2015 GSI/EnKF Community Tutorial
August 11-14, 2015, Boulder, CO

GSI Fundamentals (5): Review and Applications

Chunhua Zhou
Ming Hu and Kathryn Newman

Developmental Testbed Center (DTC)

Outline

- GSI fundamentals (1): Setup and Compilation
- GSI fundamentals (2): Run and Namelist
- GSI fundamentals (3): Diagnostics
- GSI fundamentals (4): BE, obs error and GSI hybrid
- **GSI fundamentals (5): Review and Applications**
  - ✓ Successfully set up regional GSI
    - ✓ Conventional observations
    - ✓ Radiance data
    - ✓ GPS RO data
  - ✓ Successfully set up global GSI
  - ✓ Learn to check run status
  - ✓ Learn to understand diagnostics in the context of a particular data source and/or application
  - ✓ Ensure the run was successful!
- This talk is tailored to Chapter 5 of the GSI User's Guide for community release v3.4 and builds on knowledge from:

  'GSI Fundamentals (2): Run and Namelist' & 'GSI Fundamentals (3): Diagnostics'

Introduction

- Steps to running a successful GSI Analysis:

1. Obtain background field
2. Grab desired observational data
3. Choose the appropriate anavinfo file (and modify it) to have the same vertical levels as the GSI background
4. Modify run script to suit specific application and properly link observational data
   - ✓ Additional steps specific to observational data (e.g: thinning and bias correction for radiance)
5. Run GSI
6. Check run status and completion of each step of the GSI analysis *(stdout)*
7. Diagnose analysis results *(fit files)*
8. Check analysis increment, cost function/norm of gradient *(DTC graphics utilities available)*

➢ This case study available at: http://www.dtcenter.org/com-GSI/users/tutorial/online_tutorial/index_v3.4.php *(practice case one)*

Step 1 and 2: Case data

- Regional cases using WRF-ARW
  - WRF- NMM similar
- Land mask *(shown below)* of the background used in case study
- Horizontal resolution 30-km & 51 vertical sigma levels

1) Background:

wrfinput_<domain>.<yyyymmddhh>

- Obtained from the GFS forecast through WRF WPS/REAL

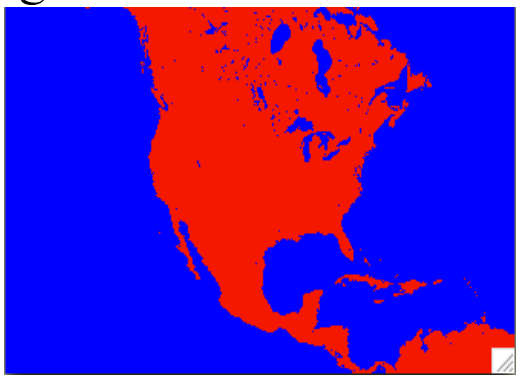


Fig: Landmask of case study background

2) Real-time and archived observation data available

- Case Study data available at:

http://www.dtcenter.org/com-GSI/users/downloads/cases/index.php

Step 3: anavinfo file

- ncdump –h wrfinput_d01 | more
- fix/anavinfo_arw_netcdf

```
netcdf wrfinput_d01
{dimensions:
Time = UNLIMITED ; // (1 currently)
DateStrLen = 19 ;
west_east = 332 ;
south_north = 215 ;
bottom_top = 50 ;
bottom_top_stag = 51 ;
soil_layers_stag = 4 ;
west_east_stag = 333 ;
south_north_stag = 216 ;
DIM0009 = 5 ;
land_cat_stag = 24 ;
soil_cat_stag = 16 ;
num_ext_model_couple_dom_stag = 1 ;
 …  …
```

```
… …
state_derivatives
::!var   level   src
  ps      1      met_guess
  u      50      met_guess
  v      50      met_guess
  tv     50      met_guess
  q      50      met_guess
  oz     50      met_guess
  cw     50      met_guess
 Prse    51      met_guess
::
 … …
```

Conventional Observation Assimilation

- Run Script
- Run Status & Completion
- Analysis Fit to Observations
- Minimization
- Analysis Increment

- This case study available at: http://www.dtcenter.org/com-GSI/users/tutorial/online_tutorial/index_v3.4.php *(practice case one)*

Step 4: Run Script

run_gsi_regional.ksh

- Set up GSI run script following 'GSI Run and Namelist' talk
  - Set paths to data, exe, fix files, etc:

Experimental Setup

```
ANAL_TIME=2014061700
WORK_ROOT=gsiprd_${ANAL_TIME}_prepbufr
BK_ROOT=data/${ANAL_TIME}/arw
BK_FILE=${BK_ROOT}/wrfinput_d01.2014061700
OBS_ROOT=/data/${ANAL_TIME}/obs
PREPBUFR=${OBS_ROOT}/nam.t00z.prepbufr.tm00.nr
CRTM_ROOT=CRTM_REL-2.1.3
GSI_ROOT=comGSIv3.4_EnKFv1.0/
FIX_ROOT=${GSI_ROOT}/fix
GSI_EXE=${GSI_ROOT}/run/gsi.exe
GSI_NAMELIST=${GSI_ROOT}/run/comgsi_namelist.sh
bk_core=ARW
bkcv_option=NAM
if_clean=clean
```

Location of PREPBUFR data

Location of GSI namelist script

- Namelist using default options in the sample script

  (*see: GSI Fundamentals (2): Run and Namelist*)

Step 5: Run Status

While GSI is still running…

- In ${WORK_ROOT} contents should include:

```
imgr_g12.TauCoeff.bin   ssmi_f15.SpcCoeff.bin
imgr_g13.SpcCoeff.bin   ssmi_f15.TauCoeff.bin
imgr_g13.TauCoeff.bin   ssmis_f16.SpcCoeff.bin
```

  - Indicates CTRM coefficients linked to this run directory

stdout:	*standard out file*
wrf_inout:	*background file*
gsiparm.anl:	*GSI namelist*
prepbufr:	*PREPBUFR file for conventional observation*
convinfo:	*data usage control for conventional data*
berror_stats:	*background error file*
errtable:	*observation error file*

  - Indicates run scripts have successfully setup a run environment for GSI and the .exe is running

Step 5: Run Status

While GSI is still running…

- Check the content of the standard out file to monitor the stage of the GSI analysis:
- **`tail -f stdout`**

1st outer loop

Inner iteration

```
GLBSOI:  START pcgsoi jiter=           1
pcgsoi: gnorm(1:2),b=  1.13152054892331361 8E+05  1.13152054892331361 8E+05  0.00000000000000000 0E+00
Initial cost function =  3.24958551456716595 6E+04
Initial gradient norm =  3.36380818258609053 8E+02
cost,grad,step,b,step? =   1   0  3.24958551456716595 6E+04  3.36380818258609053 8E+02
   2.54855354723164014 0E-02  0.00000000000000000 0E+00  good
pcgsoi: gnorm(1:2),b=  9.71189065354544873 1E+04  9.71189065354545455 2E+04  8.58304399578664445 3E-01
cost,grad,step,b,step? =   1   1  2.96121144369476642 2E+04  3.11639064520888496 2E+02
   2.51452180550157285 1E-02  8.58304399578664445 3E-01  good
pcgsoi: gnorm(1:2),b=  8.59709165907255664 9E+04  8.59709165907252172 5E+04  8.85212979198241378 7E-01
```

- Shows that GSI is in the optimal interation stage.

