2017 GSI Community Tutorial
July 11-14, 2017. College Park, MD

GSI Fundamentals (3): Diagnostics

Guoqing Ge,

Kathryn Newman, Ming Hu & Chunhua Zhou

Developmental Testbed Center (DTC)

Developmental Testbed Center

Outline

- GSI fundamentals (1): Setup and Compilation
- GSI fundamentals (2): Run and Namelist
- GSI fundamentals (3): Diagnostics
  - Standard output
  - Observation innovation statistics and binary diagnostic files
  - Convergence information
  - Analysis increments
- GSI fundamentals (4): BE, Obs Error
- GSI fundamentals (5): Prep BUFR and BUFR format
- GSI fundamentals (6): Review and Applications

Standard Output (stdout)

Details in User's Guide Section 4.1

✓Critical information about the GSI analysis can be obtained

✓Users can check:

1. Did GSI successfully complete?
2. Does the optimal minimization look correct?
3. Are the background, analysis and increment fields reasonable?

✓why my GSI run failed?

stdout: structure

- The structure of stdout is as follows:

1. Read in prerequisite info and get prepared for doing analysis:
   1) Read in configuration (namelist)
   2) Read in background
   3) Read in observations
   4) Partition domain and data for parallel analysis
   5) Read in constant fields (fixed files)

2. Optimal minimization (analysis)

3. Save analysis result

stdout: start of GSI running

```
* . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * .
     PROGRAM GSI_ANL HAS BEGUN. COMPILED 1999232.55     ORG: NP23
     STARTING DATE-TIME  JUL 06,2017  13:59:14.782  187  THU   2457941
```

stdout:

Initialize
met_guess variables
State variables
Control variables

Controlled by the
anavinfo file

```
 gsi_metguess_mod*init_:  2D-MET STATE VARIABLES:
+--  2 lines: ps----------------------------------------
 gsi_metguess_mod*init_:  3D-MET STATE VARIABLES:
+--  8 lines: u-----------------------------------------
 gsi_metguess_mod*init_: ALL MET STATE VARIABLES:
 u
 v
 div
 vor
 tv
 q
 oz
 cw
 ps
 z
 state_vectors*init_anasv:  2D-STATE VARIABLES ps
 sst
 state_vectors*init_anasv:  3D-STATE VARIABLES u
+--  4 lines: v                                  tv
 state_vectors*init_anasv: ALL STATE VARIABLES u
 v                                 tv
 tsen                              q
 oz                                cw
 prse                              ps
 sst
 control_vectors*init_anacv: 2D-CONTROL VARIABLES ARE
 ps                                sst
 control_vectors*init_anacv: 3D-CONTROL VARIABLES ARE
+--  3 lines: sf                                 vp
 control_vectors*init_anacv: MOTLEY CONTROL VARIABLES
 stl                               sti
 control_vectors*init_anacv: ALL CONTROL VARIABLES
 sf                                vp
 ps                                t
 q                                 oz
 sst                               cw
 stl                               sti
```

stdout:

Read in and
Print out namelist
configurations

```
 GSI_4DVAR:  nobs_bins =            3
 SETUP_4DVAR: nobs_bins =    3, ntlevs_ens =    1
 SETUP_4DVAR: allocate array containing time levels for ensemble
 SETUP_4DVAR: timelevel =     1 , ens_fhrlevs =     6
+-- 34 lines: SETUP_4DVAR: l4dvar= F-----------------------------------------------
 &SETUP
+--147 lines: GENCODE =   78.0000000000000     ,-------------------------------------
 /
 &GRIDOPTS
+-- 26 lines: JCAP    =         62,-----------------------------------------------
 /
 &BKGERR
 VS       =   1.00000000000000     ,
+-- 18 lines: NHSCRF  =           3,-----------------------------------------------
 /
 &ANBKGERR
 ANISOTROPIC     = F,
+-- 23 lines: ANCOVMDL        =           0,---------------------------------------
 /
 &JCOPTS
+--  8 lines: LJCDFI  = F,-------------------------------------------------------
 /
 &STRONGOPTS
+--  8 lines: REG_TLNMC_TYPE  =           1,---------------------------------------
 /
 &OBSQC
+-- 29 lines: DFACT   =  0.750000000000000     ,-------------------------------------
 /
 EXT_SONDE on type 120 = T
  ngroup =            2  dmesh =    120.000000000000
   60.0000000000000
+-- 81 lines: prepbufr             ps                    ps
 &SUPEROB_RADAR
+--  8 lines: DEL_AZIMUTH     =   5.00000000000000     ,----------------------------
 /
 &LAG_DATA
+--  8 lines: LAG_ACCUR       =  1.000000000000000E-006,---------------------------
 /
 &HYBRID_ENSEMBLE
+-- 34 lines: L_HYB_ENS       = F,-----------------------------------------------
 /
 &RAPIDREFRESH_CLDSURF
+-- 38 lines: DFI_RADAR_LATENT_HEAT_TIME_PERIOD       =   30.0000000000000     ,----
 /
 &CHEM
+-- 26 lines: BERROR_CHEM     = F,-----------------------------------------------
 /
 &NST
+--  6 lines: NST_GSI =           0,-----------------------------------------------
 /
```

stdout: check background input

```
dh1  =              3
iy,m,d,h,m,s= 2017        5      13     12          0
nlon,lat,sig_regional=  332      215       50
rmse var = T
ndim1 =  3  dh1 =  3
WrfType =  104 ierr  =  0
ordering = XYZ staggering =  N/A
start_index =        1              1          1          0
end_index =        332            215         50          0
```

k,max,min,mid T=	1	321.6354	280.1795	308.9041
k,max,min,mid T=	2	321.6452	281.1811	308.9584
k,max,min,mid T=	3	321.4893	282.7797	309.0582
…				
k,max,min,mid T=	48	624.4512	576.9553	596.8545
k,max,min,mid T=	49	649.7656	613.7888	630.5093
k,max,min,mid T=	50	680.9388	653.7093	668.4511
	K	Maximum	Minimum	Domain Center

