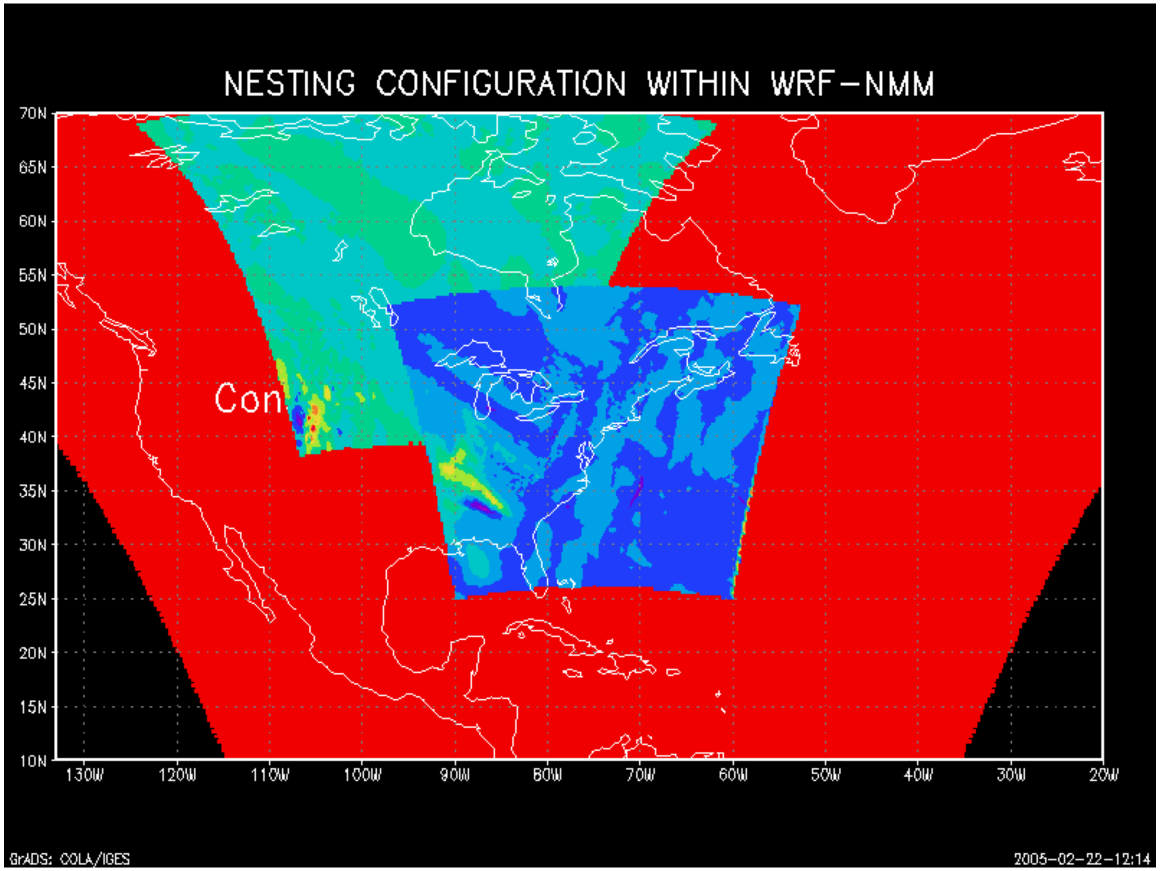
Nesting in WRF

Dave Gill
Matthew Pyle



Nesting Basics - What is a nest

- A nest is a ***finer-resolution*** model run. It may be ***embedded*** simultaneously within a coarser-resolution (parent) model run, or ***run independently*** as a separate model forecast.
- The nest ***covers a portion*** of the parent domain, and is driven along its ***lateral boundaries*** by the parent domain.
- Nesting enables running at finer resolution without the following problems:
  - Uniformly high resolution over a large domain - prohibitively expensive
  - High resolution for a very small domain with mismatched time and spatial lateral boundary conditions

Nesting Basics - NMM

- Static, one- or two-way nesting

  - Static: The nest location is fixed in space
  - One-way: Information exchange between the parent and the nest is strictly down-scale. The nest solution does not feedback to the coarser/parent solution.
  - Two-way: Information exchange between the parent and the nest is bi-directional. The nest feedback impacts the coarse-grid domain's solution.
  - Fine grid input is for non-meteorological variables.

- Automatic moving nests are available, primarily for hurricane tracking (HWRF)

