GSI Diagnostic

Ming Hu

Developmental Testbed Center

What is this talk for?

- From previous lectures, you learned how to:
  - Install GSI
  - Run GSI cases
- Now, let's see how to diagnose GSI analysis results by checking:
  - Standard output
  - Observation fitting statistic files
  - Optimization convergence information
  - Analysis increment figure

Standard Output

Details in User's Guide Section 4.1

Highlight several important points

Check Background Input

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 -7269735
0:  end_index= 69 64 45 -7269735
0:  k,max,min,mid T=  1 309.9411316 264.5114136 289.7205811
0:  k,max,min,mid T=  2 310.6200562 269.5698547 295.0413208
0:  k,max,min,mid T=  3 311.5386047 272.4312744 296.7247009

0:  k,max,min,mid T= 43 486.2092896 436.1306763 461.4866943
0:  k,max,min,mid T= 44 498.1362000 456.4700012 478.7089233
0:  k,max,min,mid T= 45 510.0127563 472.5627441 494.7407227
```

K | Maximum | Minimum | Central grid

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

DTC Developmental Testbed Center

Check Fix Files Input

convinfo

```
0:READ_CONVINFO: ps      120  0  1  3.0  0  0  0  4.0  3.0  1.0  4.0
0:READ_CONVINFO: ps      132  0 -1  3.0  0  0  0  4.0  3.0  1.0  4.0
0:READ_CONVINFO: t       120  0  1  3.0  0  0  0  8.0  5.6  1.3  8.0
0:READ_CONVINFO: t       180  0  1  3.0  0  0  0  7.0  5.6  1.3  7.0
0:READ_CONVINFO: uv      220  0  1  3.0  0  0  0  8.0  6.0  1.4  8.0
0:READ_CONVINFO: uv      280  0  1  3.0  0  0  0  6.0  6.1  1.4  6.0
0:READ_CONVINFO: spd     283  0  1  3.0  0  0  0  8.0  6.1  1.4  8.0
```

CRTM coefficients

0: **Read_SpcCoeff_Binary**(INFORMATION) : FILE: ./**hirs3_n16.SpcCoeff.bin**;
0: SpcCoeff RELEASE.VERSION: 7.01 N_CHANNELS=19
0: **Read_TauCoeff_Binary**(INFORMATION) : FILE: **./hirs3_n16.TauCoeff.bin**;
0: TauCoeff RELEASE.VERSION: 5.04 N_ORDERS=10 N_PREDICTORS= 6 N_ABSORBERS= 3 N_CHANNELS= 19 N_SENSORS= 1
0: Read_CloudCoeff_Binary(INFORMATION) : FILE: ./CloudCoeff.bin;
0: CloudCoeff RELEASE.VERSION: 2.02 N_FREQUENCIES(MW)= 31 N_FREQUENCIES(IR)= 701 N_RADII(MW)= 6 N_RADII(IR)= 6 N_TEMPERATURES= 5 N_DENSITIES= 3 N_LEGENDRE_TERMS=38 N_PHASE_ELEMENTS= 6

Check Observations Input

```
   0: READ_OBS:  read  1 ps       ps                    using ntasks= 1 0 2 1
   0: READ_OBS:  read  2 t        t                     using ntasks= 1 0 3 1
   0: READ_OBS:  read  3 q        q                     using ntasks= 1 0 0 1
   0: READ_OBS:  read  4 uv       uv                    using ntasks= 1 0 1 1
   0: READ_OBS:  read  5 spd      spd                   using ntasks= 1 0 2 1
   0: READ_OBS:  read  7 dw       dw                    using ntasks= 1 0 3 1
   0: READ_OBS:  read  8 sst      sst                   using ntasks= 1 0 0 1

READ_PREPBUFR: messages/reports = 681 / 71658 ntread = 1
READ_PREPBUFR: file=prepbufr type=t sis=t nread= 11452 ithin= 0 rmesh=120.000
               isfcalc= 0 ndata= 11280 ntask= 1
READ_PREPBUFR: messages/reports = 681 / 71658 ntread = 1
READ_PREPBUFR: file=prepbufr type=q sis=q nread= 11379 ithin= 0 rmesh=120.000
               isfcalc= 0 ndata= 10186 ntask= 1
```

Observation distribution in an analysis using 4 processors

