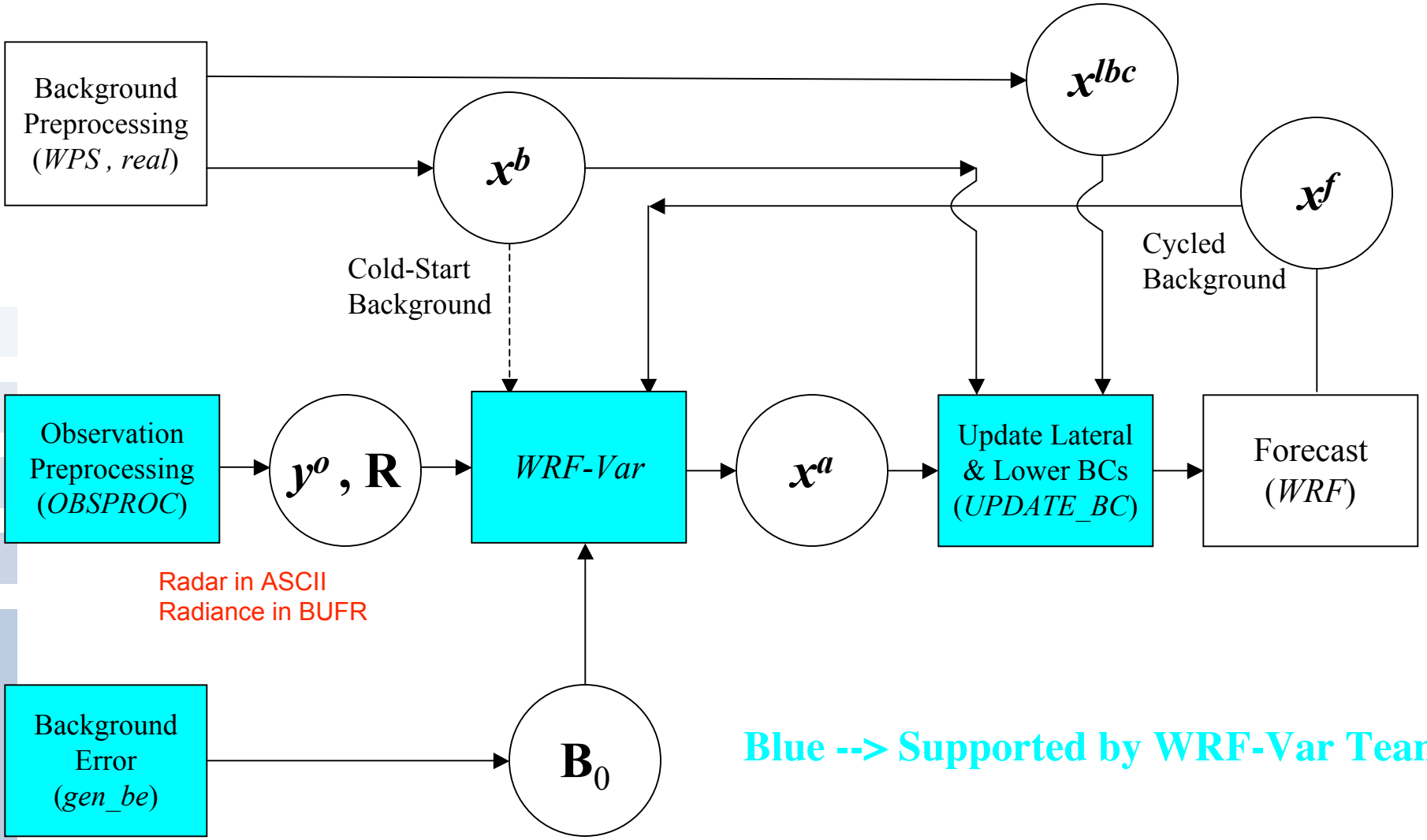
WRF-Var Setup, Run and Diagnostics

Hui Shao (huishao@ucar.edu)

Hui-Chuan Lin, Meral Demirtas, Yongrun Guo, Syed Rizvi,
Xin Zhang and Hans Huang

WRF-Var Tutorial, July 20-22, 2009

• WRF-Var in the WRF Modeling System



Blue --> Supported by WRF-Var Team

Outline

- Installing WRF-Var
- Running WRF-Var code
- Running UPDATE_BC
- WRF-Var diagnostics
  - Also check "**WRF-Var Tools and Verification Package**" talk.
- Basic runtime options (namelist)

This talk is tailored based on WRF-Var V3.1.1.

Installing WRF-Var

3D-Var

Source Codes

NCAR

- Download WRF-Var source code (WRFDAv3_1_1.tar.gz) from http://www.mmm.ucar.edu/wrf/users/download/get_sources.html

FORTRAN90 Compiler

NCAR

We have tested

- IBM: XLF
- SGI Altix: INTEL
- PC/Linux: PGI, INTEL, GFORTRAN
- MAC: G95, PGI

More? Please let us know.

Libraries Required by WRF-Var

- NetCDF: Network Common Data Form
  - http://www.unidata.ucar.edu/software/netcdf/
- BLAS: Basic Linear Algebra Subprograms
  - http://netlib.org/blas/
- LAPACK: Linear Algebra PACKage
  - http://netlib.org/lapack/

Provided in WRFDA V3.1.1:

WRFDA/var/external

Set environment variables:

> setenv NETCDF $your_installation_dir/netcdf

✓Make sure the required libraries are all compiled using the same compiler that will be used to build WRF-Var, since the libraries produced by one compiler may not be compatible with code compiled with another.

Optional Libraries

NCAR

- BUFR: Provided in WRFDA V3.1.1 (WRFDA/var/external).

- CRTM: Community Radiative Transfer Model
  Required only if assimilate radiance data.
  - ftp://ftp.emc.ncep.noaa.gov/jcsda/CRTM/
- RTTOV: Radiative Transfer for TOVS
  Required only if assimilate radiance data.
  - http://www.metoffice.gov.uk/science/creating/working_together/nwpsaf_public.html

Set environment variables:

```
>setenv BUFR    1
>setenv CRTM    $your_installation_dir/crtm
>setenv RTTOV   $your_installation_dir/rttov
```

✓ Link $RTTOV/librttov.a to $RTTOV/src/librttov8.7.a
✓ Link $CRTM/libcrtm.a to $CRTM/src/libCRTM.a

