NESTING IN WRF

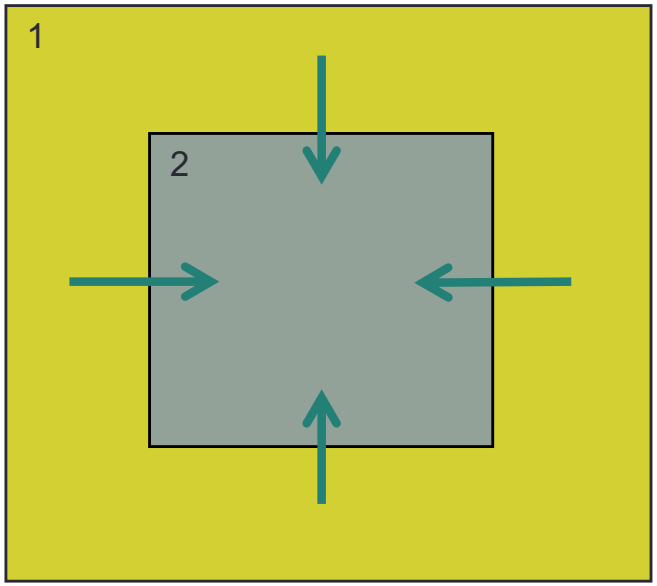
Kelly Werner

January 2020

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

What is a nest?

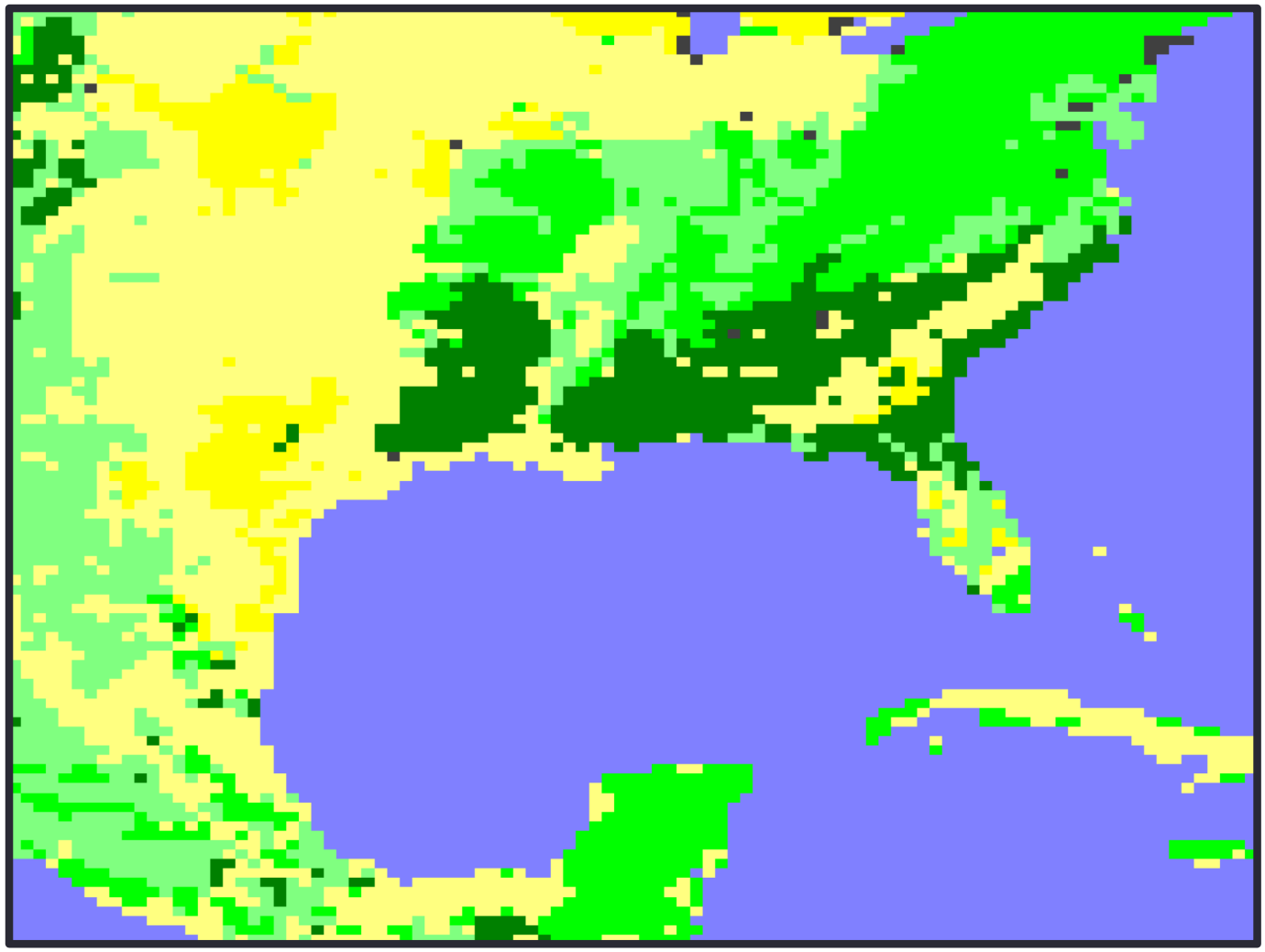
- A finer-resolution domain embedded in a coarser-resolution domain, and run together with the coarse domain
- Driven along its lateral boundaries by the parent domain
- May feedback the computed values back to the parent domain



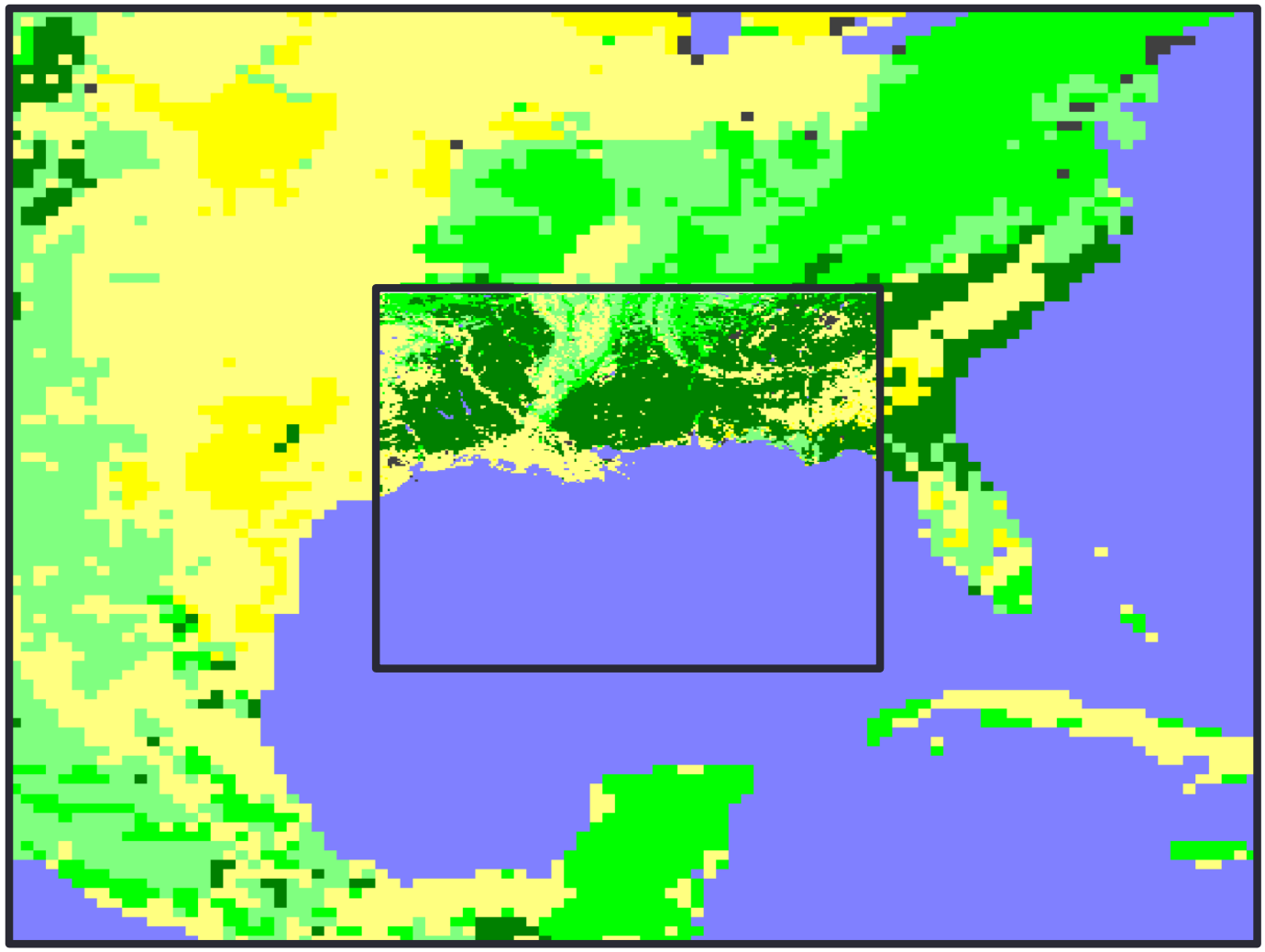
When Should I Use Nests?

