NESTING IN WRF

Kelly Werner

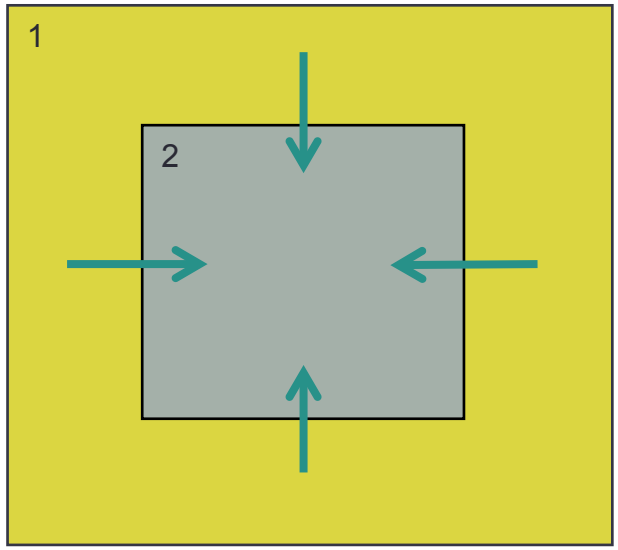
July 2017

What is a nest?

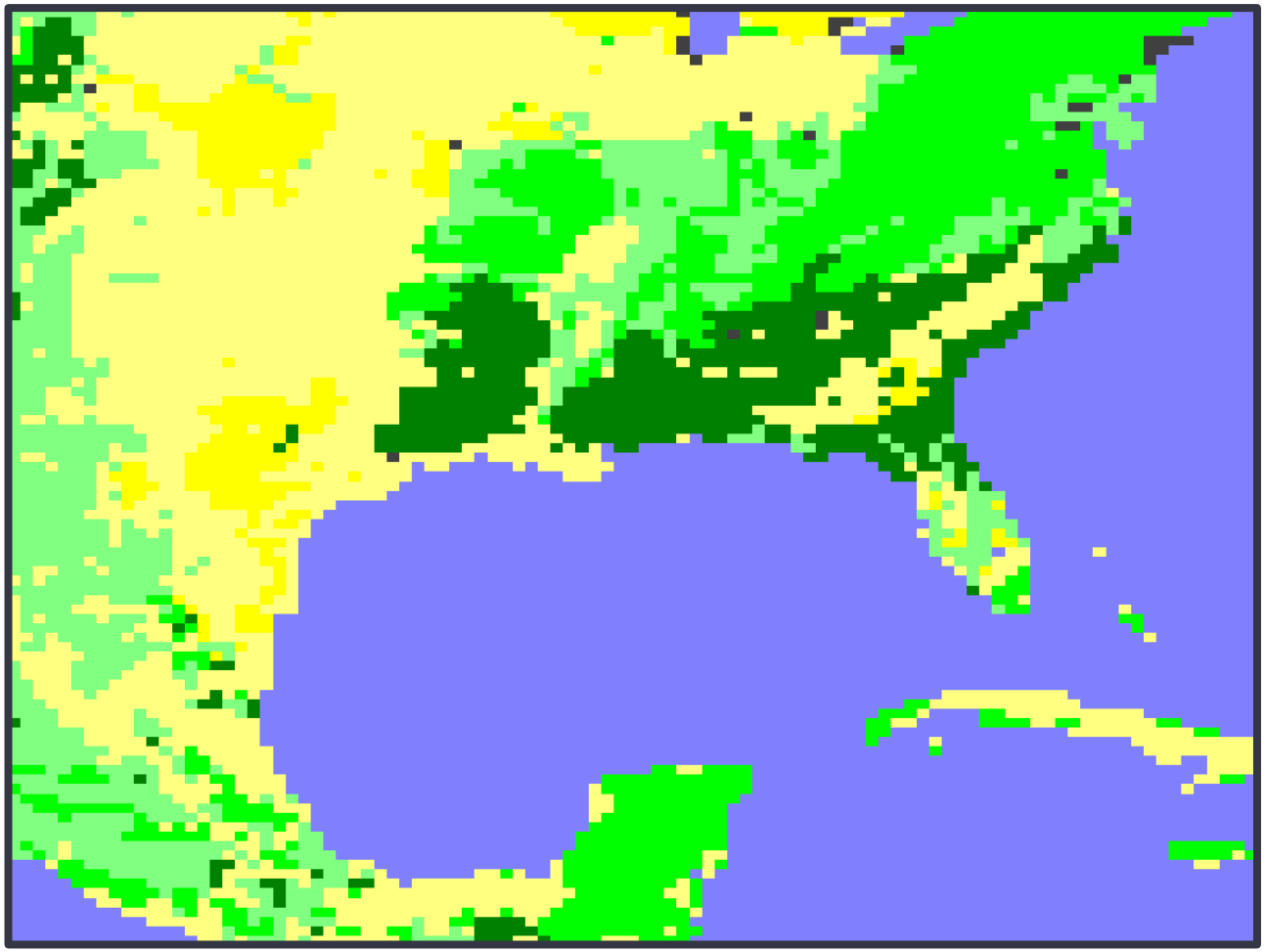
- A *finer-resolution* domain used during a model run
- Enables running at a higher-resolution without:
  - Uniformly high-resolution over a large domain – VERY expensive
  - High resolution for a very small domain, with mismatched time and spatial lateral boundary conditions

What is a nest?

