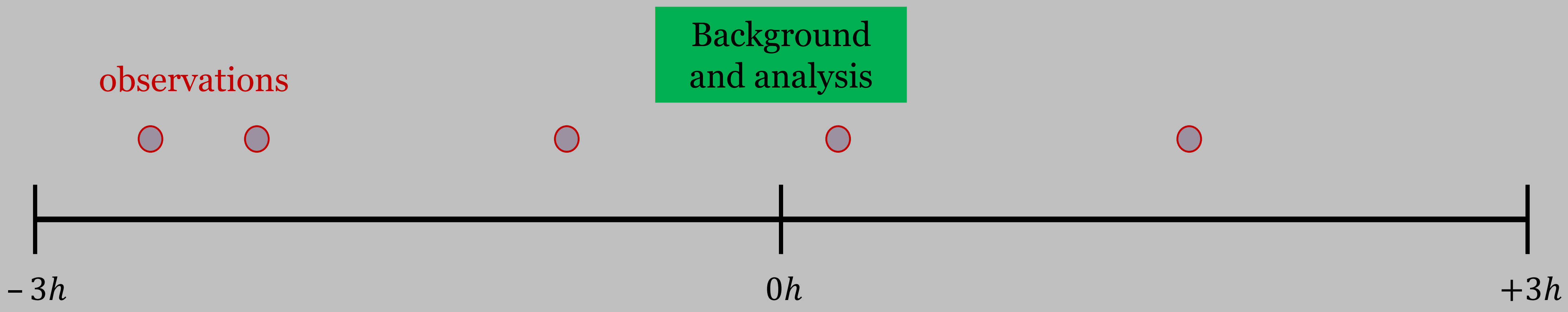
1. Variational Cost Function
2. Ensemble Error Covariance Matrix
3. **Overview of 3DEnVar**
4. Setting up a .yaml file for 3DEnVar
5. Overview of 4DEnVar
6. Setting up a .yaml file for 4DEnVar

3DEnVar

$$J(x) = \frac{1}{2}(x - x_b)^T \boldsymbol{B}^{-\mathbf{1}}(x - x_b) + \frac{1}{2}(h(x) - y)^T \boldsymbol{R}^{-\mathbf{1}}(h(x) - y)$$

- We assume that all observations $\boldsymbol{y_o}$ are valid at the same time.
- Usually valid at the center of the window (i.e. at the same time as $\boldsymbol{x}$ and $\boldsymbol{x_b}$)

3DEnVar using a 6h assimilation window



- All observations in 3DEnVar are assumed to be valid at the same time as the background

Overview

1. Variational Cost Function
2. Ensemble Error Covariance Matrix
3. Overview of 3DEnVar
4. **Setting up a .yaml file for 3DEnVar**
5. Overview of 4DEnVar
6. Setting up a .yaml file for 4DEnVar

Configure the analysis time for 3DEnvar

```yaml
_member config: &memberConfig
  date: &analysisdate '2018-04-15T00:00:00Z'
  state variables: &incvars
  - temperature
  - spechum
  - uReconstructZonal
  - uReconstructMeridional
  - surface_pressure
  stream name: ensemble
cost function:
  cost type: 3D-Var
  window begin: '2018-04-14T21:00:00Z'
  window length: PT6H
  geometry:
    nml_file: "./Data/480km/namelist.atmosphere_2018041500"
    streams_file: "./Data/480km/streams.atmosphere"
    deallocate non-da fields: true
  analysis variables: *incvars
  background:
    state variables: [temperature, spechum, uReconstructZonal, uReconstructMeridional, surface_pressure,
                      theta, rho, u, qv, pressure, landmask, xice, snowc, skintemp, ivgtyp, isltyp,
                      snowh, vegfra, u10, v10, lai, smois, tslb, pressure_p]
    filename: "./Data/480km/bg/restart.2018-04-15_00.00.00.nc"
    date: *analysisdate
```

analysis time (center of window)

Start of assimilation window
length of assimilation window

First guess (should be at analysis time)

Configure the ensemble B

```yaml
background error:
  covariance model: ensemble
  localization:
    localization method: SABER
    saber central block:
      saber block name: BUMP_NICAS
      active variables: *incvars
      read:
        io:
          files prefix: Data/bump/mpas_parametersbump_loc
        drivers:
          multivariate strategy: duplicated
          read local nicas: true
  members:
  - filename: Data/480km/bg/ensemble/mem01/x1.2562.init.2018-04-15_00.00.00.nc
    <<: *memberConfig
  - filename: Data/480km/bg/ensemble/mem02/x1.2562.init.2018-04-15_00.00.00.nc
    <<: *memberConfig
  - filename: Data/480km/bg/ensemble/mem03/x1.2562.init.2018-04-15_00.00.00.nc
    <<: *memberConfig
  - filename: Data/480km/bg/ensemble/mem04/x1.2562.init.2018-04-15_00.00.00.nc
    <<: *memberConfig
  - filename: Data/480km/bg/ensemble/mem05/x1.2562.init.2018-04-15_00.00.00.nc
    <<: *memberConfig
```

set ensemble *B* for 3DEnVar

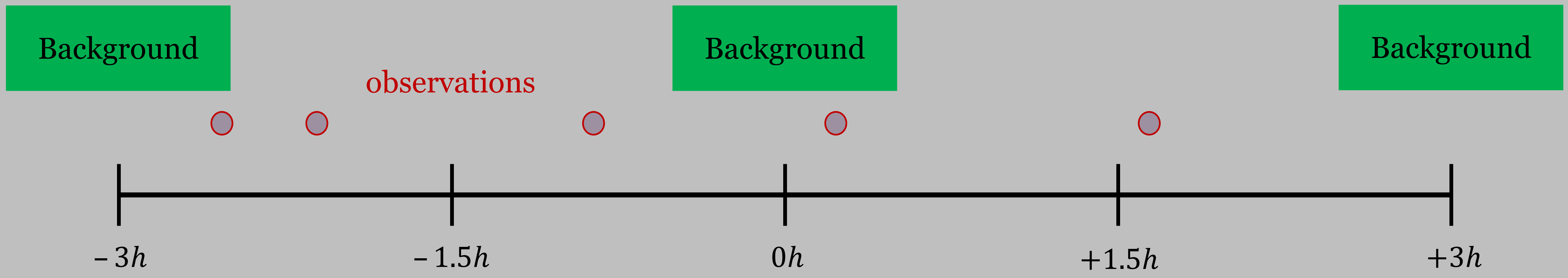
Specifying members used to compute ensemble B

Overview

1. Variational Cost Function
2. Ensemble Error Covariance Matrix
3. Overview of 3DEnVar
4. Setting up a .yaml file for 3DEnVar
5. **Overview of 4DEnVar**
6. Setting up a .yaml file for 4DEnVar

4DEnVar

$$J(x) = \frac{1}{2}(x - x_b)^T B^{-1}(x - x_b) + \frac{1}{2}\sum_{k=1}^{K}(Hx_k - y_k)^T {R_k}^{-1}(Hx_k - y_k)$$

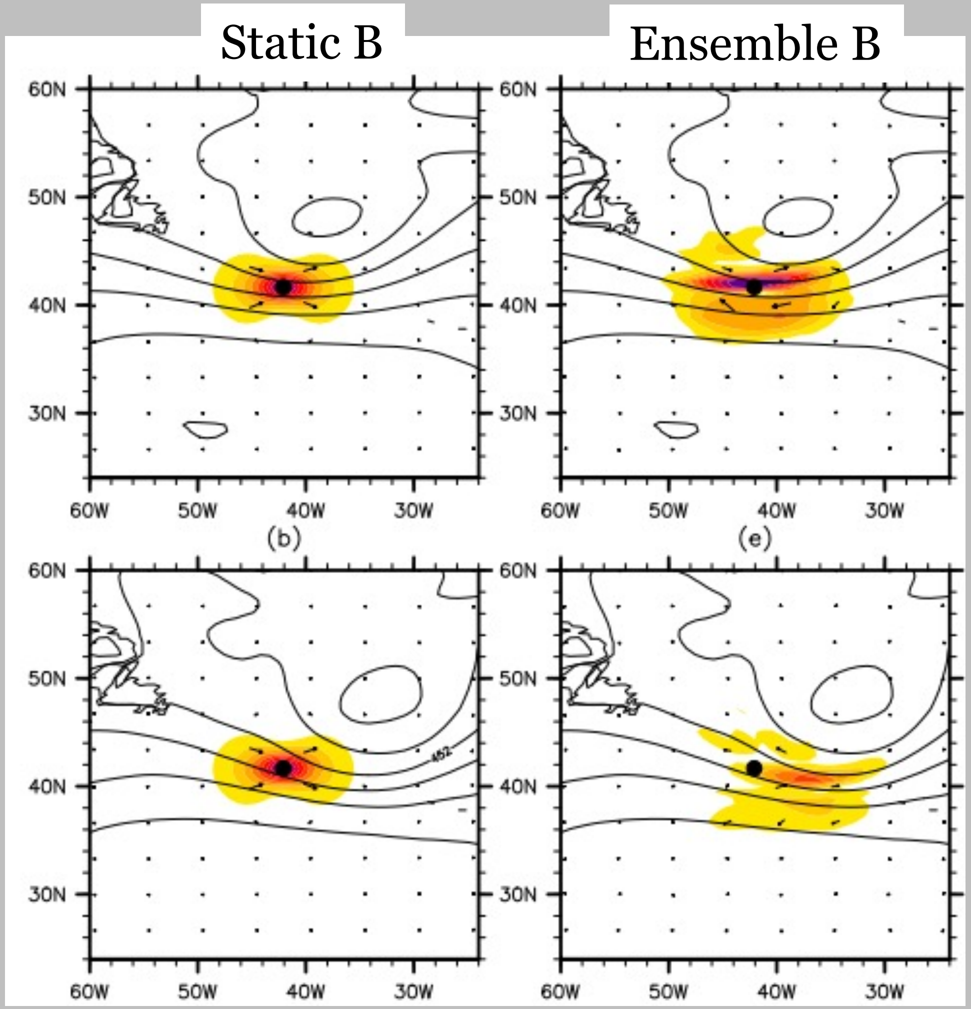


- All observations in 4DEnVar are binned within a smaller subwindow and innovations $(Hx - y_o)$ are calculated relative to background valid at that time.
- Ensemble needed at the center of each subwindow (***K*** ensemble required).

The 4D ensemble *B* is used to propagate the innovation

Start of window

end of window



Lorenc et al. (2015)

Overview

1. Variational Cost Function
2. Ensemble Error Covariance Matrix
3. Overview of 3DEnVar
4. Setting up a .yaml file for 3DEnVar
5. Overview of 4DEnVar
6. **Setting up a .yaml file for 4DEnVar**

Configure the analysis times for 4DEnvar

```yaml
_member config 1: &memberConfig1
  date: &date1 '2018-04-14T21:00:00Z'
  state variables: &incvars
  - temperature
  - spechum
  - uReconstructZonal
  - uReconstructMeridional
  - surface_pressure
  stream name: ensemble
_member config 2: &memberConfig2
  <<: *memberConfig1
  date: &date2 '2018-04-15T00:00:00Z'
_member config 3: &memberConfig3
  <<: *memberConfig1
  date: &date3 '2018-04-15T03:00:00Z'
cost function:
  cost type: 4D-Ens-Var
  window begin: '2018-04-14T21:00:00Z'
  window length: PT6H
  subwindow: PT3H
```

subwindow1

subwindow2

subwindow3

Background needed for each subwindow

```
cost function:
  cost type: 4D-Ens-Var
  window begin: '2018-04-14T21:00:00Z'
  window length: PT6H
  subwindow: PT3H
  geometry:
    nml_file: "./Data/480km/namelist.atmosphere_2018041500"
    streams_file: "./Data/480km/streams.atmosphere"
  analysis variables: *incvars
  background:
    states:
    - state variables: &stvars
                        [temperature, spechum, uReconstructZonal, uReconstructMeridional, surface_pressure,
                        theta, rho, u, qv, pressure, landmask, xice, snowc, skintemp, ivgtyp, isltyp,
                        snowh, vegfra, u10, v10, lai, smois, tslb, pressure_p]
      filename: "./Data/480km/bg/restart.2018-04-14_21.00.00.nc"
      date: *date1
    - state variables: *stvars
      filename: "./Data/480km/bg/restart.2018-04-15_00.00.00.nc"
      date: *date2
    - state variables: *stvars
      filename: "./Data/480km/bg/restart.2018-04-15_03.00.00.nc"
      date: *date3
```

bg (subwindow 1)

bg (subwindow 2)

bg (subwindow 3)

Configure the ensemble B

```
background error:
  covariance model: ensemble
  localization:
    localization method: SABER
    saber central block:
      saber block name: BUMP_NICAS
      active variables: *incvars
      read:
        io:
          files prefix: Data/bump/mpas_parametersbump_loc
        drivers:
          multivariate strategy: duplicated
          read local nicas: true
```

set ensemble *B* for 4DEnVar

Member file needed for each subwindow

```yaml
members:
- states:
  - filename: Data/480km/bg/ensemble/mem01/x1.2562.init.2018-04-14_21.00.00.nc
    <<: *memberConfig1
  - filename: Data/480km/bg/ensemble/mem01/x1.2562.init.2018-04-15_00.00.00.nc
    <<: *memberConfig2
  - filename: Data/480km/bg/ensemble/mem01/x1.2562.init.2018-04-15_03.00.00.nc
    <<: *memberConfig3
- states:
  - filename: Data/480km/bg/ensemble/mem02/x1.2562.init.2018-04-14_21.00.00.nc
    <<: *memberConfig1
  - filename: Data/480km/bg/ensemble/mem02/x1.2562.init.2018-04-15_00.00.00.nc
    <<: *memberConfig2
  - filename: Data/480km/bg/ensemble/mem02/x1.2562.init.2018-04-15_03.00.00.nc
    <<: *memberConfig3
- states:
  - filename: Data/480km/bg/ensemble/mem03/x1.2562.init.2018-04-14_21.00.00.nc
    <<: *memberConfig1
  - filename: Data/480km/bg/ensemble/mem03/x1.2562.init.2018-04-15_00.00.00.nc
    <<: *memberConfig2
  - filename: Data/480km/bg/ensemble/mem03/x1.2562.init.2018-04-15_03.00.00.nc
    <<: *memberConfig3
- states:
  - filename: Data/480km/bg/ensemble/mem04/x1.2562.init.2018-04-14_21.00.00.nc
    <<: *memberConfig1
  - filename: Data/480km/bg/ensemble/mem04/x1.2562.init.2018-04-15_00.00.00.nc
    <<: *memberConfig2
  - filename: Data/480km/bg/ensemble/mem04/x1.2562.init.2018-04-15_03.00.00.nc
    <<: *memberConfig3
- states:
  - filename: Data/480km/bg/ensemble/mem05/x1.2562.init.2018-04-14_21.00.00.nc
    <<: *memberConfig1
  - filename: Data/480km/bg/ensemble/mem05/x1.2562.init.2018-04-15_00.00.00.nc
    <<: *memberConfig2
  - filename: Data/480km/bg/ensemble/mem05/x1.2562.init.2018-04-15_03.00.00.nc
    <<: *memberConfig3
```

References

- Liu, Z., and Coauthors, 2022: Data assimilation for the Model for Prediction Across Scales - Atmosphere with the Joint Effort for Data assimilation Integration (JEDI-MPAS 1.0.0): EnVar implementation and evaluation. *Geosci. Model Dev.*, **15**, 7859–7878, https://doi.org/10.5194/gmd-15-7859-2022.

- Lorenc, A. C., N. E. Bowler, A. M. Clayton, S. R. Pring, and D. Fairbairn, 2015: Comparison of hybrid-4DEnVar and hybrid-4DVar data assimilation methods for global NWP. *Mon. Weather Rev.*, **143**, 212–229, https://doi.org/10.1175/MWR-D-14-00195.1.

This material is based upon work supported by the NSF National Center for Atmospheric Research, which is a major facility operated by the U.S. National Science Foundation under Cooperative Agreement No. 1852977.

Observations (2):
Assimilating conventional observations in MPAS-JEDI

Presented by Ivette Hernández Baños based on materials prepared by Junmei Ban

Prediction, Assimilation, and Risk Communication Section
Mesoscale & Microscale Meteorology Laboratory
National Center for Atmospheric Research

MONAN: INPE MPAS-JEDI Training 2024, Cachoeira Paulista, São Paulo, Brazil
August 15-16, 2024

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

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

Hybrid **B**

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

$\mathbf{B}_s$: static B; $\mathbf{B}_e$: 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 obs2ioda 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

Setting up a yaml file

(focus on observations)

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

```
observations:
  observers:
  - obs space:
      name: Aircraft
      obsdatain:
        engine:
          type: H5File
          obsfile: ./aircraft_obs_2018041500.h5
      obsdataout:
        engine:
          type: H5File
          obsfile: ./obsout_da_aircraft.h5
      simulated variables: [airTemperature, win
    obs error:          dEastward, windNorthward, specificHumidity]
      covariance model: diagonal
    obs operator:
      name: VertInterp
    obs filters:
    - filter: PreQC
      maxvalue: 3
    - filter: Background Check
      threshold: 3.0
```

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

PreQC is assigned from **obs2ioda-v2** converter in subroutine filter_obs_conv (as in GSI's read_prepbufr.f90)

Assimilating non-radiance observations

Setting up a yaml file

(focus on observations)

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

```
    obs error:
      covariance model: diagonal
    obs operator:
      name: VertInterp
      observation alias file: obsop_name_map.yaml
    obs filters:
    - filter: PreQC
      maxvalue: 3
    # Assign the initial observation error, based on height/pressure
    - filter: Perform Action
      filter variables:
      - name: windEastward
      - name: windNorthward
      action:
        name: assign error
        error function:
          name: ObsFunction/ObsErrorModelStepwiseLinear
          options:
            xvar:
              name: MetaData/pressure
            xvals: [100000, 95000, 80000, 65000, 60000, 55000, 50000, 45000, 40000, 35000, 30000, 25000, 20000, 15000, 10000]
            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]
    - filter: Bounds Check
      filter variables:
      - name: windEastward
      - name: windNorthward
      test variables:
      - name: ObsErrorData/windEastward
      - name: ObsErrorData/windNorthward
      minvalue: 0.0
      maxvalue: 200.0
    - filter: Gaussian Thinning
      horizontal_mesh: 145.0
    - filter: Background Check
      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

Setting up a yaml file

(focus on observations)

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

```yaml
    obs operator:
      name: GnssroRefNCEP
      obs options:
        use_compress: 0
    obs filters:
    - filter: Domain Check
      where:
      - variable:
          name: MetaData/height
        minvalue: 0.0
        maxvalue: 30000.0
      - variable:
          name: MetaData/earthRadiusCurvature
        minvalue: 6250000.0
        maxvalue: 6450000.0
      - variable:
          name: MetaData/geoidUndulation
        minvalue: -200.0
        maxvalue: 200.0
    - filter: ROobserror
      variable: refractivity
      errmodel: NCEP
      apply at iterations: 0,1,2
    - filter: Background Check
      threshold: 3.0
      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.

ROobserror (errmodel: NCEP): RO specific filter

Assimilating non-radiance observations

Setting up a yaml file

(focus on observations)

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

Surface pressure

```
obs operator:
  name: SfcPCorrected
  da_psfc_scheme: UKMO   # or WRFDA
linear obs operator:
  name: Identity
  observation alias file: obsop_name_map.yaml
obs filters:
- filter: PreQC
  maxvalue: 3
- filter: Difference Check
  reference: MetaData/stationElevation
  value: GeoVaLs/surface_altitude
  threshold: 200.0
- filter: Background Check
  threshold: 3.0
  apply at iterations: 0,1
```

SfcPCorrected 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

Radiance

GNSS RO

- Vertical Interpolation
- Atmosphere Vertical Layer Interpolation
- Averaging Kernel Operator
- Community Radiative Transfer Model (CRTM)
- RTTOV
- Aerosol Optical Depth (AODCRTM)
- Aerosol Optical Depth (AOD) for dust (Met Office)
- 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)
- Ground Based GNSS observation operator (Met Office)
- Identity observation operator
- Product observation operator
- In situ particulate matter (PM) operator
- Radar Radial Velocity
- Scatterometer neutral wind (Met Office)
- SfcPCorrected
- 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

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
- Domain Check Filter
- BlackList Filter
- RejectList Filter
- AcceptList Filter
- Perform Action Filter
- Thinning Filter
- Gaussian Thinning Filter
- Temporal Thinning Filter
- Poisson Disk 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

Background

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

Additional QC Filter Options

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

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

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

executable

YAML file

output log file

mpasjedi_3denvar.log:

OOPS_STATS Run end - Runtime: 134.43 sec, Memory: total: 23.00 Gb, per task: min = 594.06 Mb, max = 1101.14 Mb

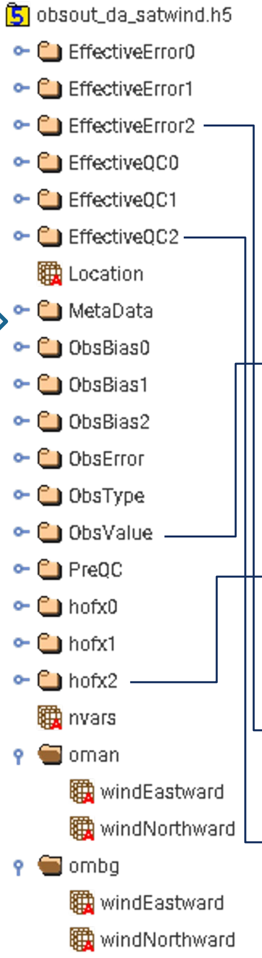
Run: Finishing oops::Variational<MPAS, UFO and IODA observations> with status = 0
OOPS Ending 2023-09-15 09:24:26 (UTC-0600)

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

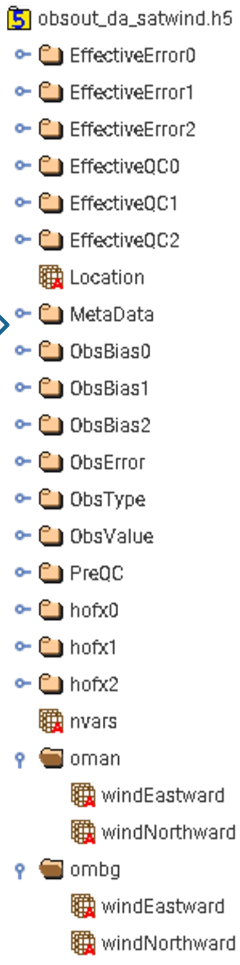
Selection of common group names and meanings

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
```

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

Diagnostics

Check output log file:

QC counts for surface pressure

```
QC SfcPCorrected stationPressure: 66147 missing values.
QC SfcPCorrected stationPressure: 549 rejected by pre QC.
QC SfcPCorrected stationPressure: 533 rejected by first-guess check.
QC SfcPCorrected stationPressure: 13122 rejected by difference check.
QC SfcPCorrected stationPressure: 54233 passed out of 134584 observations.
```

QC counts for satwnd (U component)

```
QC Satwnd windEastward: 413874 rejected by pre QC.
QC Satwnd windEastward: 4282 out of bounds.
QC Satwnd windEastward: 170237 removed by thinning.
QC Satwnd windEastward: 176 rejected by first-guess check.
QC Satwnd windEastward: 7468 passed out of 596037 observations.
```

Cost function and norm reduction

```
Quadratic cost function: J   ( 1) = 507631.5061956716
Quadratic cost function: Jb  ( 1) = 6.828370375967046
Quadratic cost function: JoJc( 1) = 507624.6778252956
Quadratic cost function: J   ( 2) = 495129.1315379007
Quadratic cost function: Jb  ( 2) = 39.53971478609463
Quadratic cost function: JoJc( 2) = 495089.5918231146
Quadratic cost function: J   ( 3) = 478221.3655824636
......

