

WRFDA-3DVar Setup, Run, and Diagnostics

Craig Schwartz

Hui-Chuan Lin, Yong-Run Guo,
Syed Rizvi, Xin Zhang, and Hans Huang

WRFDA Tutorial, July 20-22, 2011



Outline

- Installing WRFDA-3DVar
- Setting-up a WRFDA-3DVar run (namelist configurations)
- Making a WRFDA-3DVar run
- Running UPDATE_BC
- WRFDA-3DVar diagnostics

This presentation is based on WRFDA V3.3

Installing WRFDA-3DVar

Source Code

- Download WRFDA source code (**WRFDAV3.3.TAR.gz**) from http://www.mmm.ucar.edu/wrf/users/wrfda/download/get_source.html

After following the instructions at the above URL, on the actual download page, look for

WRF-DA Code and Test Data V3.3 (Released April 6, 2011)

[WRF-DA source code tar file](#) (Version 3.3) - [Details](#).

[WRF-DA home page](#).

FORTRAN 90 Compilers

Supported and tested platforms:

- IBM with XLF compiler V12.1
- Linux with PGI compiler V8.0-4 64-bit
- Linux with INTEL compiler V11.1
- Mac with PGI compiler V10.3-0 64-bit
- Mac with GFORTRAN compiler V4.4.0
- Mac with G95 compiler V4.0.3 (please download the patch of WRFPLUS on WRFDA home page)

✓ To check version numbers, use `xlf -qversion`, `g95 -v`, `pgf90 -V`, `gfortran -v`

Unsupported platforms

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

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

WRFDA does not work on

- PGI 5.x
- Windows

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

Libraries Required by WRFDA

- **NetCDF:** Network Common Data Form:

<http://www.unidata.ucar.edu/software/netcdf/>

Set environmental variable for the NetCDF library:

```
> setenv NETCDF $your_installation_dir/netcdf
```

- ✓ Make sure the NetCDF library is compiled using the same compiler that will be used to build WRFDA, since libraries produced by one compiler may not be compatible with code compiled by another.

Optional Libraries

- If using PREPBUFR or radiance data:
 - BUFR: Binary Universal Form for the Representation of meteorological data
<http://www.nco.ncep.noaa.gov/sib/decoders/BUFRLIB/>
Included in WRFDA (as of V3.1.1)
- If assimilating radiance data, either CRTM or RTTOV is required:
 - CRTM: Community Radiative Transfer Model (version REL_2.0.2)
<ftp://ftp.emc.ncep.noaa.gov/jcsda/CRTM/>
Included in WRFDA (as of V3.2.1)

Set environmental variables:

```
> setenv BUFR 1
```

```
> setenv CRTM 1
```

- RTTOV: Radiative Transfer for TOVS (version 10.0)

<http://research.metoffice.gov.uk/research/interproj/nwpsaf/rtm>

```
> setenv RTTOV path_where lib/librttov10.1.0_*.a
```

libraries are located

Configure WRFDA

- > `cd $your_sourcecode_dir/WRFDA`
- > `./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, smpar, dm+sm may not work properly because the MPI library may not be thread-safe.

Compile WRFDA

- > **./compile all_wrfvar >&! log_compile**
 - 41 executables in the var/build directory (linked to var/da):
 - da_wrfvar.exe: WRFDA
 - da_update_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 the -r4 option while WRFDA compiles with -r8.** For this reason, WRF and WRFDA cannot reside and be compiled under the same directory.

Setting-up a WRFDA-3DVar run

WRFDA-3DVar Equation

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y} - H(\mathbf{x}))$$

$J(\mathbf{x})$: Scalar cost function

$\mathbf{x}$: The analysis: **what we're trying to find!**

$\mathbf{x}_b$: Background field

$\mathbf{B}$: Background error covariance matrix

$\mathbf{y}$: Observations

H : Observation operator: **computes model-simulated obs**

$\mathbf{R}$: Observation error covariance matrix

User-provided Data

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y} - H(\mathbf{x}))$$

$J(\mathbf{x})$: Scalar cost function.....WRFDA output

$\mathbf{x}$: The analysis.....WRFDA output

$\mathbf{x}_b$: Background field.....user input

$\mathbf{B}$: Background error covariance matrix.....user input

$\mathbf{y}$: Observations.....user input

H : Observation operator.....included in WRFDA

$\mathbf{R}$: Observation error covariance matrix.....user input

Sources of User-provided Data

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y} - H(\mathbf{x}))$$

- Where do the input files come from?

Symbol	Description	Source
$\mathbf{x}_b$	Background (“first-guess”)	real.exe or previous WRF forecast
$\mathbf{B}$	Background error covariances	“gen_be” or default file provided with WRFDA
$\mathbf{y}$	Observations	“obsproc” output or NCEP BUFR files
$\mathbf{R}$	Observation error covariances	“obsproc” output or NCEP BUFR files

User-determined run-time options... the Namelist

- ✓ The namelist variables discussed in the following slides refer to WRFDA-3DVar runs and conventional data assimilation only.
- ✓ Please refer to specific lectures (background error covariance, radiance assimilation, ...) for other namelist options.

What is a Namelist?

- The Fortran namelist ([namelist.input](#)) file allows the user to configure a WRFDA run without recompiling the code.
 - Specific Fortran 90 namelist format:

```
&namelistname      - start  
...  
/                  - end
```

- Descriptions of WRFDA namelist variables are given in the [WRF User's Guide](#) and [README.namelist](#) in the WRFDA release (**WRFDA/var/README.namelist**).

WRFDA Namelist

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

```
&wrfvar1  
/  
&wrfvar2  
/  
...  
&wrfvar22  
/  
&time_control  
/  
&fdda  
/  
...  
&namelist_quilt  
/
```

1) WRFDA namelist options:

Running options for WRFDA code.

2) WRF namelist options:

WRFDA needs certain information from the WRF configuration including domain and time settings.

✓ Append your WRF namelist.input to the end of &wrfvar22 to create a complete/consistent namelist.input for WRFDA

Run-time Configurations

- The next several slides pose configuration questions that should be considered before making a WRFDA-3DVar run.
 - Important to carefully think about your configurations!
- The appropriate namelist parameters associated with these questions are discussed.

Background Options

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y} - H(\mathbf{x}))$$

• ***What's the format of my background file?***

&WRFVAR3

- fg_format: Format of the first guess field
 - fg_format = 1 : ARW regional, default
 - fg_format = 2 : WRF-NMM regional (not tested)
 - fg_format = 3 : ARW global (not tested)
 - fg_format = 4 : KMA global (not tested)

Background Error Covariance Options

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y} - H(\mathbf{x}))$$

- ***What type of background error covariance do I want to use?***

&WRFVAR7

- **cv_options**: Background error covariance option
 - cv_options = 3** : global, default...see .../var/run/be.dat.cv3
 - cv_options = 5** : regional, generated by “gen_be”
 - cv_options = 6** : regional, generated by “gen_be” with multivariate moisture correlation, new in WRFDA V3.3

Observation Options

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y} - H(\mathbf{x}))$$

• ***What's the format of my conventional observations?***

&WRFVAR3

- **ob_format**: The format of the conventional and satellite retrieval observation data going into WRFDA
 - ob_format = 1** : NCEP PREPBUFR (ob.bufr, gpsro.bufr)
 - ob_format = 2** : ASCII (ob.ascii), default
 - ob_format = 3 : MADIS data format (not tested)

Observation Options

- *What observation types do I want to assimilate?*

&WRFVAR4

```
USE_SYNOPOBS = T,  
USE_SHIPSOBS = T,  
USE_METAROBS = T,  
USE_SOUNDROBS = T,  
USE_PILOTROBS = T,  
USE_AIREPOBS = T,  
USE_GEOAMVOBS = T,  
USE_POLARAMVOBS = T,  
USE_BUOYROBS = T,  
USE_PROFILERROBS = T,  
USE_SATEMROBS = T,  
USE_GPSZTROBS = F,  
USE_GPSPWROBS = T,  
USE_GPSREFROBS = T,  
USE_QSCATROBS = T,  
USE_RADARROBS = F,  
USE_RADAR_RV = F,  
USE_RADAR_RF = F,  
USE_AIRSRETOBS = T,
```

Assimilate this observation type?

