GSI Fundamentals (3): Diagnostics

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 V3.1

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

Variable name in netcdf file

```
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
```

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

stdout: Check Fix Files Input

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
```

CRTM coefficients

```
INIT_CRTM: crtm_init() on path "./"
Read_SpcCoeff_Binary(INFORMATION) : FILE: ./amsua_n15.SpcCoeff.bin;
SpcCoeff RELEASE.VERSION:  7.03  N_CHANNELS=15
amsua_n15 AntCorr RELEASE.VERSION:  1.04  N_FOVS=30  N_CHANNELS=15
Read_ODPS_Binary(INFORMATION) : FILE: ./amsua_n15.TauCoeff.bin;
   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;
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
```

Developmental Testbed Center

stdout: Check Observations Input

```
READ_OBS:  read   21 hirs4   hirs4_metop-a     using ntasks=  2   2   2
READ_OBS:  read   36 amsub   amsub_n17         using ntasks=  2   0   2
READ_OBS:  read   37 mhs     mhs_n18           using ntasks=  2   2   2
READ_OBS:  read   38 mhs     mhs_metop-a       using ntasks=  2   0   2
READ_OBS:  read   63 mhs     mhs_n19           using ntasks=  2   2   2
READ_OBS:  read    1 ps      ps                using ntasks=  1   0   1

READ_BUFRTOVS: file=hirs4bufr  type=hirs4   sis=hirs4_metop-a       nread=
189563 ithin= 1 rmesh=120.000000 isfcalc= 0 ndata=   1178 ntask=  2
READ_BUFRTOVS: file=hirs3bufr  type=hirs3   sis=hirs3_n17  nread=  202426
ithin= 1 rmesh=120.000000 isfcalc= 0 ndata=  1235 ntask=  2
READ_PREPBUFR: file=prepbufr   type=uv      sis=uv   nread= 91930
ithin= 0 rmesh=120.000000 isfcalc= 0 ndata=     71038 ntask=  1
READ_BUFRTOVS: file=amsuabufr  type=amsua   sis=amsua_n18   nread=  63825
ithin= 2 rmesh= 60.000000 isfcalc= 0 ndata=     47704 ntask=  1
```

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     18109      5725
OBS_PARA: sst             0         0          6        3
OBS_PARA: gps_ref         3538      5580      2277      6768
OBS_PARA: hirs3     n17   0         0         30        35
```

Developmental Testbed Center

stdout: Check outer loop and inner iteration

1st outer loop

Inner iteration

```
grepcost J,Jb,Jo,Jc,Jl =       1      0     1.430230185505848494E+05 0.000000000000000000E+00
1.430230185505848494E+05 0.000000000000000000E+00 0.000000000000000000E+00
grepgrad grad,reduction=       1      0     3.400606852473948493E+03 1.000000000000000000E+00
pcgsoi: cost,grad,step =       1      0     1.430230185505E+05 3.4006068524E+03 2.4398954427E-03
pcgsoi: gnorm(1:2),b= 1.870988434476E+06 1.870988434476 4E+06 1.617924500590633419E-01
stprat 0.680228405155974
stprat 4.467539812170933E-015
Minimization iteration 1
grepcost J,Jb,Jo,Jc,Jl =       1      1     1.148077578695608245E+05 6.884228595044261567E+01
1.147389155836103746E+05 0.000000000000000000E+00 0.000000000000000000E+00
grepgrad grad,reduction=       1      1     1.367840792810494804E+03 4.022343223284949310E-01
```

2nd outer loop

Inner iteration

```
grepcost J,Jb,Jo,Jc,Jl =       2      0     9.709172928659517493E+04 5.242424325932335705E+03
9.184930496066283376E+04 0.000000000000000000E+00 0.000000000000000000E+00
grepgrad grad,reduction=       2      0     2.458528763445824097E+03 1.000000000000000000E+00
pcgsoi: cost,grad,step =       2      0     9.709172928665E+04 2.45852876344E+03 2.0910838314E-03
pcgsoi: gnorm(1:2),b= 1.39687516886273E+06 1.39687516886273E+06 2.311037592468580E-01
stprat 0.326022963208770
stprat 3.222291707173269E-015
Minimization iteration 1
grepcost J,Jb,Jo,Jc,Jl =       2      1     8.445245812257242505E+04 5.411029823593031324E+03
7.904142829897940101E+04 0.000000000000000000E+00 0.000000000000000000E+00
grepgrad grad,reduction= 2 1 1.181894736794584333E+03 4.807325236000348778E-01
```