Norm reduction ( 1) = 1.280374518688759
Norm reduction ( 2) = 0.9192503145984233
Norm reduction ( 3) = 0.8992375745724203
Norm reduction ( 4) = 0.8075275442766622
Norm reduction ( 5) = 0.6653240040986598
......
```

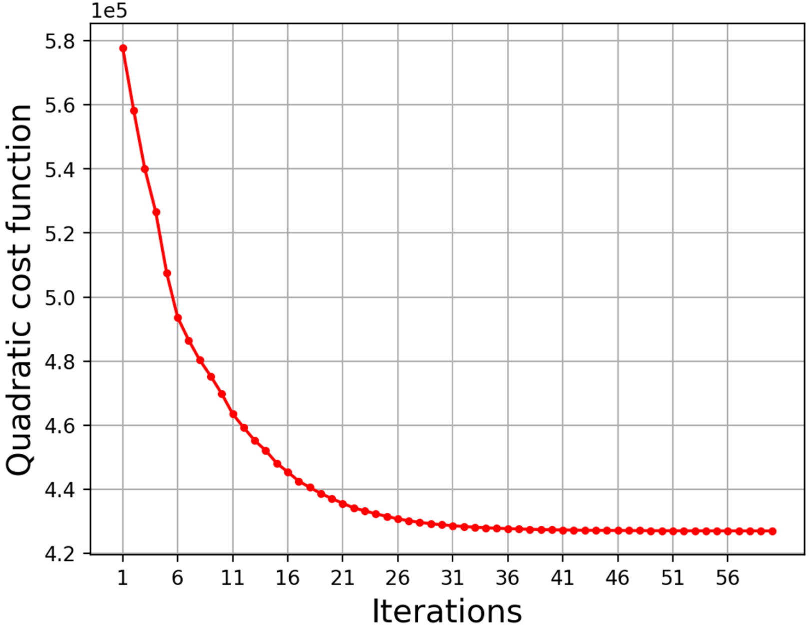
Diagnostics

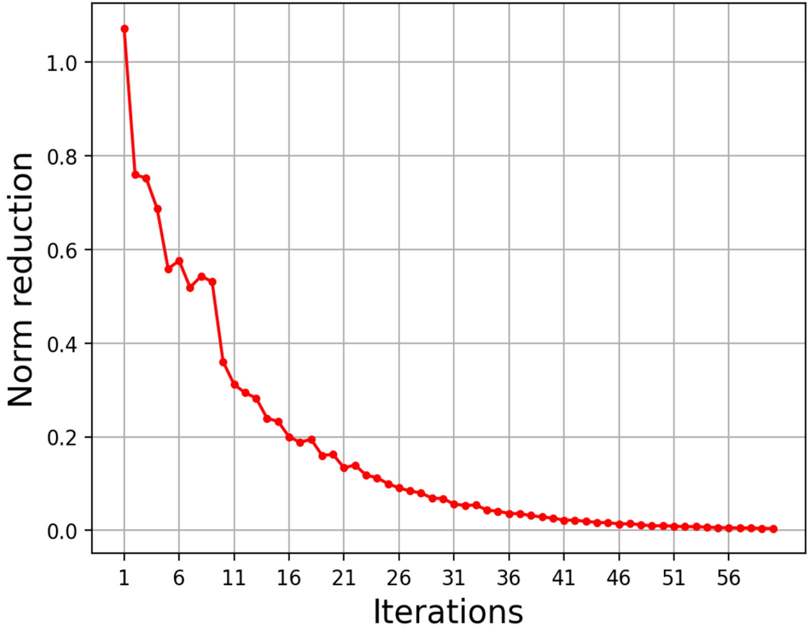
Under **graphics**:

├── standalone

  ├── plot_obs_loc_tut.py

Check cost function and norm reduction:





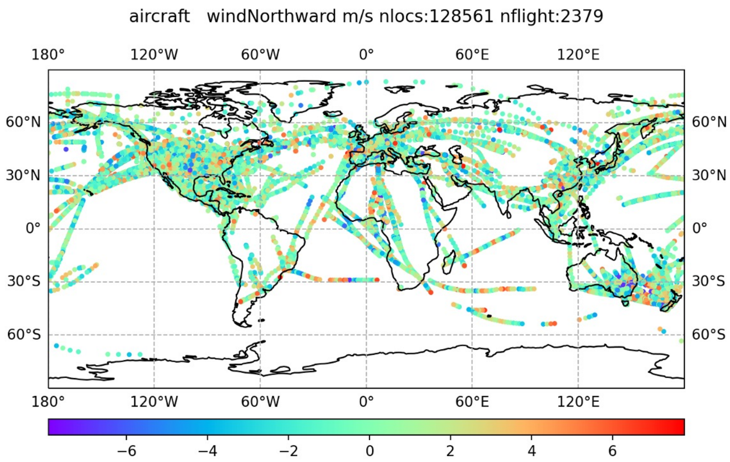
Diagnostics

Under **graphics**:

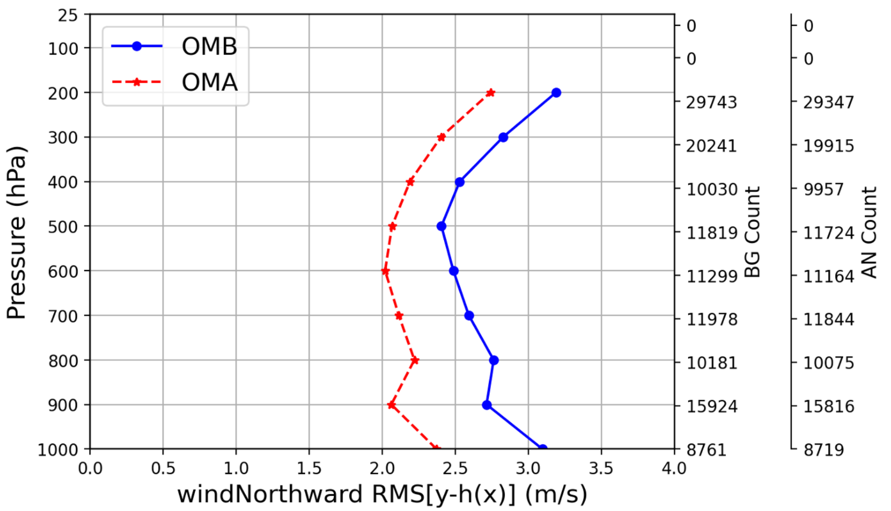
```
├── standalone
    ├── plot_diag.py
    ├── plot_diag_omboma_tut.py
```

Check observation departures figures:

OmA distribution



RMS of OmB/OmA profile



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

3DVar, B modeling, hybrid-EnVar

Presented by Jake Liu (liuz@ucar.edu)
Partially based on materials prepared by BJ Jung

Prediction, Assimilation, and Risk Communication Section
Mesoscale & Microscale Meteorology Laboratory
National Center for Atmospheric Research

NCAR

MPAS-JEDI Tutorial, INPE, 15-16 August, 2024

NSF

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

What problem a minimization algorithm solves?

Cost function in incremental form:

$$J(\delta \boldsymbol{x}) = \frac{1}{2}(\delta \boldsymbol{x} - \delta \boldsymbol{x}_{\mathrm{g}})^{\mathrm{T}}\mathbf{B}^{-1}(\delta \boldsymbol{x} - \delta \boldsymbol{x}_{\mathrm{g}}) + \frac{1}{2}(\mathbf{H}\delta \boldsymbol{x} - \boldsymbol{d})^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{H}\delta \boldsymbol{x} - \boldsymbol{d})$$

Gradient of cost function:

$$\nabla_{\delta \boldsymbol{x}} J(\delta \boldsymbol{x}) = \mathbf{B}^{-1}(\delta \boldsymbol{x} - \delta \boldsymbol{x}_{\mathrm{g}}) + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{H}\delta \boldsymbol{x} - \boldsymbol{d}) = 0$$

Analytical solution of analysis increment:

$$(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})\delta \boldsymbol{x}_a = \mathbf{B}^{-1}\delta \boldsymbol{x}_{\mathrm{g}} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\boldsymbol{d}$$

$$\Downarrow$$

$$\mathbf{A}\delta \boldsymbol{x}_a = \boldsymbol{b}$$

Final linear algebra system to solve iteratively through minimization algorithms available in OOPS

No need for computing B^{-1} in each iteration!

Instead, in each iteration of a minimization algorithm, we compute

$$\mathbf{B}\boldsymbol{r}_k \qquad \boldsymbol{r}_k = \boldsymbol{b} - \mathbf{A}\delta\boldsymbol{x}_k$$

Further reading for minimization algorithms in OOPS

https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/inside/jedi-components/oops/algorithmic_details/solvers.html

Analytical solution of analysis increment:

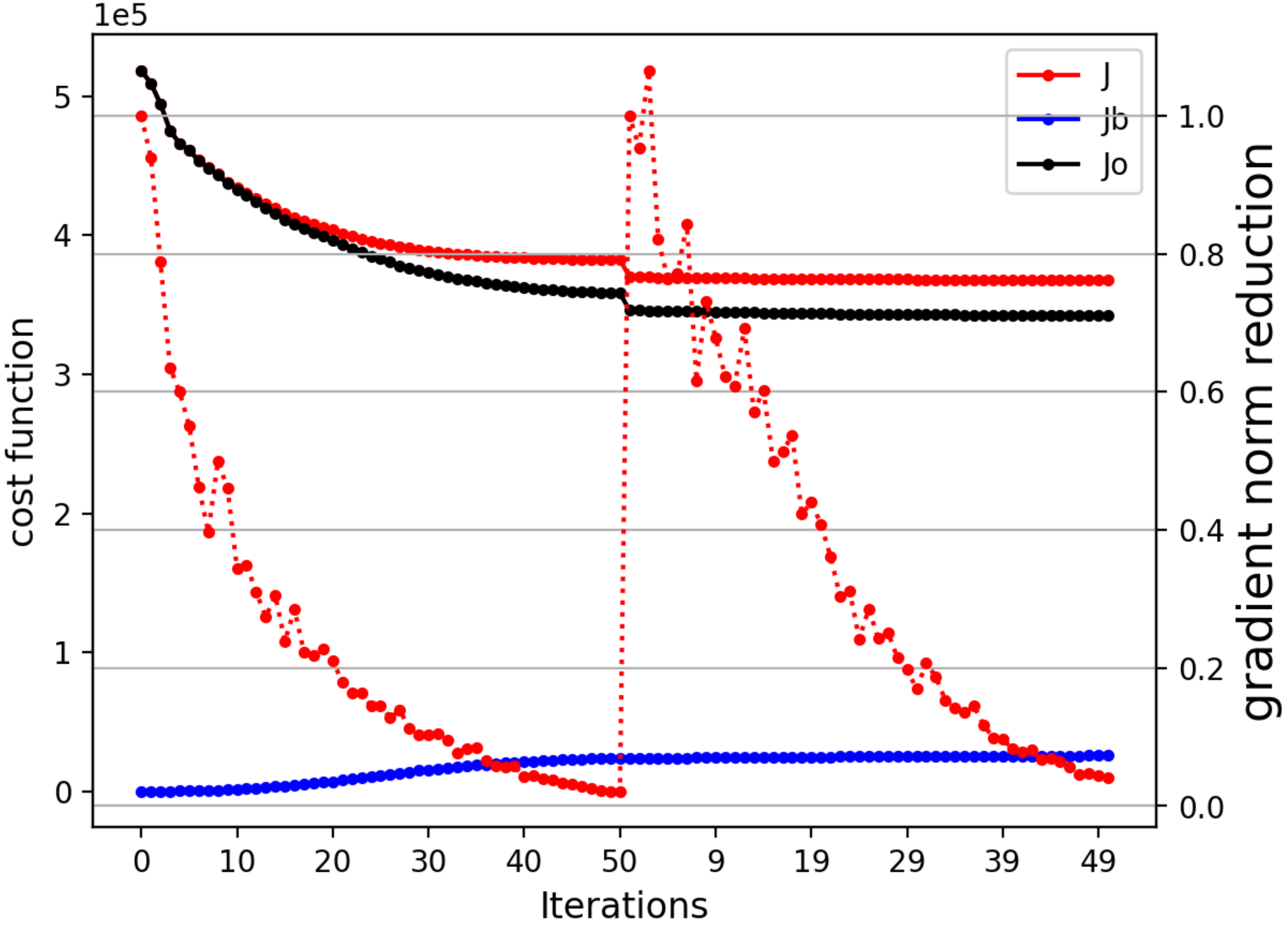
$$(\mathbf{B}^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H})\delta\boldsymbol{x}_a = \mathbf{B}^{-1}\delta\boldsymbol{x}_g + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\boldsymbol{d}$$

$$\downarrow$$

$$\mathbf{A}\delta\boldsymbol{x}_a = \boldsymbol{b}$$

Final linear algebra system to solve iteratively through minimization algorithms available in OOPS

Cost function and gradient norm reduction



How B is modeled in MPAS-JEDI's 3DVar?

$$B=K_1K_2\Sigma C\Sigma^TK_2^TK_1^T$$

- B is decomposed as a sequence of operators (or linear variable changes) (**K_1 , K_2 Σ, and C)** and their adjoint operators **(K_1^T, K_2^T**)
- Reason for doing this is that, mathematically, B matrix is a very large-dimension matrix, we can not store the full matrix in memory. We have to apply these operators in local grid points.

$\mathbf{B = K_1 K_2 \Sigma C \Sigma^T K_2^T K_1^T}$

- $\mathbf{K_1}$ is a linear variable change from stream function ($\delta\psi$) and velocity potential ($\delta\chi$) to zonal (δu) and meridional (δv) winds. This is similar to GSI or WRFDA.

$$\begin{bmatrix} \delta u \\ \delta v \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\partial}_y & -\boldsymbol{\partial}_x \\ \boldsymbol{\partial}_x & -\boldsymbol{\partial}_y \end{bmatrix} \begin{bmatrix} \delta\psi \\ \delta\chi \end{bmatrix}$$

- $\mathbf{K_1^T}$ is a corresponding adjoint operator.

$B = K_1 K_2 \Sigma C \Sigma^T K_2^T K_1^T$

- K_2 applies the linear variable change from 'unbalanced' variables to full variables. This is also similar to GSI or WRFDA

$$\begin{bmatrix} \delta\psi \\ \delta\chi \\ \delta T \\ \delta Q \\ \delta p_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{L} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{M} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} \\ \boldsymbol{N} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \delta\psi \\ \delta\chi_u \\ \delta T_u \\ \delta Q \\ \delta p_{s,u} \end{bmatrix}$$

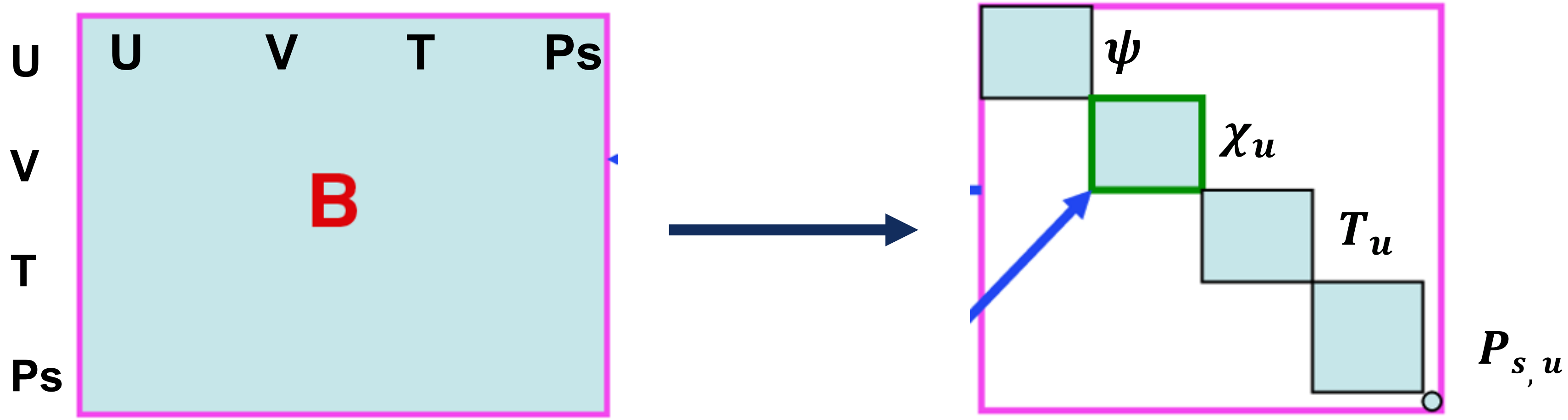
- $\delta\chi = \delta\chi_b + \delta\chi_u = \boldsymbol{L}\delta\psi + \delta\chi_u$
- $\delta T = \delta T_b + \delta T_u = \boldsymbol{M}\delta\psi + \delta T_u$
- $\delta p_s = \delta p_{s,b} + \delta p_{s,u} = \boldsymbol{N}\delta\psi + \delta\chi_u$

  - $\delta\psi$ is a predictor for the balanced part of $\delta\chi$, δT , and δp_s.
  - Full matrix for **M** & **N**, diagonal matrix for **L**

- K_2^T is a corresponding adjoint operator.

$B = K_1 K_2 \Sigma C \Sigma^T K_2^T K_1^T$

- $\Sigma C \Sigma^T$ represents the spatial covariance for $\{\delta\psi, \delta\chi_u, \delta T_u, \delta Q, \delta p_{s,u}\}$. These variables are assumed to have not cross-variable correlations.
- $\Sigma = \Sigma^T$ is a diagonal matrix with error standard deviation
- **C** is a block diagonal matrix. Each block represents the spatial correlation for $\{\delta\psi, \delta\chi_u, \delta T_u, \delta Q, \delta p_{s,u}\}$



$\mathbf{B=K_1K_2\Sigma \textcolor{red}{C} \Sigma^T K_2^T K_1^T}$

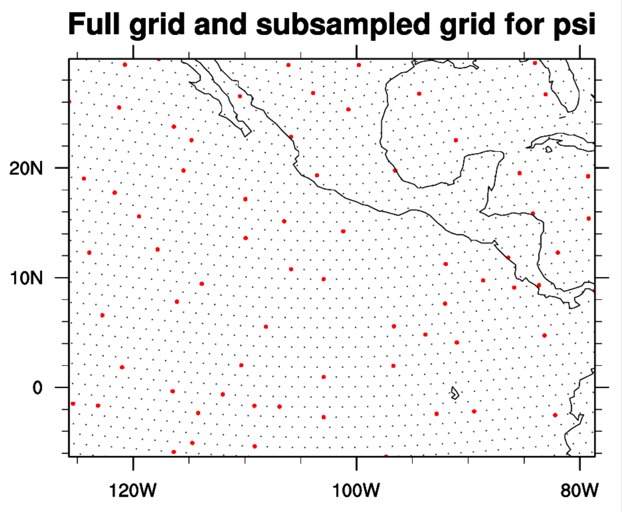
- Even with a single variable, the dimension for spatial correlation is still large.
- SABER/BUMP-NICAS applies the spatial correlation at a coarse grid ($\mathbf{C}^s$).

$$\mathbf{C} = \mathbf{N}\mathbf{S}\mathbf{C}^s\mathbf{S}^\mathrm{T}\mathbf{N}^\mathrm{T}$$

$\mathbb{R}^{m\times m}$ $\mathbb{R}^{m_s\times m_s}$ with $m_s \ll m$

$\mathbf{N}$: diagonal matrix for normalization
(to ensure the diagonal component of C equals "1")

$\mathbf{S} = \mathbf{S}^\mathrm{v}\mathbf{S}^\mathrm{h}$: Interpolation from coarse grid to full grid



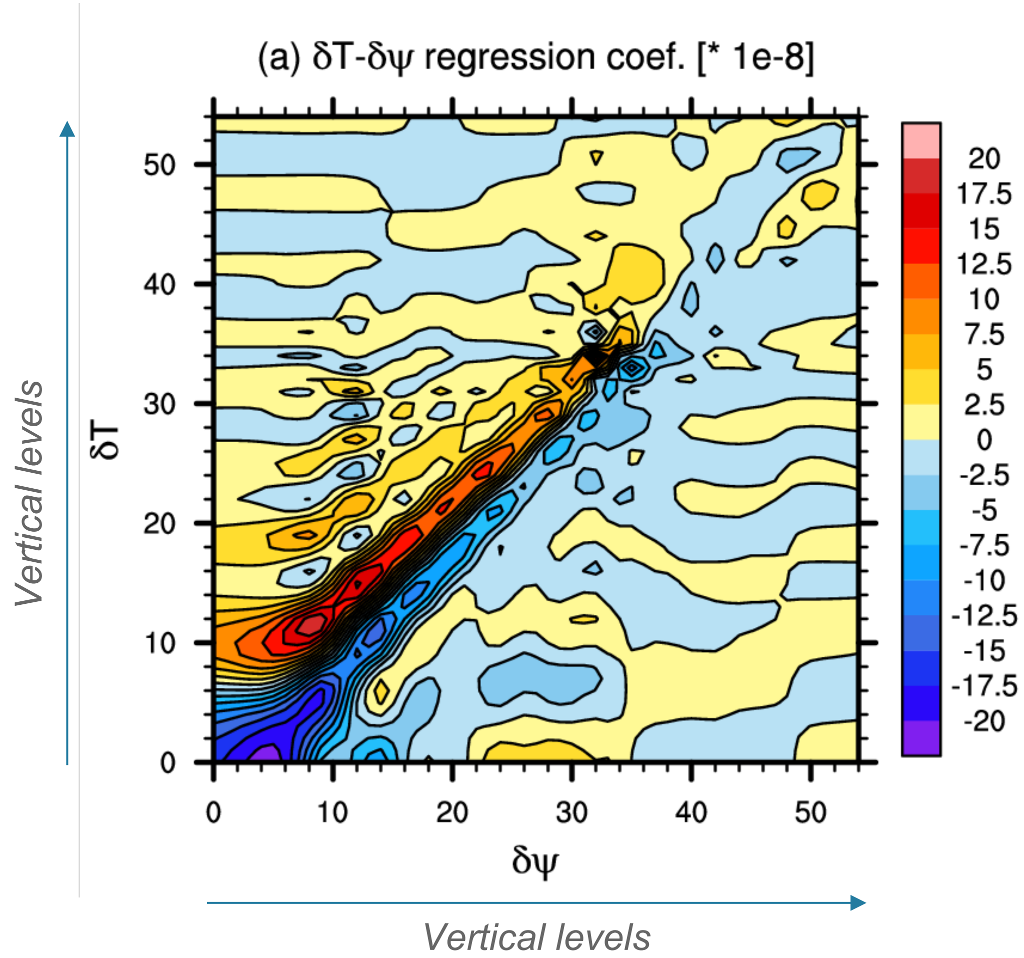
Matrix $\mathbf{C}^\mathbf{S}$ are pre-computed and stored in files according to statistics for correlation length-scales of each variable

How B (K_1, K_2, Σ, C^S) is estimated?

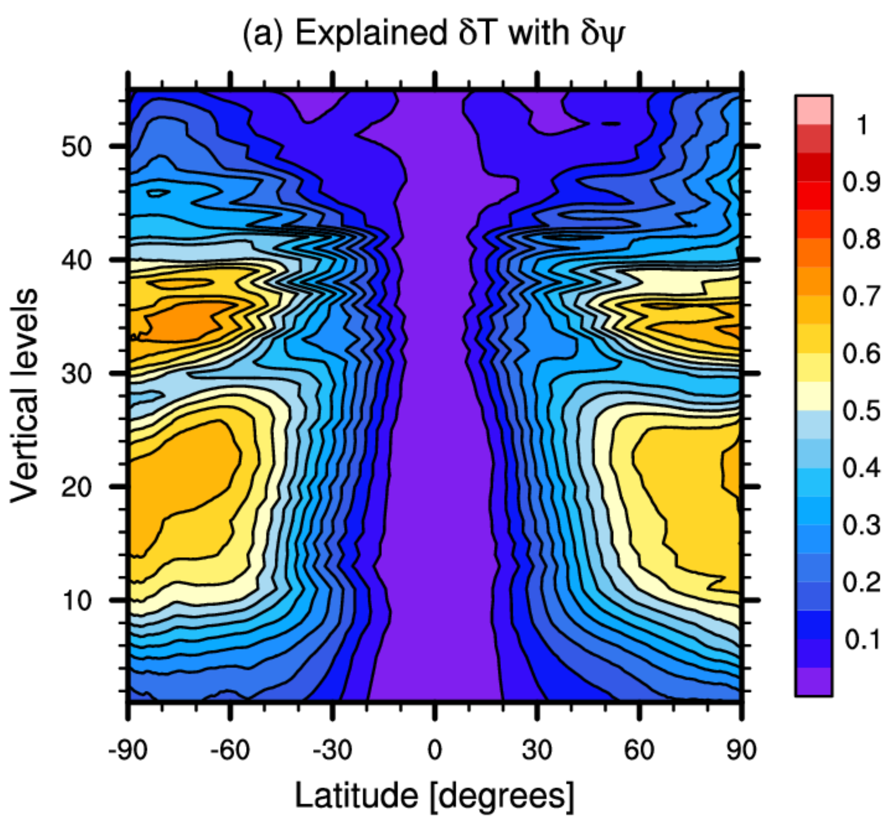
- Through the so-called 'NMC' method, which uses forecast difference pairs to do statistics, e.g., B provided in the tutorial practice is generated with
  - 366 pairs (over 3 months) of GFS 48 hour and 24 hour forecast differences at MPAS 60 km mesh.
- Additional tunings are applied to the estimated B.
  - Reducing the error STD for all variables by a factor of 1/3
  - Reducing the diagnosed horizontal lengths for $\delta\psi$ and $\delta\chi_u$ by a factor of 1/2

NOT ready to support B estimation tool

Estimated *M* at 34.8° N latitude

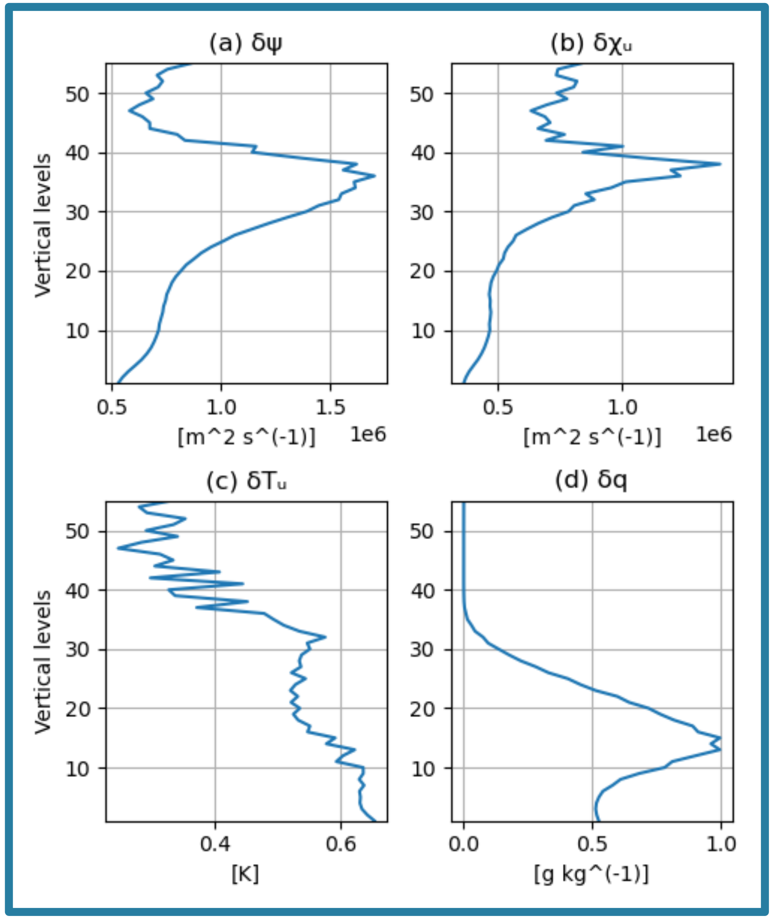


Ratio of balanced variance to total variance

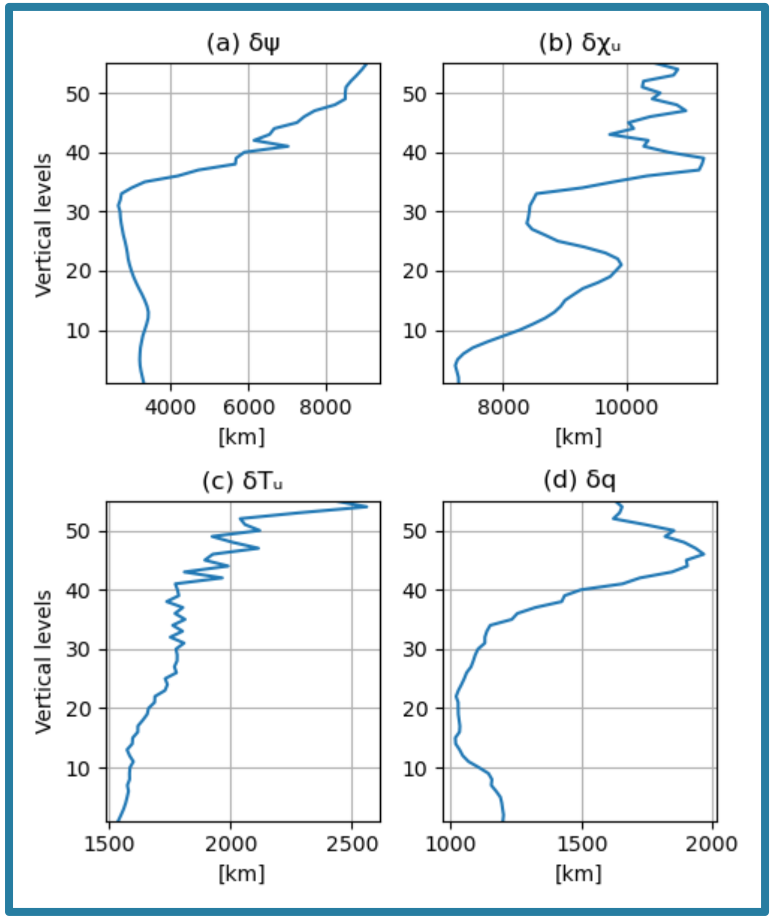


$$\frac{\langle \delta T_b, \delta T \rangle}{\langle \delta T, \delta T \rangle}$$

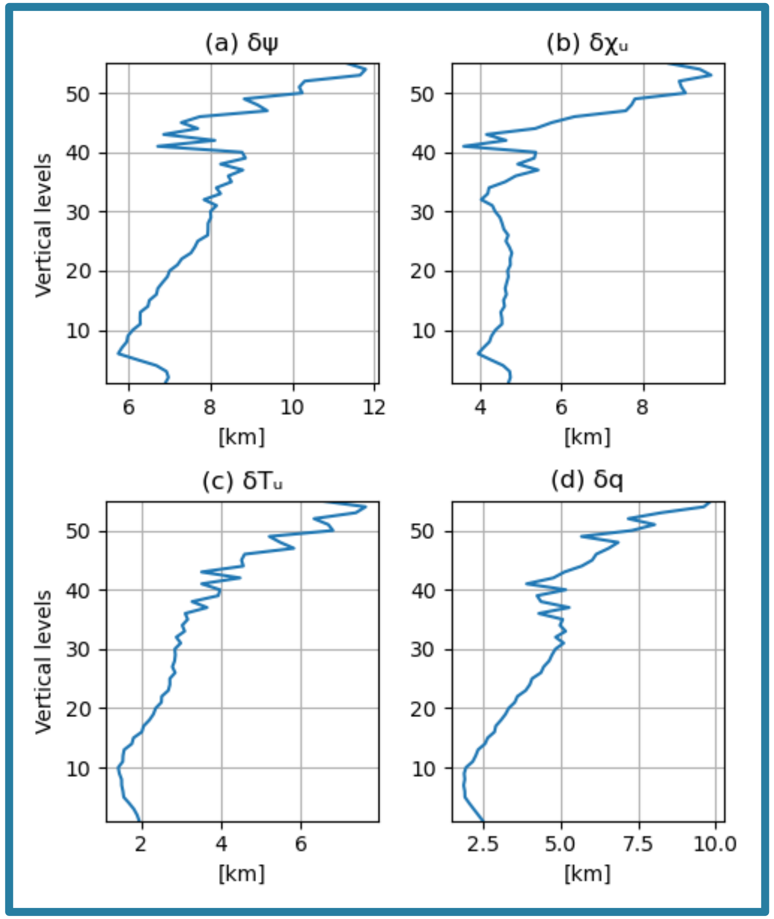
Estimated **Σ**



Estimated Horizontal correlation length-scale



Estimated vertical Correlation length-scales

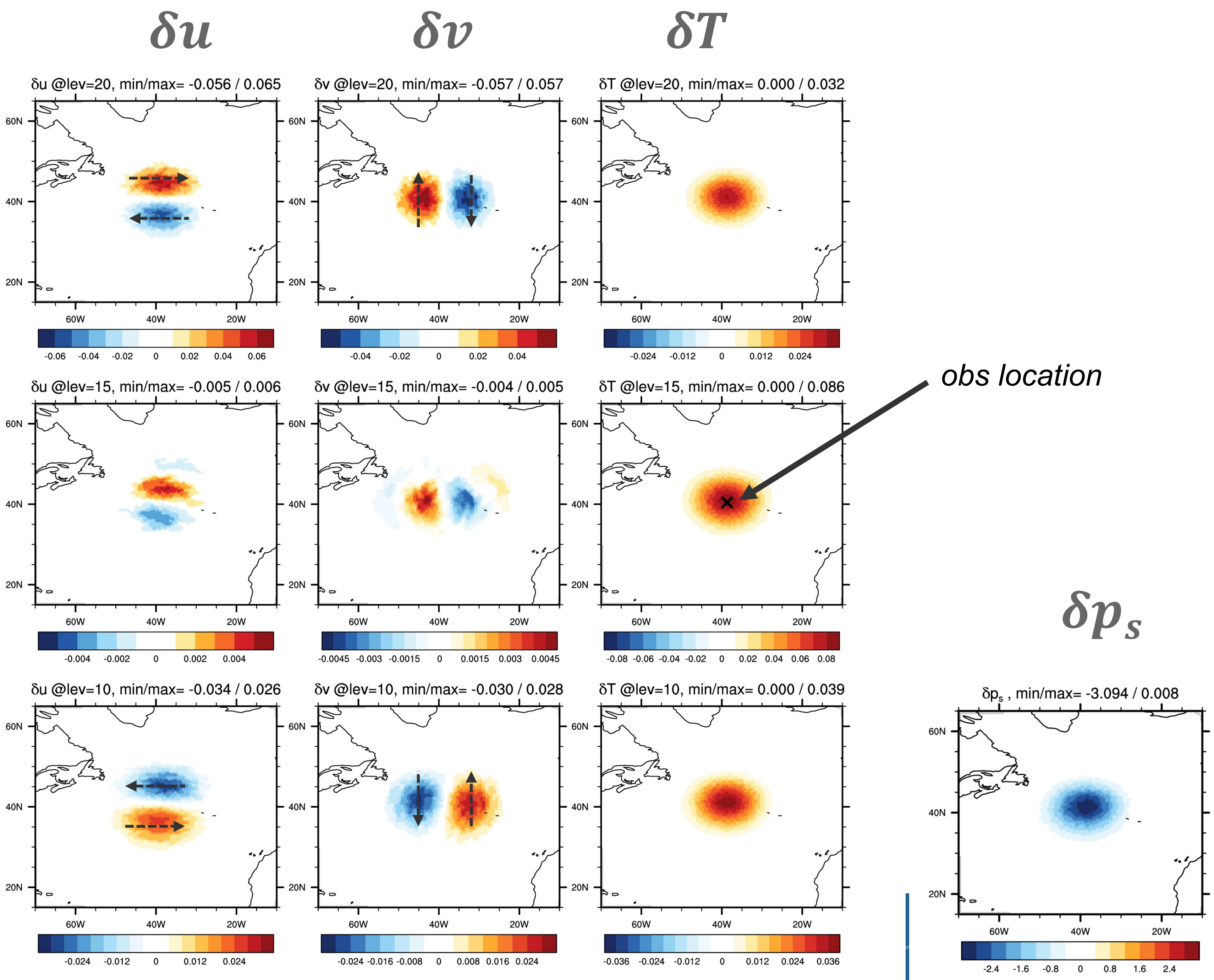


Single T obs test

Model level = 20

Model level = 15

Model level = 10



NCAR
UCAR

Previous slides present ‘multivariate’ B, MPAS-JEDI can easily do ‘univariate’ B, in that case:

$$\mathbf{B}=\mathbf{\Sigma C \Sigma^T}$$

- i.e., no cross-variable correlation between analysis variables (U, V, T, Q, Ps)

YAML configuration for 3DVar (1/6)