Set to either **True** or **False**

Observation Options

- *How much do I want to thin my PREPBUFR obs?*

&WRFVAR4—only for NCEP PREPBUFR OBS (ob_format = 1)

- thin_conv: For thinning NCEP PREFPBUFR obs
 - thin_conv = .true. : default, *should always set to true*
 - thin_conv = .false. : Used only for debugging purposes.
- thin_mesh_conv (max_instruments): Thinning mesh (km) for each type of conventional observation. The observation index/order follows the definitions in WRFDA/var/da/da_control/da_control.f90 (e.g., sound =1, synop =2, ...)
 - By default, thin_mesh_conv = 20.0 (km)

Observation Options

- *What time window for my observations do I want to use?*

&WRFVAR21

- `time_window_min` = "2008-02-05_10:30:00"

&WRFVAR22

- `time_window_max` = "2008-02-05_13:30:00"

- Obs between `time_window_min` and `time_window_max` are processed.
- Note the "WRF format" of the times

Observation Options

• *How strictly do I want to reject conventional observations?*

&WRFVAR5

- `check_max_iv`: Turns on/off an “outlier check” to reject observations whose innovations (O-B) are larger than a value defined as a multiple (a) of the observation error (σ_o) for each observation: i.e., when $O-B > (a * \sigma_o)$, the ob is rejected.
 - `check_max_iv = .true.` : default, typically set to true
 - `check_max_iv = .false.` : Use this option only if the observation data are known to have good quality.
- `max_error_t`, `max_error_uv`, `max_error_pw`, `max_error_q`, `max_error_ref`, `max_error_rv`, `max_error_p`: The factors (a) that multiply σ_o in the `check_max_iv` test. Can be set individually for different meteorological variables.
 - By default, `max_error* = 5.0` for all meteorological variables

Observation Options

• *How do I want to handle surface observations?*

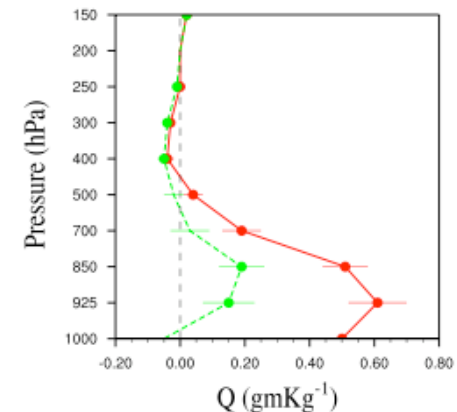
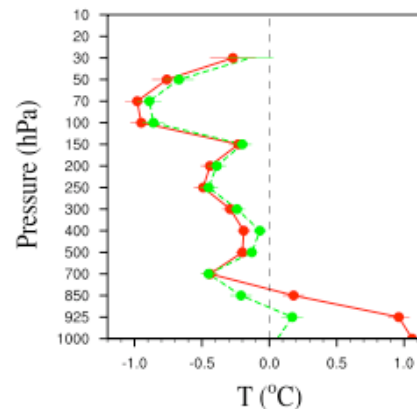
&WRFVAR11

- `sfc_assi_options`:

`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 100 meters.

`sfc_assi_options = 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.



Analysis Options

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y} - H(\mathbf{x}))$$

• *At what time is my analysis valid?*

&WRFVAR18

- `analysis_date` = "2008-02-05_12:00:00"
(should be the same time as in your first-guess file)

Analysis Options

•How do I want to configure minimization options?

&WRFVAR6

- **max_ext_its**: Number of **outer loops**.
 - 1: default. Only one outer loop.
 - Maximum number of outer loops is 10; common application is 2.
- **eps**: Value for minimization convergence criterion. It is an array of dimension=max_ext_its.
 - 0.01(max_ext_its): default. The minimization is considered to converge when the norm of the cost function gradient is reduced at least 2 orders.
- **ntmax**: Maximum number of iterations in an **inner loop** for the minimization in WRFDA.
 - 200: default. The minimization in the inner loop stops when either the **ntmax**th iteration is reached or the **eps** criterion is met.

Analysis Options

- ***What type of analysis do I want to perform?***

&WRFVAR17

- analysis_type: Indicates job type of WRFDA.
 - analysis_type = “3D-VAR” (default): Run 3DVar data assimilation.
 - analysis_type = “RANDOMCV”: Creates ensemble perturbations.
 - analysis_type = “VERIFY” : Run WRFDA in verification mode (forces check_max_iv=.false. and ntmax=0).
 - ✓ Please refer to “**WRFDA Tools and Verification package**” talk.
 - analysis_type = “QC-OBS”: Run 3DVar data assimilation and produce filtered_obs file.
 - ✓ By combining with check_max_iv=.true. and ntmax=0, you can produce a WRFDA filtered (QCd) observation data set (filtered_obs) without running data assimilation.

Analysis Options

• *How much output information do I want?*

&WRFVAR1

- print_detail_grad

print_detail_grad = .false. (default)

print_detail_grad = .true.

Output cost function and gradient values of every observation type each iteration into standard output files (rsl.out)

Analysis Options

•How much output information do I want?

&WRFVAR11

- `calculate_cg_cost_fn`:

`calculate_cg_cost_fn = .false.`

(default) : Only the initial and final cost functions are computed and output in files called “`cost_fn`” and “`grad_fn`”.

`calculate_cg_cost_fn = .true.` :

The cost functions are derived and output at every iteration for diagnostic purposes in “`cost_fn`” and “`grad_fn`”.