Repeats for all fields at each vertical level: `P_TOP, ZNU, ZNW, RDX, RDY, MAPFAC_M, XLAT, XLONG, MUB, MU, PHB, QVAPOR, U, V, LANDMASK, SEAICE, SST,IVGTYP, ISLTYP, VEGFRA, SNOW, U10, V10, SMOIS, TSLB, TSK …`

stdout: check convinfo

```
READ_CONVINFO: tcp      112    0  1   3.00000        0   0   0   75.0000       5.00000
READ_CONVINFO: ps       120    0  1   3.00000        0   0   0   4.00000       3.00000
READ_CONVINFO: ps       132    0 -1   3.00000        0   0   0   4.00000       3.00000
+--178 lines: READ_CONVINFO: ps       180    0  1   3.00000        0   0   0   4.00000
READ_CONVINFO: gps      440    0 -1   3.00000        0   0   0   10.0000       10.0000
READ_CONVINFO: pm2_5    102    0 -1   1.00000        0   0   0   100.000       1.50000
READ_CONVINFO: pm10     102    0 -1   1.00000        0   0   0   150.000       1.50000
READ_CONVINFO: gps      750    0 -1   3.00000        0   0   0   10.0000       10.0000
READ_CONVINFO: gps      751    0 -1   3.00000        0   0   0   10.0000       10.0000
READ_CONVINFO: gps      752    0 -1   3.00000        0   0   0   10.0000       10.0000
READ_CONVINFO: gps      753    0 -1   3.00000        0   0   0   10.0000       10.0000
READ_CONVINFO: gps      754    0 -1   3.00000        0   0   0   10.0000       10.0000
READ_CONVINFO: gps      755    0 -1   3.00000        0   0   0   10.0000       10.0000
READ_CONVINFO: gps      724    0 -1   3.00000        0   0   0   10.0000       10.0000
READ_CONVINFO: gps      725    0 -1   3.00000        0   0   0   10.0000       10.0000
READ_CONVINFO: gps      726    0 -1   3.00000        0   0   0   10.0000       10.0000
READ_CONVINFO: gps      727    0 -1   3.00000        0   0   0   10.0000       10.0000
READ_CONVINFO: gps      728    0 -1   3.00000        0   0   0   10.0000       10.0000
READ_CONVINFO: gps      729    0 -1   3.00000        0   0   0   10.0000       10.0000
READ_CONVINFO: gps       44    0 -1   3.00000        0   0   0   10.0000       10.0000
```

stdout: observation ingest

GSI resets file status depending on observation time & checks for consistency with usage in *satinfo* files

```
read_obs_check: bufr file date is   2017051312 prepbufr ps
read_obs_check: bufr file uv                   not available satwndbufr
read_obs_check: bufr file rw                   not available radarbufr
read_obs_check: bufr file date is   2014061700 prepbufr t
read_obs_check: bufr file date is   2014061700 prepbufr q
read_obs_check: bufr file date is   2014061700 amsuabufr amsua     n18
        •••••
data type gos_ctp            not used in info file -- do not read file prepbufr
data type hirs3_n16          not used in info file -- do not read file hirs3bufr
```

Not available at ob time

Ob time match

Not used in satinfo file

Read observations

```
READ_OBS:  read   19 hirs4    hirs4_metop-a          using ntasks=   2   0   2
READ_OBS:  read   33 mhs      mhs_n18                using ntasks=   2   2   2
READ_OBS:  read    1 ps       ps                     using ntasks=   1   2   1
READ_OBS:  read    2 t        t                      using ntasks=   1   3   1

    •••••          •••••            •••••

 READ_PREPBUFR: file=prepbufr   type=uv        sis=uv                  nread=     91930
ithin= 0 rmesh=120.000000 isfcalc= 0 ndata=    72392 ntask=  1
 READ_BUFRTOVS: file=amsuabufr  type=amsua     sis=amsua_n18           nread=     30690
ithin= 2 rmesh= 60.000000 isfcalc= 0 ndata=    25575 ntask=  1
```

stdout: Check Observations Input

Observation distribution in an analysis using 4 processors

```
OBS_PARA: ps                          2607      2878      9565      3019
OBS_PARA: t                           5172      4743     13902      5590
OBS_PARA: q                           4107      4197     11998      4090
OBS_PARA: pw                           296        92       475        83
OBS_PARA: uv                          6640      5439     18365      6147
OBS_PARA: sst
OBS_PARA: gps_ref
OBS_PARA: hirs3      n17
OBS_PARA: hirs4      metop-a             0         0       416       731
OBS_PARA: amsua      n15              2563      1323      1048      1669
OBS_PARA: amsua      n18              1002      2119         0       390
OBS_PARA: amsua      metop-a             0         0      1268      2279
OBS_PARA: amsub      n17                 0         0      1716      2891
OBS_PARA: mhs        n18              1446      2932         0       809
OBS_PARA: mhs        metop-a             0         0      1600      2839
OBS_PARA: hirs4      n19               244      1093         0       236
OBS_PARA: amsua      n19               651      3486         0       469
OBS_PARA: mhs        n19               936      4272         0       848
```

Conventional obs fairly evenly distributed

1	3
0	2

Subdomain 0 1 2 3

stdout: observation innovation

```
GLBSOI: jiter,jiterstart,jiterlast,jiterend=     1          1          2          1
```

First outer analysis loop

Calculate observation innovation for each data type in first outer loop:

Set up RHS of analysis equation

```
SETUPALL:,obstype,isis,nreal,nchanl=ps        ps                    22
          0
SETUPALL:,obstype,isis,nreal,nchanl=t         t                     24
          0
           •••••
 SETUPALL:,obstype,isis,nreal,nchanl=gps_ref   gps                   16
          0
 SETUPALL:,obstype,isis,nreal,nchanl=amsua     amsua_n15             33
         15
crtm_interface*init_crtm: crtm_init() on path "./"
 ACCoeff ReadFile (Binary)(INFORMATION) : FILE: ./amsua_n15.SpcCoeff.bin; ^M
 ACCoeff RELEASE.VERSION:  1.04  N_FOVS=30  N_CHANNELS=15
 Read_ODPS_Binary(INFORMATION) : FILE: ./amsua_n15.TauCoeff.bin; ^M
   ODPS RELEASE.VERSION:  2.01  N_LAYERS=100  N_COMPONENTS=2  N_ABSORBERS=1  N_CHANNELS=15
N_COEFFS=21600
 IRwaterCoeff_ReadFile(INFORMATION) : FILE: ./Nalli.IRwater.EmisCoeff.bin; ^M
 IRwaterCoeff RELEASE.VERSION:  3.02  N_ANGLES= 76  N_FREQUENCIES= 2223  N_WIND_SPEEDS= 11
 SETUPRAD:  write header record for amsua_n15                    7          30
          8          0          0          17          0        30303
  to file pe0000.amsua_n15_01   2011032212
```

