

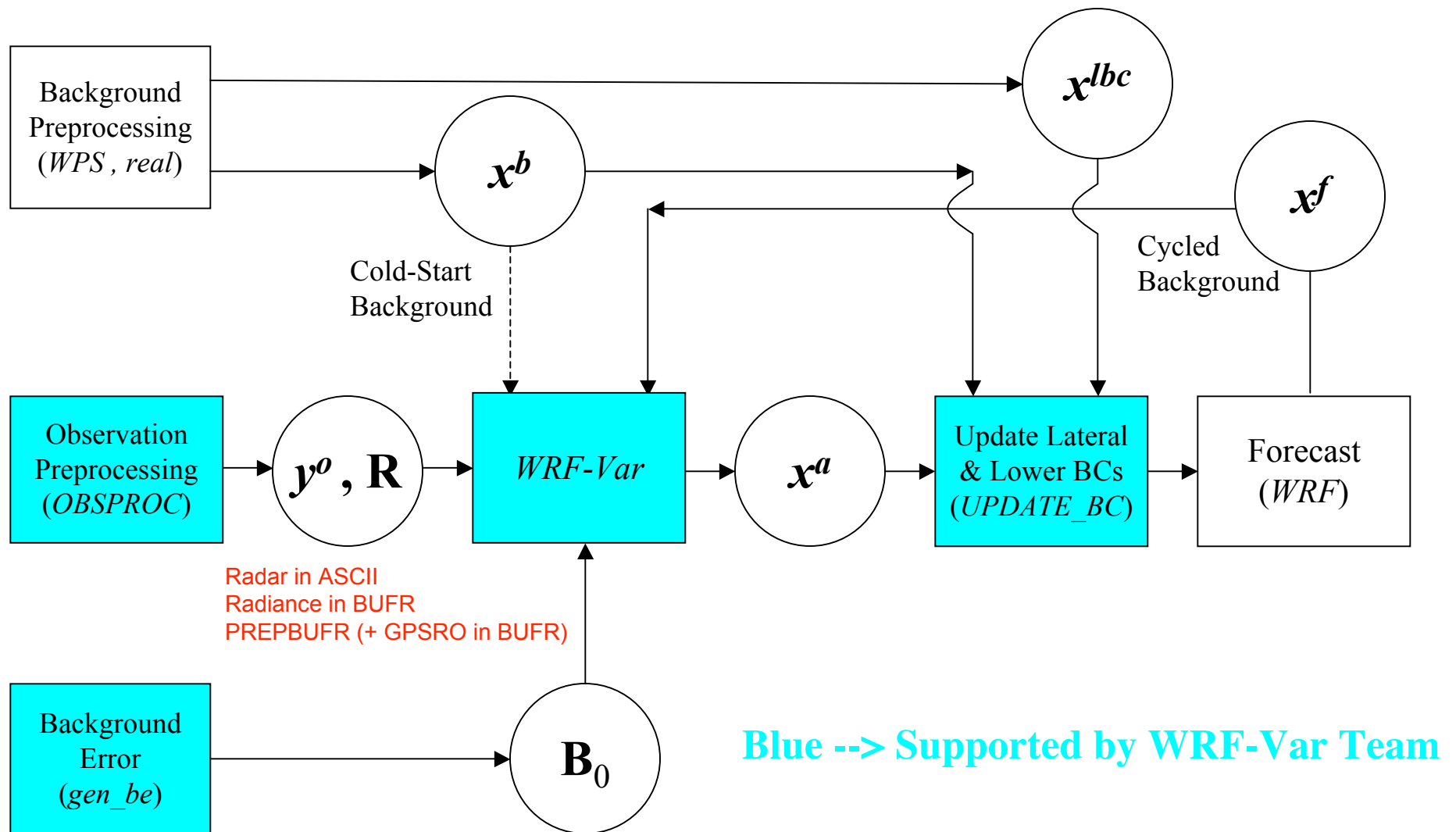
WRF-Var Setup, Run and Diagnostics

Hui-Chuan Lin

Hui Shao, Meral Demirtas, Yong-Run Guo, Syed Rizvi,
Xin Zhang and Hans Huang

WRFDA Tutorial, February 01-03, 2010

- **WRF-Var in the WRF Modeling System**



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 WRFDA V3.1.1.

Installing WRF-Var 3D-Var

Source Codes

- Download WRF-Var source code (**WRFDV3.1.1.TAR.gz**) from
http://www.mmm.ucar.edu/wrf/users/wrfda/download/get_source.html

After following the instruction from the above URL, on the actual download page, look for

WRF-Var Code and Test Data V3.1.1 (Released July 31, 2009)

[WRF-DA source code tar file](#) (Version 3.1.1) - [Details](#).

WRF-Var Code and Test Data V3.1 (Released April 9, 2009)

[WRF-DA source code tar file](#) (Version 3.1) - [Details](#).

[WRF DA test data](#) for V3.

To learn more about WRF-Var and how to use it, please visit the [WRF-DA home page](#).

FORTRAN90 Compiler

Supported and tested platforms

- IBM with XLF
- Powerpc Mac with G95 (GCC 4.0.3 (g95 0.91!) Sep 12 2007)
- Intel Mac with PGI and G95 (GCC 4.0.3 (g95 0.91!) Sep 12 2007)
- Linux i686 with PGI, INTEL and GFORTRAN (gcc version 4.3.3 20090123)
- ✓ to check version numbers, use xlf -qversion, g95 -v, pgf90 -V, gfortran -v
- ✓ IBM xlf 11.01.0000.0002, the da_minimization.f might stuck with the -O2 optimization, -O0 can get it through.

Supported but tested by a 3rd party

- SGI Altix with ifort
- cray with ftn 6.0.0.2.1

Unsupported platforms

Due to the lack of access to these platforms and compilers, we are unable to provide support.

- Sun (solaris)
- DEC alpha
- Other SGI
- Absoft
- Lahey
- Salford
- Pathscale

Does not work on

- PGI 5.x
- Windows

Please share with us (and other fellow users) your experience on the various platforms and compilers.