- Need to simulate localized phenomena: convection, topography, landuse-forced, etc.
  - Input data resolution is too coarse by more than a factor of 5-10x
  - Computing resources not available for uniform coverage
- What resolution is necessary to resolve what you are interested in?

Coarse landuse over large area



Nested fine-resolution landuse



Types of Nesting

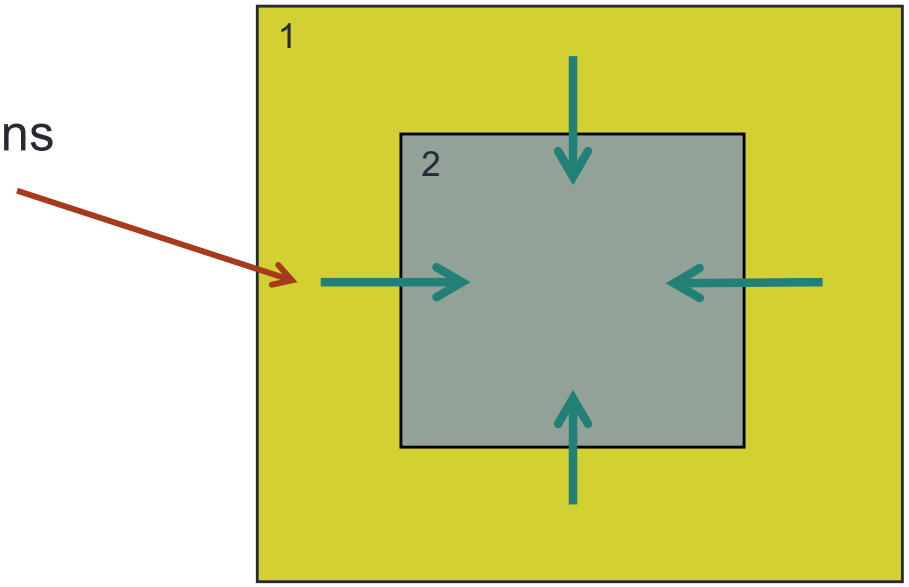
- Using a single input domain (met_em.d01*)
  - No met_em.d02* files are used
  - All fields are interpolated from the model coarse grid
  - Only recommended if nest is over the ocean
- Using multiple input domains
  - Each domain contains full input data files (including topography, landuse, etc.)
- Specified move
  - Build WRF with “2=preset moves”
  - Must specify every move
  - Can use, but tedious to set-up
- Automatic move
  - Build WRF with “3=vortex following”
  - Only for tropical cyclone tracking
  - Expensive for single large nest
- ndown.exe
  - Use coarser WRF model output to drive finer resolution domains (i.e. ‘downscaling’)
  - If you have run a long coarse domain simulation (years) and later decide you want to have a nest with higher resolution.
  - If using several nests and domain size for the fine-resolution domain is much different than resolution for coarse domain

Types of Nesting

One-way/two-way nesting

- Determined by the namelist parameter “feedback”
  - **feedback = 0 (turned off/one-way)**

Lateral boundary conditions are fed to the nest, from the parent.