```
   3:OBS_PARA: ps                     291     508    1140    1561
   3:OBS_PARA: t                      484     887    1730    2729
   3:OBS_PARA: q                      464     868    1642    2568
   3:OBS_PARA: uv                    1146    2067    2766    5162
   3:OBS_PARA: sst                      0       0      47       0
   3:OBS_PARA: pw                      13      18      44      17

   3:OBS_PARA: hirs3     n16           21      21       0       0
   3:OBS_PARA: amsua     n15          179     189     149     224
```

Check outer loop and inner iteration

1st outer loop

Inner iteration

```
0:grepcost J,Jb,Jo,Jc,Jl =   1   0  1.6048921E+04   0.0E+0   1.60489216E+04 0.0E+0 0.0E+0
0:grepgrad grad,reduction=   1   0  4.622906854790679176E+02  1.000000000000000000E+00
0:pcgsoi: cost,grad,step =   1   0  1.60489216E+04  4.6229068547E+02  1.4816403979E-02
0:pcgsoi: gnorm(1:2),b=  1.24089076555E+05  1.240890765553E+05  5.8063507409044E-01
0:  stprat  0.5188388147779914O3E-01
0:  stprat  0.2463118129990371 77E-15
0: Minimization iteration 1
0:grepcost J,Jb,Jo,Jc,Jl =   1   1  1.288246E+04   4.6915E+01  1.283555E+04 0.0E+0 0.0E+0
0:grepgrad grad,reduction=   1   1  3.522627947361617657E+02  7.619941430814570760E-01
0:pcgsoi: cost,grad,step =   1   1  1.288246824862E+04  3.522627947E+02  1.408558895E-02
0:pcgsoi: gnorm(1:2),b=  3.9226971306926E+04  3.9226971306926E+04  3.1611945544163E-01
0:  stprat  0.364906062575707624
0:  stprat  0.15643113915167427 8E-14224
```

2nd outer loop

Inner iteration

```
0: Minimization iteration 0
0:grepcost J,Jb,Jo,Jc,Jl =   2   0  9.9610405E+03   8.5076644E+02  9.1102740E+03 0.0 0.0
0:grepgrad grad,reduction=   2   0  2.291813874420873560E+02  1.000000000000000000E+00
0:pcgsoi: cost,grad,step =   2   0  9.96104052193E+03  2.29181387442E+02  7.15279261E-03
0:pcgsoi: gnorm(1:2),b=  1.13035507282E+04  1.13035507282E+04  2.15206903713E-01
0:  stprat  0.319264972580440953
0:  stprat  0.577830902990640559E-14
0: Minimization iteration 1
0:grepcost J,Jb,Jo,Jc,Jl =   2   1  9.58534646E+03  8.7366356E+02  8.7116829E+03 0.0 0.0
0:grepgrad grad,reduction=   2   1  1.063181580364677217E+02  4.639039811351767240E-01
0:pcgsoi: cost,grad,step =   2   1  9.58534646763E+03  1.06318158036E+02  1.05074549E-02
0:pcgsoi: gnorm(1:2),b=  5.4234537548554E+03  5.4234537548554E+03  4.7980089488985E-01
0:  stprat  0.410968433100896591
0:  stprat  0.157538354385690312E-13
```

Check Analysis Result Output

```
0:   ordering=XY
0:   WrfType,WRF_REAL= 104 104
0:   ndim1= 2
0:   staggering= N/A
0:   start_index= 1 1 1 -7269735
0:   end_index1= 69 64 45 -7269735
0:   k,max,min,mid T=   1  309.9584656  264.4796753  290.9471130
0:   k,max,min,mid T=   2  310.6225281  269.6447144  296.2458191