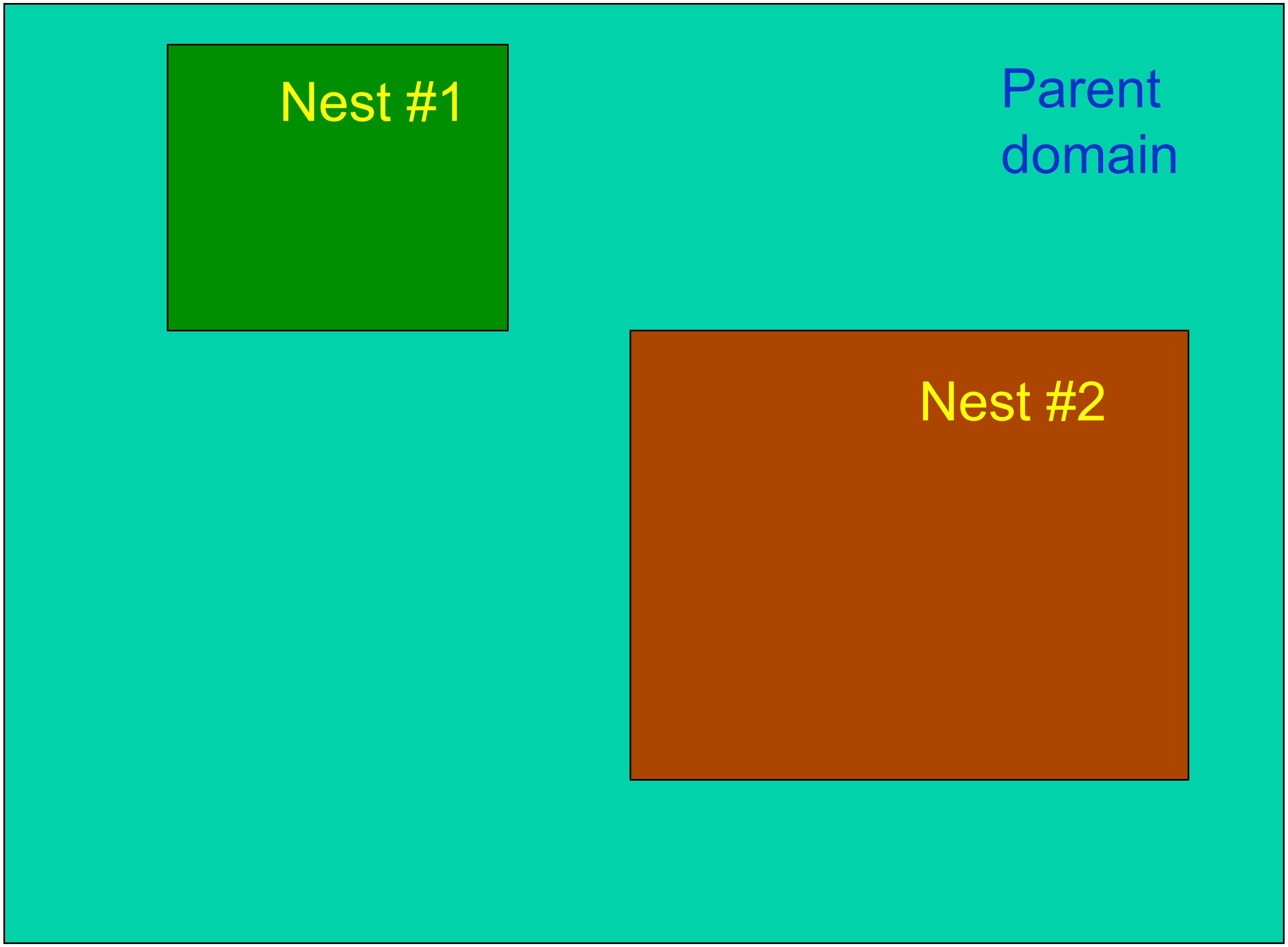
Nesting Basics - ARW

- One-way nesting via multiple model forecasts
- One-way nesting with a single model forecast, without feedback

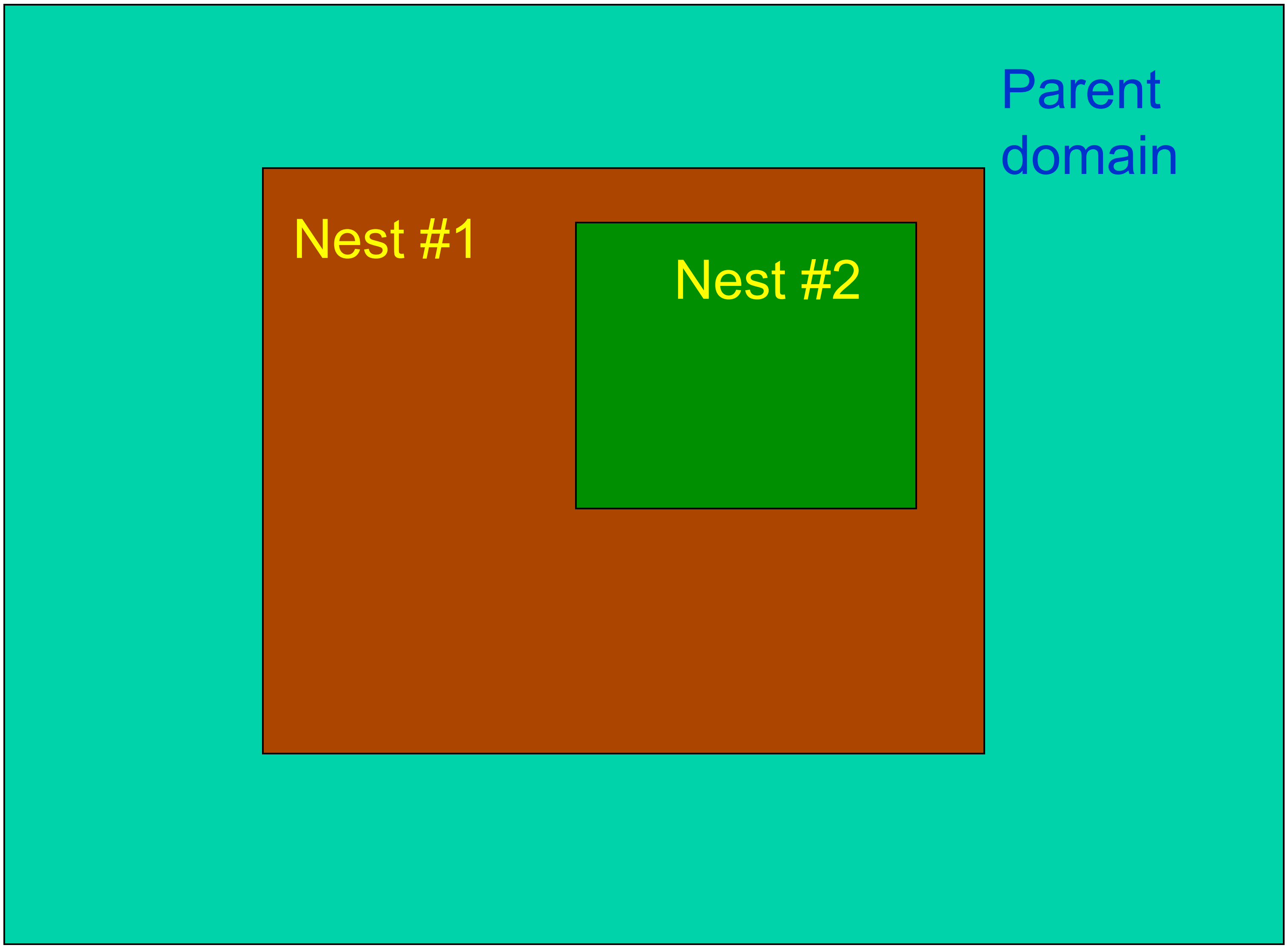
- One-way/two-way nesting with a single input file, all fields interpolated from the coarse grid
- One-way/two-way nesting with multiple input files, each domain with a full input data file
- One-way/two-way nesting with the coarse grid data including all meteorological fields, and the fine-grid domains including only the static files

- One-way/two-way nesting with a specified move for each nest
- One-way/two-way nesting with an automatic move on the nest determined through 500 mb low tracking