```
cost function:
  cost type: 3D-Var
  window begin: 2018-04-14T21:00:00Z
  window length: PT6H
  analysis variables: &incvars
[spechum,surface_pressure,temperature,uReconstructMeridional,uReconstructZonal]
  background:
    state variables:
[spechum,surface_pressure,temperature,uReconstructMeridional,uReconstructZonal,theta,rho,u,qv,pressure,landmask,xice,snowc,skintemp,ivgtyp,isltyp,snowh,vegfra,u10,v10,lai,smois,tslb,pressure_p]
    filename: ./bg.2018-04-15_00.00.00.nc
    date: &analysisDate 2018-04-15T00:00:00Z
```

YAML configuration for 3DVar (2/6)

$$B = K_1 K_2 \Sigma C \Sigma^T K_2^T K_1^T$$

```
cost function:
  ...
  background error:
    covariance model: SABER
    saber central block:
      saber block name: BUMP_NICAS
      ... more config ...
    saber outer blocks:
    - saber block name: StdDev
      ... more config ...
    - saber block name: BUMP_VerticalBalance
      ... more config ...
    linear variable change:
      linear variable change name: Control2Analysis
      ... more config ...
```

C

Σ

K_2

K_1

YAML configuration for 3DVar (3/6)

$$B=K_1K_2\Sigma C\Sigma^TK_2^TK_1^T$$

```
  background error:
    covariance model: SABER
    saber central block:
      saber block name: BUMP_NICAS
      active variables: &ctlvars
[stream_function,velocity_potential,temperature,spechum,surface_pressure]
      read:
        io:
          data directory: ./BUMP_files/bump_nicas
          files prefix: bumpcov_nicas
        drivers:
          multivariate strategy: univariate
          read local nicas: true
```

YAML configuration for 3DVar (4/6)

$$B = K_1 K_2 \Sigma C \Sigma^T K_2^T K_1^T$$

```
background error:
  covariance model: SABER
  saber central block:
    saber block name: BUMP_NICAS
    ... more config ...
  saber outer blocks:
  - saber block name: StdDev
    read:
      model file:
        filename: ./BUMP_files/stddev/mpas.stddev_0p33.2018-04-15_00.00.00.nc
        date: *analysisDate
        stream name: control
```

YAML configuration for 3DVar (5/6)

$B = K_1 K_2 \Sigma C \Sigma^T K_2^T K_1^T$

```
- saber block name: BUMP_VerticalBalance
  read:
    io:
      data directory: ./BUMP_files/bump_vertical_balance
      files prefix: bumpcov_vbal
    drivers:
      read local sampling: true
      read vertical balance: true
    vertical balance:
      vbal:
      - balanced variable: velocity_potential
        unbalanced variable: stream_function
        diagonal regression: true
      - balanced variable: temperature
        unbalanced variable: stream_function
      - balanced variable: surface_pressure
        unbalanced variable: stream_function
```

YAML configuration for 3DVar (6/6)

$$B = K_1 K_2 \Sigma C \Sigma^T K_2^T K_1^T$$

```
background error:
  covariance model: SABER
  saber central block:
    saber block name: BUMP_NICAS
    ... more config ...
  saber outer blocks:
  - saber block name: StdDev
    ... more config ...
  - saber block name: BUMP_VerticalBalance
    ... more config ...
  linear variable change:
    linear variable change name: Control2Analysis
    input variables: *ctlvars
    output variables: *incvars
```

YAML configuration for Hybrid-3DEnVar (1/2)

- 3DVar setting

```
background error:
  covariance model: SABER
  ... more config ...
```

- 3DEnVar setting

```
background error:
  covariance model: ensemble
  ... more config ...
```

- We can configure the hybrid covariance as a linear combination of two Bs !

$$\mathbf{B}_{\text{hybrid}} = \alpha \mathbf{B}_{\text{static}} + \beta \mathbf{B}_{\text{ensemble}}$$

(Hamill and Snyder, 2000)

YAML configuration for Hybrid-3DEnVar (2/2)

- We can configure the hybrid covariance as a linear combination of two Bs !

```
background error:
   covariance model: hybrid
   components:
   - weight:
       value: 0.5
     covariance:
       covariance model: SABER
       ... more config ...
   - weight:
       value: 0.5
     covariance:
       covariance model: ensemble
       ... more config ...
```

$$\mathbf{B}_{\mathbf{hybrid}} = \alpha\mathbf{B}_{\mathbf{static}} + \beta\mathbf{B}_{\mathbf{ensemble}}$$

2-stream I/O (1/3)

- To reduce disk space usage, we use “mpasout” file instead of “restart” file for MPAS-JEDI’s background and analysis file.

- Also “time invariant” fields in a separate file and “mpasout” file excludes those “time invariant” fields and also physical tendency fields.

- So MPAS-JEDI will need to read in two streams (two files)
    - “**invariant**” stream: mesh info, sfc input variables (landmask, shdmin, 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 (2/3)

- For a cold start forecast, “invariant” stream file should be set to the “invariant.nc” file, generated by MPAS *init_atmosphere* executable.
  - In “namelist.atmosphere”

```
&restart
    config_do_DAcycling = false
/
```

an invariant.nc file is linked or copied to the working directory

- For forecast step of cycling exp, “input” stream should point the file generated from “da_state” stream.
  - In “namelist.atmosphere”

```
&restart
    config_do_DAcycling = true
/
```

an invariant.nc file is linked or copied to the working directory

2-stream I/O (3/3)

- For DA step of cycling exp, setting will be
  - In “namelist.atmosphere”

