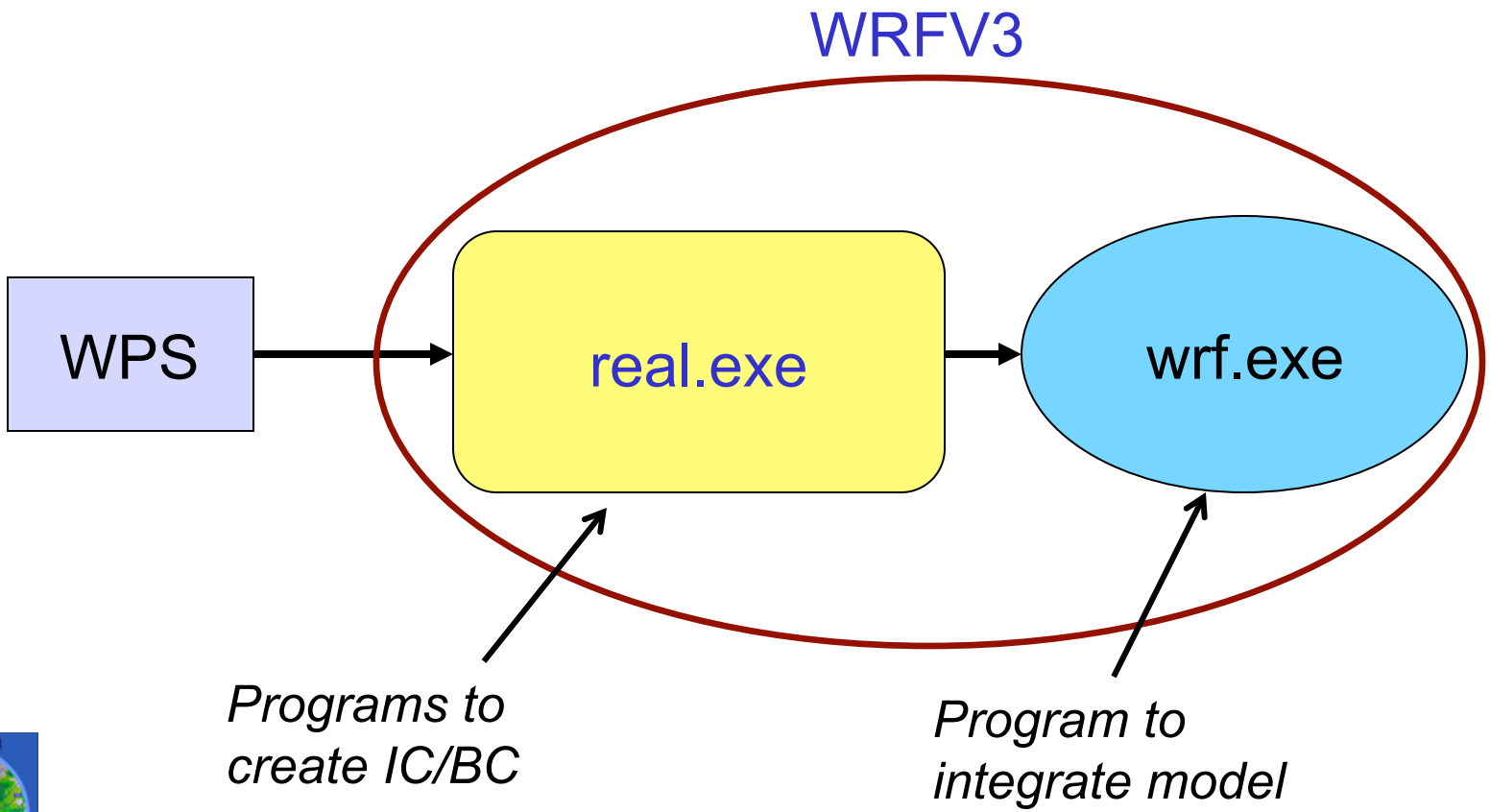
Real and WRF Namelist

Wei Wang, Dave Gill

NCAR/MMM

August 2016

WRF System Flowchart



What is a namelist?