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

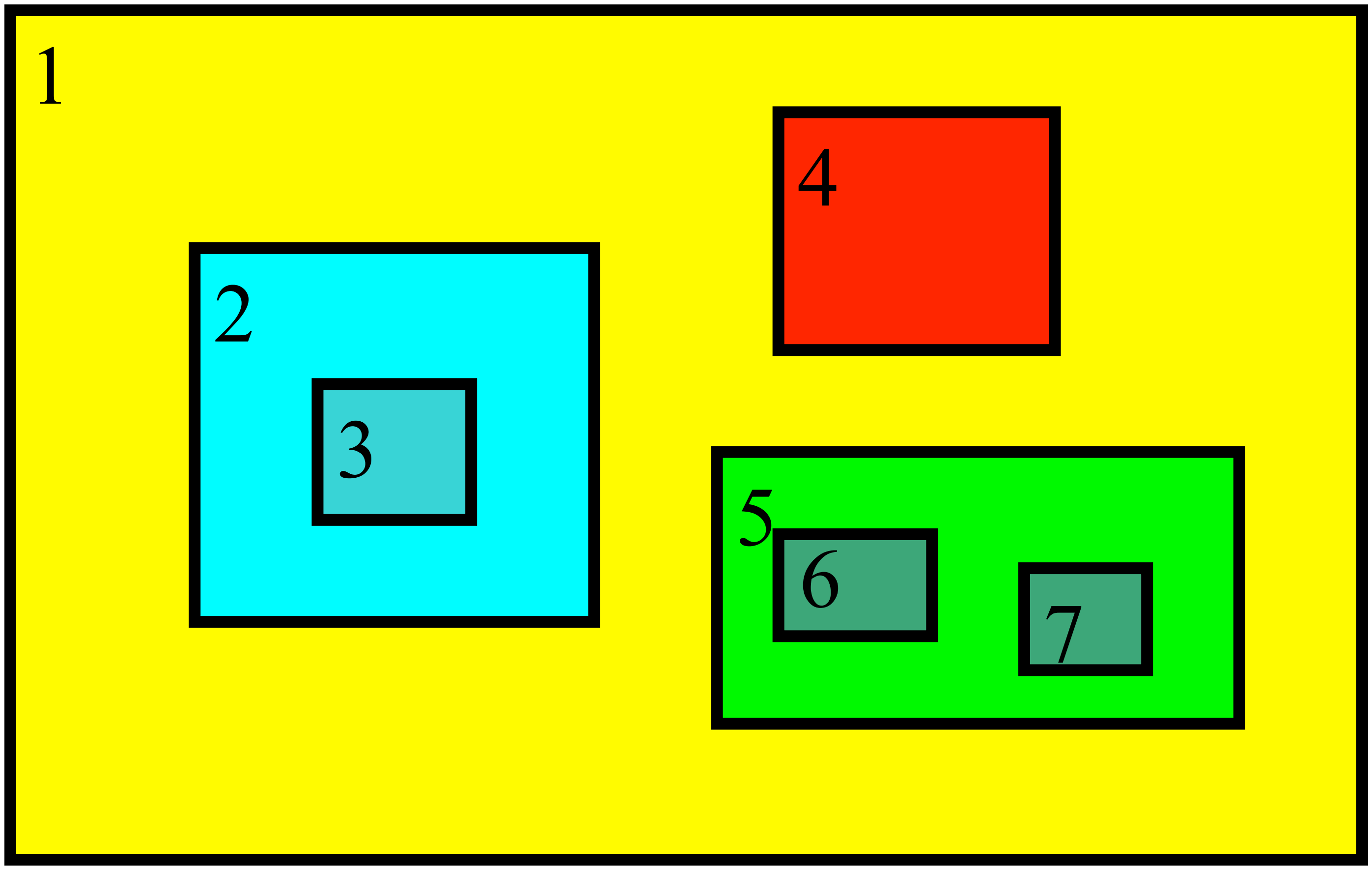


Two levels of nests, with nest #1 acting as the parent for nest #2



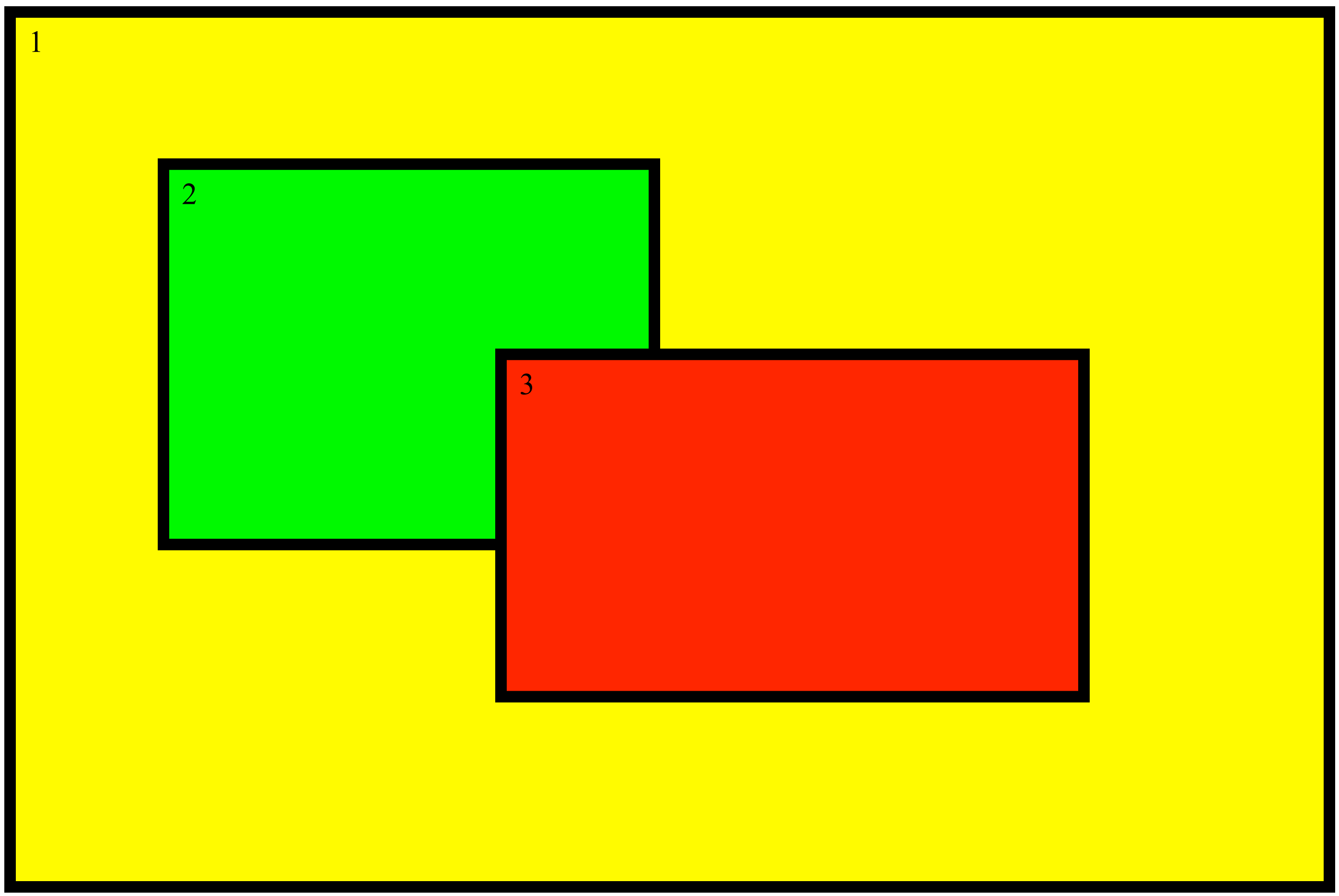
These are all OK

Telescoped to any depth