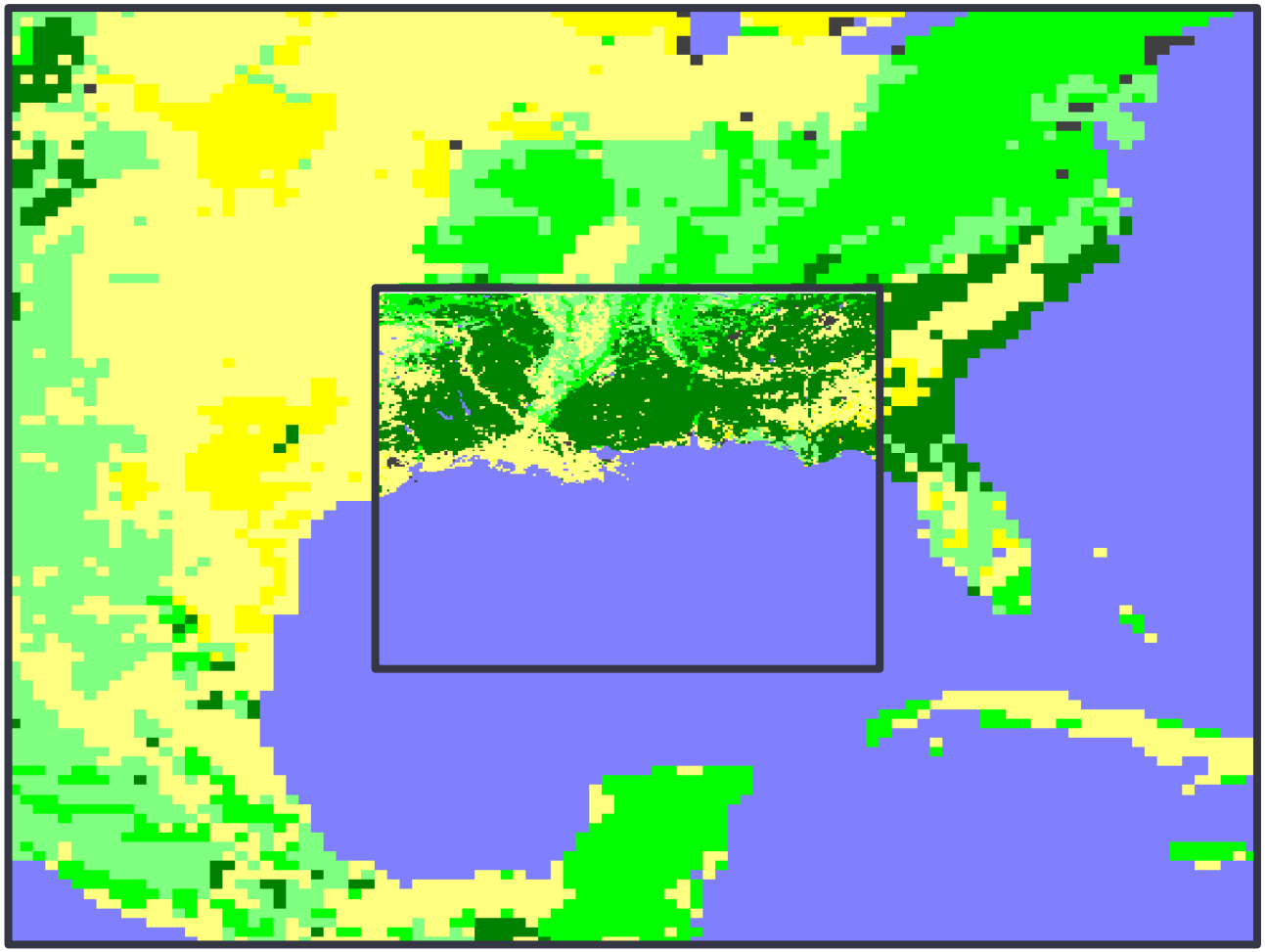
- Covers a portion of the parent domain, and is fully contained by the parent domain
- Driven along its lateral boundaries by the parent domain



When Should I Use Nests?



When Should I Use Nests?



When Should I Use Nests?

- Need to simulate localized phenomena: convection, topography, landuse-forced, etc.
  - What resolution is necessary to resolve what you are interested in?
  - Input data resolution is too coarse by more than a factor of 5-10x
  - Would like to provide better boundary conditions for the area of interest
    - BC's for external sources are typically 3-6 hours and do not have tendencies for all predicted fields
  - Computing resources not available for uniform coverage

Types of Nesting

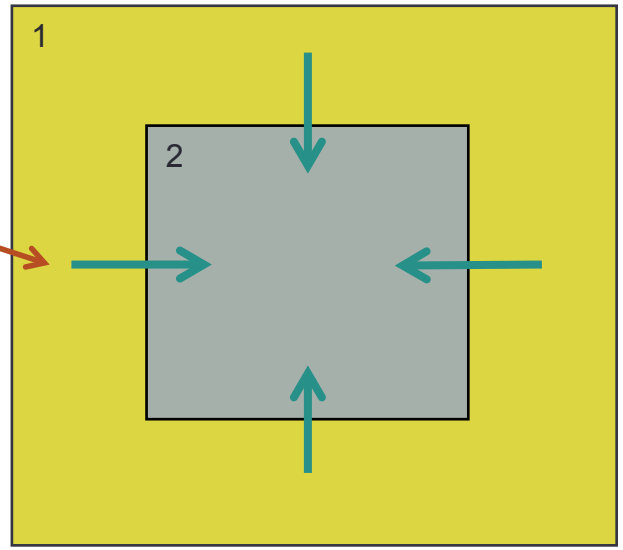
- Using a single input domain (met_em.d01*)
  - No met_em.d02* files are used
  - All fields are interpolated from the coarse grid
  - Only recommended if nest is over the ocean
- Using multiple input domains
  - Each domain contains full input data files
- Specified move
  - Originally used as a testing facility – can use, but tedious to set-up
  - Must specify every move
- Automatic move
  - Build WRF with “3=vortex following”
  - Only for tropical cyclone tracking
  - Expensive for single large nest
- ndown.exe
  - If you have run a long coarse domain simulation (years) and later decide you want to have a nest with higher resolution.

Types of Nesting

One-way/two-way nesting

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

Lateral boundary information is fed to the nest, from the parent.