Configure WRF-Var

> cd $your_sourcecode_dir/WRFDA

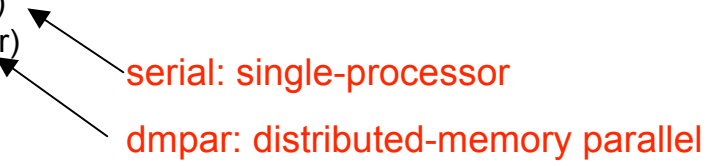
> ./configure wrfda

– configure.wrf is created.

```
checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /users/noname/work/external/g95/netcdf-3.6.1
PHDF5 not set in environment. Will configure WRF for use without.
$JASPERLIB or $JASPERINC not found in environment, configuring to build without grib2 I/O...
------------------------------------------------------------------------
Please select from among the following supported platforms.

1. Darwin (MACOS) PGI compiler with pgcc (serial)
2. Darwin (MACOS) PGI compiler with pgcc (dmpar)
3. Darwin (MACOS) intel compiler with icc (serial)
4. Darwin (MACOS) intel compiler with icc (dmpar)
5. Darwin (MACOS) intel compiler with cc (serial)
6. Darwin (MACOS) intel compiler with cc (dmpar)
7. Darwin (MACOS) g95 with gcc (serial)
8. Darwin (MACOS) g95 with gcc (dmpar)
9. Darwin (MACOS) xlf (serial)
10. Darwin (MACOS) xlf (dmpar)

Enter selection [1-10] : 7
------------------------------------------------------------------------
Compile for nesting? (0=no nesting, 1=basic, 2=preset moves, 3=vortex following) [default 0]:
Configuration successful. To build the model type compile .
```



Compile WRF-Var

> ./compile all_wrfvar

- 31 executables in the var/da (linked to var/build):
  - da_wrfvar.exe: WRF-Var
  - da_updated_bc.exe: update_bc
  - gen_be_stage0_wrf.exe, …: gen_be
  - da_advance_time.exe: time manipulation
- OBSPROC executable in the var/obsproc/src
  - obsproc.exe: OBSPROC

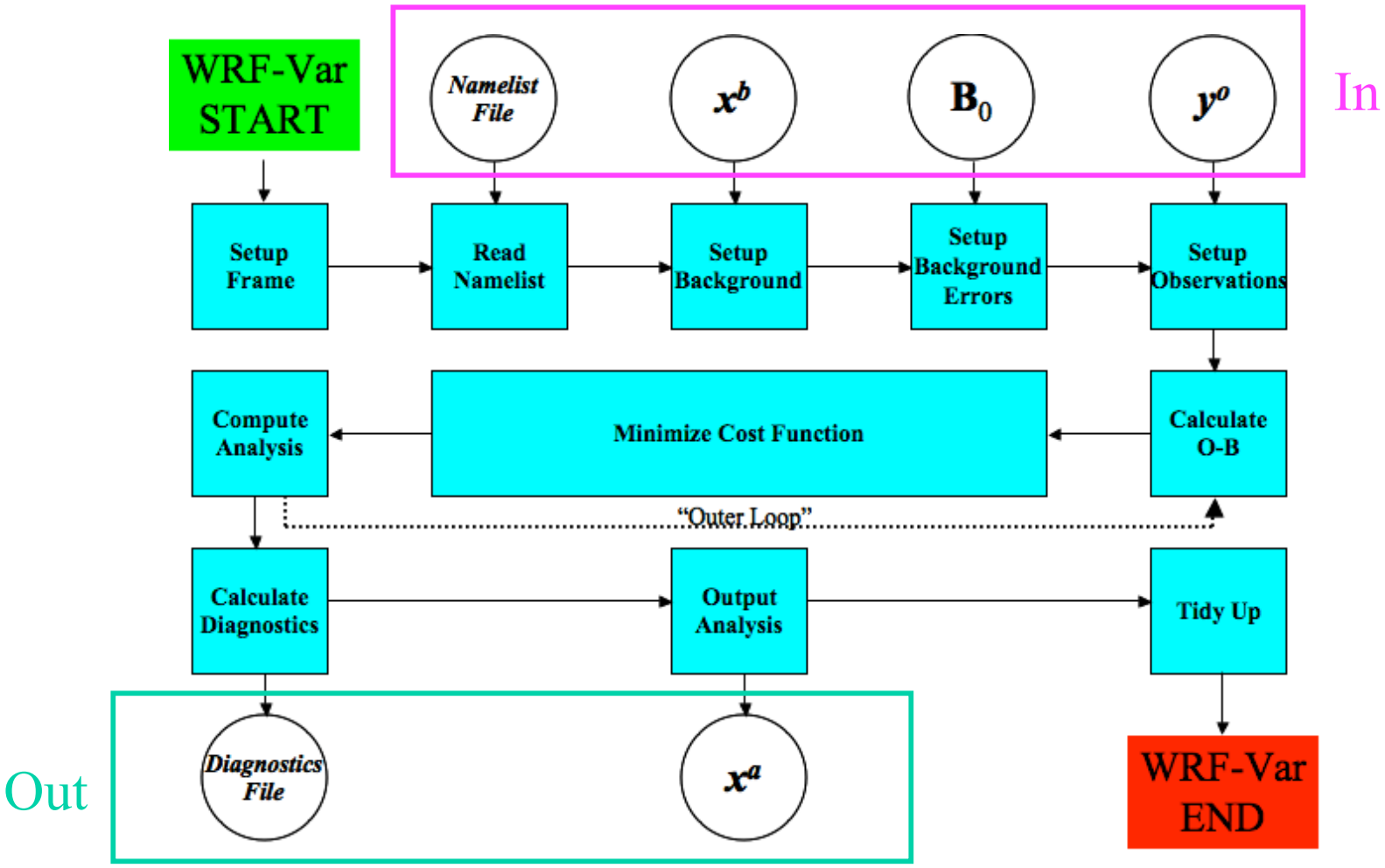
- Clean the WRFDA directory before making your next compilation:

  > clean -a

✓**Note: WRF compiles with -r4 option while WRFDA compiles with -r8.** For this reason, WRF and WRFDA cannot reside and be compiled under the same directory.

Running WRF-Var
3D-Var

• WRF-Var Code Flow



Before You Run …

- Check WRF-Var executable has been created appropriately:
  - WRFDA/var/da/da_wrfvar.exe

- Get input files:
  - The following test data (WRFV3.1-Var-testdata.tar.gz) can be downloaded from http://www.mmm.ucar.edu/wrf/users/download/get_sources.htm.

  - Extract the test data into your local data directory, e.g., "*your_choice_of_dat_dir*".

