Bringing GSI Background Error Covariance Capability to JEDI

Ricardo Todling

NASA Global Modeling and Assimilation Office

Unifying Innovations in Forecasting Capabilities Workshop (EPIC/NOAA)

July 18-22, 2022

Collaborators: Dan Holdaway(1), Benjamin Menetrier(2), Russ Treadon(3) & Catherine Thomas(3)

(1) NASA/Global Modeling and Assimilation Office
(2) Institut de Recherche en Informatique de Toulouse
(3) NOAA/National Centers for Environmental Prediction

Background: early FSOI experiment with 3DEnVar

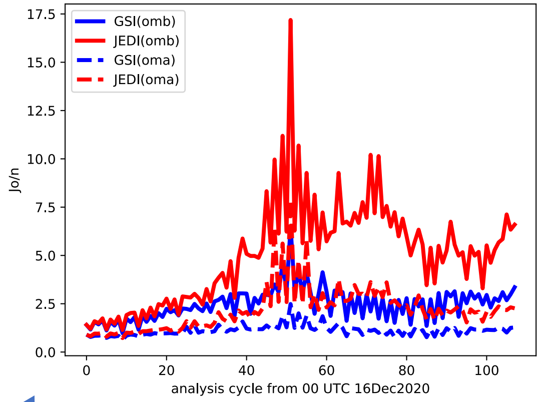
Some time ago, we implemented and tested a FSOI knob for JEDI.

To provide for a fair comparison we removed from GSI:

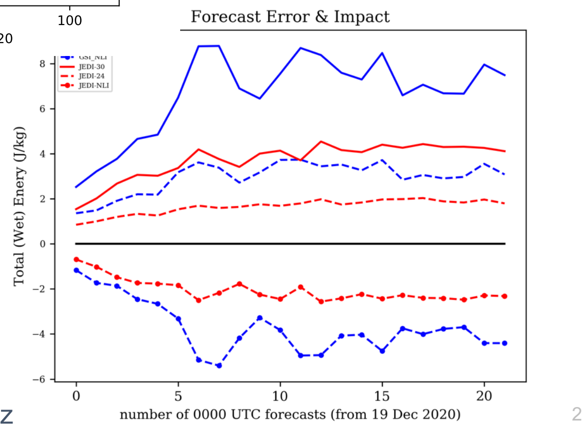
- ➢ TLNMC
- ➢ Dry Mass Constraint
- ➢ 3DEnVar due to lack of B-Clim in JEDI

The RAOB-only results showed:

- ➢ JEDI not drawing as hard to obs as GSI
- ➢ JEDI forecast errors much small than GSI’s (in self-verification this is typically a problem).



We want to revisit these results by running with 3DVar instead of 3DEnVar; tunning the ensemble was not an easy task, and it was not good enough.



With D. Holdaway & F. R. Diniz

Initial Blurb

- Initial implementation of JEDI-based DAS to follow same configuration as current operational systems: Hybrid 4DEnVar.

- For this reason, both NCEP & GMAO require JEDI to provide a climatological B.

- Just as done with UFO where GSI and JEDI have strived to get Obs-minus-Bkg as close as possible to each other, it would be nice to use the same prescription of B-Clim as that used in our operational systems to reduce differences in the analysis.

- Just as with UFO, results between GSI & JEDI from B-Clim tests will not be identical to each other given differences in underlying grids (cubed vs lat-lon or Gaussian), and interpolation operators (Atlas vs GSI).

- Goal of this initial effort: re-run RAOB-only FSOI experiments using 3DVar.

Steps to Bring GSI B-Clim to JEDI

1. Disentangling GSI B-Clim from GSI, i.e., produce standalone code, with minimal dependences.

2. GMAO version of GSI has an implementation (done some time back) to allow applying a vector to B – notice, this is not an increment test, this is a covariance test.

3. GMAO version of GSI has a knob to replace the binary B-Clim Coeffs file (berror_stats) w/ NC4.

4. Embed new library in JEDI infrastructure: setup test cases.

5. *Allow for state dependent covariances: setup test cases.*

6. *Build more realistic exercise; compare GSI/JEDI analyses; re-run FSOI in 3DVar mode.*

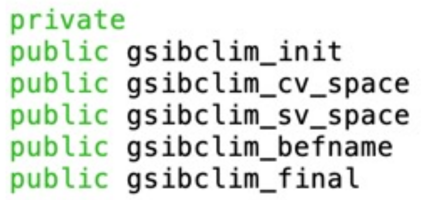
Steps 1-4 are pretty much done; starting to work on Step 5.

