GSI Fundamentals (4): Applications

Kathryn M. 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): Applications
  - ✓ Where to obtain observation data
  - ✓ Successfully set up GSI for various data sources
    - ✓ Conventional observations
    - ✓ Radiance data
    - ✓ GPS RO data
  - ✓ Learn to check run status
  - ✓ Learn to understand diagnostics in the context of a particular data source
  - ✓ Ensure the run was successful!
- This talk is tailored to Chapter 5 of the GSI User's Guide for community release v3.1 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. Modify run script to properly link observational data
   - ✓ Additional steps specific to observational data (e.g: thinning and bias correction for radiance)
4. Run GSI
5. Check run status and completion of each step of the GSI analysis *(stdout)*
6. Diagnose analysis results *(fit files)*
7. 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_v3.1.php

(practice case three)

Step 1 and 2: Case data

- 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>_<yyyy-mm-dd_hh:mm:ss>

- Obtained from the GFS forecast through WRF WPS/REAL

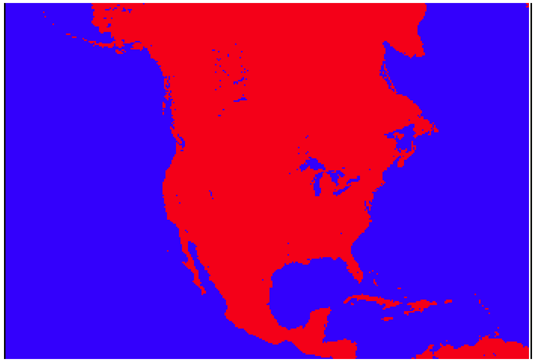


Fig: Landmask of case study background

2) Real-time and archived observation data available *(refer to 'Community Tools (1): PrepBUFR/BUFR for more detail)*

- Case Study data available at:

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

Conventional Observation Assimilation

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

Step 3: Run Script

run_gsi.ksh

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

Experimental Setup

```
WORK_ROOT=/ptmp/test/gsiprd_${ANAL_TIME}_prepbufr
BK_FILE=/ptmp/GSI/data/DTC/NA30km/bk/wrfinput_d01_2011-03-22_12:00:00
OBS_ROOT=/ptmp/GSI/data/DTC/NA30km/obs20110322
PREPBUFR=/ptmp/GSI/data/DTC/NA30km/obs20110322/nam.t12z.prepbufr.tm00.nr
FIX_ROOT=/blhome/GSI/comGSI_v3.1/fix
CRTM_ROOT=/ptmp/GSI/CRTM/CRTM_Coefficients-2.0.5
GSI_EXE=/blhome/GSI/comGSI_v3.1/run/gsi.exe
bk_core=ARW
bkcv_option=NAM
if_clean=clean
```

Location of PREPBUFR data

- Namelist using default options in the sample script

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

Step 4: 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 4: Run Status

While GSI is still running…

- Check the content of the standard out file to monitor the stage of the GSI analysis:
- `be1105en% tail -f stdout`

1st outer loop

Inner iteration

```
grepcost J,Jb,Jo,Jc,Jl =   1   1  5.825495408135202160E+04  2.145440277602700405E+02
   5.804041005359175324E+04  0.000000000000000000E+00  0.000000000000000000E+00

grepgrad grad,reduction=   1   1  5.979012335995843159E+02  7.302068672950051686E-01

pcgsoi: cost,grad,step =   1   1  5.825495408135202160E+04  5.979012335995843159E+02
   2.019904649670459587E-02

pcgsoi: gnorm(1:2),b=  2.237482294472646608E+05  2.237482294472649228E+05
   6.258938849003770066E-01

  stprat   0.226513888727740

  stprat   1.049725208537512E-015
```

- Shows that GSI is in the optimal interation stage.