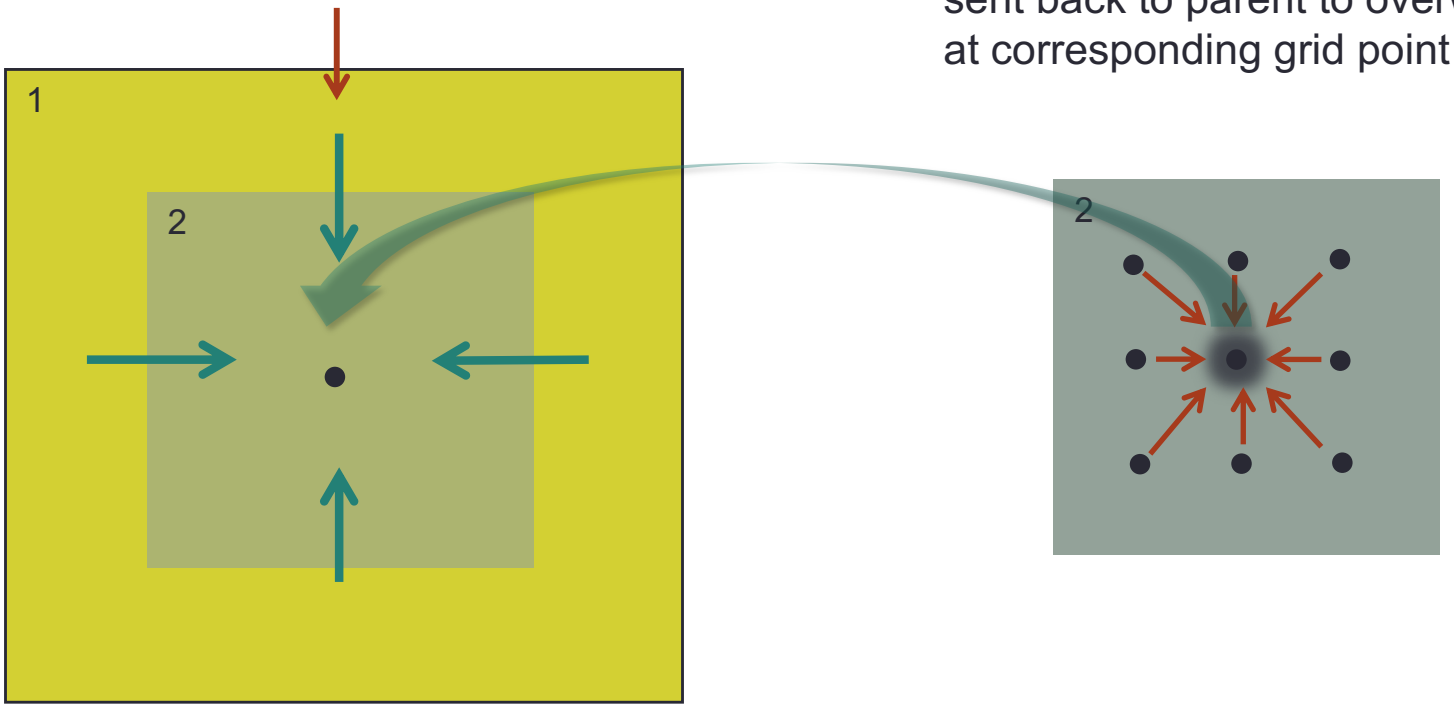
Types of Nesting

One-way/**two-way nesting**

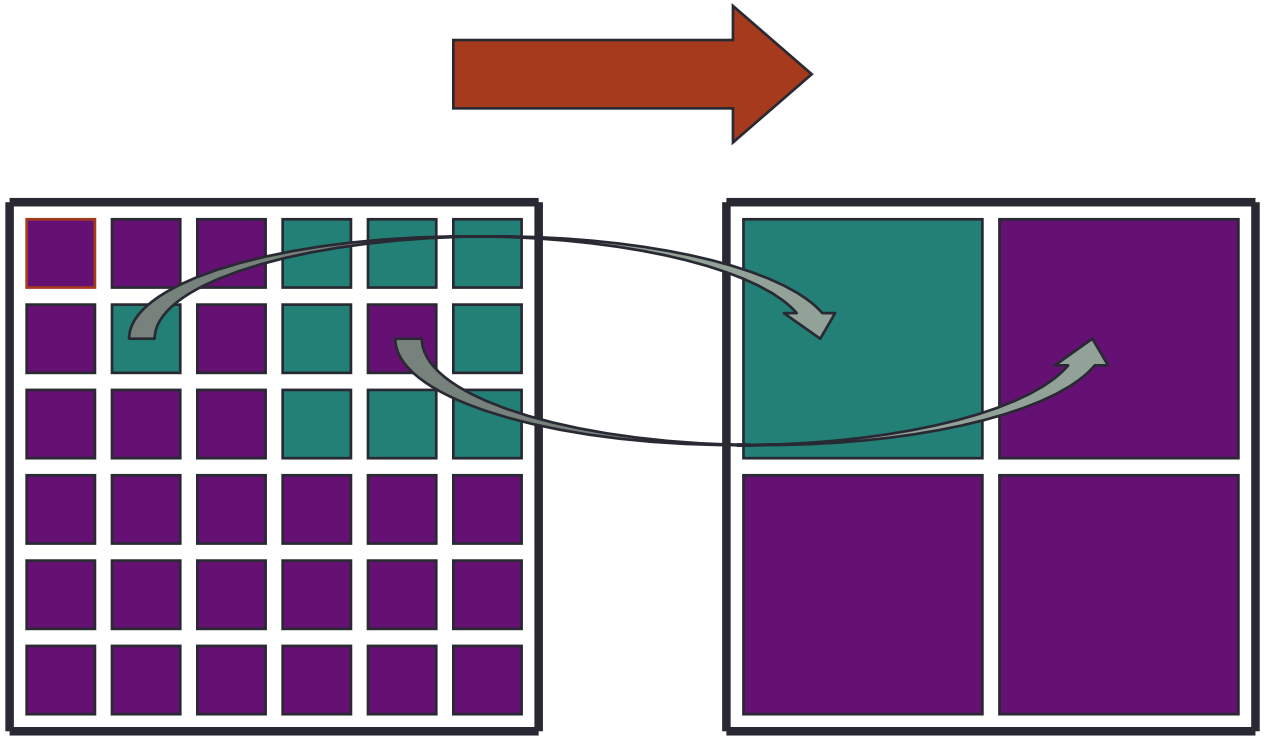
- Determined by the namelist parameter “feedback”
  - **feedback = 1 (turned on/two-way)**

(1) Lateral boundary conditions are fed to the nest, from the parent.

(2) Child values are averaged, and then sent back to parent to overwrite value at corresponding grid point

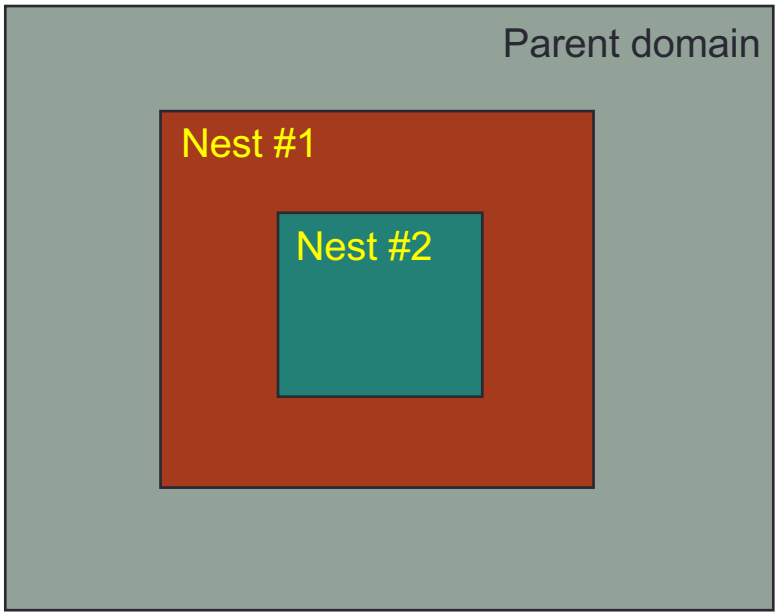


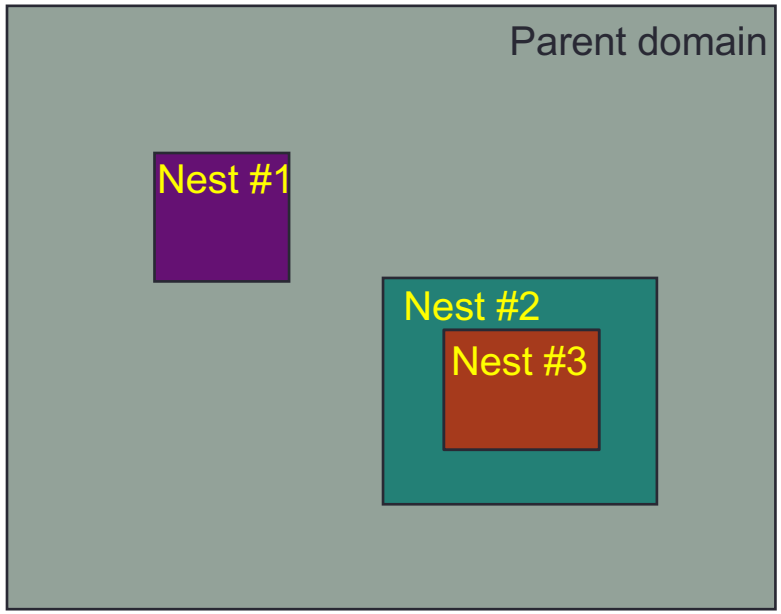
Masked Feedback



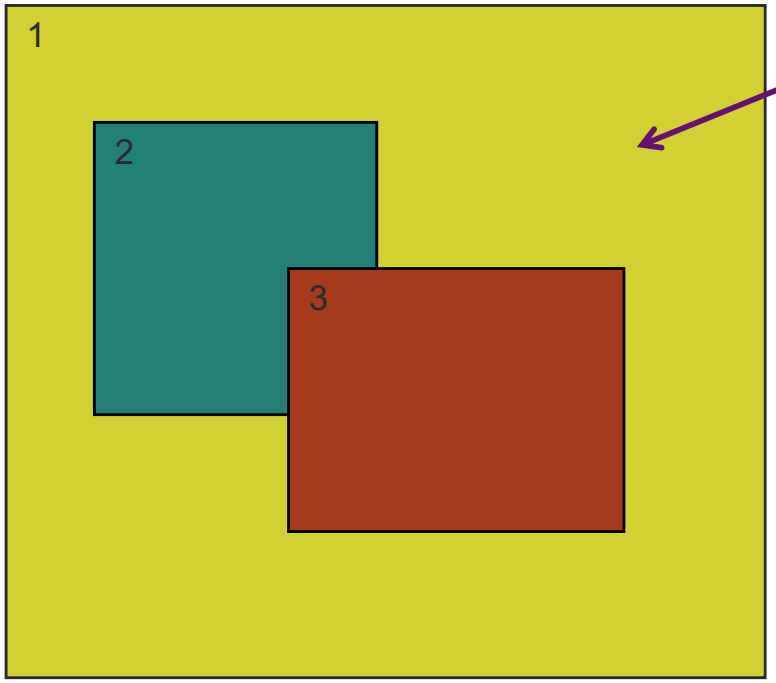
Single grid value feedback for categorical and masked data