`calculate_cg_cost_fn = .false.`

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

`calculate_cg_cost_fn = .true.`

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

List of some namelist variables that are most likely to be user-modified (for conventional observations...red, discussed herein)

&WRFVAR1

PRINT_DETAIL_GRAD = F,

&WRFVAR3

FG_FORMAT = 1,

OB_FORMAT = 2,

NUM_FGAT_TIME = 1

&WRFVAR4

THIN_CONV = T,

THIN_MESH_CONV = 30*20.0

USE_SYNOBOBS = T,

USE_SHIPSOBS = T,

USE_METAROBS = T,

USE_SOUNDOBS = T,

USE_PILOTOBS = T,

USE_AIREPOBS = T,

USE_GEOAMVOBS = T,

USE_POLARAMVOBS = T,

USE_BUOYOBBS = T,

USE_PROFILEROBS = T,

USE_SATEMOBS = T,

USE_GPSZTDOBS = F,

USE_GPSPWOBS = T,

USE_GPSREFOBS = T,

USE_QSCATOBS = T,

USE_RADAROBS = F,

USE_RADAR_RV = F,

USE_RADAR_RF = F,

USE_AIRSRETOBS = T,

&WRFVAR5

CHECK_MAX_IV = T,

MAX_ERROR_T = 5.0,

MAX_ERROR_UV = 5.0,

MAX_ERROR_PW = 5.0,

MAX_ERROR_REF = 5.0,

MAX_ERROR_Q = 5.0,

MAX_ERROR_P = 5.0,

MAX_ERROR_RV = 5.0,

MAX_ERROR_RF = 5.0,

&WRFVAR6

MAX_EXT_ITS = 1,

NTMAX = 200,

EPS = 0.01, 0.01, 0.01

&WRFVAR7

CV_OPTIONS = 5,

AS1 = 0.25, 1.0, 1.5,

AS2 = 0.25, 1.0, 1.5,

AS3 = 0.25, 1.0, 1.5,

AS4 = 0.25, 1.0, 1.5,

AS5 = 0.25, 1.0, 1.5,

RF_PASSES = 6,

VAR_SCALING1 = 1.0,

VAR_SCALING2 = 1.0,

VAR_SCALING3 = 1.0,

VAR_SCALING4 = 1.0,

VAR_SCALING5 = 1.0,

LEN_SCALING1 = 1.0,

LEN_SCALING2 = 1.0,

LEN_SCALING3 = 1.0,

LEN_SCALING4 = 1.0,

LEN_SCALING5 = 1.0,

&WRFVAR11

CHECK_RH = 0,

SFC_ASSI_OPTIONS = 1,

CALCULATE_CG_COST_FN = F,

&WRFVAR15

NUM_PSEUDO = 0,

PSEUDO_X = 1.0,

PSEUDO_Y = 1.0,

PSEUDO_Z = 1.0,

PSEUDO_VAL = 1.0,

PSEUDO_ERR = 1.0

&WRFVAR17

ANALYSIS_TYPE = "3D-VAR"

&WRFVAR18

ANALYSIS_DATE = "2008-02-05_12:00:00"

&WRFVAR19

PSEUDO_VAR = "t"

&WRFVAR21

TIME_WINDOW_MIN = "2008-02-05_10:30:00"

&WRFVAR22

TIME_WINDOW_MAX = "2008-02-05_13:30:00"

✓ Append your WRF namelist.input to the end of &wrfvar22 to create a complete/consistent namelist.input for WRFDA

Running WRFDA-3DVar

Before You Run...

- Ensure the WRFDA executable has been created successfully
 - WRFDA/var/build/da_wrfvar.exe
- Get input files:
 - The test data (WRFDAV3.3-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 environmental variable \$DAT_DIR:
 - > `setenv DAT_DIR your_choice_of_dat_dir`

Before You Run...

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y} - H(\mathbf{x}))$$

- Check input files:
 - **Background** ($\mathbf{x}_b$): \$DAT_DIR/rc/2008020512/wrfinput_d01
 - NETCDF format.
 - For cold-start mode, $\mathbf{x}_b$ is generated by WRF “real.exe”
 - For cycling mode, $\mathbf{x}_b$ is generated by WRF from a previous cycle’s forecast.
 - **Background Error Statistics** ($\mathbf{B}$): \$DAT_DIR/be/be.dat
 - Binary format.
 - Generated by “gen_be” for this specific test case domain.
 - Please refer to “**WRFDA Background Error Estimations**” talk.
 - **Observations** ($\mathbf{y}, \mathbf{R}$) : \$DAT_DIR/ob/2008020512/ob.ascii (conventional obs only)
 - ASCII or PREPBUFR format.
 - Generated by OBSPROC from obs.2008020512, included in the tar file of the test data.
 - Please refer to “**Radar Data Assimilation**” and “**Radiance Data Assimilation**” talks for assimilating radar and radiance data.
- Prepare a WRFDA **namelist** containing runtime options:
 - WRFDA/var/test/tutorial/namelist.input (example)

WRFDA-3DVar Input

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y} - H(\mathbf{x}))$$

Symbol	Description	WRFDA names
$\mathbf{x}_b$	Background (“first-guess”)	<code>./fg</code>
$\mathbf{y}$	Observations	<code>./ob.ascii</code> <u>OR</u> <code>./ob.bufr</code>
$\mathbf{R}$	Observation error covariances	<code>./ob.ascii</code> <u>OR</u> <code>./ob.bufr</code>
$\mathbf{B}$	Background error covariances	<code>./be.dat</code>
N/A	User-defined run-time options (namelist)	<code>./namelist.input</code>
N/A	Land-use table	<code>./LANDUSE.TBL</code>
N/A	WRFDA executable	<code>./da_wrfvar.exe</code>

Working Directory - Input

- Create a working directory, for example, “*your_choice_of_working_dir*”.
 - > `mkdir -p 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 WRFDA:
 - > `ln -sf WRFDA/var/build/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/2008020512/ob.ascii` **`./ob.ascii`**
 - > `cp WRFDA/var/test/namelist.input` **`./namelist.input`**
(or use your own namelist)

Running WRFDA

```
> ./da_wrfvar.exe >&! wrfda.log
```

```
> mpirun -np 8 ./da_wrfvar.exe
```

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

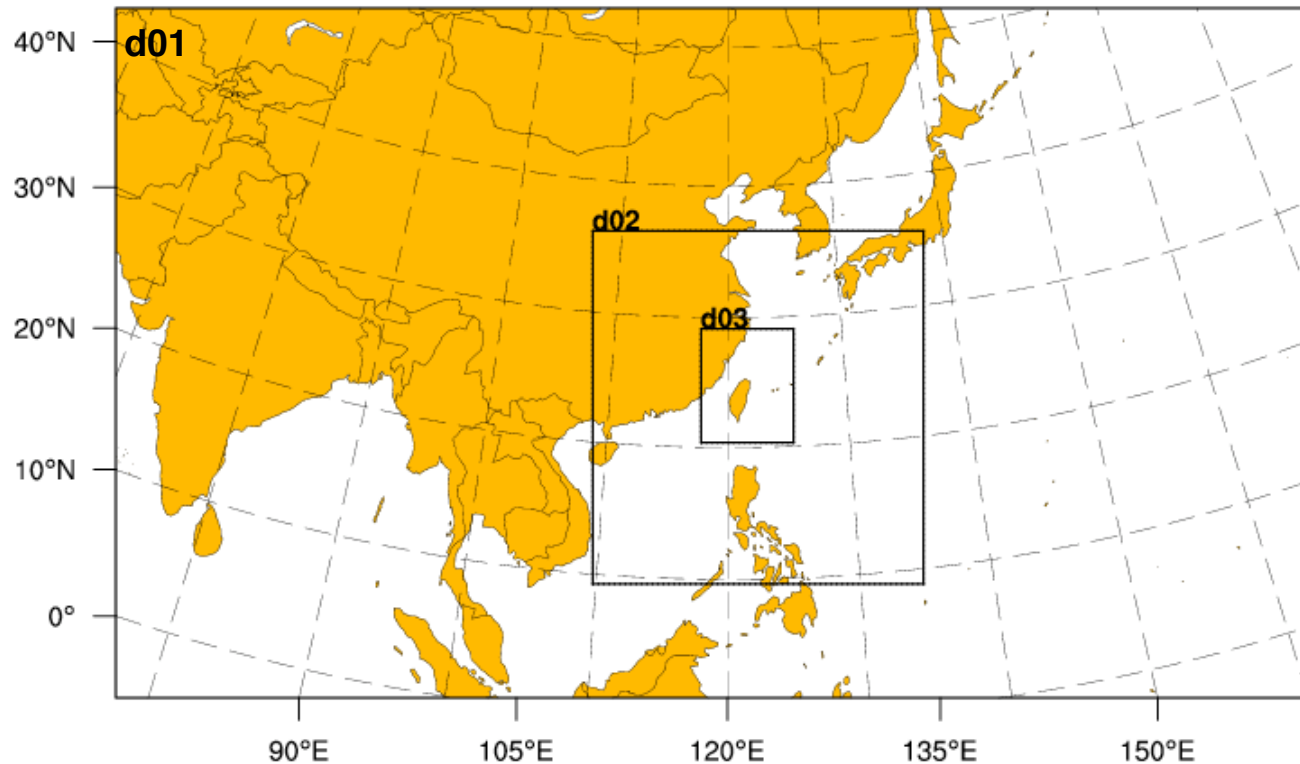
In *your_choice_of_working_dir*, you should have at least the following files after WRFDA is successfully completed:

- **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-wide O-B and O-A statistics)
- **wrfvar_output** (**Analysis x**, the input to the WRF model)

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

A word about nested domains

- WRFDA can only process one file/domain at a time
 - If you wish to run WRFDA for multiple nests, need to run WRFDA separately for each nest

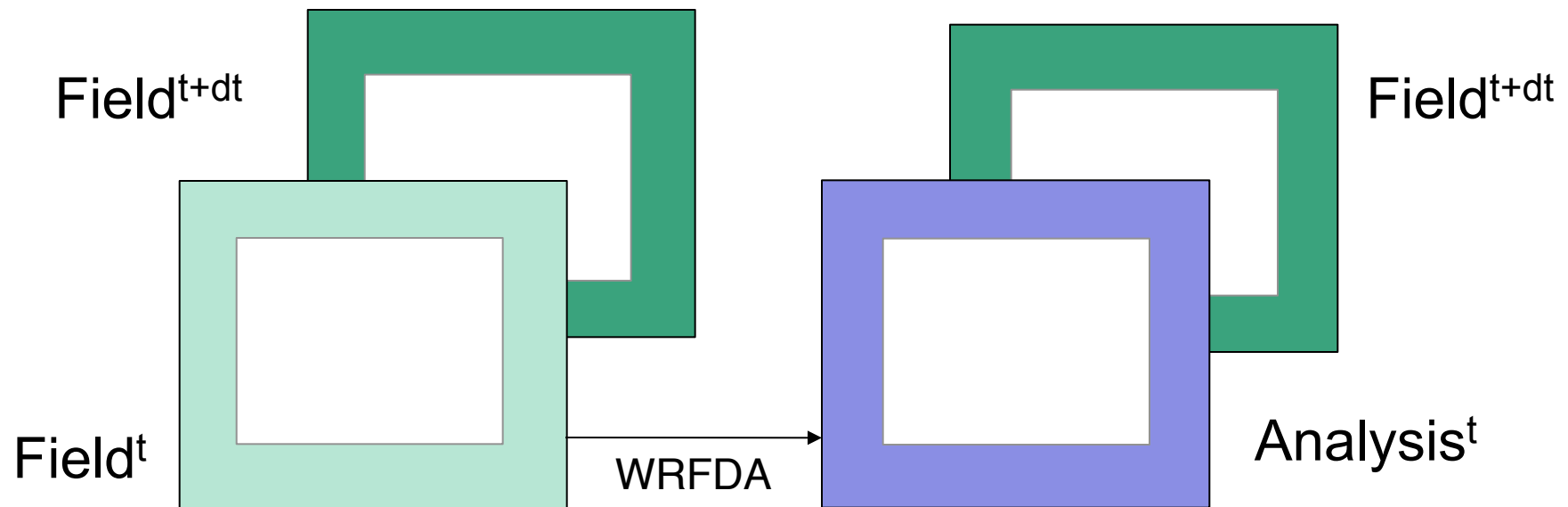


Running update_bc

update_bc

• *Why do we need update_bc?*

- Need to update lateral boundary points to reflect our analysis
- Need to update lateral boundary *tendencies* for the first time
 - Can also update lower boundary conditions



wrfbdy contains the *tendency*
 $(\text{Field}^{t+dt} - \text{Field}^t)/dt$

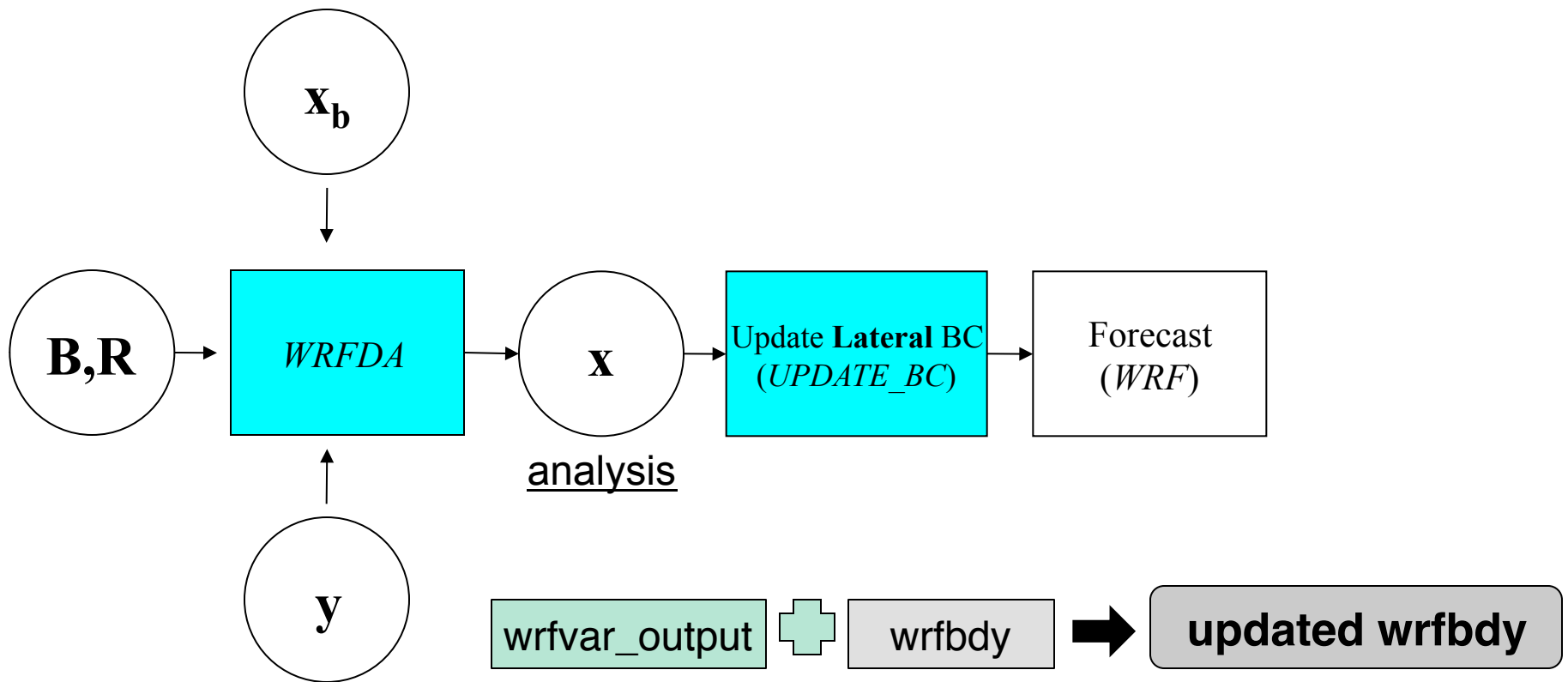
wrfbdy needs to be updated
to be $(\text{Field}^{t+dt} - \text{Analysis}^t)/dt$
after WRFDA