Step 5: Run Completion

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

```
anavinfo                        fort.206          gsiparm.anl
berror_stats                    fort.207          l2rwbufr
convinfo                        fort.208          list_run_directory
diag_conv_anl.2011032212        fort.209          ozinfo
diag_conv_ges.2011032212        fort.210          pcpbias_out
errtable                        fort.211          pcpinfo
fit_p1.2011032212               fort.212          prepbufr
fit_q1.2011032212               fort.213          prepobs_prep.bufrtable
fit_rad1.2011032212             fort.214          satbias_angle
fit_t1.2011032212               fort.215          satbias_in
fit_w1.2011032212               fort.217          satbias_out
fort.201                        fort.218          satinfo
fort.202                        fort.219          stdout
fort.203                        fort.220          stdout.anl.2011032212
fort.204                        fort.221          wrfanl.2011032212
fort.205                        gsi.exe           wrf_inout
```

- 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 5: Run Completion

stdout: Reading in namelist

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

```
GSI_4DVAR:  nobs_bins =            1
 SETUP_4DVAR: l4dvar= F
 SETUP_4DVAR: l4densvar= F
 SETUP_4DVAR: winlen=   3.00000000000000
 SETUP_4DVAR: winoff=   3.00000000000000
 SETUP_4DVAR: hr_obsbin=   3.00000000000000
```

- *Indication GSI is reading the background fields:*

stdout: Reading in background field

```
end_index= 348     247     50     0
max,min XLAT(:,1)= 13.41553 2.813988
max,min XLAT(1,:)= 47.02686 2.813988
xlat(1,1),xlat(nlon,1)= 2.813988 4.969986
xlat(1,nlat),xlat(nlon,nlat)= 47.02686 52.26216
rmse_var=XLONG
...............
rmse_var=U
ordering=XYZ
WrfType,WRF_REAL= 104     104
ndim1=     3
staggering= N/A
start_index= 1     1     1     0
end_index= 349     247     50     0
```

	K	Maximum	Minimum	Central grid
k,max,min,mid U=	1	20.05023	-21.65548	-6.996003
k,max,min,mid U=	2	20.64079	-22.58930	-7.982791
k,max,min,mid U=	3	21.84538	-24.38444	-9.791903
k,max,min,mid U=	4	24.33893	-27.59095	-12.01432
k,max,min,mid U=	5	27.30596	-29.83475	-14.94501

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

Step 5: Run Completion

stdout: Reading in observational data

- In the middle of the stdout file:

OBS_PARA: ps	2352	2572	8367	2673
OBS_PARA: t	4617	4331	12418	4852
OBS_PARA: q	3828	3908	11096	3632
OBS_PARA: pw	89	31	141	23
OBS_PARA: uv	5704	4835	15025	4900
OBS_PARA: sst	0	0	2	0

Observation Type

Distribution of observations in each sub-domain

✓ This table is important to see if the observations have been read in properly

Step 5: Run Completion

stdout: optimal iteration

The namelist specified 2 outer loops with 50 inner loops

- The optimal interation step will look as follows:

```
Minimization iteration          0

grepcost J,Jb,Jo,Jc,Jl =   1   0  7.024833842077980808E+04  0.000000000000000000E+00  7.024833842077980808E+04
   0.000000000000000000E+00  0.000000000000000000E+00

grepgrad grad,reduction=   1   0  8.188107512799259666E+02  1.000000000000000000E+00

pcgsoi: cost,grad,step =   1   0  7.024833842077980808E+04  8.188107512799259666E+02  1.788853101746809213E-02

pcgsoi: gnorm(1:2),b=  3.574858851399047417E+05  3.574858851399045670E+05  5.332020690447850653E-01

  stprat   0.114387353859176

  stprat   4.672705475826555E-016
```

- … last iteration:

The iteration met the stop threshold before meeting the maximum iteration

```
Minimization iteration          40

grepcost J,Jb,Jo,Jc,Jl =   2  40  3.664942287987162126E+04  6.573582409713609195E+03  3.007584047015801480E+04
   0.000000000000000000E+00  0.000000000000000000E+00

grepgrad grad,reduction=   2  40  7.094588989162899789E-03  2.206814011091630791E-04

pcgsoi: cost,grad,step =   2  40  3.664942287987162126E+04  7.094588989162899789E-03  4.844607290004876443E-02

 PCGSOI: WARNING **** Stopping inner iteration ***

 gnorm  0.750736287079361097E-10 less than  0.100000000000000004E-09

 Minimization final diagnostics
```