Following output will repeat multiple times in *stdout*

radiance obs innovation computation followed by CRTM coefficients info (only for 1st outer loop)

stdout: Check outer loop and inner iteration

Print Jo components at beginning of the inner loop:

```
 Begin J table inner/outer loop          0          1
     J term                                    J
surface pressure              3.1647295169634604E+03
temperature                   4.1891962408742038E+04
wind                          4.5359281326225479E+04
moisture                      2.7516195492749889E+03
gps                           5.3375455660772332E+02
radiance                      6.7641749172470163E+03
 -----------------------------------------------------
 J Global                     1.0046552227506071E+05
 End Jo table inner/outer loop          0          1
```

Print cost function values for each inner iteration:

```
cost,grad,step,b,step? =   1   0  1.0046552227506069 18E+05  6.3599452640463880 47E+01  1.2772752414237220 63E+00  0.0000000000000000 00E+00  good
pcgsoi: gnorm(1:2),b=  2.0135898256137741 10E+03  2.0135898256137752 47E+03  4.9781072868582382 23E-01
cost,grad,step,b,step? =   1   1  9.5299083943310019 09E+04  4.4873041189714051 31E+01  3.4185183097185753 99E+00  4.9781072868582382 23E-01  good
pcgsoi: gnorm(1:2),b=  1.2369600976390861 43E+03  1.2369600976390845 52E+03  6.1430589383418221 51E-01
cost,grad,step,b,step? =   1   2  8.8415590256186318 58E+04  3.5170443523491229 60E+01  1.8387080463289509 07E+00  6.1430589383418221 51E-01  good
pcgsoi: gnorm(1:2),b=  8.2903931142214082 68E+02  8.2903931142214082 68E+02  6.7022316484135580 07E-01
cost,grad,step,b,step? =   1   3  8.6141181771669449 52E+04  2.8793042760745880 32E+01  2.2245171129958918 32E+00  6.7022316484135580 07E-01  good
pcgsoi: gnorm(1:2),b=  7.4909421327669804 21E+02  7.4909421327669792 84E+02  9.0356898998154333 57E-01
cost,grad,step,b,step? =   1   4  8.4296969636064575 75E+04  2.7369585551789015 20E+01  3.2562999482423129 47E+00  9.0356898998154333 57E-01  good
pcgsoi: gnorm(1:2),b=  4.3338304813946268 72E+02  4.3338304813946285 78E+02  5.7854277934380626 82E-01
cost,grad,step,b,step? =   1   5  8.1857694188143039 36E+04  2.0817854071432595 48E+01  3.3902275264773895 64E+00  5.7854277934380626 82E-01  good
pcgsoi: gnorm(1:2),b=  4.9652102614888946 16E+02  4.9652102614888985 95E+02  1.1456863120984588 37E+00
cost,grad,step,b,step? =   1   6  8.0388427048831959 84E+04  2.2282751763390656 66E+01  2.8505075164434994 23E+00  1.1456863120984588 37E+00  good
```

• • • • •

stdout: Check outer loop and inner iteration

Print diagnostics about guess field after adding analysis increment:

```
================================================================================
Status    Var          Mean                  Min                   Max
analysis  U      1.000121864994E+01  -2.419309362670E+01   5.851826277025E+01
analysis  V     -6.135617740694E-02  -5.248606119488E+01   5.502160512845E+01
analysis  TV     2.437949825619E+02   1.932559352375E+02   3.050689176747E+02
analysis  Q      1.886748199518E-03   1.000000000000E-07   2.064960777388E-02
analysis  TSEN   2.434680623906E+02   1.932555080178E+02   3.015036100285E+02
analysis  OZ     1.000000000000E-15   1.000000000000E-15   1.000000000000E-15
analysis  CW     0.000000000000E+00   0.000000000000E+00   0.000000000000E+00
analysis  DIV    0.000000000000E+00   0.000000000000E+00   0.000000000000E+00
analysis  VOR    0.000000000000E+00   0.000000000000E+00   0.000000000000E+00
analysis  PRSL   4.233881117321E+01   5.306372580295E+00   1.026146077849E+02
analysis  PS     9.667713698921E+01   6.811830115039E+01   1.029057548376E+02
analysis  SST    2.876061426024E+02   2.519014434814E+02   3.026017761230E+02
analysis  radb   2.312546270209E-02  -1.001980000000E+02   1.463245040000E+02
analysis  pcpb   0.000000000000E+00   0.000000000000E+00   0.000000000000E+00
analysis  aftb   0.000000000000E+00   0.000000000000E+00   0.000000000000E+00
```

Print diagnostics about analysis increment:

```
================================================================================
increment u                          4.245564934418E-02  -5.704651324022E+00   8.360870264603E+00
increment v                          2.194016252748E-02  -5.903095661442E+00   5.363745636870E+00
increment tv                         1.514239663130E-01  -3.243805033460E+00   5.308469697861E+00
increment tsen                       1.502589299933E-01  -3.241358667819E+00   5.307975573878E+00
increment q                          5.596441882602E-06  -3.163107489857E-03   5.929173274032E-03
increment oz                         0.000000000000E+00   0.000000000000E+00   0.000000000000E+00
increment cw                         0.000000000000E+00   0.000000000000E+00   0.000000000000E+00
increment prse                      -2.401097530955E-03  -1.294451891771E-01   6.549264125341E-02
increment ps                        -5.844547637771E-03  -1.294451891771E-01   6.549264125341E-02
increment sst                       -6.303690981444E-03  -4.179376399069E-01   4.961453154801E-01
```

stdout: start of 2nd outer loop

```
GLBSOI: jiter,jiterstart,jiterlast,jiterend=          2          1
          2           1
```

stdout: check Jo components

Before 1st outer loop

```
 Begin J table inner/outer loop          0          1
     J term                                        J
surface pressure              3.1647295169634604E+03
temperature                   4.1891962408742038E+04
wind                          4.5359281326225479E+04
moisture                      2.7516195492749889E+03
gps                           5.3375455660772332E+02
radiance                      6.7641749172470163E+03
 -----------------------------------------------------
 J Global                     1.0046552227506071E+05
 End Jo table inner/outer loop          0          1
```