Any number of siblings



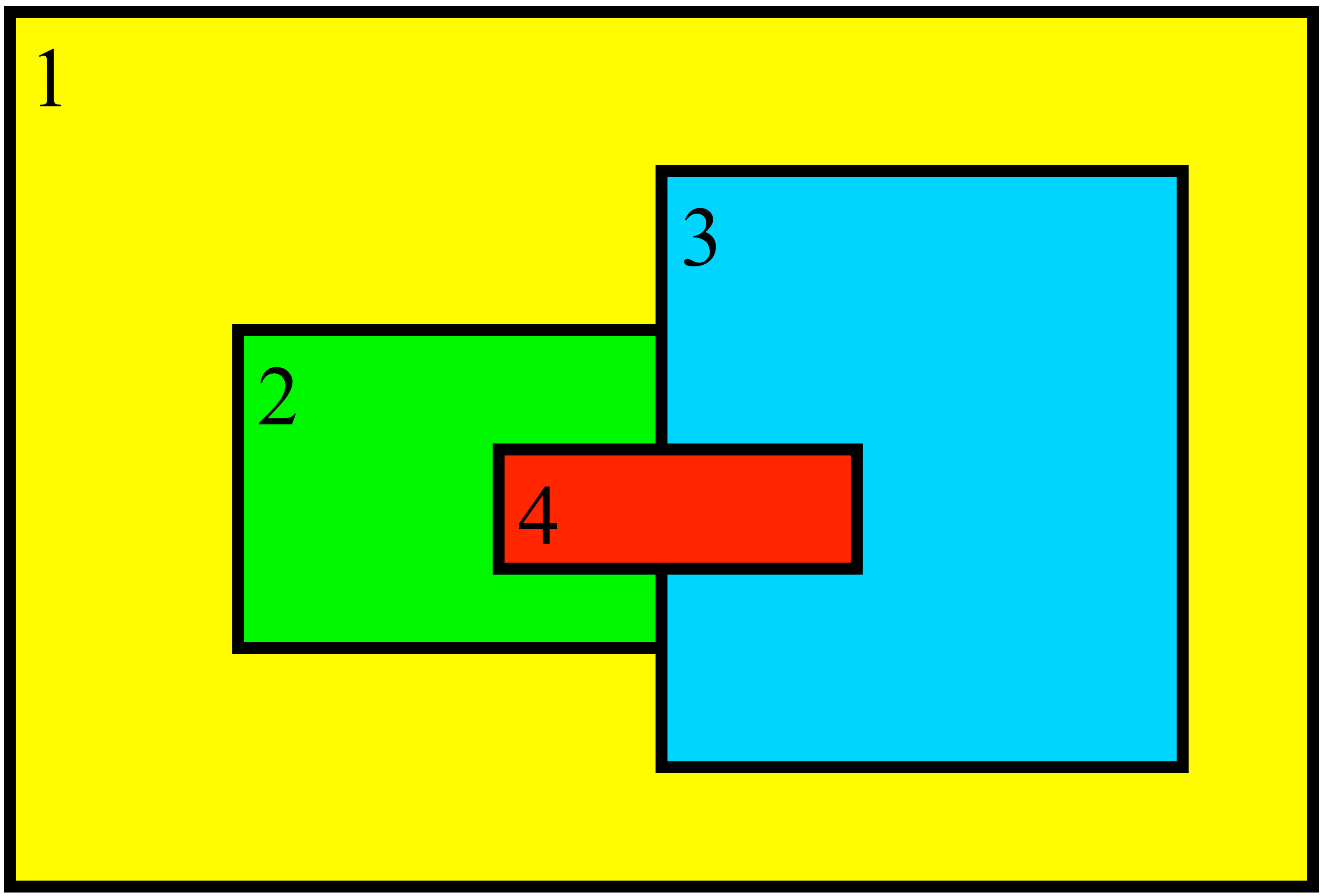
Not OK for 2-way

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



Not OK either

Domains have one, and only one, parent -
(domain 4 is NOT acceptable even with 1-way nesting)