Iteration check: The J value should descend through each iteration

Step 5: Run Completion

stdout: Write out analysis results

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

```
max,min MU=    3993.281     -2043.336
rmse_var=MU
ordering=XY
WrfType,WRF_REAL=   104      104
ndim1=     2
staggering= N/A
start_index=    1           1           1           0
end_index1=   348         247          50           0
k,max,min,mid T=          1    310.9237       233.9436       280.6658
k,max,min,mid T=          2    311.3734       235.8086       280.8300
k,max,min,mid T=          3    311.6932       238.1501       281.0092
k,max,min,mid T=          4    312.7413       243.5058       281.4198
k,max,min,mid T=          5    313.2800       242.0858       282.9984
```

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

stdout: Successful GSI run

```
ENDING DATE-TIME    JUL 14,2012  23:50:12.367  196  SAT   2456123
    PROGRAM GSI_ANL HAS ENDED.  IBM RS/6000 SP
* . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * .
```

✓**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 6: 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	185	527	573	995	1117	637	226	443	642	979	855	8692
o-g 01	t	120	0000	bias	0.62	0.88	-0.22	-0.20	-0.19	-0.38	-1.07	-0.89	-0.95	-0.99	-1.63	-0.92
o-g 01	t	120	0000	rms	2.62	2.57	2.01	1.31	0.90	1.02	1.66	1.71	1.94	1.86	2.67	2.27
o-g 01	t	130	0000	count	0	0	0	0	0	11	177	626	67	0	0	881
o-g 01	t	130	0000	bias	0.00	0.00	0.00	0.00	0.00	0.18	0.06	-0.02	-1.23	0.00	0.00	-0.09
o-g 01	t	130	0000	rms	0.00	0.00	0.00	0.00	0.00	0.96	0.97	1.48	2.28	0.00	0.00	1.46
o-g 01	t	180	0000	count	1260	28	0	0	0	0	0	0	0	0	0	1288
o-g 01	t	180	0000	bias	0.67	0.80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.68
o-g 01	t	180	0000	rms	1.76	1.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.75
o-g 01		all		count	1445	555	573	995	1117	648	403	1069	709	979	855	10861
o-g 01		all		bias	0.67	0.88	-0.22	-0.20	-0.19	-0.37	-0.57	-0.38	-0.97	-0.99	-1.63	-0.66
o-g 01		all		rms	1.89	2.52	2.01	1.31	0.90	1.02	1.40	1.58	1.97	1.86	2.67	2.16

O-B

Data types used:
120: rawinsonde
130: AIREP/PIREP aircraft
180: surface marine

Data type 120 has 1117 obs in level 400.0-600.0 mb, bias=-0.19, rms =0.90

Whole atmosphere/all data: O-A – 10861 obs, bias = -0.66, rms = 2.16

Step 6: Analysis fit to Observations

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

o-g	01	all	count	1445	555	573	995	1117	648	403	1069	709	979	855	10861
o-g	01	all	bias	0.67	0.88	-0.22	-0.20	-0.19	-0.37	-0.57	-0.38	-0.97	-0.99	-1.63	-0.66
o-g	01	all	rms	1.89	2.52	2.01	1.31	0.90	1.02	1.40	1.58	1.97	1.86	2.67	2.16

O-B

o-g	03	all	count	1445	555	573	995	1117	648	403	1069	709	979	855	10861
o-g	03	all	bias	0.38	0.50	-0.15	-0.04	-0.02	0.03	-0.15	-0.03	-0.06	-0.17	-0.31	-0.03
o-g	03	all	rms	1.56	2.01	1.60	1.00	0.61	0.59	0.84	1.06	1.29	1.36	1.93	1.43

O-A

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

Check other parameters!

