2013 GSI Community Tutorial
August 05-07, 2013, NCWCP, MD

GSI Fundamentals (3): Diagnostics

Ming Hu and Chunhua Zhou

Developmental Testbed Center

Outlines

- GSI fundamentals (1): Setup and Compilation
- GSI fundamentals (2): Run and Namelist
- GSI fundamentals (3): Diagnostics
  - Standard output
  - Observation fitting statistics and diagnosis files
  - Convergence information
  - Analysis increment
- GSI fundamentals (4): Applications

This talk is tailored based on Chapter 4 of the GSI User's Guide for Community Release V3.2

Standard Output (stdout)

Details in User's Guide Section 4.1

Highlight several important points

stdout: Check Background Input

P_TOP, ZNU, ZNW, RDX, RDY, MAPFAC_M, XLAT, XLONG, MUB, MU, PHB

```
0:  rmse_var=T
0:  ordering=XYZ
0:  WrfType,WRF_REAL= 104 104
0:  ndim1= 3
0:  staggering= N/A
0:  start_index= 1 1 1 0
0:  end_index= 348 247 50 0
0:  k,max,min,mid T=  1   310.8738   234.0430   280.8286
0:  k,max,min,mid T=  2   311.2928   235.4056   280.9952
0:  k,max,min,mid T=  3   311.4995   237.6656   281.3005

0:  k,max,min,mid T= 48   702.1189   616.5657   651.2628
0:  k,max,min,mid T= 49   770.3488   684.2148   718.2207
0:  k,max,min,mid T= 50   837.6133   760.6921   792.9601
                      K   Maximum    Minimum    Central grid
```

Variable name in netcdf file

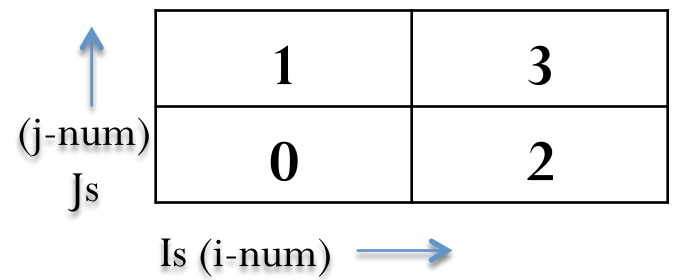
QVAPOR, U, V, LANDMASK, SEAICE, SST,IVGTYP, ISLTYP, VEGFRA, SNOW, U10, V10, SMOIS, TSLB, TSK

Developmental Testbed Center

stdout: Check domain and convinfo

Domain partition information:

	Task ID	Js	Is	j-num	i-num
general_DETER_SUBDOMAIN: task,istart,jstart,ilat1,jlon1=	0	1	1	124	174
general_DETER_SUBDOMAIN: task,istart,jstart,ilat1,jlon1=	1	125	1	123	174
general_DETER_SUBDOMAIN: task,istart,jstart,ilat1,jlon1=	2	1	175	124	174
general_DETER_SUBDOMAIN: task,istart,jstart,ilat1,jlon1=	3	125	175	123	174



convinfo

Obs type | usage

```
READ_CONVINFO: tcp 112   0  1   3.00000   0   0   0   75.0000   5.00000   1.00000   75.0000
0.00000   0   0.00000   0.00000 0
READ_CONVINFO: ps  120   0  1   3.00000   0   0   0   4.00000   3.00000   1.00000   4.00000
0.30E-03   0   0.00000   0.00000  0
READ_CONVINFO: ps  180   0  1   3.00000   0   0   0   4.00000   3.00000   1.00000   4.00000
0.30E-03   0   0.00000   0.00000  0
READ_CONVINFO: t   120   0  1   3.00000   0   0   0   8.00000   5.60000   1.30000   8.00000
0.10E-05   0   0.00000   0.00000  0
READ_CONVINFO: gps 440   0 -1   3.00000   0   0   0  10.0000   10.0000   1.00000   10.0000
0.00000    0   0.00000   0.0000   0
```