  - Set up your environment variable $DAT_DIR:
    > Setenv DAT_DIR *your_choice_of_dat_dir*

Before You Run …

- Check input files:
  - Background (x^b): $DAT_DIR/rc/2008020512/wrfinput_d01
    - NETCDF format.
    - For cold-start mode, x^b is generated by WRF *real*.
    - For cycling mode, x^b is generated by WRF from previous cycle (e.g., 6hr forecast).
  - Background Error Statistics: $DAT_DIR/be/be.dat
    - Binary format.
    - Generated by *gen_be* for this specific test case domain.
    - Please refer to "**WRF-Var Background Error Estimations**" talk.
  - Observations (y^o) : $DAT_DIR/ob/2008020512/obs_gts_2008-02-05_12:00:00.3DVAR (GTS data only)
    - ASCII format.
    - Generated by OBSPROC from ob.little_r included in the tar file of the test data.
    - Please refer to "**Radar Data Assimilation**" and "**Radiance Data Assimilation**" talks for assimilations of radar and radiance data.
- Prepare a WRFVAR namelist for runtime options:
  - WRFDA/var/test/tutorial/namelist.input (example)

Working Directory - Input

NCAR

- Create a working directory, for example, "*your_choice_of_ working_dir*".
  > mv *your_choice_of_working_dir*
- Go into the working directory:
  > cd *your_choice_of_working_dir*

- Prepare the input files for running WRF-Var:
  > ln -sf WRFDA/var/da/da_wrfvar.exe **./da_wrfvar.exe**

  > ln -sf WRFDA/run/LANDUSE.TBL **./LANDUSE.TBL**

  > ln -sf $DAT_DIR/rc/2008020512/wrfinput_d01 **./fg**

  > ln -sf $DAT_DIR/be/be.dat **./be.dat**

  > ln -sf $DAT_DIR/ob/obs_gts_2008-02-05_12:00:00.3DVAR **./ob.ascii**

  > cp WRFDA/var/test/namelist.input **./namelist.input**

  (or use your own namelist)

Running WRF-Var

> **da_wrfvar.exe** >&! wrfda.log (or your own log file name)

If running in distributed-memory mode, you need to set up the computer resources (e.g., processor numbers, memory, wallclock…) based on the platform you are using. The log file names would be rsl.out.0000, rsl.out.0001,…

Working Directory - Output

In *your_choice_of_working_dir*, you should at least have the following files after running WRF-Var successfully:

- cost_fn (Cost function)
- grad_fn (Gradient of cost function)
- gts_omb_oma_01 (O, O-A information, etc)
- namelist.output (Complete namelist)
- statistics (Averaged O-B & O-A information)
- **wrfvar_output (Analysis x^a)**

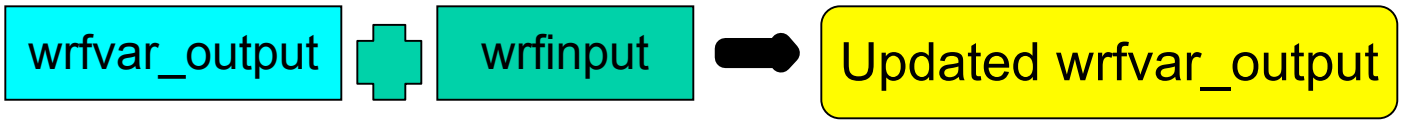
O: Observation
A: Analysis
B: Background (first-guess)

“update_bc” Basic

Input and Output for update_bc

- Update the ***lateral boundary*** condition:

- Update ***the low boundary*** condition:



- Input to update_bc:

  wrfvar_output from **WRF-Var** assimilation

  wrfbdy, wrfinput from **WPS** at analysis time.

- Output from update_bc:

  updated wrfbdy and updated wrfvar_output

  are used to run **WRF** model.

Applications of update_bc

NCAR

- Cold-start run

  only lateral boundary update needed

- Warm-start (cycling) run

  both lateral and low boundaries update needed

- Coarse and fine domains in nested model run

➢ for coarse domain (domain_id = 1),

  both lateral and low boundaries updated

➢ for fine mesh domains (domain_id > 1),

  low boundary updated only

Running update_bc

Steps to Run update_bc

- Check UPDATE_BC executable has been created appropriately:
  - WRFDA/var/da/da_update_bc.exe

- Go into the working directory:
  > cd *your_choice_of_working_dir*

- Prepare the namelist for update_bc: parame.in

```
&control_param
 wrfvar_output_file = './wrfvar_output'      - Analysis generated from WRF-Var
 wrf_bdy_file       = './wrfbdy_d01'         - BC generated from WPS and WRF real
 wrf_input          = './wrfinput_d01'       - IC generated from WPS and WRF real
 domain_id          =1
 cycling = .false.  Set to .true. if WRF-Var first guess comes from a previous WRF forecast.
 debug   = .true.
 low_bdy_only = .false.
 update_lsm = .false.
/
```

Steps to Run update_bc

•Prepare the input files for running WRF-Var:

> ln -sf WRFDA/var/da/da_update_bc.exe **./da_update_bc.exe**

> cp -p $DAT_DIR/rc/2007010200/wrfbdy_d01 **./wrfbdy_d01**

> cp -p $DAT_DIR/rc/2007010200/wrfinput_d01 **./wrfinput_d01**

> cp WRFDA/var/test/param.in **./parame.in** (or use your own file)

And **wrfvar_output** (generated from WRF-Var run)

• Run UPDATE_BC:

> da_update_bc.exe > &! da_update_bc.log

• Check output: **wrfvar_output**, **wrfbdy_d01** (overwrite the original ones!)

WRF-Var Diagnostics

ASCII output files in the WRF-Var working directory:

- wrfda.log or rsl.out.0000
- namelist.output
- filtered_obs (analysis_type="QC-OBS")
- rej_obs_conv_01.000
- qcstat_conv_01
- cost_fn
- grad_fn
- gts_omb_oma_01
- statistics
- jo

wrfda.log

- Very important information about your WRF-Var run, including observation summary, values of cost function and its gradient, etc.

- Additional diagnostics may be printed in these files by including various "print_detail" WRF-Var namelist options (Using these options, the log file size could become really large).

```
***  VARIATIONAL ANALYSIS ***
 DYNAMICS OPTION: Eulerian Mass Coordinate
 WRF NUMBER OF TILES =   1
Set up observations (ob)
```

```
Final:  15 iter, J= 1.76436785D+04, g= 2.06098421D+00
---------------------------------------------------------

Diagnostics
   Final cost function J      =     17643.68
   Total number of obs.       =    26726
   Final value of J           =    17643.67853
   Final value of Jo          =    15284.64894
   Final value of Jb          =     2359.02958
   Final value of Jc          =        0.00000
   Final value of Je          =        0.00000
   Final value of Jp          =        0.00000
   Final J / total num_obs    =        0.66017
   Jb factor used(1)          =        1.00000
   Jb factor used(2)          =        1.00000
   Jb factor used(3)          =        1.00000
   Jb factor used(4)          =        1.00000
   Jb factor used(5)          =        1.00000
   Jb factor used             =        1.00000
   Je factor used             =        1.00000
   VarBC factor used          =        1.00000

 *** WRF-Var completed successfully ***
```

namelist.output

- When WRF-Var is run, a namelist.output file will be produced with all values of namelist variables (default and/or from namelist.input).

