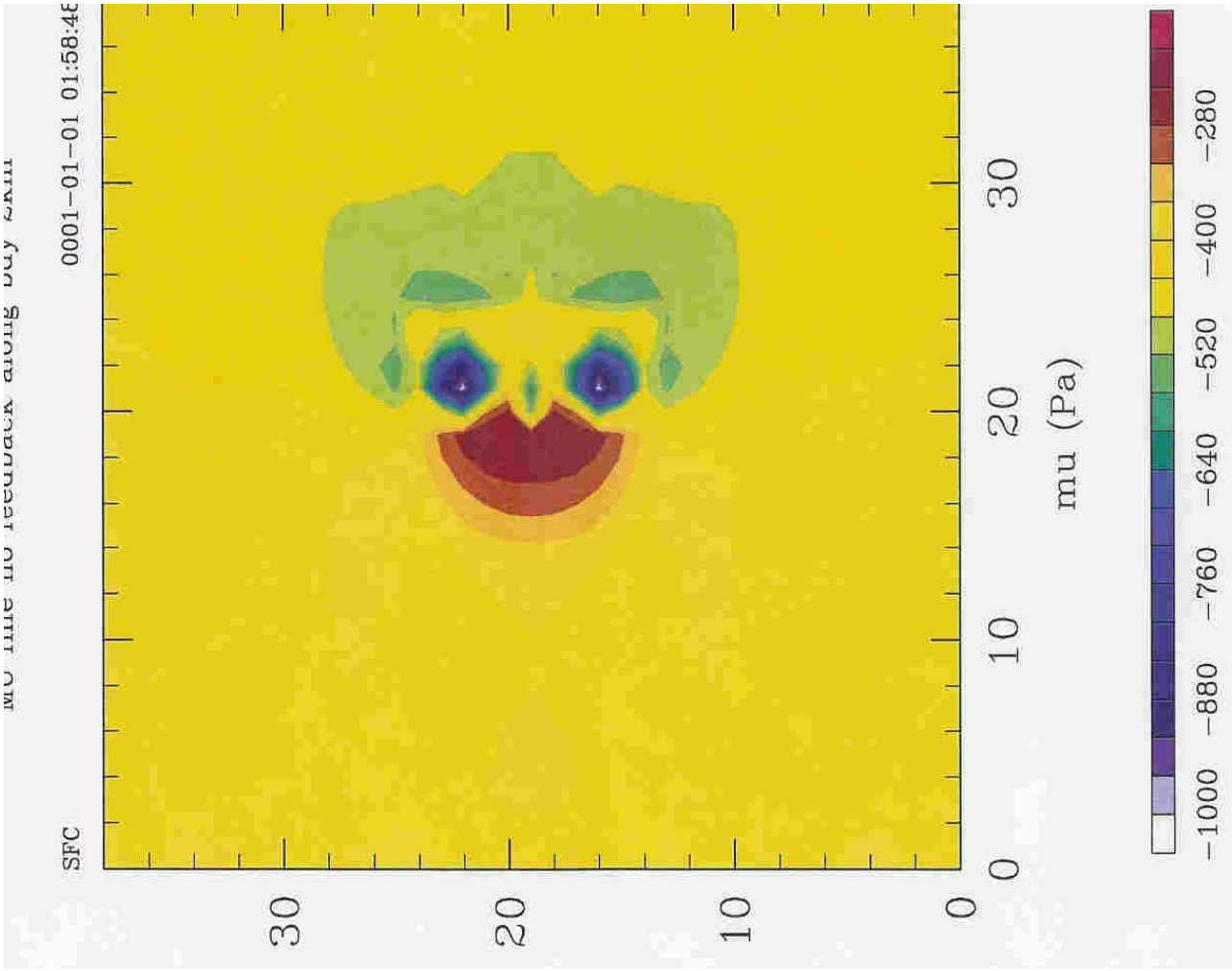
WRF Nesting: Set Up and Run

Wei Wang

Dave Gill

NCAR/ESSL/MMM

WRF Nesting


0001-01-01 01:58:48
SFC
0
10
20
30
mu (Pa)
-1000
-880
-760
-640
-520
-400
-280

Outline

- General comments
- Nest namelist options
- Running WRF with nests
  - ARW case: two-way nesting
  - ARW moving nest
  - ARW one-way nesting
- Summary

Before You Run ..