GSIBCLIM Library: Disentangling B-Clim from GSI

1. Took the minimal set of routines from GSI

- Ifdef'ed parts of the code out with flag: USE_ALL_ORIGINAL
- Bypassed regional option
- Bypassed hybrid knobs
- Bypassed balance operators
- Bypassed lsqrtb option
- ifdef'ed out aircraft predictors from berror (from f90 to F90) – wrong place for this!
- Converted some code from *.f90 to *.F90

Interface from GSI B-Clim to JEDI:
m_gsibclim.F90



```
private
public gsibclim_init
public gsibclim_cv_space
public gsibclim_sv_space
public gsibclim_befname
public gsibclim_final
```

This is the only module referenced in JEDI and the public procedures here are the only ones made available to JEDI.

Another procedure is being added to handle the BKG

2. Set up & executable

Namelist: gsiberror.nml → this file combines both the typical GSI-like namelist and anavinfo

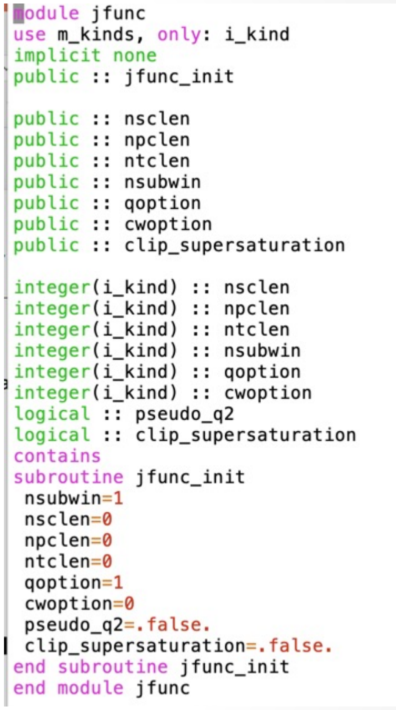
Executable: test_bkerror_clim.x → program this can be used for testing application of SV or CV to B

3. The only dependence left (aside from MPI) is libsp.a (& only needed in Gaussian grid case).

GSIBCLIM Library: Disentangling B-Clim from GSI

4. Be aware

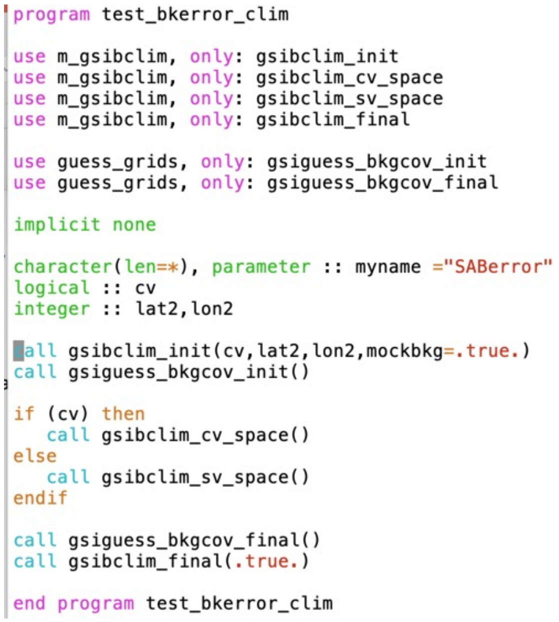
➢ In some cases, the file name is familiar, but its content is substantially different.



```
module jfunc
use m_kinds, only: i_kind
implicit none
public :: jfunc_init

public :: nsclen
public :: npclen
public :: ntclen
public :: nsubwin
public :: qoption
public :: cwoption
public :: clip_supersaturation

integer(i_kind) :: nsclen
integer(i_kind) :: npclen
integer(i_kind) :: ntclen
integer(i_kind) :: nsubwin
integer(i_kind) :: qoption
integer(i_kind) :: cwoption
logical :: pseudo_q2
logical :: clip_supersaturation
contains
subroutine jfunc_init
 nsubwin=1
 nsclen=0
 npclen=0
 ntclen=0
 qoption=1
 cwoption=0
 pseudo_q2=.false.
 clip_supersaturation=.false.
end subroutine jfunc_init
end module jfunc
```