Libraries Required by WRF-Var

- NetCDF: Network Common Data Form
 - <http://www.unidata.ucar.edu/software/netcdf/>
 - BLAS: Basic Linear Algebra Subprograms
 - LAPACK: Linear Algebra PACKage
- } Included in WRFDA V3.1.1:
WRFDA_V3.1.1/var/external
- <http://netlib.org/blas/>
- <http://netlib.org/lapack/>

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

- BUFR: Included in WRFDA V3.1.1 (WRFDA_V3.1.1/var/external).
- CRTM: Community Radiative Transfer Model (version REL_1.2)
Required only if assimilating radiance data.
 - <ftp://ftp.emc.ncep.noaa.gov/jcsda/CRTM/>
- RTTOV: Radiative Transfer for TOVS (version 8.7)
Required only if assimilating 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_V3.1.1
> ./configure wrfda
    – configure.wrf will be created.
```

```
checking for perl5... no
checking for perl... found /usr/bin/perl (perl)
Will use NETCDF in dir: /usr/local/netcdf-3.6.0-p1-g95ppc
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 (smpar)
3. Darwin (MACOS) PGI compiler with pgcc (dmpar)
4. Darwin (MACOS) PGI compiler with pgcc (dm+sm)
5. Darwin (MACOS) intel compiler with icc (serial)
6. Darwin (MACOS) intel compiler with icc (smpar)
7. Darwin (MACOS) intel compiler with icc (dmpar)
8. Darwin (MACOS) intel compiler with icc (dm+sm)
9. Darwin (MACOS) intel compiler with cc (serial)
10. Darwin (MACOS) intel compiler with cc (smpar)
11. Darwin (MACOS) intel compiler with cc (dmpar)
12. Darwin (MACOS) intel compiler with cc (dm+sm)
13. Darwin (MACOS) g95 with gcc (serial)
14. Darwin (MACOS) g95 with gcc (dmpar)
15. Darwin (MACOS) xlf (serial)
16. Darwin (MACOS) xlf (dmpar)

serial: single-processor

dmpar: distributed-memory parallel

smpar: shared-memory parallel

dm+sm: distributed-memory with shared-memory parallel

Enter selection [1-16] :

✓ On linux, dm+sm may not work properly because the MPI library may not be thread-safe

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**

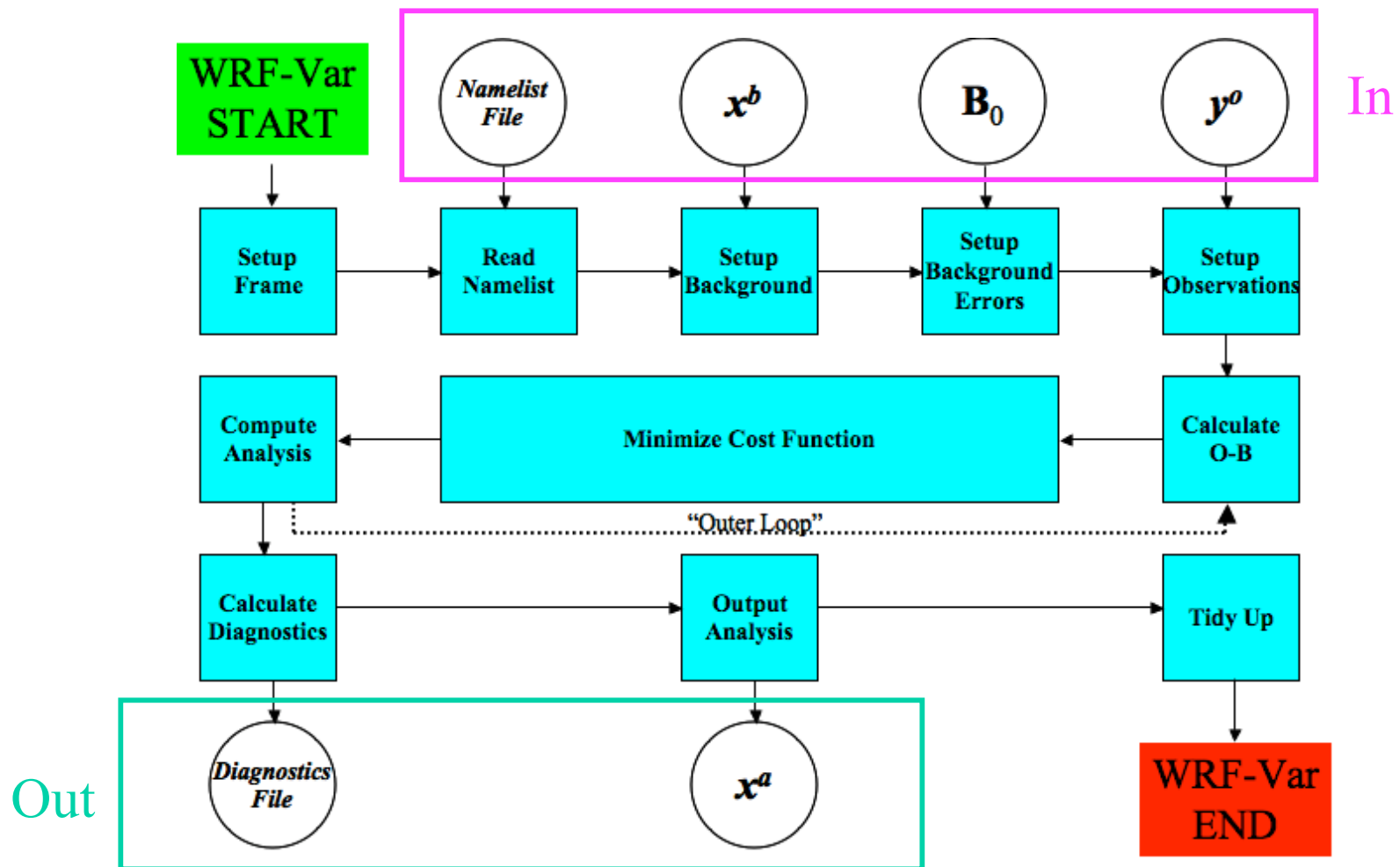
- Clean the WRFDA directory before making your next compilation:

> **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 ...