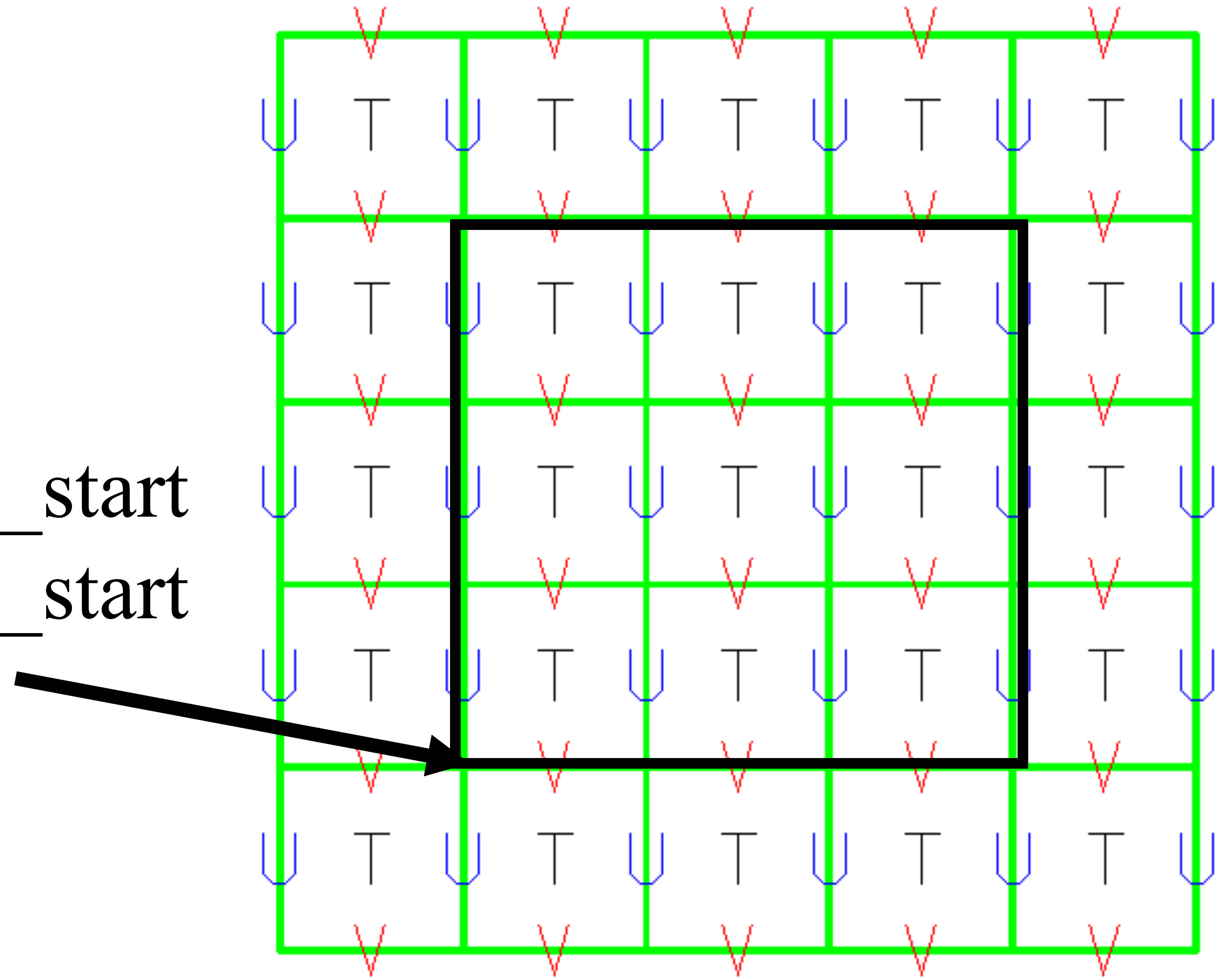


WRF Coarse-Fine Overlap

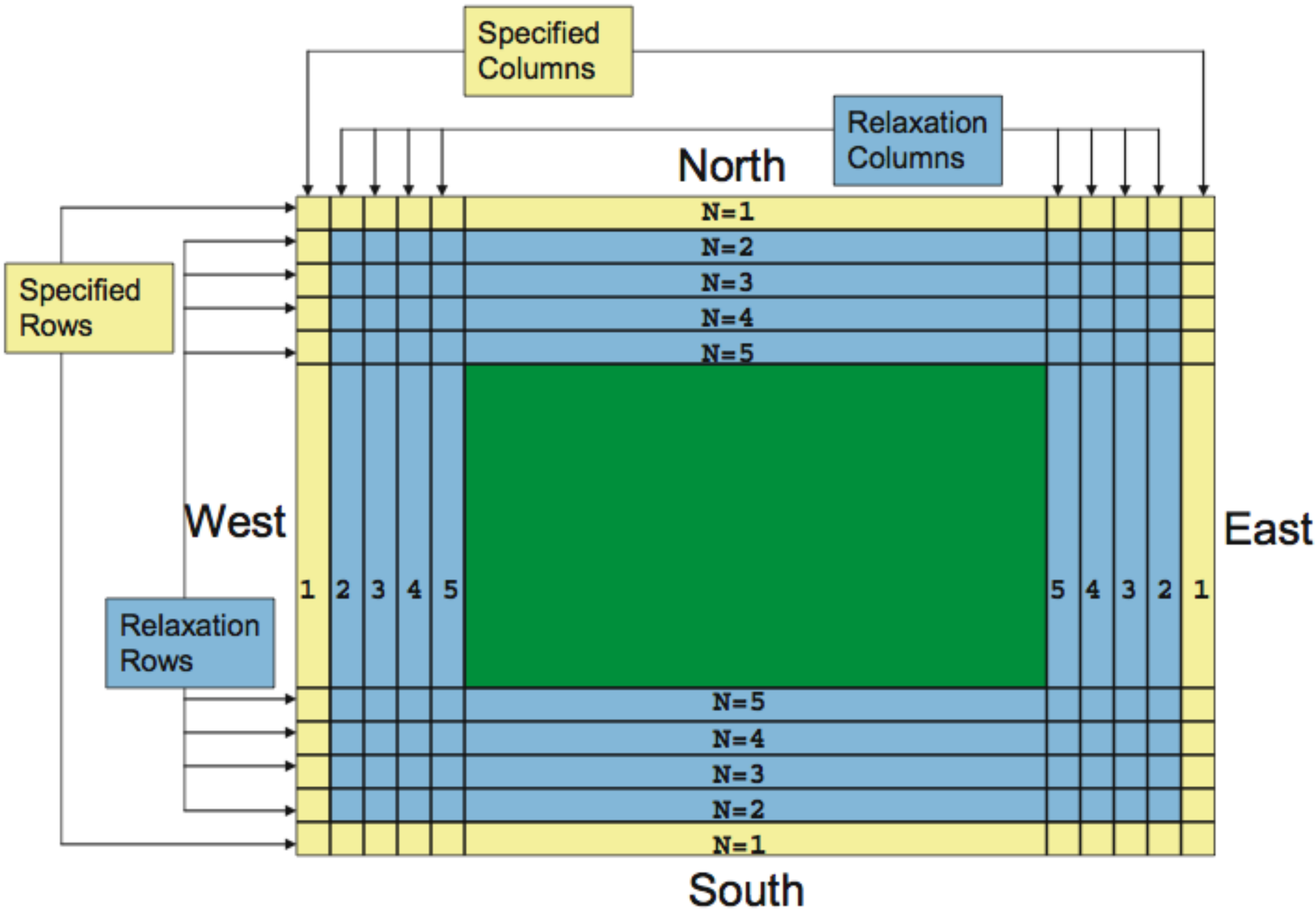
- The rectangular fine grid is coincident with a portion of the high-resolution grid that covers the entire coarse grid cell
- The nested domain can be placed anywhere within the parent domain and the nested grid cells will exactly overlap the parent cells at the coincident cell boundaries.
- Coincident parent/nest grid points eliminate the need for complex, generalized remapping calculations, and enhances model performance and portability.
- The grid design was created with moving nests in mind.