stdout: Check Observations Input

```
read_obs_check: bufr file date is   2011032212 prepbufr ps
read_obs_check: bufr file uv                   not available satwnd
read_obs_check: bufr file rw                   not available radarbufr
read_obs_check: bufr file pcp_tmi   trmm       not available tmirrbufr
read_obs_check: bufr file date is   2011032212 prepbufr t
read_obs_check: bufr file date is   2011032212 prepbufr q
    ••••••
data type hirs2_n14          not used in info file -- do not read file
hirs2bufr
data type hirs3_n16          not used in info file -- do not read file
hirs3bufr
```

```
 READ_BUFRTOVS: file=hirs4bufr  type=hirs4      sis=hirs4_metop-a        nread=
189563 ithin= 1 rmesh=120.000000 isfcalc= 1 ndata=     21736 ntask=  2
 READ_BUFRTOVS: file=hirs3bufr  type=hirs3      sis=hirs3_n17            nread=
202426 ithin= 1 rmesh=120.000000 isfcalc= 1 ndata=     23636 ntask=  2
 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=
63825 ithin= 2 rmesh= 60.000000 isfcalc= 1 ndata=     52309 ntask=  1
```

Developmental Testbed Center

stdout: Check Observations Input

Observation distribution in an analysis using 4 processors

			Task 0	Task 1	Task 2	Task 3
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		0	0	6	3
OBS_PARA:	gps_ref		3538	5580	2277	6768
OBS_PARA:	hirs3	n17	0	0	478	773
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

1	3
0	2

Developmental Testbed Center

stdout: observation innovation

```
 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
 INIT_CRTM: crtm_init() on path "./"
 Read_SpcCoeff_Binary(INFORMATION) : FILE: ./amsua_n15.SpcCoeff.bin; ^M
 SpcCoeff RELEASE.VERSION:  7.03  N_CHANNELS=15^M
 amsua_n15 AntCorr 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
 Read_EmisCoeff_Binary(INFORMATION) : FILE: ./EmisCoeff.bin; ^M
 EmisCoeff RELEASE.VERSION:  2.02  N_ANGLES= 16  N_FREQUENCIES= 2223
N_WIND_SPEEDS= 11
 SETUPRAD:  write header record for amsua_n15                      5
30
          8           0           0          15           0       13784
  to file pe0000.amsua_n15_01   2011032212
```

CRTM: binary order and versions

stdout: Check outer loop and inner iteration

1st outer loop

Inner iteration

```
GLBSOI:  START pcgsoi jiter=         1
Initial cost function =  1.946084290880794579E+05
Initial gradient norm =  5.576089529423489694E+03
cost,grad,step,b,step? =   1   0  1.946084290880794579E+05  5.576089529423489694E+03  9.446046100069309653E-04
0.000000000000000000E+00  good
cost,grad,step,b,step? =   1   1  1.652380510140116094E+05  5.918533543369932886E+03  5.770901706622101049E-04
1.126597414824659582E+00  good
cost,grad,step,b,step? =   1   2  1.450231367439358146E+05  2.250104752168165305E+03  1.248362877688804704E-03
1.445364045468512382E-01  good
cost,grad,step,b,step? =   1   3  1.387027112027065014E+05  3.053100435499959531E+03  1.047494021606896770E-03
1.841097162253764630E+00  good
```

2nd outer loop

Inner iteration

```
GLBSOI:  START pcgsoi jiter=         2
Initial cost function =  1.415039414838706725E+05
Initial gradient norm =  3.380943386335680316E+03
cost,grad,step,b,step? =   2   0  1.415039414838706725E+05  3.380943386335680316E+03  1.489035006995881186E-03
0.000000000000000000E+00  good
cost,grad,step,b,step? =   2   1  1.244831126142531430E+05  2.748895456590210870E+03  4.496514460165658249E-04
6.610596506387631521E-01  good
cost,grad,step,b,step? =   2   2  1.210853546326485375E+05  1.624514629971304885E+03  1.669025207983985048E-03
3.492454901594346239E-01  good
cost,grad,step,b,step? =   2   3  1.166807173577626236E+05  1.826733003801378345E+03  6.055466310511921964E-04
1.264453599015726848E+00  good
```