- Make sure WRF-Var executable has been created successfully:
 - `WRFDA_V3.1.1/var/build/da_wrfvar.exe`
- Get input files:
 - The test data (WRFV3.1-Var-testdata.tar.gz) can be downloaded from <http://www.mmm.ucar.edu/wrf/users/wrfda/download/testdata.html>
 - 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/ob.ascii (GTS data only)
 - ASCII format.
 - Generated by OBSPROC from obs.2008020512.gz 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_V3.1.1/var/test/tutorial/namelist.input (example)

Working Directory - Input

- Create a working directory, for example, “*your_choice_of_working_dir*”.
 `> mkdir your_choice_of_working_dir`
- Go into the working directory:
 `> cd your_choice_of_working_dir`
- Prepare the input files (link or copy) for running WRF-Var:
 `> ln -sf WRFDA_V3.1.1/var/build/da_wrfvar.exe ./da_wrfvar.exe`
 `> ln -sf WRFDA_V3.1.1/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/2008020512/ob.asci ./ob.asci`
 `> cp WRFDA_V3.1.1/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 will 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 WRF-Var is successfully completed:

- **cost_fn** (Cost function)
- **grad_fn** (Gradient of cost function)
- **gts_omb_oma_01** (point-by-point O, O-B, O-A information, etc)
- **namelist.output** (Complete namelist)
- **statistics** (domain-wise O-B and O-A statistics)
- **wrfvar_output** (**Analysis x^a** , the input to WRF model)

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

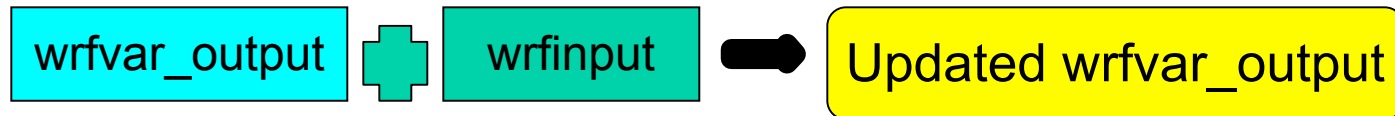
“update_bc” Basic

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 the **WRF** model.

Applications of update_bc

- Cold-start run
only lateral boundary update needed
- Warm-start (cycling) run
both lateral and low boundaries updates needed
- Coarse and fine domains in nested model run
 - for coarse domain ($\text{domain_id} = 1$),
both lateral and low boundaries updated
 - for fine mesh domains ($\text{domain_id} > 1$),
low boundary updated only

Running update_bc

Steps to Run update_bc

- Make sure UPDATE_BC executable has been created successfully:
 - WRFDA_V3.1.1/var/build/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'
wrf_bdy_file       = './wrfbdy_d01'
wrf_input          = './wrfinput_d01'
domain_id         = 1
cycling = .false.
debug   = .true.
low_bdy_only = .false.
update_lsm = .false.
/
```

 - Analysis generated from WRF-Var
 - BC generated from WPS and WRF real
 - IC generated from WPS and WRF real

Set to .true. if WRF-Var first guess comes from a previous WRF forecast.

Steps to Run update_bc

- Prepare the input files for running update_bc:

```
> ln -sf WRFDA_V3.1.1/var/build/da_update_bc.exe
```

```
./da_update_bc.exe
```

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

```
> ln -sf $DAT_DIR/rc/2007010200/wrfinput_d01 ./wrfinput_d01
```

```
> cp WRFDA_V3.1.1/var/test/update_bc/parame.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** (the original ones are overwritten!)

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 (rsl.out.0000)

- 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_xxx” 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
print_detail_grad=.true.
/
&wrfvar2
/
&wrfvar3
ob_format=2,
num_fgat_time=1,
/
&wrfvar4
use_synopobs=.false.
/
```

namelist.output