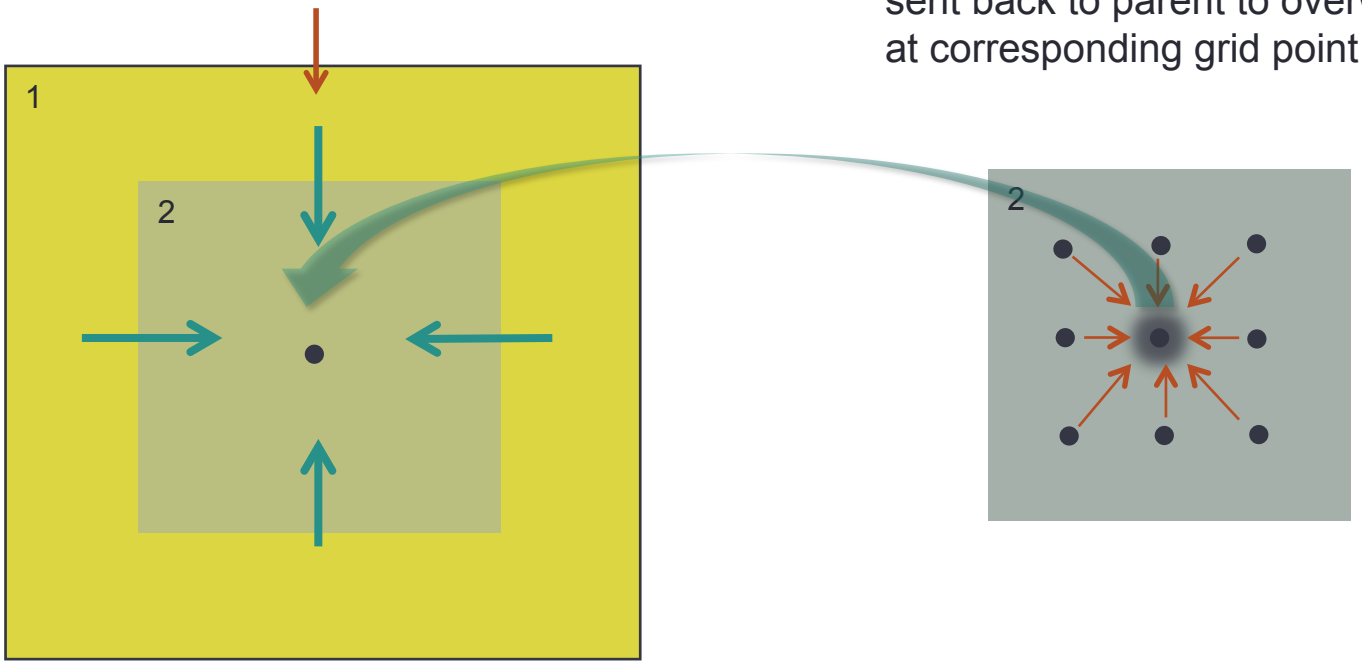
Types of Nesting

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

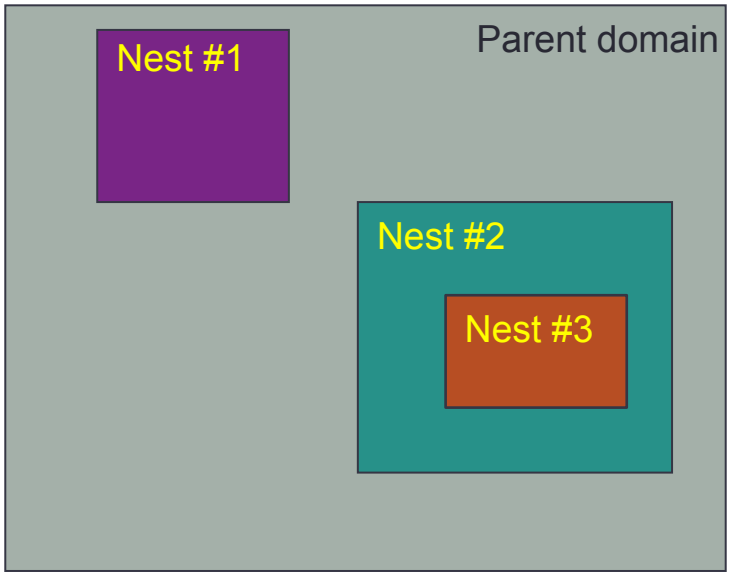
- Determined by the namelist parameter “feedback”
  - **feedback = 1 (turned on/two-way)**

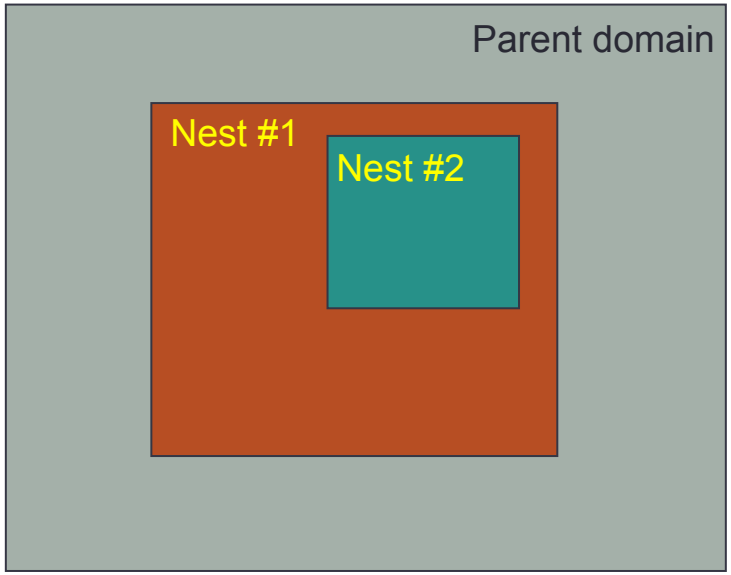
(1) Lateral boundary information is fed to the nest, from the parent.