Nests that are OK



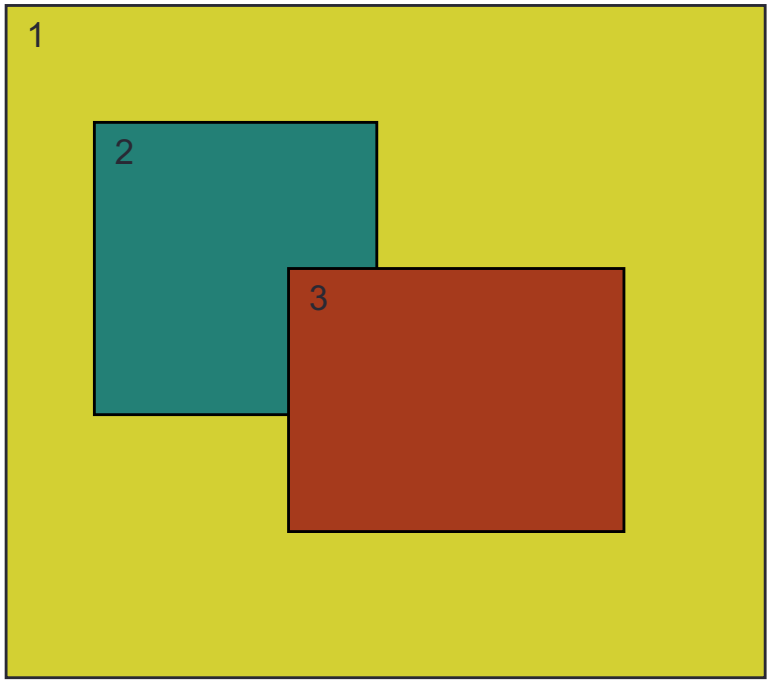


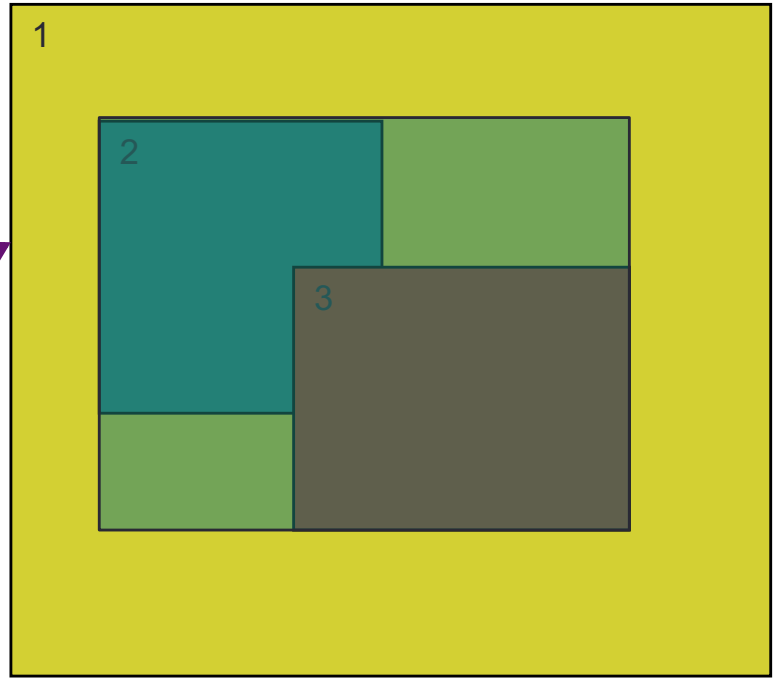
Nests that are NOT OK



Child domains *may not* have overlapping points in the parent domain (possible if Feedback is off).