```
&WRFVAR1
WRITE_INCREMENTS = F, WRFVAR_MEM_MODEL = 0, VAR4D = F,
MULTI_INC = 0, VAR4D_COUPLING = 2, PRINT_DETAIL_RADAR = F,
PRINT_DETAIL_RAD = F, PRINT_DETAIL_XA = F, PRINT_DETAIL_XB = F,
PRINT_DETAIL_OBS = F, PRINT_DETAIL_F_OBS = F, PRINT_DETAIL_MAP = F,
PRINT_DETAIL_GRAD = T, PRINT_DETAIL_REGRESSION = F,
PRINT_DETAIL_SPECTRAL = F,
PRINT_DETAIL_TESTING = F, PRINT_DETAIL_PARALLEL = F,
PRINT_DETAIL_BE = F,
CHECK_MAX_IV_PRINT = T, CHECK_BUDDY_PRINT = F,
/
&WRFVAR2
ANALYSIS_ACCU = 900, CALC_W_INCREMENT = F, DT_CLOUD_MODEL = F,
WRITE_MOD_FILTERED_OBS = F,
/
&WRFVAR3
FG_FORMAT=1, OB_FORMAT=2, NUM_FGAT_TIME=1
/
&WRFVAR4
USE_SYNOPOBS=F, USE_SHIPSOBS=T, USE_METAROBS=T, USE_SOUNDOBS=T,
USE_MTGIRSOBS=T, USE_PILOTOBS=T,
```

filtered_obs

- Similar to ob.ascii (observation input file of WRF-Var) but it contains 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 values defined as a multiple of the observation error for each observation (obs_error*factor). The default maximum value is 5 times the observation error. the factor of 5 can be changed through max_error_* namelist settings.

Obs_type	Variable	Lat	Lon	Pressure
sound	T	50.68	-127.36	215.00
sound	Q	50.68	-127.36	215.00
sound	U	47.46	-111.38	850.00
sound	V	31.86	-106.70	400.00
synop	U	50.11	-127.93	991.10
synop	V	48.76	-123.11	994.50
synop	Ps	53.43	-114.71	1013.01
synop	Q	53.43	-114.71	1013.01
gpsref	GpsR	36.26	-71.36	53.34
qscat	V	23.20	-74.22	1013.25

qcstat_conv_01

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

[illegible]

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

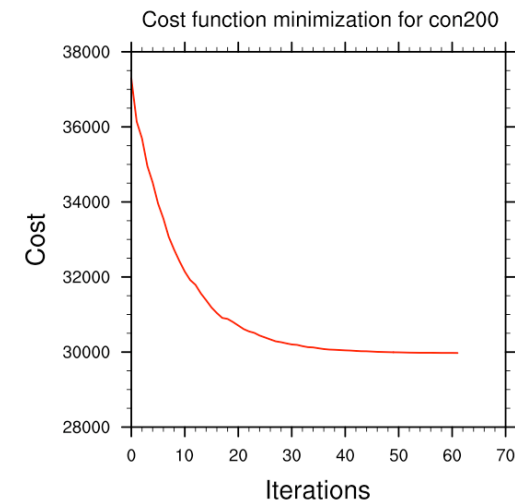
grad_fn

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

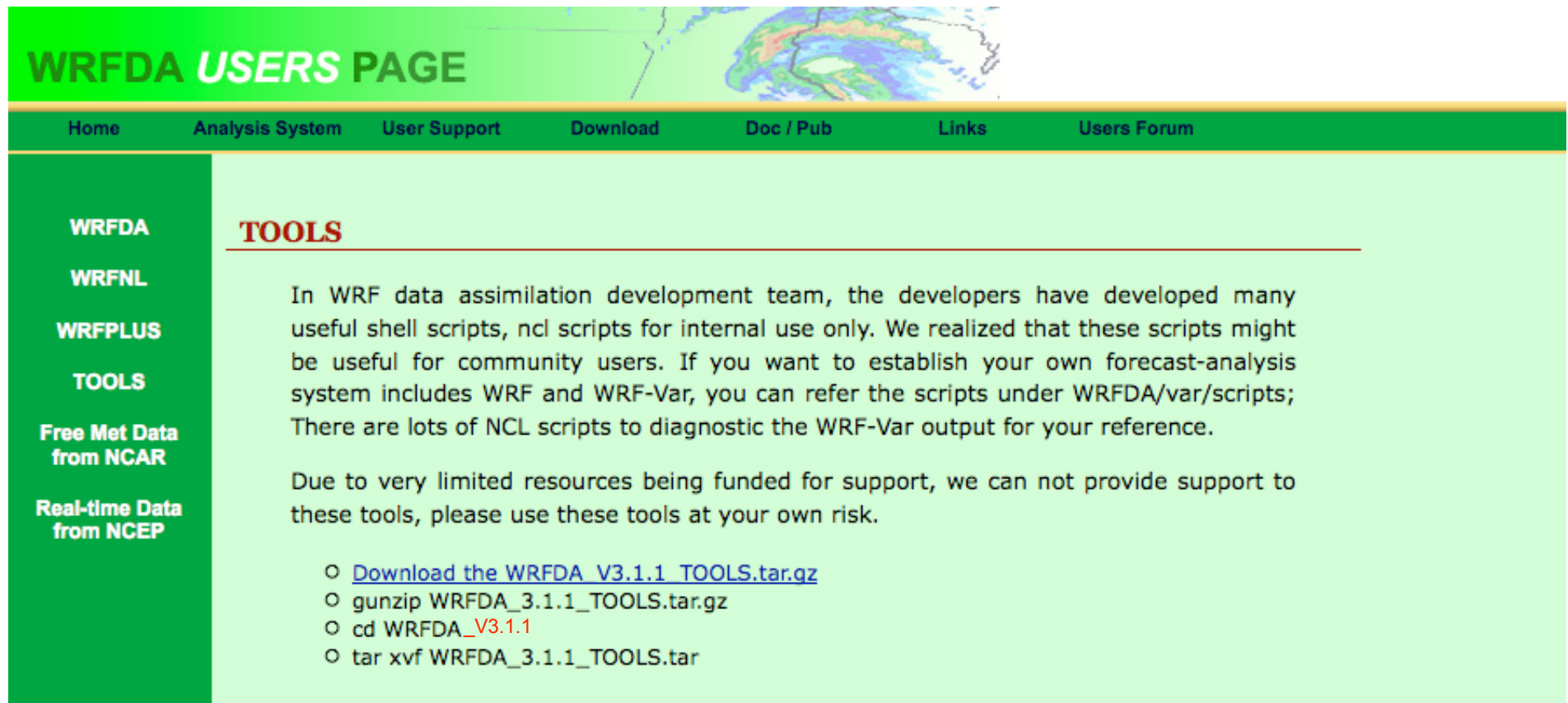
b: background term
 o: observation term
 c: JcDFI term
 e: alpha term
 p: radiance variational bias correction term
 s: skin temperature or cloud cover term

- 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`* (see the next slide)



<http://www.mmm.ucar.edu/wrf/users/wrfda/download/tools.html>

The screenshot shows the WRFDA Users Page. At the top, there is a green header with the text "WRFDA USERS PAGE" in white. Below the header is a navigation bar with links: Home, Analysis System, User Support, Download, Doc / Pub, Links, and Users Forum. On the left side, there is a green sidebar with links: WRFDA, WRFNL, WRFPLUS, TOOLS, Free Met Data from NCAR, and Real-time Data from NCEP. The main content area has a light green background. The "TOOLS" section is highlighted with a red underline. It contains a paragraph about the development team's scripts, a disclaimer about limited resources, and a list of instructions for downloading and extracting the tools. A weather map is visible in the top right corner of the page.

WRFDA USERS PAGE

Home Analysis System User Support Download Doc / Pub Links Users Forum

WRFDA
WRFNL
WRFPLUS
TOOLS
Free Met Data from NCAR
Real-time Data from NCEP

TOOLS

In WRF data assimilation development team, the developers have developed many useful shell scripts, ncl scripts for internal use only. We realized that these scripts might be useful for community users. If you want to establish your own forecast-analysis system includes WRF and WRF-Var, you can refer the scripts under WRFDA/var/scripts; There are lots of NCL scripts to diagnostic the WRF-Var output for your reference.

Due to very limited resources being funded for support, we can not provide support to these tools, please use these tools at your own risk.