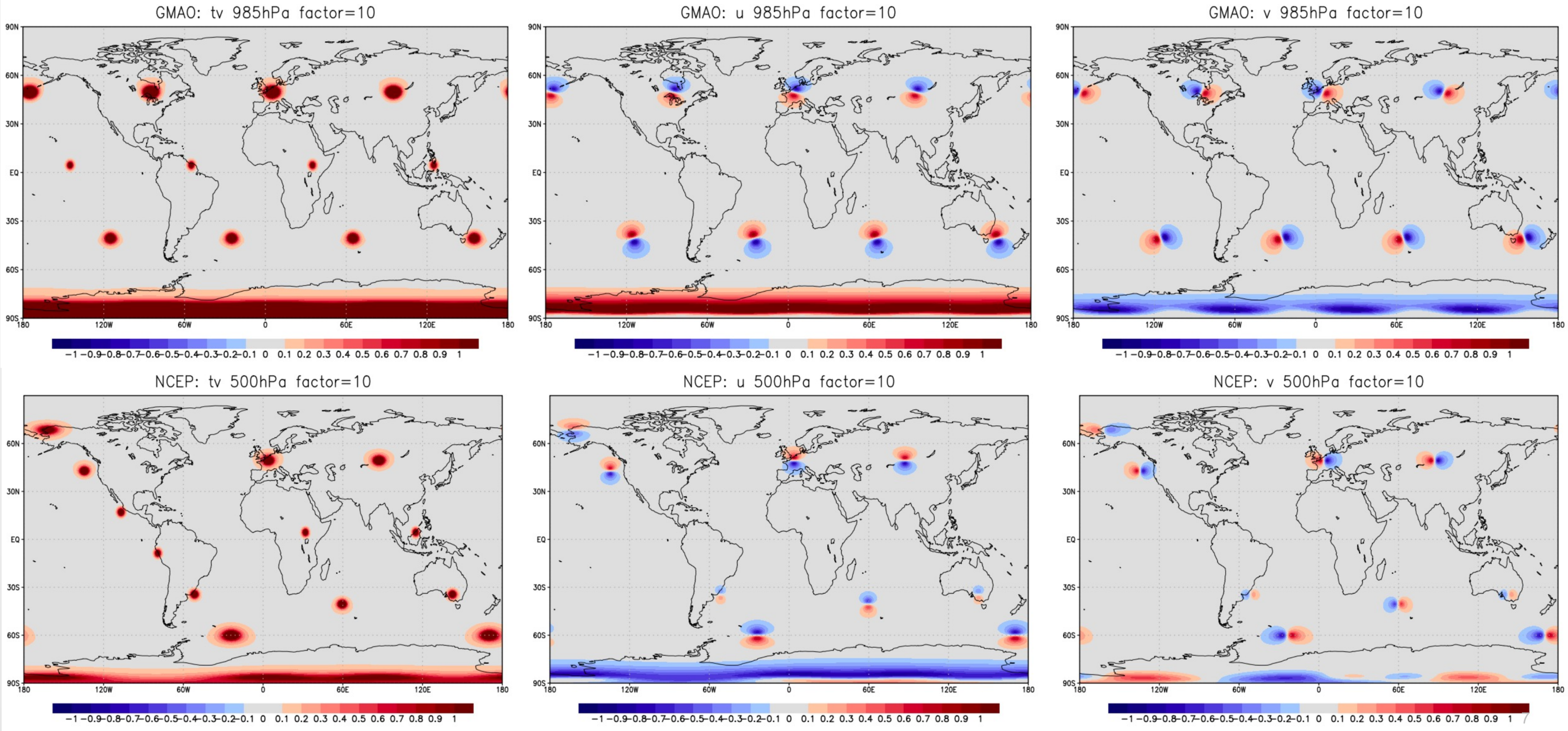
A case in point is jfunc.f90; it is an extreme example.

5. The offline program that allows for testing GSIBCLIM is very simple.



```
program test_bkerror_clim

use m_gsibclim, only: gsibclim_init
use m_gsibclim, only: gsibclim_cv_space
use m_gsibclim, only: gsibclim_sv_space
use m_gsibclim, only: gsibclim_final

use guess_grids, only: gsiguess_bkgcov_init
use guess_grids, only: gsiguess_bkgcov_final

implicit none

character(len=*), parameter :: myname ="SABerror"
logical :: cv
integer :: lat2,lon2

call gsibclim_init(cv,lat2,lon2,mockbkg=.true.)
call gsiguess_bkgcov_init()

if (cv) then
   call gsibclim_cv_space()
else
   call gsibclim_sv_space()
endif

call gsiguess_bkgcov_final()
call gsibclim_final(.true.)

end program test_bkerror_clim
```

T Dirac's @ lowest level and Corresponding Wind Covariances: offline GSI B-Clim



Link GSIBCLIM Library to JEDI

1. We have added a knob to SABER to either stub or use GSIBCLIM.

2. FV3-JEDI's TL/AD variable conversion programs have been updated to handle additional variables, such as O3, RH, QI, QL.