➢ 10861 total observations: from the background to the analysis the bias reduced from **-0.66 to -0.03** & rms reduced from **2.16 to 1.43**. ~34% 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 6: 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 ''penalty,grad ,a,b=' fort.220 | sed -e 's/penalty,grad ,a,b=//g' >
cost_gradient.txt
```

- *cost_gradient.txt* will have 6 columns (4 shown below)

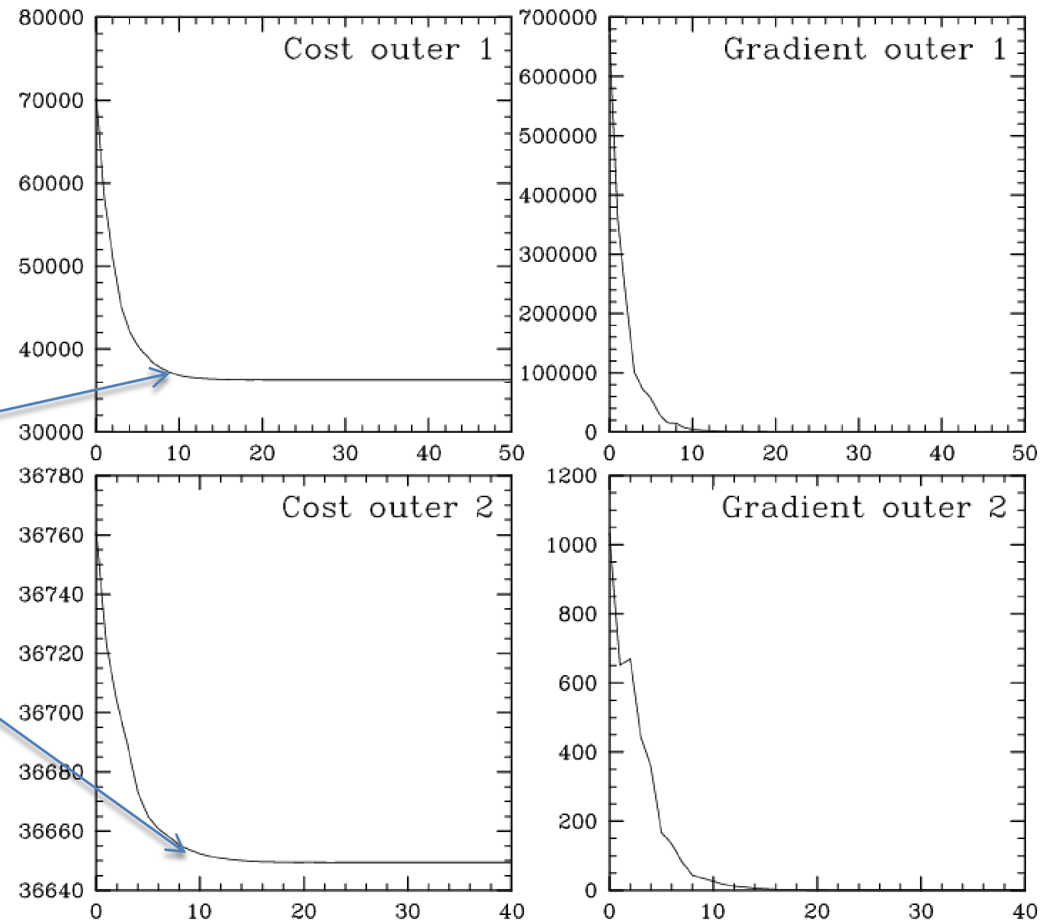
1	0	0.702483384207798081E+05	0.670451046411596704E+06
1	1	0.582549540813520216E+05	0.357485885139904742E+06
1	2	0.510340800654954874E+05	0.223748229447264661E+06
1	3	0.451910522149672179E+05	0.100693261241731772E+06
1	4	0.421994383952682037E+05	0.726689181584937469E+05
1	5	0.403738674444606295E+05	0.565482157493619452E+05
…			
2	35	0.366494229475844040E+05	0.654227325952160183E-03
2	36	0.366494229180964758E+05	0.328958749059858212E-03
2	37	0.366494229007449394E+05	0.215132065808595375E-03
2	38	0.366494228897850917E+05	0.140109917756355446E-03