- [Download the WRFDA V3.1.1 TOOLS.tar.gz](#)
- gunzip WRFDA_3.1.1_TOOLS.tar.gz
- cd WRFDA_V3.1.1
- tar xvf WRFDA_3.1.1_TOOLS.tar

- ✓ var/graphics/ncl contains various NCL plotting scripts, see var/graphics/ncl/README

gts_omb_oma_01

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

obs_type		Number of obs		Obs index, Level index, station ID, lat, lon, pressure						For u: Obs, O-B, QC flag, Obs error, O-A			
synop	995	1	1	176556	21.51	-104.90	89973.8836463	3.3147587	1.2193668	2	1.1000000	0.1849281	-1.5412909
				-1.4225501	2	1.1000000	-1.6862257	295.5511624	2.5999150	2	2.0000000	1.3689324	89973.8836463
				-273.5464584	2	100.0000000	-236.6028635	0.0134689	0.0048657	0	0.0036749	0.0050584	

- Measured quantities for each observation type vary:

synop: u, v, t, p, q

metar: u, v, t, p, q

ship: u, v, t, p, q

geoamv: u, v

airep: u, v, t

pilot: u, v

satem: thickness

qscat: u, v

polaramv: u, v

gpspw: tpw

sound: u, v, t, q

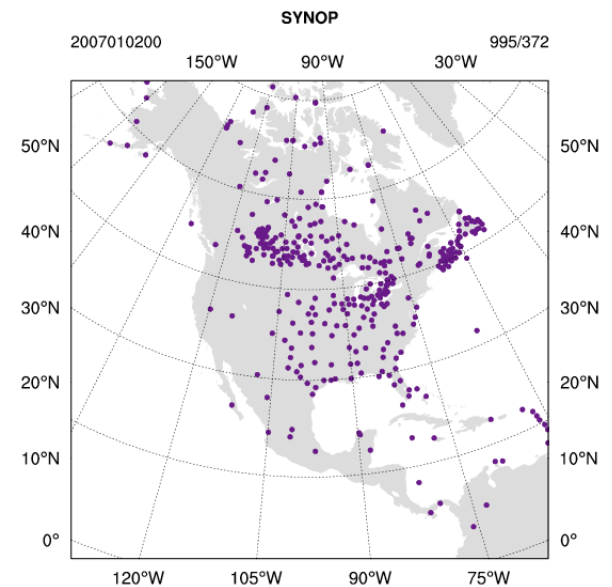
sonde_sfc: u, v, t, p, q

profiler: u, v

buoy: u, v, t, p, q

airsr: t, q

gpsref: ref



✓ WRF-Var tools: plot_gts_omb_oma.ncl

✓ WRF-Var tools: plot_ob_ascii_loc.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

- Contains cost function of 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) should be equal to 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 tuning factors used in WRF-Var:

Tuned obs_error = obs_error * tuning_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_V3.1.1/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  
/  
&fdda  
/  
...  
&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

- `print_detail_grad`

- `.false.` : Default



Outer Iter	EPS Iter	Inner	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 of each observation type at each iteration into **standard output files (rsl.out)** and cost function and gradient values at each iteration into the files called **“cost_fn”** and **“grad_fn”**.



Outer Iter	EPS Iter	Inner	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 $|\text{analysis time} - \text{first guess time}| > \text{Analysis_accu}$, WRF-Var will give a warning like “*Wrong xb time found???*”, but won’t abort.

- ✓ Make sure the time of the first guess file matches the analysis time.

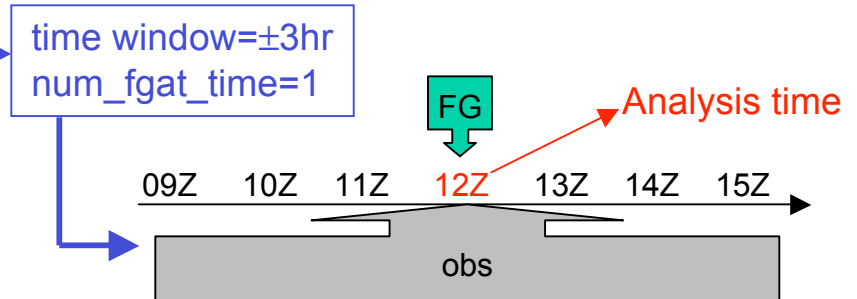
Namelist - WRFVAR3

- **fg_format**: Format of the first guess of WRF-Var.
 - 1 = ARW regional: Default
 - 2 = WRF-NMM regional (not tested)
 - 3 = ARW global (not tested)
 - 4 = KMA global (not tested)
- **ob_format**: The format of the conventional and satellite retrieval observation data going into WRF-Var.
 - 1 = NCEP PREPBUFR (ob.bufr) (gpsro.bufr)
 - 2 = ASCII (ob.ascii): Default
 - 3 = MADIS data format (not tested)

Namelist - WRFVAR3

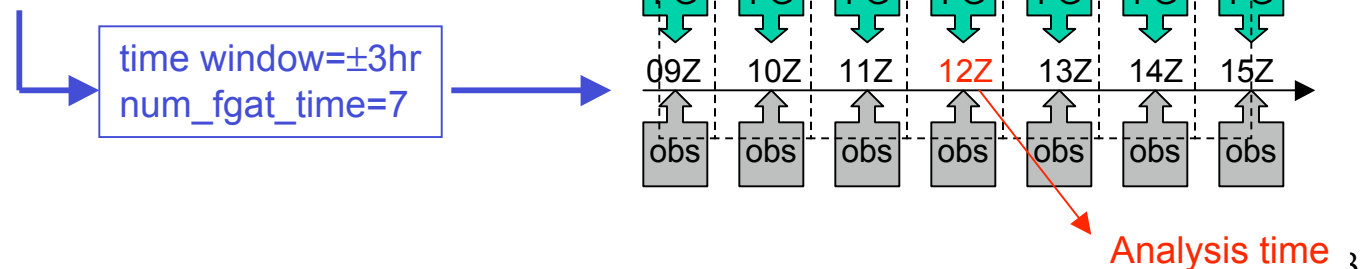
- **num_fgat_time**: Number of data time windows (slots) used in WRF-Var.

- 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.



Running WRF-Var with FGAT

1. prepare hourly obs files using OBSPROC
2. prepare hourly first guess files from previous WRF forecasts
3. when running WRF-Var,
 - a) set num_fgat_time = 7 in namelist.input &wrfvar3
 - b) link hourly obs to be ob01.ascii, ob02.ascii, ..., ob07.ascii
 - c) link hourly first guess (previous WRF hourly forecasts) to be fg01, fg02, ..., fg07
 - d) link first guess valid at analysis time to be fg

OBSPROC

&record9 of namelist.3dvar_obs