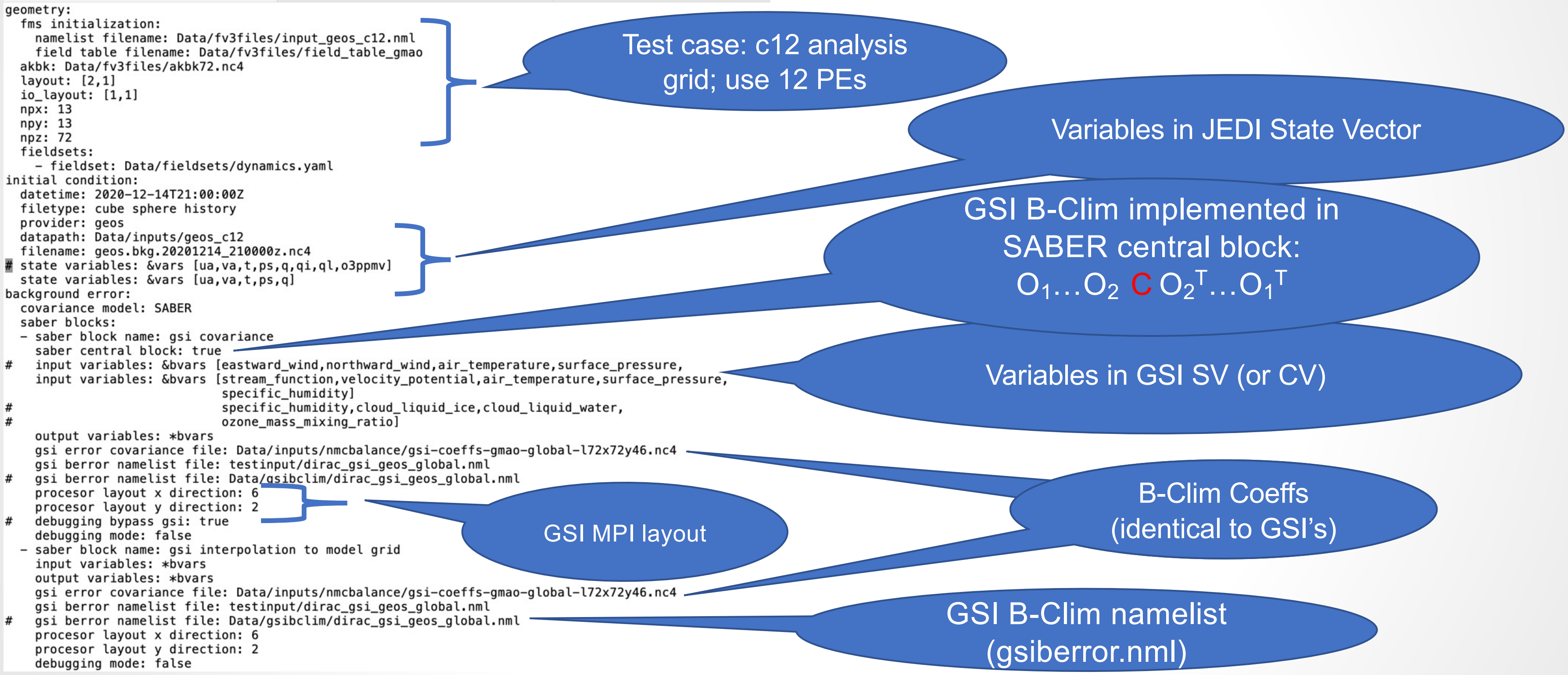
3. JEDI program tested so far: fv3jedi_dirac.x; fv3jedi_var.x

4. Relevant yaml files:

- dirac_geos_gsi_global.yaml
- dirac_gfs_gsi_global.yaml
- geos_gsi_global.yaml
- gfs_gsi_global.yaml