- Make sure you have selected basic nest compile options and appropriate executables are created in **`WRFV3/main/`** directory:

  For ARW:
  - **`real.exe`**
  - **`wrf.exe`**
  - **`ndown.exe`**
  - **`tc.exe`**

- If you are running a real-data case, be sure that files for *nest* domains from WPS are generated:
  - **`met_em.d01.<date>`**, **`met_em.d0*.<date>`** for ARW

Steps to Run (same as 1 domain)

1. cd to *run/* or one of the *test case* directories
2. Link or copy WPS output files to the directory for real-data cases
3. Edit *namelist.input* file for the appropriate grid and times of the case
4. Run initialization program, *real.exe*, as in the single domain case
5. Run model executable, *wrf.exe*

All in the namelist…

- Nearly all control for a nested run can be achieved by editing the namelist file.
- Look at nest specific namelist options

Important to note:

- Key variable: **`max_dom`** must be set to >= 2
- Need to pay attention to multi-column namelists

Nest namelist Options

&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
```

First column: domain 1 option

These control the start and end times of the nests. They can be different from the parent domain, but must fit in the time window of the parent domain

&time_control

```
interval_seconds     = 21600
history_interval     = 180,  60,   60,
frame_per_outfile    = 1000, 1000, 1000,
restart_interval     = 360,
```

History files may be split into multiple pieces

- History files are written separately for each domain
- History intervals may be different for different domains
- restart files are also written each per domain

&time_control

Nest input option: ARW only

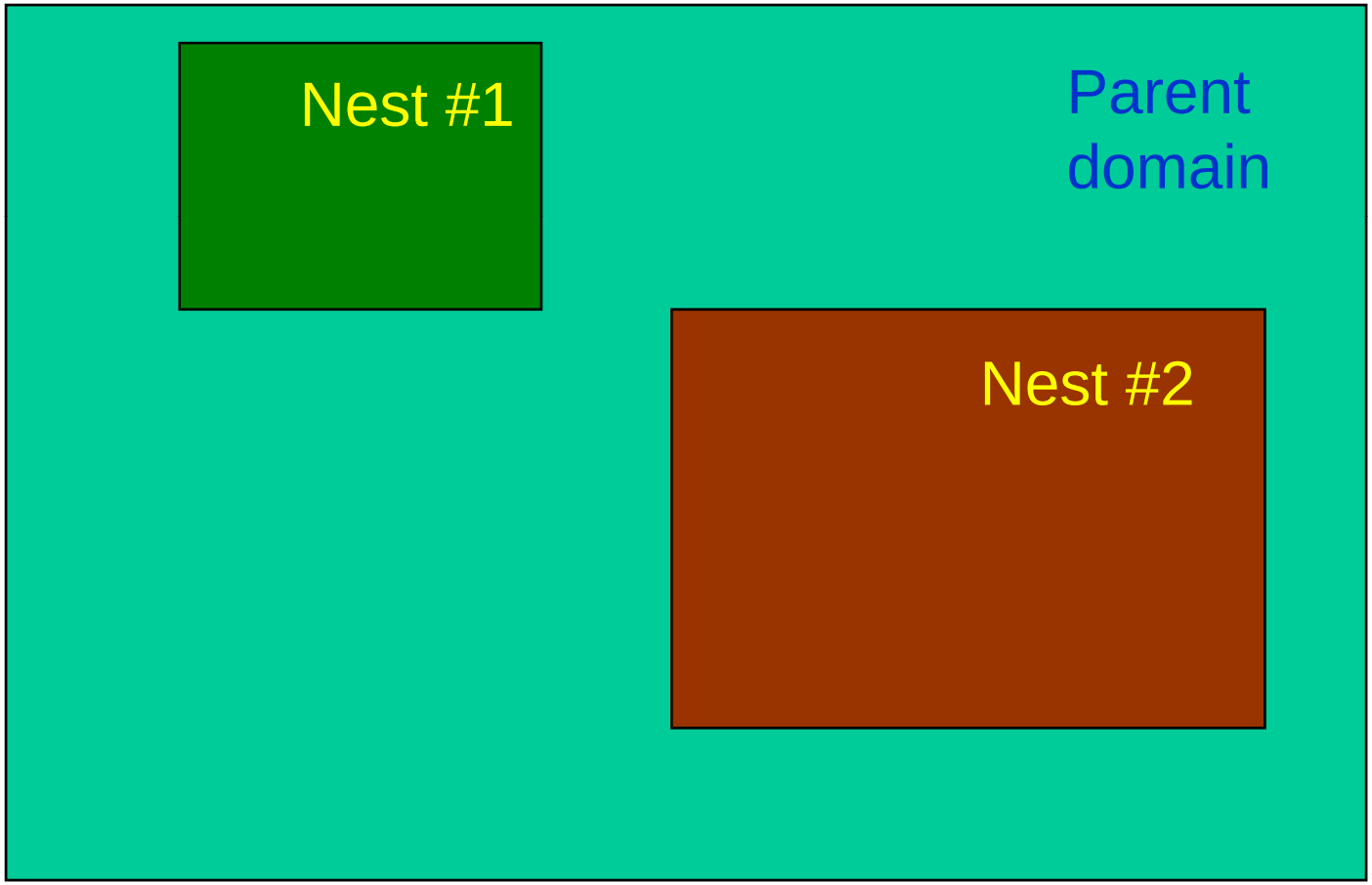
```
input_from_file  = .true., .true., .true.,
fine_input_stream   = 0, 2, 2,
```

Specify what fields to use in nest input:
a) all = 0
b) data specified in I/O stream 2 in Regsitry = 2