namelist.input

```
&wrfvar1
write_increments=true,
var4d=false,
multi_inc=0,
global=false,
/
&wrfvar2
/
&wrfvar3
ob_format=2,
num_fgat_time=1,
/
&wrfvar4
/
```

namelist.output

```
&WRFVAR1
WRITE_INCREMENTS=T, WRFVAR_MEM_MODEL=0, VAR4D=F, MULTI_INC=0,
VAR4D_COUPLING=2, GLOBAL=F, PRINT_DETAI
L_AIREP=F, PRINT_DETAIL_RADAR=F, PRINT_DETAIL_RAD=F,
PRINT_DETAIL_XA=F, PRINT_DETAIL_XB=F, PRINT_DETAI
L_OBS=F, PRINT_DETAIL_F_OBS=F, PRINT_DETAIL_MAP=F,
PRINT_DETAIL_GRAD=F, PRINT_DETAIL_REGRESSION=F, PRI
NT_DETAIL_SPECTRAL=F, PRINT_DETAIL_TESTING=F,
PRINT_DETAIL_PARALLEL=F, PRINT_DETAIL_BE=F,
PRINT_DETAIL_TIMING=F, CHECK_MAX_IV_PRINT=T
 /
 &WRFVAR2
 ANALYSIS_ACCU=900, CALC_W_INCREMENT=F, DT_CLOUD_MODEL=F,
WRITE_QCW=F, WRITE_QRN=F, WRITE_QCI=F, WRITE_QSN=F, WRITE_QGR=F,
WRITE_FILTERED_OBS=F
 /
 &WRFVAR3
 FG_FORMAT=1, OB_FORMAT=2, NUM_FGAT_TIME=1
 /
 &WRFVAR4
 USE_SYNOPOBS=T, USE_SHIPSOBS=T, USE_METAROBS=T,
USE_SOUNDOBS=T, USE_MTGIRSOBS=T, USE_PILOTOBS=T,
```

filtered_obs

NCAR

• Similar to ob.ascii (observation input file of WRF-Var) but with the observations filtered by WRF-Var using the following WRF-Var namelist option:

analysis_type = QC-OBS

ob.ascii

```
TOTAL =  21649, MISS. =-888888.,
SYNOP =   186, METAR =   762, SHIP  =    96, BUOY  =   100, BOGUS =     0, TEMP  =    45,
AMDAR =    43, AIREP =    16, TAMDAR=     0, PILOT =    35, SATEM =   123, SATOB =  12984,
GPSPW =   100, GPSZD =     0, GPSRF =     6, GPSEP =     0, SSMT1 =     0, SSMT2 =     0,
TOVS  =     0, QSCAT =  7125, PROFL =    28, AIRSR =     0, OTHER =     0,
```

filtered_obs

```
TOTAL =  21198, MISS. =-888888.,
SYNOP =   138, METAR =   704, SHIP  =    94, BUOY  =    97, TEMP  =    44, AIREP =    59,
PILOT =    35, GeAMV =  12919, PoAMV =     0, GPSPW =     0, GPSRF =     0, PROFL =     0,
QSCAT =  7108, SSMT1 =     0, SSMT2 =     0, SATEM =     0, BOGUS =     0, AIRSR =     0,
MTGIRS=     0, TAMDAR=     0, OTHER =     0,
```

rej_obs_conv_01.000

- Contains observations that fail check_max_iv check.
  - ✓ 01: outer loop index.
  - ✓ 000: processor id.

- Observations are rejected if the innovation (O-B) values are larger than certain maximum errors (max_error). Currently the max_error values are defined through namelist options max_error_t(uv, pw,…).

Obs_type	Variable	Lat	Lon	Value
synop	U	4.81	-75.73	-8888.88
synop	V	4.81	-75.73	-8888.88
qscat	U	2.20	-79.96	1013.25
qscat	U	2.38	-80.23	1013.25
qscat	U	2.43	-80.01	1013.25
qscat	U	2.44	-79.77	1013.25
qscat	U	2.65	-80.05	1013.25
qscat	U	2.66	-79.83	1013.25
sound	T	4.71	-74.15	107.00
sound	T	4.71	-74.15	31.00
sound	U	2.83	-60.70	925.00
sound	T	2.83	-60.70	93.00
sonde_sfc	Ps	2.83	-60.70	1004.00
sonde_sfc	Q	2.83	-60.70	1004.00

qcstat_conv_01

NCAR

- Contains the number of observations that pass or fail WRF-Var internal QC (e.g, check_max_iv check).
  - ✓ 01: outer loop index.

WRF-Var data utilization statistics for outer iteration 1

obs type	var		ptop / pbot	1000.0 / 1200.0	900.0 / 999.9	800.0 / 899.9	600.0 / 799.0	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
sound	U	used		20	29	37	48	91	41	41	44	45	79	99	574
		rej		1	2	0	0	0	0	0	0	0	0	0	3
sound	V	used		21	30	37	48	91	41	41	44	45	79	99	576
		rej		0	1	0	0	0	0	0	0	0	0	0	1
sound	T	used		32	135	130	452	447	200	118	68	113	191	293	2179
		rej		0	2	0	0	0	0	0	0	1	6	5	14
sound	Q	used		32	135	130	451	439	193	105	53	81	159	218	1996
		rej		0	0	0	0	4	3	0	1	1	4	2	15
synop	U	used		83	0	0	0	0	0	0	0	0	0	0	83
		rej		1	0	0	0	0	0	0	0	0	0	0	1
synop	V	used		83	0	0	0	0	0	0	0	0	0	0	83
		rej		1	0	0	0	0	0	0	0	0	0	0	1
synop	T	used		137	0	0	0	0	0	0	0	0	0	0	137
		rej		0	0	0	0	0	0	0	0	0	0	0	0
synop	Q	used		130	0	0	0	0	0	0	0	0	0	0	130
		rej		4	0	0	0	0	0	0	0	0	0	0	4

cost_fn and grad_fn

- Contain values of cost function and its gradient at each iteration.
  - If print_detail_grad=.false., only the initial and final values of the cost and gradient functions are output as follows:

cost_fn

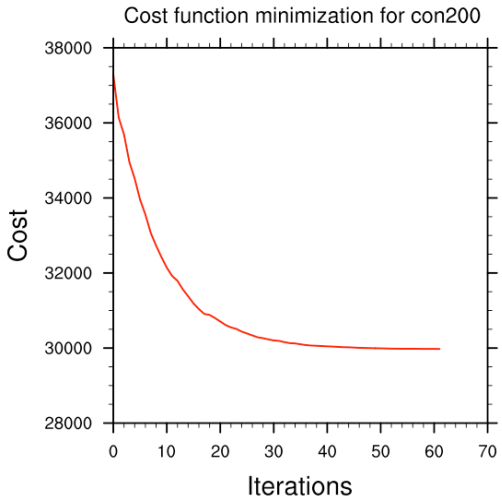
Outer Iter	EPS	Inner Iter	J	Jb	Jo	Jc	Je	Jp	Js
1	0.100E-01	0	24322.148	0.000	24322.148	0.000	0.000	0.000	0.000
1	0.100E-01	21	16141.945	1847.293	14294.652	0.000	0.000	0.000	0.000

grad_fn

Outer Iter	EPS	Inner Iter	G	Gb	Go	Ge	Gp	Gs
1	0.100E-01	0	543.846	0.000	543.846	0.000	0.000	0.000
1	0.100E-01	21	4.767	60.783	60.970	0.000	0.000	0.000

b: background term
o: observation term
c: JcDFI term
e: alpha term
p: radiance variational bias correction term
s: skin temperature or cloud cover term