```
&record9
  use_for='FGAT'
  num_slots_past=3
  num_slots_ahead=3
```

WRF model

add the following settings (write_input, inputout_interval, input_outname, inputout_begin_h, inputout_end_h) in &time_control of namelist.input

```
&time_control
  write_input      = .true.
  inputout_interval = 60
  input_outname    = 'wrfinput_d<domain>_<date>'
  inputout_begin_h = 3
  inputout_end_h   = 9
```

WRF-Var

&wrfvar3 record of namelistinput

```
&wrfvar3
  num_fgat_time = 7
```

```
ln -sf obs_gts_2007-01-01_21:00:00.FGAT ob01.ascii
ln -sf obs_gts_2007-01-01_22:00:00.FGAT ob02.ascii
ln -sf obs_gts_2007-01-01_23:00:00.FGAT ob03.ascii
ln -sf obs_gts_2007-01-02_00:00:00.FGAT ob04.ascii
ln -sf obs_gts_2007-01-02_01:00:00.FGAT ob05.ascii
ln -sf obs_gts_2007-01-02_02:00:00.FGAT ob06.ascii
ln -sf obs_gts_2007-01-02_03:00:00.FGAT ob07.ascii
```

```
ln -sf wrfinput_d01_2007-01-01_21:00:00 fg01
ln -sf wrfinput_d01_2007-01-01_22:00:00 fg02
ln -sf wrfinput_d01_2007-01-01_23:00:00 fg03
ln -sf wrfinput_d01_2007-01-02_00:00:00 fg04
ln -sf wrfinput_d01_2007-01-02_01:00:00 fg05
ln -sf wrfinput_d01_2007-01-02_02:00:00 fg06
ln -sf wrfinput_d01_2007-01-02_03:00:00 fg07
```