stdout: check Jo components

Before 1st outer loop

Observation Type	Nobs	Jo	Jo/n
surface pressure	15600	1.1832900994269770E+04	0.759
temperature	11913	1.9028504337386752E+04	1.597
wind	36726	4.0133809857345797E+04	1.093
moisture	4416	4.6817824733959324E+03	1.060
gps	8709	1.8551360623628108E+04	2.130
radiance	114977	1.0038007080205309E+05	0.873
	Nobs	Jo	Jo/n
Jo Global	192341	1.9460842908807946E+05	1.012

After 1st outer loop

J_o

surface pressure	15617	6.9931034829496348E+03	0.448
temperature	11913	1.2304487849735804E+04	1.033
wind	36800	2.7098051897512487E+04	0.736
moisture	4416	2.9381913471101529E+03	0.665
gps	8811	1.0898349569787561E+04	1.237
radiance	157545	7.8484444589549588E+04	0.498
	Nobs	Jo	Jo/n
Jo Global	235102	1.3871662873664522E+05	0.590

After 2nd outer loop

surface pressure	15619	6.3786866162558572E+03	0.408
temperature	11913	1.1099886560557266E+04	0.932
wind	36798	2.4257649951877418E+04	0.659
moisture	4416	2.5071927158240760E+03	0.568
gps	8860	9.6425658325086042E+03	1.088
radiance	172626	6.0208464023162502E+04	0.349
	Nobs	Jo	Jo/n
Jo Global	250232	1.1409444570018572E+05	0.456

stdout: Check Analysis Result Output

```
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    311.0948    234.0167    280.5982
k,max,min,mid T=   2    311.5334    235.6663    280.7569
   . . .    . . .

k,max,min,mid T=  49    762.8171    679.6612    706.3553
k,max,min,mid T=  50    828.8514    754.9369    777.9498
rmse_var=T
```

K	Maximum	Minimum	Central grid

QVAPOR, U, V, SEAICE, SST, TSK

↑

Variable name in netcdf file

Stdout: Final normal exit information

```
   ENDING DATE-TIME    JUN 20,2013  23:20:16.431  171  THU   2456464
   PROGRAM GSI_ANL HAS ENDED.  IBM RS/6000 SP
. * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * . * .
```

```
0: ****************RESOURCE STATISTICS******************************
0:
0: The total amount of wall time                    = 551.066561
0:
    ( …  … )

0: ****************END OF RESOURCE STATISTICS*************************
```

Computer system other than IBM may not have "RESORCE STATISTICS" section

Developmental Testbed Center

Observation Fitting Statistics and Diagnostic Files

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

Why need to check fitting statistics

- Data Analysis: adjust background fields based on observation data so that analysis fields fit the observation better.

- GSI has a series of text files to provide statistic information on how background or analysis fields fit to the observations in each outer loop (fort.2*)

- GSI also has a series of binary files to save diagnostic information for each observation (diag*)

Statistic fitting files: 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 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 q 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	

File names are from fort.201 to fort.215 (or fit_*variable.analysis_time*)
Each file is for one observation variable

Statistic fitting files (Continue)

File name	Variables in file	Ranges/units
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), mhs_n18(, metop-a, n19), atms_npp, cris_npp, iasi616_metop-a, airs281SUBSET_aqua, avhrr3_n18(, n19, metop-b), amsre_aqua, ssmi_f15, ssmis_f16(, f17,f18,f19,f20), sndrD?_g11(, g12, g13, g14, g15), imgr_g11(, g12, g13, g14, g15), seviri_m08(,m09 m10)	
fort.208	fit of pcp_ssmi, pcp_tmi	
fort.209	fit of radar radial wind (rw)	
fort.210	fit of lidar wind (dw)	
fort.211	fit of super wind data (srw)	
fort.212	fit of GPS data	fractional difference
fort.213	fit of conventional sst data	C
fort.214	Tropical cyclone central pressure	
fort.215	Lagrangian data	