Applications of update_bc

- Cold-start initial conditions (i.e., first-guess from “real.exe”):
 - Update **lateral** boundaries **after** running WRFDA
 - No need to update low boundary before running WRFDA
- Cyclic initial conditions (i.e., first-guess from previous forecast):
 - Update **low** BC **before** running WRFDA
 - Update **lateral** BCs **after** running WRFDA
- Dealing with nested domains:
 - For coarse domain (domain_id = 1), update **low** boundary **before** running WRFDA (if cycling) and **lateral** boundaries **after** running WRFDA
 - For fine mesh domains (domain_id > 1) update **low** boundary for each nest **before** running WRFDA (if cycling)...DO NOT update lateral boundaries

update_bc (lateral boundary condition)

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1}(\mathbf{y} - H(\mathbf{x}))$$



- Always update **lateral BC after** running WRFDA for outermost domain

Steps to Run update_bc (for lateral BC)

- Make sure UPDATE_BC executable has been created successfully:
 - WRFDA/var/build/da_update_bc.exe
- Go into the working directory and prepare the input files for update_bc:
 - > cd *your_choice_of_working_dir*
 - > **cp** \${DAT_DIR}/rc/2008020512/wrfbdy_d01 ./wrfbdy_d01
 - > ln -sf WRFDA/var/build/da_update_bc.exe ./da_update_bc.exe
- Prepare the namelist for update_bc: parame.in

```
&control_param
da_file          = './wrfvar_output'      - Analysis from WRFDA
wrf_bdy_file     = './wrfbdy_d01'         - BC from WPS and WRF real
debug            = .true.
update_lateral_bdy = .true.
update_low_bdy    = .false.
iswater          = 16                     - Should be 17 if using MODIS land-use
/
```

- ./da_update_bc.exe > &! da_update_bc_latbdy.log

(updating **low** boundary for **cycling** runs)

da_update_bc: **update_low_bdy**

TSK: surface skin temperature (over water)

TMN: soil temperature at lower boundary

SST: sea surface temperature

VEGFRA: vegetation fraction

ALBBCK: background snow-free albedo

SEAICE: sea ice flag

IVGTYP: dominant vegetation category (integer)

ISLTYP: dominant soil category (integer)

LANDMASK: 1=land, 0=water

XLAND: 1=land, 2=water

SNOW: snow water equivalent

SNOWC: snow cover

SNOWH: snow depth

} snow over water
needs to be
removed

} fields need to
be consistent
with **SEAICE**

da_update_bc: update_low_bdy & **update_lsm**

SNOW: snow water equivalent

CANWAT: canopy water

RHOSN: snow density

SNOWH: snow depth

SNOWC: snow cover

TSLB: soil temperature

SMOIS: soil moisture

SH2O: soil liquid water

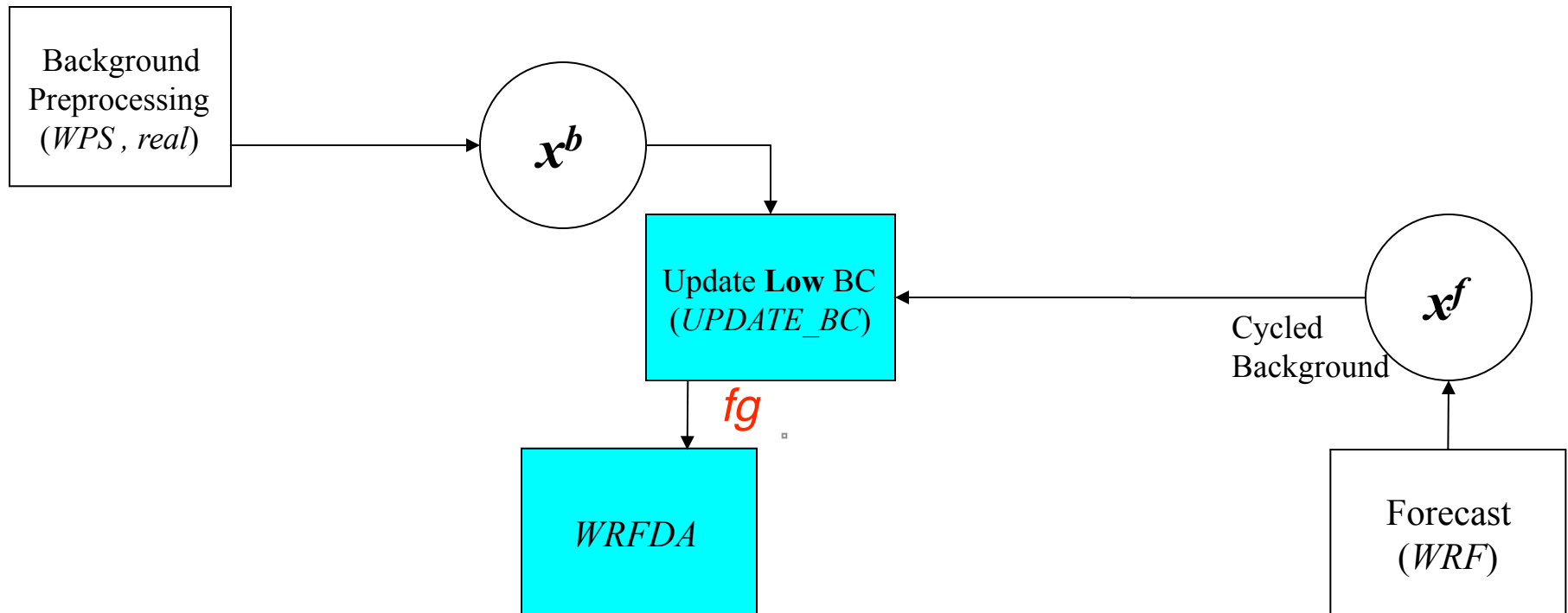
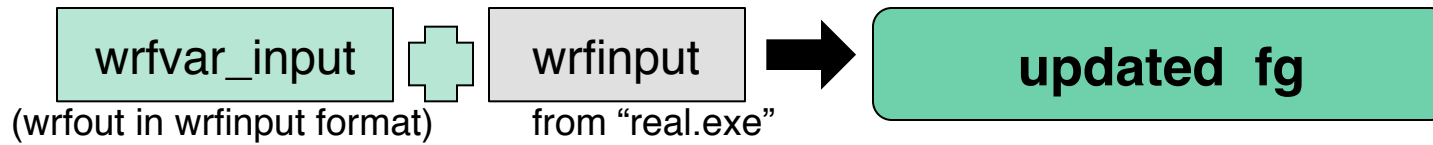
WRFDA adds increments to