```
ln -sf wrfinput_d01_2007-01-02_00:00:00 fg
```

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_V3.1.1/var/da/da_control/da_control.f90 (e.g., `sound =1`, `synop =2`, ...)
 - 20.0 (km): Default.

Namelist - WRFVAR4

- `use_XXXobs`: Set to true to use particular observation types.
 - e.g, `use_gpsrefobs=.true.:` Assimilate GPS refractivity observations if `gpsref` data are available in the data file.

`XXX`: `synop`, `ships`, `metar`, `sound`, `pilot`, `airep`, `geoamv`, `polaramv`, `buoy`, `profiler`, `satem`, `gpspw`, `gpsref`, `qscat`, `radar`, `airsret`, `hirs2`, `hirs3`, `hirs4`, `mhs`, `msu`, `amsua`, `amsub`, `airs`, `eos_amsua`, `hsb`, `ssmis`
- `use_obs_errfac`: Option for using **tuned observation error** (conventional data only).
 - `.false.` : Default.
 - `.true.` : Use tuned observation error statistics (a file `errfac.dat` needs to be created beforehand by using `da_tune_obs_desrozier.f`).

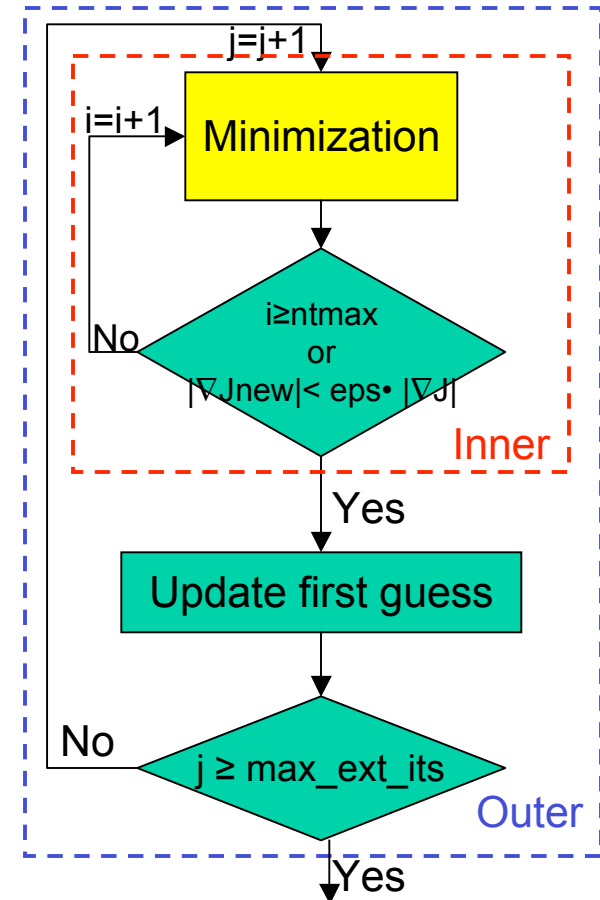
Namelist - WRFVAR5

- `check_max_iv`: Options for rejecting the observations whose innovations (O-B) are larger than a maximum value defined as a multiple of the observation error for each observation. i.e., when $O-B > (obs_error * factor)$, then `fails_error_max`. The default maximum value is 5 times the observation error. The factor of 5 can be changed through `max_error_*` settings.
 - `.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, q, ...): maximum error factor allowed in `check_max_iv` check for t (u/v, pw, ref, q, ...).

Namelist - WRFVAR6

The following namelist variables are for minimization options:

- **max_ext_its**: Number of **outer loops**.
 - 1: Default. Only one outer loop.
 - allowed maximum outer loop number is 10, while common application is 2.
- **ntmax**: Maximum number of iterations in an **inner loop** for the minimization in WRF-Var.
 - 200: Default. The minimization in the inner loop stops when either **ntmax** iteration is reached or **eps** criterion is met.
- **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.



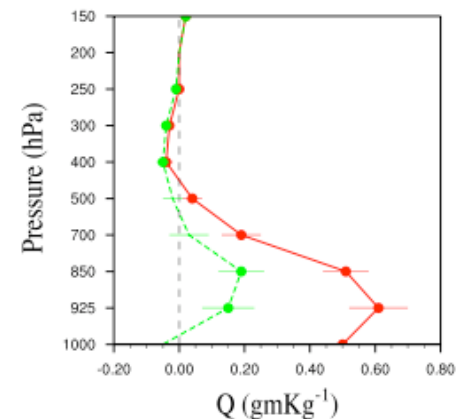