0:   k,max,min,mid T=  44  496.4794922  457.3758850  478.2719116
0:   k,max,min,mid T=  45  509.2687378  475.0305481  494.7050171
0:   rmse_var=T            K  Maximum      Minimum      Central grid
```

QVAPOR, U, V, SMOIS, XICE, SST, TSK

Variable name in netcdf file

Observation Fitting Statistic

Details in User's Guide Section 4.5

Why need to check fitting statistic

- 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 outer loop fields fit to the certain observation variable (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	current fit of precipitation water data	mm
fort.206	fit of ozone observations from sbuv6_n14 (, _n16, _n17, _n18) and sbuv8_n16 (, _n17, _n18, _n19)	

File names are from fort.201 to fort.215 (or fit_*)
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 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	pcp_ssmi, pcp_tmi	
fort.209	Rw	
fort.210	Dw	
fort.211	srw1, srw2	
fort.212	GPS RO	
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_t1.2010050700 (fort.203)

current fit of temperature data, ranges in K

it	obs	type	styp	ptop / pbot	1000.0 / 1200.0	900.0 / 999.9	800.0 / 899.9	600.0 / 799.9	400.0 / 599.9	300.0 / 399.9	250.0 / 299.9	200.0 / 249.9	150.0 / 199.9	100.0 / 149.9	50.0 / 99.9	0.0 / 2000.0
o-g 01	t	120	0000	count	44	180	214	381	405	200	93	147	247	334	396	3432
o-g 01	t	120	0000	bias	3.67	2.12	0.51	0.23	-0.33	-0.63	-0.66	-1.49	-0.45	-1.04	-1.21	-0.40
o-g 01	t	120	0000	rms	4.36	2.77	1.45	1.19	0.83	1.04	1.43	2.09	1.74	1.89	2.38	**1.81**
o-g 01	t	130	0000	count	0	0	0	0	0	0	3	12	2	0	0	18
o-g 01	t	130	0000	bias	0.00	0.00	0.00	0.00	0.00	0.00	-0.55	-0.50	2.33	0.00	0.00	-0.04
o-g 01	t	130	0000	rms	0.00	0.00	0.00	0.00	0.00	0.00	0.57	1.31	2.51	0.00	0.00	1.48
o-g 01	t	180	0000	count	714	79	0	0	0	0	0	0	0	0	0	793
o-g 01	t	180	0000	bias	2.65	-0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.38
o-g 01	t	180	0000	rms	3.66	1.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.51
o-g 01		all		count	758	259	214	381	405	200	96	159	249	334	396	4243
o-g 01		all		bias	2.70	1.46	0.51	0.23	-0.33	-0.63	-0.65	-1.41	-0.42	-1.04	-1.21	0.12
o-g 01	O-B	all		rms	3.70	2.47	1.45	1.19	0.83	1.04	1.41	2.04	1.74	1.89	2.38	**2.23**

current fit of temperature data, ranges in K

it	obs	type	styp	ptop / pbot	1000.0 / 1200.0	900.0 / 999.9	800.0 / 899.9	600.0 / 799.9	400.0 / 599.9	300.0 / 399.9	250.0 / 299.9	200.0 / 249.9	150.0 / 199.9	100.0 / 149.9	50.0 / 99.9	0.0 / 2000.0
o-g 03	t	120	0000	count	44	180	214	381	405	200	93	147	247	334	396	3432
o-g 03	t	120	0000	bias	2.38	1.36	0.13	0.13	-0.07	-0.10	-0.03	-0.50	0.03	-0.24	-0.27	0.02
o-g 03	t	120	0000	rms	2.75	1.94	1.06	0.90	0.59	0.61	1.02	1.23	1.37	1.19	1.90	**1.34**
o-g 03	t	130	0000	count	0	0	0	0	0	0	3	12	2	0	0	18
o-g 03	t	130	0000	bias	0.00	0.00	0.00	0.00	0.00	0.00	-0.06	-0.04	1.70	0.00	0.00	0.29
o-g 03	t	130	0000	rms	0.00	0.00	0.00	0.00	0.00	0.00	0.26	1.08	1.87	0.00	0.00	1.24
o-g 03	t	180	0000	count	714	79	0	0	0	0	0	0	0	0	0	793
o-g 03	t	180	0000	bias	0.84	-0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.74
o-g 03	t	180	0000	rms	2.34	1.38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.27
o-g 03		all		count	758	259	214	381	405	200	96	159	249	334	396	4243
o-g 03		all		bias	0.93	0.88	0.13	0.13	-0.07	-0.10	-0.03	-0.46	0.04	-0.24	-0.27	0.16
o-g 03	O-A	all		rms	2.37	1.79	1.06	0.90	0.59	0.61	1.01	1.22	1.37	1.19	1.90	**1.56**

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

Observation departure for each obs

- Diagnostic files:

```
diag_amsua_metop-a_anl.2010050700   diag_amsub_n16_anl.2010050700
diag_amsua_metop-a_ges.2010050700   diag_amsub_n16_ges.2010050700
diag_amsub_n17_anl.2010050700       diag_conv_anl.2010050700
Diag_amsub_n17_ges.2010050700       diag_conv_ges.2010050700
```

- 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/diag)
    - read_diag_conv.f90 (diag_conv*)
    - read_diag_rad.f90 (diag_amsub_n16* …)
  - see User's Guide appendix A.2 for details

Convergence information

Details in User's Guide Section 4.6

Find convergence information

- From stdout
  - See example on this talk
- From fort.220
  - Include many details
    - Cost function and gradient
    - Contribution from background and each data type
    - …
  - DTC provides a tool to read this file
    - util/cost_grad/read_fort220.f90
  - DTC provides a NCL scripts to make plot
    - util/cost_grad/GSI_cost_gradient.ncl

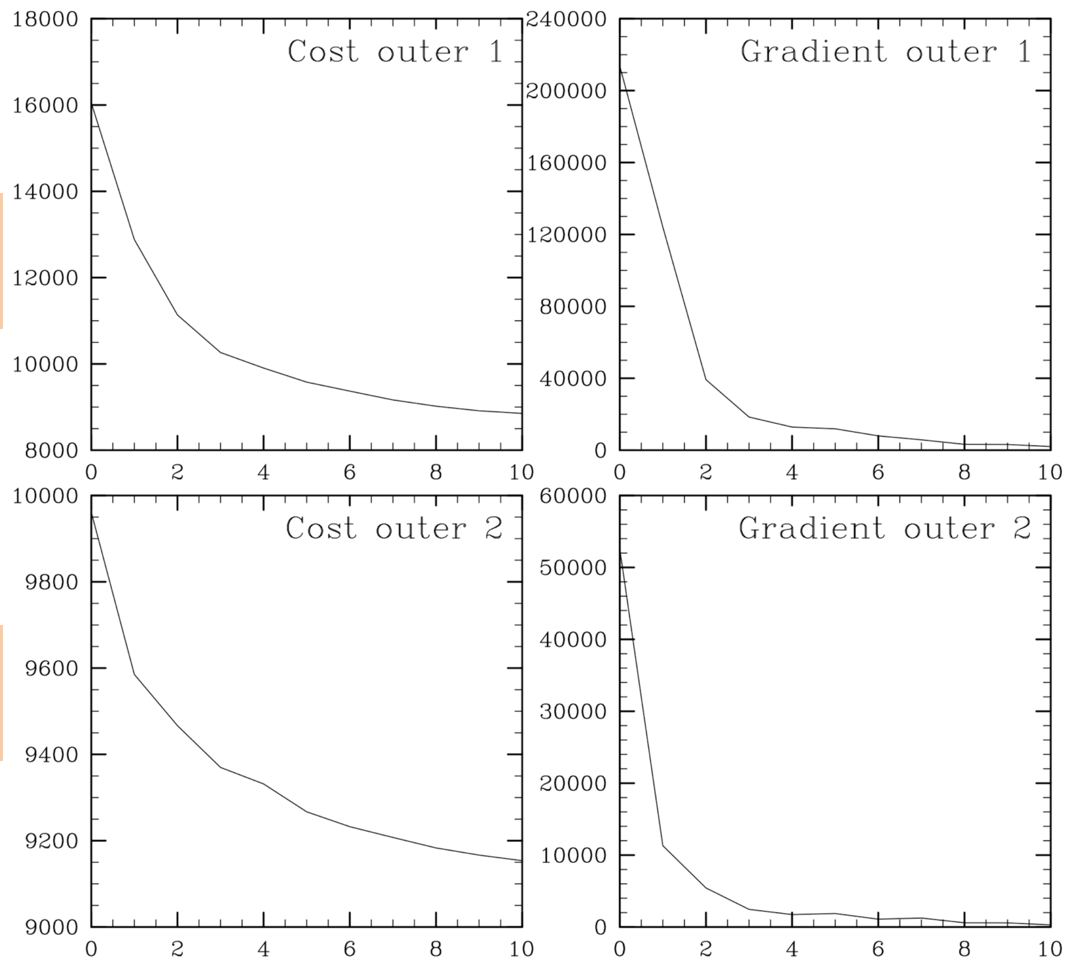
Example figures: check convergence