- U
- V
- T
- PSFC
- QVAPOR

and modifies

- PH
- P
- MU
- U10
- V10
- T2
- Q2

update_bc (low boundary condition)



- Only need to update low BC if using cyclic initial conditions

Steps to Run update_bc (for low BC)

- Make sure UPDATE_BC executable has been created successfully:
 - WRFDA/var/build/da_update_bc.exe
- Go into the working directory and prepare the input files for update_bc:
 - > cd *your_choice_of_working_dir*
 - > **cp** \${your_wrf_run_dir}/wrfvar_input_d01 ./fg
 - > ln -sf WRFDA/var/build/da_update_bc.exe ./da_update_bc.exe
- Prepare the namelist for update_bc: parame.in

```
&control_param
da_file      = './fg'                - First guess (wrfout in wrfinput format) for WRFDA
wrf_input    = '${DAT_DIR}/rc/2008020512/wrfinput_d01' - IC from WPS and WRF real
debug        = .true.
update_lateral_bdy = .false.
update_low_bdy    = .true.
iswater      = 16                    - Should be 17 if using MODIS land-use
/
```

- ./da_update_bc.exe > &! da_update_bc_lowbdy.log

WRFDA-3DVar Diagnostics

ASCII output files in the WRFDA 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

After each WRFDA run, it is important to:

- ✓ Check the log file (or rsl.out.0000) to see if WRFDA has completed successfully, how many iterations it took to converge, etc.
- ✓ Check the statistics file to see if the values are reasonable

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

The screenshot shows the 'WRFDA USERS PAGE' with a green header and a navigation bar. The left sidebar contains links to WRFDA, WRFNL, WRFPLUS, TOOLS, Free Met Data from NCAR, and Real-time Data from NCEP. The main content area is titled 'TOOLS' and contains a paragraph about shell scripts, a disclaimer about support, and a list of commands to download and extract the tools.

WRFDA USERS PAGE

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

WRFDA

TOOLS

WRFNL

WRFPLUS

TOOLS

Free Met Data from NCAR

Real-time Data from NCEP

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 V3.3_r4036_TOOLS.tar.gz](#)
- gunzip WRFDA_V3.3_r4036_TOOLS.tar.gz
- cd WRFDA
- tar xvf WRFDA_V3.3_r4036_TOOLS.tar

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

wrfda.log (rsl.out.0000)

- Very important information about your WRFDA 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`” WRFDA namelist options (`&wrfvar1`) (using these options, the log file size could become quite 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 WRFDA is run, a namelist.output file is 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 to WRFDA) but contains the observations filtered by WRFDA
- To output this file, set WRFDA namelist option: `analysis_type = "QC-OBS"` (`&wrfvar17`)
- What is filtered_obs for?
 - Can be used for checking what observations are actually assimilated in WRFDA
 - Can be used for running WRFDA in VERIFY mode with `analysis_type = "VERIFY"`
- filtered_obs should NOT be used for running regular WRFDA

rej_obs_conv_01.000

- Contains observations that fail check_max_iv check (if `check_max_iv = .true.`)
 - ✓ 01: outer loop index.
 - ✓ 000: processor id.
- See slide 25 and `&wrfvar5`

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 WRFDA's internal QC (e.g., check_max_iv check) for observations with pressure as a vertical coordinate
 ✓ 01: outer loop index.

WRF-Var data utilization statistics for **outer iteration 1**

			ptop	1000.0	900.0	800.0	600.0	400.0	300.0	250.0	200.0	150.0	100.0	50.0	0.0
obs	type	var	pbot	1200.0	999.9	899.9	799.0	599.9	399.9	299.9	249.9	199.9	149.9	99.9	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

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 equal the printout value in WRFDA log file, e.g., rsl.out.0000:

Final value of Jo = 28880.81069

- Numbers in **blue boxes** are observation error tuning factors used in WRFDA:

Tuned obs_error = obs_error * tuning_factor

Where obs_error values are assigned by OBSPROC and tuning_factor=1 by default.

cost_fn and grad_fn

- Contains values of the cost function and its gradient.
 - If `calculate_cg_cost_fn = .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

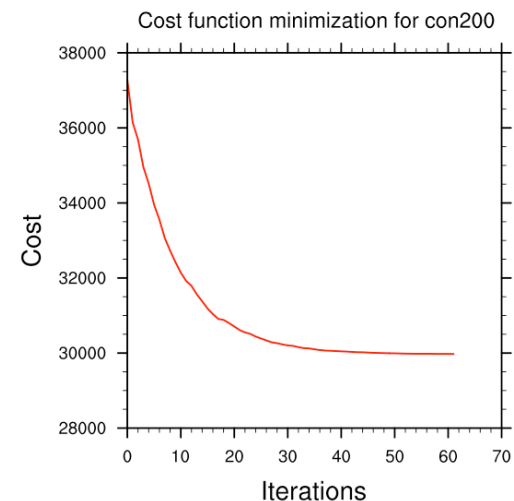
grad_fn

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

✓ WRFDA tools: `plot_cost_grad_fn.ncl`



gts_omb_oma_01

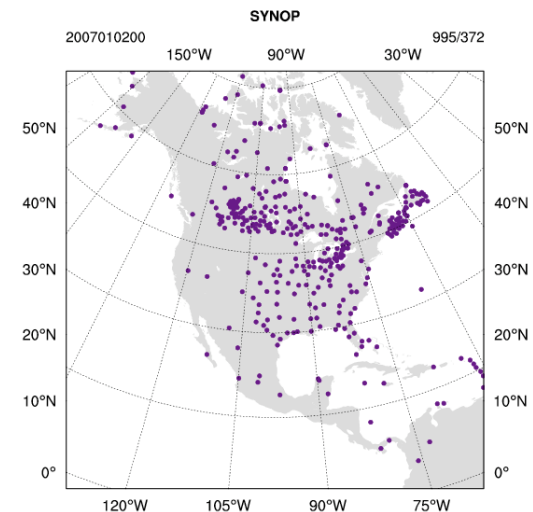
- Contains complete point-by-point, detailed observation information.

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	Number of levels													
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
 gpssp: t, p, q
 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