```
&restart
    config_do_DAcycling = true
/
&assimilation
    config_jedi_da = true
/
```

References

Bannister, R.N. (2008a), A review of forecast error covariance statistics in atmospheric variational data assimilation. I: Characteristics and measurements of forecast error covariances. Q.J.R. Meteorol. Soc., 134: 1951-1970. https://doi.org/10.1002/qj.339

Bannister, R.N. (2008b), A review of forecast error covariance statistics in atmospheric variational data assimilation. II: Modelling the forecast error covariance statistics. Q.J.R. Meteorol. Soc., 134: 1971-1996. https://doi.org/10.1002/qj.340

Derber, J. and Bouttier, F.: A reformulation of the background error covariance in the ECMWF global data assimilation system, Tellus A: Dynamic Meteorology and Oceanography, 51, 195–221, https://doi.org/10.3402/tellusa.v51i2.12316, 1999.

Gaspari, G. and Cohn, S. E.: Construction of correlation functions in two and three dimensions, Quarterly Journal of the Royal Meteorological Society, 125, 723–757, https://doi.org/https://doi.org/10.1002/qj.49712555417, 1999.

Hamill, T. M. and Snyder, C.: A Hybrid Ensemble Kalman Filter–3D Variational Analysis Scheme, Monthly Weather Review, 128, 2905 – 2919, https://doi.org/10.1175/1520-0493(2000)128<2905:AHEKFV>2.0.CO;2, 2000.

Jung, B.-J., Ménétrier, B., Snyder, C., Liu, Z., Guerrette, J. J., Ban, J., Baños, I. H., Yu, Y. G., and Skamarock, W. C.: Three-dimensional variational assimilation with a multivariate background error covariance for the Model for Prediction Across Scales–Atmosphere with the Joint Effort for data Assimilation Integration (JEDI-MPAS 2.0.0-beta), Geosci. Model Dev. Discuss. [preprint], https://doi.org/10.5194/gmd-2023-131, in review, 2023.

Ménétrier, B.: Normalized Interpolated Convolution from an Adaptive Subgrid documentation, https://github.com/benjaminmenetrier/nicas_doc/blob/master/nicas_doc.pdf, 2020.

Wu, W.-S., Purser, R. J., and Parrish, D. F.: Three-dimensional variational analysis with spatially inhomogeneous covariances, Monthly Weather Review, 130, 2905–2916, https://doi.org/10.1175/1520-0493(2002)130<2905:TDVAWS>2.0.CO;2, 2002.

This material is based upon work supported by the NSF National Center for Atmospheric Research, which is a major facility operated by the U.S. National Science Foundation under Cooperative Agreement No. 1852977.

Observations (3):
Satellite Radiance Data Assimilation

Presented by Ivette Hernández Baños based on materials prepared by Zhiquan (Jake) Liu

Prediction, Assimilation, and Risk Communication Section
Mesoscale & Microscale Meteorology Laboratory
National Center for Atmospheric Research

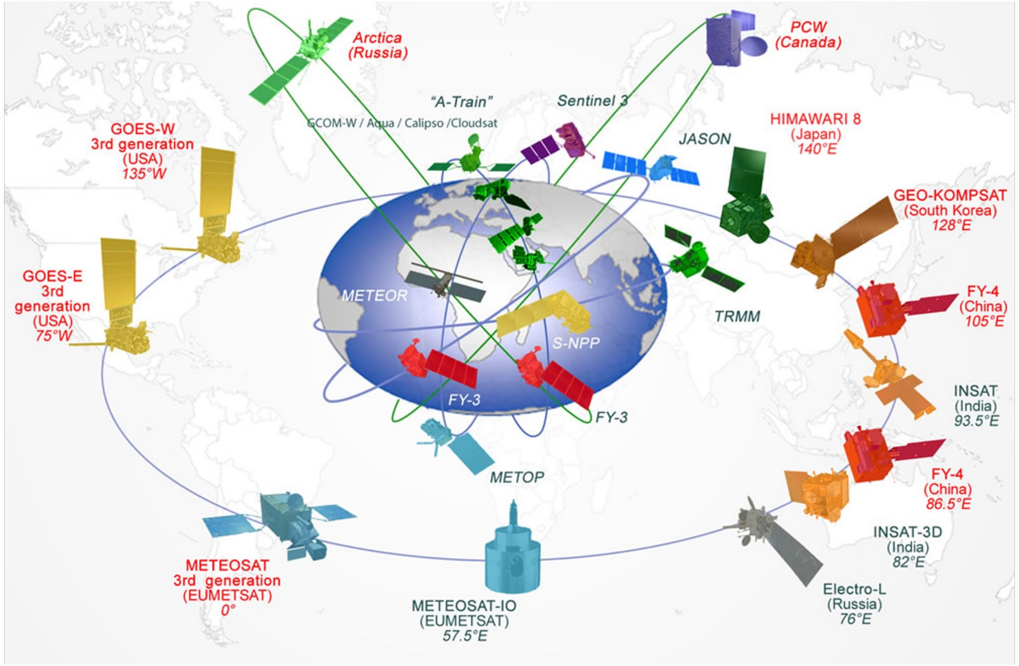
MONAN: INPE MPAS-JEDI Training 2024, Cachoeira Paulista, São Paulo, Brazil
August 15-16, 2024

Outline

1. Background
2. Principles of satellite measurements
3. Radiative Transfer Model
4. Radiance DA setting with MPAS-JEDI
5. Variational Bias Correction
6. All-sky radiance DA

Background

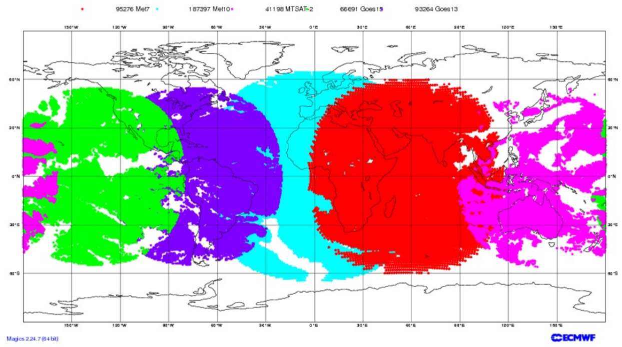
Environmental monitoring satellites

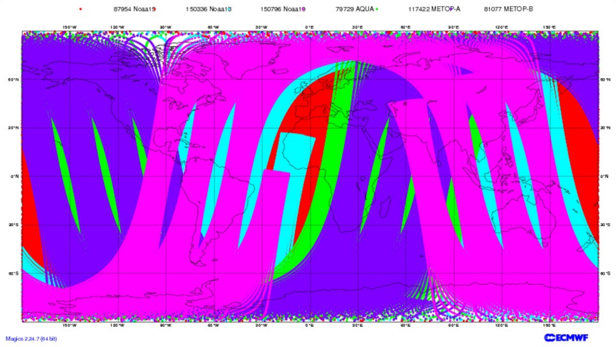


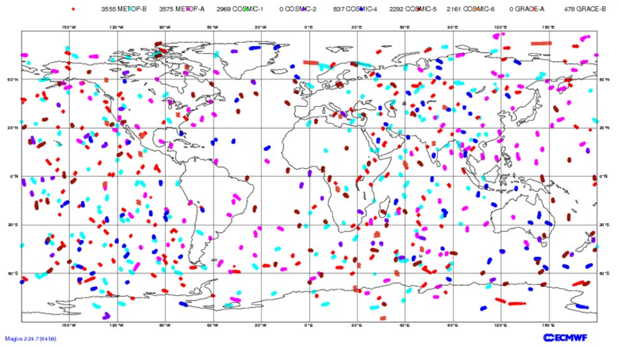
Polar-orbiting satellites vs. Geostationary satellites

Background

ECMWF data coverage for 06 UTC 05/Jul/2015 (All obs DA)







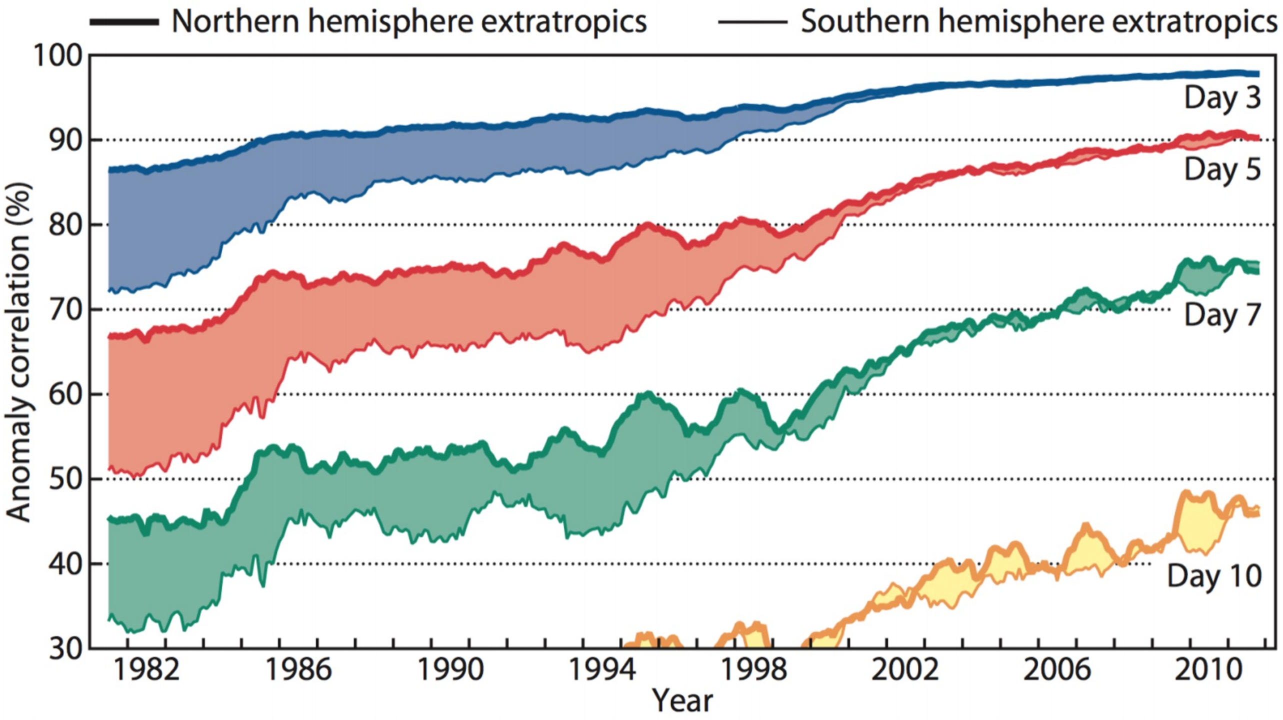
GRAD
Total obs: 483826

AMSU-A
Total obs: 777314

GPSRO
Total obs: 15867

Background

Global forecast improvement over time at ECMWF 2012



Background

Current status (2023) of satellite radiance DA at ECMWF

(Courtesy of Niels Bormann, ITSC-24)

A – Assimilated; P – Passively monitored; E – Under evaluation; X – Failed or data excluded due to quality/transmission issues; 🌦 – All-sky treatment
Changes since ITSC-23 are highlighted through orange shading.

Satellite	Present orbit position (LTAN, approx.)	MW temperature sounder	MW humidity sounder	MW imager	IR broadband sounder or imager	IR hyper-spectral sounder
NOAA-15	19:30	A 🌦	X		X	
NOAA-18	22:30	A 🌦	X		X	
NOAA-19	20:30	A 🌦	A 🌦		P	
NOAA-20	13:30	A	A			A
NOAA-21	13:30	E	E			
Aqua	13:30	X	X			A
S-NPP	13:30	A	A			A
Metop-B	21:30	A 🌦	A 🌦		X	A
Metop-C	21:30	A 🌦	A 🌦			A
FY-3C	19:00	X	A 🌦	X		
FY-3D	14:00	P 🌦	A 🌦	P 🌦 & X		E
FY-3E	17:30	E 🌦	A 🌦			
DMSP-F17	18:30		A 🌦	A 🌦		
DMSP-F18	16:00		A 🌦	P 🌦 & E		
GCOM-W1	13:30			A 🌦		
GPM	Mid-incl.		A 🌦	A 🌦		
Meteosat-9	45.5°E				A	
Meteosat-11	0°				A	
GOES-16	75.2°W				A	
GOES-18	137°W				A	
Himawari-9	140.7°E				A	
FY-4A	104.7°E					E
FY-4B	133°E					E

Principles of satellite measurements

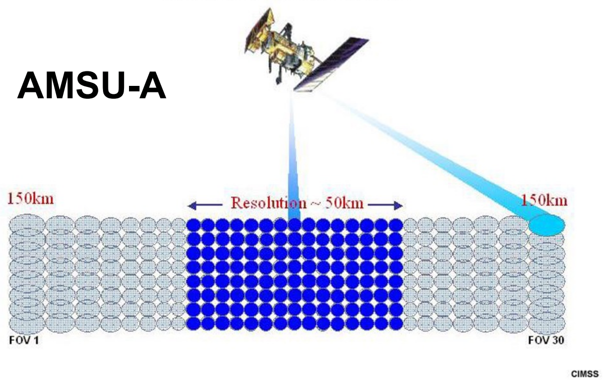
Types of sensors → | Passive | Active | GNSS radio occultation

Scan strategies and viewing geometry affect coverage and field-of-view (FOV) resolution:

<u>cross-track scan</u>

- Resolution degrades toward the edge of the swath because the viewing angle changes across the swath

AMSU-A



<u>conical scan</u>

- Constant ground resolution
- Generally narrower swaths than cross-track scan swaths

GMI

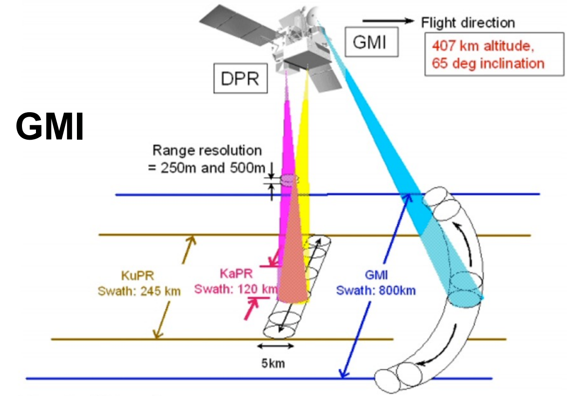
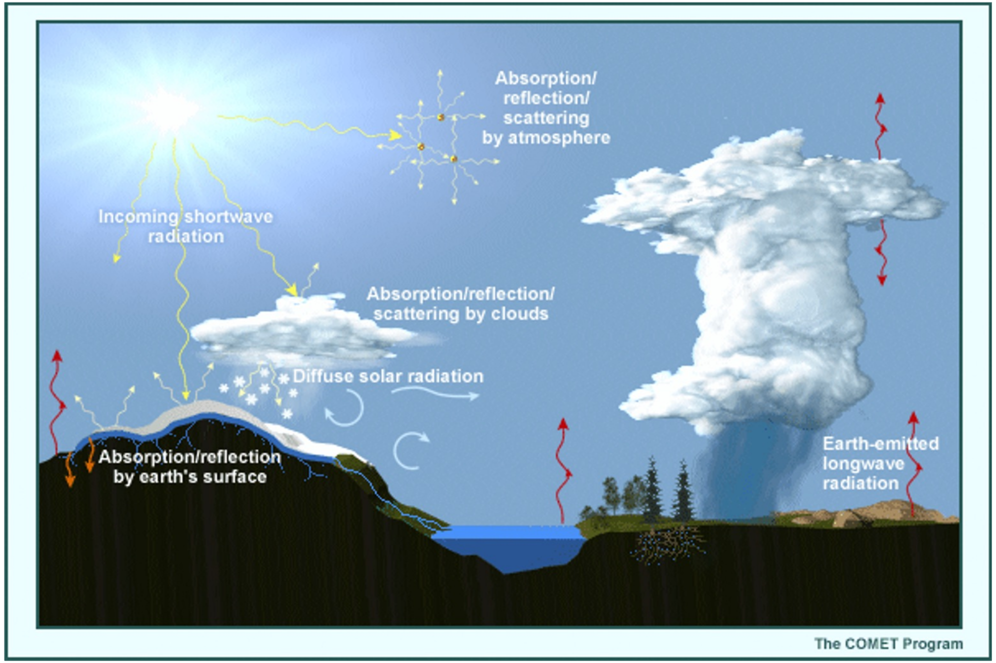


Figure 2. GPM swath measurements

Principles of satellite measurements

What do satellite instruments measure?

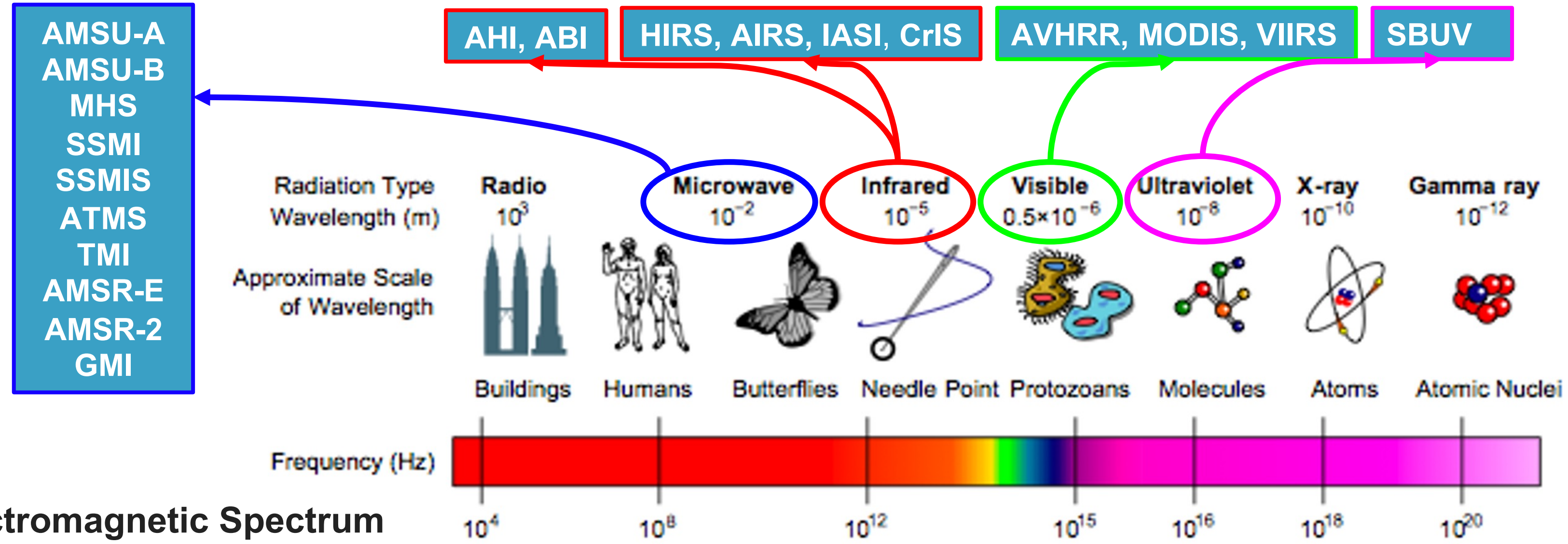
- **Satellite passive sensors** observe radiation emitted and scattered from Earth’s surface and atmosphere at discrete wavelength intervals



Principles of satellite measurements

What do satellite instruments measure?

⇒ Different sensors measure radiation at different wavelengths (e.g., MW, IR, VIS)



Electromagnetic Spectrum

Principles of satellite measurements

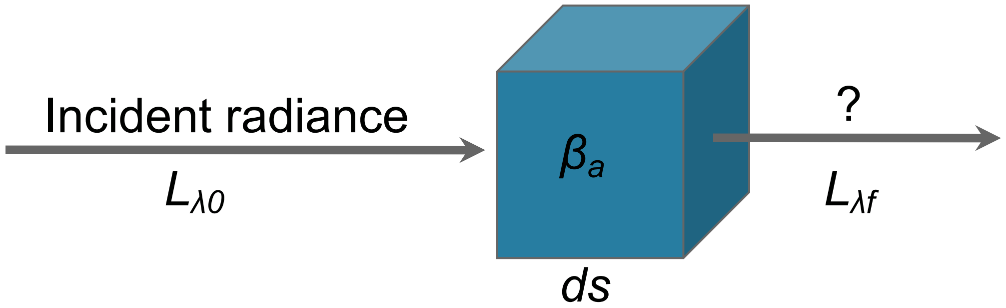
What is radiance?

- Radiance (L) is the amount of energy per unit area per unit time per unit solid angle emitted at a wavelength λ (or frequency v)
  - Recall, $c = \lambda v$, where c is the speed of light.
- Physically, can think of radiance as the “brightness” of an object
- Radiance is related to geophysical atmospheric variables by the radiative transfer equation
- Radiances are often converted to **brightness temperature** (equivalent blackbody temperature, by inverting Plank function)

Principles of satellite measurements

Atmospheric Transmittance

- Consider radiation at wavelength λ with radiance $L_{\lambda 0}$ incident upon an absorbing medium of thickness *ds*
  - Use an absorption coefficient (β_a ; units m^{-1}) to quantify degree of absorption
- Ignore emission from the medium and scattering
- What is the radiance on the other side of the surface?



Principles of satellite measurements

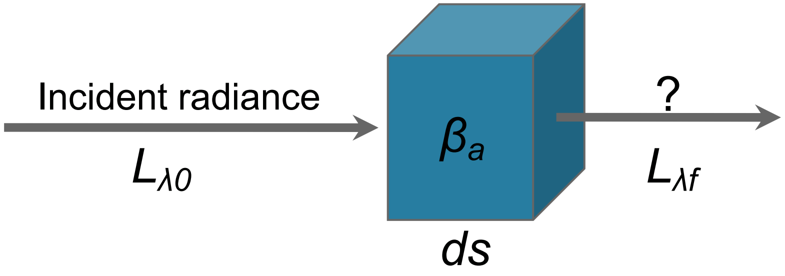
Atmospheric Transmittance

- Beer's Law gives the amount of radiation emerging from the material:

$$L_{\lambda f} = L_{\lambda 0} \exp\left[-\int_{s_1}^{s_2} \beta_a(s) ds \right]$$

- The ratio of the amount of radiation that emerges from the cube to the amount that entered is the transmittance:

$$\tau_\lambda = \frac{L_{\lambda f}}{L_{\lambda 0}} = \exp\left[-\int_{s_1}^{s_2} \beta_a(s) ds \right]$$



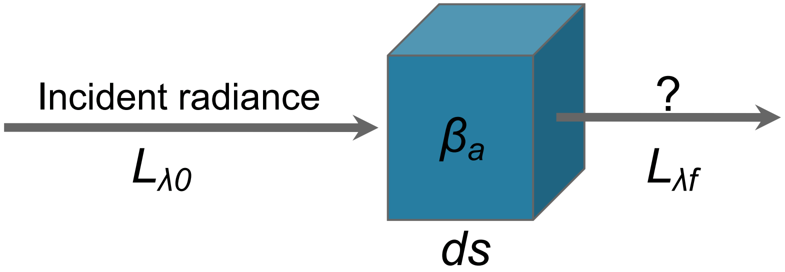
Principles of satellite measurements

Atmospheric Transmittance

- Transmittance in the real atmosphere varies in space (<u>especially in the vertical</u>) and time

- Letting a_λ denote the <u>absorption</u> of the medium at wavelength λ, then in the absence of scattering:

$$\boxed{a_\lambda + \tau_\lambda = 1}$$



Radiative Transfer Model

$$L(\nu) = \int_0^\infty B(\nu, T(z)) \left[\frac{d\tau(\nu)}{dz} \right] dz + \text{Surface} + \text{Cloud/Rain Aerosol}$$

TOA radiance at frequency ν

Planck function

Atmospheric Absorption (weighting function)

Emission/reflection

Diffusion/scattering

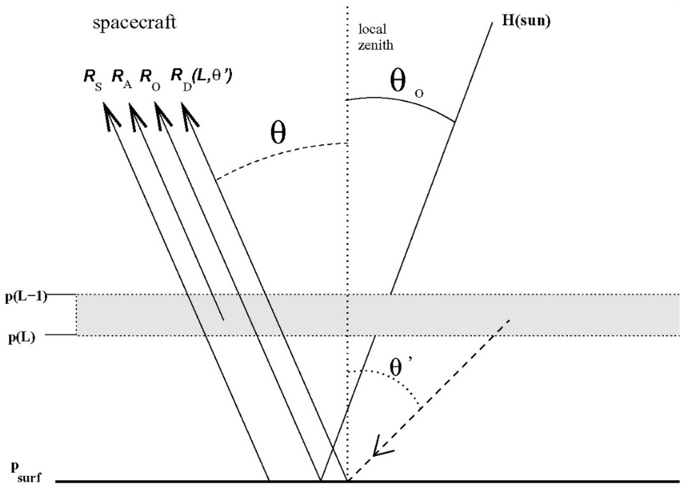
Surface emission R_s

Upwelling atmosphere emission R_A

Reflected solar radiation R_O

Down-welling & reflected atmos.

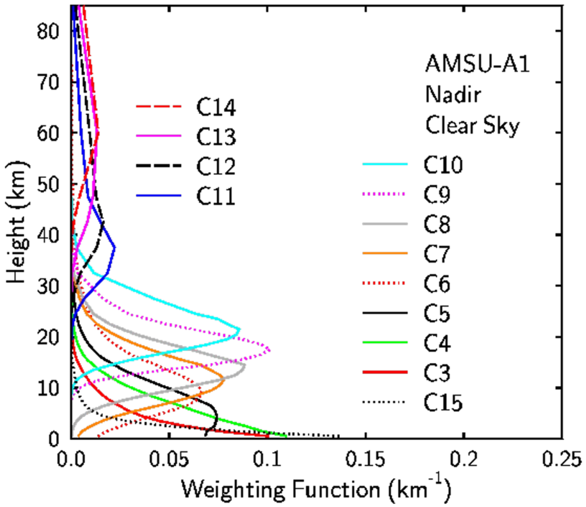
Emission (R_D)



Radiative Transfer Model

Weighting functions

- ❑ Weighting functions indicate the contribution to the outgoing radiance from various layers of the atmosphere
- ❑ Weighting functions are frequency (channel) dependent



<u>Channel selection for NWP data assimilation</u>

- **Atmospheric sounding channels** (measured radiance has no contribution from the surface)
- **Window channels** are sensitive to properties associated with earth and ocean surfaces as well as clouds

Radiance DA setting with MPAS-JEDI

YAML setting for radiative transfer model

```yaml
_clear crtm: &clearCRTMObsOperator
  name: CRTM
  SurfaceWindGeoVars: uv
  Absorbers: [H2O, O3]
  linear obs operator:
    Absorbers: [H2O]
  obs options: &CRTMObsOptions
    EndianType: little_endian
    CoefficientPath: ./crtm_coeffs_v2/
    IRVISlandCoeff: USGS
```

```yaml
- obs space:
    <<: *ObsSpace
    name: amsua_n18
    obsdatain:
      engine:
        type: H5File
        obsfile: ./amsua_n18_obs_2018041500.h5
    obsdataout:
      engine:
        type: H5File
        obsfile: ./obsout_da_amsua_n18.h5
    simulated variables: [brightnessTemperature]
    channels: &amsua_n18_channels 1-15
  obs error: *ObsErrorDiagonal
  obs operator:
    <<: *clearCRTMObsOperator
    obs options:
      <<: *CRTMObsOptions
      Sensor_ID: amsua_n18
  get values:
```

Radiance DA setting with MPAS-JEDI

YAML settings for channel selection and quality control

```yaml
  obs filters:
  - filter: PreQC
    maxvalue: 0
#  Useflag check #amsua-n18
  - filter: Bounds Check
    filter variables:
    - name: brightnessTemperature
      channels: *amsua_n18_channels
    test variables:
    - name: ObsFunction/ChannelUseflagCheckRad
      channels: *amsua_n18_channels
      options:
        channels: *amsua_n18_channels
        use_flag: [-1, -1, -1, -1,  1,
                    1,  1,  1,  1, -1,
                   -1, -1, -1, -1, -1 ]
    minvalue: 1.0e-12
    action:
      name: reject
  - filter: Background Check
    threshold: 3.0
    <<: *multiIterationFilter
```

Much more you can set
for quality control, but not able
to cover too much this time

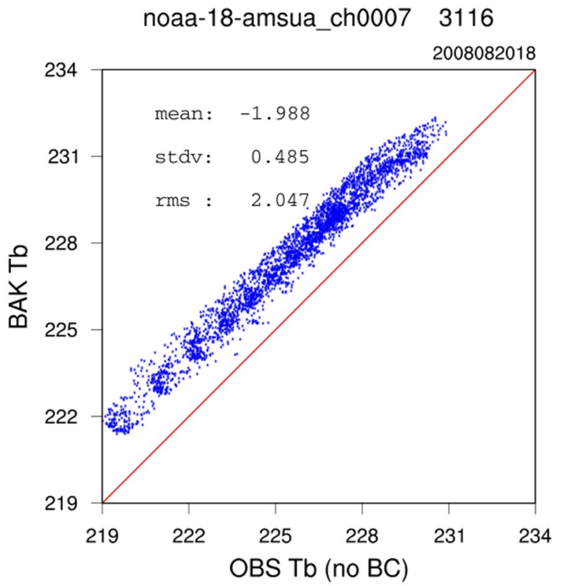
Variational Bias Correction (VarBC)

Modeling errors for satellite radiances

$$y = H(x_t) + B(\beta) + \varepsilon \quad \begin{cases} \langle \varepsilon \rangle = 0 \\ B(\beta) = \sum_{i=1}^{N} \beta_i p_i \end{cases}$$

Bias-correction coefficients

Predictors: e.g.,
- **Offset (i.e., 1)**
- **Temperature lapse rate**
- **Scan, Scan^2, Scan^3**

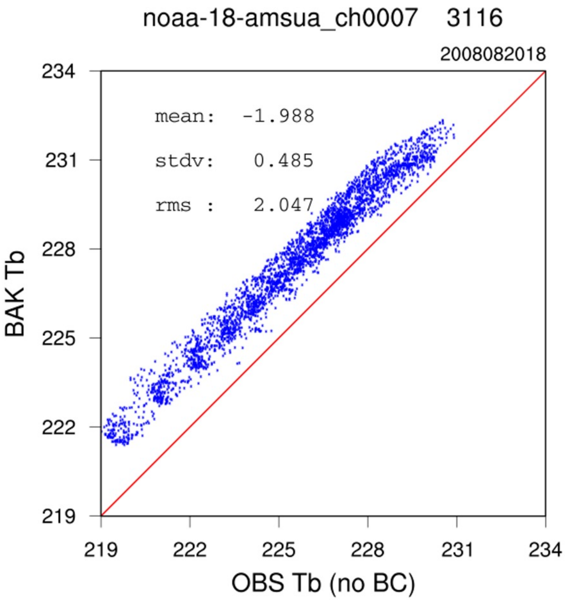


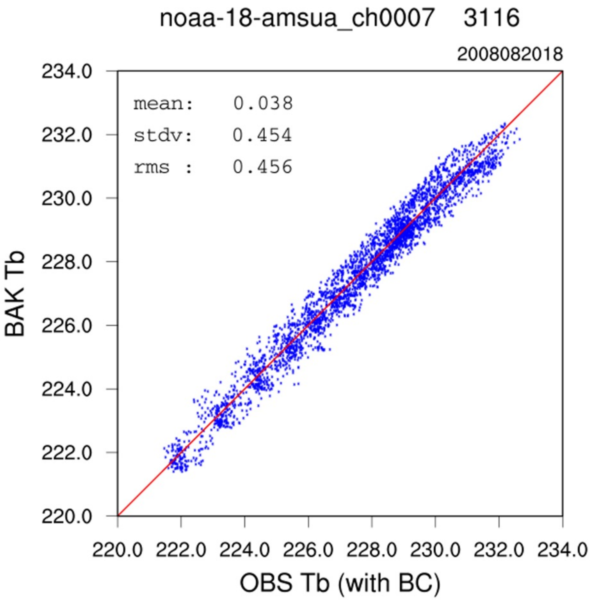
$$\mathbf{J}(\mathbf{x},\beta) = \underbrace{(\mathbf{x}_b - \mathbf{x})^T \mathbf{B}_x^{-1} (\mathbf{x}_b - \mathbf{x})}_{J_b\text{: background term for x}} + \underbrace{[\mathbf{y} - H(\mathbf{x}) - B(\beta)]^T \mathbf{R}^{-1} [\mathbf{y} - H(\mathbf{x}) - B(\beta)]}_{J_o\text{: corrected observation term}}$$
$$+ \underbrace{(\beta_b - \beta)^T \mathbf{B}_\beta^{-1} (\beta_b - \beta)}_{J_p\text{: background term for } \beta}$$

Variational Bias Correction (VarBC)

Modeling errors for satellite radiances

No bias correction





With bias correction

Variational Bias Correction (VarBC)

JEDI's bias correction coefficient file