Satellite radiance

Radar radial wind

GPS

Developmental Testbed Center

Example: fit_p1.2011032212 (fort.201)

```
current fit of surface pressure data, ranges in mb
--------------------------------------------------
 pressure levels (hPa)=    0.0 2000.0
     it     obs      type  stype       count        bias         rms        cpen       qcpen
 o-g 01      ps       120   0000         141      0.2326      1.0637      1.6900      1.6613
 o-g 01      ps       180   0000        2210      0.2623      1.1016      1.4846      1.2854
 o-g 01      ps       181   0000         725      0.1493      1.2380      2.0757      1.8844
 o-g 01      ps       187   0000       12524      0.5000      1.0227      0.5437      0.5061
 o-g 01               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      type  stype       count        bias         rms        cpen       qcpen
 o-g 03      ps       120  0000          141     -0.1201      0.7662      0.8333      0.8317
 o-g 03      ps       180  0000         2213     -0.0232      0.8324      0.7786      0.7038
 o-g 03      ps       181  0000          733     -0.0313      1.0366      1.2592      1.1803
 o-g 03      ps       187  0000        12533      0.0266      0.6664      0.2326      0.2269
 o-g 03               all              15620      0.0155      0.7143      0.3636      0.3446
```

O-A

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

Example: fit_w1.2011032212 (fort.202)

Data Used in Analysis

					ptop	1000.0	900.0	800.0	600.0	100.0	50.0	0.0
	it	obs	type	styp	pbot	1200.0	1000.0	900.0	800.0	150.0	100.0	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	0.69	0.92	1.21	1.30	1.74	1.53	1.39
o-g	01	uv	220	0000	qcpen	0.69	0.92	1.21	1.30	1.72	1.52	1.38
o-g	01		all		count	1962	1400	1488	2688	1014	1071	18363
o-g	01		all		bias	-0.31	0.92	0.56	0.00	0.21	0.29	0.27
o-g	01		all		rms	3.02	3.88	4.68	4.65	5.59	5.36	4.74
o-g	01		all		cpen	0.45	0.53	0.85	0.94	1.62	1.49	1.09

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	80	24	8	19	2	1	489
o-g	01	rej	all		bias	5.40	2.74	2.94	6.31	24.83	28.94	1.94
o-g	01	rej	all		rms	12.45	10.82	23.91	44.74	61.69	59.27	21.23
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	00.00	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	4648	9606	1653	958	51	63	17344
o-g	01	mon	all		bias	-0.78	-0.25	0.18	-4.74	-0.64	0.15	-0.77
o-g	01	mon	all		rms	2.87	3.22	5.93	11.09	19.81	12.48	5.04
o-g	01	mon	all		cpen	1.16	1.27	0.87	0.68	0.00	0.00	1.14

Example: fit_rad1.2011032212 (fort.207)

```
RADINFO_READ:  jpch_rad=   2680
1 amsua_n15   chan=  1 var=   3.000 varch_cld=  9.100 use=  1 ermax=   4.500 b_rad=   10.00 pg_rad=   0.00 icld_det= 1
2 amsua_n15   chan=  2 var=   2.000 varch_cld= 13.500 use=  1 ermax=   4.500 b_rad=   10.00 pg_rad=   0.00 icld_det= 1
3 amsua_n15   chan=  3 var=   2.000 varch_cld=  7.100 use=  1 ermax=   4.500 b_rad=   10.00 pg_rad=   0.00 icld_det= 1
    . . .    . . .
  RADINFO_READ:  guess air mass bias correction coefficients below
   1            amsua_n15  -61.293985   -8.222002   55.207233  -59.849819  323.077772
   2            amsua_n15 -116.098935   -2.336239  124.728090    5.170702  436.024189
    . . .    . . .
```

```
it       satellite instrument    # read   # keep # assim  penalty     qcpnlty      cpen   qccpen
o-g 01 rad  n17        hirs3     202426    23636        0   0.0000     0.0000   0.0000   0.0000
o-g 01 rad  n18        hirs4          0        0        0   0.0000     0.0000   0.0000   0.0000
o-g 01 rad  metop-a    hirs4     189563    21736     4124   3954.6     3954.6  0.95892  0.95892
o-g 01 rad  g11        sndr           0        0        0   0.0000     0.0000   0.0000   0.0000
```