- A Fortran namelist contains a list of *runtime* options for the code to read in during its execution. Use of a namelist allows one to change runtime configuration without the need to recompile the source code.
- A Fortran namelist primary building block is a namelist record, which is given a unique name:

```
&BOB
/
```

- A namelist file may contain a number of records
- In the WRF system, the REAL and WRF programs share a namelist file: **`namelist.input`**

What is a namelist?

- Fortran namelist has very specific format, so edit with care:

```
&BOB
variable_a = value ,
variable_r = 1., 2., 3.,
variable_i = 1, 2, 3,
variable_L = .true.
variable_c = 'Some string'
/
You can put stuff here …
Anything outside of the
NML record & and / couplet
is ignored
```

What is a namelist?

- Fortran namelist has very specific format, so edit with care:

```
&BOB
variable_a = value ,
variable_r = 1., 2., 3.,
variable_i = 1, 2, 3,
variable_L = .true.
variable_c = 'Some string'
/
You can put stuff here …
Anything outside of the
NML record & and / couplet
is ignored
```

Always starts with a "&" character and the name of the namelist record

Always ends with a forward slash "/"

What is a namelist?

- Fortran namelist has very specific format, so edit with care:

```
&BOB
variable_a = value ,
variable_r = 1., 2., 3.,
variable_i = 1, 2, 3,
variable_L = .true.
variable_c = 'Some string'
/
You can put stuff here …
Anything outside of the
NML record & and / couplet
is ignored
```

Can have single valued entries

Can have array type assignments, comma separated

Trailing commas "," are optional

What is a namelist?

- Fortran namelist has very specific format, so edit with care:

```
&BOB
variable_a = value ,
variable_r = 1., 2., 3.,
variable_i = 1, 2, 3,
variable_L = .true.
variable_c = 'Some string'
/
You can put stuff here …
Anything outside of the
NML record & and / couplet
is ignored
```

All standard Fortran data types may be used

REAL

INTEGER

LOGICAL

CHARACTER

What is a namelist?

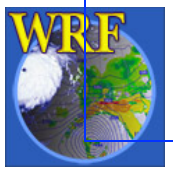
- Fortran namelist has very specific format, so edit with care:

```
&BOB
variable_a = value ! ignored
variable_r = 1., 2., 3.,
variable_i = 1, 2, 3,
variable_L = .true.
variable_c = 'Some string'
/
You can put stuff here …
Anything outside of the
NML record & and / couplet
is ignored
```

Two types of comments are available

Standard Fortran exclamation point "!" syntax

Outside of NML record



What is in the WRF namelist?

- The WRF namelist is broken down into a few larger sections, where each namelist record tends to deal with one aspect of the processing for REAL and WRF