```
netcdf satbias_amsua_n18 {
dimensions:
        nchannels = 15 ;
        npredictors = 12 ;
variables:
        float bias_coeff_errors(npredictors, nchannels) ;
        float bias_coefficients(npredictors, nchannels) ;
        int channels(nchannels) ;
        int nchannels(nchannels) ;
                nchannels:suggested_chunk_dim = 15LL ;
        int npredictors(npredictors) ;
                npredictors:suggested_chunk_dim = 12LL ;
        float number_obs_assimilated(nchannels) ;
        string predictors(npredictors) ;

// global attributes:
                string :_ioda_layout = "ObsGroup" ;
                :_ioda_layout_version = 0 ;
```

satbias_amsua_n18.h5
satbias_cov_mhs_n18.h5

```
predictors = "constant", "zenith_angle", "cloud_liquid_water",
   "lapse_rate_order_2", "lapse_rate",
   "cosine_of_latitude_times_orbit_node", "sine_of_latitude", "emissivity",
   "scan_angle_order_4", "scan_angle_order_3", "scan_angle_order_2",
   "scan_angle" ;
```

Variational Bias Correction (VarBC)

YAML setting for VarBC

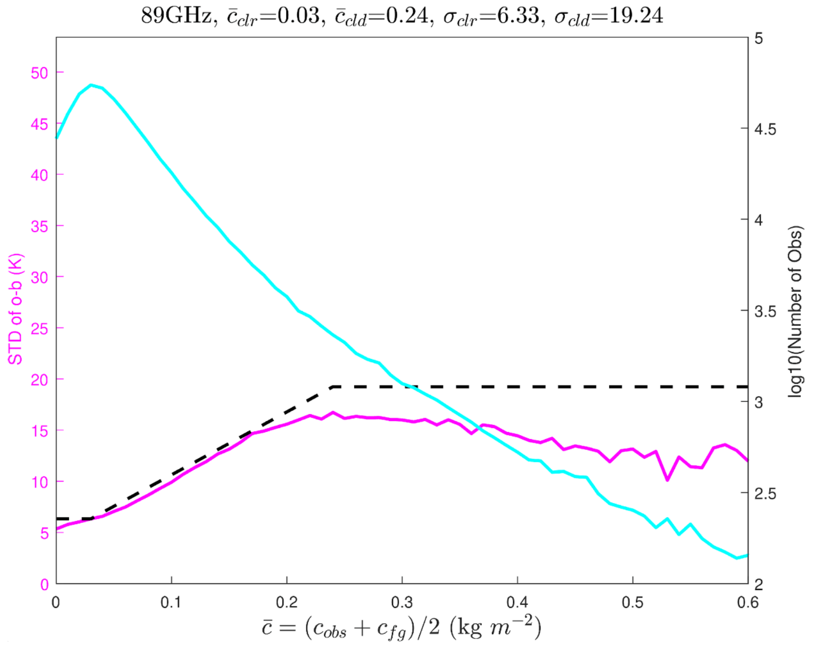
```yaml
obs bias:
  input file: {{biasCorrectionDir}}/satbias_amsua_n18.h5
  output file: {{OutDBDir}}{{MemberDir}}/satbias_amsua_n18.h5
  variational bc:
    predictors: &predictors3
    - name: constant
    - name: lapse_rate
      order: 2
      tlapse: &amsua18tlap {{fixedTlapmeanCov}}/amsua_n18_tlapmean.txt
    - name: lapse_rate
      tlapse: *amsua18tlap
    - name: emissivity
    - name: scan_angle
      order: 4
    - name: scan_angle
      order: 3
    - name: scan_angle
      order: 2
   covariance:
     minimal required obs number: 20
     variance range: [1.0e-6, 10.]
     step size: 1.0e-4
     largest analysis variance: 10000.0
     prior:
       input file: {{biasCorrectionDir}}/satbias_cov_amsua_n18.h5
       inflation:
         ratio: 1.1
         ratio for small dataset: 2.0
     output file: {{OutDBDir}}{{MemberDir}}/satbias_cov_amsua_n18.h5
```

$$B(\beta) = \sum_{i=1}^{N} \beta_i p_i$$

$$\mathbf{J}(\mathbf{x},\beta) = \underbrace{(\mathbf{x}_\mathrm{b} - \mathbf{x})^\mathrm{T} \mathbf{B}_\mathrm{x}^{-1} (\mathbf{x}_\mathrm{b} - \mathbf{x})}_{\mathrm{J_b: \ background\ term\ for\ x}} + \underbrace{\left[\mathbf{y} - H(\mathbf{x}) - B(\beta)\right]^\mathrm{T} \mathbf{R}^{-1} \left[\mathbf{y} - H(\mathbf{x}) - B(\beta)\right]}_{\mathrm{J_o: \ corrected\ observation\ term}}$$
$$+ \underbrace{(\beta_\mathrm{b} - \beta)^\mathrm{T} \mathbf{B}_\beta^{-1} (\beta_\mathrm{b} - \beta)}_{\mathrm{J_p: \ background\ term\ for\ \beta}}$$

All-sky radiance DA

Situation-dependent all-sky obs error model

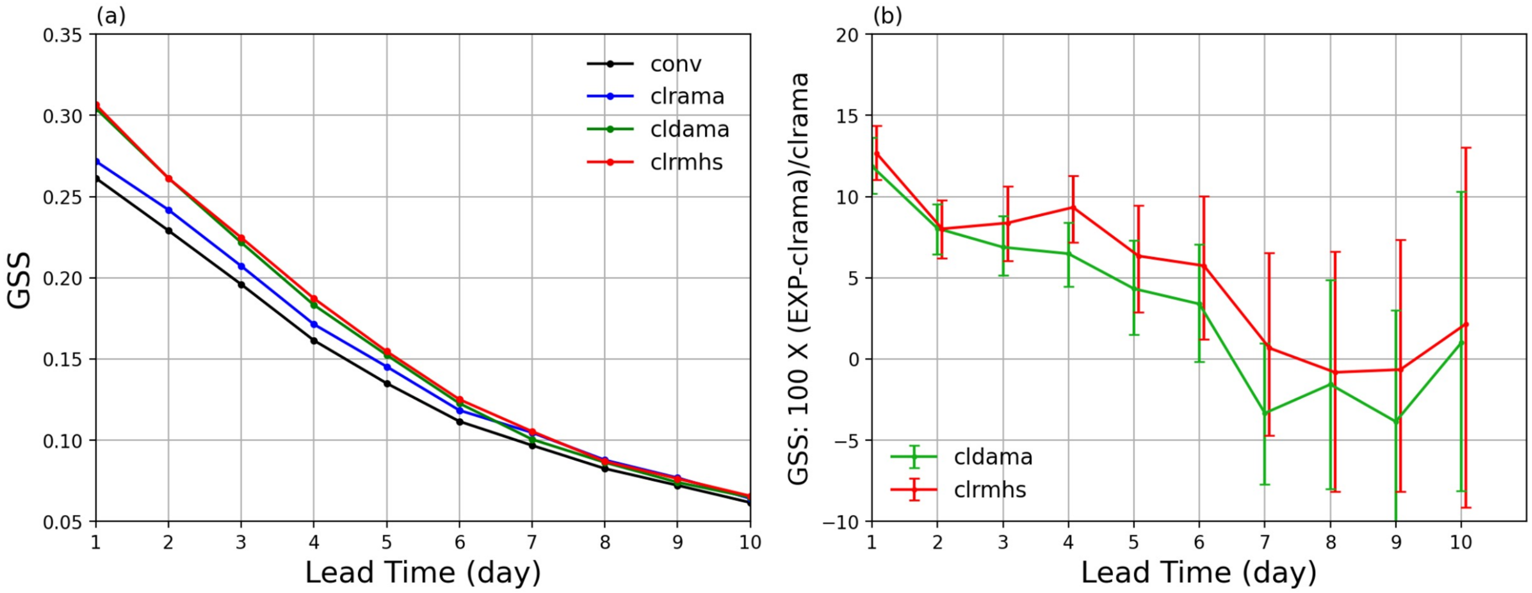


All-sky obs error model for AMSU-A channel 15:

- ❑ Observation error is a function of cloud liquid water path retrieved from channel 1 and 2's brightness temperature

All-sky radiance DA

Gilbert Skill Score of 1-10-day rainfall FC w.r.t. CMORPH obs

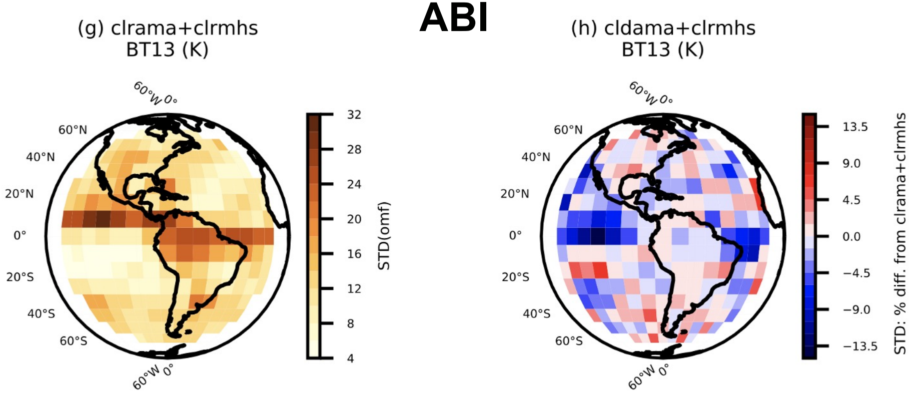


Liu et al., 2022

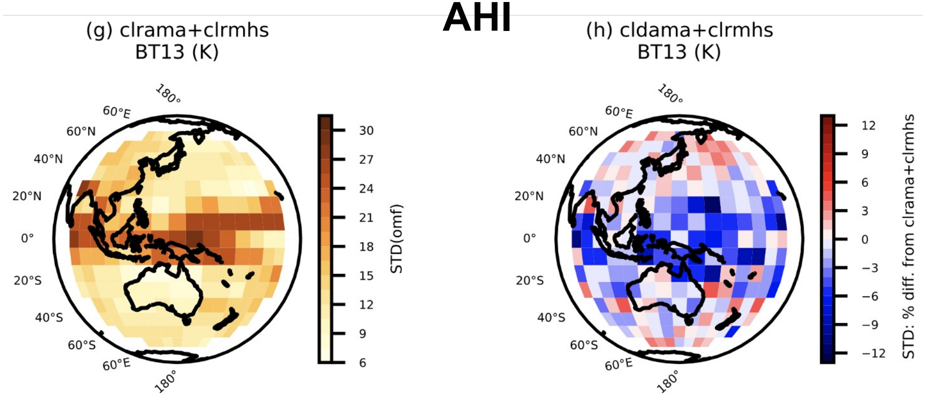
All-sky radiance DA

Added value of all-sky AMSU-A

ABI



AHI



Day-1 forecast

Error STD reduction

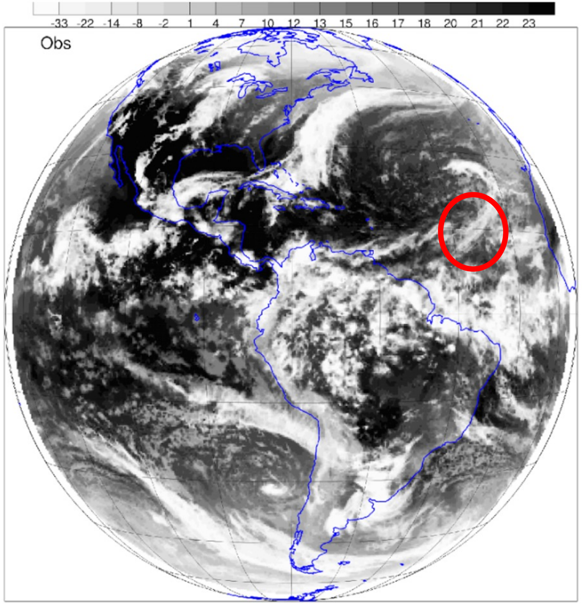
Improvement concentrated in cloudy regions of Tropics
Up to 12-14%

All-sky radiance DA

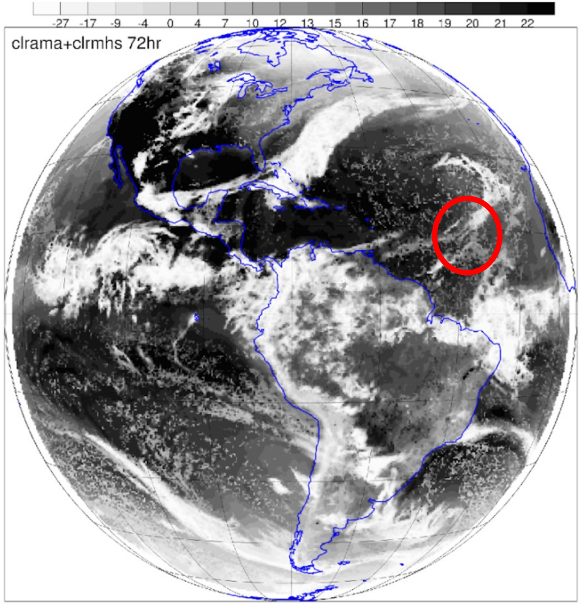
Observations vs. Day-3 forecast

ABI channel 13 BTs (degree C) valid at 00 UTC 9 May 2018

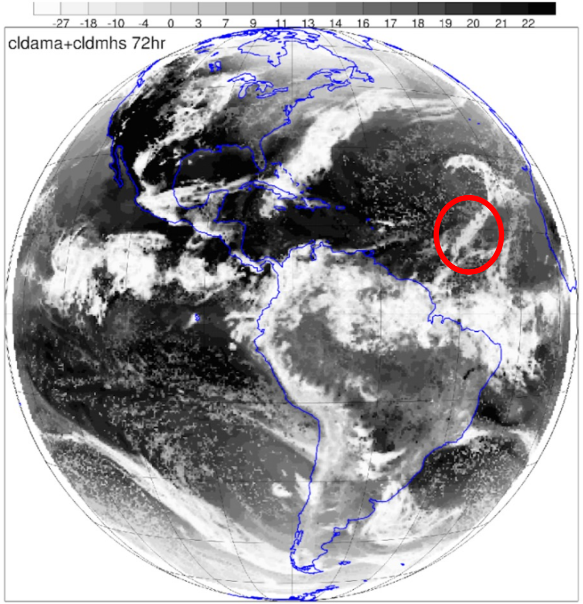
Observations



Clear-sky DA

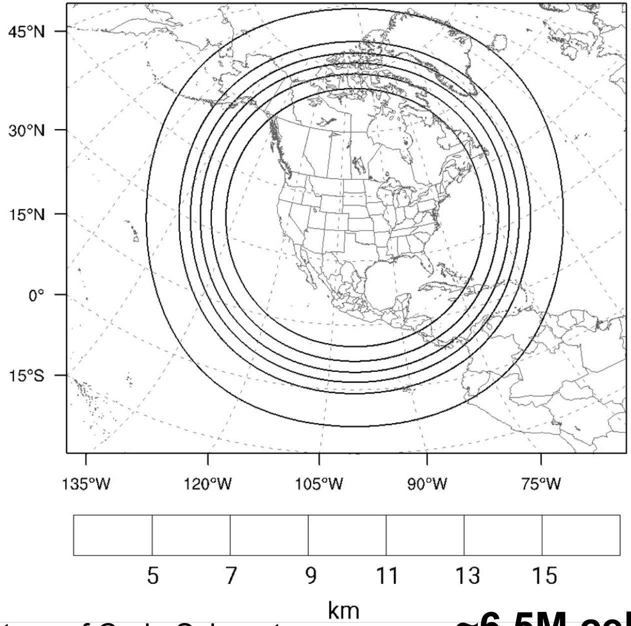


All-sky DA



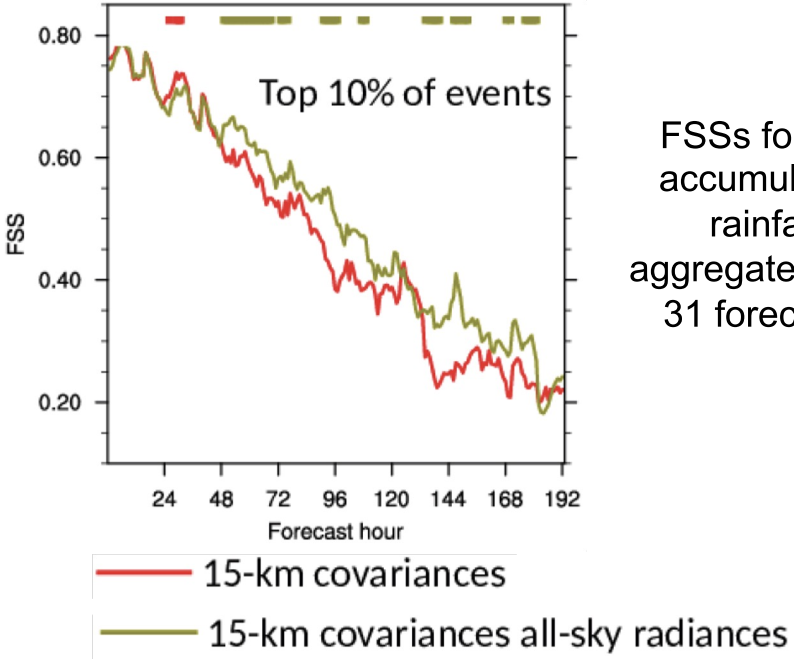
All-sky radiance DA

3DEnVar exps @ global 15km-3km variable-resolution mesh (centered over US) with the 80-member 15km ensemble input



Courtesy of Craig Schwartz

~6.5M cells



FSSs for 1-h accumulated rainfall aggregated over 31 forecasts

Concluding Remarks

- Radiance DA is complex
  - Cloudy radiative transfer, QC, bias correction, all-sky obs error model
  - Different complexity for assimilating different sensors' data

- Much more to explore for satellite DA in general
  - Visible band, near IR, active sensors, small satellites, …

- JEDI framework allows much greater flexibility to configure/tune without code change, ease science discovery
  - e.g., you can combine the use of CRTM and RTTOV in the same run!

Ensemble Data Assimilation in MPAS-JEDI: EDA and LETKF

Presented by Jake Liu
Based on materials prepared by Tao Sun

Prediction, Assimilation, and Risk Communication Section
Mesoscale & Microscale Meteorology Laboratory
National Center for Atmospheric Research

NCAR

MPAS-JEDI Tutorial, INPE, 15-16 August, 2024

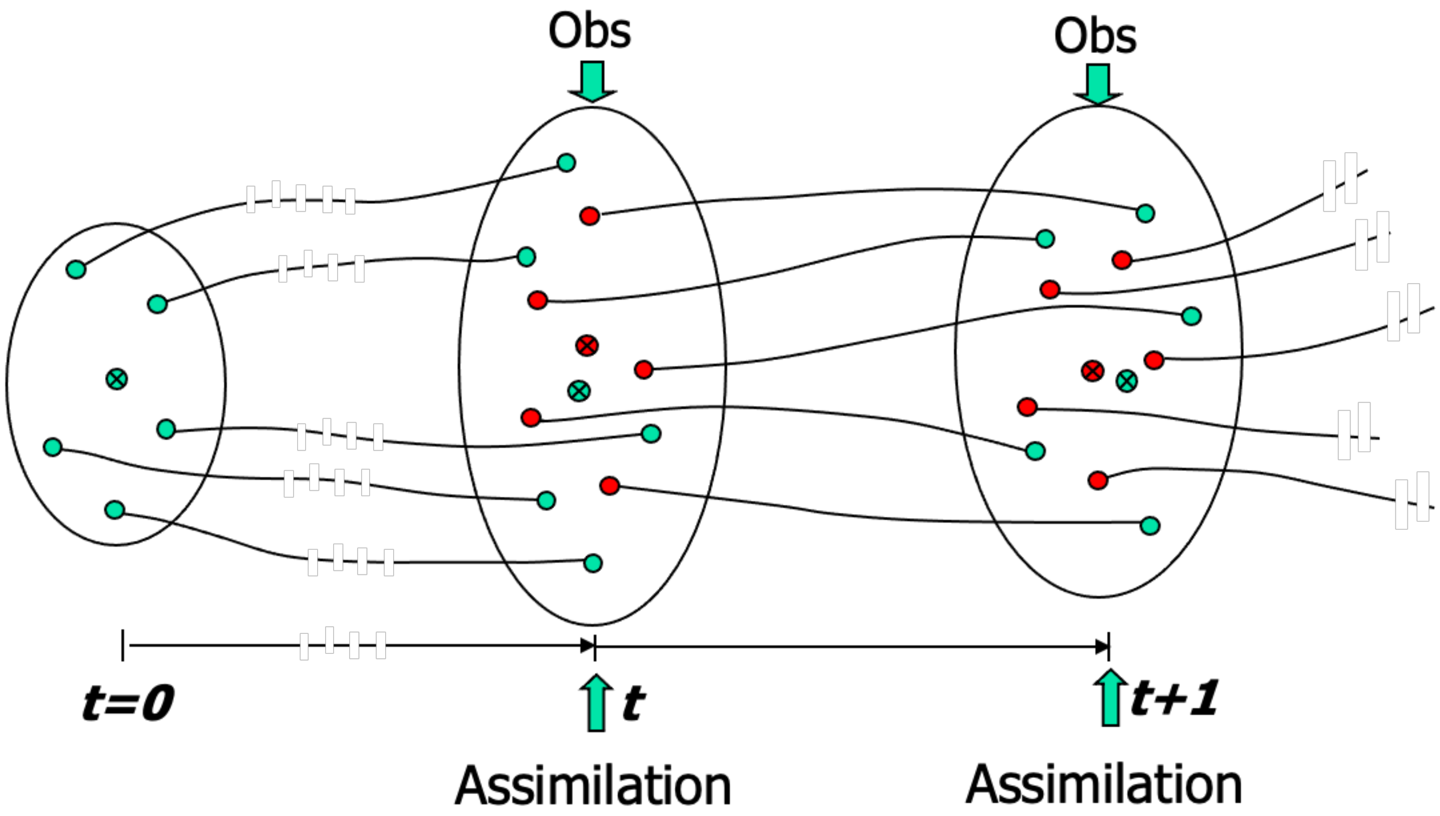
NSF

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

Outline

- Ensemble data assimilation methods
- Ensemble of data assimilation (EDA)
- Local ensemble transform Kalman filter (LETKF)
- Comparison between EDA and LETKF

Ensemble data assimilation methods



Flow-chart of ensemble data assimilation

Benefits of ensemble DA:

❖ Provide uncertainty estimate of forecast and analysis;

❖ Provide flow-dependent background error covariance (BEC) for deterministic DA methods.

Ensemble data assimilation methods

Two methods of ensemble data assimilation available for MPAS-JEDI:

- The ensemble of data assimilation (**EDA**) method.

- , Local Ensemble Transform Kalman Filter (LETKF) and gain form of LETKF

Outline

- Ensemble data assimilation methods
- Ensemble of data assimilation (EDA)
- Local ensemble transform Kalman filter (LETKF)
- Comparison between EDA and LETKF

Ensemble of Data Assimilation (EDA)

In EDA, the ensemble analysis are obtained by solving *N* independent variational cost functions with perturbed observations, where the *i*th EDA cost function is:

$$J(\mathbf{x}_i) = \frac{1}{2}\left(\mathbf{x}_i - \mathbf{x}_i^b\right)^{\mathrm{T}} \mathbf{B}_i^{-1}\left(\mathbf{x}_i - \mathbf{x}_i^b\right) + \frac{1}{2}\left[H(\mathbf{x}_i) - \mathbf{y}^o - \boldsymbol{\epsilon}_i\right]^{\mathrm{T}} \mathbf{R}^{-1}\left[H(\mathbf{x}_i) - \mathbf{y}^o - \boldsymbol{\epsilon}_i\right]$$

$\mathbf{x}_i^b$: *i*th background states;

$\mathbf{x}_i$: *i*th analysis states;

$\mathbf{B}_i$: *i*th background error covariance matrix;

$\mathbf{R}$: observation error covariance matrix;

$\mathbf{y}^o$: observation states;

$\boldsymbol{\epsilon}_i$: *i*th random observation errors; $\boldsymbol{\epsilon} \sim N(0, \mathbf{R})$, and $\sum_{i=1}^{N} \boldsymbol{\epsilon}_i = 0$.

Configure and Run EDA

For each EDA member, a specific yaml file is needed, i.e., *3denvar_${memebr}.yaml*.

❖ Each EDA member can be done with a single command:
$mpirun ./mpasjedi_variational.x 3denvar_${memebr}.yaml

Configure and Run EDA

The configuration of each EDA assimilation member is very similar to the common variational DA, but some parameters need setting.

❖ Introduction of observation random errors

Set “obs perturbations” to true in the observations section:

```
observations:
    obs perturbations: true
```

For each observation type, the observation error should be

```
obs error:
    covariance model: diagonal
    zero-mean perturbations: true
    member: 1 # index of EDA member
    number of members: 20 # ensemble size
```

Configure and Run EDA

The configuration of each EDA assimilation member is very similar to the common variational DA, but some parameters need setting.

❖ Self-exclusion in ensemble BEC

```
members from template:
    template:
      <<: *memberConfig
      filename: ../../bg/mem%iMember%/bg.2018-04-15_00.00.00.nc
    pattern: %iMember%
    start: 1
    zero padding: 3
    nmembers: 19 #Number of EDA member -1
    except: [1] # Index of EDA member
```

Configure and Run EDA

Posterior Inflation: Relaxation To Prior Perturbation (RTPP)

After all EDA members are updated, a posterior inflation is needed to keep the ensemble spread using an external executable **mpasjedi_rtpp.x**

$mpirun ./mpasjedi_rtpp.x rtpp.yaml

```
_state read: &stateReadConfig
  date: 2018-04-15T00:00:00Z
  state variables: [spechum,surface_pressure,temperature,uReconstructMeridional,uReconstructZonal,pressure_p,pressure,rho,theta,u,qv]
  stream name: background
output:
  filename: ${OUTPUT_DIR}/mem%{member}%/an.$Y-$M-$D_$h.$m.$s.nc
  stream name: analysis
geometry:
  nml_file: namelist.atmosphere
  streams_file: streams.atmosphere
  deallocate non-da fields: true
analysis variables: [spechum,surface_pressure,temperature,uReconstructMeridional,uReconstructZonal,pressure_p,pressure,rho,theta,u,qv]
background:
  members:
  - <<: *stateReadConfig
    filename: ${BAKDIR}/mem001/bg.2018-04-15_00.00.00.nc
  ...
analysis:
  members:
  - <<: *stateReadConfig
    filename: ${ANA_DIR}/mem001/an.2018-04-15_00.00.00.nc
  ...
factor: 0.8
```

Output directory. Should be the same as the analysis member.

List of variables to be inflated

List of all background members

List of all analysis members

RTPP inflation factor

Outline

- Ensemble data assimilation methods
- Ensemble of data assimilation (EDA)
- Local ensemble transform Kalman filter (LETKF)
- Comparison between EDA and LETKF

Local Ensemble Transform Kalman Filter (LETKF)

Background error covariance $\mathbf{P}^{\boldsymbol{b}} = \dfrac{1}{N-1}\mathbf{X}^b(\mathbf{X}^b)^{\mathrm{T}}$

$$\mathbf{x}_i^{\boldsymbol{b}} = \bar{\mathbf{x}}^b + \mathbf{X}_i^b$$

Analysis error covariance $\mathbf{P}^a = \dfrac{1}{N-1}\mathbf{X}^a(\mathbf{X}^a)^{\mathrm{T}} = \mathbf{X}^b\widetilde{\mathbf{P}}^a(\mathbf{X}^b)^{\mathrm{T}}$

$$\widetilde{\mathbf{P}}^a = [(N-1)\mathbf{I}/\rho + (\mathbf{Y}^b)^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{Y}^b)]^{-1}$$ **transform matrix**

$$\mathbf{Y}^b = \mathbf{H}\left(\mathbf{X}^b\right) \approx H(\mathbf{x}^b) - \bar{\mathbf{y}}^b$$ **obs-space ens. perturbation**

Ensemble mean updating $\bar{\mathbf{x}}^a = \bar{\mathbf{x}}^b + \mathbf{X}^b\widetilde{\mathbf{P}}^a(\mathbf{Y}^b)^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}^o - \bar{\mathbf{y}}^b) = \bar{\mathbf{x}}^b + \mathbf{X}^b\overline{\mathbf{w}}^a$

Ens. Perturbation updating $\mathbf{X}^a = \mathbf{X}^b[(N-1)\widetilde{\mathbf{P}}^a]^{\frac{1}{2}} = \mathbf{X}^b\mathbf{W}^a$ **weighting vector**

$$\mathbf{W}^a = \mathbf{U}\mathbf{S}^{\frac{1}{2}}\mathbf{U}^{\mathrm{T}}$$ **Singular vector decomposition**

To update analysis states at every grid point, the LETKF assimilates only **local observations** within a certain distance from each grid point.

Hunt et al. 2007; Frolov 2023

Configure and Run LETKF

Increment variables, background, and output section:

```
increment variables: ${an_variables}

background:
  members from template:
    template:
      date: &analysisDate YYYY-MM-DDTHH:MN:SSZ
      state variables: [${state_variables}]
      stream name: background
      filename: ${bg_dir}/mem%iMember%/${bg_file}
    pattern: %iMember% # 001, 002, …, 020
    start: 1
    zero padding: 3
    nmembers: 20 # Number of ensemble

output:
  filename: ${an_dir}/mem%{member}%/${an_file}
  stream name: analysis
```

MPAS-JEDI will overwrite analysis variables in ${an_file}, so we need to copy ${bg_file} to ${an_file} before running LETKF.

Configure and Run LETKF

Observation space localization:

The Observation section in JEDI are similar to that in variational DA except for the observation space localization configurations.

Horizontal localization

obs localizations:

localization method: ***Horizontal Gaspari-Cohn***/ *SOAR/ Box car*

lengthscale: ***${horizontal localization scale}***

search method: ***kd_tree***/ *brute_force*

distance type: ***geodesic***/ *cartesian*

max nobs: *maximum obs umber for localization*

…

Vertical localization

obs localization:

localization method: ***Vertical localization***

vertical lengthscale: ***${vertical localization scale}***

apply log transformation: *commonly used for pressure*

ioda vertical coordinate: *height/pressure/…*

ioda vertical coordinate group: *MetaData*

localization function: *Box Car/ Gaspari Cohn/ SOAR*

…

Configure and Run LETKF

Local ensemble DA section:

This section relates to the local ensemble DA methods and the variance inflation schemes.

local ensemble DA:

solver: LETKF/ GETKF

LETKF: *Vertical localization is done in the observation space;*

GETKF: *using modulated ensembles to emulate model-space vertical localization.*

Variance inflation

mult: prior multiplicative inflation $\mathbf{P}^{b'} = \alpha \mathbf{P}^{b}$

rtpp: post relaxation to prior perturbation $\mathbf{X}_i^{a'} = \alpha \mathbf{X}_i^{a} + (1-\alpha)\mathbf{X}_i^{b}$

rtps: post relaxation to prior spread $\mathbf{X}_i^{a'} = \mathbf{X}_i^{a}(1 + \alpha \frac{\sigma_b - \sigma_a}{\sigma_a})$

Configure and Run LETKF

LETKF analysis procedure can be divided into three steps: **Observer**, **Solver**, **and DiagOMA**

Observer

driver:
    run as observer only: true
    update obs config with geometry info: false

This step will only calculate the HofX of all members and ensemble mean and then write them out;

Quality control of LETKF is done based on the ensemble mean states.

In this step, the *observation distribution* can be set to *RoundRobin* to be more efficient.

Solver

driver:
    read HX from disk: true
    do posterior observer: false
    save posterior ensemble: true
    save posterior mean: true

In this step, if "*read HX from disk*" is set to true, it will read the HofX of all members and ensemble mean from the **Observer** step, and then run LETKF solver;

The *obsdatain* should be changed to the *obsdataout* that is used in Observer step.

The *observation distribution* should be set to *Halo.*

Configure and Run LETKF

LETKF analysis procedure can be divided into three steps: **Observer**, **Solver**, and **DiagOMA**

Observer

```
background:
  members from template:
     template: <<: *memberConfig
     filename: ../../bg/mem%iMember%/bg.2018-04-15_00.00.00.nc
    pattern: %iMember%
    start: 1
    zero padding: 3
    nmembers: 20 _obs

_obsdatain &ObsDataIn
    engine:
       type: H5File
       obsfile: ../../dbIn/sfc_obs_2018041500.h5
_obsdataout: &ObsDataOut
     engine:
        type: H5File
        obsfile: ../../dbOut/obsout_da_sfc.h5
```

DiagOMA

```
background:
  members from template:
     template: <<: *memberConfig
     filename: ../../an/mem%iMember%/an.2018-04-15_00.00.00.nc
    pattern: %iMember%
    start: 1
    zero padding: 3
    nmembers: 20

_obsdatain &ObsDataIn
    engine:
       type: H5File
       obsfile: ../../dbIn/sfc_obs_2018041500.h5
_obsdataout: &ObsDataOut
     engine:
        type: H5File
        obsfile: ../../dbAna/obsout_da_sfc.h5
```

Outline

- Ensemble data assimilation methods
- Ensemble of data assimilation (EDA)
- Local ensemble transform Kalman filter (LETKF)
- Comparison between EDA and LETKF

Comparison between EDA and KETKF

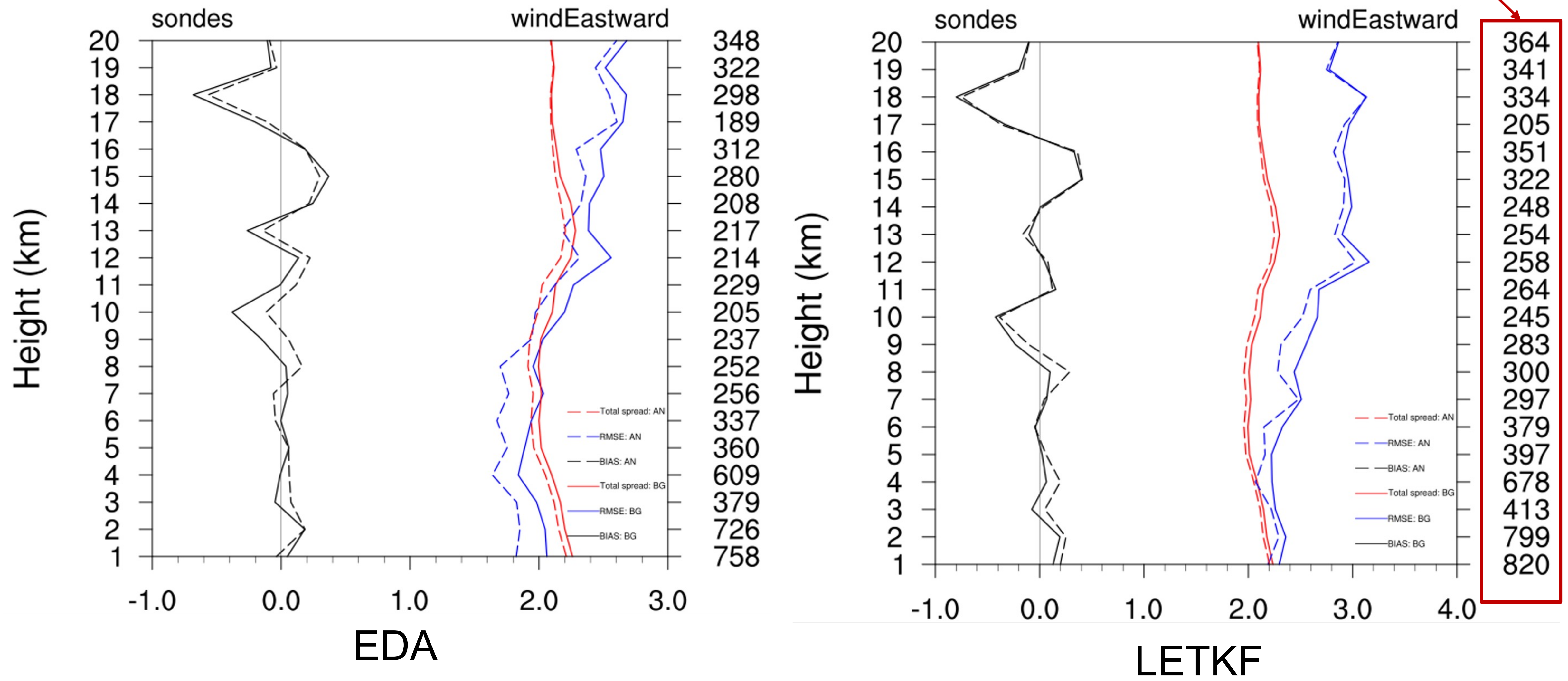
General comparison

DA methods	EDA	LETKF
Algorithm	Based on the variational framework	An EnKF-based method
Uncertainty	use perturbated observations	No need to perturb observations
Localization	Model space localization	Observation space localization
Inflation	External posterior inflation	Interior prior/posterior inflation
Computational cost	All members can be updated in parallel	All members are updated simultaneously

Comparison between EDA and KETKF

Vertical profiles of total spread, RMSE and BIAS

More observations are assimilated in LETKF



References

- Guerrette, J. J., et al., 2023: Data assimilation for the Model for Prediction Across Scales – Atmosphere with the Joint Effort for Data assimilation Integration (JEDI-MPAS 2.0.0-beta): ensemble of 3D ensemble-variational (En-3DEnVar) assimilations, Geosci. Model Dev.
- Hunt B.R., Kostelich E.J., Szunyogh I., 2007: Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter, Physica D: Nonlinear Phenomena, 230, 112-126, https://doi.org/10.1016/j.physd.2006.11.008.
- Sergey Frolov, Anna Shlyaeva, Wei Huang, et al., 2023: Local volume solvers for Earth system data assimilation: implementation in the framework for Joint Effort for Data Assimilation Integration, Journal of Advances in Modeling Earth Systems, under review.
- https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/latest/inside/jedi-components/oops/applications/localensembleda.html

This material is based upon work supported by the NSF National Center for Atmospheric Research, which is a major facility operated by the U.S. National Science Foundation under Cooperative Agreement No. 1852977.

MPAS-Workflow and graphics package: an overview

Ivette Hernández Baños
Prediction, Assimilation, and Risk Communication Section
Mesoscale & Microscale Meteorology Laboratory
National Center for Atmospheric Research

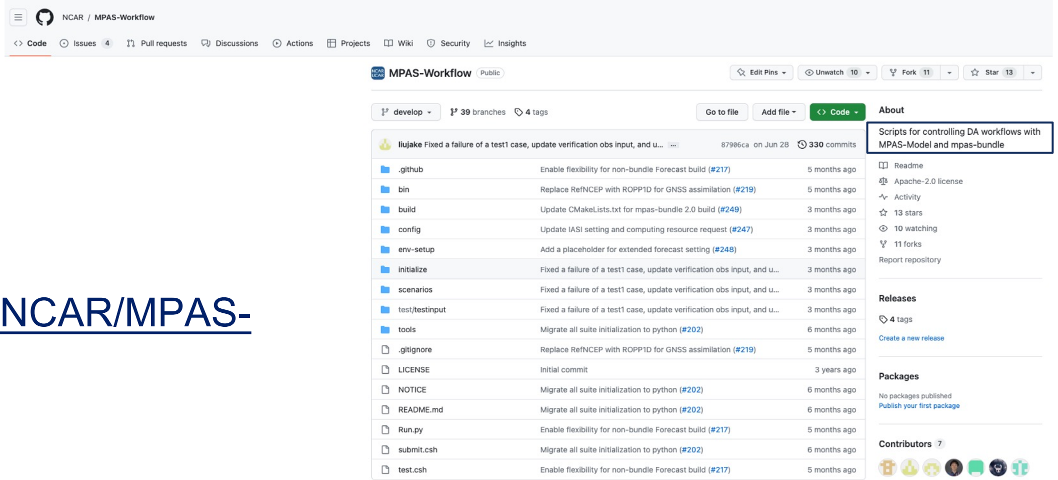
MONAN: INPE MPAS-JEDI Training 2024, Cachoeira Paulista, São Paulo, Brazil
August 15-16, 2024

Outline

1. MPAS-Workflow
    a. Applications
    b. Data
    c. Post-processing
    d. Scenario YAMLS
    e. Predefined tests
    f. Tips
2. Graphics package
    a. Functionalities
    b. Examples

MPAS-Workflow

- ➢ Developed at NSF NCAR/MMM to aid cycling experiments with MPAS and MPAS-JEDI
  - ○ Tailored for the PANDAC specific use
  - ○ last version: 2.1.0
- ➢ CYLC-based workflow manager (v8.2.2) + Python + C-Shell scripts
- ➢ Currently, only operates on NSF NCAR's Derecho HPC



- ➢ Open-source: https://github.com/NCAR/MPAS-Workflow

but **NOT** supported

MPAS-Workflow

- ❑ constructs each JEDI application YAML, with high flexibility for a number of configurations
    - ❑ e.g., do variational bias correction or not, SST and XICE update, number of outer loops, number of ensemble members, observers, etc.
- ❑ links all necessary input data
- ❑ can be used for cycling and no cycling experiments
    - ❑ e.g., generate observations, generate GFS analyses in MPAS ICs format, generate free forecast from GFS analyses
- ❑ can handle cold and warm start
- ❑ constructs and submit the CYLC suite for the cycling (and no cycling) experiment
- ❑ can be use to run real-time experiments with 3DVar data assimilation

MPAS-Workflow: applications

- **Data assimilation**
  - Variational
    - 3DVar
    - 3/4DEnVar
    - Hybrid
    - EDA
      - Application specific YAML templates: `config/jedi/applications/*.yaml`
      - Task script: `bin/Variational.csh` `bin/EnsembleOfVariational.csh`
  - EnKF
    - LETKF
    - GETKF
      - Application YAML template: `config/jedi/applications/enkf.yaml`
      - Task script: `bin/EnKF.csh` `bin/EnKFObserver.csh`
  - RTPP
    - Application YAML template: `config/jedi/applications/rtpp.yaml`
    - Task script: `bin/RTPP.csh`
  - HofX
    - Application YAML template: `config/jedi/applications/hofx.yaml`
    - Task script: `bin/HofX.csh`

- Observation YAML stubs:
  ```
  config/jedi/ObsPlugs/da
  ├── base/*.yaml
  ├── bias/*.yaml
  ├── filters/*.yaml
  └── filtersWithBias/*.yaml
  ```
- YAMLS are further populated by `bin/PrepJEDI.csh` and/or `bin/InitVariationals.csh` `bin/InitEnKF.csh`

- Initialization scripts (`initialize/`) set up the suite:
  1. Read/parse scenario YAML
  2. Define c-shell environment variables for use by task scripts in the run directory
  3. Define inter-task and inter-cycle dependencies
  4. Submit Cylc suite

MPAS-Workflow: applications

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

YAML file

How we set the YAML file?

MPAS-Workflow: applications

Data assimilation:

- 3denvar.yaml

<u>Configurable options:</u>

`InnerNamelistFile`, `InnerStreamsFile`, `thisISO8601Date`, `AnalysisVariables`, `VariationalMinimizer`, `VariationalIterations`, `StateVariables`, `EnsemblePbMembers`, `Observers`, …

```yaml
_iteration: &iterationConfig
iteration: &iterationConfig
  geometry:
    nml_file: {{InnerNamelistFile}}
    streams_file: {{InnerStreamsFile}}{{StreamsFileMember}}
    deallocate non-da fields: true
    interpolation type: unstructured
  gradient norm reduction: 1e-3
_member: &memberConfig
  date: &analysisDate {{thisISO8601Date}}
  state variables: &incvars [{{AnalysisVariables}}]
  stream name: ensemble
output:
  filename: {{anStateDir}}{{MemberDir}}/{{anStatePrefix}}.$Y-$M-$D_$h.$m.$s.nc
  stream name: analysis
variational:
  minimizer:
{{VariationalMinimizer}}
  iterations:
{{VariationalIterations}}
final:
  diagnostics:
    departures: oman
cost function:
  cost type: 3D-Var
  window begin: {{windowBegin}}
  window length: {{windowLength}}
  jb evaluation: false
  geometry:
    nml_file: {{OuterNamelistFile}}
    streams_file: {{OuterStreamsFile}}{{StreamsFileMember}}
    deallocate non-da fields: true
    interpolation type: unstructured
  analysis variables: *incvars
  background:
    state variables: [{{StateVariables}}]
    filename: {{bgStateDir}}{{MemberDir}}/{{bgStatePrefix}}.{{thisMPASFileDate}}.nc
    date: *analysisDate
  background error:
    covariance model: ensemble
    localization:
      localization method: SABER
      saber central block:
        saber block name: BUMP_NICAS
        active variables: *incvars
        read:
          io:
            data directory: {{bumpLocDir}}
            files prefix: {{bumpLocPrefix}}
          drivers:
            multivariate strategy: duplicated
            read local nicas: true
          model:
            level for 2d variables: last
{{EnsemblePbMembers}}
{{EnsemblePbInflation}}
  observations:
    obs perturbations: {{ObsPerturbations}}
    observers:
{{Observers}}
```

MPAS-Workflow: applications

Data assimilation:

- enkf.yaml

Configurable options:

`driver`, `thisISO8601Date`, `AnalysisVariables`, `EnKFNamelistFile`, `EnKFStreamsFile`, `StateVariables`, `EnsembleMembers`, `localEnsembleDASolver`, `verticalLocalizationLengthscale`, …

```yaml
_member: &memberConfig
  date: &analysisDate {{thisISO8601Date}}
  state variables: [{{StateVariables}}]
  stream name: background

_as observer: &asObserver
  run as observer only: true
  update obs config with geometry info: false

_as solver: &asSolver
  read HX from disk: true
  do posterior observer: false
  save posterior ensemble: true
  save posterior mean: true