2	39	0.366494228834806418E+05	0.873987085206934423E-04
2	40	0.366494228798716213E+05	0.503331929251514545E-04

First 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 0.70 E+05 to 0.37 E+05
Norm of gradient reduced
from 0.67 E+06 to 0.50 E-04

Step 6: Checking Minimization

- To gain a complete picture of the minimization process: plot cost function and norm of gradient
  - Script available in v3.1 release: *./util/Analysis_Utilities/plot_ncl/GSI_cost_gradient.ncl*

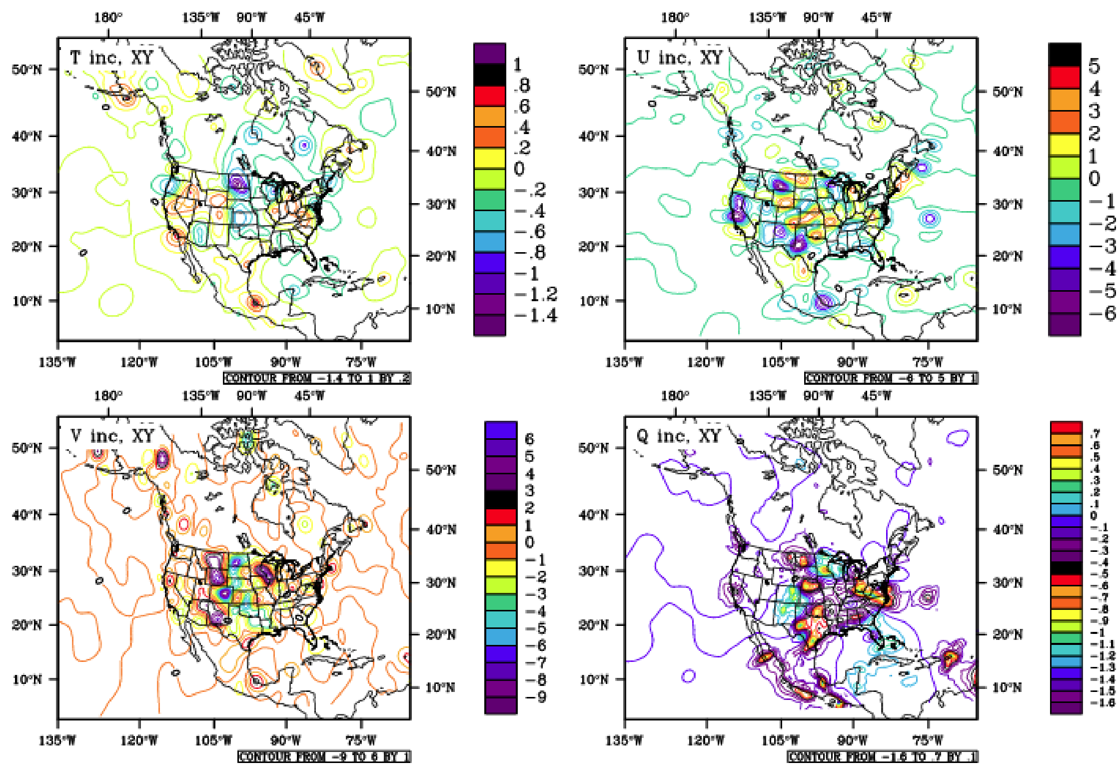


Cost function and norm of gradient descend in both outer loops

Step 7: 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 v3.1 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 3: Run Script *run_gsi.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.t12z.1bamua.tm00.bufr_d*

  AMSU-B: *gdas1.t12z.1bamub.tm00.bufr_d*

  HRS4: *gdas1.t12z.1bhrs4.tm00.bufr_d*

- The location of these data is indicated in *OBS_ROOT***
- Insert below link to PREPBUFR data in *run_gsi.ksh*:

```
ln -s ${OBS_ROOT}/gdas1.t12z.1bamua.tm00.bufr_d amsuabufr
ln -s ${OBS_ROOT}/gdas1.t12z.1bamub.tm00.bufr_d amsubbufr
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_ROOT:*

```
dfile(30)='amsuabufr',dtype(30)='amsua',dplat(30)='n17',dsis(30)='amsua_n17',dval(30)=0.0,dthin(30)=2,
```

  - The AMSU-A observation from NOAA-17 will be read in from BUFR file '**amsuabufr**'

Step 3: Radiance Data Thinning

- Radiance data thinning is setup in the namelist section *&OBS_ROOT:*

```
dmesh(1)=120.0,dmesh(2)=60.0,dmesh(3)=60.0,dmesh(4)=60.0,dmesh(5)=120

dfile(30)='amsuabufr',dtype(30)='amsua',dplat(30)='n17',dsis(30)='amsua_n17',dval
(30)=0.0,dthin(30)=2,
```

- *&OBS_ROOT* has thinning grid array `dmesh`
- For each data type line, the last column:'`dthin(30)=2`', is used to select the mesh grid used in the thinning.
- In this case, data thinning for NOAA-17 AMSU-A observation is 60 km

Step 3: Radiance Bias Correction

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

```
SATANGL=${FIX_ROOT}/global_satangbias.txt ./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 User's Guide v3.1 – section 8.4*

Step 5: Run Completion *stdout: Reading in observational data*

- ✓ *[While GSI is running]* Working directory will look same as 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                      2352      2572      8367      2673
OBS_PARA: t                       4617      4331     12418      4852
OBS_PARA: q                       3828      3908     11096      3632
OBS_PARA: pw                        89        31       141        23
OBS_PARA: uv                      5704      4835     15025      4900
OBS_PARA: sst                        0         0         2         0
OBS_PARA: hirs4      metop-a         0         0        28        35
OBS_PARA: amsua      n15          2564      1333       133       195
OBS_PARA: amsua      n18          1000      2117         0        90
OBS_PARA: amsua      metop-a         0         0        58        67
OBS_PARA: amsub      n17             0         0        57        70
OBS_PARA: hirs4      n19           246      1097         0        37
OBS_PARA: amsua      n19           656      3503         0        93
```

7 new radiance data types have been read in

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

Step 6: Diagnosing Analysis Results

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

	it	satellite		instrument	#read	#keep	#assim	penalty	qcpnlty	cpen	qccpen
o-g	01	rad	n15	amsua	128055	53932	14552	24875.	24875.	1.7094	1.7094
o-g	01	rad	n17	amsub	213920	254	0	0.000	0.0000	0.0000	0.0000
…											
o-g	03	rad	n15	amsua	128055	53932	36345	9993.3	9993.3	0.27496	0.27496
o-g	03	rad	n17	amsub	213920	254	0	0.0000	0.0000	0.0000	0.0000

Penalty

O-B

within the analysis time window and domain

after thinning

used in analysis

O-A

✓ The penalty for n15 decreased from 24875 to 9993.3 after 2 outer loops

✓ n17 (amsub) had 213920 within the analysis time window and domain…254 after thinning…none used in analysis

✓ When checking values: number passing quality checks similar, but final penalty smaller