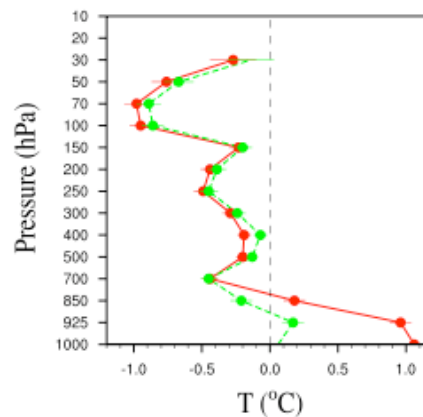
Namelist - WRFVAR11

- `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.

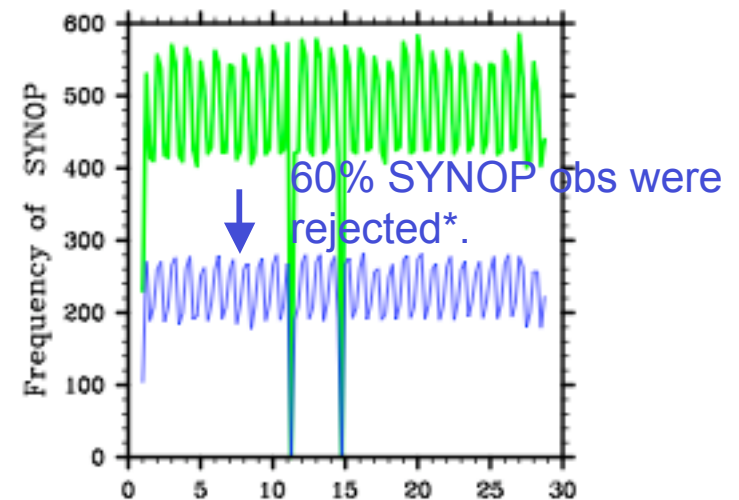


Namelist - WRFVAR11

- `calculate_cg_cost_fn`:
 - `.false.(default)` : Only the initial and final cost functions are computed and output.
 - `.true.` : The cost functions are derived and output into standard output files (`rsl.out`) 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.` (in `&wrfvar1`) to compute and 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**: for creating ensemble perturbations.
 - **VERIFY**: Run WRF-Var verification mode (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

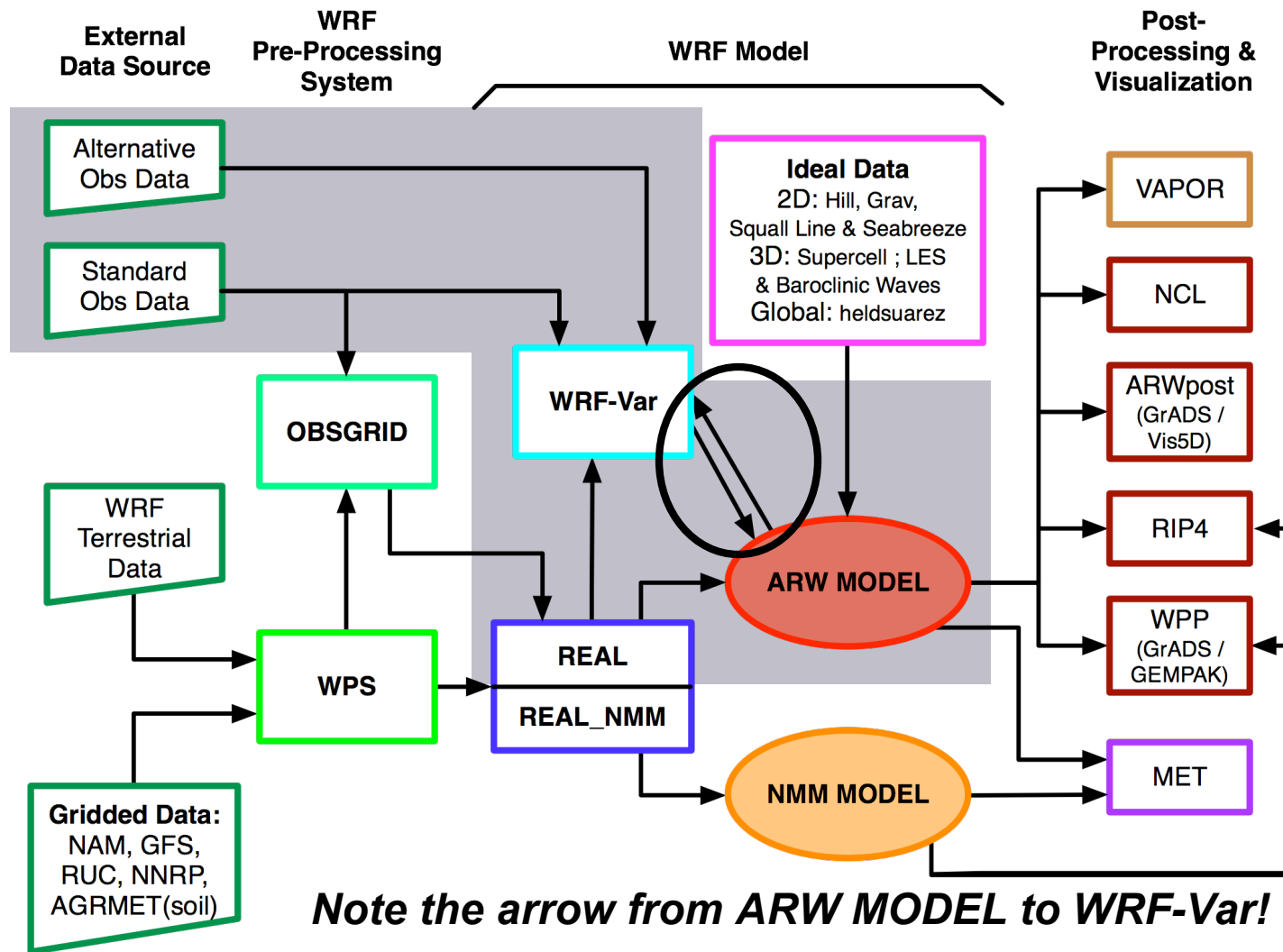
- **analysis_date**: Specify the analysis time. It should be consistent with the first guess time.
 - ✓ If time difference between analysis_date and date info read in from first guess is larger than analysis_accu, WRF-Var will issue a warning message "Wrong xb time found???", but won't abort.

Namelist - WRFVAR21/22

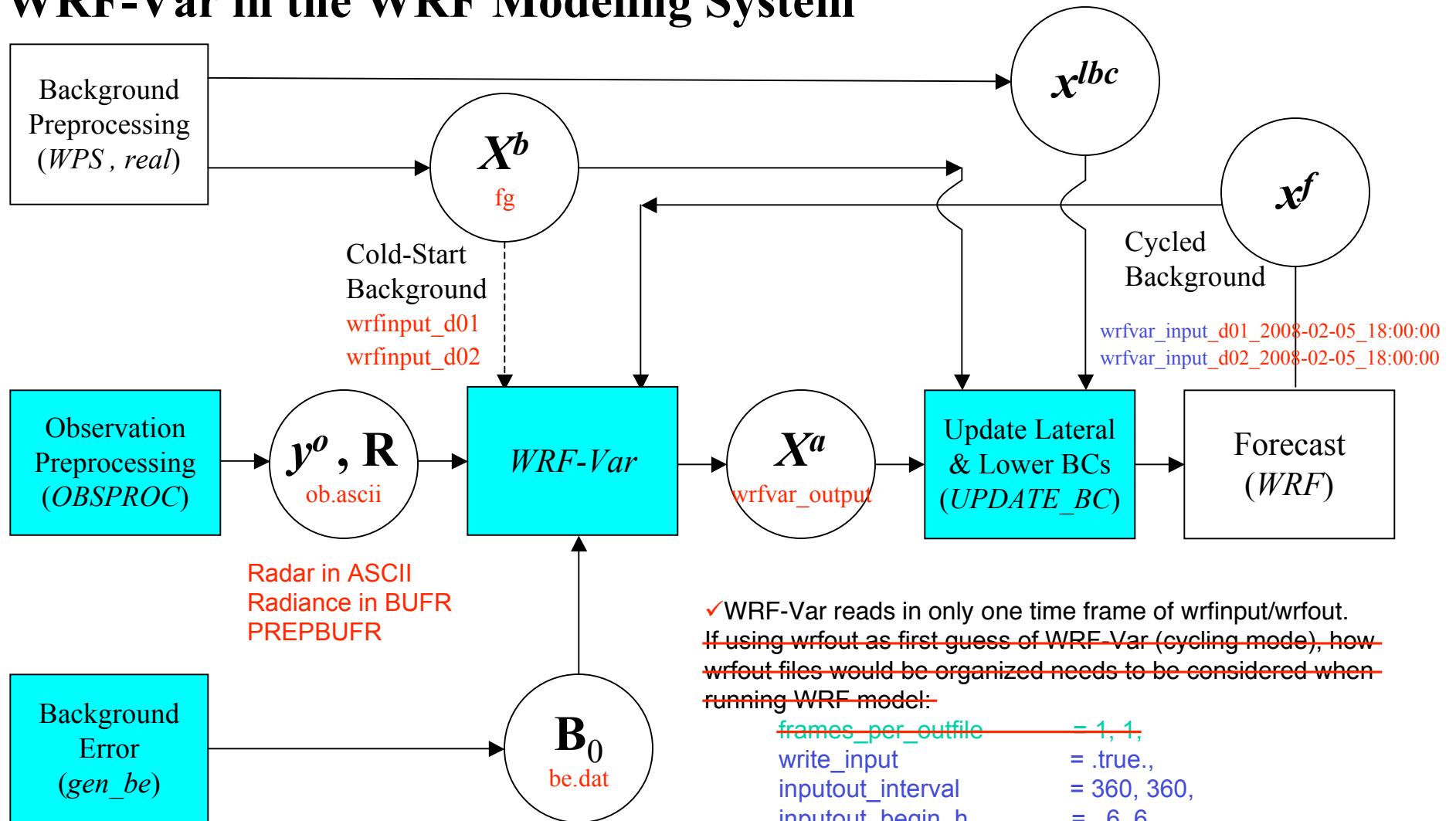
- **time_window_min(max)**: Specify the lower (upper) time values of the assimilation time window.

WRF-Var in the WRF Modeling System

WRF Modeling System Flow Chart



• WRF-Var in the WRF Modeling System



✓ WRF-Var reads in only one time frame of wrfinput/wrfout.
~~If using wrfout as first guess of WRF Var (cycling mode), how wrfout files would be organized needs to be considered when running WRF model:~~

```
frames_per_outfile = 1, 1,
write_input         = .true.,
inputout_interval   = 360, 360,
inputout_begin_h    = 6, 6,
inputout_end_h      = 6, 6,
input_outname="wrfvar_input_d<domain>_<date>",
```

✓ WRF-Var processes only one domain at a time

WRFDA USERS PAGE

<http://www.mmm.ucar.edu/wrf/users/wrfda/index.html>

WRFDA USERS GUIDE

http://www.mmm.ucar.edu/wrf/users/wrfda/Docs/user_guide_V3.1.1/users_guide_chap6.htm

README files contained in the tar file

WRFDA_V3.1.1/README.VAR

WRFDA_V3.1.1/var/README.namelist

WRFDA_V3.1.1/var/README.basics

WRFDA_V3.1.1/var/README.radiance

wrfhelp@ucar.edu