Useful for nest starting at a later time.

Whether to produce in *real* and use nest wrfinput files in *wrf*. This is usually the case for real-data runs.

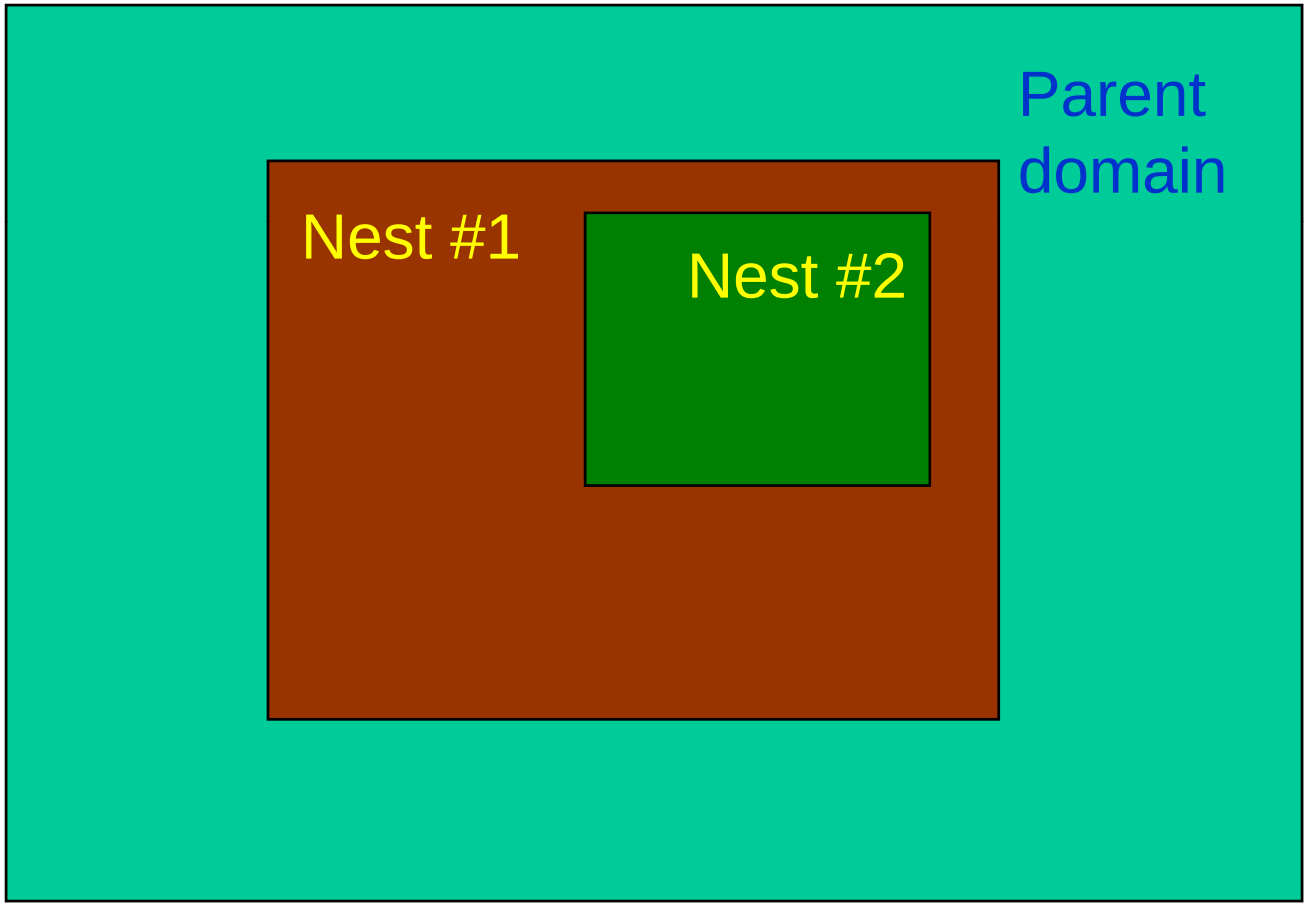
&domains

Two nests on the same “level”, with a common parent domain



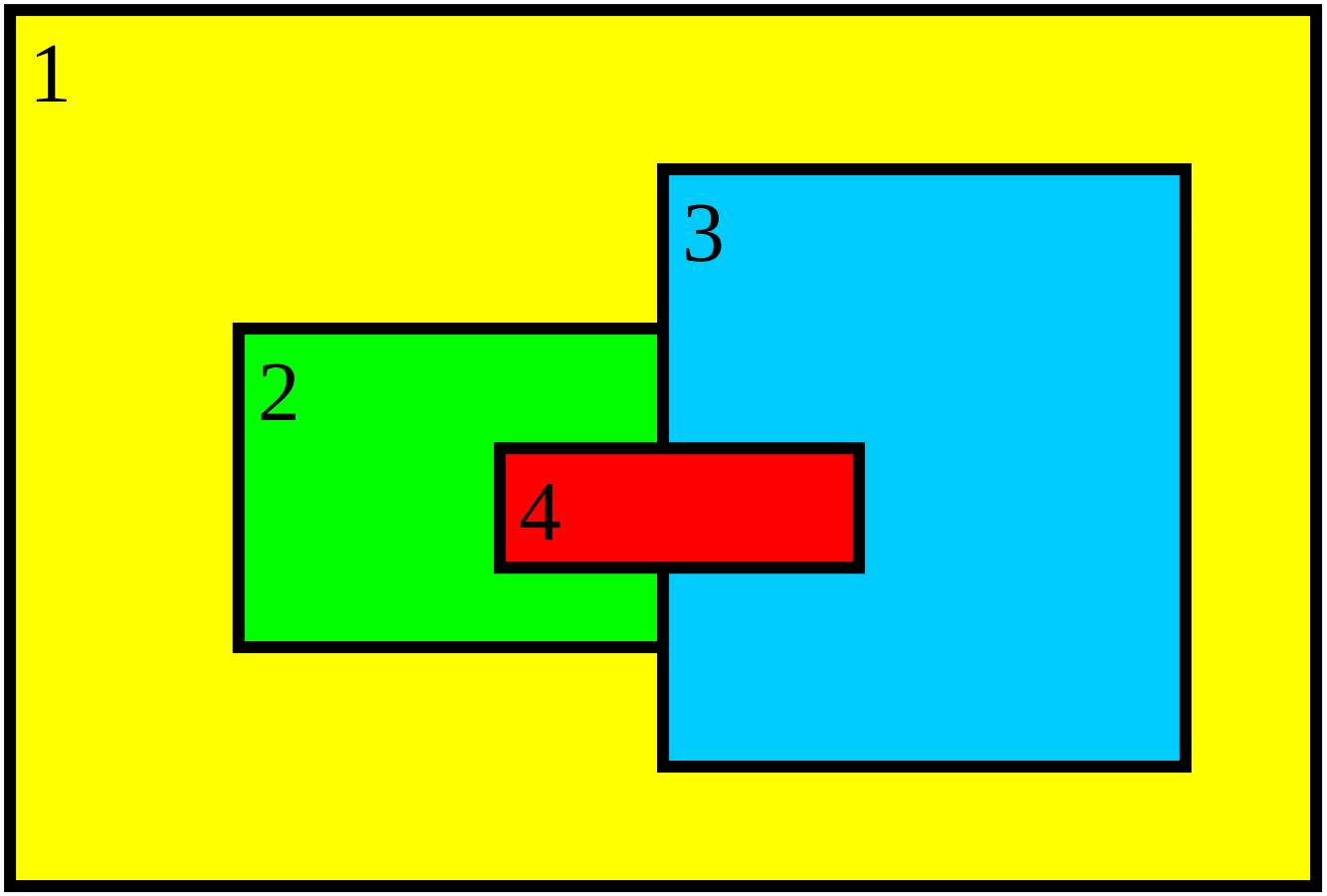
&domains

Two levels of nests, with nest #1 acting as parent to nest #2



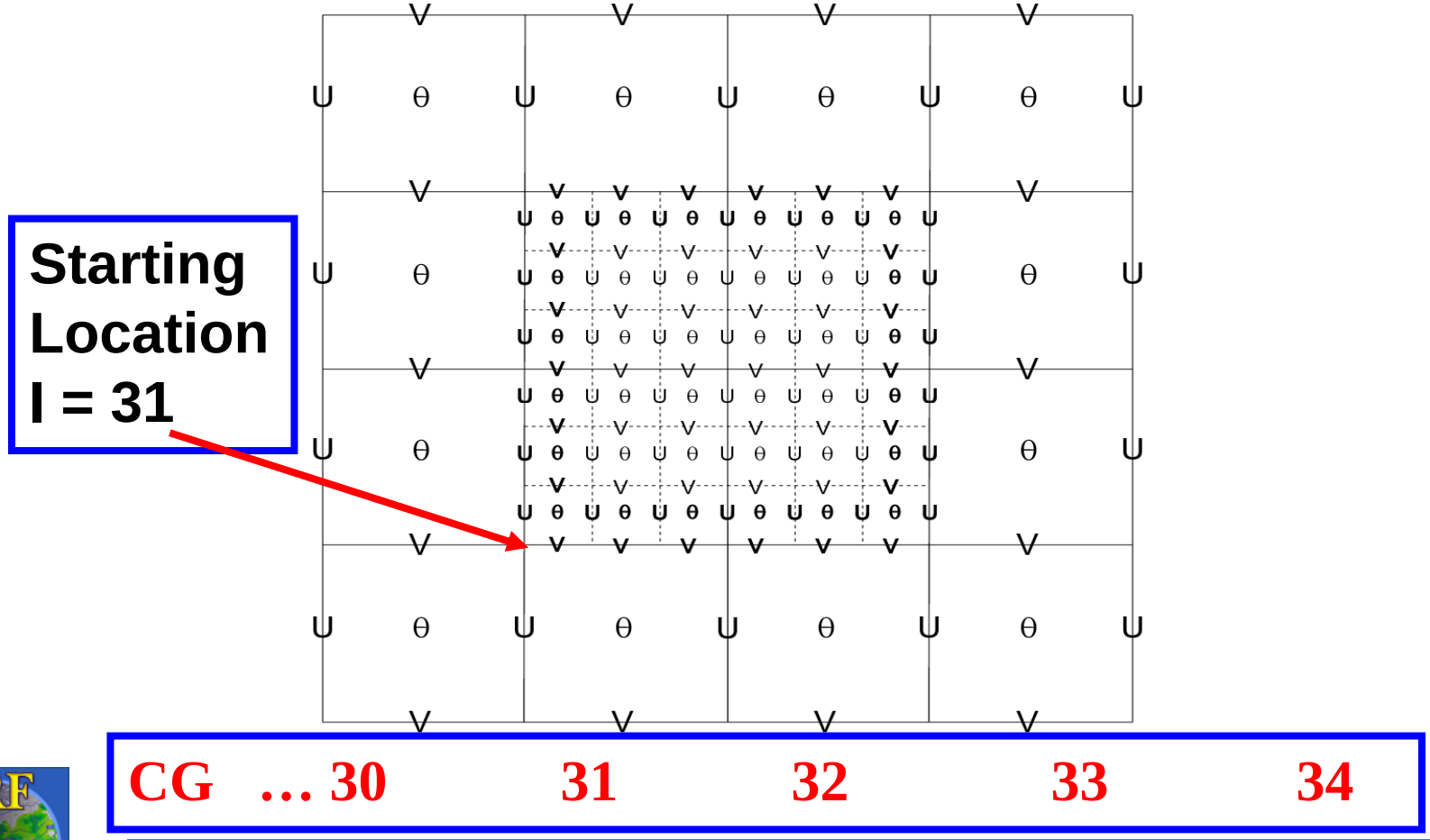
&domains

Domains have only a single parent. This is never allowed.



&domains

Starting locations for nests are the lower left corner, wrt parent.

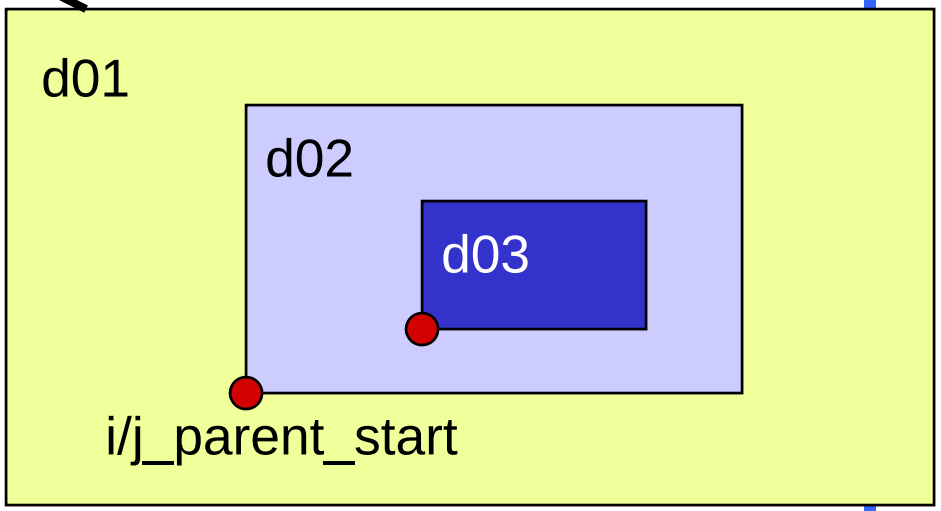


&domains