Step 5: Run Completion

- Upon successful completion – the run directory should look like:

```
anavinfo                    fort.202  fort.214  fort.228            satbias_angle
berror_stats                fort.203  fort.215  fort.229            satbias_in
convinfo                    fort.204  fort.217  fort.230            satbias_out
diag_conv_anl.2014061700    fort.205  fort.218  gsi.exe             satinfo
diag_conv_ges.2014061700    fort.206  fort.219  gsiparm.an          stdout
errtable                    fort.207  fort.220  l2rwbufr            prepbufr
fit_p1.2014061700           fort.208  fort.221  list_run_directory  wrfanl.2014061700
fit_q1.2014061700           fort.209  fort.223  ozinfo              wrf_inout
fit_rad1.2014061700         fort.210  fort.224  pcpbias_out
fit_t1.2014061700           fort.211  fort.225  pcpinfo
fit_w1.2014061700           fort.212  fort.226  stdout.anl.2014061700
fort.201                    fort.213  fort.227  prepobs_prep.bufrtable
```

- Number of files will be greatly reduced from the run stage due to the 'clean' option in the run script.

✓ Important! Always check for successful completion of GSI analysis
  - Completion of GSI without crashing does **not** guarantee a successful analysis

Step 6: Run Completion

stdout: Reading in namelist

- *Indication GSI started normal and has read in the namelist:*

```
SETUP_4DVAR: l4dvar=  F
SETUP_4DVAR: l4densvar=  F
SETUP_4DVAR: winlen=    3.000000000000000
   …  …
 &SETUP
 GENCODE =   78.0000000000000     ,
 FACTQMIN        =  0.000000000000000E+000,
```

- *Indication GSI is reading the background fields:* *stdout: Reading in background field*

```
end_index =            332          215          50          0
max,min XLAT(:,1)=     9.928894        1.901657
max,min XLAT(1,:)=     48.19735        1.901657
xlat(1,1),xlat(nlon,1)=    1.901657        1.901657
xlat(1,nlat),xlat(nlon,nlat)=    48.19735        48.19735
…  …
rmse_var = U
ndim1=            3
WrfType =           104
WRF_REAL=          104
ierr  =             0
ordering = XYZ
staggering =  N/A
start_index =             1            1            1            0
end_index =           333          215          50          0
k,max,min,mid U=          1    18.50961      -17.84097      -0.8667576
k,max,min,mid U=          2    18.68178      -18.39229      -0.8647658
k,max,min,mid U=          3    19.28049      -19.42709      -0.8610985
k,max,min,mid U=          4    19.60607      -21.29182      -0.8547171
k,max,min,mid U=          5    21.58153      -24.50086      -0.8405453
                          K    Maximum       Minimum      Central grid
```

Domain information

Check the range of the minimum and maximum values to indicate if background fields are normal

Step 6: Run Completion

stdout: Reading in observational data

- In the middle of the stdout file:

```
OBS_PARA: ps        1429    3190    4655    6774
OBS_PARA: t         2564    5200    7057   11128
OBS_PARA: q         2346    4626    6148    8128
OBS_PARA: pw          65      80      63      49
OBS_PARA: uv        3358    6453    8091   11998
```

Observation Type

Distribution of observations in each sub-domain

- ✓ This table is important to see if the observations have been read in properly.
- ✓ But the data have to go through QC process to be finally used in the analysis, more detailed information in fit files

Step 6: Run Completion

stdout: optimal iteration

The namelist specified 2 outer loops with 50 inner loops

- The optimal iteration step will look as follows:

```
GLBSOI:  START pcgsoi jiter=            1
pcgsoi: gnorm(1:2),b=  1.13152054892331361 8E+05  1.13152054892331361 8E+05  0.00000000000000000E+00
Initial cost function =  3.24958551456716595 6E+04
Initial gradient norm =  3.36380818258609053 8E+02
cost,grad,step,b,step? =   1   0  3.24958551456716595 6E+04  3.36380818258609053 8E+02
 2.54855354723164014 0E-02  0.00000000000000000E+00  good
```

- … last iteration:

```
cost,grad,step,b,step? =   2  45  2.28306638487603049 7E+04  3.97619938528637195 7E-03
  2.25313861190973123 4E-02  9.33373053899598348 3E-01  good
pcgsoi: gnorm(1:2),b=  7.56593578602322126 1E-06  7.56599549180765260 1E-06
  4.78552699612678600 3E-01
cost,grad,step,b,step? =   2  46  2.28306638484040886 4E+04  2.75062461743205179 1E-03
  5.60348131013475095 3E-02  4.78552699612678600 3E-01  good
PCGSOI: WARNING **** Stopping inner iteration ***
gnorm  0.66865208883943888 9E-10 less than  0.10000000000000000 4E-09
update_guess: successfully complete
```

Iteration check: The J value should descend through iterations

The iteration met the stop threshold before meeting the maximum iteration

Step 6: Run Completion

stdout: Write out analysis results

- Final step (write out results) looks very similar to section reading in background fields:

```
max,min MU=    2799.898       -1195.195
rmse_var=MU
ordering=XY
WrfType,WRF_REAL=          104          104
ndim1=            2
staggering= N/A
start_index=            1            1            1            0
end_index1=           332          215           50            0
k,max,min,mid T=            1    321.6157        270.8151        309.3634
k,max,min,mid T=            2    321.7112        270.9660        309.4070
k,max,min,mid T=            3    321.4324        271.2166        309.3831
k,max,min,mid T=            4    321.2418        271.6100        309.3864
```

- As an indication that GSI has successfully run – the following lines will appear at the end of the file:

```
ENDING DATE-TIME    JUN 06,2014  16:47:10.518  157  FRI   2456815
                     PROGRAM GSI_ANL HAS ENDED.
```

✓ **It can be concluded GSI successfully ran through every step with no run issues. It *cannot* be concluded that GSI did a successful analysis until more diagnosis has been completed…**

Step 7: Analysis Fit to Observations

- The analysis uses the observations to correct the background fields to push the analysis results to fit the observations under certain constraints.
- Easiest way to confirm the GSI analysis fit the observations better than the background?
  - ✓ Check fort files!
- Example: fort.203 (t1)