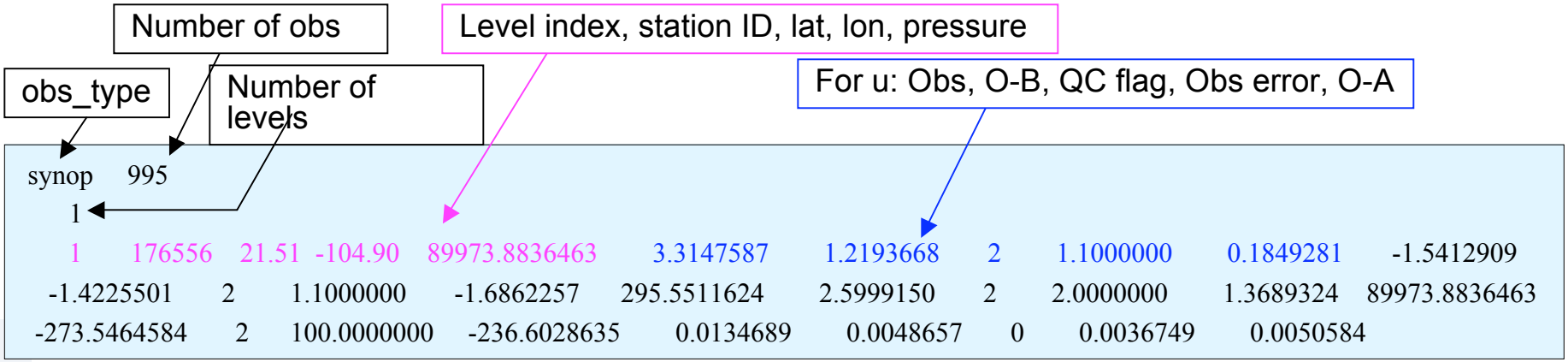
  - If print_detail_grad=.true., the cost function and its gradient at each iteration will be computed and written into **cost_fn** and **grad_fn**.



✓ WRF-Var tools: plot_cost_grad_fn.ncl

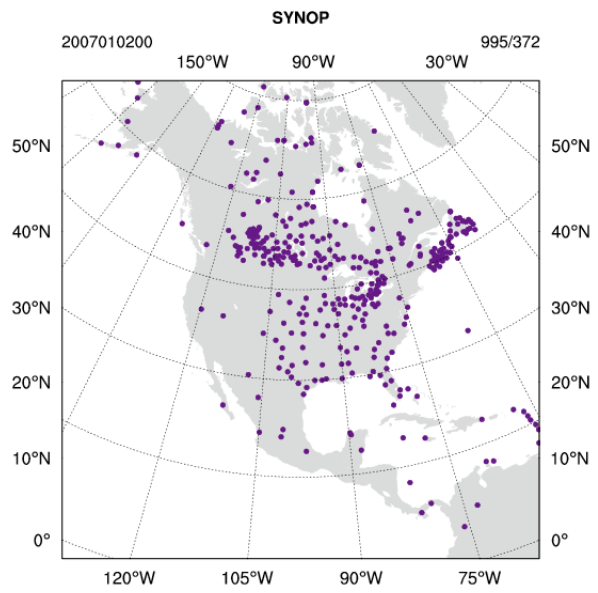
gts_omb_oma_01

- Contain complete point-by-point, detailed observation information.



- Measured quantities for each observation type vary:

Synop: u, v, t, p, q	Polaramv: u, v
Metar: u, v, t, p, q	Gpspw: tpw
Ship: u, v, t, p, q	Sound: u, v, t, q
Geoamv: u, v	Sonde_sfc: u, v, t, p, q
Airep: u, v, t	Profiler: u, v
Pilot: u, v	Buoy: u, v, t, p, q
Satem: thickness	Airsr: t, q
Qscat: u, v	Gpsref: ref



✓ WRF-Var tools: plot_gts_omb_oma.ncl

statistics

- Contains domain-wise O-B and O-A information:

Diagnostics of OI for synop

var	u (m/s)	n	k	v (m/s)	n	k	t (K)	n	k	p (Pa)	n	k	q (kg/kg)	n	k
Number:	331			332			355			330			361		
Minimum(n,k):	-5.4017	363	0	-5.4086	878	0	-9.7206	592	0	-390.7893	931	0	-0.4461E-02	719	0
Maximum(n,k):	5.0466	886	0	5.2878	630	0	7.7302	421	0	471.9343	944	0	0.5408E-02	787	0
Average :	-0.8471			-0.1995			-1.1171			20.4177			-0.2525E-03		
RMSE :	2.3023			2.1150			3.1978			116.1518			0.8045E-03		

Diagnostics of AO for synop

var	u (m/s)	n	k	v (m/s)	n	k	t (K)	n	k	p (Pa)	n	k	q (kg/kg)	n	k
Number:	331			332			355			330			361		
Minimum(n,k):	-4.2496	172	0	-5.0463	683	0	-8.9005	583	0	-472.9290	931	0	-0.4152E-02	719	0
Maximum(n,k):	5.5540	886	0	5.7990	630	0	8.8192	421	0	392.4096	944	0	0.5058E-02	1	0
Average :	-0.0847			-0.0376			-0.4283			1.1709			0.1625E-04		
RMSE :	1.8650			1.8093			2.1990			101.3816			0.5958E-03		

Minimum of gridded analysis increments

Lvl	u	i	j	v	i	j	t	i	j	p	i	j	q	i	j
1	-1.8915	17	32	-1.9965	36	24	-5.2526	20	35	-314.7470	44	1	-0.1451E-02	18	32
2	-1.9476	16	32	-2.0070	36	24	-3.0142	21	36	-311.2885	44	1	-0.1438E-02	18	33

Maximum of gridded analysis increments

Lvl	u	i	j	v	i	j	t	i	j	p	i	j	q	i	j
1	1.3750	41	8	1.5739	28	12	3.2994	24	20	197.8351	28	2	0.1401E-02	39	8
2	1.4844	40	8	1.6180	28	13	1.7471	7	20	195.5165	28	2	0.1591E-02	39	8

Mean of gridded analysis increments

Lvl	u	v	t	p	q
1	-0.0327	0.0632	-0.1477	17.4414	-0.1047E-03
2	-0.0031	0.0736	0.0116	17.2543	-0.8066E-04