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

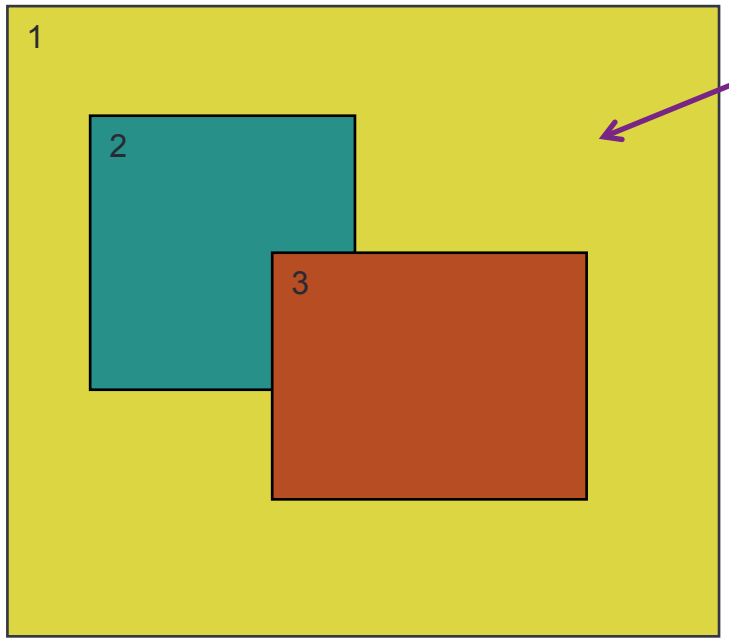


Nests that are OK



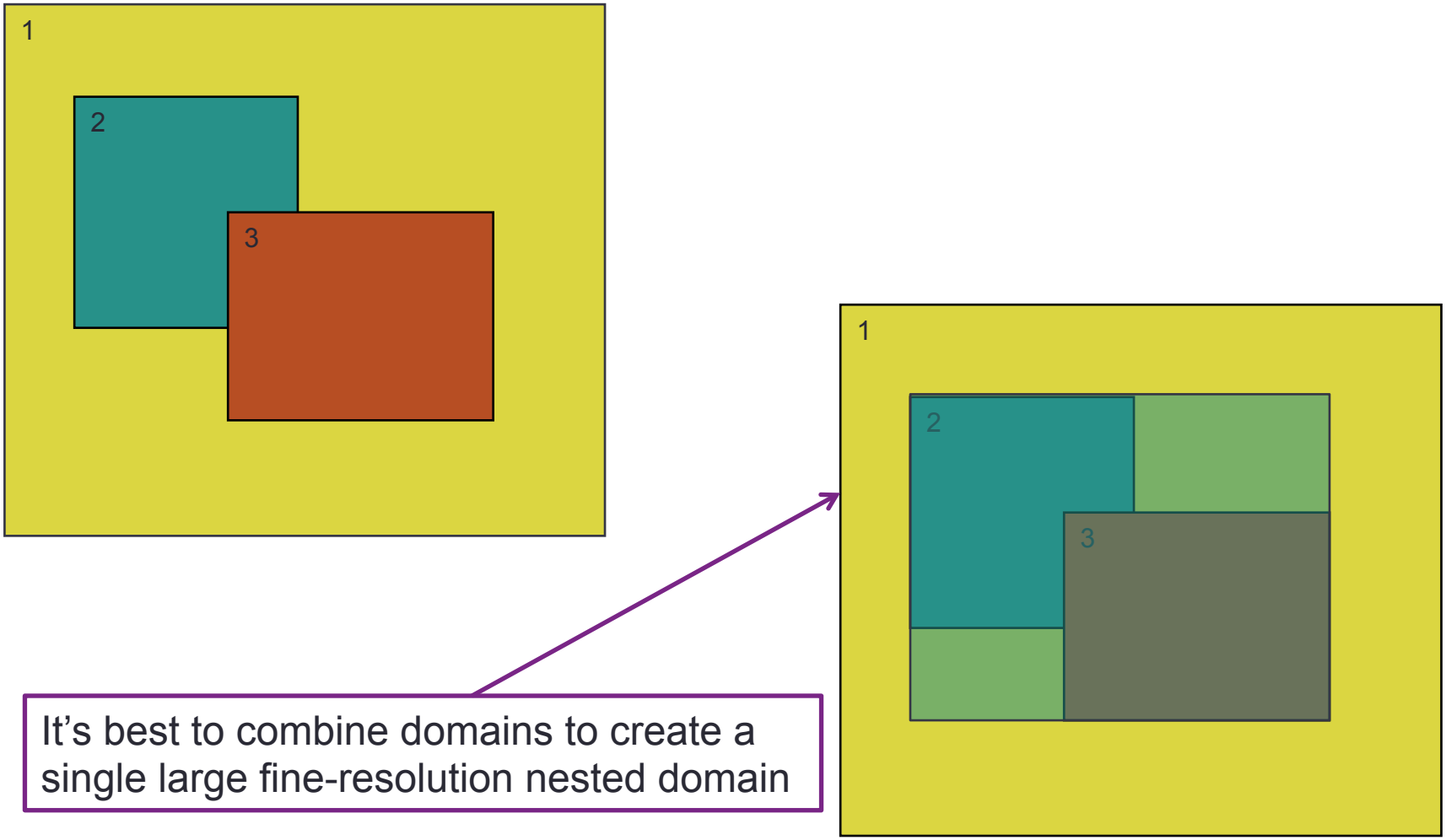


Nests that are NOT OK

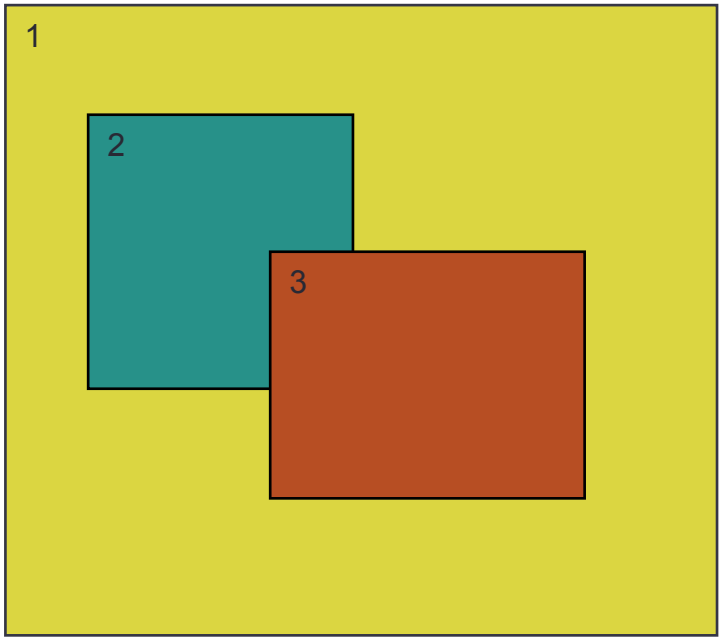


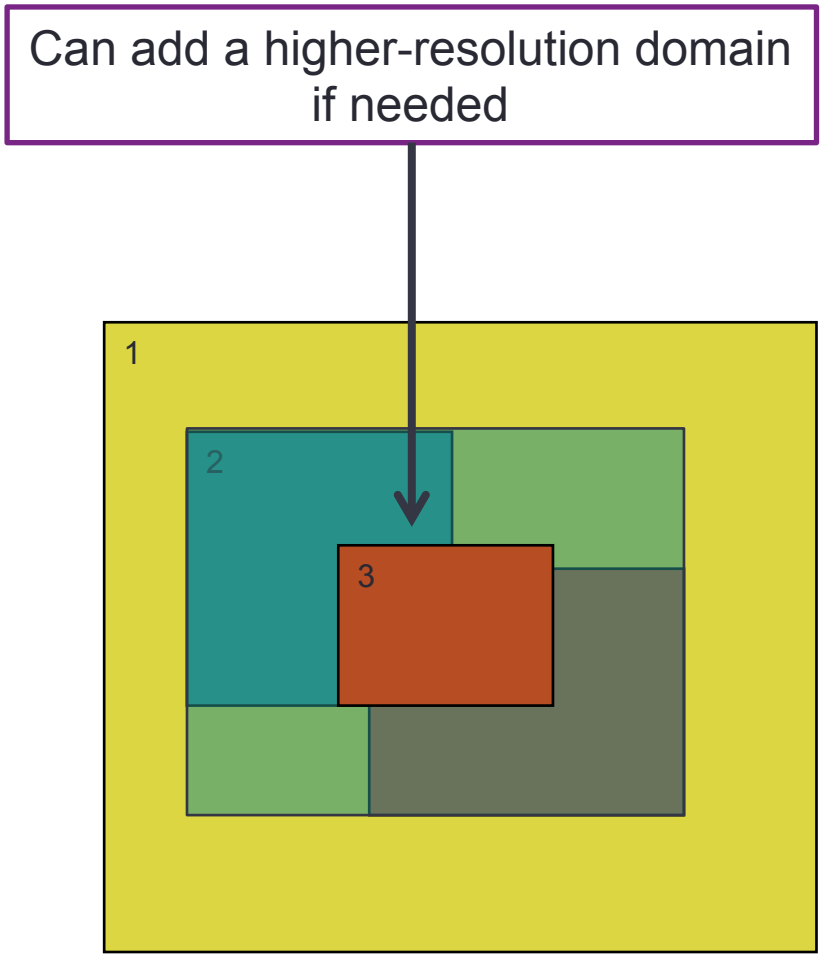
Child domains *may not* have overlapping points in the parent domain (1-way nesting excluded).

Nests that are NOT OK

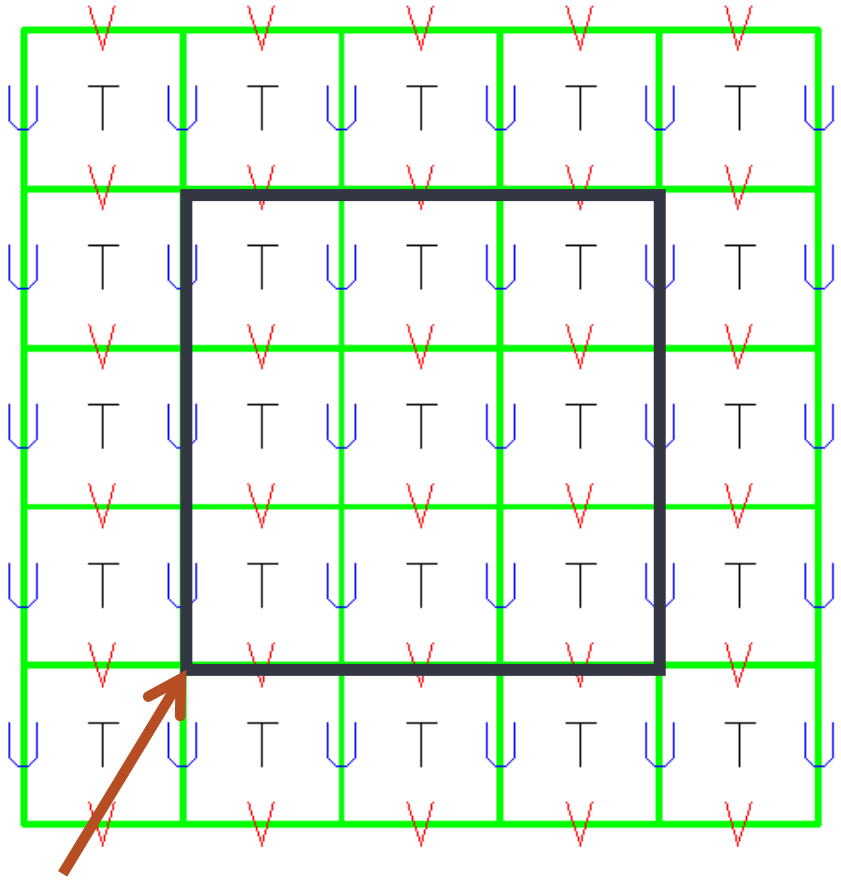


Nests that are NOT OK





WRF Coarse-fine Domain Overlap



i_parent_start
j_parent_start

- The nested domain can be placed *anywhere* within the parent domain and the nested grid cells will exactly overlap the parent cells at the coincident cell boundaries
- Coincident parent/nest grid points eliminate the need for complex, generalized remapping calculations, and enhances model performance and portability.