GSIBCLIM Library: Disentangling B-Clim from GSI

A quick look at the namelist: dirac_geos_gsi_global.yaml



```
geometry:
  fms initialization:
    namelist filename: Data/fv3files/input_geos_c12.nml
    field table filename: Data/fv3files/field_table_gmao
  akbk: Data/fv3files/akbk72.nc4
  layout: [2,1]
  io_layout: [1,1]
  npx: 13
  npy: 13
  npz: 72
  fieldsets:
    - fieldset: Data/fieldsets/dynamics.yaml
initial condition:
  datetime: 2020-12-14T21:00:00Z
  filetype: cube sphere history
  provider: geos
  datapath: Data/inputs/geos_c12
  filename: geos.bkg.20201214_210000z.nc4
# state variables: &vars [ua,va,t,ps,q,qi,ql,o3ppmv]
  state variables: &vars [ua,va,t,ps,q]
background error:
  covariance model: SABER
  saber blocks:
  - saber block name: gsi covariance
    saber central block: true
#   input variables: &bvars [eastward_wind,northward_wind,air_temperature,surface_pressure,
    input variables: &bvars [stream_function,velocity_potential,air_temperature,surface_pressure,
                             specific_humidity]
#                            specific_humidity,cloud_liquid_ice,cloud_liquid_water,
#                            ozone_mass_mixing_ratio]
    output variables: *bvars
    gsi error covariance file: Data/inputs/nmcbalance/gsi-coeffs-gmao-global-l72x72y46.nc4
    gsi berror namelist file: testinput/dirac_gsi_geos_global.nml
#   gsi berror namelist file: Data/gsibclim/dirac_gsi_geos_global.nml
    procesor layout x direction: 6
    procesor layout y direction: 2
#   debugging bypass gsi: true
    debugging mode: false
  - saber block name: gsi interpolation to model grid
    input variables: *bvars
    output variables: *bvars
    gsi error covariance file: Data/inputs/nmcbalance/gsi-coeffs-gmao-global-l72x72y46.nc4
    gsi berror namelist file: testinput/dirac_gsi_geos_global.nml
#   gsi berror namelist file: Data/gsibclim/dirac_gsi_geos_global.nml
    procesor layout x direction: 6
    procesor layout y direction: 2
    debugging mode: false
```

Test case: c12 analysis grid; use 12 PEs

Variables in JEDI State Vector

GSI B-Clim implemented in SABER central block: $O_1 \ldots O_2\ C\ O_2^T \ldots O_1^T$

Variables in GSI SV (or CV)

B-Clim Coeffs (identical to GSI's)