First outer loop with 10 inter iteration

Second outer loop with 10 inter iteration

Cost function

Norm of gradient



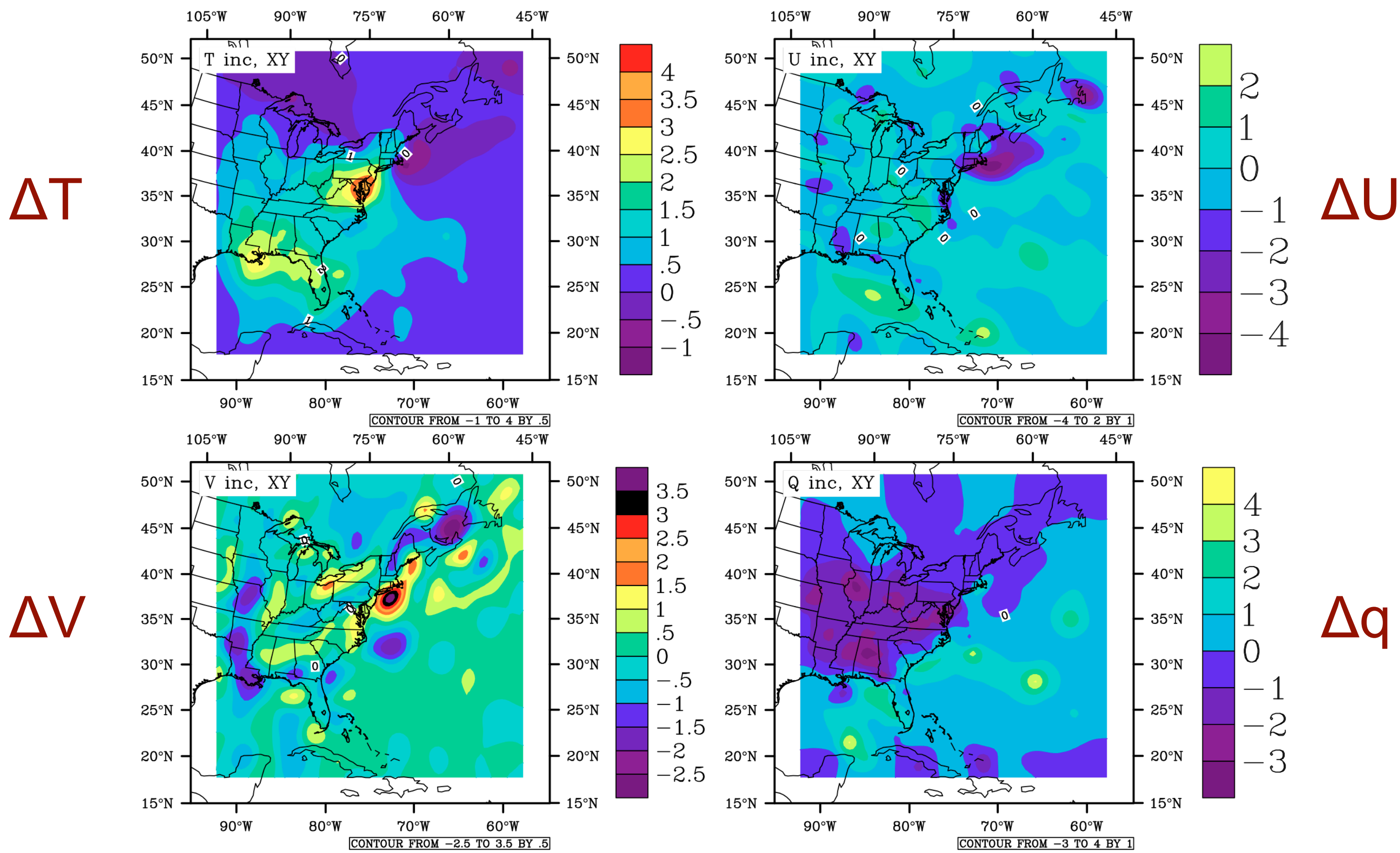
Iteration step

Iteration step

Analysis Increment

Details in User's Guide Section 4.7

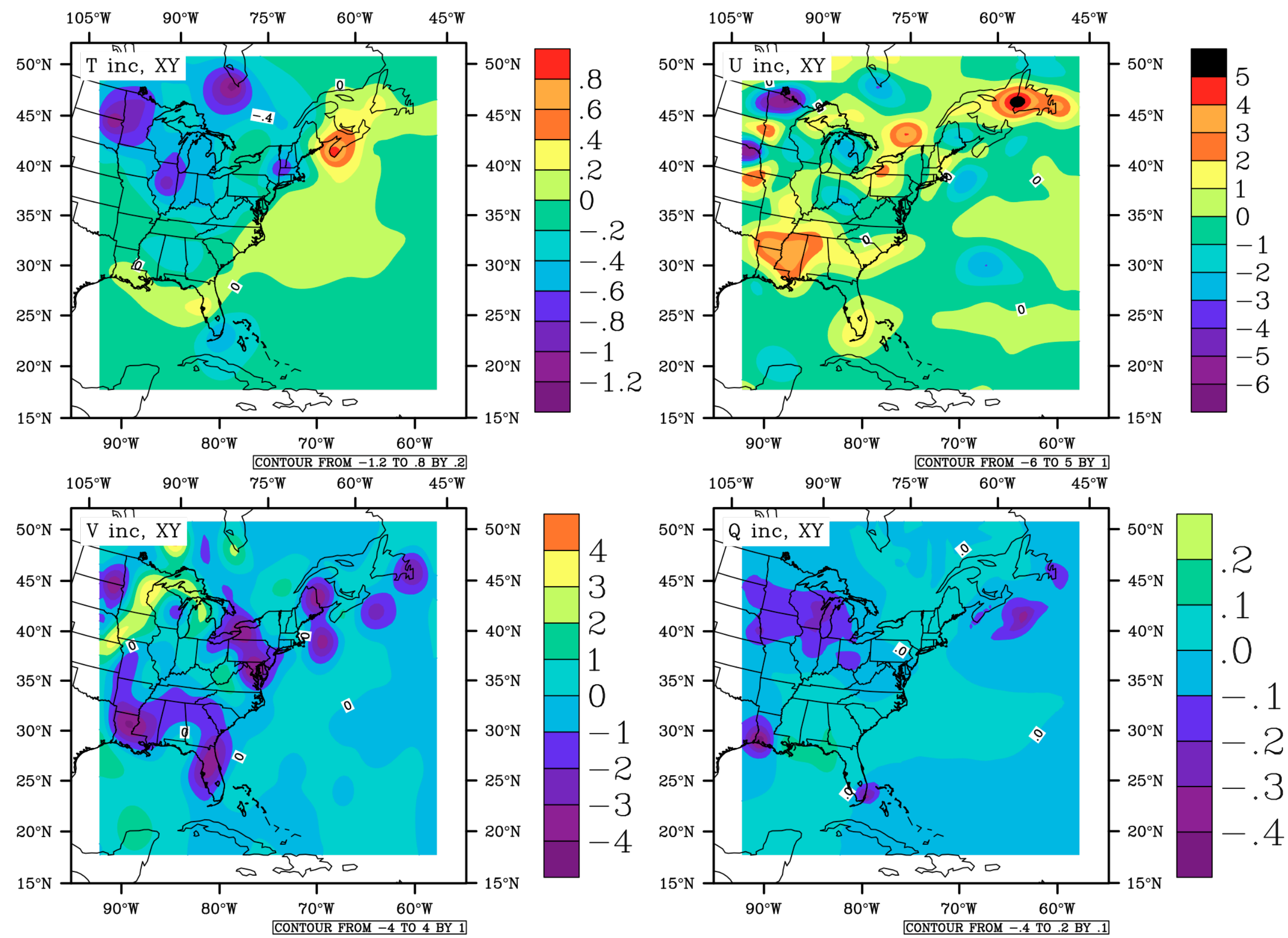
Analysis increment: surface



GSI analysis with conventional data

Analysis increment: upper level

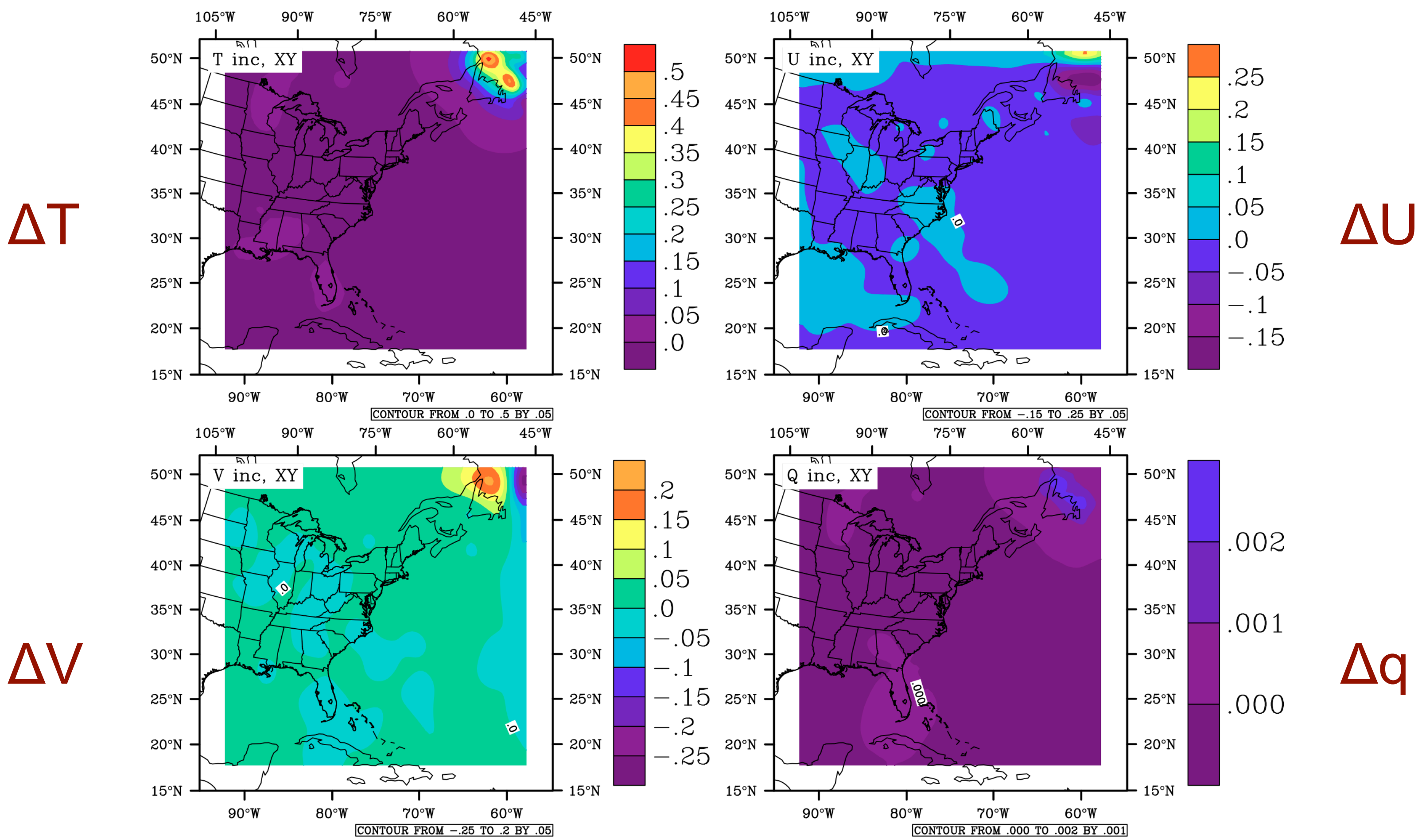
ΔT ΔU



ΔV Δq

GSI analysis with conventional data

Data impact: radiance (metop-a amsua)



Difference between analyses with and without radiance. Both use conventional data.

Developmental Testbed Center

Summary

Many methods can be used:

- We can diagnose GSI analysis by checking:
  - Standard output
  - Observation fitting statistic
  - Optimization convergence
  - Analysis increment figures

- There are more:
  - Single observation test
  - Observation innovation overlay analysis increment figure
  - Forecast
  - …

Questions?

gsi_help@ucar.edu