```
max_dom   = 3,
e_we      = 74, 112,94,
e_sn      = 61, 97, 91,
e_vert    = 28, 28, 28,
grid_id         = 1,  2,  3,
parent_id       = 0,  1,  2,
i_parent_start = 0, 31, 30,
j_parent_start = 0, 17, 30,
```

Activate nests: # of domains to run

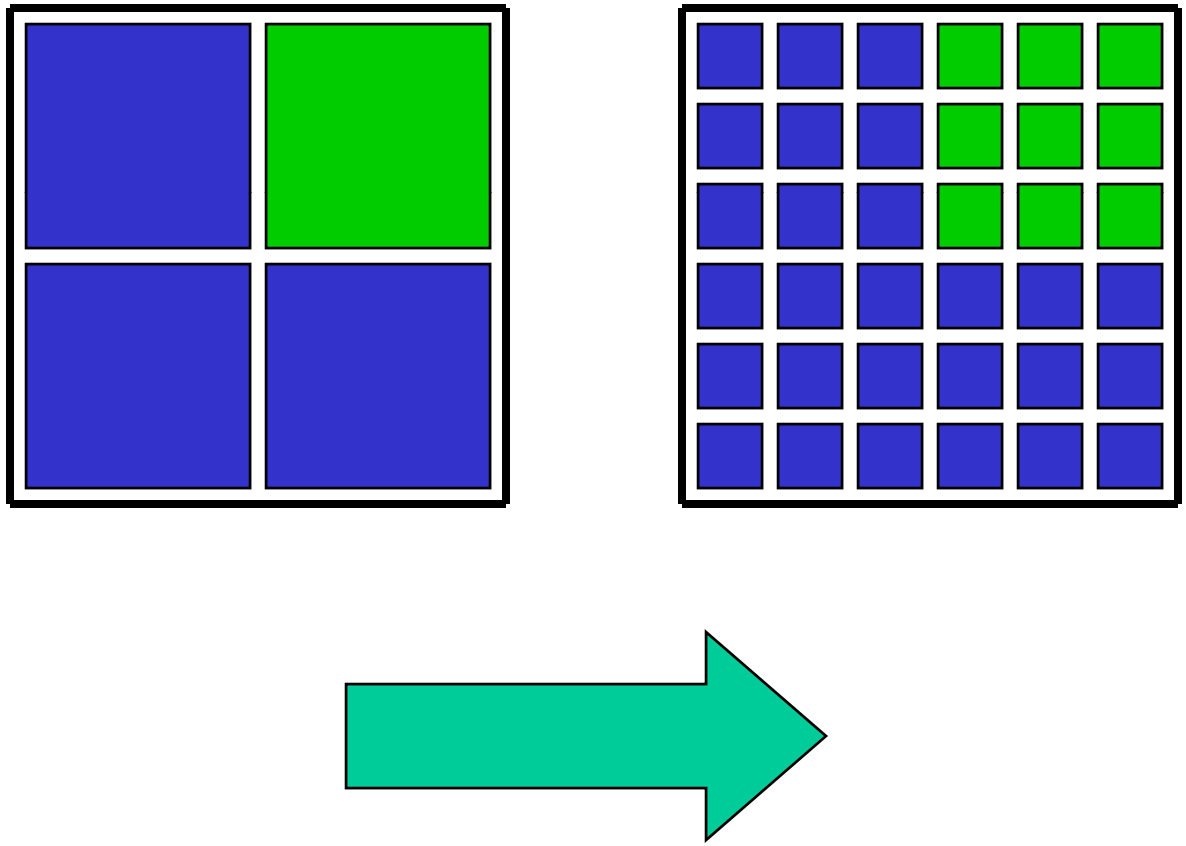
Dimensions of all domains; **same as in WPS**.



Make sure the nest domain parameters match those defined in WPS

&domains

Mesh refinement, 3:1 ratio



&domains