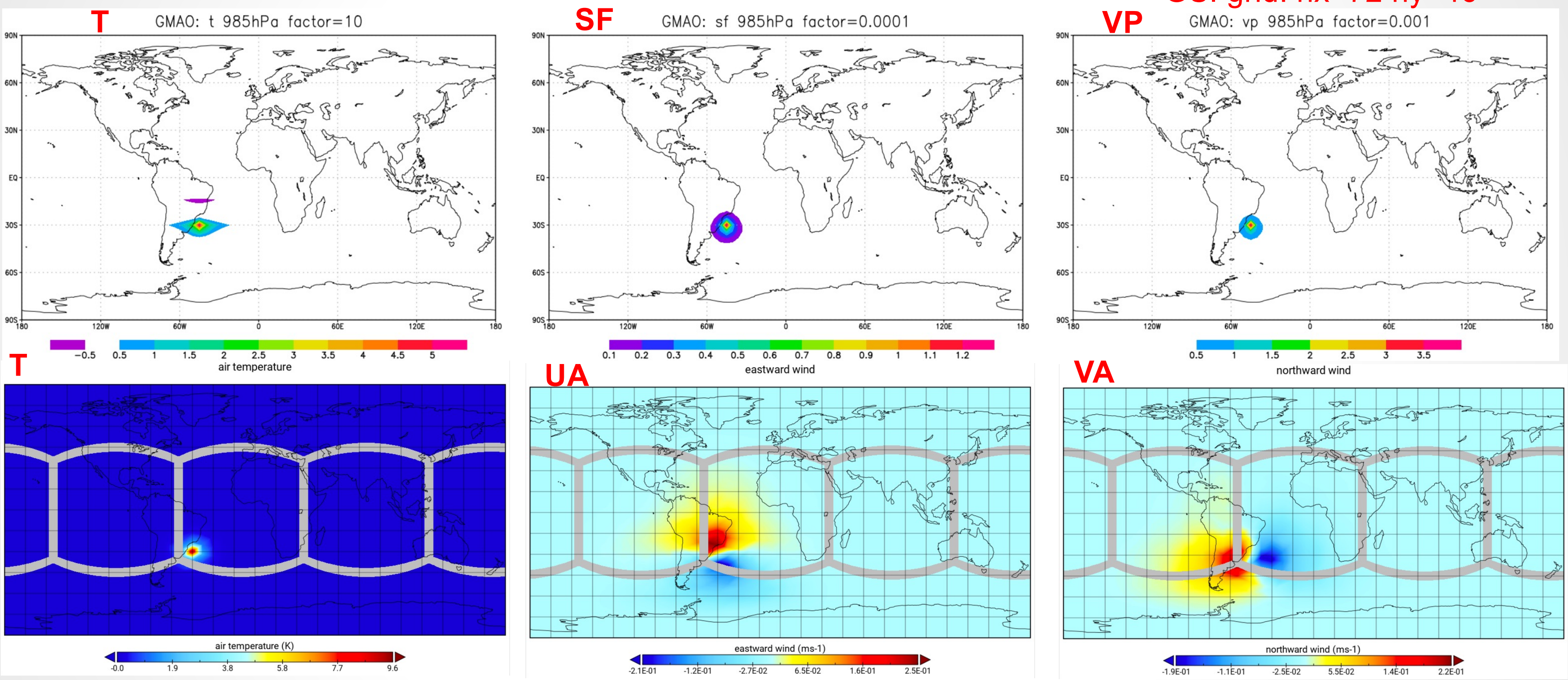
GSI MPI layout

GSI B-Clim namelist (gsiberror.nml)