ARW Coarse Grid Staggering

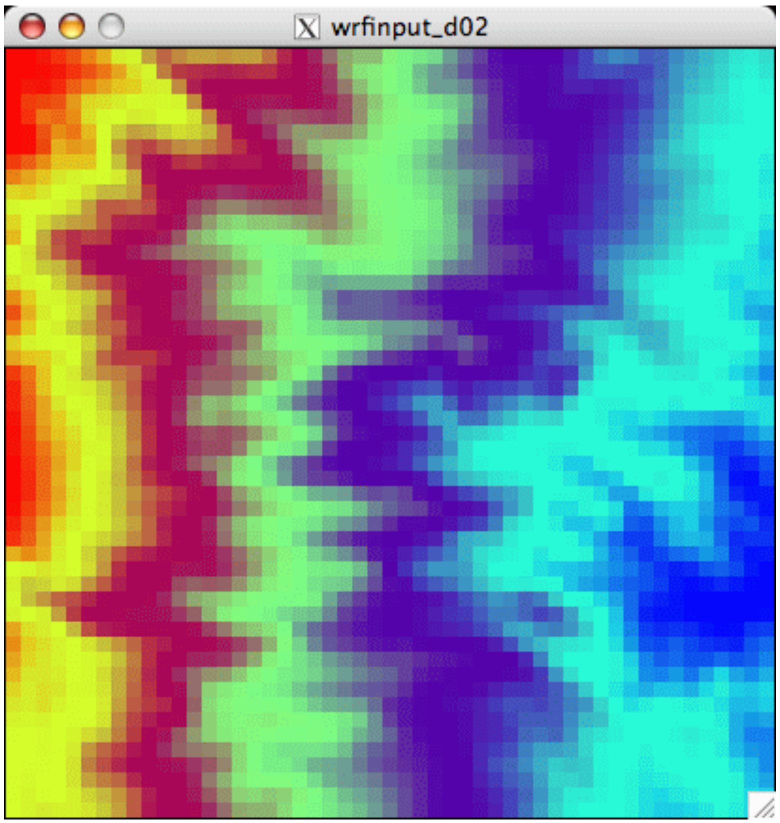
i_parent_start
j_parent_start

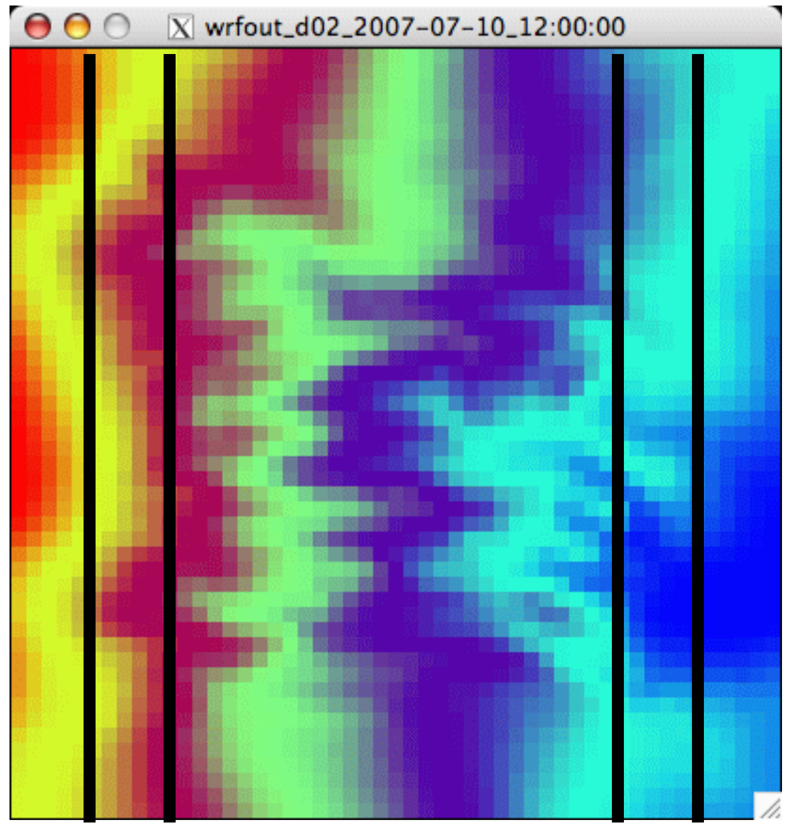


Real-Data Lateral Boundary Condition: Location of Specified and Relaxation Zones



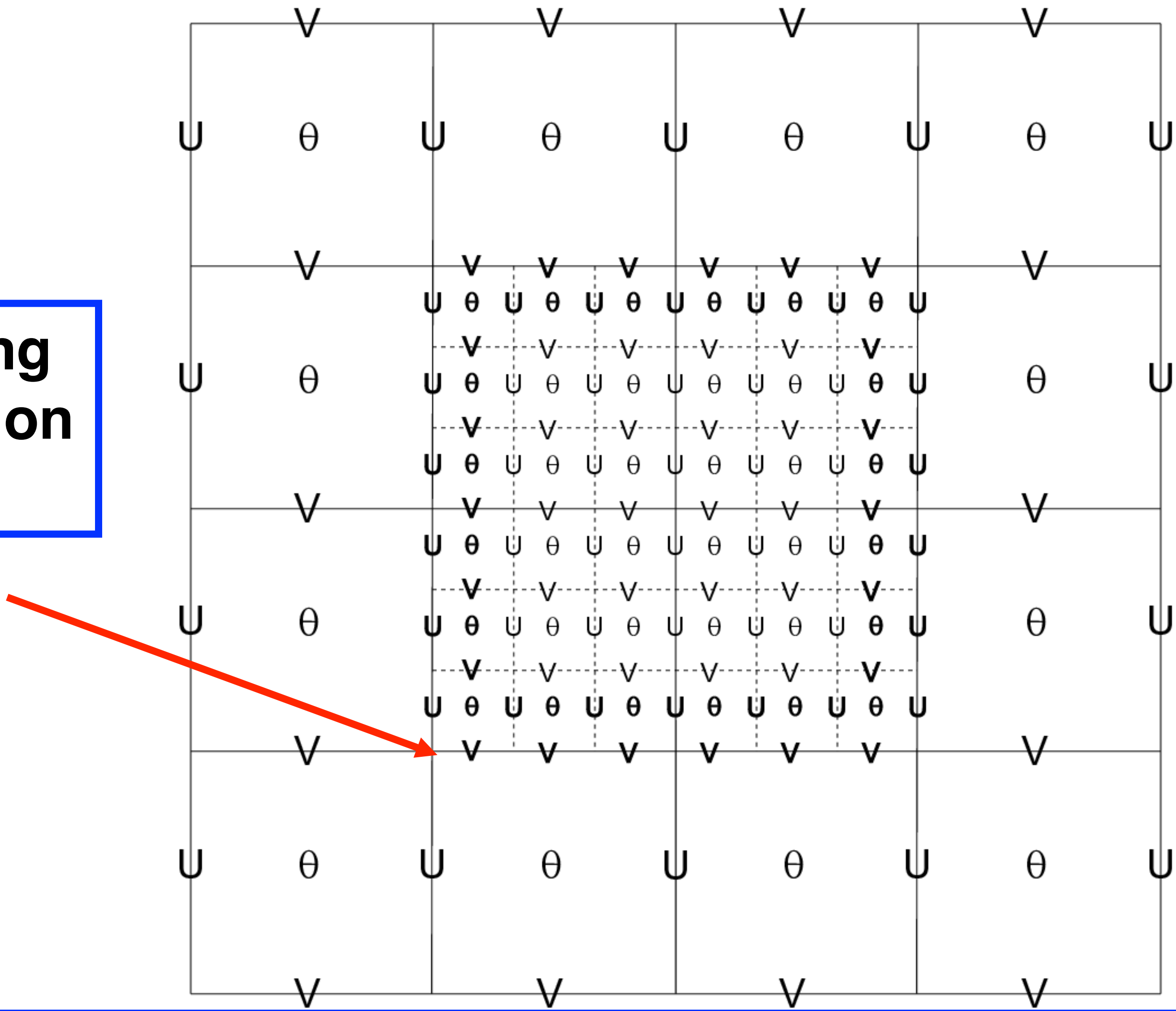
ARW Lateral Smoothing