Is the bias and RMS in reasonable range? – should be checked

				ptop	1000.0	900.0	800.0	600.0	400.0	300.0	250.0	200.0	150.0	100.0	50.0	0.0
it	obs	type	styp	pbot	1200.0	1000.0	900.0	800.0	600.0	400.0	300.0	250.0	200.0	150.0	100.0	2000.0
o-g 01	t	120	0000	count	107	350	357	866	**1153**	719	252	450	551	884	745	7188
o-g 01	t	120	0000	bias	0.80	0.32	-0.10	-0.12	**-0.15**	-0.20	-0.24	-0.60	-0.22	0.15	-0.10	-0.07
o-g 01	t	120	0000	rms	2.06	1.55	0.83	0.77	**0.69**	0.66	0.73	1.20	1.44	1.65	1.65	1.23
o-g 01	t	180	0000	count	1683	50	0	0	0	0	0	0	0	0	0	1733
o-g 01	t	180	0000	bias	0.69	2.04	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.73
o-g 01	t	180	0000	rms	1.99	4.50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.11
o-g 01		all		count	1792	405	358	871	1172	725	325	800	651	884	745	**9482**
o-g 01		all		bias	0.69	0.53	-0.10	-0.12	-0.15	-0.19	-0.09	-0.50	-0.04	0.15	-0.10	**0.08**
o-g 01		all		rms	1.99	2.14	0.83	0.77	0.69	0.67	0.84	1.32	1.58	1.65	1.65	**1.45**

O-B

Data types used:
120: rawinsonde
180: surface marine

Data type 120 has 1153 obs in level 400.0-600.0 mb, bias=-0.15, rms =0.69

Whole atmosphere/all data: O-A – 9482 obs, bias = 0.08, rms = 1.45

Step 7: Analysis fit to Observations

- fort.203 (t) cont'
- Whole atmosphere, all data only: quick view of fitting

o-g 01	all	count	1792	405	358	871	1172	725	325	800	651	884	745	9482
o-g 01	all	bias	0.69	0.53	-0.10	-0.12	-0.15	-0.19	-0.09	-0.50	-0.04	0.15	-0.10	0.08
o-g 01	all	rms	1.99	2.14	0.83	0.77	0.69	0.67	0.84	1.32	1.58	1.65	1.65	1.45

O-B

o-g 03	all	count	1792	406	358	871	1172	725	325	800	651	884	745	9483
o-g 03	all	bias	0.21	0.34	-0.04	-0.02	-0.04	-0.02	0.04	-0.13	0.06	0.06	0.04	0.05
o-g 03	all	rms	1.71	1.61	0.69	0.61	0.49	0.43	0.61	0.94	1.26	1.40	1.59	1.22

O-A

- fort.202 (w)
- fort.201 (p)
- fort.204 (q)

Check other parameters!

➢ 9482 total observations from the background to 9483 for the analysis; the bias reduced from **0.08 to 0.05** & rms reduced from **1.45 to 1.22**. Reduction reasonable for large scale analysis

✓ Statistics show analysis results fit to observation closer than background… how close analysis fit is to observation is based on ratio of background error variance and observation error.

Step 7: Checking Minimization

- In addition to stdout, GSI writes fort.220 with more detailed information on minimization
- Quick check of the trend of the cost function and norm of the gradient:
  - Dump information from fort.220 to an output file:
    - `grep 'cost,grad,step,b' fort.220 | sed -e 's/cost,grad,step,b,step? =  //g' | sed -e 's/good//g' > cost_gradient.txt`
- *cost_gradient.txt* will have 6 columns (4 shown below)

1	**0**	**3.24958551456716595 6E+04**	**3.36380818258609053 8E+02**
1	1	2.96121144369476642 2E+04	3.11639064520888496 2E+02
1	2	2.71700383548489626 3E+04	2.93207974978044489 9E+02
1	3	2.62788851849405764 4E+04	2.38401255718702231 0E+02
1	4	2.51715036756383669 9E+04	1.80735370903411535 4E+02
		… …	
2	42	2.28306638519430816 8E+04	6.12413471509185188 0E-03
2	43	2.28306638506101844 5E+04	7.27540284515745035 9E-03
2	44	2.28306638495001971 0E+04	4.11566384548528625 7E-03
2	45	2.28306638487603049 7E+04	3.97619938528637195 7E-03
2	**46**	**2.28306638484040886 4E+04**	**2.75062461743205179 1E-03**

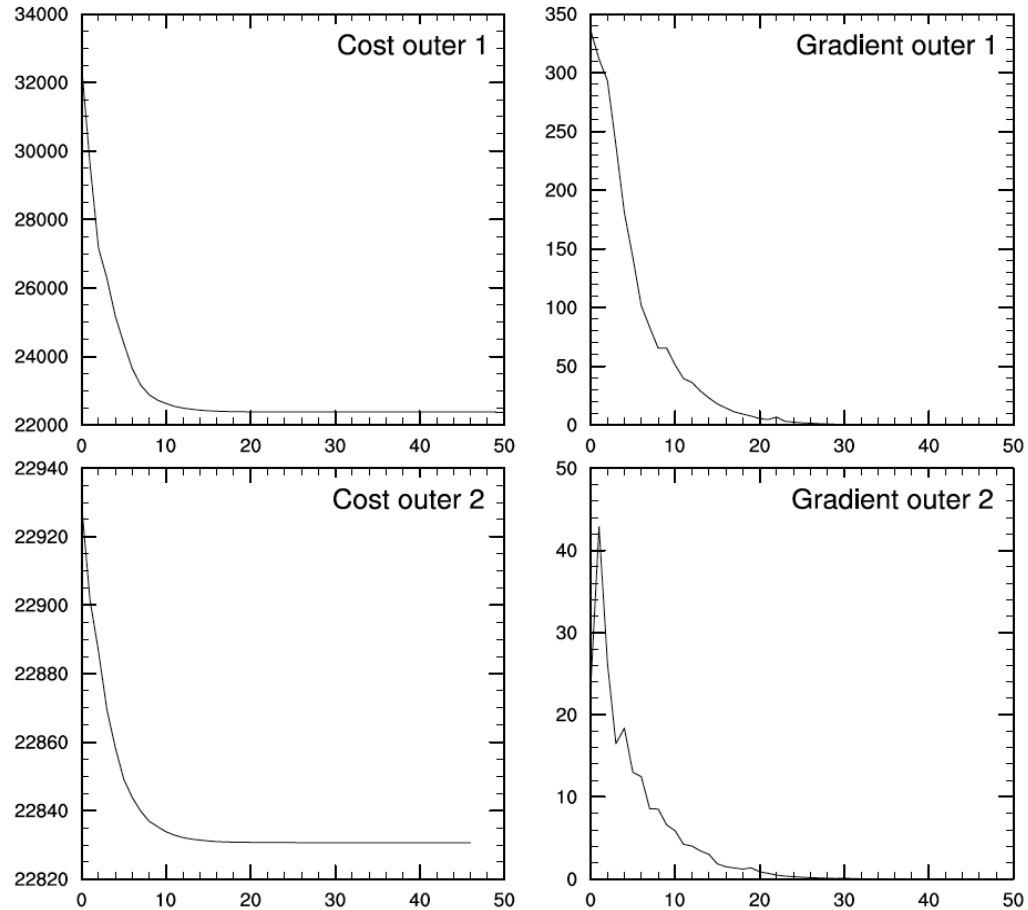
Outer loop | Inner iteration number | Cost function | Norm of gradient

✓ Both the cost function and the norm of gradient are descending with each iteration:
Cost function reduced
From 3.24 E+04 to 2.83 E+04
Norm of gradient reduced
From 3.36 E+02 to 2.75 E-03

Step 8: Checking Minimization

- To gain a complete picture of the minimization process: plot cost function and norm of gradient
  - Script available in v3.4 release: *./util/Analysis_Utilities/plot_ncl/GSI_cost_gradient.ncl*