_letkf geometry: &3DLETKFGeometry
  iterator dimension: 3

_letkf geometry: &2DLETKFGeometry
  iterator dimension: 2

_lgetkf geometry: &3DGETKFGeometry
  iterator dimension: 2

geometry:
  <<: *{{localizationDimension}}{{localEnsembleDASolver}}Geometry
  nml_file: {{EnKFNamelistFile}}
  streams_file: {{EnKFStreamsFile}}
  deallocate non-da fields: true

window begin: {{windowBegin}}
window length: {{windowLength}}

background:
{{EnsembleMembers}}

increment variables: [{{AnalysisVariables}}]

observations:
  observers:
{{Observers}}

driver: *{{driver}}

local ensemble DA:
  solver: {{localEnsembleDASolver}}
  vertical localization:
    fraction of retained variance: 0.95
    lengthscale: {{verticalLocalizationLengthscale}}
    lengthscale units: modellevel

output:
  filename: {{anStateDir}}/mem%{member}%/{{anStatePrefix}}.$Y-$M-$D_$h.$m.$s.nc
  stream name: analysis
```

MPAS-Workflow: applications

Data assimilation:

- Observers: e.g., amsua_n15

aircraft, sondes, sfc, satwind, satwnd, gnssro ⇒ base + filters
amsua, mhs ⇒ base + filters or base + bias + filtersWithBias

base → **bias** → **filtersWithBias**

```
- obs space:
    <<: *ObsSpace
    name: amsua_n15
    _obsdatain: &ObsDataIn
      engine:
        type: H5File
        obsfile: {{InDBDir}}/amsua_n15_obs_{{thisValidDate}}.h5
    _obsdataout: &ObsDataOut
      engine:
        type: H5File
        obsfile:
{{OutDBDir}}{{MemberDir}}/{{obsPrefix}}_amsua_n15{{ObsOut
Suffix}}.h5
    obsdatain: *{{ObsDataIn}}
    {{ObsDataOut}}
    simulated variables: [brightnessTemperature]
    channels: &amsua_n15_channels 1-15
  obs error: *ObsErrorDiagonal
  <<: *horizObsLoc
  obs operator:
    <<: *clearCRTMObsOperator
    obs options:
      <<: *CRTMObsOptions
      Sensor_ID: amsua_n15
  get values:
    <<: *GetValues
```

```
obs bias:
  input file: {{biasCorrectionDir}}/satbias_amsua_n15.h5
  output file: {{OutDBDir}}{{MemberDir}}/satbias_amsua_n15.h5
  variational bc:
    predictors: &predictors2
    - name: constant
    - name: lapse_rate
      order: 2
      tlapse: &amsua15tlap {{fixedTlapmeanCov}}/amsua_n15_tlapmean.txt
    - name: lapse_rate
      tlapse: *amsua15tlap
    - name: emissivity
    - name: scan_angle
      order: 4
    - name: scan_angle
      order: 3
    - name: scan_angle
      order: 2
    - name: scan_angle
  covariance:
    minimal required obs number: 20
    variance range: [1.0e-6, 10.]
    step size: 1.0e-4
    largest analysis variance: 10000.0
    prior:
      input file: {{biasCorrectionDir}}/satbias_cov_amsua_n15.h5
      inflation:
        ratio: 1.1
        ratio for small dataset: 2.0
    output file: {{OutDBDir}}{{MemberDir}}/satbias_cov_amsua_n15.h5
```

```
obs filters:
- filter: Domain Check
  where:
  - variable:
      name: MetaData/sensorZenithAngle
    maxvalue: 45.0
# CLW Retrieval Check
- filter: Bounds Check
  filter variables:
  - name: brightnessTemperature
    channels: 1-6, 15
  test variables:
  - name: ObsFunction/CLWRetMW
    options:
      clwret_ch238: 1
      clwret_ch314: 2
      clwret_types: [ObsValue]
  maxvalue: 999.0
  action:
    name: reject
```

Functions in filters see: https://jointcenterforsatellitedataassimilation-jedi-docs.readthedocs-hosted.com/en/stable/index.html

MPAS-Workflow: data

```
initialize/data
├── DataList.py
├── ExternalAnalyses.py
├── FirstBackground.py
├── Model.py
├── ObsEnsemble.py
├── Observations.py
├── StateEnsemble.py
└── StaticStream.py
```

```
benchmarkObservations = [
  # anchor
  'aircraft',
  'gnssrobndropp1d',
  'satwind',
  'satwnd',
  'sfc',
  'sondes',
  # MW satellite-based
  'amsua_aqua',
  'amsua_metop-a',
  'amsua_metop-b',
  'amsua_n15',
  'amsua_n18',
  'amsua_n19',
  'mhs_metop-a',
  'mhs_metop-b',
  'mhs_n18',
  'mhs_n19',
]
```

defaults = 'scenarios/defaults/observations.yaml'

- resources:
  - NCEPFTPOnline
  - GladeRDAOnline
  - PANDACArchive
  - PANDACArchiveForVarBC
  - GenerateObs

Other resources can be added as needed

outerMesh`, `innerMesh`, `ensembleMesh`, `GraphInfoDir`

MPAS-Workflow: Post-processing

- ❑ Verify vs. GFS analyses: VerifyModel
  - ➢ Inputs: MPAS forecast and GFS analyses on MPAS format
- ❑ Verify vs. observations: VerifyObs
  - ➢ Inputs: HofX or DA observation feedback files:
    - ■ DA: omb/oma obsout diagnostics (same assimilated observations)
    - ■ model on observations space: HofX obsout diagnostics + VerifyObs (instantiates it own HofX)

Observers

- **Sondes**, **aircraft**, **satellite-derived 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) and **AHI** (Himawari-8)

MPAS-Workflow: Post-processing/applications

HofX:

- `bin/HofX.csh`: Carries out multiple observation operators (“h(x)”) on 1 or more MPAS-Atmosphere forecasts

Input:

- state (single or ensemble members) ⇒ previously generated
- static files
- lookup tables
- mesh graph info
- namelist and streams files
- mpasjedi_hofx3d.x executable
- geovars.yaml
- observations in `/dbIn` folder (observers specified in `initialize/applications/HofX.py`)

Standalone application used to verify MPAS 6-hr forecasts on observation space
Facilitates verifying independent observations

YAML: hofx.yaml

MPAS-Workflow: scenarios

- Configuration for a particular instance of an MPAS-Workflow Cylc suite
- Nested key-value parameters that users can specify for their particular needs
- Include default YAMLS that describe options that users may select, such as the observations resource, the first background, etc…

`scenarios/defaults/*.yaml`

source env-script/machine.${YourShell}

Running:

./Run.py ./scenarios/{{scenario}}.yaml

OR

./Run.py ./test/testinput/{{scenario}}.yaml

MPAS-Workflow: scenarios

Top scenario YAML file containing most possible user configurable variables:

scenarios/3dhybrid_O30kmIE60km_SpecifiedEnsemble_VarBC_**allConfig**.yaml

```yaml
suite: Cycle

experiment:
  user directory child: pandac
  suffix: '_ensB-SE80+RTPP70_VarBC_allConfig_TEST'
  #prefix: ''
  #name: ''

build:
  mpas bundle: /glade/campaign/mmm/parc/ivette/pandac/codeBuild/mpasBundle_saca_dev_10Jun2024/build_SP
  #forecast directory: ''
  bundle compiler used: gnu-openmpi

hpc:
  CriticalAccount: NMMM0015
  CriticalQueue: main
  NonCriticalAccount: NMMM0015
  NonCriticalQueue: economy
  SingleProcAccount: NMMM0015
  SingleProcQueue: casper@casper-pbs
```

```yaml
    GraphInfoDir: /glade/campaign/mmm/parc/liuz/pandac_common/static_from_duda
    precision: single
    MPThompsonTablesDir: /glade/campaign/mmm/parc/ivette/pandac/saca/thompson_tables

  staticstream:
    resource: "PANDAC"
    resources:
    PANDAC:
        60km: # only available 20180414T18, 20200723T18
          directory: /glade/campaign/mmm/parc/liuz/pandac_common/fixed_input/GEFS/init/000hr/{{FirstCycleDate}}
          maxMembers: 80
          memberFormat: /{:02d}

  externalanalyses:
    resource: "GFS.PANDAC"
    resources:
      GFS:
        PANDAC: # only available 20180418T00-20180524T00
          30km:
            directory: /glade/campaign/mmm/parc/liuz/pandac_common/30km/30km_GFSANA
          60km:
            directory: /glade/campaign/mmm/parc/liuz/pandac_common/60km/60km_GFSANA
```

⇒ can help us to better locate the default paths and variable values we are using in each section for this experiment

MPAS-Workflow: predefined tests

`/test/testinput`

Pre-defined scenarios that exercise functionality in the workflow

(WarmStart == offline 1st state; ColdStart == online 1st state)

`test1.yaml`

scenarios: [
  3denvar_O30kmIE60km_WarmStart.yaml
  3denvar_OIE120km_IAU_WarmStart.yaml
  3dvar_O30kmIE60km_ColdStart.yaml
  3dvar_OIE120km_ColdStart.yaml
  3dvar_OIE120km_WarmStart_PostProcess.yaml
  3dvar_OIE120km_WarmStart.yaml
  eda_OIE120km_WarmStart.yaml
  ForecastFromGFSAnalysesMPT.yaml
  getkf_OIE120km_WarmStart.yaml
  letkf_OIE120km_WarmStart.yaml]

Run:

./test.csh

./Run.py test/testinput/test1.yaml

or

./Run.py test/testinput/test2.yaml

MPAS-Workflow: tips

For debugging, you have a couple of ways to check what is happening:

1. the CYLC gui interface will tell you the status of each job
2. check if the job is actually submitted by issuing 'qstat -u $USER'
3. check the log file of the application that seems to be submitted/failed/etc
    a. e.g., HofX or DA: you can check the jedi.log/jedi.log.all files in the cycle date)
4. check the CYLC log file in the cylc-run directory (/glade/scratch/<username>/cylc-run)

Useful CYLC line commands:

cylc scan

cylc trigger suitename "*.*:failed"

cylc restart   --until=final_end_point suitename | add restart point in the scenario and run it

cylc reset -s succeeded suitename *:failed

How to run 6hr cycling, assuming all dependencies have been prepared in advance?

MPAS-Workflow

```
./Run.py scenarios/3dvar_OIE120km_WarmStart.yaml
```

WarmStart

Pre-generated first forecast for 6hr data assimilation cycling

Long term forecasts (aka. extended forecasts) with cycling analyses

Default is post-processing
To turn it off:

```
forecast
    post: []
variational:
          post: []
```

Already generated/archived observations in IODA format

```
experiment:
  name: '3dvar_OIE120km_WarmStart_TEST'
externalanalyses:
  resource: "GFS.PANDAC"
firstbackground:
  resource: "PANDAC.GFS"
forecast:
  # turn off post to reduce overhead
  post: □
hpc:
  CriticalQueue: economy
  NonCriticalQueue: economy
members:
  n: 1
model:
  outerMesh: 120km
  innerMesh: 120km
  ensembleMesh: 120km
observations:
  resource: PANDACArchive
variational:
  DAType: 3dvar
  nInnerIterations: [15,]
  # turn off post to reduce overhead
  post: □
workflow:
  first cycle point: 20180414T18
  final cycle point: 20180415T06
```

MPAS-Workflow

YAML configuration for **extended forecasts (larger than 6hr):**

```
extendedforecast:
  meanTimes: T00,T06,T12,T18
  lengthHR: 240
  outIntervalHR: 12
  post: [verifyobs, verifymodel]
forecast:
  execute: False
  post: [ ]
variational:
  execute: False
  post: [ ]
```

```
./Run.py
scenarios/3dvar_OIE120km_WarmStart.yaml
```

Triggers HofX application

***Cycling analyses and 6hr forecast for cycling DA won't be executed**

MPAS-Workflow

Experiment folders structure: ivette_3dvar_OIE120km_WarmStart

```
├── CyclingDA
│   ├── 2018041500
│   ├── …
│   └── 2018041600
├── CyclingFC
│   ├── 2018041418
│   ├── …
│   └── 2018041600
├── ExtendedFC
│   ├── 2018041500
│   ├── …
│   ├── …
│   └── 2018041600
├── ExternalAnalyses
│   ├── 120km
├── MPAS-Workflow
│   ├── bin
│   ├── config
│   ├── scenarios
│   ├── test
│   └── tools
└── Verification
    ├── bg
    ├── da
    └── fc
```

- CyclingDA: for cycling DA (analysis + 6-h fcst) → Executes mpasjedi_variational.x with corresponding yaml file
- CyclingFC: → Executes mpas_atmosphere for 6-h with corresponding namelist and stream files
- ExtendedFC: 240-h deterministic forecast → Executes mpas_atmosphere for 240-h with corresponding namelist and stream files
- ExternalAnalyses: For the SST and XICE update at each cycle, and verification of the MPAS 6-h fcst → Link of external GFS analyses converted to MPAS initial conditions format and grid
- MPAS-Workflow: Local copy of the workflow with corresponding changes for this experiment
- Verification: For the MPAS 6-h fcst verification against observation and GFS analyses, and extended fcst against GFS analyses → Runs HofX application and diagnostic scripts

How to port the workflow for your own machine?

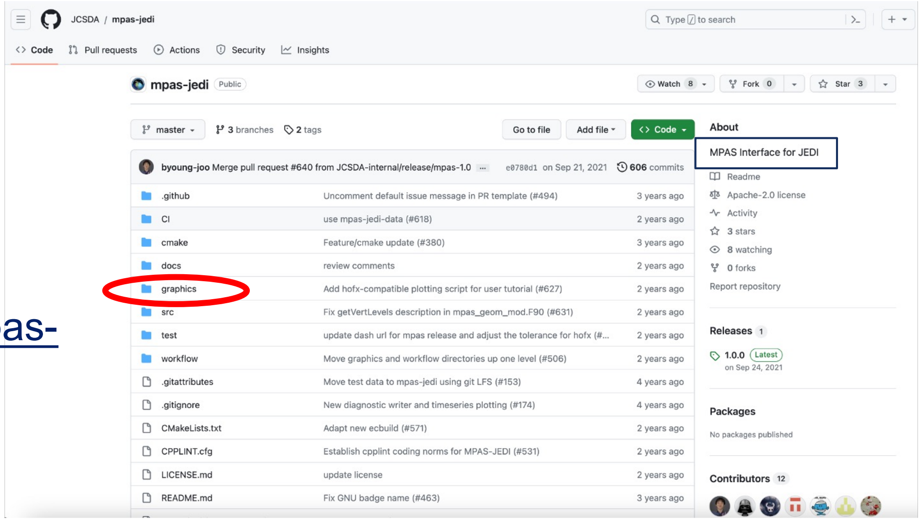
Porting MPAS-Workflow

1. ***Install spack-stack and compile mpas-bundle***
2. Install CYLC v8.2.2 (or v7.3 for older versions) (and needed Python)
3. Prepare your corresponding machine.${YourShell} with needed environment modules
4. Clone the MPAS-Workflow
5. Copy your machine.${YourShell} under env-script folder
6. Copy all necessary files to run experiments (mesh, static files, B and localization files, 1st background, ensemble forecasts, observations, external analyses on MPAS mesh for verification) to your machine
7. Set up paths for files location (check `scenarios/3dhybrid_O30kmIE60km_SpecifiedEnsemble_VarBC_`**`allConfig`**`.yaml` for variables to update)
8. Create new scenario YAML ⇒ make copy of existing scenario and taylor it for your own experiment
9. Execute Run.py with your new scenario

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 NCAR’s Cheyenne HPC

➢ Open-source: https://github.com/JCSDA/mpas-jedi/tree/release/2.0.0/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

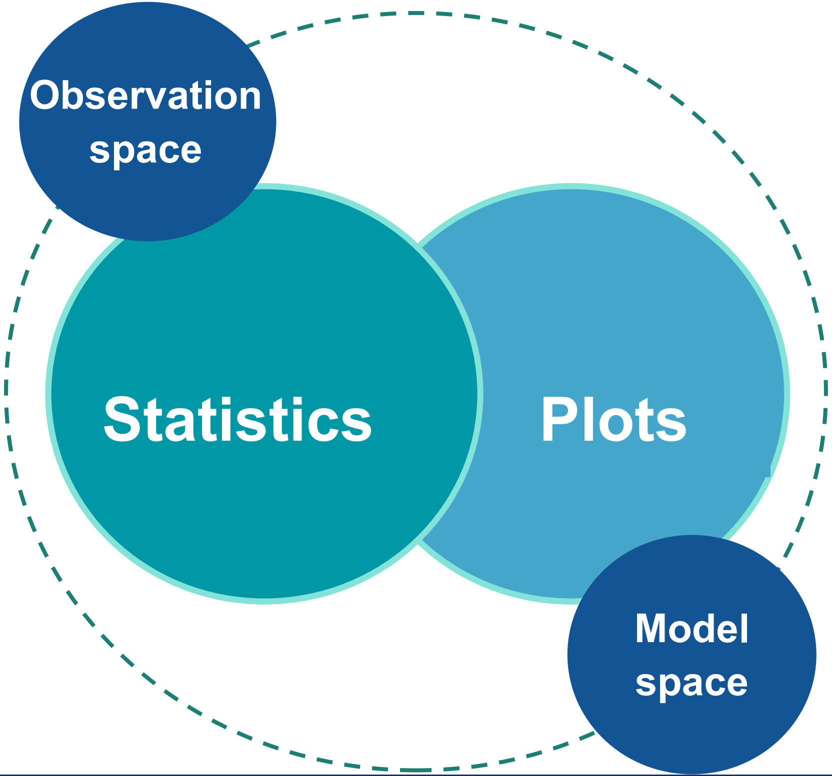
AMSU-A (NOAA-15, NOAA-18, NOAA-19, METOP-A, METOP-B)

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

Statistics

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** YYYYMMMDDDHHH -n 36 -r ./x1.$\underbrace{655362}_{30km}$.init

Graphics package: examples

Statistics

Experiment folders structure: ivette_3dvar_OIE120km_WarmStart

```
└── Verification
    ├── bg
    │   ├── mean/YYYYMMMDDDHHH/diagnostic_stats
    ├── da
    │   ├── mean/YYYYMMMDDDHHH/diagnostic_stats
    ├── fc
    │   ├── mean/YYYYMMMDDDHHH/diagnostic_stats
```

```
├── 0hr
├── 120hr
├── 24hr
├── 48hr
├── 72hr
└── 96hr
    ├── diag.2018-05-16_00.00.00.nc ->
    ├── model
    ├── obs
    └── restart.2018-05-16_00.00.00.nc ->
```

```
└── obs
    ├── stats_da_aircraft.h5
    ├── stats_da_amsua_aqua.h5
    ├── stats_da_amsua_metop-a.h5
    ├── stats_da_amsua_metop-b.h5
    ├── stats_da_amsua_n15.h5
    ├── stats_da_amsua_n18.h5
    ├── stats_da_amsua_n19.h5
    ├── stats_da_gnssrobndropp1d.h5
    ├── stats_da_mhs_metop-a.h5
    ├── stats_da_mhs_metop-b.h5
    ├── stats_da_mhs_n18.h5
    ├── stats_da_mhs_n19.h5
    ├── stats_da_satwind.h5
    ├── stats_da_satwnd.h5
    ├── stats_da_sfc.h5
    ├── stats_da_sondes.h5
```

```
└── model
    ├── stats_mpas.h5
```

Graphics package: functionalities

Plots

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,gnssro,satw

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

Model space (vs GFS analysis):

./**SpawnAnalyzeStats.py** -d mpas

Plots

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)
```

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'
```

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+RTPP70_VarBC_RefNCEP_2ndDoaDob' + \
  deterministicVerifyDir
```

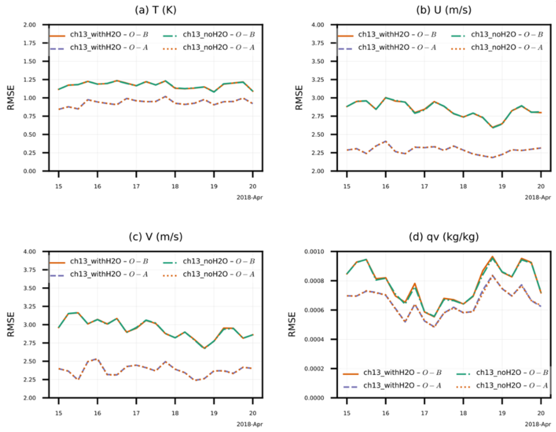
Graphics package: examples

Plots

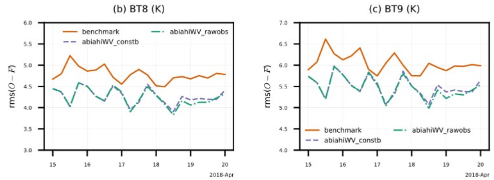
Observation space

DiagSpace_analyses
- BinValAxes2D
- BinValAxisProfileDiffCI
- CYandBinValAxes2D
- CYAxisExpLines

aircraft: OmA/OmB



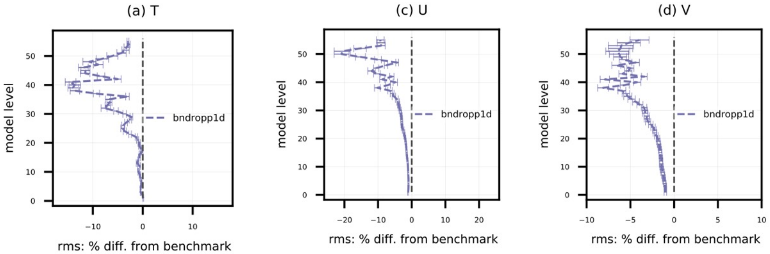
ABI: OmB (HofX)

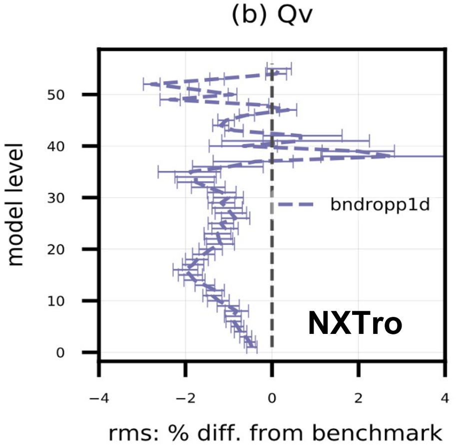


Model space

mpas_analyses
- BinValAxes2D
- BinValAxisProfileDiffCI
- CYandBinValAxes2D
- CYAxisExpLines

MPAS 6-h verification vs GFS analysis



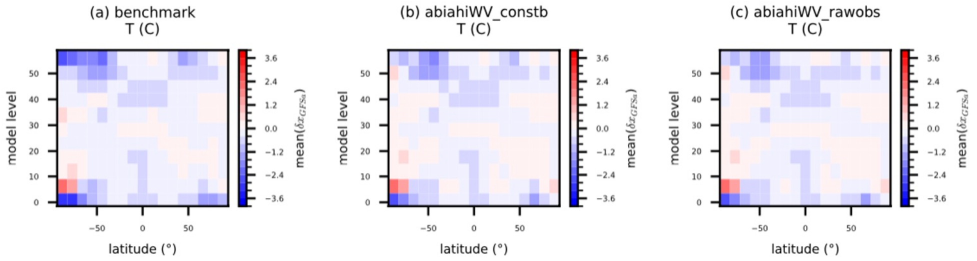


Plots

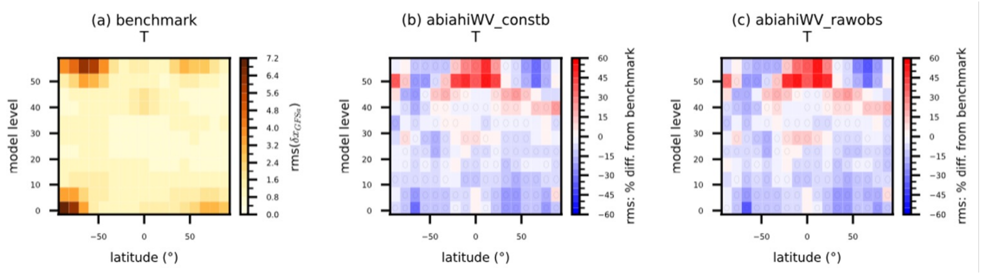
Graphics package: examples

MPAS 6-h verification vs GFS analysis

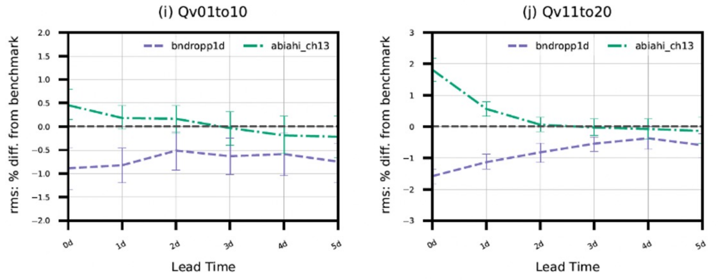
BIAS

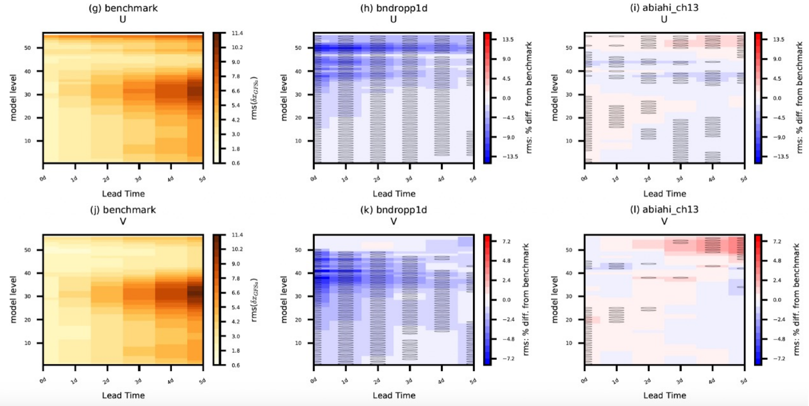


RMSE



MPAS 5-days verification vs GFS analysis





Contributions for new diagnostics/capabilities are welcome!!!

https://github.com/JCSDA/mpas-jedi/tree/develop/graphics

Regional MPAS-JEDI

Jake Liu

Prediction, Assimilation, and Risk Communication Section
Mesoscale & Microscale Meteorology Laboratory
National Center for Atmospheric Research

NCAR

MPAS-JEDI Tutorial, INPE, 15-16 August, 2024

NSF

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

What are differences from global MPAS-JEDI?