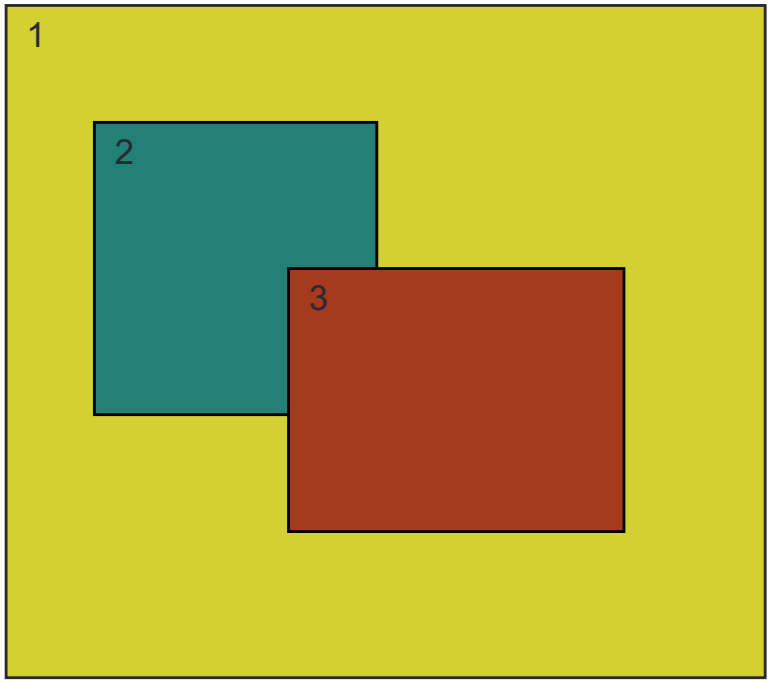
Nests that are NOT OK

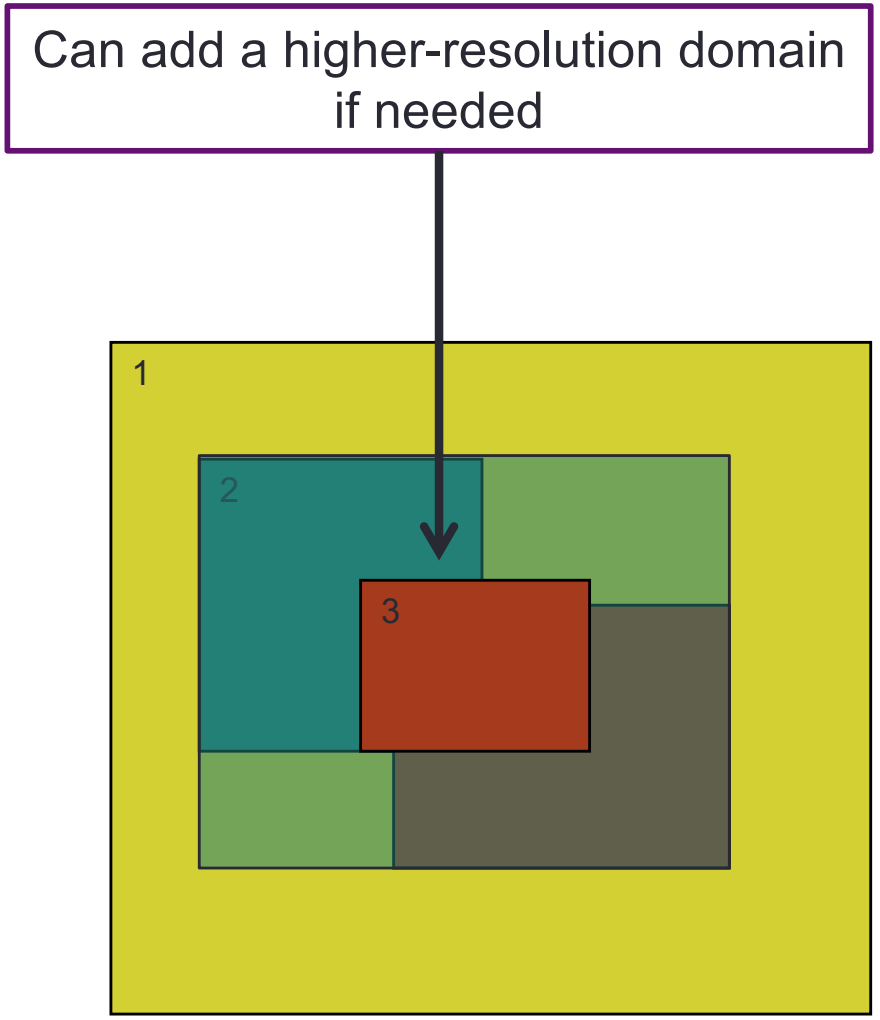




It's best to combine domains to create a single large fine-resolution nested domain

Nests that are NOT OK

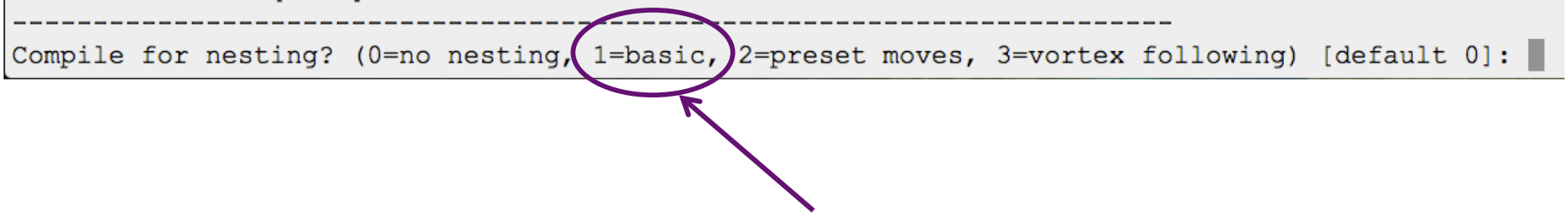




Nesting Set-up and Run

Compiling for Nesting (WRF)

```
------------------------------------------------------------------------
Please select from among the following Darwin ARCH options:

  1. (serial)   2. (smpar)   3. (dmpar)   4. (dm+sm)   PGI (pgf90/pgcc)
  5. (serial)   6. (smpar)   7. (dmpar)   8. (dm+sm)   INTEL (ifort/icc)
  9. (serial)  10. (smpar)  11. (dmpar)  12. (dm+sm)   INTEL (ifort/clang)
 13. (serial)                14. (dmpar)               GNU (g95/gcc)
 15. (serial)  16. (smpar)  17. (dmpar)  18. (dm+sm)   GNU (gfortran/gcc)
 19. (serial)  20. (smpar)  21. (dmpar)  22. (dm+sm)   GNU (gfortran/clang)
 23. (serial)                24. (dmpar)               IBM (xlf90_r/cc)
 25. (serial)  26. (smpar)  27. (dmpar)  28. (dm+sm)   PGI (pgf90/pgcc): -f90=pgf90

Enter selection [1-28] : 9
------------------------------------------------------------------------
Compile for nesting? (0=no nesting, 1=basic, 2=preset moves, 3=vortex following) [default 0]:
```



Compile with nesting option (1=basic)

*Note: Unless compiling for a moving nest, or 2D idealized case, there's no reason to not always choose "basic." It takes no longer to build.

namelist.wps - WPS

namelist.wps set-up: &share

To edit the namelist.wps file, make sure you are in the WPS/ directory

```
&share
   wrf_core = 'ARW',
   max_dom = 2,
   start_date = '2012-01-27_00:00:00', 2012-01-27_00:00:00'
   end_date = '2012-01-28_00:00:00', 2012-01-27_00:00:00'
   interval_seconds = 21600
   io_form_geogrid = 2,
/
```

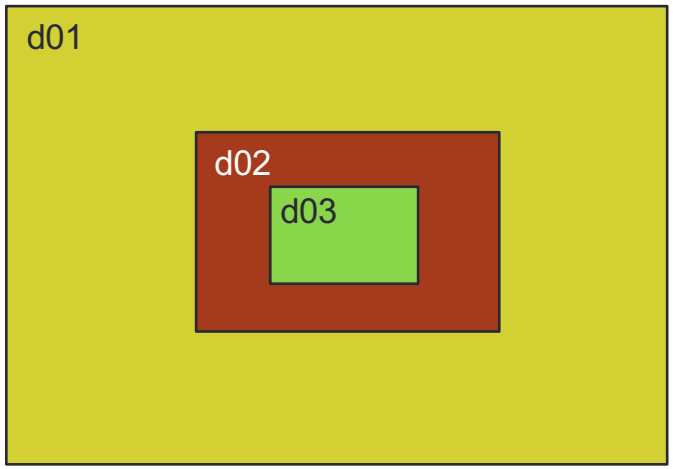
real.exe only requires initial time for fine domain (unless doing nudging or SST-update in the nest)

Make sure to edit start/end dates for all domains!

namelist.wps set-up: &geogrid

```
&geogrid

   parent_id             =   1,        1,
   parent_grid_ratio     =   1,        3,
   i_parent_start        =   1,       70,
   j_parent_start        =   1,       67,

   e_we                  =  175,     181,
   e_sn                  =  145,     181,
   geog_data_res         =  'default', 'default',


   dx                    = 30000,
   dy                    = 30000,
   map_proj     = 'lambert',
   ref_lat      = 37.0,
   ref_lon      = -97.0,
   truelat1     = 45.0,
   truelat2     = 30.0,
   stand_lon    = -97.0,
   geog_data_path = '/data/static/geog/'
/
```

parent_id:
The domain # of the nest's parent

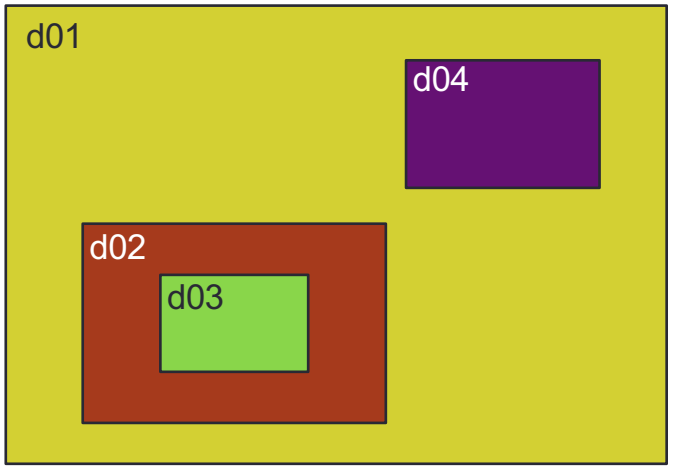


parent_id = 1, 1, 2

namelist.wps set-up: &geogrid

```
&geogrid

   parent_id             =   1,      1,
   parent_grid_ratio     =   1,      3,
   i_parent_start        =   1,     70,
   j_parent_start        =   1,     67,

   e_we                  =  175,   181,
   e_sn                  =  145,   181,
   geog_data_res         =  'default', 'default',


   dx                    = 30000,
   dy                    = 30000,
   map_proj      = 'lambert',
   ref_lat       = 37.0,
   ref_lon       = -97.0,
   truelat1      = 45.0,
   truelat2      = 30.0,
   stand_lon     = -97.0,
   geog_data_path = '/data/static/geog/'
/
```