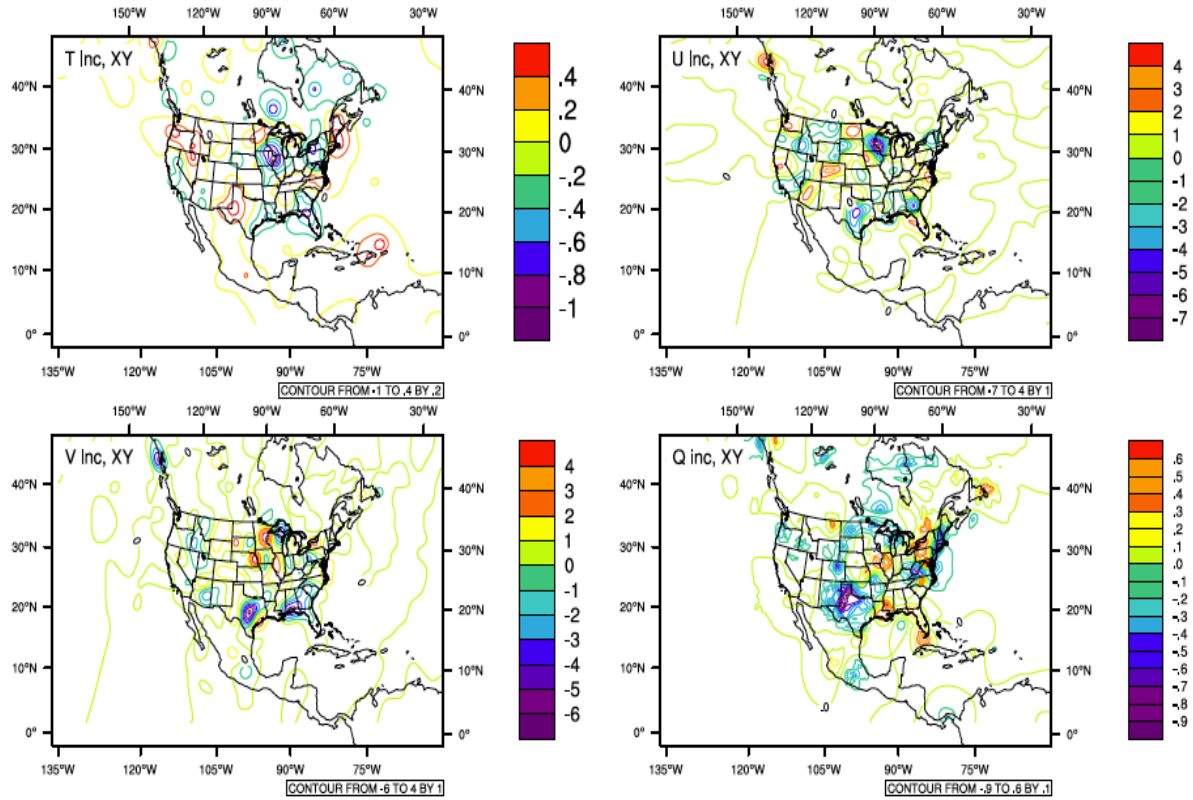
Cost function and norm of gradient descend in both outer loops



Step 8: Checking Analysis Increment

- Analysis increment gives an idea where and how much the background fields have been changed by the observations
  - Graphic tool available in v3.4 release: *./util/Analysis_Utilities/plot_ncl/Analysis_increment.ncl*

Analysis Increment at the 15th Level



Note: different scales for each plot

✓ The U.S. CONUS domain has many upper level observations and the data availability over the ocean is sparse

Radiance Assimilation

In addition to conventional:

- Run Script
- Data thinning and Bias correction
- Run Status & Completion
- Diagnosing analysis results

Step 4: Run Script *run_gsi_regional.ksh*

- Key difference from conventional assimilation: properly link radiance BUFR files to the GSI run directory
- To add the following radiance BUFR files:

AMSU-A: *gdas1.002z.1bamua.tm00.bufr_d*

HIRS4: *gdas1.t00z.1bhrs4.tm00.bufr_d*

- The location of these data is indicated in *OBS_INPUT***
- Insert the following data links after PREPBUFR data in *run_gsi_regional.ksh*:

```
ln -s ${OBS_ROOT}/gdas1.t12z.1bamua.tm00.bufr_d amsuabufr
ln -s ${OBS_ROOT}/gdas1.t12z.1bhrs4.tm00.bufr_d hirs4bufr
```

- Keep link to prepbufr when assimilating both prepbufr and radiance…

```
ln -s ${PREPBUFR} ./prepbufr
```

**To ensure correct name for radiance BUFR file, check namelist section *&OBS_INPUT:*

```
dmesh(1)=120.0,dmesh(2)=60.0,dmesh(3)=30,time_window_max=1.5,ext_sonde=.true.
! dfile      dtype    dplat    dsis        dval    dthin    dsfcalc
amsuabufr   amsua    n15      amsua_n15   10.0    2        0
```

- The AMSU-A observation from NOAA-15 will be read in from BUFR file '**amsuabufr**'

Step 4: Radiance Data Thinning

- Radiance data thinning is set up in the namelist section *&OBS_INPUT:*

dmesh(1)=120.0,dmesh(2)=60.0,dmesh(3)=30,time_window_max=1.5,ext_sonde=.true.

! dfile	dtype	dplat	dsis	dval	dthin	dsfcalc
amsuabufr	amsua	n15	amsua_n15	10.0	2	0

- *&OBS_ROOT* has thinning grid array `dmesh`
- For each data type line, the second last column: `dthin`, is used to select the mesh grid used in the thinning.
- In this case, data thinning for NOAA-15 AMSU-A observation is 60 km

Step 4: Radiance Bias Correction – method 1 (default)

- Radiance bias correction is very important for a successful radiance data analysis.
- *run_gsi_regional.ksh* includes:

```
SATANGL=${FIX_ROOT}/global_satangbias.txt
cp $SATANGL ./satbias_angle
cp ${FIX_ROOT}/sample.satbias ./satbias_in
```

✓ **`satbias_angle`** tells GSI the angle bias (*calculated outside GSI*)

✓ **`satbias_in`** tells GSI the mass bias (*calculated inside GSI from the previous cycle*)

The files **`global_satangbias.txt`** and **`sample.satbias`** can be found in ./fix for an ***example*** of bias correction coefficients.