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, 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 program only requires initial time for fine domain (unless doing nudging or SST-update)

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,     20,
   j_parent_start    =   1,     17,

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


   dx                = 15000,
   dy                = 15000,
   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/'
/
```

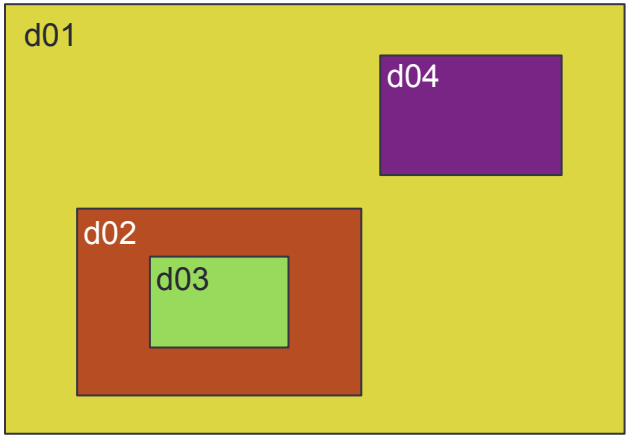
Used for nesting purposes
- What is the grid ratio for each nest?
- Where is it located inside its parent?
- parent_grid_ratio: integer ratio required

Domain sizes: How many grid points does each domain have?

namelist.*wps* set-up: *&geogrid*

```
&geogrid

   parent_id           =   1,       1,
   parent_grid_ratio   =   1,       3,
   i_parent_start      =   1,      20,
   j_parent_start      =   1,      17,

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


   dx                  = 15000,
   dy                  = 15000,
   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



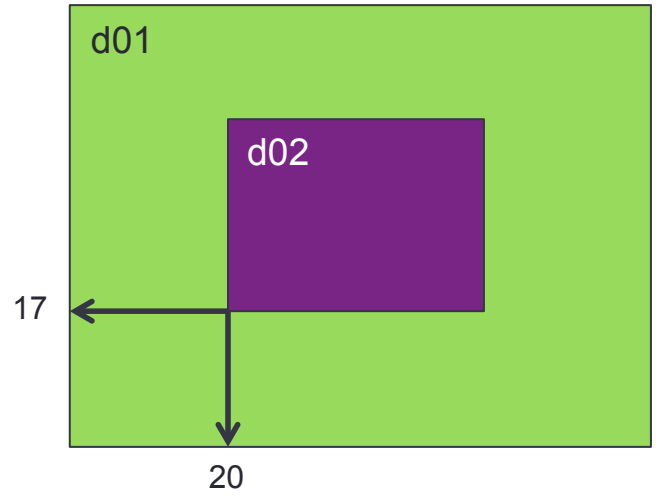
parent_id = 1, 1, 2, 1

namelist.*wps* set-up: *&geogrid*

```
&geogrid

   parent_id           =   1,      1,
   parent_grid_ratio   =   1,      3,
   i_parent_start      =   1,     20,
   j_parent_start      =   1,     17,

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


   dx                  = 15000,
   dy                  = 15000,
   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:
recommended ratios are 3:1 or 5:1
(odd ratios, less than 7)

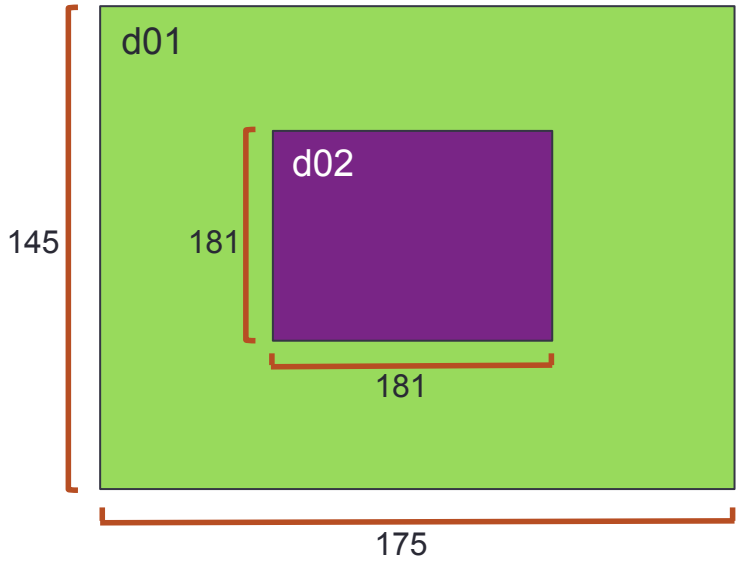
i/j_parent_start:



namelist.*wps* set-up: *&geogrid*

```
&geogrid

  parent_id             =   1,        1,
  parent_grid_ratio     =   1,        3,
  i_parent_start        =   1,       20,
  j_parent_start        =   1,       17,

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


  dx                    = 15000,
  dy                    = 15000,
  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:

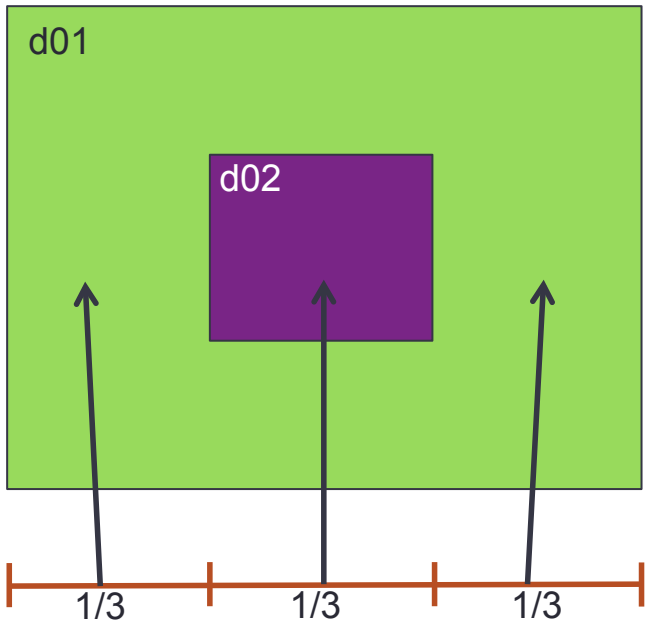
- Domains should be no smaller than about 100x100
- Avoid placing any boundaries over complex terrain

namelist.*wps* set-up: *&geogrid*

```
&geogrid

   parent_id             =   1,        1,
   parent_grid_ratio     =   1,        3,
   i_parent_start        =   1,       20,
   j_parent_start        =   1,       17,

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


   dx                    = 15000,
   dy                    = 15000,
   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
- need MUCH larger buffer zone



- Good practice to have ~1/3 of coarse-grid surrounding each side of nest
- Nest can be placed a bit downstream of the inflow boundary

namelist.*wps* set-up: *&geogrid*

```
&geogrid

   parent_id           =   1,        1,
   parent_grid_ratio   =   1,        3,
   i_parent_start      =   1,       20,
   j_parent_start      =   1,       17,

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


   dx                  = 15000,
   dy                  = 15000,
   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:**
No changes need to be made to the &ungrib and &metgrid namelists records for nesting purposes

namelist.input (WRFV3)

namelist.*input* set-up: *&time_control*

```
&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
 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 *WRFV3/test/em_real/* (or *WRFV3/run/*) directory

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

history_interval:
May choose to have more frequent output time for nests

frames_per_outfile:
May choose to have all history outputs in a single file, or in multiple files - for particular netcdf conventions (e.g., ncview), it's necessary to have 1 file per time period.

namelist.*input* set-up: *&domains*

```
&domains
 time_step                  = 180,
 time_step_fract_num        = 0,
 time_step_fract_den        = 1,
 max_dom                    = 2,
 e_we                       = 74,   112,   94,
 e_sn                       = 61,   97,    91,
 e_vert                     = 30,   30,    30,
 p_top_requested            = 5000,
 num_metgrid_levels         = 27,
 num_metgrid_soil_levels    = 4,
 dx                         = 30000, 10000,  3333.33,
 dy                         = 30000, 10000,  3333.33,
 grid_id                    = 1,    2,     3,
 parent_id                  = 0,    1,     2,
 i_parent_start             = 1,    31,    30,
 j_parent_start             = 1,    17,    30,
 parent_grid_ratio          = 1,    3,     3,
 parent_time_step_ratio     = 1,    3,     3,
 feedback                   = 1,
 smooth_option              = 0
 /
```

max_dom:
Activate nests - # of domains to run
VERY IMPORTANT!

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

namelist.*input* set-up: *&domains*