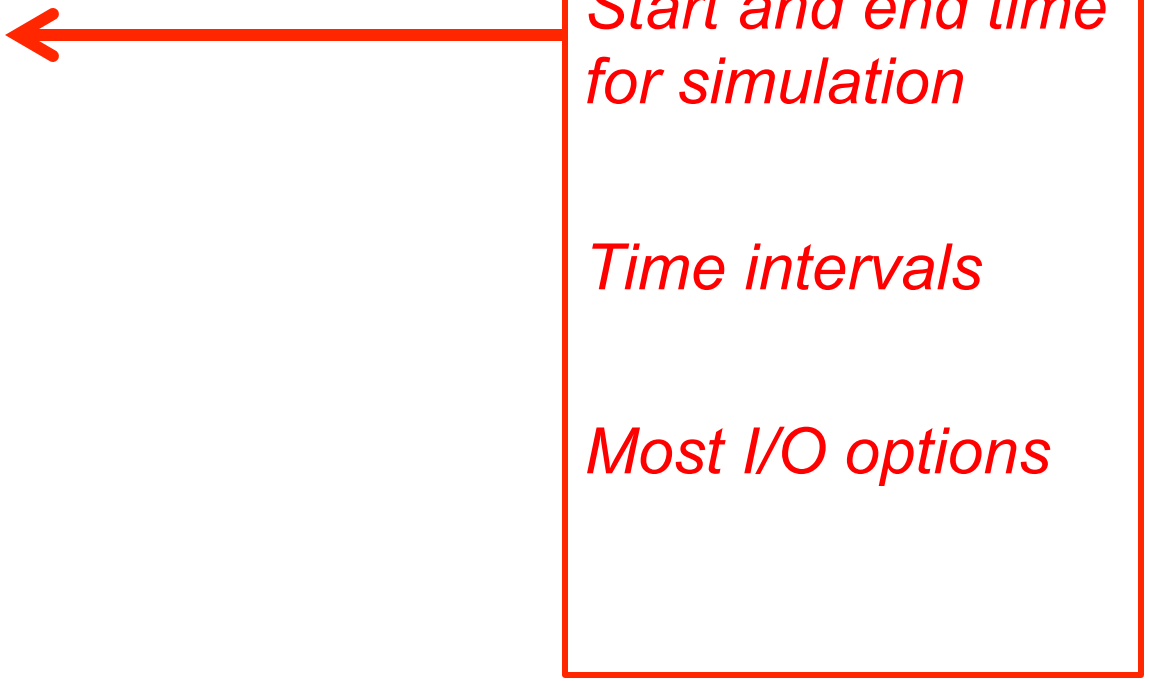
time_control

domains

physics

dynamics

bdy_control



What is in the WRF namelist?

- The WRF namelist is broken down into a few larger sections, where each namelist record tends to deal with one aspect of the processing for REAL and WRF

time_control

domains

physics

dynamics

bdy_control

Number of cells, and physical cell size, lid pressure

Time steps (sec)

Most REAL options

What is in the WRF namelist?

- The WRF namelist is broken down into a few larger sections, where each namelist record tends to deal with one aspect of the processing for REAL and WRF

time_control

domains

physics ←

dynamics

bdy_control

Which choice of physical parameterization schemes to run

***All** are available as run-time options*

What is in the WRF namelist?

- The WRF namelist is broken down into a few larger sections, where each namelist record tends to deal with one aspect of the processing for REAL and WRF

```
time_control
domains
physics
dynamics
bdy_control
```

Various filters, advection and diffusion options

What is in the WRF namelist?

- The WRF namelist is broken down into a few larger sections, where each namelist record tends to deal with one aspect of the processing for REAL and WRF

```
time_control
domains
physics
dynamics
bdy_control
```

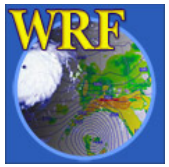
Lateral boundary conditions

What is in the WRF namelist?

- The WRF namelist is broken down into a few larger sections, where each namelist record tends to deal with one aspect of the processing for REAL and WRF

```
time_control
domains
physics
dynamics
bdy_control
```

Lateral boundary conditions



What is in the WRF namelist?

- The WRF namelist is broken down into a few larger sections, where each namelist record tends to deal with one aspect of the processing for REAL and WRF

time_control

domains

physics

dynamics

bdy_control

Focus on options that tend to need to be changed more frequently

namelist record &time_control

```
run_days                            = 0,
run_hours                           = 24,
run_minutes                         = 0,
run_seconds                         = 0,
start_year                          = 2000, 2000, 2000,
start_month                         = 01,   01,   01,
start_day                           = 24,   24,   24,
start_hour                          = 12,   12,   12,
start_minute                        = 00,   00,   00,
start_second                        = 00,   00,   00,
end_year                            = 2000, 2000, 2000,
end_month                           = 01,   01,   01,
end_day                             = 25,   25,   25,
end_hour                            = 12,   12,   12,
end_minute                          = 00,   00,   00,
end_second                          = 00,   00,   00,
interval_seconds                    = 21600
history_interval                    = 180,  60,   60,
frames_per_outfile                  = 1000, 1000, 1000,
restart_interval                    = 360,
restart                             = .true.,
```

Notes on `&time_control`

- *start_** and *end_** time variables:
- *interval_seconds*:

  Time interval between WPS output times, and lateral BC (and lower BC) update frequency