National Aeronautics and
Space Administration

Treat JEDI Control Vector as GSI Control Vector

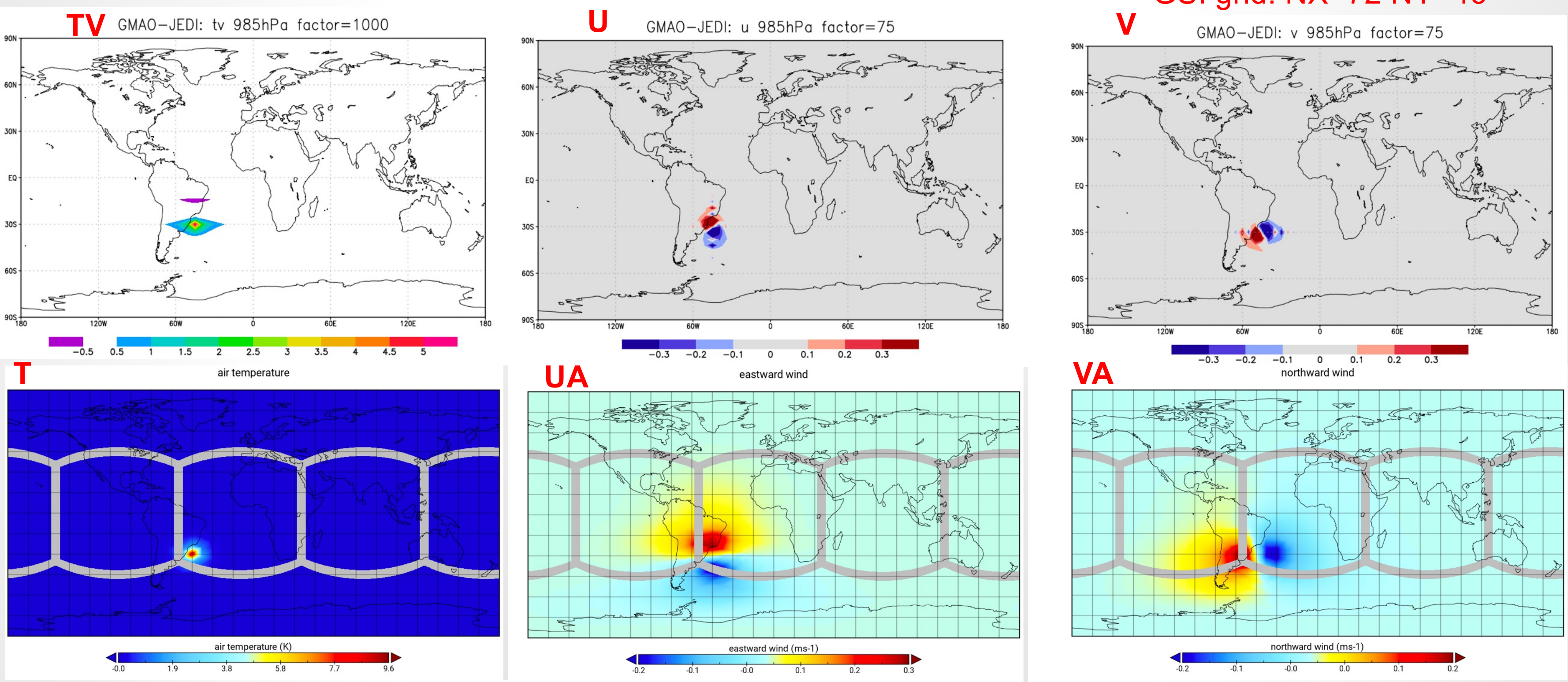
GSI grid: nx=72 ny=46



JEDI grid: c12

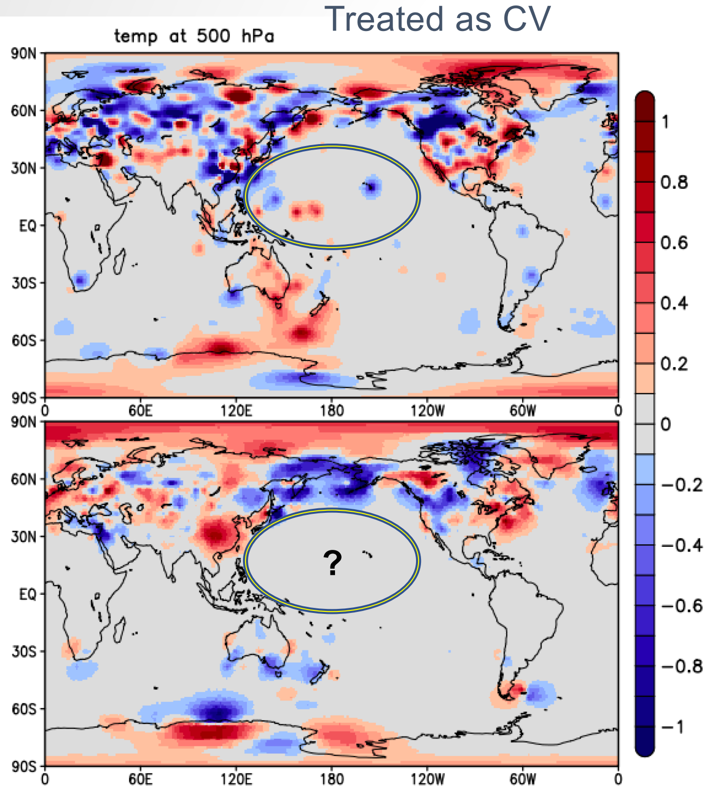
Treat JEDI Control Vector as GSI State Vector

GSI grid: NX=72 NY=46



JEDI grid: c12

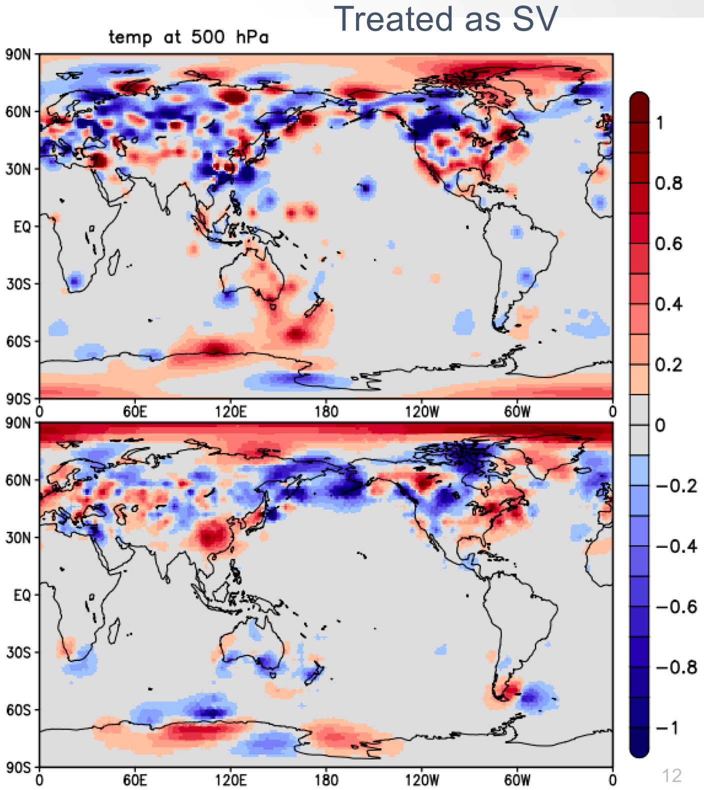
Increments from Radiosonde-only test: c90


Treated as CV

Somehow there is not a one-to-one match of sonde observations between GSI and JEDI.