O-B

```
it       satellite instrument    # read   # keep # assim  penalty     qcpnlty      cpen   qccpen
o-g 03 rad   n17       amsua          0        0        0  0.0000     0.0000      0.0000   0.0000
o-g 03 rad   n18       amsua      63825    52309    31731  7162.7     7162.7     0.22573  0.22573
o-g 03 rad   metop-a   amsua      67800    49515    32366  14665.     14665.     0.45311  0.45311
o-g 03 rad   aqua      amsua          0        0        0  0.0000     0.0000      0.0000   0.0000
```

O-A

Developmental Testbed Center

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

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, has to 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 obs

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

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

GSI diagnosis file

Text file to save the content of diagnosis file

Example content of results_conv_anl

station ID	obs type	obs time	obs latitude	obs longitude	obs elev	usag	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',
 /
```

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
  - Include 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  1.882839321839552431E+05  5.284623958904349820E+03  1.697988409415417902E-03  0.000000000000000E+00
```

Inner iteration

Convergence information

- *cost_gradient.txt*

outer loop | Inner iteration

1	0	0.143023018550584849E+06	0.115641269650927745E+08	0.243989544270778198E-02	0.000000000000000000E+00
1	1	0.114807757869560824E+06	0.187098843447644310E+07	0.763011937910831189E-02	0.161792450059063342E+00
1	2	0.100531892757574562E+06	0.170165204992092540E+07	0.554614957747754343E-02	0.909493623030918297E+00
1	3	0.910942759598918346E+05	0.753744504713076283E+06	0.111976598917916254E-01	0.442948665532476193E+00
		. . .	. . .		
2	8	0.720276521927204303E+05	0.109468035525767671E+06	0.574413289558039185E-02	0.860466507061116603E+00
2	9	0.713988532488423079E+05	0.967618081750838755E+05	0.680342752911228341E-02	0.883927510988420262E+00
2	10	0.707405412993372593E+05	0.107314979924136685E+06	0.434178833835923046E-02	0.110906339957968947E+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