Jo decreased after 1st outloop

Before 2nd outer loop

```
 Begin J table inner/outer loop          0          2
     J term                                        J
background                    5.0586314376624878E+03
surface pressure              2.1287629750238493E+03
temperature                   2.4163225495490991E+04
wind                          3.4131578938344865E+04
moisture                      1.2014140664772476E+03
gps                           2.6824775912451992E+02
radiance                      4.0097736541798968E+03
 -----------------------------------------------------
 J Global                     7.0961634326303858E+04
 End Jo table inner/outer loop          0          2
```

stdout: Check Analysis Result Output

Save analysis result

```
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    321.6622    279.6388    309.0912
k,max,min,mid T=    2    321.6799    280.6721    309.1487
. . .
. . .
k,max,min,mid T=   49    653.1034    614.2672    634.6012
k,max,min,mid T=   50    684.9056    655.0489    672.6349
rmse_var=T
```

K	Maximum	Minimum	Domain Center

QVAPOR, U, V, SEAICE, SST, TSK …

Variable name in netcdf file

Stdout: Final normal exit information

Additional resource stats

```
 glbsoi: complete
[000]gsisub(): : complete.


     ENDING DATE-TIME    JUL 06,2017  14:02:24.923  187  THU   2457941
     PROGRAM GSI_ANL HAS ENDED.
* . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . *
****************RESOURCE STATISTICS*******************************
The total amount of wall time                        = 190.142072
The total amount of time in user mode                = 186.204692
The total amount of time in sys mode                 = 2.268655
The maximum resident set size (KB)                   = 384912
Number of page faults without I/O activity           = 61615
Number of page faults with I/O activity              = 0
Number of times filesystem performed INPUT           = 0
Number of times filesystem performed OUTPUT          = 0
Number of Voluntary Context Switches                 = 8090
Number of InVoluntary Context Switches               = 637
****************END OF RESOURCE STATISTICS*************************
```

Observation Fitting Statistics and Diagnostic Files

Details in User's Guide Section 4.5 and A.2

2 ways to check fitting statistics

- RULE: The analysis should fit better to the observation than the background

- **text files** (fort.201, … , fort.232):
  statistic information at the beginning of each outer loop

- **binary files** (diag*ges.*, diag*anl.*) :
  OMB and OMA information for each observation

Observation Innovation Statistics

GSI User's Guide Table 4.4

File name	Variables in file	Ranges/units
fort.201 or fit_p1.analysis_time	fit of surface pressure data	mb
fort.202 or fit_w1.analysis_time	fit of u, v wind data	m/s
fort.203 or fit_t1.analysis_time	fit of temperature data	K
fort.204 or fit_q1.analysis_time	fit of moisture data	percent of guess $q_{saturation}$
fort.205	fit of precipitation water data	mm
fort.206	fit of ozone observations from sbuv6_n14 (, _n16, _n17, _n18), sbuv8_n16 (, _n17, _n18, _n19), omi_aura, gome_metop-a/b, mls_aura	

Conventional observations

Each file is for one observation variable

fit* files are generated in run_gsi_regional.ksh

Observation Innovation Statistics (Cont')

fort.207 or fit_rad1.analysis_time	fit of satellite radiance data, such as: amsua_n15(, n16, n17, n18, metop-a, aqua, n19), amsub_n17, hirs3_n17, hirs4_n19 (, metop-a), etc	Satellite radiance
fort.208	fit of prepcipitation rate (pcp_ssmi, pcp_tmi)	
fort.209	fit of radar radial wind (rw)	
fort.210	fit of lidar wind (dw)	
fort.211	fit of radar superob wind data (srw)	Radar radial wind
fort.212	fit of GPS data (refractivity or bending angle)	fractional difference
fort.213	fit of conventional sst data	GPS C
fort.214	Tropical cyclone central pressure	
fort.215	Lagrangian tracer data	
Fort.217	Fit of aerosol product (aod)	
Fort.218	Fit of wind gust	
Fort.219	Fit of visibility	

Now up to fort.232

Example: fit_p1 (fort.201)

```
current fit of surface pressure data, ranges in mb
--------------------------------------------------
 pressure levels (hPa)=   0.0 2000.0
     it    obs use type stype    count      bias       rms      cpen     qcpen
 o-g 01     ps asm  120  0000      141    0.2326    1.0637    1.6900    1.6613
 o-g 01     ps asm  180  0000     2210    0.2623    1.1016    1.4846    1.2854
 o-g 01     ps asm  181  0000      725    0.1493    1.2380    2.0757    1.8844
 o-g 01     ps asm  187  0000    12524    0.5000    1.0227    0.5437    0.5061
 o-g 01         asm  all         15600    0.4476    1.0455    0.7585    0.6910
```

O-B

```
current fit of surface pressure data, ranges in mb
--------------------------------------------------
 pressure levels (hPa)=   0.0 2000.0
     it    obs asm type stype    count      bias       rms      cpen     qcpen
 o-g 03     ps asm  120 0000      141   -0.0965    0.8260    0.7879    0.7875
 o-g 03     ps asm  180 0000     2212    0.0210    0.8709    0.9194    0.8221
 o-g 03     ps asm  181 0000      732   -0.0356    1.1377    0.7439    0.7396
 o-g 03     ps asm  187 0000    12533    0.0390    0.7038    0.2583    0.2504
 o-g 03         asm  all         15618    0.0318    0.7561    0.3794    0.3592
```

O-A

Results from test case using 2 outer loops with 10 inner iterations in each outer loop

Example: fit_w1 (fort.202)