parent_id:
The domain # of the nest's parent



parent_id = 1, 1, 2, 1

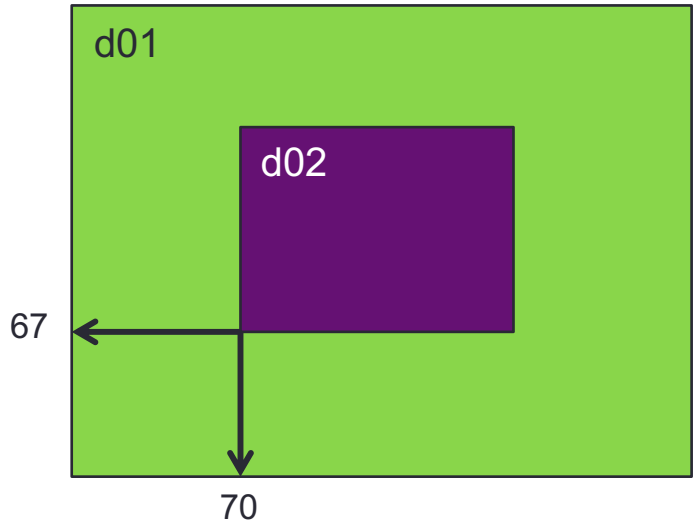
namelist.wps set-up: &geogrid

```
&geogrid

   parent_id           =   1,        1,
   parent_grid_ratio   =   1,        3,
   i_parent_start      =   1,       70,
   j_parent_start      =   1,       67,

   e_we                =  175,     181,
   e_sn                =  145,     181,
   geog_data_res       =  'default', 'default',


   dx                  = 30000,
   dy                  = 30000,
   map_proj     = 'lambert',
   ref_lat      = 37.0,
   ref_lon      = -97.0,
   truelat1     = 45.0,
   truelat2     = 30.0,
   stand_lon    = -97.0,
   geog_data_path = '/data/static/geog/'
/
```

parent_grid_ratio: The grid resolution ratio of the child to its parent (must be an integer).

recommended ratios are 3:1 or 5:1 (odd ratios, less than 7)

i/j_parent_start:

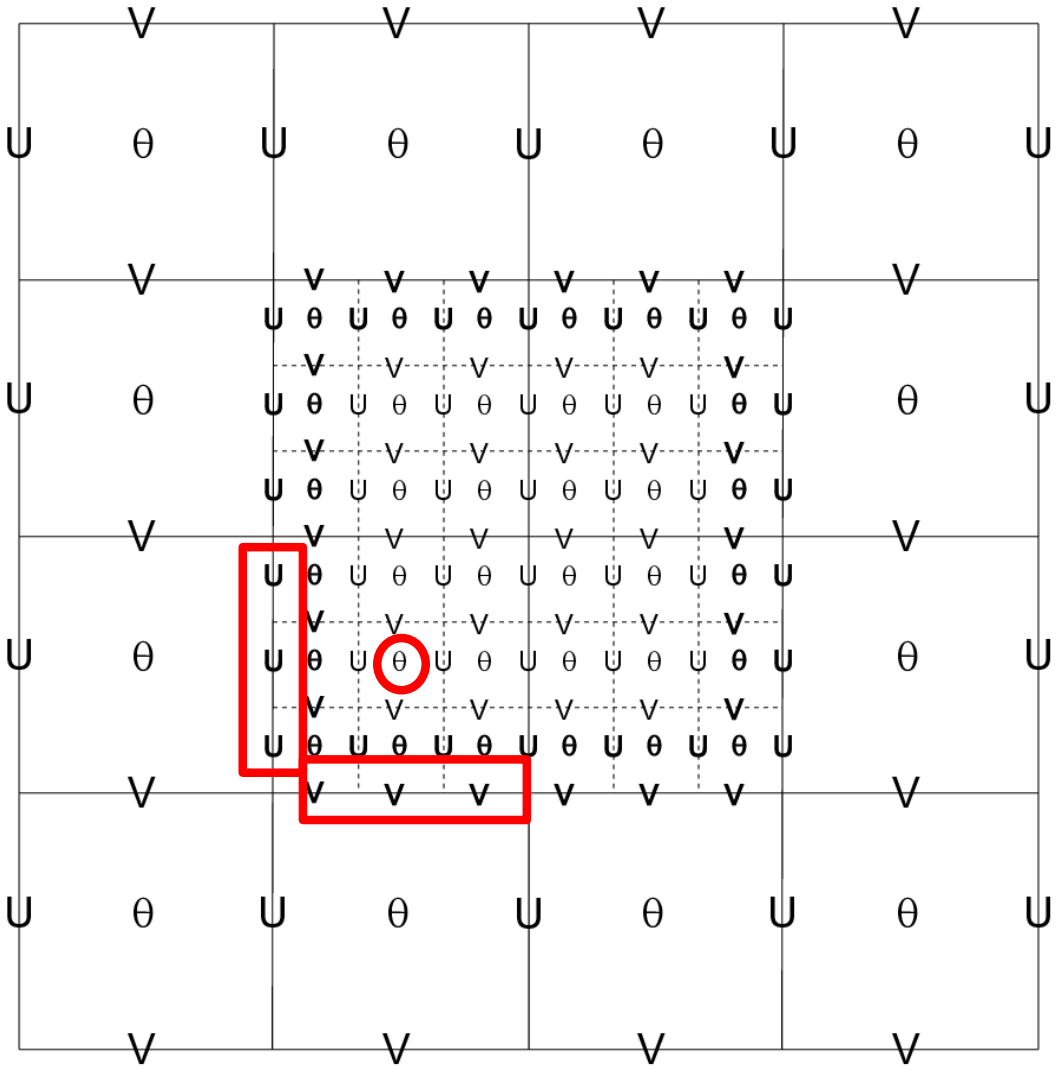


Feedback 3:1 Ratio

When using feedback, conditions are fed back to the parent domain from the child along the rows and columns, and at the mass points (center)

U: east-west velocities
V: south-north velocities
Θ: all other meteorological data

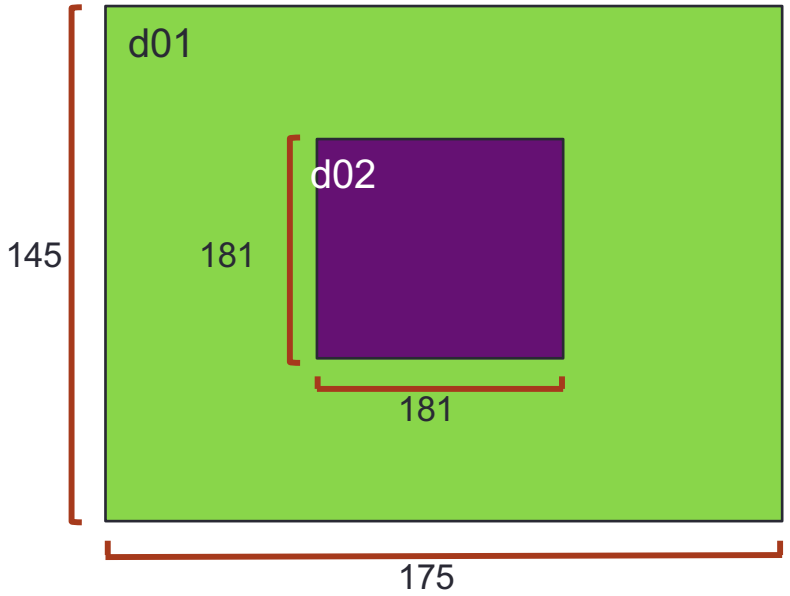
➡ Averaging is performed



namelist.wps set-up: &geogrid

```
&geogrid

 parent_id             =   1,        1,
 parent_grid_ratio     =   1,        3,
 i_parent_start        =   1,       70,
 j_parent_start        =   1,       67,

 e_we                  =  175,     181,
 e_sn                  =  145,     181,
 geog_data_res         =  'default', 'default',


 dx                    = 30000,
 dy                    = 30000,
 map_proj     = 'lambert',
 ref_lat      = 37.0,
 ref_lon      = -97.0,
 truelat1     = 45.0,
 truelat2     = 30.0,
 stand_lon    = -97.0,
 geog_data_path = '/data/static/geog/'
/
```

e_we and e_sn:
Each domain's full west-east and south-north dimensions



Notes:

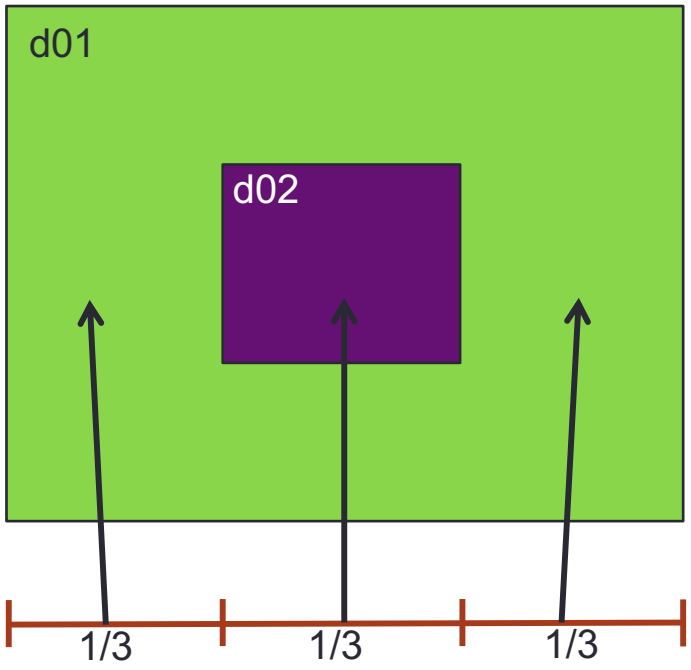
- Domains should be > 100x100
- Avoid placing any boundaries over complex terrain ***(if possible)***
- Keep nest away from coarse domain

namelist.wps set-up: &geogrid

```
&geogrid

 parent_id             =   1,        1,
 parent_grid_ratio     =   1,        3,
 i_parent_start        =   1,       70,
 j_parent_start        =   1,       67,

 e_we                  =  175,     181,
 e_sn                  =  145,     181,
 geog_data_res         =  'default', 'default',


 dx                    = 30000,
 dy                    = 30000,
 map_proj     = 'lambert',
 ref_lat      = 37.0,
 ref_lon      = -97.0,
 truelat1     = 45.0,
 truelat2     = 30.0,
 stand_lon    = -97.0,
 geog_data_path = '/data/static/geog/'
/
```

Minimum distance between nest boundary and parent boundary:

- 4 grid cells (for communication)
- need MUCH larger buffer zone



- Good practice to have ~1/3 of coarse-grid surrounding each side of nest

namelist.wps set-up: &geogrid

```
&geogrid

 parent_id             =   1,        1,
 parent_grid_ratio     =   1,        3,
 i_parent_start        =   1,       70,
 j_parent_start        =   1,       67,

 e_we                  =  175,     181,
 e_sn                  =  145,     181,
 geog_data_res         =  'default', 'default',


 dx                    = 30000,
 dy                    = 30000,
 map_proj     = 'lambert',
 ref_lat      = 37.0,
 ref_lon      = -97.0,
 truelat1     = 45.0,
 truelat2     = 30.0,
 stand_lon    = -97.0,
 geog_data_path = '/data/static/geog/'
/
```

dx and dy:
Only need the coarse domain resolution. The geogrid program calculates the nest resolution(s) using the "parent_grid_ratio"

Note
Do not add values for additional columns if default namelist does not have values in more than d01 column.

Note:
No changes need to be made to the &ungrib and &metgrid namelists records for nesting purposes

namelist.input (WRF)

namelist.input set-up: &time_control

```
&time_control
 run_days                 = 0,
 run_hours                = 24,
 run_minutes              = 0,
 run_seconds              = 0,
 start_year               = 2012,  2012,   2012,
 start_month              = 01,    01,     01,
 start_day                = 27,    27,     27,
 start_hour               = 00,    00,     00,
 end_year                 = 2012,  2012,   2012,
 end_month                = 01,    01,     01,
 end_day                  = 28,    28,     28,
 end_hour                 = 00,    00,     00,
 interval_seconds         = 10800
 input_from_file          = .true.,.true.,  .true.,
 history_interval         = 360,    60,     60
 frames_per_outfile       = 1000,   1,       1
 restart                  =  .false.
 restart_interval         = 180
 io_form_history          = 2
 io_form_restart          = 2
/
```

** To edit the namelist.input file, make sure you are in the *WRF/test/em_real/* (or *WRF/run/*) directory

start/end date/times:
These values *typically* will be the same for all domains

history_interval:
How often history is written out.
*May choose to have more frequent output time for nests

frames_per_outfile:
The number of history intervals in a single file.

Time and Frequency Clarification

```
&time_control
 interval_seconds        = 10800
 history_interval        =  60,     60,      60
 frames_per_outfile      = 1,       1,       1
/

&domains
 time_step               = 180
/
```

interval_seconds:
The number of seconds between each data input file (met_em*)
10800 = 3-hourly input data

time_step:
How often the model integrates forward (in seconds). 180 = every 3 mins of simulation time

frames_per_outfile:
The number of history intervals in a single file.
In our example above, if it's set to 1, you should get a wrfout file for each simulation hour.

history_interval:
Frequency (in simulation minutes) that data is written/recorded.
= 60: history is recorded every 1 hour.

Since time_step=180, each history recording includes 20 time steps of integration.

namelist.input set-up: &domains

```
&domains
 time_step                  = 180,
 time_step_fract_num        = 0,
 time_step_fract_den        = 1,
 max_dom                    = 2,
 e_we                       = 175,   181,   94,
 e_sn                       = 145,   181,   91,
 e_vert                     = 40,    40,    40,
 p_top_requested            = 5000,
 num_metgrid_levels         = 32,
 num_metgrid_soil_levels    = 4,
 dx                         = 30000, 10000,  3333.33,
 dy                         = 30000, 10000,  3333.33,
 grid_id                    = 1,     2,     3,
 parent_id                  = 1,     1,     2,
 i_parent_start             = 1,     70,    30,
 j_parent_start             = 1,     67,    30,
 parent_grid_ratio          = 1,     3,     3,
 parent_time_step_ratio     = 1,     3,     3,
 feedback                   = 1,
/
```

max_dom:
Activate nests - # of domains to run

e_we and e_sn:
should match namelist.wps values

e_vert:
All columns usually have the same value

dx/dy:
must set values for each domain.
make sure values correspond with "parent_grid_ratio"
- for non-integer grid resolutions, use at least two decimal places

**** Beginning V4.2 – only need d01****

namelist.input set-up: &domains

```
&domains
.....
grid_id                             = 1,    2,    3,
parent_id                           = 1,    1,    2,
i_parent_start                      = 1,    70,   30,
j_parent_start                      = 1,    67,   30,
parent_grid_ratio                   = 1,    3,    3,
parent_time_step_ratio              = 1,    3,    3,
feedback                            = 1,
/
```