```
&domains
.....
grid_id                             = 1,    2,    3,
parent_id                           = 0,    1,    2,
i_parent_start                      = 1,    31,   30,
j_parent_start                      = 1,    17,   30,
parent_grid_ratio                   = 1,    3,    3,
parent_time_step_ratio              = 1,    3,    3,
feedback                            = 1,
smooth_option                       = 0
/
```

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!

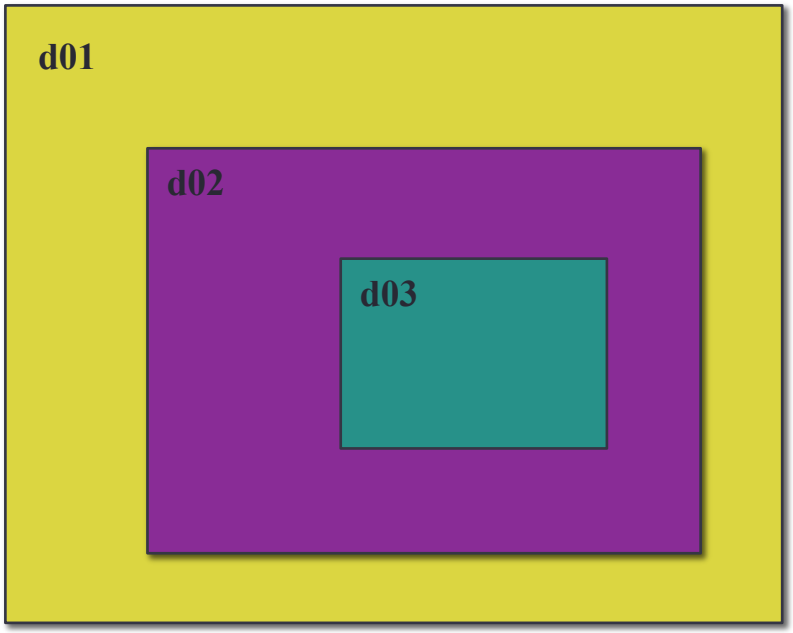
namelist.*input* set-up: *&dynamics*

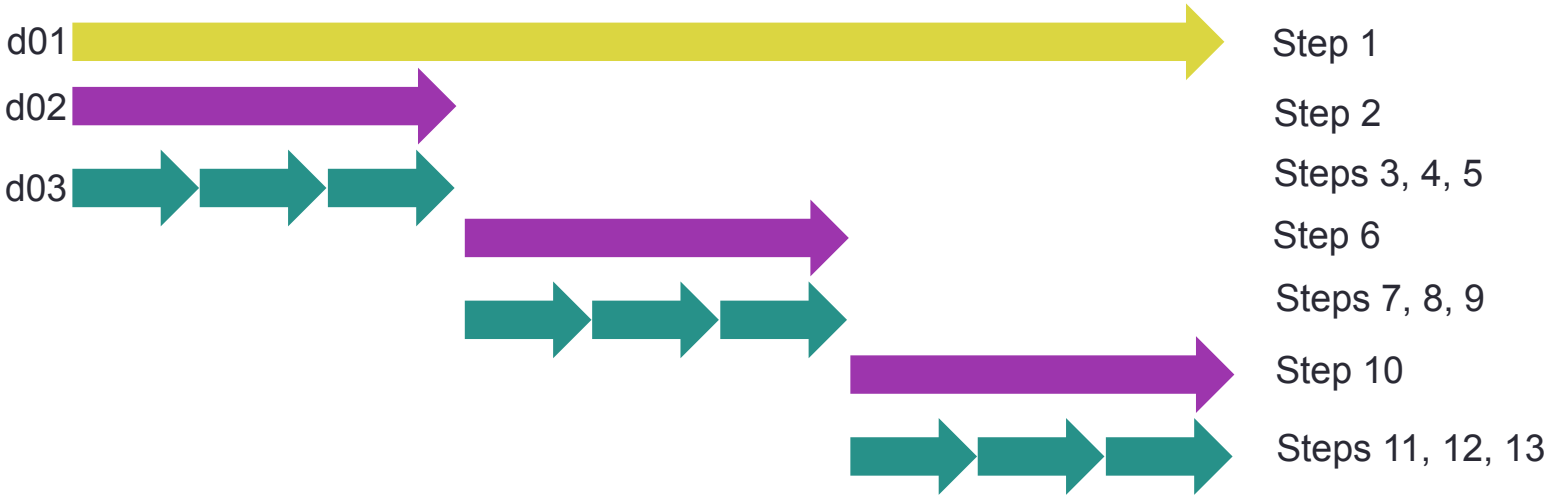
```
&dynamics
 ..........
 hybrid_opt          = 2,
/
```

Hybrid Vertical Coordinate Option:
Must be consistent between real and WRF
(set the same for both)

Nested 3:1 Time Step Ratio

- Example: 3-domain nested run
  - D01: a single 3 min dt
  - D02: a single 1 min dt
  - D03: 20 second pieces, up to 1 min





namelist.*input* set-up: *&physics*

- You must 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); NOTE: cudt=5 => run CU every 5 min

namelist.*input* set-up: *&physics*

- You must 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); NOTE: cudt=5 => run CU every 5 min

Where do I start?

- Always start with a *namelist* template provided in the WRFV3/test/em_real (or WRFV3/run/) directory

- Use documents/websites to guide your namelist modifications
  - WRFV3/run/*README.namelist*
  - WRFV3/test/em_real/*examples.namelist*
  - Users' Guide, Chapter 5
    - http://www2.mmm.ucar.edu/wrf/users/docs/user_guide_V3.8/users_guide_chap5.htm
  - Namelist Best Practice web pages:
    - WPS: http://www2.mmm.ucar.edu/wrf/users/namelist_best_prac_wps.html
    - WRFV3: http://www2.mmm.ucar.edu/wrf/users/namelist_best_prac_wrf.html

- Not all namelist options are domain dependent. If in doubt:
  - Check WRFV3/Registry/*Registry.EM_COMMON* or *registry.io_boilerplate* (grep for parameter names)
  - Check WRFV3/run/*README.namelist* (grep for parameter names)
  - Rule of thumb: If default namelist only has 1 column, don't add values for other columns!

Steps to run with a nest

- WPS: Identical to single domain run:
  1) Make sure you are in the WPS/ directory
  2) Make necessary changes to the *namelist.wps* file
  3) Run geogrid.exe, ungrib.exe, and metgrid.exe

```
./geogrid.exe
./ungrib.exe
./metgrid.exe
```

- WRFV3: Identical to single domain run:
  1) Make sure you are in the *WRFV3/test/em_real (*or *WRFV3/run/) directory*
  2) Move or link WPS output files (*met_em.d0**) to your running directory

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

  3) Edit *namelist.input* file for the appropriate grid and times of the case
  4) Run initialization program (assuming a dmpar compile):

```
mpirun -np n ./real.exe
```

- "n": number of processors used

  1) Run model executable (assuming a dmpar compile):

```
mpirun -np n ./wrf.exe
```

Nesting in real.exe

- Real program can read multiple domain input files from metgrid (met_em_d0*)
- There is no horizontal interpolation taking place between parent and child domains, at this stage (this is handled during the WRF model run)
- There are no consistency check between domains (this is handled in the feedback step for the WRF model)
- real.exe must be re-run if you make changes to:
  - Date/time
  - Domain – size, location, quanitity
  - Land surface model option (sf_surface_physics)
  - Input data

Successful *real.exe* Run

- If *real.exe* was successful, you should see this at the end of your rsl.error.0000 file (assuming a dmpar compile):
  - `tail rsl.error.0000`
  - SUCCESS COMPLETE REAL_EM INIT

- You should have these files in your running directory:
  - wrfbdy_d01 :
    - time level data at model's start time (includes all domains)
  - wrfinput_d01, wrfinput_d02, ….
    - time_level data at the lateral boundary for all times
    - 1 file per domain

Successful *wrf.exe* Run

- If *wrf.exe* was successful, you should see this at the end of your rsl.error.0000 file (assuming a dmpar compile):
  - `tail rsl.error.0000`
  - SUCCESS COMPLETE WRF

- You should have these files in your running directory:
  - wrfout_d01_2005-08-28_00:00:00
  - wrfout_d02_2005-08-28_00:00:00
    - One for each domain, for each history time (depending on how you set 'frames_per_outfile')
  - wrfrst_d01_2005-08-28_00:00:00
  - wrfrst_d02_2005-08-28_00:00:00
    - If "restart_interval" is less than or equal to the integration time

Questions?