Vertical levels

					ptop	1000.0	900.0	800.0	600.0	100.0	50.0	0.0
	it	obs	use	type	styp	pbot	1200.0	1000.0	900.0	800.0	150.0	100.0	2000.0
Data Used in Analysis													
o-g	01	uv	asm	220	0000	count	135	428	427	838	688	978	8061
o-g	01	uv	asm	220	0000	bias	-0.43	0.93	0.85	0.61	0.31	0.23	0.64
o-g	01	uv	asm	220	0000	rms	3.62	3.68	4.21	4.13	5.61	5.33	4.84
o-g	01	uv	asm	220	0000	cpen	0.70	0.95	1.26	1.33	1.39	1.30	1.25
o-g	01	uv	asm	220	0000	qcpen	0.70	0.95	1.26	1.33	1.78	1.29	1.25
o-g	01		asm	all		count	1962	1398	1440	2623	1014	1071	18248
o-g	01		asm	all		bias	-0.31	0.93	0.59	0.05	0.21	0.29	0.28
o-g	01		asm	all		rms	3.02	3.87	4.34	4.35	5.59	5.36	4.67
o-g	01		asm	all		cpen	0.50	0.68	1.07	1.12	1.35	1.28	1.05
Rejected													
o-g	01	uv	rej	220	0000	count	0	0	0	0	0	0	292
o-g	01	uv	rej	220	0000	bias	0.00	0.00	0.00	0.00	0.00	0.00	2.81
o-g	01	uv	rej	220	0000	rms	0.00	0.00	0.00	0.00	0.00	0.00	8.69
o-g	01	uv	rej	220	0000	cpen	0.00	0.00	0.00	0.00	0.00	0.00	0.00
o-g	01	uv	rej	220	0000	qcpen	0.00	0.00	0.00	0.00	0.00	0.00	0.00
o-g	01		rej	all		count	77	22	139	366	2	1	1089
o-g	01		rej	all		bias	5.58	2.18	-3.75	-10.98	24.83	28.94	-5.73
o-g	01		rej	all		rms	12.51	10.85	11.12	18.02	61.69	59.27	18.49
o-g	01		rej	all		cpen	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Monitoring													
o-g	01	uv	mon	220	0000	count	3	2	5	3	11	40	145
o-g	01	uv	mon	220	0000	bias	-1.55	-2.31	7.31	0.04	-3.49	2.02	1.74
o-g	01	uv	mon	220	0000	rms	3.92	3.68	15.18	2.94	10.65	8.67	8.93
o-g	01	uv	mon	220	0000	cpen	0.00	0.00	0.00	0.00	0.00	0.00	0.00
o-g	01	uv	mon	220	0000	qcpen	0.00	0.00	0.00	0.00	0.00	0.00	0.00
o-g	01		mon	all		count	4651	9610	1570	676	51	63	16859
o-g	01		mon	all		bias	-0.78	-0.25	0.53	-0.79	-0.64	0.15	-0.38
o-g	01		mon	all		rms	2.88	3.22	5.71	8.18	19.81	12.48	4.24
o-g	01		mon	all		cpen	0.88	0.93	0.65	0.64	0.00	0.00	0.86

Example: fit_w1 (fort.202)

O-B

it	obs	type	styp	ptop / pbot	1000.0 / 1200.0	900.0 / 1000.0	800.0 / 900.0	600.0 / 800.0	100.0 / 150.0	50.0 / 100.0	0.0 / 2000.0
o-g 01	uv	220	0000	count	135	428	427	838	688	978	8061
o-g 01	uv	220	0000	bias	-0.43	0.93	0.85	0.61	0.31	0.23	0.64
o-g 01	uv	220	0000	**rms**	3.62	3.68	4.21	4.13	5.61	5.33	**4.84**
o-g 01	uv	220	0000	cpen	0.70	0.95	1.26	1.33	1.39	1.30	1.25

O-A After 1st outer loop

it	obs	type	styp	ptop / pbot	1000.0 / 1200.0	900.0 / 1000.0	800.0 / 900.0	600.0 / 800.0	100.0 / 150.0	50.0 / 100.0	0.0 / 2000.0
o-g 02	uv	220	0000	count	135	428	427	838	688	978	8061
o-g 02	uv	220	0000	bias	-0.27	0.87	0.64	0.66	0.25	0.56	0.69
o-g 02	uv	220	0000	**rms**	3.29	3.26	3.43	3.28	5.15	5.08	**4.31**
o-g 02	uv	220	0000	cpen	0.56	0.74	0.82	0.84	1.18	1.17	0.96

O-A After 2st outer loop

it	obs	type	styp	ptop / pbot	1000.0 / 1200.0	900.0 / 1000.0	800.0 / 900.0	600.0 / 800.0	100.0 / 150.0	50.0 / 100.0	0.0 / 2000.0
o-g 03	uv	220	0000	count	135	428	427	838	688	978	8061
o-g 03	uv	220	0000	bias	-0.27	0.84	0.60	0.63	0.20	0.57	0.67
o-g 03	uv	220	0000	**rms**	3.24	3.11	3.24	3.07	5.01	5.00	**4.17**
o-g 03	uv	220	0000	cpen	0.54	0.67	0.73	0.74	1.12	1.14	0.89

Example: fit_rad1 (fort.207)