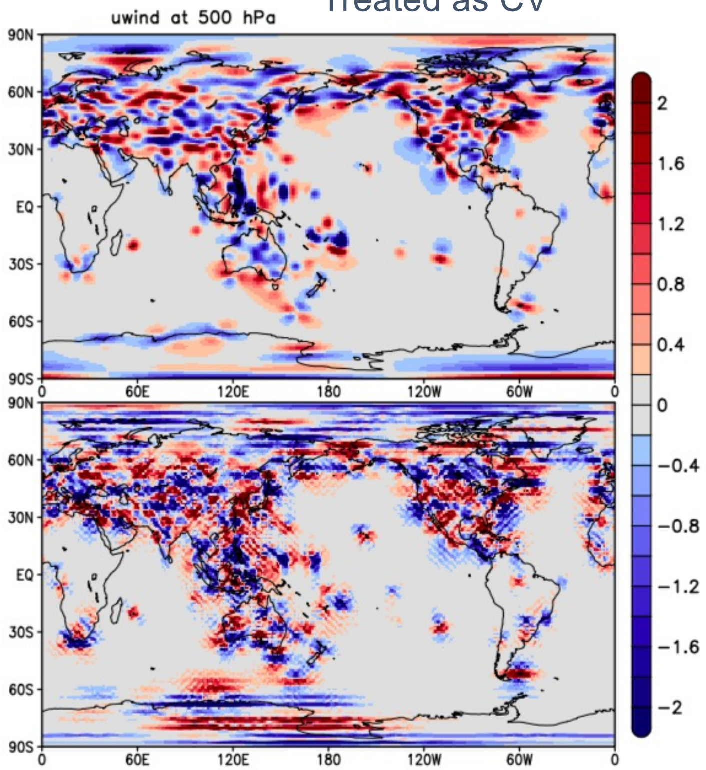
- Conversion of SF/VP by JEDI very sensitive;
- LL or Gaussian vs Cubed grid will always introduce undesirable errors.

The solution is trivial: Calculate increment on LL or Gaussian grid; only final increment needs to be interpolated to model (cubed) grid as presently done in GEOS and GFS.


Treated as SV

Increments from Radiosonde-only test: c90

Treated as CV

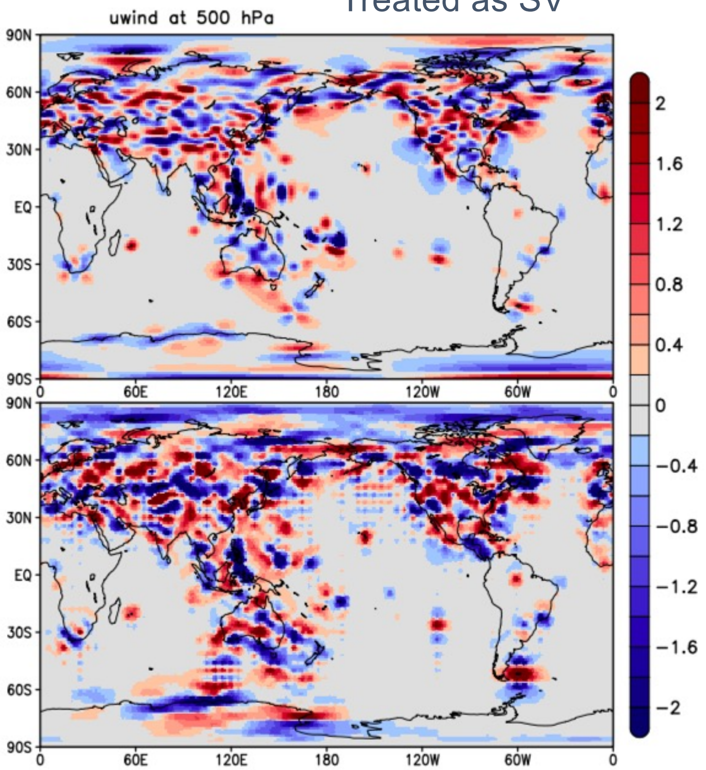


Different grid for JEDI increment and GSI-B calculation leads to unnecessary errors:

- ➢ Conversion of SF/VP by JEDI very sensitive, better handled as SV;
- ➢ LL or Gaussian vs Cubed grid introduces undesirable features.

The solution is trivial: Calculate increment on LL or Gaussian grid; only final increment needs to be interpolated to model (cubed) grid as presently done in GEOS and GFS.

Treated as SV



Much to work on still

- Ozone and Q-option(s) are state-dependent and therefore need the background fields:
  - This capability has been added to SABER but not yet fully exercised.

- A follow up to this presentation will include more details about the code.

- Once a reasonable results has been obtained comparing GSI vs JEDI analyses:
  - Bring in full GSI hybrid B capability to JEDI.
  - Bring in TLNMC capability to JEDI.

- Stepping back and forth onto and out of Gaussian/LL grid will have undesirable consequences to analysis: this will be avoided by redesigning the JEDI interface to the background (and analysis grid) and have it operating on the Gaussian/LL grid instead of the Cubed-grid.