These two files should be changed using case data or real-time data

✓ *More details on radiance bias correction in the GSI Advanced User's Guide v3.4 – Chapter 8*

Step 4: Radiance Bias Correction – method 2

- Radiance angle dependent bias correction can also be done variationally within GSI, together with the mass bias correction.
- *run_gsi_regional.ksh*:

```
cp ${FIX_ROOT}/gdas1.t00z.abias.new ./satbias_in
```

- *comgsi_namelist.sh:*

```
$SETUP
adp_anglebc=.true.
```

✓ **`satbias_in`** tells GSI the combined angle and mass bias (*calculated inside GSI from the previous cycle*)

The file **`gdas1.t00z.abias.new`** can be found in ./fix as an ***example*** of new bias correction coefficients. Users can also download the satbias coefficient file (i.e., gdas1.t00z.abias) from the NOMADS ftp site as the input file (starting spring 2015, the GDAS satbias files have adopted the new format): ftp://nomads.ncdc.noaa.gov/GDAS/YYYYMM/YYYYMMDD/gdas1.tHHz.abias

Step 6: Run Completion *stdout: Reading in observational data*

- ✓ *[While GSI is running]* Working directory will look similar to conventional, with additional links to the radiance BUFR files
- ✓ Check stdout status and successful completion of each part of the analysis processes
- The radiance data should have been read in and distributed to each sub domain:

```
OBS_PARA: ps                          1429      3190      4655      6774
OBS_PARA: t                           2564      5200      7057     11128
OBS_PARA: q                           2346      4626      6148      8128
OBS_PARA: pw                            65        80        63        49
OBS_PARA: uv                          3358      6453      8091     11998
OBS_PARA: hirs4     metop-a              0         2      1601      2425
OBS_PARA: hirs4     n19                496      1347         0         0
OBS_PARA: hirs4     metop-b              0         0       154       737
OBS_PARA: amsua     n15               1921      2362       996       391
OBS_PARA: amsua     n18               2697      2789       262        36
OBS_PARA: amsua     n19                326      1137         0         0
OBS_PARA: amsua     metop-a              0         0      1321      1967
OBS_PARA: amsua     metop-b              0         0       386      2238
```

8 new radiance data types have been read in

Most radiance data read in are from AMSU-A NOAA-15 and NOAA-18

Step 7: Diagnosing Analysis Results

- The *fort.207* is statistics file for radiance data (*similar to fort.203 for t*)
- Similar to conventional - has statistics for each outer loop:

Penalty

```
     it   satellite instrument  # read   # keep  # assim penalty  qcpnlty     cpen   qccpen
o-g  01   rad  n15    amsua     104190   72740   33302   23716.  23716.  0.71216  0.71216
o-g  01   rad  n18    amsua     104715   85857   35576   19844.  19844.  0.55779  0.55779
```

O-B

...

within the analysis time window and domain

after thinning

used in analysis

```
o-g  03   rad  n15    amsua     104190   72740   40370   6456.7  6456.7   0.15994 0.15994
o-g  03   rad  n18    amsua     104715   85857   41795   11666.  11666.   0.27911 0.27911
```

O-A

- ✓ The penalty for n15 decreased from 23716 to 6465.7 after 2 outer loops
- ✓ The data assimilated increased from 33302 to 40370 after 2 outer loops

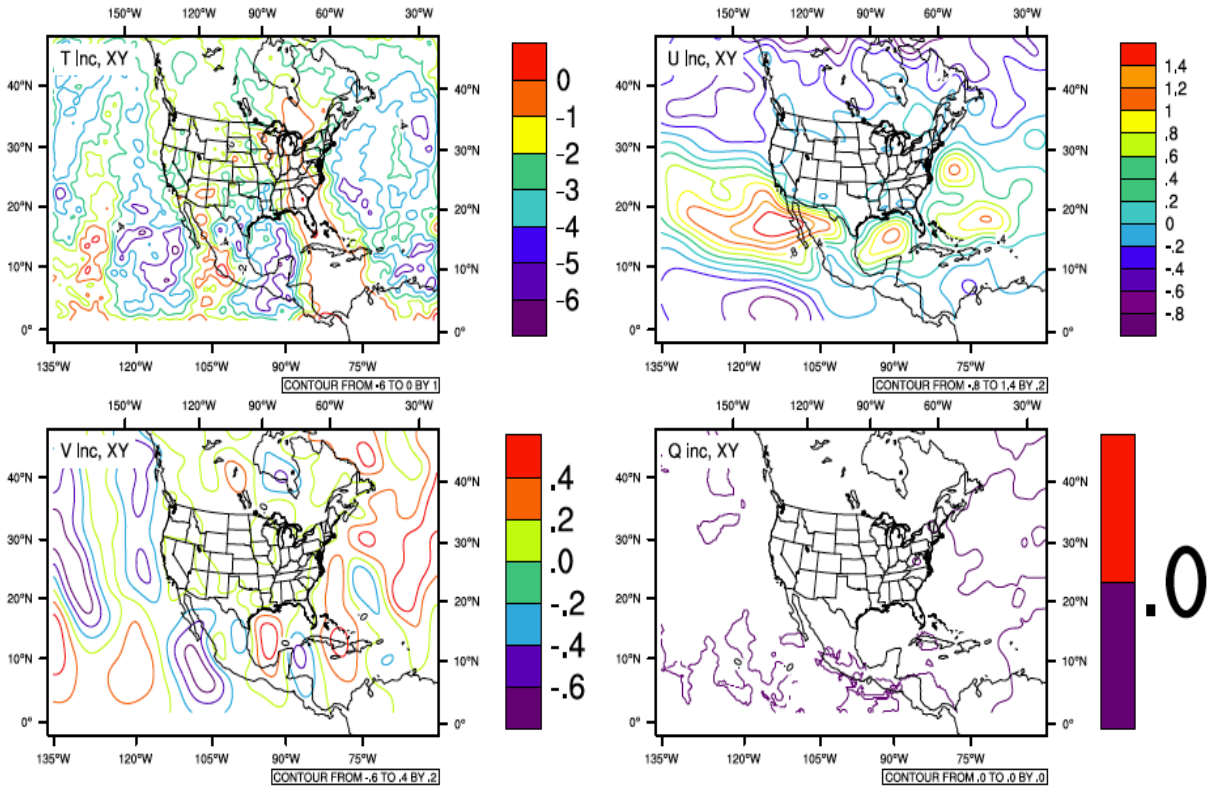
Statistics can also be viewed for each channel in fort.207

Step 8: Checking Analysis Impact

- Analysis increment plotted comparing the analysis results with radiance & conventional and conventional only.
  - Graphic tool available in the v3.4 release: *./util/Analysis_Utilities/plot_ncl/Analysis_increment.ncl*

Analysis Increment at the 49th Level

Increment = (AMSU-A+HIRS4+ PREPBUFR) - PREPBUFR



✓ Impact of radiance data compared to conventional alone most evident over data sparse oceans

Understanding Analysis Results

1. Understand the weighting functions of each channel & data coverage at the analysis time
2. The usage of each channel is located in the file 'satinfo' *(see 'GSI Fundamentals (3): Diagnostics' for more detail)*
3. Understand the implications of thinning
4. Bias correction is very important for successful radiance data analysis

- Radiance bias correction for regional analysis is a difficult issue because of limited coverage of radiance data
  - To be considered/understood when using GSI with radiance applications

GPS RO Assimilation

In addition to conventional/radiance

- Run Script
- Run Status & Completion
- Diagnosing analysis results
- Analysis Increment

Step 4: Run Script *run_gsi_regional.ksh*

- Key difference from conventional/radiance assimilation: properly link to GPSRO BUFR data file in the GSI run directory
- The location of these data is indicated in *OBS_INPUT***
- Insert below link to GPS RO data in *run_gsi_regional.ksh*:

```
ln -s ${OBS_ROOT}/gdas1.t00z.gpsro.tm00.bufr_d gpsrobufr
```