  Both program *REAL* and WRF use

  REAL – Which metgrid time periods to process

  WRF – When to start and end the simulation

Notes on `&time_control`

- *history_interval*: (default units: minutes)
  - Time interval when a history output is written (*note* output is instantaneous)
  - If the **`time_step`** cannot be evenly divided by **`history_interval`**, then nearest time step output is used
  - The time stamp in a history file name is the time when the history file is first written, and multiple time periods may be written in one file. e.g. a history file for domain 1 that is first written for 1200 UTC Jan 24 2000 is

    `wrfout_d01_2000-01-24_12:00:00`

Notes on `&time_control`

- *frames_per_outfile*:

  Number of history times written to one file

  Currently “1” is preferable, as it fixes some NETCDF CF compliancy issues

Notes on &time_control

Example 1: all output times are in a single file

```
history_interval    = 180,  60,   60,
frames_per_outfile = 1000, 1000, 1000,
```

wrfout_d01_2000-01-24_12:00:00

Example 2: each output file only contains a single time

```
history_interval   = 180, 60,  60,
frames_per_outfile = 1,   1,   1,
```

wrfout_d01_2000-01-24_12:00:00
wrfout_d01_2000-01-24_15:00:00
wrfout_d01_2000-01-24_18:00:00

&time_control

```
io_form_history      = 2,
io_form_restart      = 2,
io_form_input        = 2,
io_form_boundary     = 2,
```

IO format options:
- = 1, binary
- = 2, netcdf (most common)
- = 4, PHDF5
- = 5, Grib 1
- =10, Grib 2
- =11, pnetCDF

history = standard WRF model output: wrfout_d0*x*_<*date*>
input = standard REAL output: wrfinput_d0*x*

namelist record **&domains**

```
time_step                           = 180
max_dom                             = 1,
e_we                                = 74,    112,    94,
e_sn                                = 61,    97,     91,
e_vert                              = 28,    28,     28,
num_metgrid_levels                  = 21
num_metgrid_soil_levels             = 4
dx                                  = 30000, 10000,  3333,
dy                                  = 30000, 10000,  3333,
p_top_requested                     = 5000,
```

nest options

Notes on `&domains`

- *time_step*:
  - Time step for model integration in seconds.
  - Typically 5 to 6*DX (DX is grid distance in km)

- *e_we, e_sn, e_vert*:
  - Model grid dimensions (staggered) in X, Y and Z directions.

- *p_top_requested*:
  - Pressure value at the model top.
  - Constrained by the available data from WPS.
  - Default is 5000 Pa (about 20 km)

Notes on **&domains**

- *num_metgrid_levels*:
  - Number of *metgrid* (input) data levels.

- *num_metgrid_soil_levels*:
  - Number of soil data levels in the input data

- *dx, dy*:
  - grid distance: in meters

- Found by typing

ncdump -h met_em.d01.<*date*> | more

namelist record &bdy_control

Do not change second, third, etc column

```
spec_bdy_width      = 5,
spec_zone           = 1,
relax_zone          = 4,
specified           = .true., .false.,.false.,
nested              = .false., .true., .true.,
```

(spec_zone + relax_zone = spec_bdy_width)

Where do I start?

- Always start with a *namelist* template provided in a test case directory
- A number of namelist templates are provided in *test/em_real/* directory

  `namelist.input.4km` ~ 4 km grid size

  `namelist.input.jun01` ~ 10 km grid size

  `namelist.input.jan00` ~ 30 km grid size

Where do I start?

- For different applications, please refer to p5-33 to 5-35 of the ARW User's Guide:
  - 2 or 4 km microphysics-only runs
  - 20 – 30 km, 2 – 3 day runs
  - Antarctic region
  - Tropical storm forecasting
  - Regional climate

Where do I start?

- Use document to guide the modification of the namelist values:
  - **`run/README.namelist`**
  - **`test/em_real/examples.namelist`**
  - User's Guide, Chapter 5 (online version has the latest)
  - Full list of namelists and their default values can be found in Registry files: Registry.EM_COMMON and registry.io_boilerplate (for IO options) (look for character string '*namelist*')