All must be set to the same values used in namelist.wps

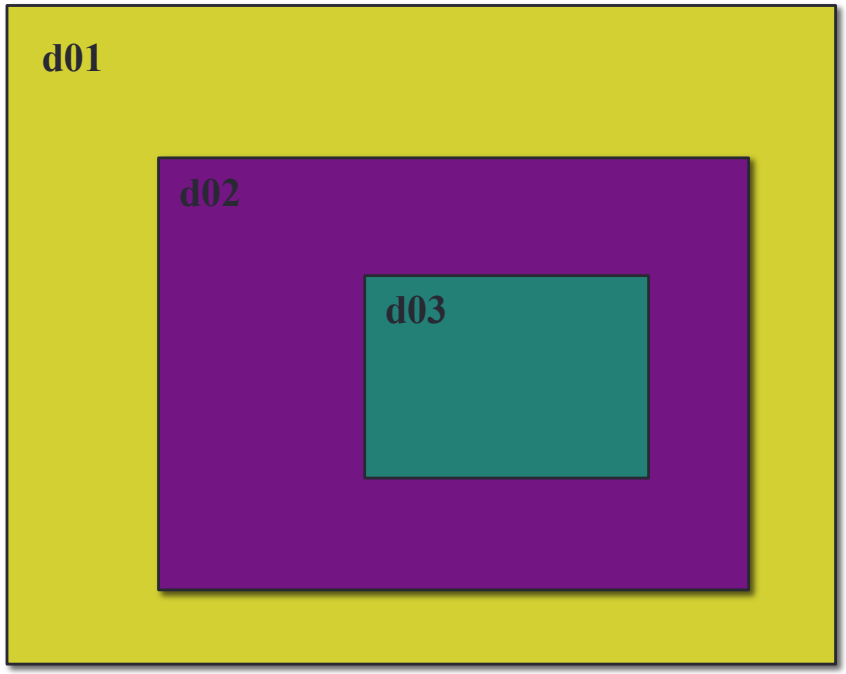
feedback:
Whether a nest will overwrite parent results
- 2-way nesting: feedback = 1
- 1-way nesting: feedback = 0

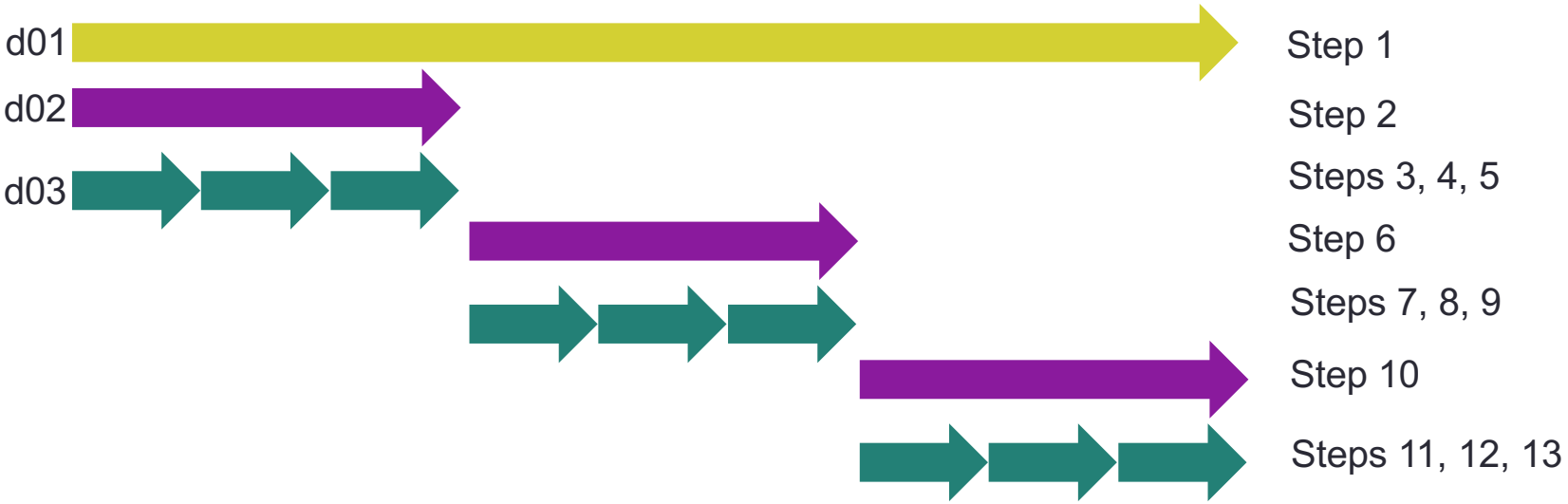
parent_time_step_ratio:
See next slide!

Nested 3:1 Parent Time Step Ratio

- Example: 3-domain nested run
- time_step = 180

  parent_time_step_ratio = 1, 3, 3,
  - D01: a single 3 min dt
  - D02: a single 1 min dt
  - D03: 20 second intervals, up to 1 min





namelist.input set-up: &physics

- You should use the same physics options for all domains for all schemes
  - **Exceptions**:
    - cumulus_scheme (cu_physics): may need to be turned off for a nest that has a grid distance of only a few kilometers
    - may turn off PBL scheme for resolutions close to 100 m

- Use same values for physics calling frequency parameters (for each domain)
  - radt: radiation time step
  - bldt: boundary layer time step
  - cudt: cumulus scheme time step

Computationally inexpensive – no reason to not always set to zero (run every time step)

Where do I start?

- Namelist templates provided in test/em_real

- Helpful documentation:
  - **README.namelist** (in test/em_real/)
  - **examples.namelist** (in test/em_real/)
  - **Users' Guide, Chapter 5**
    - http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_v4/V4.1/users_guide_chap5.htm
  - **Namelist Best Practice web pages:**
    - WPS: http://www2.mmm.ucar.edu/wrf/users/namelist_best_prac_wps.html
    - WRF: http://www2.mmm.ucar.edu/wrf/users/namelist_best_prac_wrf.html

- Not all namelist options are domain dependent. If in doubt:
  - **Registry.EM_COMMON** or **registry.io_boilerplate** (found in WRF/Registry/)
    * grep for parameter names – look for "max_dom" (max_dom means expected value for each domain)
  - Rule of thumb: If default namelist only has 1 column, don't add values for other columns!

Successful Nested Run: WPS

- Modify namelist.wps for multiple domains (additional columns)

- Use same executables for running with a single domain
  - geogrid.exe output: **`geo_em.d01.nc`**, **`geo_em.d02.nc`**, etc.
  - ungrib.exe output: same as single domain – not domain dependent
  - metgrid.exe output: **`met_em.d01*`**, **`met_em.d02*`**, etc.

Successful Nested Run: WRF

- Modify namelist.input for multiple domains (additional columns)

- Link in the met_em* files and issue same executables for running with a single domain

real.exe output:

wrfbdy_d01

- Lateral boundary data for all times (domain 01 only)

wrfinput_d01, wrfinput_d02, etc.

- Single time-level data at the model's start time (for each domain)
- 1 file per domain

wrf.exe output:

wrfout_d01*, wrfout_d02*, etc.

- One for each domain, for each history time (depending on 'frames_per_outfile')

wrfrst_d01*, wrfrst_d02*, etc.

- If "restart_interval" is less than or equal to the integration time

Summary

- Decide the best strategy to run your simulation
  - Based on resolution needed to resolve phenomenon, vs. resolution of input data
- If nesting is required, design your nest configuration
  - Design the coarse domain first
  - Determine the beginning and ending indices of the nest on the coarse domain
- Choose the appropriate nesting strategy:
  - one-way, two-way, or one-way via *ndown*

Questions?