**To ensure correct name for GPS RO BUFR file, check namelist section *&OBS_INPUT:*

```
!  dfile          dtype      dplat     dsis            dval    dthin dsfcalc
   gpsrobufr      gps_ref    null      gps             1.0     0     0
```

- In sample run script, GSI is expecting a GPS refractivity BUFR file named '**gpsrobufr**'

Step 6: Run Completion *stdout: Reading in observational data*

- ✓ While GSI is running, the working directory will look the same, with the additional links to the GPS refractivity BUFR file used
- ✓ Check **stdout** status and successful completion of each part of the analysis processes
- The GPS RO data should have been read in and distributed to each sub domain:

```
OBS_PARA: ps                    1429      3190      4655      6774
OBS_PARA: t                     2564      5200      7057     11128
OBS_PARA: q                     2346      4626      6148      8128
OBS_PARA: pw                      65        80        63        49
OBS_PARA: uv                    3358      6453      8091     11998
OBS_PARA: gps_ref               1799      1368      2664      3520
```

GPS RO refractivity data have been read in and distributed to four sub-domains successfully

Step 7: Diagnosing Analysis Results

- The *fort.212* is statistic file for GPS RO data (*similar to fort.203 for t*)
- Statistics for each outer loop:

it	obs	type	styp	ptop / pbot	1000.0 / 1200.0	900.0 / 1000.0	800.0 / 900.0	600.0 / 800.0	400.0 / 600.0	300.0 / 400.0	250.0 / 300.0	200.0 / 250.0	150.0 / 200.0	100.0 / 150.0	50.0 / 100.0	0.0 / 2000.0
o-g 01		all		count	0	13	58	223	355	342	232	261	326	440	729	3740
o-g 01		all		bias	0.00	-0.76	-0.03	-0.06	-0.04	0.01	-0.03	0.04	-0.04	-0.16	-0.18	-0.14
o-g 01		all		rms	0.00	1.41	0.75	0.96	0.79	0.35	0.32	0.42	0.54	0.57	0.55	0.59
o-g 03		all		count	1	18	65	229	355	342	231	266	330	440	731	3776
o-g 03		all		bias	-0.40	-0.43	0.03	0.02	-0.02	-0.01	-0.02	0.00	0.01	-0.01	-0.02	0.00
o-g 03		all		rms	0.40	1.03	0.54	0.59	0.70	0.26	0.14	0.20	0.24	0.28	0.39	0.41

O-B

most GPS RO observations located in upper levels

O-A

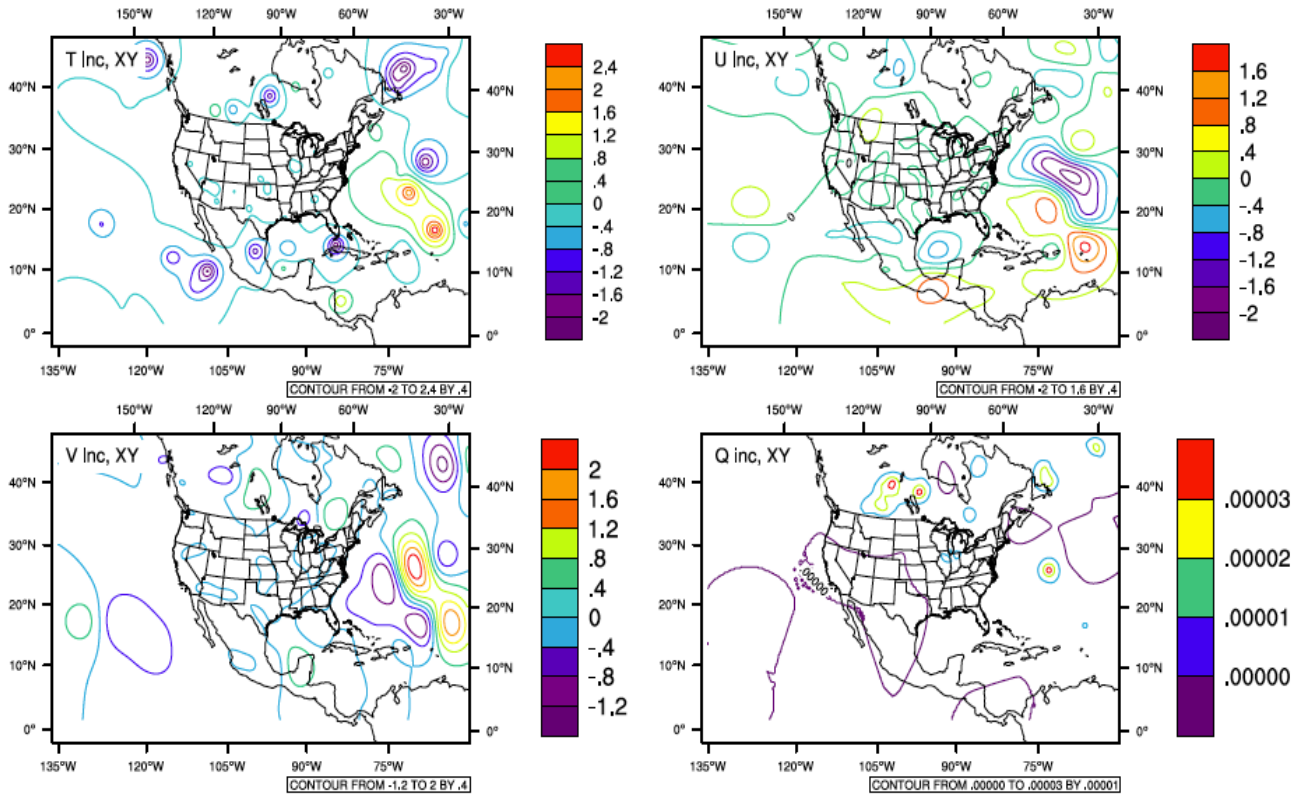
- ✓ 3740 obs used in analysis during 1st outer loop, 3776 used to calculate O-A
- ✓ Bias -0.14 to 0.00 after analysis. RMS reduced from 0.59 to 0.41 after analysis.