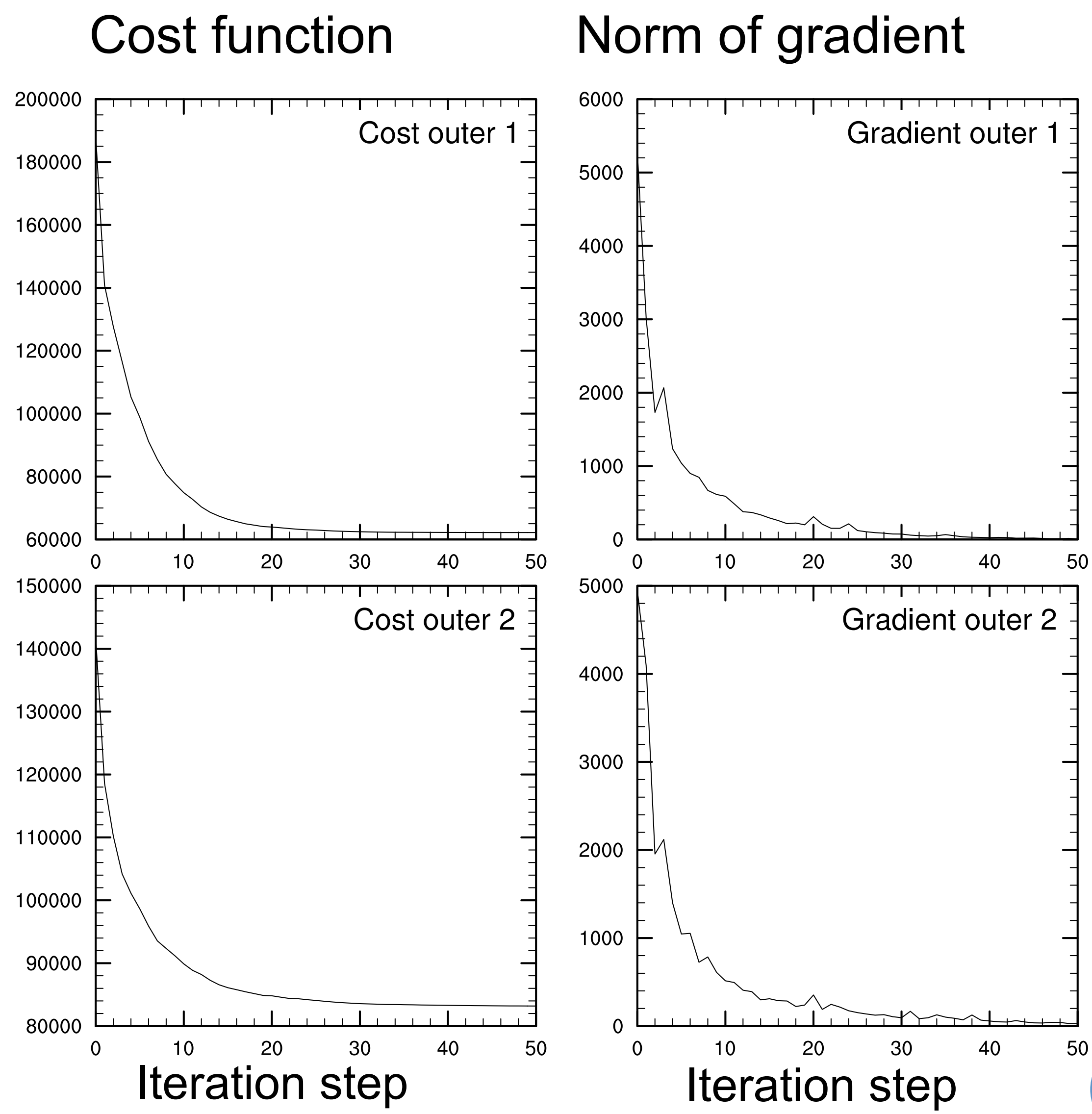
# of iterations in 2nd outer loop

figure file name and format

Check convergence: example figure

1st outer loop with 50 inter iteration

2nd outer loop with 50 inter iteration



Developmental Testbed Center

Analysis Increment

Details in User's Guide Section 4.8 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

begin

  cdf_analysis = addfile(”./gsiprd_2011032212_prepbufr/wrf_inout.cdf","r")

  cdf_bk = addfile(”./DTC/NA30km/bk/wrfinput_d01_2011-03-22_12:00:00.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: plot for NMM case