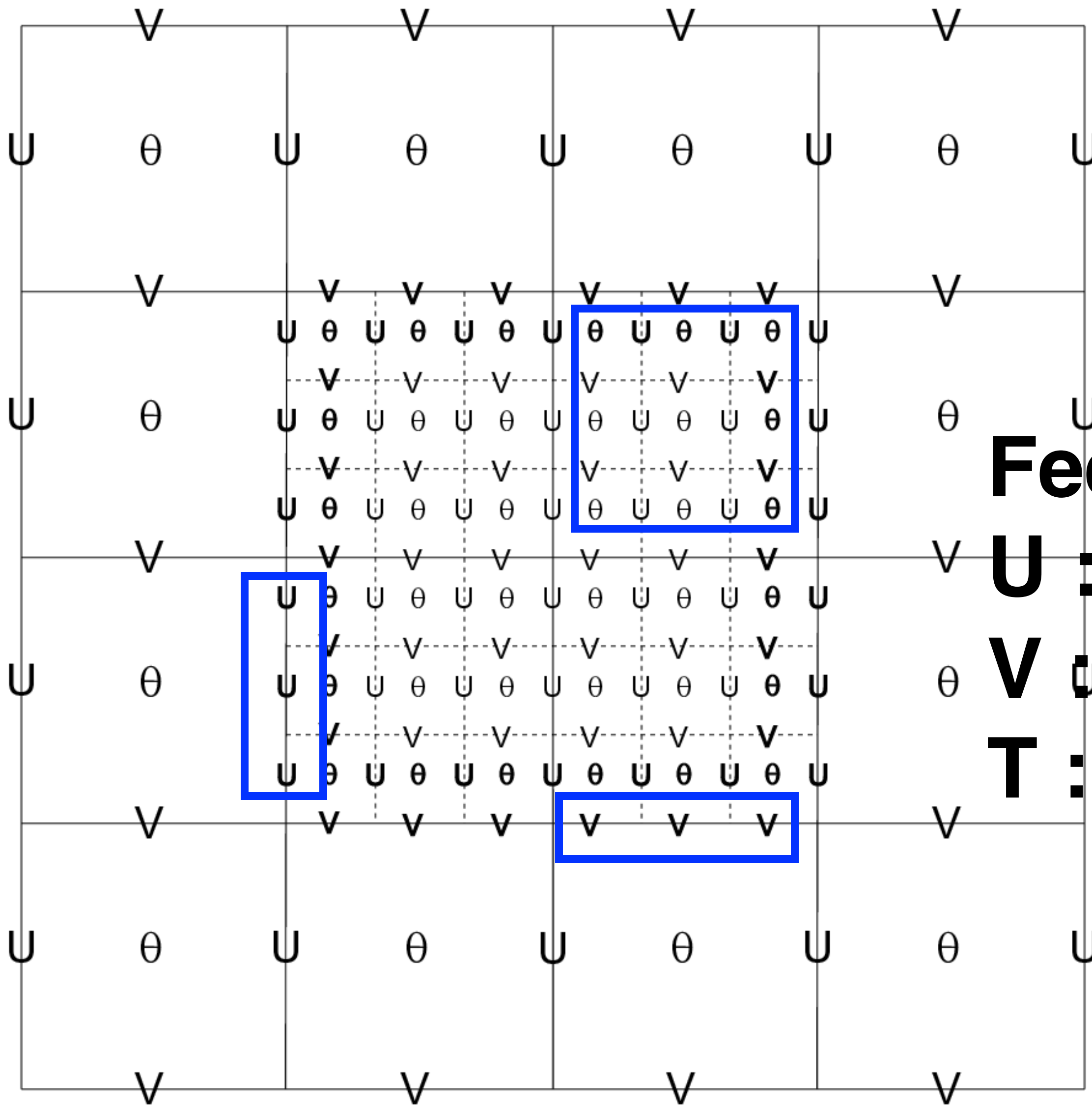
ARW Coarse Grid Staggering 3:1 Ratio

Starting Location
I = 31



CG … 30 31 32 33 34

ARW Coarse Grid Staggering 3:1 Ratio



Feedback:
U : column
V : row
T : cell

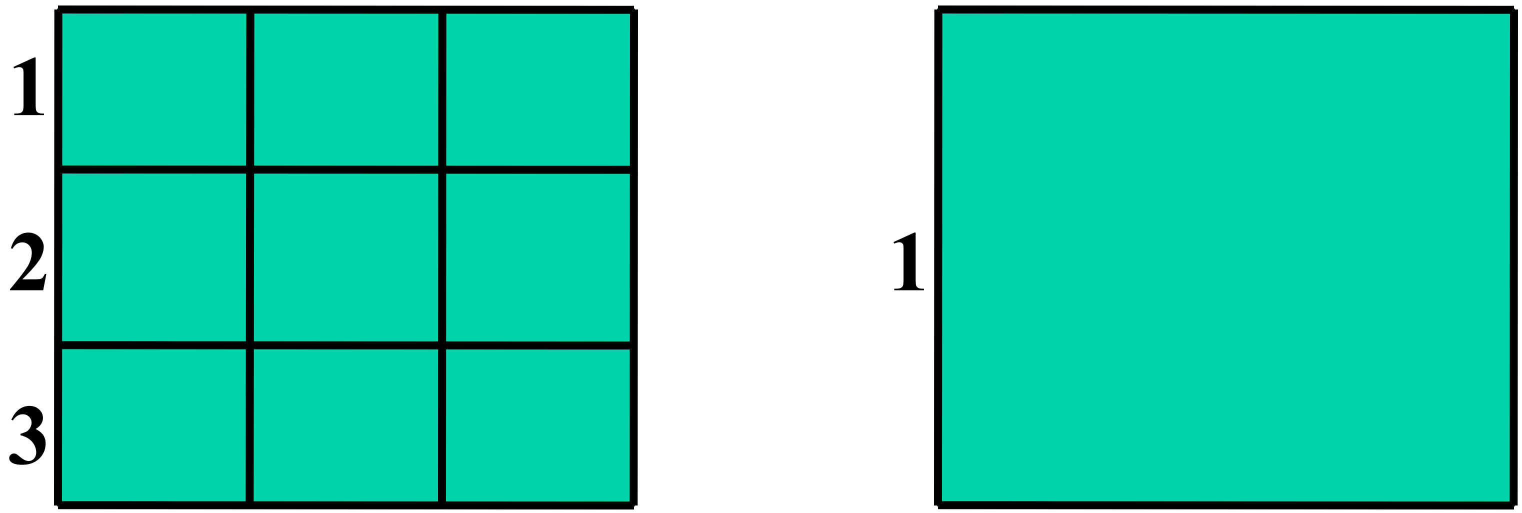
ARW Coarse Grid Staggering 3:1 Ratio