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	53749	4.8794660264558479E+04	0.908
	Nobs	Jo	Jo/n
Jo Global	**131113**	**1.4302301855058485E+05**	**1.091**

After 1st outer loop

J_o

surface pressure	15620	6.0353797873191743E+03	0.386
Temperature	11913	1.0272593707203929E+04	0.862
wind	36796	2.2776352264954159E+04	0.619
Moisture	4416	2.2085428390180150E+03	0.500
gps	8860	8.9698796153000221E+03	1.012
Radiance	84607	4.1586556746867529E+04	0.492
	Nobs	Jo	Jo/n
Jo Global	**162212**	**9.1849304960662834E+04**	**0.566**

After 2nd outer loop

surface pressure	15620	5.6788282318981810E+03	0.364
temperature	11913	9.6271243544834779E+03	0.808
Wind	36794	2.1433478710069721E+04	0.583
Moisture	4417	2.0790935066632146E+03	0.471
gps	8868	8.1188601604511678E+03	0.916
Radiance	100900	3.2527599143479922E+04	0.322
	Nobs	Jo	Jo/n
Jo Global	**178512**	**7.9464984107045690E+04**	**0.445**

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 JUL 13,2012 21:58:55.998 195 FRI 2456122
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

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

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 including: amsua_n15(, n16, n17, n18, metop-a, aqua, n19), amsub_n15(, n16, n17,), hirs3_n16(, n17), hirs4_n18 (, metop-a, n19), mhs_n18(, metop-a, n19), hirs2_n14, iasi616_metop-a, airs281SUBSET_aqua, avhrr3_n16(, n17, n18), amsre_aqua, ssmi_f13(, f14, f15), ssmis_f16, msu_n14, sndr_g11(, g12, g13), imgr_g11(, g12, g13), sndrD1(, 2, 3, 4)_g11(, g12,g13),	
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 srw1, srw2	
fort.212	fit of GPS data	[illegible] 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 RO

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

it		obs	type	styp	top / 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	0.69	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
o-g	01		all		qcpen	0.45	0.53	0.84	0.92	1.60	1.48	1.08

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	34	10	8	19	2	1	429
o-g	01		rej	all		bias	10.52	4.64	2.94	6.31	24.83	28.94	2.00
o-g	01		rej	all		rms	17.64	13.49	23.91	44.74	61.69	59.27	22.52
o-g	01		rej	all		cpen	0.00	0.00	0.00	0.00	0.00	0.00	0.00
o-g	01		rej	all		qcpen	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	0.65	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	4091	9546	1653	958	51	63	16727
o-g	01		mon	all		bias	-0.92	-0.25	0.18	-4.74	-0.64	0.15	-0.81
o-g	01		mon	all		rms	2.92	3.22	5.93	11.09	19.81	12.48	5.11
o-g	01		mon	all		cpen	1.32	1.28	0.87	0.68	0.00	0.00	1.18
o-g	01		mon	all		qcpen	1.26	1.24	0.83	0.60	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
   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
   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
     . . .    . . .
 RADINFO_READ:  guess air mass bias correction coefficients below
   1            amsua_n15     0.472353   -0.231512    0.291223    0.000634   -0.148959
   2            amsua_n15    -0.677697    0.382025    1.424922   -0.000061    0.016514
     . . .    . . .
```

```
it  satellite instrument   # read  # keep # assim  penalty    qcpnlty    cpen       qccpen
o-g 01 rad  n16     hirs3       0       0       0  0.0000     0.0000   0.0000       0.0000
o-g 01 rad  n17     hirs3  202426    1235       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    1178     264  274.88     274.88   1.0412       1.0412
```

O-B

```
it satellite instrument  # read  # keep  # assim  penalty    qcpnlty   cpen       qccpen
o-g 03 rad  n16     hirs3      0       0       0  0.0000     0.0000  0.0000       0.0000
o-g 03 rad  n17     hirs3  02426    1235       0  0.0000     0.0000  0.0000       0.0000
o-g 03 rad  n18     hirs4      0       0       0  0.0000     0.0000  0.0000       0.0000
o-g 03 rad metop-a hirs4  189563    1178     256  217.41     217.41  0.84926      0.84926
```

O-A

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*

Developmental Testbed Center

Observation departure for each obs

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