```
dx = 30000, 10000, 3333.33,
dy = 30000, 10000, 3333.33,
parent_grid_ratio = 1, 3, 3,
parent_time_step_ratio = 1,3,3,
```

All 4 variables must be specified.

- *grid ratio* can be any integer
- *time step ratio* can be different from grid ratio
- Grid distance is in meters, even for lat/lon map projection
- For rotated lat/lon domains, dx does not necessarily = dy

&domains

```
feedback        = 1,
smooth_option = 0,
```

When feedback is on, this option can be selected to smooth the area in the parent domain where nest is. Valid values are 0,1,2.

Whether nest will overwrite parent domain results. Setting *feedback*=0 ➔ 'one-way' nesting in a concurrent run.

&bdy_control

```
spec_bdy_width = 5,
spec_zone      = 1,
relax_zone     = 4,
specified      = .T.,.F.,.F.,
nested         = .F.,.T.,.T.,
```

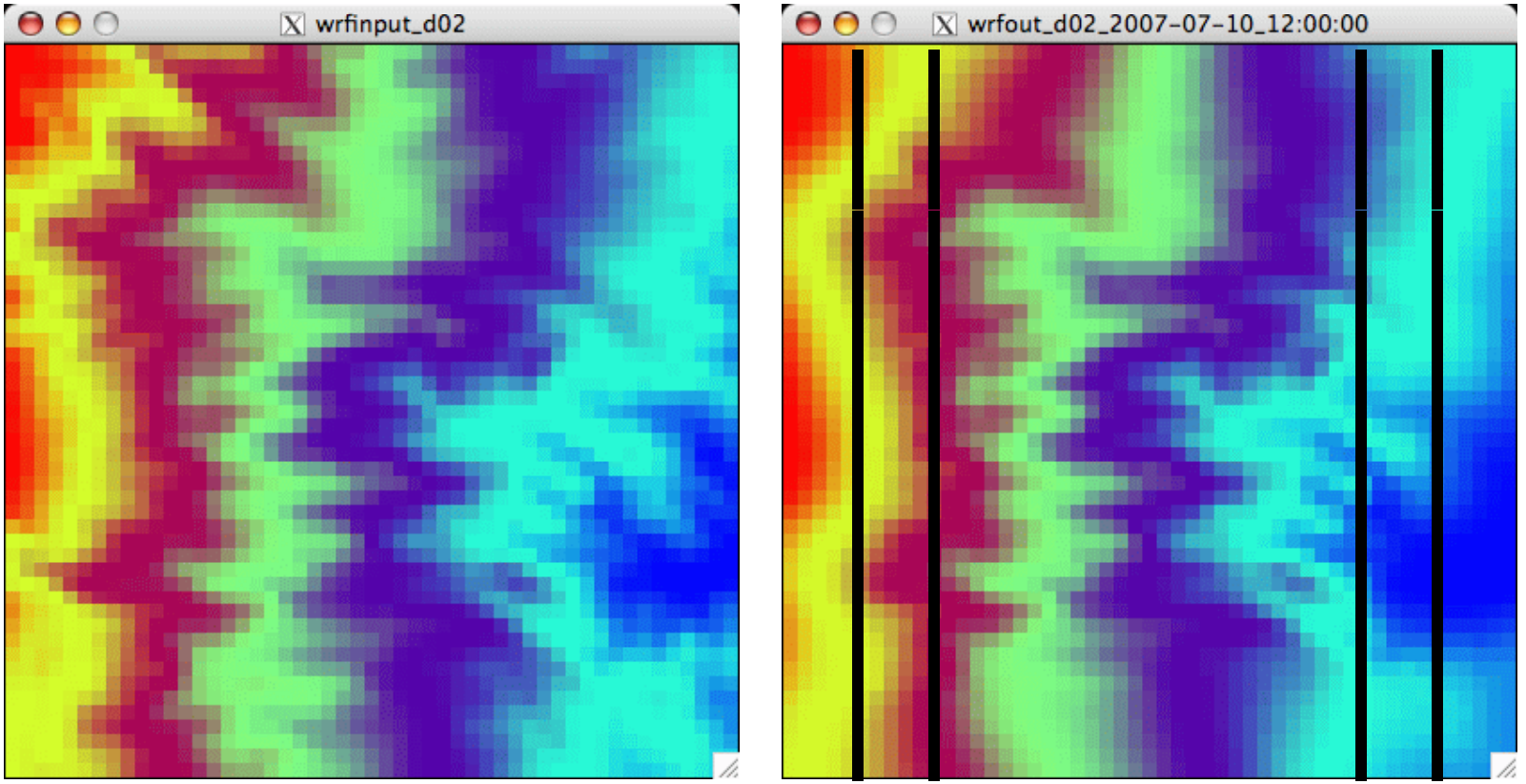
Boundary condition option for domain 1.

Boundary condition option for nests.

May change *relax_zone* and *spec_bdy_width* for ARW; sum must always be spec_bdy_width

&bdy_control

Lateral boundary smoothing, for parent/child consistency.



Other notes on namelists

- Use same physics options for all domains.
  - An exception is cumulus scheme. One may need to turn it off for a nest that has grid distance of a few kilometers or less.
- Also use same physics calling frequency (e.g. **`radt, cudt,`** etc.) in all domains.

Where do I start?

- Always start with a *namelist* template provided in a test case directory
- Not all namelists variables are function of domains. If in doubt, check Registry.EM and registry.io_boilerplate (look for strings ‘rconfig’ and ‘namelist’).
- Use document to guide the modification of the namelist values:
  - run/README.namelist
  - User’s Guide, Chapter 5

Registry.EM