RMSE of gridded analysis increments

Lvl	u	v	t	p	q
1	0.7546	0.6040	1.3120	72.0441	0.4258E-03
2	0.7995	0.6483	0.9169	71.2614	0.4476E-03

jo

NCAR

• Contains cost function for each observation type:

synop	obs,	Jo(actual)	=	1007	1709	475.29555	1.00000	448.89633	1.00000	214.58090	1.00000	169.59091	1.00000	39.54654	1.00000
metar	obs,	Jo(actual)	=	2551	4996	1142.22791	1.00000	1139.04835	1.00000	450.85222	1.00000	141.48881	1.00000	127.23786	1.00000
ships	obs,	Jo(actual)	=	270	739	295.61942	1.00000	328.81980	1.00000	38.63147	1.00000	76.05158	1.00000	10.88285	1.00000
geoamv	ob,	Jo(actual)	=	18216	35619	4375.80943	1.00000	4291.11244	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000
gpspw	obs,	Jo(actual)	=	113	94	42.19891	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000
sound	obs,	Jo(actual)	=	122	12507	1501.01081	1.00000	1417.89485	1.00000	2934.71994	1.00000	1412.34202	1.00000	0.00000	1.00000
sonde	obs,	Jo(actual)	=	122	12507	77.96908	1.00000	70.37029	1.00000	43.28542	1.00000	45.34806	1.00000	4.58217	1.00000
airep	obs,	Jo(actual)	=	1527	4506	699.19993	1.00000	655.45784	1.00000	776.57509	1.00000	0.00000	1.00000	0.00000	1.00000
pilot	obs,	Jo(actual)	=	112	5895	2582.21854	1.00000	2434.46137	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000
satem	obs,	Jo(actual)	=	204	2079	108.15758	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000	0.00000	1.00000
buoy	obs,	Jo(actual)	=	241	400	133.21166	1.00000	104.72975	1.00000	31.86149	1.00000	38.47701	1.00000	1.04651	1.00000

• Sum of individual Jo (numbers in red boxes) equals the printout value in WRF-Var log file, e.g., rsl.out.0000:

Final value of Jo = 28880.81069

• Numbers in blue boxes are observation error factors used in WRF-Var:

Tuned obs_error = obs_error * factor

Where obs_error values are assigned by OBSPROC and factor=1 by default (use_obs_errfac=false).

WRF-Var Running Options - Namelist

- ✓ The namelist mentioned in the following slides refer to 3D-Var runs and conventional data assimilation only. Please refer to specific lectures (BE, radiance, …) for other namelist options .

What is a Namelist?

- The Fortran namelist (namelist.input) file helps the user to configure a WRF-Var run without recompiling the code.
  - Specific Fortran 90 namelist format

```
&namelistname        - start
…
/                    - end
```

- Description of WRF-Var namelist variables are given in WRF User's Guide and README.namelist in the release tar file (WRFDA/var/README.namelist).

WRF-Var Namelist

- Default values of the namelist variables are defined by WRF-Var Registry (WRFDA/Registry/Registry.wrfvar).
- Define namelist.input with non-default and desired variable values before running WRF-Var.
- A WRF-Var namelist file includes two parts:

```
&wrfvar1
/
&wrfvar2
/
…
&wrfvar23
/
&time_control
/
&dfi_control
/
…
&namelist_quilt
/
```

WRF-Var namelist options:
Running options for WRF-Var code.

WRF namelist options:
WRF-Var needs certain information from this file including domain and time setting. Please make sure this part of the namelist file is consistent with the namelist used in your WRF *real* and WRF runs.

Namelist - WRFVAR1

- Write_increments
  - .false. : Default
  - .true. : Output analysis increment file "analysis_increments" (analysis-background). The file is a binary file, generated every time you run WRF-Var by using a FORTRAN code given in

```
-rw-r--r--    1 huishao  ncar    43271476 Jul  7 16:27 analysis_increments
lrwxrwxrwx    1 huishao  ncar          32 Jul  7 16:27 be.dat -> /ptmp/huishao/tutorial/be/be.dat
-rw-r--r--    1 huishao  ncar        1600 Jul  7 16:27 check_max_iv
-rw-r--r--    1 huishao  ncar         313 Jul  7 16:27 cost_fn
```

  - ✓ You could still produce your own analysis increment file by extracting first guess from analysis files (both in netcdf format). The advantage of using this "analysis_increment" is to avoid spurious increments (because it is generated directly from the code without including the first guess).

Namelist - WRFVAR1

- Print_detail_grad
  - .false. : Default

Outer Iter	EPS	Inner Iter	J	Jb	Jo	Jc	Je	Jp
1	0.100E-01	0	11251.182	0.000	11251.182	0.000	0.000	0.000
1	0.100E-01	19	8634.570	885.427	7749.143	0.000	0.000	0.000

  - .true. : Output cost function gradient values for each observation type at each iteration into standard output files and cost function and gradient values at each iteration into the files called "cost_fn" and "grad_fn".

Outer Iter	EPS	Inner Iter	J	Jb	Jo	Jc	Je	Jp
1	0.100E-01	0	11251.182	0.000	11251.182	0.000	0.000	0.000
1	0.100E-01	1	10384.156	41.768	10342.388	0.000	0.000	0.000
1	0.100E-01	2	9633.557	184.109	9449.448	0.000	0.000	0.000
1	0.100E-01	3	9245.700	327.121	8918.579	0.000	0.000	0.000
1	0.100E-01	4	9014.861	453.787	8561.075	0.000	0.000	0.000
1	0.100E-01	5	8872.989	559.714	8313.275	0.000	0.000	0.000
1	0.100E-01	6	8777.974	652.105	8125.869	0.000	0.000	0.000
1	0.100E-01	7	8720.998	721.735	7999.263	0.000	0.000	0.000
1	0.100E-01	8	8689.342	768.464	7920.878	0.000	0.000	0.000
1	0.100E-01	9	8665.605	810.136	7855.469	0.000	0.000	0.000
1	0.100E-01	10	8654.051	833.590	7820.461	0.000	0.000	0.000
1	0.100E-01	11	8646.376	851.091	7795.285	0.000	0.000	0.000
1	0.100E-01	12	8641.869	862.515	7779.355	0.000	0.000	0.000
1	0.100E-01	13	8638.219	872.853	7765.365	0.000	0.000	0.000
1	0.100E-01	14	8636.669	877.707	7758.962	0.000	0.000	0.000
1	0.100E-01	15	8635.794	880.667	7755.127	0.000	0.000	0.000
1	0.100E-01	16	8635.176	882.929	7752.247	0.000	0.000	0.000
1	0.100E-01	17	8634.861	884.169	7750.693	0.000	0.000	0.000
1	0.100E-01	18	8634.686	884.909	7749.777	0.000	0.000	0.000
1	0.100E-01	19	8634.570	885.427	7749.143	0.000	0.000	0.000

Namelist - WRFVAR2

- Analysis_accu
  - 900 (Sec): Default.