- ✓ WRFDA tools: plot_gts_omb_oma.ncl
- ✓ WRFDA tools: plot_ob_ascii_loc.ncl

statistics

- Contains domain-wide 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

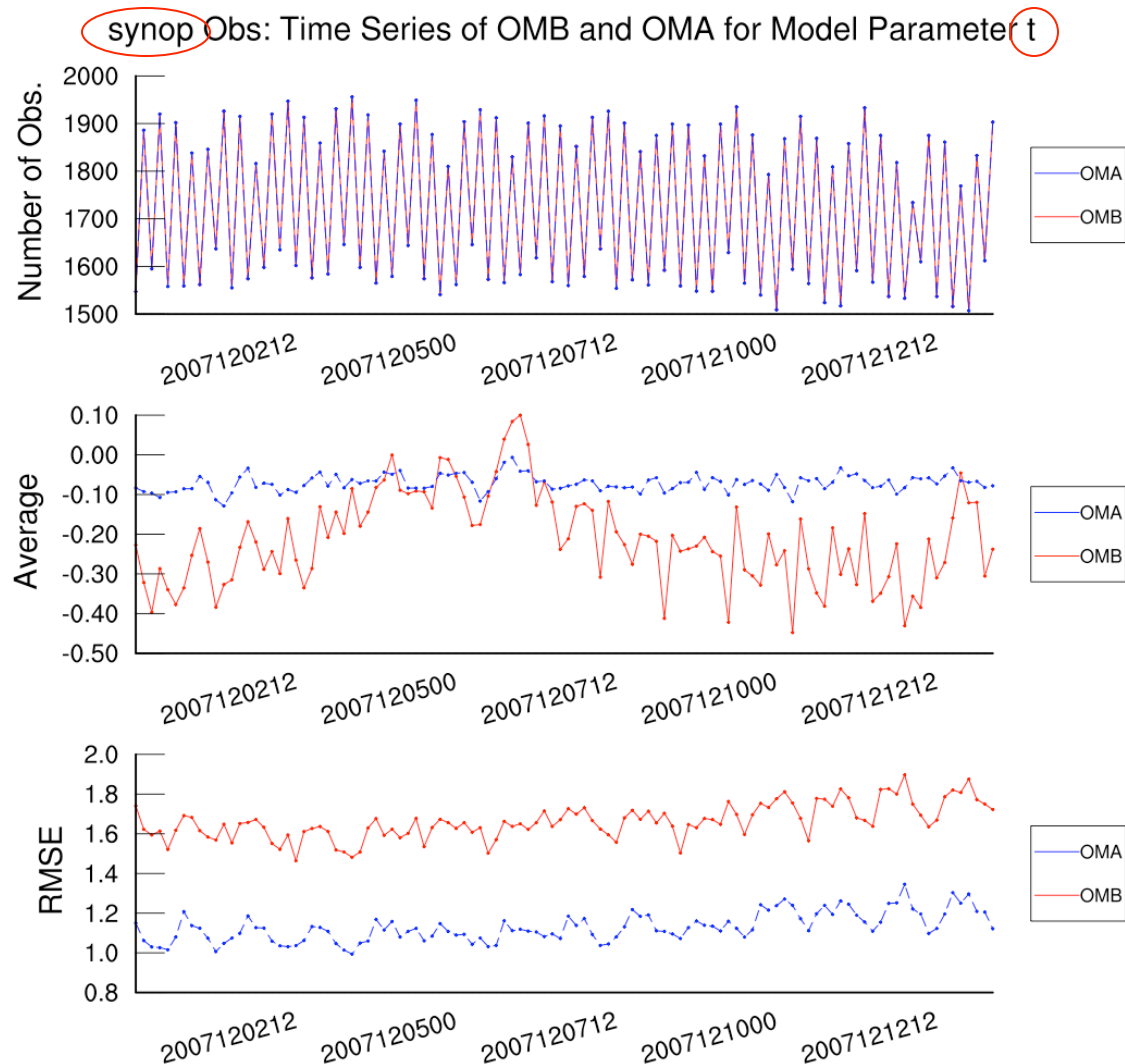
Mean of gridded analysis increments

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

Information contained in **statistics** files can be used to plot time series of O-B and O-A for each observation variable and type.



Wrap-up

A Few Things I Didn't Cover

- First-Guess at Appropriate Time (FGAT; &wrfvar3)
- Background error covariance tuning (&wrfvar7)
- Radiance assimilation (&wrfvar4, &wrfvar14)
- Pseudo-single observation tests (&wrfvar15, &wrfvar19)

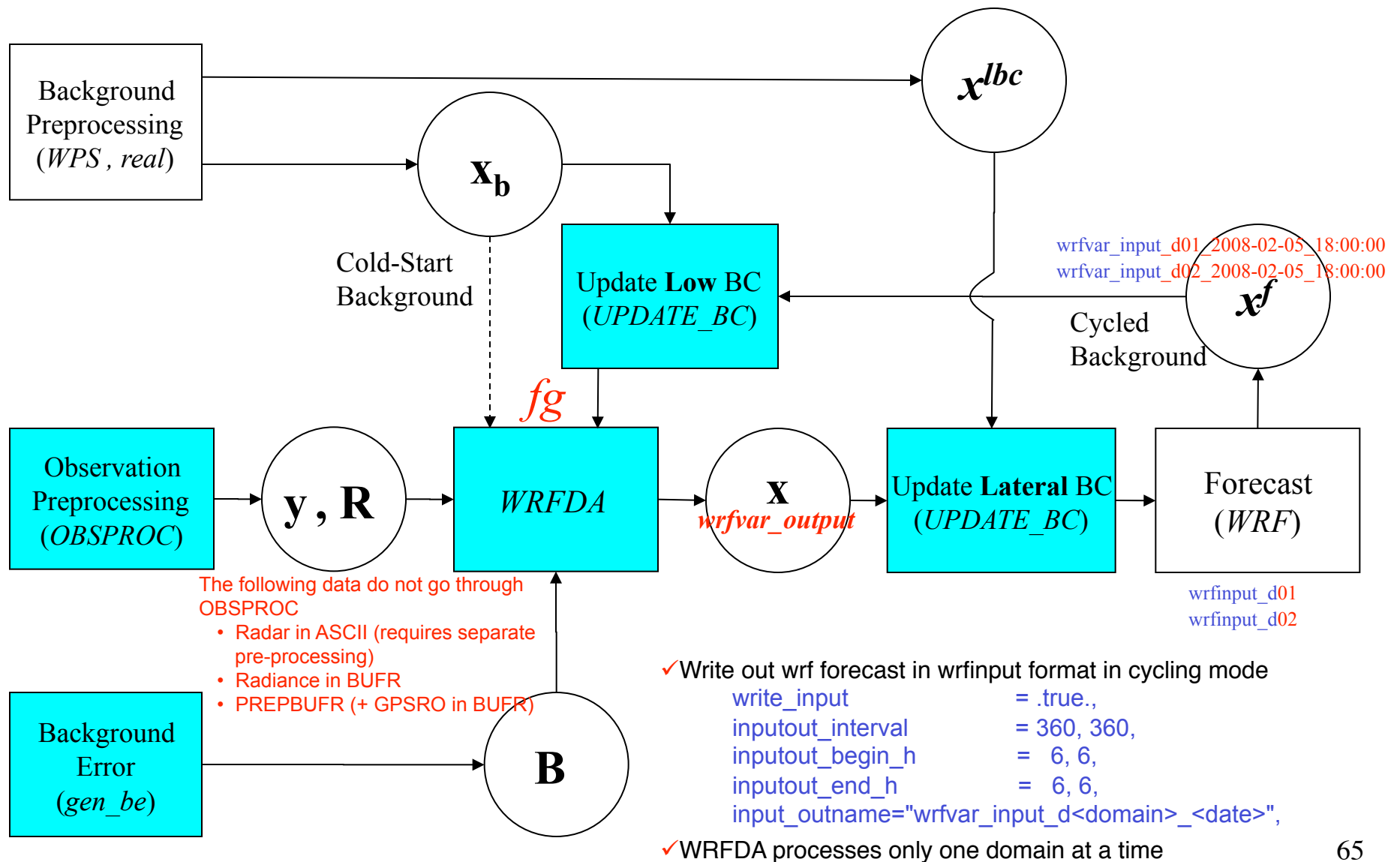
How to Cycle Properly

- In the WRF namelist, add the following (assuming 6-hr cycling):

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

- This will output “wrfvar_input_d<domain>_<date>”.
 - Use this file as your first-guess for the next data assimilation cycle
 - Don't forget to update the lower boundary when cycling!