Step 8: Checking Analysis Impact

- Analysis increment plotted comparing the analysis results with GPS RO & conventional and conventional only.
  - Graphic tool available in v3.4 release: *./util/Analysis_Utilities/plot_ncl/Analysis_increment.ncl*

Analysis Increment at the 48th Level (upper atmosphere)

Increment = (GPS RO + PREPBUFR) - PREPBUFR



Global GSI Application

- Run Script
- Run Status & Completion
- Diagnosing analysis results

Please download the fix files for global GSI application from the following link:
http://www.dtcenter.org/com-GSI/users/downloads/index.php

Step 4: Run Script

run_gsi_global.ksh

- Set up GSI run script *Experimental Setup*

```
ANAL_TIME=2014070706
GFSCASE=T254
WORK_ROOT=gsiprd_${ANAL_TIME}_${GFSCASE}
BK_ROOT=data/gfs/${GFSCASE}
OBS_ROOT=/data/gfs/${GFSCASE}
PREPBUFR=${OBS_ROOT}/gdas1.t06z.prepbufr.nr
CRTM_ROOT=CRTM_REL-2.1.3
GSI_ROOT=comGSIv3.4_EnKFv1.0/
FIX_ROOT=${GSI_ROOT}/fix/global
GSI_EXE=${GSI_ROOT}/run/gsi.exe
GSI_NAMELIST=${GSI_ROOT}/run/comgsi_namelist_gfs.sh
bk_core=ARW
bkcv_option=NAM
if_clean=clean
```

```
# Set the JCAP resolution which you want.
# All resolutions use LEVS=64
if [[ "$GFSCASE" = "T62" ]]; then
 JCAP=62
 JCAP_B=62
elif [[ "$GFSCASE" = "T126" ]]; then
 JCAP=126
 JCAP_B=126
elif [[ "$GFSCASE" = "enkf_glb_t254" ]]; then
 JCAP=254
 JCAP_B=254
elif [[ "$GFSCASE" = "T254" ]]; then
 JCAP=254
 JCAP_B=574
elif [[ "$GFSCASE" = "T574" ]]; then
 JCAP=574
 JCAP_B=1534
else
 echo "INVALID case = $GFSCASE"
 exit
fi
```

- Namelist using default options in the sample script

(see: GSI Fundamentals (2): Run and Namelist)

Step 4: Run Script

run_gsi_global.ksh

- Set up GSI run script – need GFS background at 3, 6, and 9 hours

```
# Bring over background field (it's modified by GSI so we can't link to it)
# Copy bias correction, atmospheric and surface files
… …
cp $OBS_ROOT/satbias_in          ./satbias_in
cp $OBS_ROOT/satbias_angle       ./satbias_angle
cp $BK_ROOT/sfcf03  ./sfcf03
cp $BK_ROOT/sfcf06  ./sfcf06
cp $BK_ROOT/sfcf09  ./sfcf09
cp $BK_ROOT/sigf03  ./sigf03
cp $BK_ROOT/sigf06  ./sigf06
cp $BK_ROOT/sigf09  ./sigf09
fi
```

GFS surface forecast files at 3 time levels: 3, 6, 9 hours

GFS atmosphere forecast files at 3 time levels: 3,6,9 hours

Step 5: Run Completion

- Upon successful completion – the run directory should look like: (more radiances data used in this case)

```
airsbufr                          diag_mhs_metop-a_anl.2014070706    fit_q1.2014070706    fort.230
amsuabufr                         diag_mhs_metop-a_ges.2014070706    fit_rad1.2014070706  gomebufr
amsubbufrears                     diag_mhs_n18_anl.2014070706        fit_t1.2014070706    gpsrobufr
anavinfo                          diag_mhs_n18_ges.2014070706        fit_w1.2014070706    gsi.exe
atms_beamwidth.txt                diag_mhs_n19_anl.2014070706        fort.201             gsiparm.anl
berror_stats                      diag_mhs_n19_ges.2014070706        fort.202             gsnd1bufr
bftab_sstphr                      diag_pcp_tmi_trmm_anl.2014070706   fort.203             hirs3bufrears
convinfo                          diag_pcp_tmi_trmm_ges.2014070706   fort.204             hirs4bufr
cost_gradient.txt                 diag_sbuv2_n19_anl.2014070706      fort.205             iasibufr
diag_airs_aqua_anl.2014070706     diag_sbuv2_n19_ges.2014070706      fort.206             list_direcotry
diag_airs_aqua_ges.2014070706     diag_seviri_m10_anl.2014070706     fort.207             mhsbufr
diag_amsua_aqua_anl.2014070706    diag_seviri_m10_ges.2014070706     fort.208             ozinfo
diag_amsua_aqua_ges.2014070706    diag_sndrd1_g13_anl.2014070706     fort.209             pcpbias_out
diag_amsua_metop-a_anl.2014070706 diag_sndrd1_g13_ges.2014070706     fort.210             pcpinfo
diag_amsua_metop-a_ges.2014070706 diag_sndrd2_g13_anl.2014070706     fort.211             prepbufr
diag_amsua_n15_anl.2014070706     diag_sndrd2_g13_ges.2014070706     fort.212             prepobs_prep.bufrtable
diag_amsua_n15_ges.2014070706     diag_sndrd3_g13_anl.2014070706     fort.213             satbias_angle
diag_amsua_n18_anl.2014070706     diag_sndrd3_g13_ges.2014070706     fort.214             satbias_in
diag_amsua_n18_ges.2014070706     diag_sndrd4_g13_anl.2014070706     fort.215             satbias_out
diag_amsua_n19_                                                                           satinfo
diag_amsua_n19_                                                                           satwndbufr
diag_conv_anl.2                                                                           sbuvbufr
diag_conv_ges.2                                                                           scaninfo
diag_gome_metop-a_anl.2014070706  diag_ssmis_img_f16_ges.201                              seviribufr
diag_gome_metop-a_ges.2014070706  diag_ssmis_las_f16_anl.2014070706  fort.223             sfcanl.gsi
diag_hirs4_metop-a_anl.2014070706 diag_ssmis_las_f16_ges.2014070706  fort.224             siganl
diag_hirs4_metop-a_ges.2014070706 diag_ssmis_uas_f16_anl.2014070706  fort.225             ssmisbufr
diag_hirs4_n19_anl.2014070706     diag_ssmis_uas_f16_ges.2014070706  fort.226             stdout
diag_hirs4_n19_ges.2014070706     errtable                           fort.227             stdout.anl.2014070706
diag_iasi_metop-a_anl.2014070706  filter_fort220.ksh                 fort.228             tmirrbufr
diag_iasi_metop-a_ges.2014070706  fit_p1.2014070706                  fort.229
```

sfcanl.gsi -- global analysis background for surface
siganl -- global analysis background for the atmosphere

Step 6: Run Completion

stdout: Reading in global background

- *Indication GSI is reading the global background fields:*

```
[000]gsisub(): : starting …
GESINFO:  Read NCEP sigio format file, sigf06
GESINFO:  jcap_b= 574, levs= 64, latb=  576, lonb= 1152, ntrac=  3, ncldt=  1, idvc=  2,
nvcoord=  2, idvm=  0, idsl=  0, idpsfc=  0, idthrm=  0                                  06
… …
READ_FILES:  analysis date,minutes        2014        7
READ_FILES:  atm guess file, nming2    3.000000000000000                                  20
READ_FILES:  atm guess file, nming2    6.000000000000000                                  00
READ_FILES:  atm guess file, nming2    9.000000000000000                                  80
… …
READ_FILES:  sfc guess file, nming2    3.000000000000000    0    7    7    2014    19204020
READ_FILES:  sfc guess file, nming2    6.000000000000000    0    7    7    2014    19204200
READ_FILES:  sfc guess file, nming2    9.000000000000000    0    7    7    2014    19204380
GESINFO:  Guess    date is        0        7        7        2014    6.000000000000000
GESINFO:  Analysis date is        2014        7        7        6        0   2014070706   3.000000000000000
READ_FILES:  atm fcst files used in analysis  :                                          00
3.000000000000000    6.000000000000000
READ_FILES:  sfc fcst files used in analysis:
3.000000000000000    6.000000000000000                    2
```