  If |analysis time - first guess time| > Analysis_accu, WRF-Var will give a warning like "*Wrong xb time found???*".

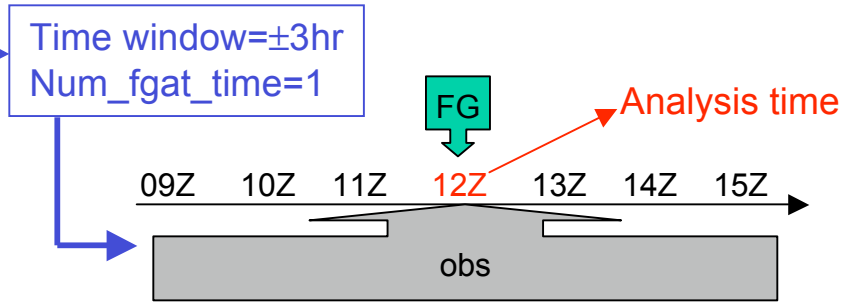
Namelist - WRFVAR3

- Fg_format: Format of the first guess of WRF-Var.
  - 1 = ARW regional: Default
  - 2 = WRF-NMM regional
  - 3 = ARW global
  - 4 = KMA global

- Ob_format: The format of the conventional and satellite retrieval observation data going into WRF-Var.
  - 1 = NCEP PREPBUFR (ob.bufr).
  - 2 = ASCII (ob.ascii): Default.
  - 3 = MADIS data format.

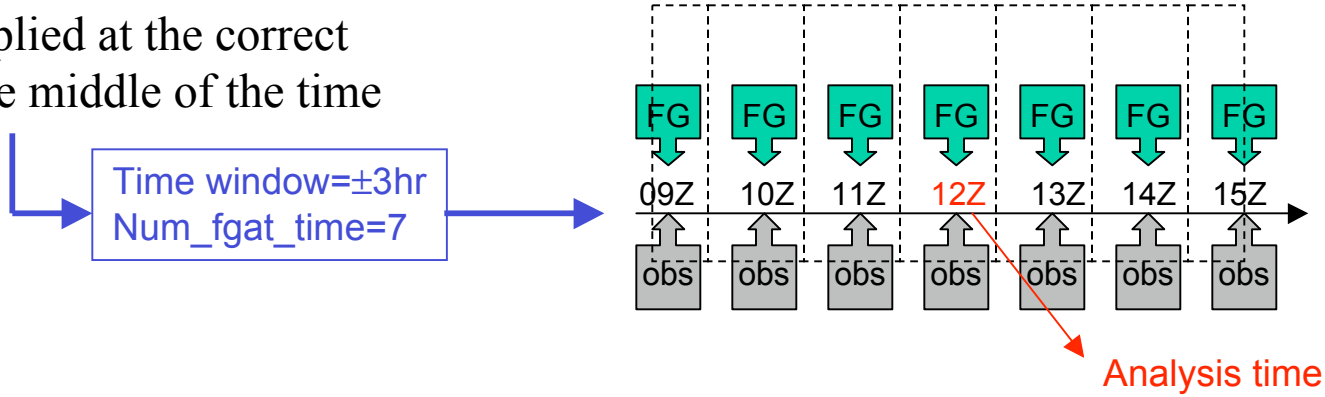
Namelist - WRFVAR3

- Num_fgat_time: Number of data time windows (slots) used in WRF-Var.
  - 1 = 3DVAR: Default.
  - >1 apply to FGAT and 4D-Var (for ob_format=2 and radiance only).



✓ **First-Guess at Appropriate Time (FGAT):**

An option in WRF-3DVar that allows the observations to be applied at the correct time, rather than at the middle of the time window.



Namelist - WRFVAR4

- thin_conv
  - .true.: Default. Mandatory for ob_format=1 (NCEP PREPBUFR) to avoid time duplication.
  - .false. Used only for debugging purpose.

- thin_mesh_conv (max_instruments): Thinning mesh for each type of conventional observations. This option is used for ob_format=1 (NCEP PREPBUFR) only. The observation index/order follows the definition in WRFDA/var/da/da_control/da_control.f90 (e.g., sound =1, synop =2, …)
  - 20.0 (km): Default.

Namelist - WRFVAR4

- Use_obstype: Set to true to use particular observation types.
  - E.g, use_gpsrefobs=.true.: Assimilate GPS refractivity observations if any available in the data file.

- Use_obs_errfac: Option for using tuned observation error.
  - .false. : Default.
  - .true. : Use tuned observation error statistics (need to produce errfac.dat beforehand created by da_tune_obs_desrozier.f).

Namelist - WRFVAR5

- Check_max_iv: Option for WRF-Var internal QC procedure, which is basically a maximum observation error check based on the innovations (Obs-Background).
  - .true. : default
  - .false: Use this option only if the observation data have been cleaned before going into WRF-Var.

- Max_error_t (uv, pw, ref, rh, …): maximum error factor allowed in check_max_iv check for t (u/v, pw, ref, rh, …).

Namelist - WRFVAR5

- Max_obstype_input: Set to restrict the maximum number of observations used in each type.
  - The restriction is applied when the observations are read, and applies to the total number of observations across all processors before quality control.
  - E.g., max_sound_input=5000: the maximum number of radiosondes is 5000.

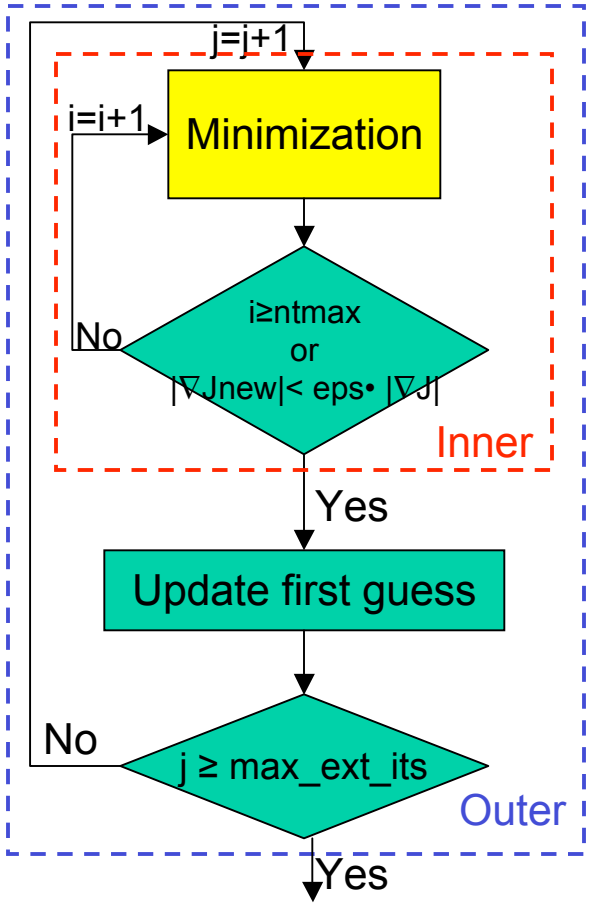
Namelist - WRFVAR6

The following namelist variables are for minimization options:

- Max_ext_its: Number of outer loops.
  - 1: Default. Only one outer loop.
  - Currently, maximum outer loop number is 10.