1. namelist.atmosphere

```
&limited_area
    config_apply_lbcs = true
/
```

2. streams.atmosphere

```
<immutable_stream name="lbc_in"
                  type="input"
                  io_type="pnetcdf,cdf5"
                  filename_template="lbc.$Y-$M-$D_$h.$m.$s.nc"
                  filename_interval="input_interval"
                  packages="limited_area"
                  input_interval="3:00:00" />
```

You need to set this, but no need of LBC file.

3. 3denvar.yaml

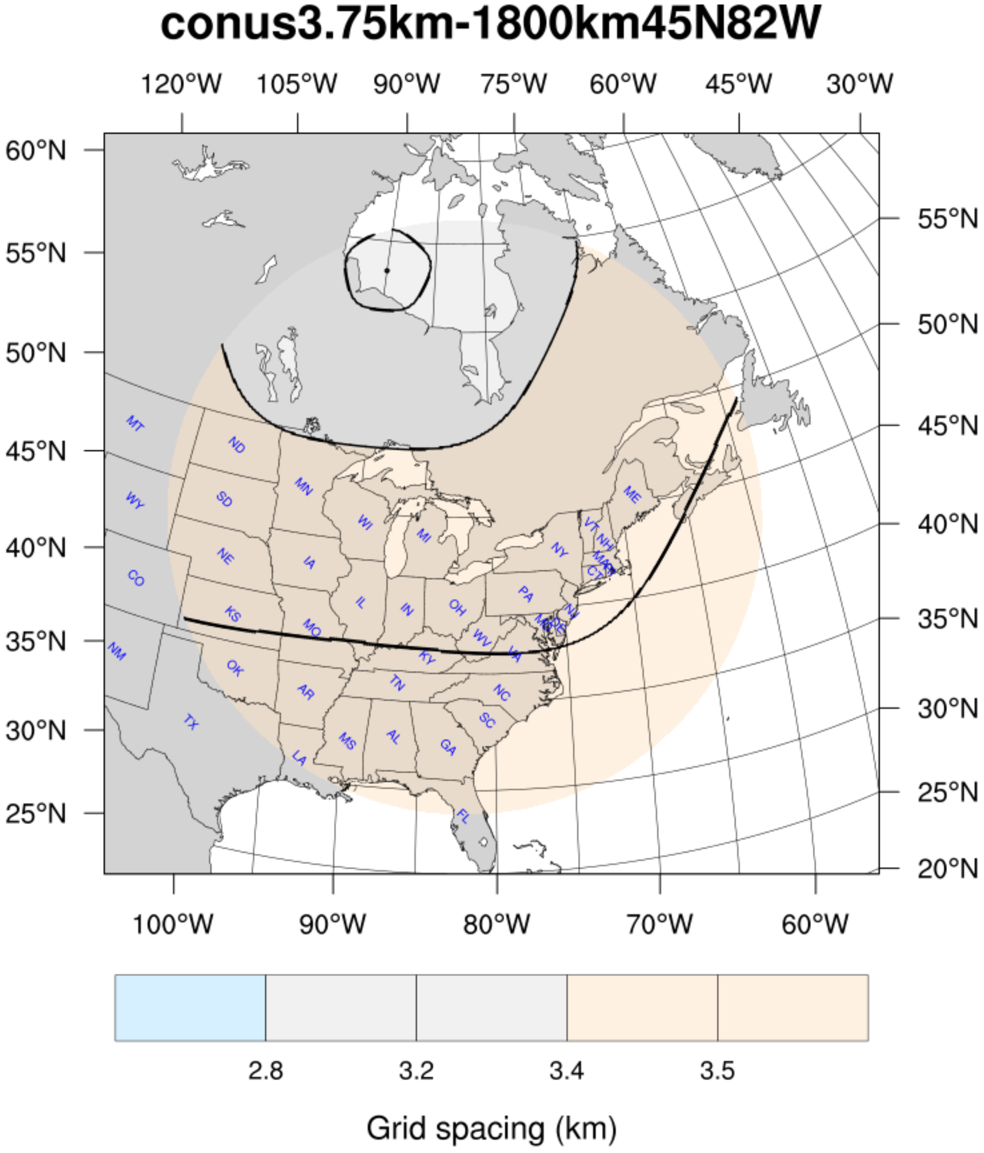
```
obs filters:
- filter: Bounds Check
  filter variables:
  - name: airTemperature
  - name: windEastward
  - name: windNorthward
  - name: specificHumidity
  test variables:
  - name:  LAMDomainCheck@ObsFunction
    options:
      map_projection: circle # an option
      save: true # will save the Derived
      cenlat: 40.0    # central lat
      cenlon: 260.0  # central lon
      radius: 2750.0  # km
  minvalue: 1.0 # will filter all obs ou
```

Reject obs outside a circular domain

Recent update about regional obs filtering

- Latest code has another more generic way to reject obs outside a regional domain of any shape

Regional **hybrid-3D/4DEnVar at 3.75km** over Eastern US



Ensemble B (weight 0.6): from 30-member ensemble input at 15km mesh from MPAS downscaled forecasts from GEFS ICs

Static B (weight 0.4): univariate, statistics from 960 downscaled 6-h ensemble forecasts

2-week period 6-hourly cycling: 7 – 18 July, 2023

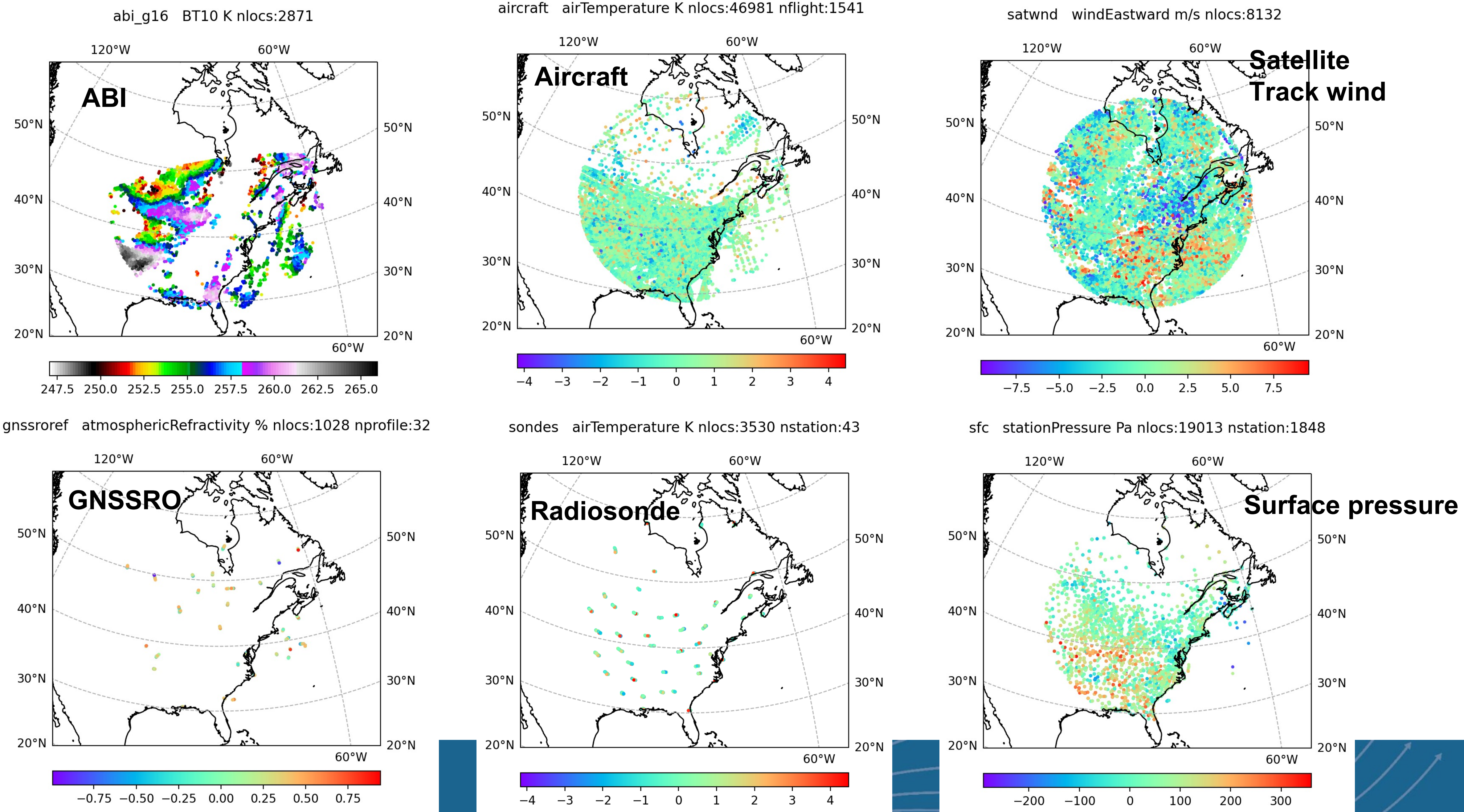
assimilates:

- T/Q/U/V from radiosonde
- T/Q/U/V from aircraft
- U/V from satellite track winds
- GNSSRO refractivity
- surface pressure
- +- 3-h time window

2 experiments:

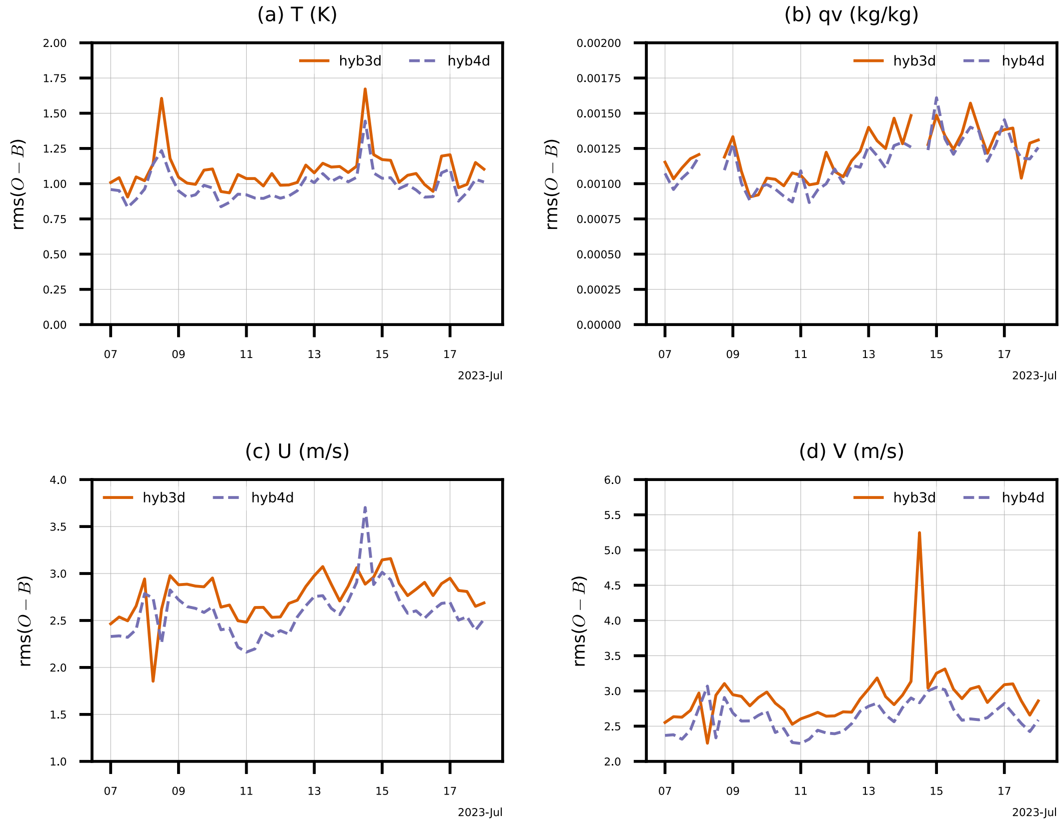
- Hybrid-3DEnVar
- Hybrid-4DEnVar

Obs coverage (all vertical levels together) at 2023070900

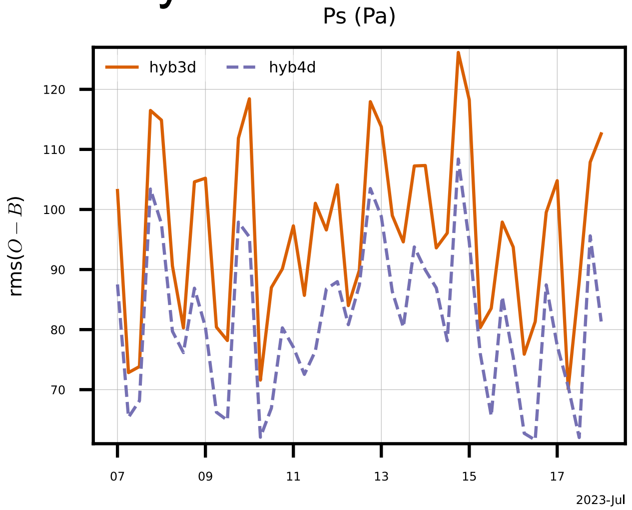


RMS of OMB: hybrid-3DEnVar vs. hybrid-4DEnVar

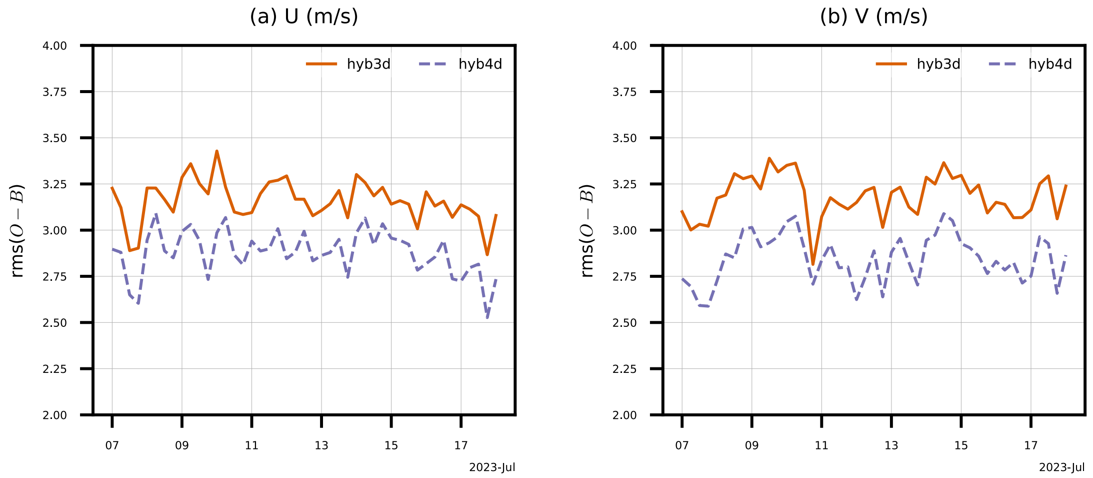
Clear better background-obs fitting from hybrid-4DEnVar



Aircraft
(spikes: times with missing obs)



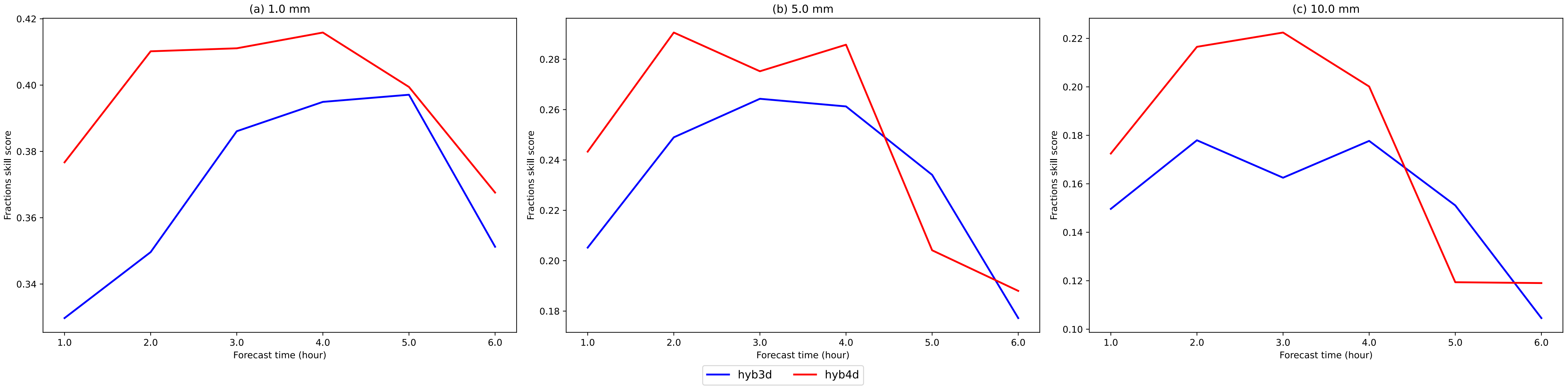
Surface Pressure



Satellite track winds

1-h accumulated rainfall forecast FSS scores: 1h - 6h lead time

Hybrid-3DEnVar vs. Hybrid-4DEnVar

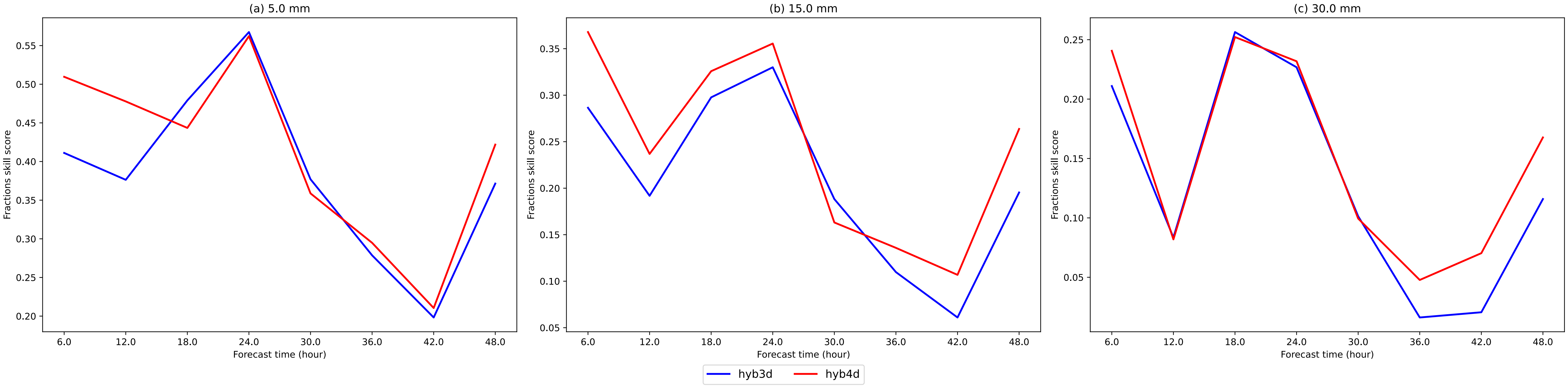


Clear improvement for the first several hours from hybrid-4DEnVar

Fraction Skill Scores (FSS) computed against Stage-IV obs with a radius of 25km, from 21 forecasts from 00 UTC 8 to 00 UTC 13 July.

6-h accumulated rainfall forecast FSS scores: up to 48-h lead time

Hybrid-3DEnVar vs. Hybrid-4DEnVar

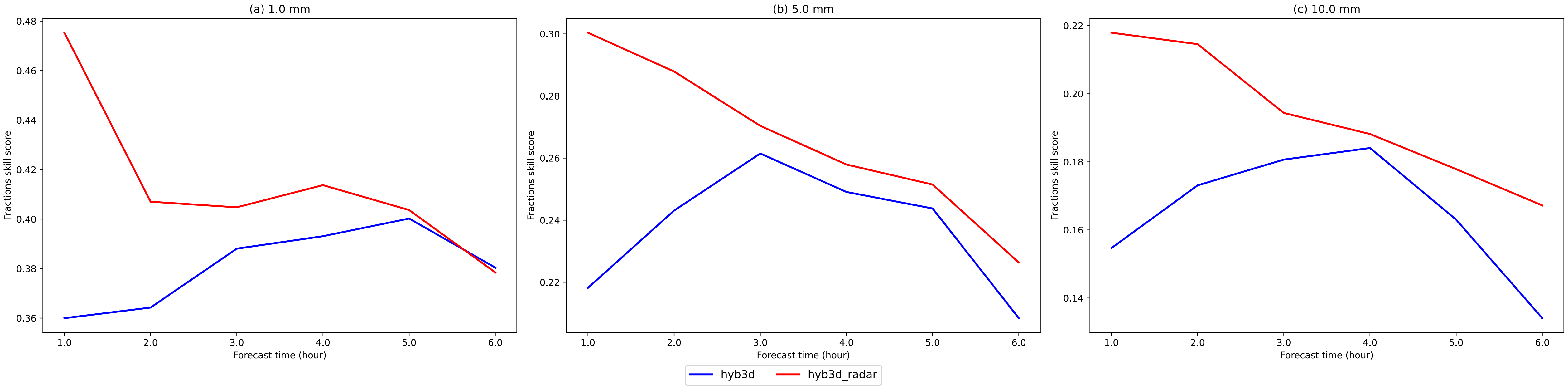


Computed from 6 forecasts at 00 UTC, 8 - 13 July, 2023

NCAR
UCAR

Preliminary Radar DA

Hybrid-3DEnVar: without vs. with radar (radial wind + reflectivity)



33 forecasts from 00 UTC 9 to 18 UTC 17, July

Future Perspectives for improvement

- Assimilate hourly ABI radiances in all-sky mode with hybrid-4DEnVar (in progress)
- Combine ABI and Radar DA
- More frequent cycling: e.g., hourly with assimilation of sub-hourly ABI and radar data
- Use MPAS-JEDI's own ensemble with more members from LETKF
- Use higher-resolution ensemble

Regional MPAS-JEDI test case

- cd ~/mpas_jedi_tutorial/conus15km

- sbatch run_conus15km.sh
  - 15km 3DEnVar with only radiosonde obs and 5-member ensemble input