```
&iosetup
   infilename='./diag_conv_anl',
   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_n15_anl',
  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 'penalty,grad ,a,b='  fort.220 | sed -e 's/penalty,grad ,a,b=//g' > cost_gradient.txt
```

```
penalty,grad ,a,b=  1   1   0.11480775E+06  0.18709884E+07  0.76301193E-02  0.16179245E+00
```

outer loop

Inner iteration

Developmental Testbed Center

Convergence information

- *cost_gradient.txt*

outer loop | Inner iteration

outer loop	Inner iteration	Penalty (cost function)	Norm of gradient		
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

DTC Developmental Testbed Center

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

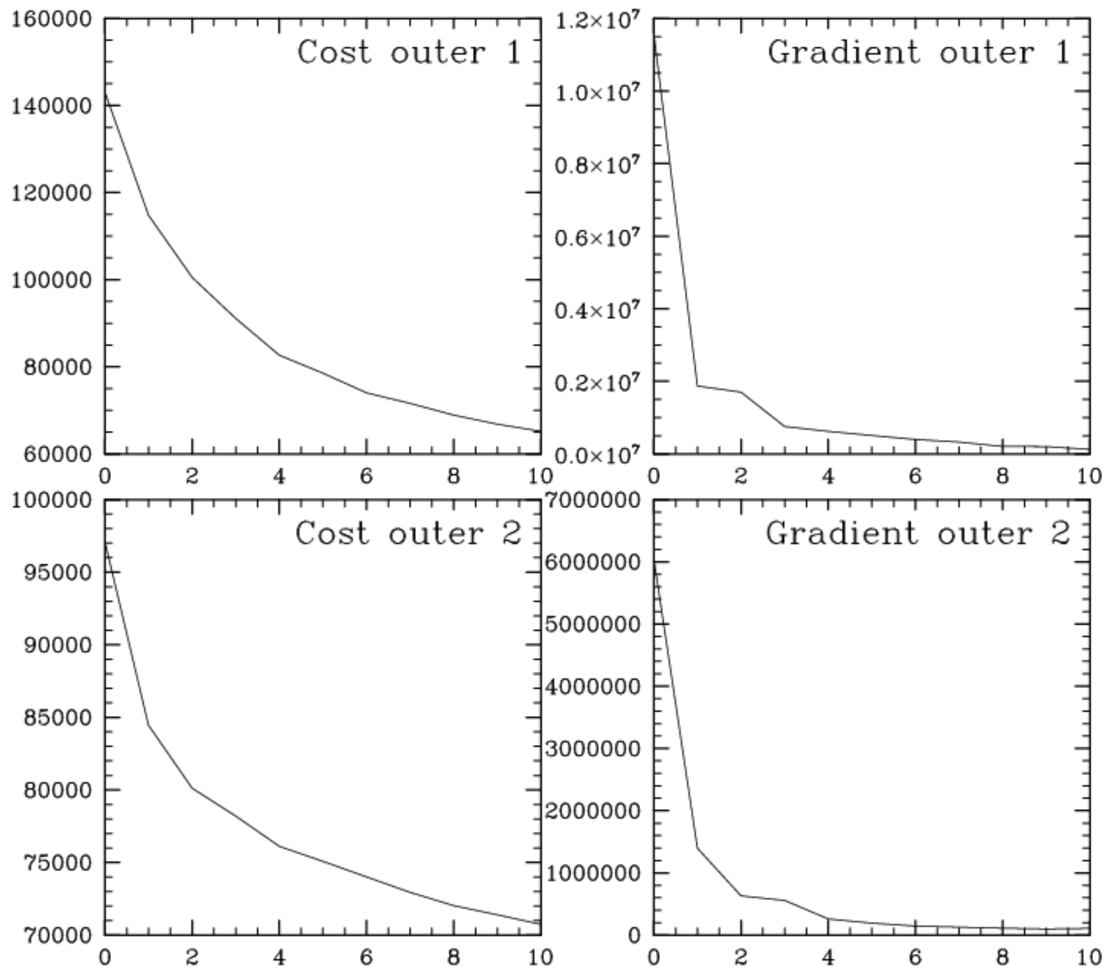
figure file name and format

Check convergence: example figure

1st outer loop with 10 inter iteration

2nd outer loop with 10 inter iteration

Cost function   Norm of gradient



Iteration step   Iteration step

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
```