Print out **satinfo** firstly, then the bc **coefficients** for each channel

```
RADINFO_READ:  jpch_rad=   4934
   1 amsua_n15            chan=    1 var=   3.000 varch_cld= 20.000 use=  1 ermax=   4.500 b_rad=   10.00 pg_rad=   0.00 icld_det=-2
   2 amsua_n15            chan=    2 var=   2.200 varch_cld= 18.000 use=  1 ermax=   4.500 b_rad=   10.00 pg_rad=   0.00 icld_det=-2
   3 amsua_n15            chan=    3 var=   2.000 varch_cld= 12.000 use=  1 ermax=   4.500 b_rad=   10.00 pg_rad=   0.00 icld_det=-2
   4 amsua_n15            chan=    4 var=   0.600 varch_cld=  3.000 use=  1 ermax=   2.500 b_rad=   10.00 pg_rad=   0.00 icld_det=-2
+--4929 lines: 5 amsua_n15            chan=    5 var=   0.300 varch_cld=  0.500 use=  1 ermax=   2.000 b_rad=   10.00 pg_rad=   0.00 icld_det=-2----------------------------
4934 ahi_himawari8        chan=   16 var=   2.200 varch_cld=  0.000 use= -1 ermax=   2.000 b_rad=   10.00 pg_rad=   0.00 icld_det=-2
 RADINFO_READ:  satbias_pc file doesnot exist - set to zero
 RADINFO_READ:  guess air mass bias correction coefficients below
   1            amsua_n15    2.680929    0.000000    0.000000    0.281644    0.735701    0.000000    0.000000   -0.016301    4.269251    3.092839   -4.917681   -2.043537
   2            amsua_n15    1.591210    0.000000    0.000000   15.459969   -3.826823    0.000000    0.000000   -0.015617    3.943127    9.279759   -4.624200   -1.794099
   3            amsua_n15    1.626770    0.000000    0.000000    1.296444   -0.634276    0.000000    0.000000   -0.005104   -2.269633   -0.670826   -1.166500    0.302704
   4            amsua_n15   -0.181336    0.000000    0.000000    0.166206   -0.024951    0.000000    0.000000    0.022909   -0.619062   -0.508514   -0.758746    0.006709
+--7165 lines: 5            amsua_n15    0.302768    0.000000    0.000000    0.028569   -0.060893    0.000000    0.000000    0.011679    0.610039   -0.936094   -1.323020
```

Statistics output as a function of observation type

Tip: search “penalty” to quickly get here

sat	type	penalty	nobs	iland	isnoice	icoast	ireduce	ivarl	nlgross
n15	amsua	2963.45197415	1341	630	89	205	216	9132	0
		qcpenalty	qc1	qc2	qc3	qc4	qc5	qc6	qc7
		2963.45197415	103	9	1029	0	0	14	14

1 radiance ob may include mult. channels, not all channels are used in analysis

Example: fit_rad1.2011032212 (fort.207)

Statistics as a function of channel:

channel			nobs	nobs_tossed	variance	Bias (before BC)	Bias (after BC)	penalty	O-B (w/BC)	Standard dev
6	6	amsua_n15	53	60	-0.230	-1.9390127	0.4251205	3.0242853	0.4520939	0.1538230
7	7	amsua_n15	1071	270	0.250	0.1640600	0.3859903	1.9443378	0.4217391	0.1699275
8	8	amsua_n15	558	783	0.275	0.3348176	0.4027184	1.5233287	0.4370213	0.1697219
9	9	amsua_n15	27	1314	0.340	0.3017604	0.5059296	1.0323963	0.5117796	0.0771593

Statistics for each observation type:

O-B

	it	satellite	instrum	# read	# keep	# assim	penalty	qcpnlty	cpen	qccpen
o-g	01 rad	n16	hirs3	0	0	0	0.0000	0.0000	0.0000	0.0000
+--	7 lines: o-g 01 rad	n17	hirs3	0	0	0	0.0000	0.0000	0.0000	0.0000
o-g	01 rad	aqua	airs	0	0	0	0.0000	0.0000	0.0000	0.0000
o-g	01 rad	n15	amsua	24855	17417	1662	2963.5	2963.5	1.7831	1.7831
o-g	01 rad	n18	amsua	46845	19260	2682	2049.0	2049.0	0.76398	0.76398
o-g	01 rad	n19	amsua	71655	24042	2619	1383.0	1383.0	0.52808	0.52808

O-A

	it	satellite	instrum	# read	# keep	# assim	penalty	qcpnlty	cpen	qccpen
o-g	03 rad	n16	hirs3	0	0	0	0.0000	0.0000	0.0000	0.0000
+--	7 lines: o-g 03 rad	n17	hirs3	0	0	0	0.0000	0.0000	0.0000	0.0000
o-g	03 rad	aqua	airs	0	0	0	0.0000	0.0000	0.0000	0.0000
o-g	03 rad	n15	amsua	24855	17417	2977	790.23	790.23	0.26545	0.26545
o-g	03 rad	n18	amsua	46845	19260	3120	834.44	834.44	0.26745	0.26745
o-g	03 rad	n19	amsua	71655	24042	3163	780.42	780.42	0.24673	0.24673

Diagnostic files (User's Guide A.2)

- Files include observation departure for each obs:

```
diag_amsua_metop-a_anl.2011032212   diag_amsua_n15_anl.2011032212
diag_amsua_metop-a_ges.2011032212   diag_amsua_n15_ges.2011032212
diag_amsub_n17_anl.2011032212       diag_conv_anl.2011032212
diag_amsub_n17_ges.2011032212       diag_conv_ges.2011032212
        . . .    . . .
```

- To get these files, turn write_diag on:

```
write_diag(1)=.true.,write_diag(2)=.false.,write_diag(3)=.true.,
```

- To read this binary information:
  - Code to read these files (util/Analysis_Utilities/read_diag)
    - read_diag_conv.f90 (`diag_conv*`)
    - read_diag_rad.f90 (`diag_amsub_n17*` …)
  - Compile: `./make`

    two executables: *read_diag_conv.exe* *read_diag_rad.exe*

Observation departure for each ob

- Run *read_diag_conv.exe*: (*namelist.conv* needed)

```
&iosetup
  infilename='./diag_conv_anl.2011032212',
  outfilename='./results_conv_anl',
/
```

GSI diagnostic file

Text file to save the content of diagnostic file

Example content of *results_conv_anl*

station ID	obs type	obs time	obs latitude	obs longitude	obs pressure	usage	obs value	O-B
ps @ 21997 :	180	0.00	42.65	190.98	1013.20	1	1013.2	-0.35
t @ PADK :	187	-0.07	51.88	183.35	993.34	1	276.0	0.33

- Run *read_diag_rad.exe*: (*namelist.rad* needed)

```
&iosetup
  infilename='./diag_amsua_n18_ges.2011032212',
  outfilename='./results_amsua_n15_anl',
/
```