- Ntmax: Maximum number of iterations in an inner loop for the minimization in WRF-Var.
  - 200: Default. The minimization in the inner loop can not exceed 200.

- Eps: Value for minimization convergence criterion. It is an array with the dimension=max_ext_its.
  - 0.01(max_ext_its): The minimization is considered to converge when the norm of the cost function gradient is reduced at least 2 orders.



Namelist - WRFVAR9

The namelist variables trace_* are for tracing:

Tracing gives additional diagnostics about program runs. It does not change results, but does slow the program down, so should be disabled in production environments.

- Trace_use: .false. (default). Use tracing function in WRF-Var if true.

Trace Output

* Calling Tree
* Local routine timings
* Overall routine timings
* Memory usage

Maximum memory usage for routines

Routine	Max in any PE (kbytes)	Overall (kbytes)	Average per PE (kbytes)
da_transfer_xatoanalysis	508076	15803513	493859
da_transfer_xatowrf	508076	15803513	493859
da_write_increments	508076	15803513	493859
da_deallocate_observations	506698	15761784	492555
da_deallocate_y	506392	15756018	492375

Namelist - WRFVAR11

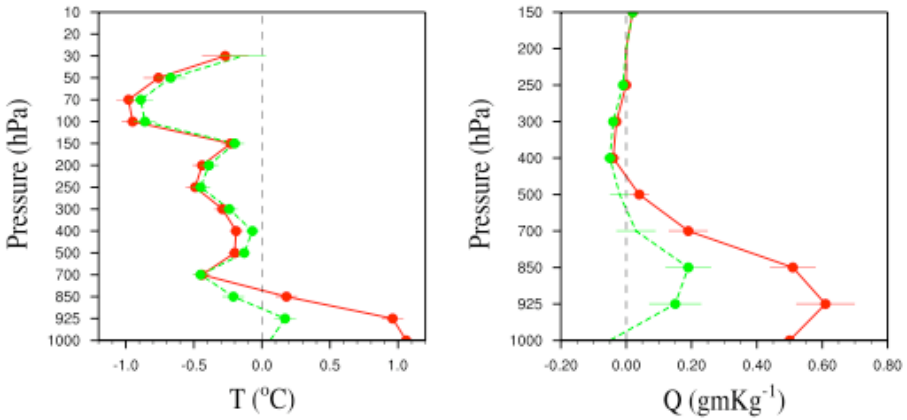
- Cv_options_hum:
  - 1 (default): Please do not change.

- Check_rh:
  - 0 : No supersaturation check after minimization.
  - 1: With the supersaturation (rh> 100%) and minimum rh (rh<10%) check, and make the local adjustment of q.
  - 2 (default): With the supersaturation (rh> 95%) and minimum rh (rh<11%) check, and make the multi-level q adjustment under the constraint of integrated water vapor in column conserved.

Namelist - WRFVAR11

- Sfc_assi_options:
  - 1 (default): The surface observations will be assimilated based on the lowest model level first guess. Observations are not used when the height difference of the elevation of observing site and the lowest model level height is larger than 100m.
  - 2: The surface observations will be assimilated based on surface similarity theory in PBL. Innovations are computed based on 10-m wind and 2-m temperature & moisture.

✓Please use this option with caution, since the results could be very sensitive.



Namelist - WRFVAR11

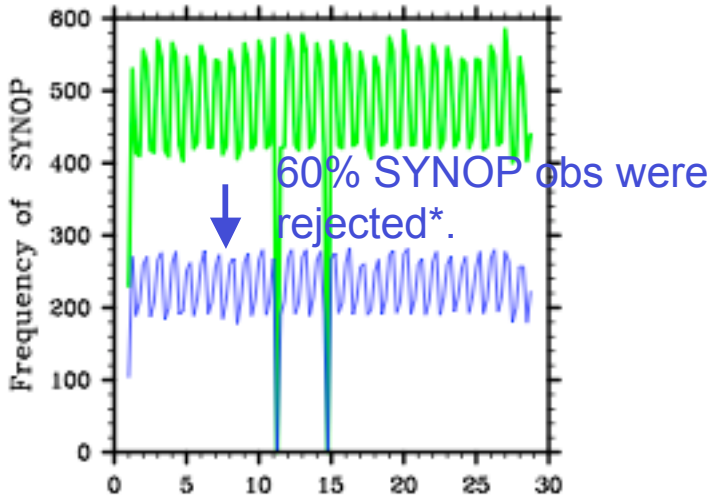
- Calculate_cg_cost_fn:
  - .false. : Only the initial and final cost functions are computed and output.
  - .true. : The cost functions are computed and output into standard output files at every iteration for diagnostic purpose.

  ✓ The conjugate gradient algorithm for the minimization does not require the computation of cost function at every iteration.

  ✓ Set print_detail_grad=.true. to output cost function and gradient values at every iteration into “cost_fn” and “grad_fn”.

Namelist - WRFVAR17

- Analysis_type: Indicate job type of WRF-Var.
  - 3D-VAR (default): Run 3D-Var data assimilation.
  - RANDOMCV: Create ensemble perturbations.
  - VERIFY: Run WRF-Var verification mode (then Check_max_iv=.false. and ntmax=0 by default).
    - ✓ Please refer to "**WRF-Var Tools and Verification package**" talk.
  - QC-OBS: Run 3D-Var data assimilation and produce filtered_obs.
    - ✓ By combining with Check_max_iv=.true. and ntmax=0, you can produce a WRF-Var filtered (QCed) observation data set (filtered_obs) without running the data assimilation.
      - 1st screen/QC procedure performed by observation preprocessor (OBSPROC).
      - 2nd screen/QC procedure performed in WRF-Var.
      - Main impact of 2nd screen/QC is on surface observations*.
      - Rejection rates will reduce with higher resolution, higher-order interpolation.



* Surface observation rejection here is mostly due to surface elevation check with sfc_assi_options=1. Such a rejection may be bypassed by using sfc_assi_options=2.

Namelist - WRFVAR18

- Analysis_date: Specify the analysis time. It should be consistent with the first guess time.

  ✓ If time difference between analysis_date and date info read in from first guess is larger than analysis_accu, WRF-Var will issue a warning message "Wrong xb time found???", but won't abort.

Namelist - WRFVAR21/22

- Time_window_min(max): Specify the lower (upper) time values of the assimilation time window.

NCAR

Thank you!

• UPDATE_BC Code Flow