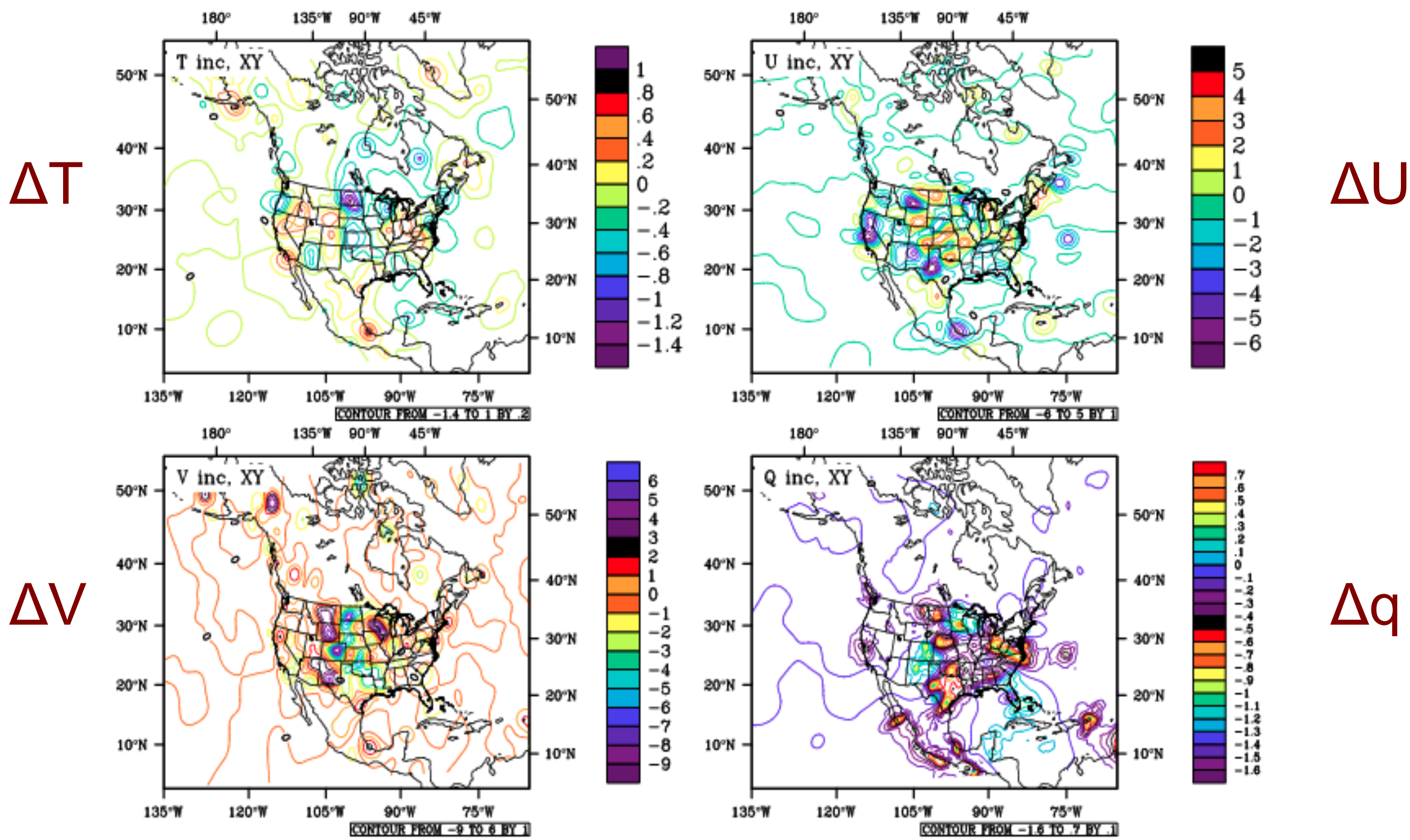
Convert from E grid to A grid

In: E grid field

Out: A grid field

Developmental Testbed Center

Analysis increment: example



GSI analysis increment at the 15th level

Developmental Testbed Center

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