Same as conventional

Convergence information

Details in User's Guide Section 4.6 and A.3

Find convergence information

- From stdout (see example in stdout of this talk)
- From *fort.220*
  - Includes many details
    - Cost function and gradient
    - Contribution from background and each data type
    - …
  - DTC provides a ksh script to filter this file
    - util/Analysis_Utilities/plot_cost_grad/filter_fort220.ksh

```
grep 'cost,grad,step,b' fort.220 | sed -e 's/cost,grad,step,b,step? =  //g'
    | sed -e 's/good//g' > cost_gradient.txt
```

outer loop

```
1  0  1.88283932183955243 1E+05  5.28462395890434982 0E+03  1.69798840941541790 2E-03  0.00000000000000E+00
```

Inner iteration

Convergence information

- *cost_gradient.txt*

outer loop | Inner iteration

1	0	0.14302301855058484 9E+06	0.11564126965092774 5E+08	0.24398954427077819 8E-02	0.00000000000000000 0E+00
1	1	0.11480775786956082 4E+06	0.18709884344764431 0E+07	0.76301193791083118 9E-02	0.16179245005906334 2E+00
1	2	0.10053189275757456 2E+06	0.17016520499209254 0E+07	0.55461495774775434 3E-02	0.90949362303091829 7E+00
1	3	0.91094275959891834 6E+05	0.75374450471307628 3E+06	0.11197659891791625 4E-01	0.44294866553247619 3E+00
		. . .	. . .		
2	8	0.72027652192720430 3E+05	0.10946803552576767 1E+06	0.57441328955803918 5E-02	0.86046650706111660 3E+00
2	9	0.71398853248842307 9E+05	0.96761808175083875 5E+05	0.68034275291122834 1E-02	0.88392751098842026 2E+00
2	10	0.70740541299337259 3E+05	0.10731497992413668 5E+06	0.43417883383592304 6E-02	0.11090633995796894 7E+01

Penalty (cost function) | Norm of gradient

Plot convergence information

- DTC provides a NCL script to make plot
  - util/Analysis_Utilities/plot_cost_grad/GSI_cost_gradient.ncl

```
load "$NCARG_ROOT/lib/ncarg/nclex/gsun/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"

begin

  nloop1=50
  nloop2=50
  step = stringtofloat(systemfunc("cut -c7-9 ./cost_gradient.txt"))
  cost = stringtofloat(systemfunc("cut -c10-35 ./cost_gradient.txt"))
  gradient = stringtofloat(systemfunc("cut -c36-61 ./cost_gradient.txt"))

  titles = new(4,string)
  titles(0)="Cost outer 1"
  titles(1)="Gradient outer 1"
  titles(2)="Cost outer 2"
  titles(3)="Gradient outer 2"

  plot = new(4,graphic)

  xwks = gsn_open_wks("pdf","GSI_cost_gradient")
```