```
rconfig integer spec_bdy_width namelist,bdy_control     1          5
rconfig integer spec zone      namelist,bdy control     1          1
rconfig integer relax_zone     namelist,bdy_control     1          4
rconfig logical specified      namelist,bdy_control max_domains .false.
```

Available types for namelist values: real, integer, logical, character

Typical entries for namelist are either “1” (for all domains) or “max_domains” (where each column/entry is for a particular domain)

Running ARW Nested Case

Running WRF ARW Nested Cases

- Files available from WPS:

  met_em.d01.<*date*>

  met_em.d02.<*date*> (at least one time)

  …

- Link or copy WPS output files to the run directory:

  cd test/*em_real*

  ln -sf ../../../WPS/met_em.* .

Running WRF ARW Nested Cases

- Edit **`namelist.input`** file for runtime options (set **`max_dom >= 2`** in **`&domains`** for a nested run)

- Run the real-data initialization program:

  `./real.exe,` if compiled serially / SMP, or

  `mpirun -np N ./real.exe,` for an MPI job

  where ***N*** is the number of processors requested

Running WRF ARW Nested Cases

- Successfully running this program will create model initial and boundary files:

wrfinput_d01
wrfinput_d02
wrfbdy_d01

Single time level data at model's start time for all domains

Multiple time level data at the lateral boundary, only for domain 1

Running WRF ARW Nested Cases

- Run the model executable by typing:

  `./wrf.exe >& wrf.out &`

  or

  `mpirun -np N ./wrf.exe &`

- Successfully running the model will create model *history* files, one for each domain:

  `wrfout_d01_2005-08-28_00:00:00`

  `wrfout_d02_2005-08-28_00:00:00`

  And *restart* files if selected:

  `wrfrst_d01_<date>, wrfrst_d02_<date>`

Moving Nest Case (ARW only)

- The main reason for using this option is to run the model economically.
- Must choose correct compile options when creating **`configure.wrf`** file
  - Choose preset move, or vortex following
- Other options are controlled by the namelists.
- Can do specified move, and automatic vortex tracking (for tropical cyclone application).
- All nest domains can move.

Specified Moving Case

- Namelists in **`&domains`**:

  `num_moves, move_id, move_interval,`
  `move_cd_x, move_cd_y, corral_dist`

  ➔ only one-grid-cell move at a time
- Must specify initial nest location

Automatic Moving Case

- Tropical cyclone applications only.
- Works better for well developed storms.
- Namelists in **`&domains`**:

  `vortex_interval` (default 15 min)

  `max_vortex_speed` (default 40 m/s)

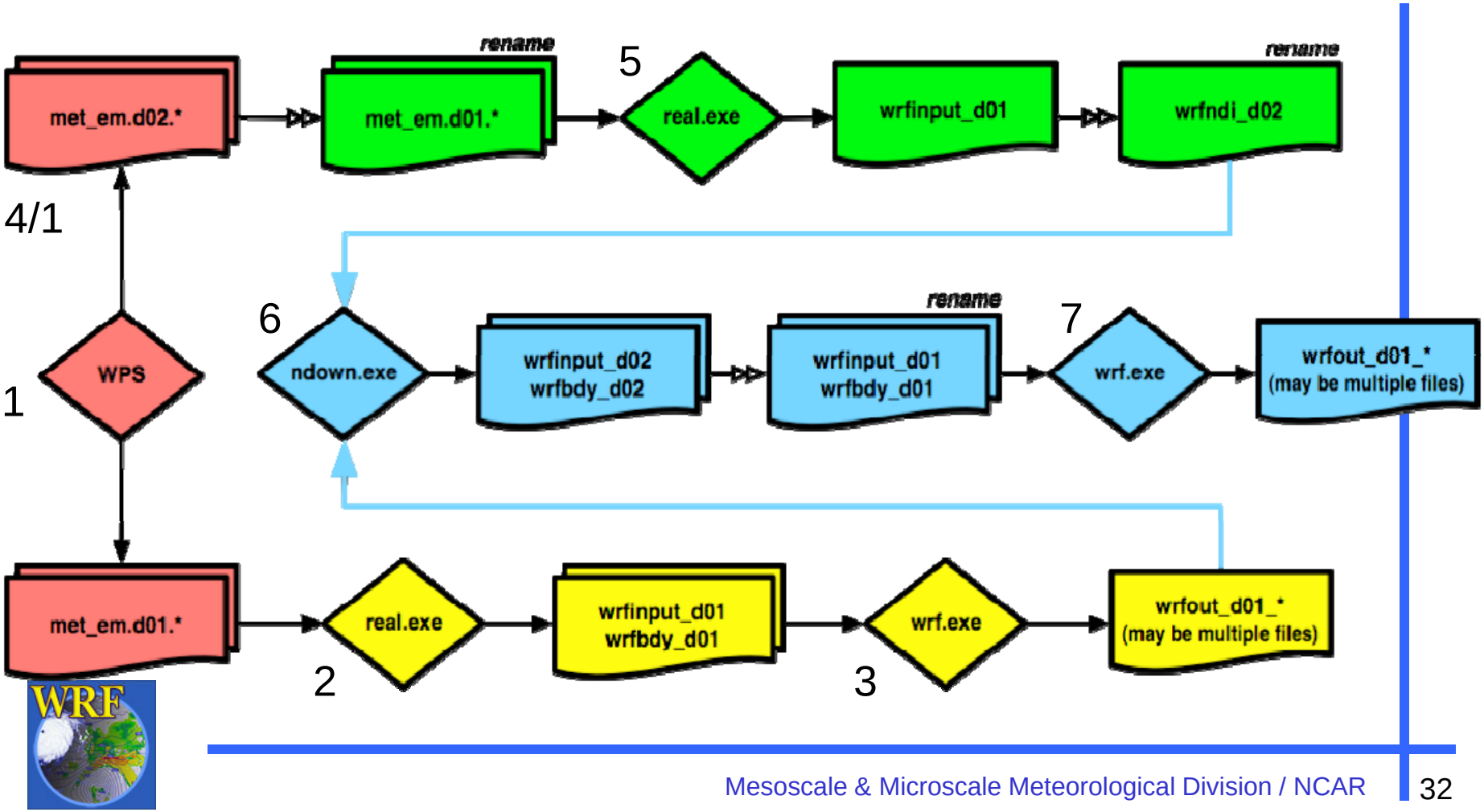
  `corral_dist` (default 8 coarse grid cells)

  `track_level` (default 50000 Pa)

  `time_to_move` (default is 0 h for all nests)
- Must specify initial nest location

One-way Nesting: Two separate runs

ARW only



Summary

- ARW:
  - Two-way, without nest input files (**`input_from_file=.f`**.)
  - Two-way, with nest input files (**`input_from_file = .t`**.)
  - Two-way, with static nest input only (**`input_from_file=.t.,`** **`fine_input_stream = 2`**)
  - One-way, concurrent run (**`feedback = 0`**)
  - One-way, separate runs (treated like two single domain runs, with ***ndown***)
  - Two-way, specified moving nest run
  - Two-way, automatic vortex tracking run

References

- Information on compiling and running WRF with nests, and a more extensive list of namelist options and their definition / explanations can be found in the ARW User's Guide, Chapter 5
- Start with namelist templates in test/ directory