GFS background file

GFS grid information

GFS atmosphere guess file: 3,6,9 hour forecast initialized at 2014070700

GFS surface guess file: 3,6,9 hour forecast initialized at 2014070700

Guess date - 2014070700
Analysis date - 2014070706

Step 5: Run Completion

stdout: Reading in observational data

- In the middle of the stdout file:

```
OBS_PARA: ps                 490    5481     731    2042     138    3833     380     4410
OBS_PARA: t                  556    7521    1608    3493      75    8881     350     7267
OBS_PARA: q                  338    5489     579    2229      42    3754     278     3763
OBS_PARA: pw                   6       1       0       2       1     134       0       81
OBS_PARA: uv               13658   13374    5273    5827    7951    3522    9998    10606
OBS_PARA: uv                1398    8149    1934    3947     970   11590     351     7766
OBS_PARA: gps_ref           4924    4138    1899    3327    3234    3757    2582     3588
OBS_PARA: pcp_tmi   trmm       0     712       0      71     362       0     623      102
OBS_PARA: hirs4     metop-a    9       0       0       1      17      15       3        0
OBS_PARA: hirs4     n19        1       6      17      14       2       0       0        7
OBS_PARA: hirs4     metop-b   16      12       0       5       0      13       0        0
OBS_PARA: airs      aqua       0       4      14      10       0       0       0       13
OBS_PARA: amsua     n15        1       4       8      17       7       0       3        2
OBS_PARA: amsua     n18        1       6       2       1       5       0      17       12
OBS_PARA: amsua     n19        0       4      15      15       1       0       0        7
OBS_PARA: amsua     metop-a    9       0       0       1      15      14       3        0
```

… …

Observation Type

Distribution of observations in each sub-domain (8)

Only showing part of the satellite radiances data!

Step 5: Run Completion

stdout: optimal iteration

2 outer loops with 50 inner iterations

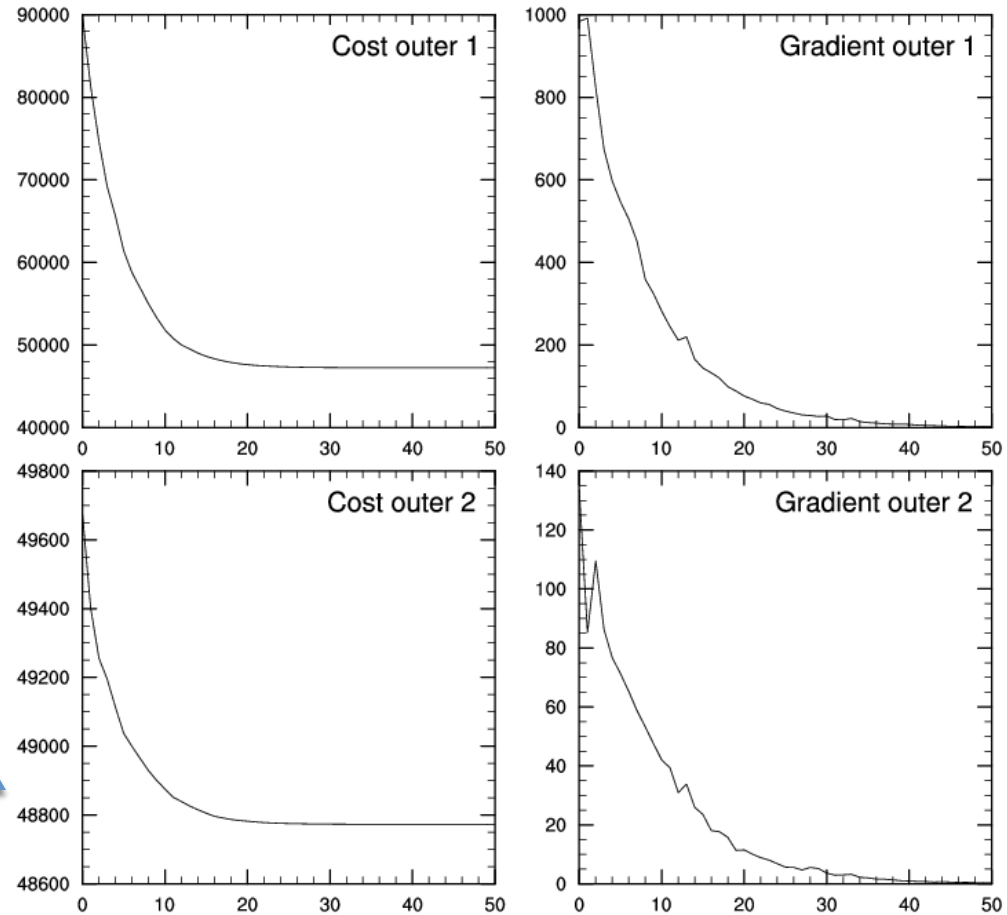
```
 GLBSOI:  START pcgsoi jiter=          1
pcgsoi: gnorm(1:2),b=  9.681685162080464652E+05  9.681685162080466980E+05
0.000000000000000000E+00
Initial cost function =  8.986310792231539381E+04
Initial gradient norm =  9.839555458495299263E+02
cost,grad,step,b,step? =   1   0  8.986310792231539381E+04  9.839555458495299263E+02
8.844696541982170063E-03  0.000000000000000000E+00  good
 … …
cost,grad,step,b,step? =   2  48  4.877345111329654173E+04  4.179057161074568039E-01
1.124861681908470752E-02  9.664540442600634274E-01  good
pcgsoi: gnorm(1:2),b=  1.085430275613184642E-01  1.085430279371430023E-01
6.215059770987519938E-01
cost,grad,step,b,step? =   2  49  4.877344914877974225E+04  3.294586887021170307E-01
1.655841298948613691E-02  6.215059770987519938E-01  good
pcgsoi: gnorm(1:2),b=  1.125369599403734100E-01  1.125369597618661732E-01
1.036795842996837802E+00
cost,grad,step,b,step? =   2  50  4.877344735147946631E+04  3.354652887265289207E-01
5.569972100056517182E-03  1.036795842996837802E+00  good
update_guess: successfully complete
```

Iteration check: The J value should descend through iterations

Step 8: Checking Minimization

- To gain a complete picture of the minimization process: plot cost function and norm of gradient
  - Script available in v3.4 release: *./util/Analysis_Utilities/plot_ncl/GSI_cost_gradient.ncl*

Cost function and norm of gradient descend in both outer loops



Summary

- Steps to running a successful GSI Analysis:

1. Obtain background field
2. Grab desired observational data
3. Choose the appropriate anavinfo file (and modify it) to have the same vertical levels as the GSI background
4. Modify run script to suit specific application and properly link observational data
   - ✓ Additional steps specific to observational data (e.g: thinning and bias correction for radiance)
5. Run GSI
6. Check run status and completion of each step of the GSI analysis *(stdout)*
7. Diagnose analysis results *(fit files)*
8. Check analysis increment, cost function/norm of gradient *(DTC graphics utilities available)*

Questions?

gsi_help@ucar.edu