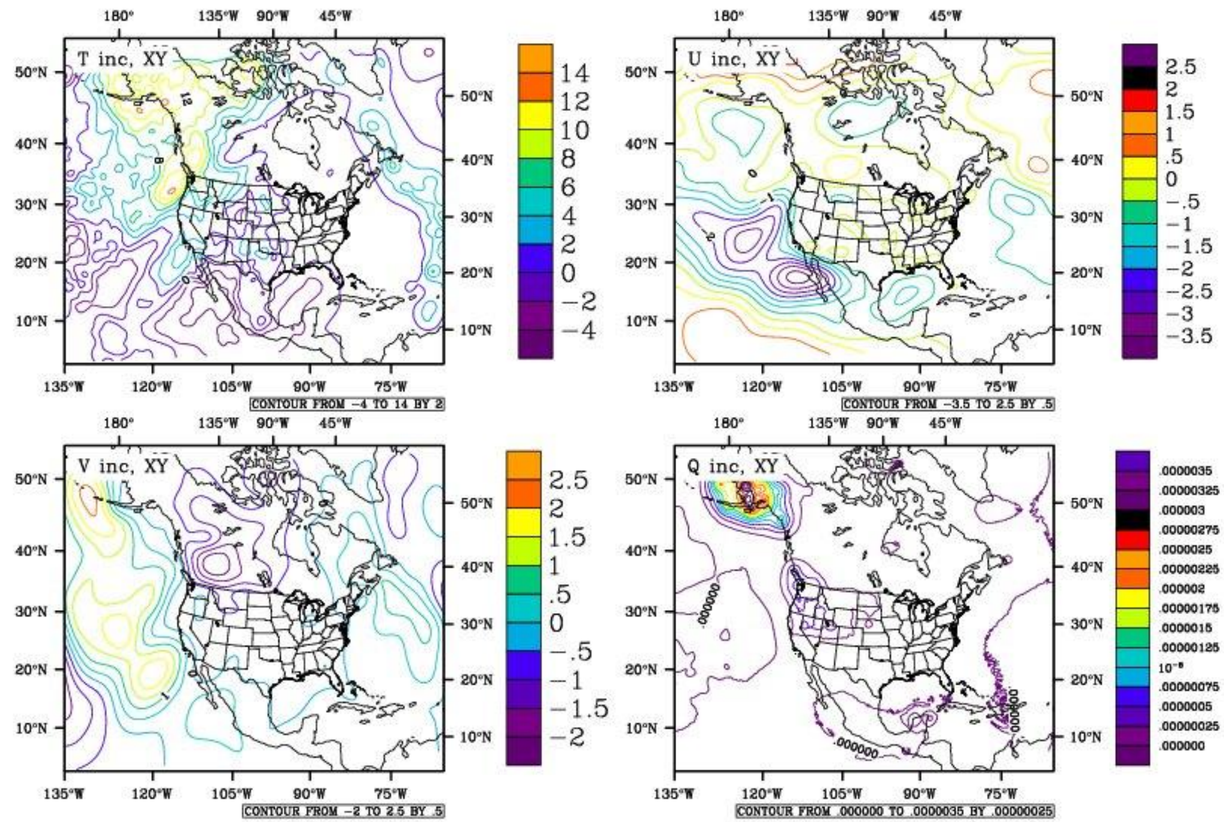
Statistics can also be viewed for each channel in fort.207

Step 7: Checking Analysis Impact

- Analysis increment plotted comparing the analysis results with radiance & conventional and conventional only.
  - Graphic tool available in the v3.1 release: *./util/Analysis_Utilities/plot_ncl/Analysis_increment.ncl*

Analysis Increment at the 49th Level

Increment = (AMSU-A+ PREPBUFR) - PREPBUFR



✓ Impact of radiance data compared to conventional alone 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 3: Run Script *run_gsi.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_ROOT***
- Insert below link to PREPBUFR data in *run_gsi.ksh*:

```
ln -s ${OBS_ROOT}/gdas1.t12z.gpsro.tm00.bufr_d gpsrobufr
```

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

```
dfile(10)='gpsrobufr',dtype(10)='gps_ref',dplat(10)='',dsis(10)='gps',dval(10)=
   1.0,dthin(10)=0,
```

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

Step 5: 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	2352	2572	8367	2673
OBS_PARA: t	4617	4331	12418	4852
OBS_PARA: q	3828	3908	11096	3632
OBS_PARA: pw	89	31	141	23
OBS_PARA: uv	5704	4835	15025	4900
OBS_PARA: sst	0	0	2	0
OBS_PARA: gps_ref	3538	5580	2277	6768

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

Step 6: Diagnosing Analysis Results

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

				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		all		count	4	87	155	548	942	675	411	497	618	855	1416	8709
o-g 01		all		bias	-0.50	-0.27	-0.18	-0.19	-0.11	-0.03	0.12	0.16	0.09	0.07	-0.06	-0.07
o-g 01		all		rms	0.77	0.87	0.83	1.02	0.74	0.43	0.44	0.50	0.54	0.62	0.57	0.66

O-B

most GPS RO observations located in upper levels

o-g 03		all		count	5	108	192	615	950	673	413	500	631	868	1417	8891
o-g 03		all		bias	-0.31	-0.11	-0.06	-0.01	-0.01	-0.04	0.00	0.01	0.01	-0.01	-0.01	0.01
o-g 03		all		rms	0.43	0.62	0.79	0.77	0.54	0.29	0.20	0.21	0.24	0.28	0.40	0.48