# of iterations in 1st outer loop

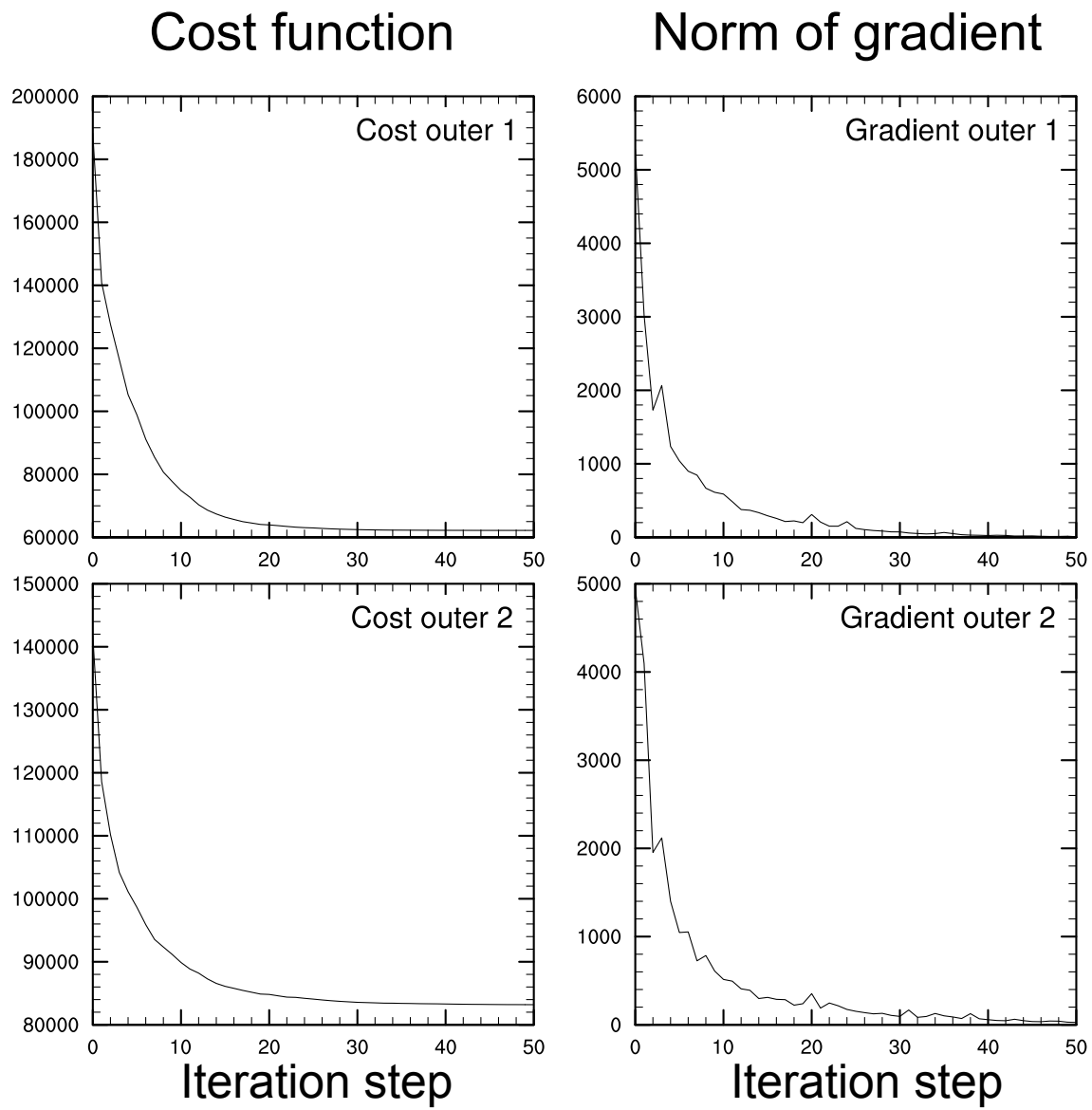
# of iterations in 2nd outer loop

figure file name and format

Check convergence: example figure

1st outer loop with 50 inter iteration

2nd outer loop with 50 inter iteration



Analysis Increment

Details in User's Guide Section 4.9 and A.4

Analysis increment: plot for ARW case

util/Analysis_Utilities/plots_ncl/Analysis_increment.ncl

util/Analysis_Utilities/plots_ncl/GSI_singleobs_arw.ncl

```
load "$NCARG_ROOT/lib/ncarg/nclex/gsun/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"

begin

  cdf_analysis = addfile("./gsiprd_2014061700_prepbufr/wrf_inout.cdf","r")

  cdf_bk = addfile("./GSI-DTC/bk/wrfinput_d01_2014-06-17_00:00:00.cdf","r")

  Ta = cdf_analysis->T(0,:,:,:)
  Tb = cdf_bk->T(0,:,:,:)

  DT = Ta - Tb
```

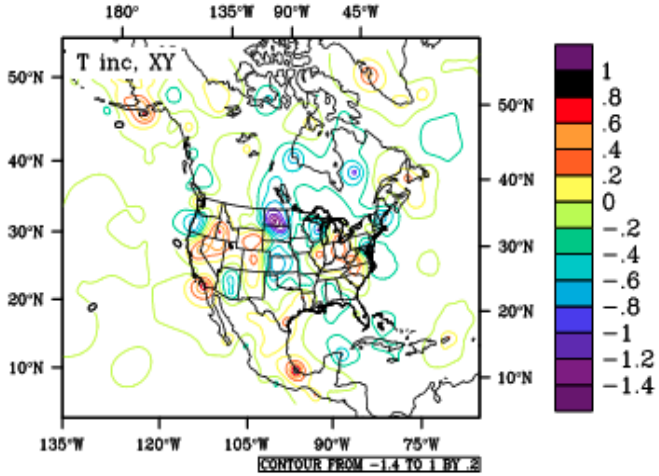
GSI analysis

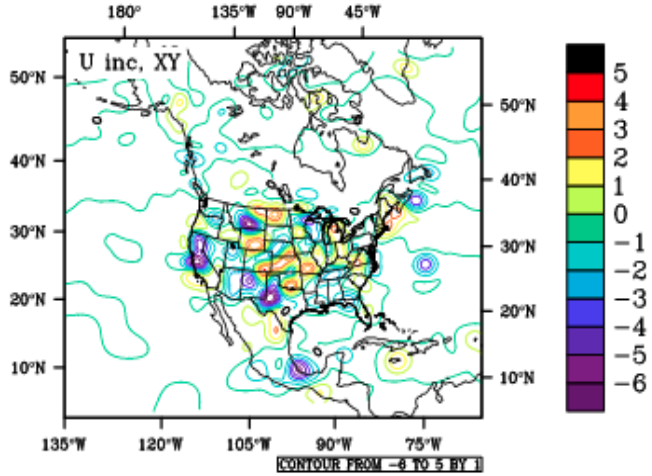
First guess

Analysis increment
= GSI analysis – First guess

Analysis increment: example

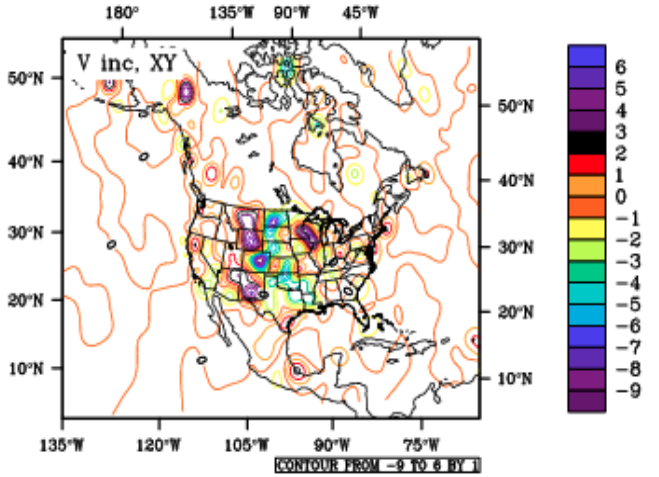
ΔT

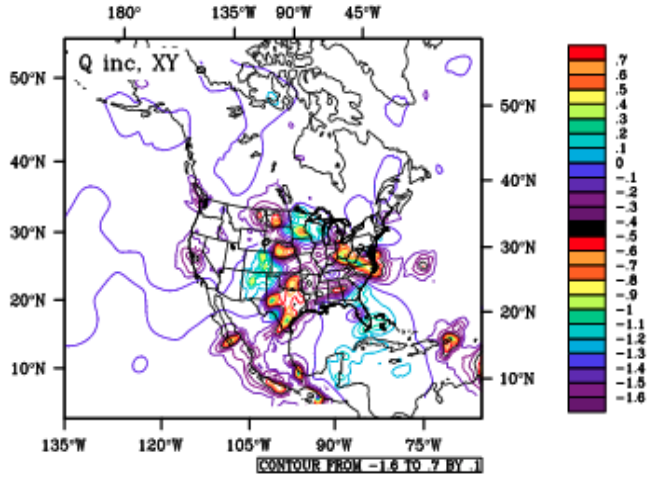




ΔU

ΔV





Δq

GSI analysis increment at the 15th level

Summary

- We can diagnose GSI analysis by checking:
  - ✓ Standard output
  - ✓ Observation fitting statistics and binary diagnostic files
  - ✓ Convergence information
  - ✓ Analysis increment
- There are more:
  - Single observation test
  - Overlay observation innovation on analysis increment
  - Forecast
  - …

Questions?

gsi-help@ucar.edu