util/Analysis_Utilities/plots_ncl/GSI_singleobs_nmm.ncl

util/Analysis_Utilities/plots_ncl/fill_nmm_grid2.ncl

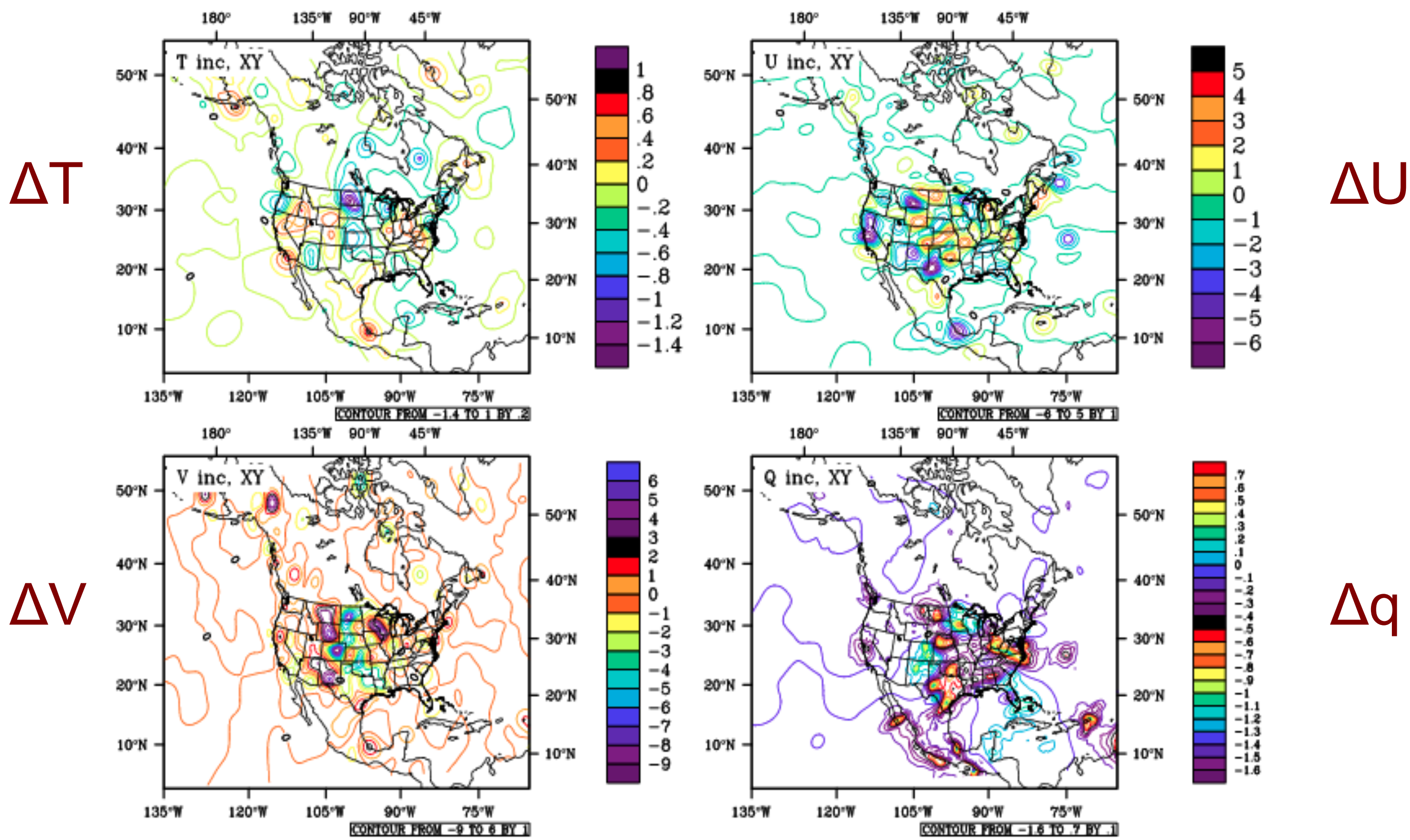
```
load "$NCARG_ROOT/lib/ncarg/nclex/gsun/gsn_code.ncl"
load "$NCARG_ROOT/lib/ncarg/nclscripts/csm/contributed.ncl"
load "./fill_nmm_grid2.ncl"

begin

  cdf_analysis = addfile(”./nmm_2007122000/gsiprd/wrf_inout.cdf","r")
  cdf_bk = addfile(”./nmm_netcdf/2007122000/wrfinput_d01_nmm_netcdf.cdf","r")
  Ta = cdf_analysis->T(0,:,:,:)
  Tb = cdf_bk->T(0,:,:,:)
  DT = Ta - Tb
    … …
  do k=0, nz-1
       fill_nmm_grid2(DT(k,:,:),nx,ny,f3d(k,:,:),1)
  end do
```

Convert from E grid to A grid

In: E grid field

Out: A grid field

Analysis increment: example



GSI analysis increment at the 15th level

Summary

Many methods can be used:

- We can diagnose GSI analysis by checking:
  - Standard output
  - Observation fitting statistics and diagnosis files
  - Convergence information
  - Analysis increment
- There are more:
  - Single observation test
  - Plot observation innovation overlay analysis increment
  - Forecast
  - …

Questions?

gsi_help@ucar.edu