O-A

✓ 8709 obs used in analysis during 1st outer loop, 8891 used to calculate O-A

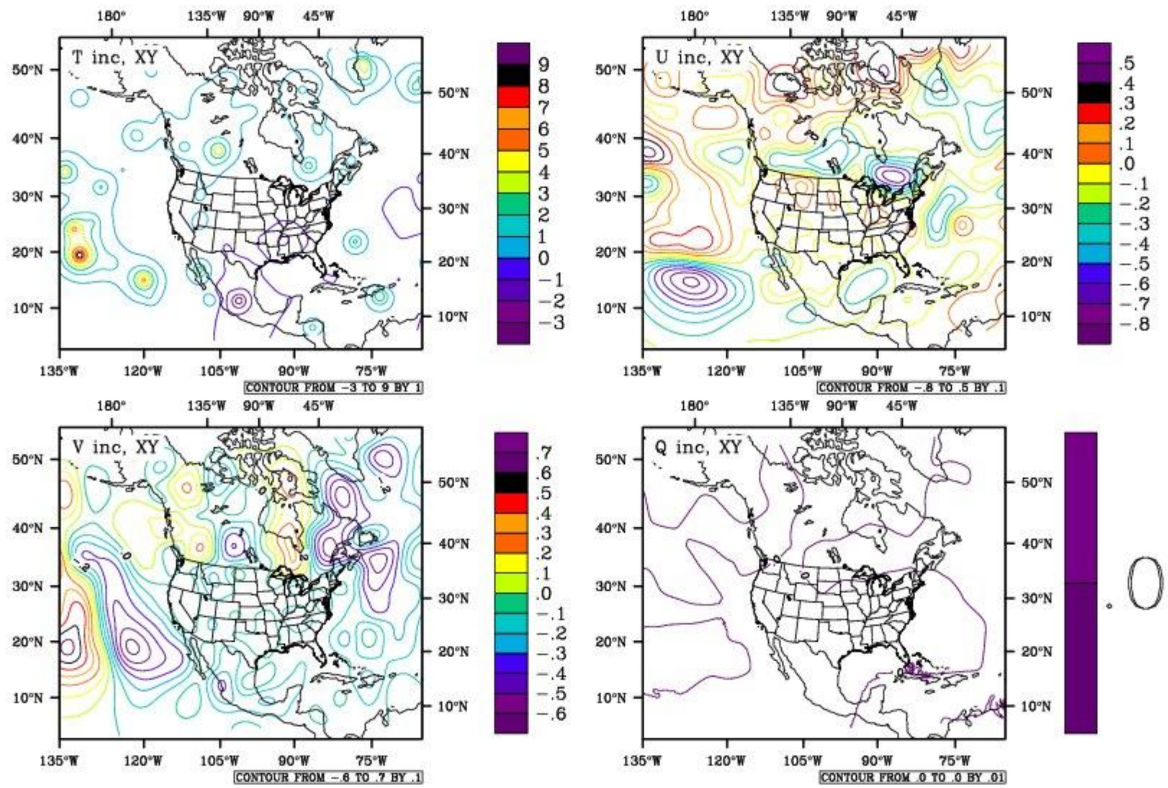
✓ Bias -0.07 to 0.01 after analysis. RMS reduced from 0.66 to 0.48 after analysis.

Step 7: Checking Analysis Impact

- Analysis increment plotted comparing the analysis results with GPS RO & conventional and conventional only.
  - Graphic tool available in v3.1 release: *./util/Analysis_Utilities/plot_ncl/Analysis_increment.ncl*

Analysis Increment at the 48th Level

Increment = (GPS RO + PREPBUFR) - PREPBUFR



✓ Impact of GPS RO data apparent over U.S. CONUS domain

Summary

- Steps to running a successful GSI Analysis:

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

➢ This case study is available at: http://www.dtcenter.org/com-GSI/users/tutorial/online_tutorial/index_v3.1.php *(practice case three)*

Questions?

gsi_help@ucar.edu