• WRFDA in the WRF Modeling System



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_V3.3/users_guide_chap6.htm](http://www.mmm.ucar.edu/wrf/users/wrfda/Docs/user_guide_V3.3/users_guide_chap6.htm)

README files contained in the tar file

WRFDA/README.VAR

WRFDA/var/README.namelist

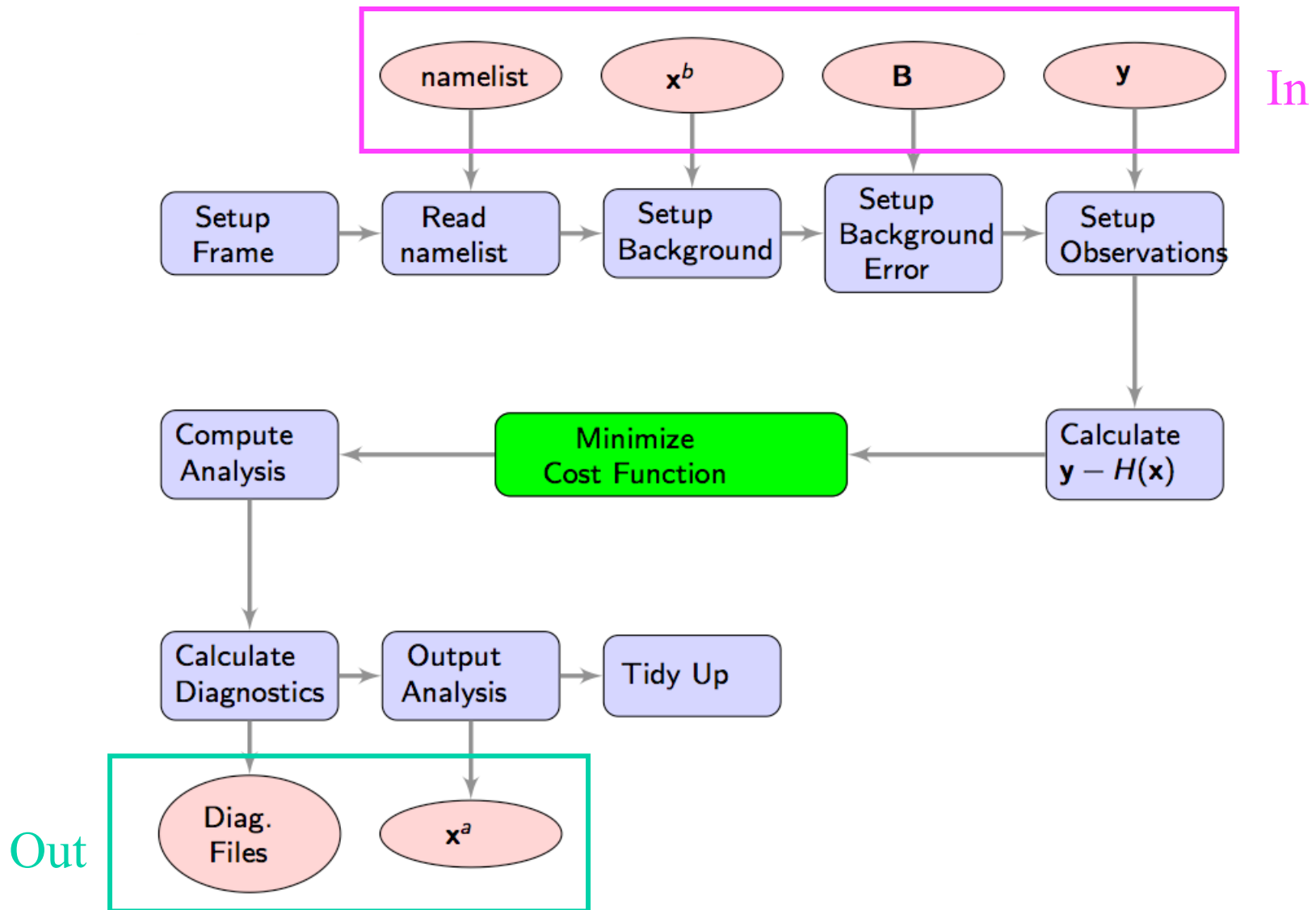
WRFDA/var/README.basics

WRFDA/var/README.radiance

wrfhelp@ucar.edu

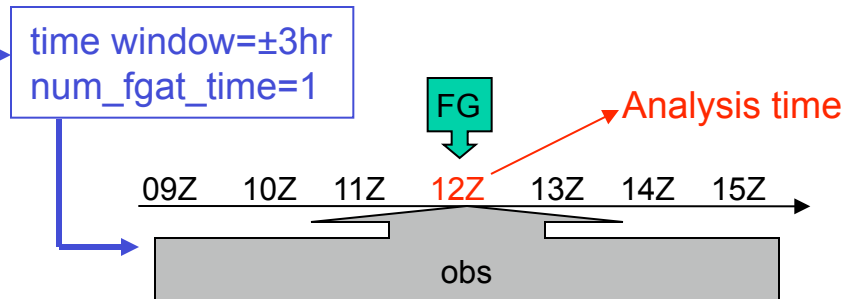
Extra Slides for your Reference

• WRFDA Code Flow



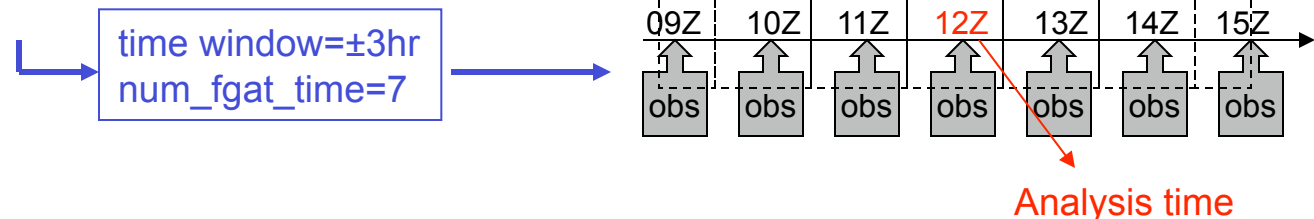
Namelist - WRFVAR3

- **num_fgat_time**: Number of data time windows (slots) used in WRFDA.
 - 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 WRFDA with FGAT

1. prepare hourly obs files using OBSPROC
2. prepare hourly first guess files from previous WRF forecasts
3. when running WRFDA-3DVar,
 - 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
```

WRFDA-3DVar

&wrfvar3 record of namelist.input

```
&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 wrfininput_d01_2007-01-01_21:00:00 fg01
ln -sf wrfininput_d01_2007-01-01_22:00:00 fg02
ln -sf wrfininput_d01_2007-01-01_23:00:00 fg03
ln -sf wrfininput_d01_2007-01-02_00:00:00 fg04
ln -sf wrfininput_d01_2007-01-02_01:00:00 fg05
ln -sf wrfininput_d01_2007-01-02_02:00:00 fg06
ln -sf wrfininput_d01_2007-01-02_03:00:00 fg07
```

```
ln -sf wrfininput_d01_2007-01-02_00:00:00 fg
```