Feedback:

U : column

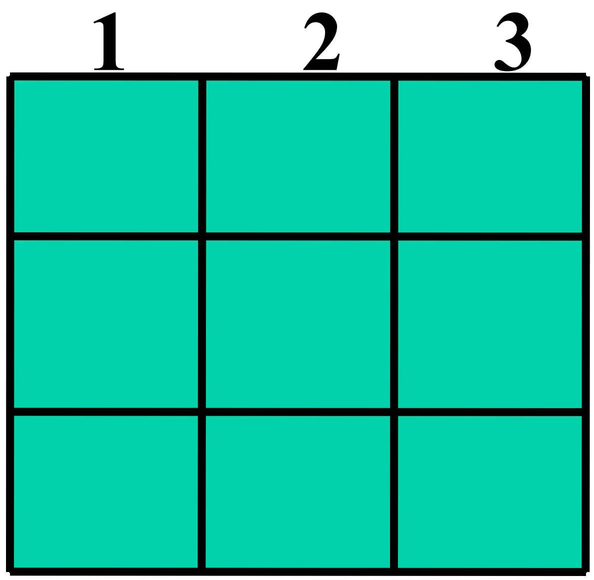
V : row

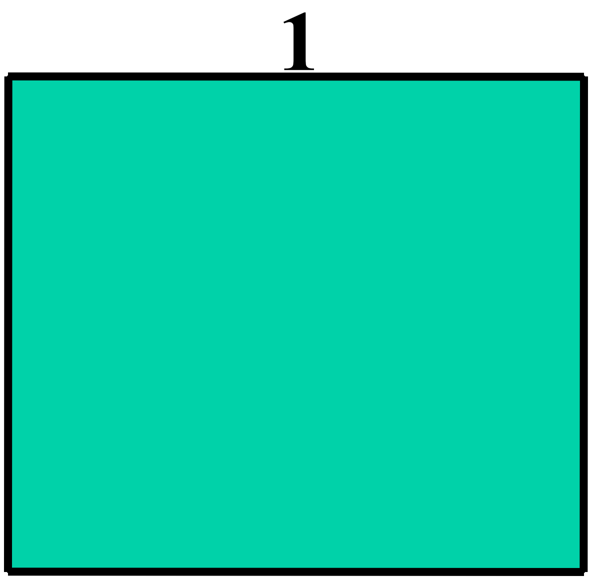
T : cell



ARW Coarse Grid Staggering 3:1 Ratio

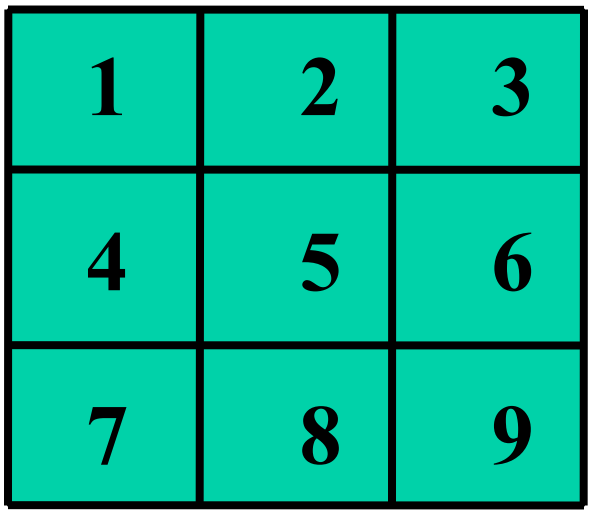
Feedback:
U : column
V : row
T : cell

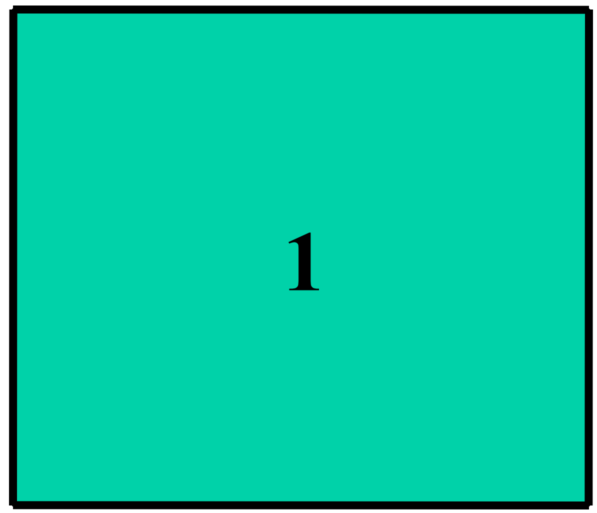




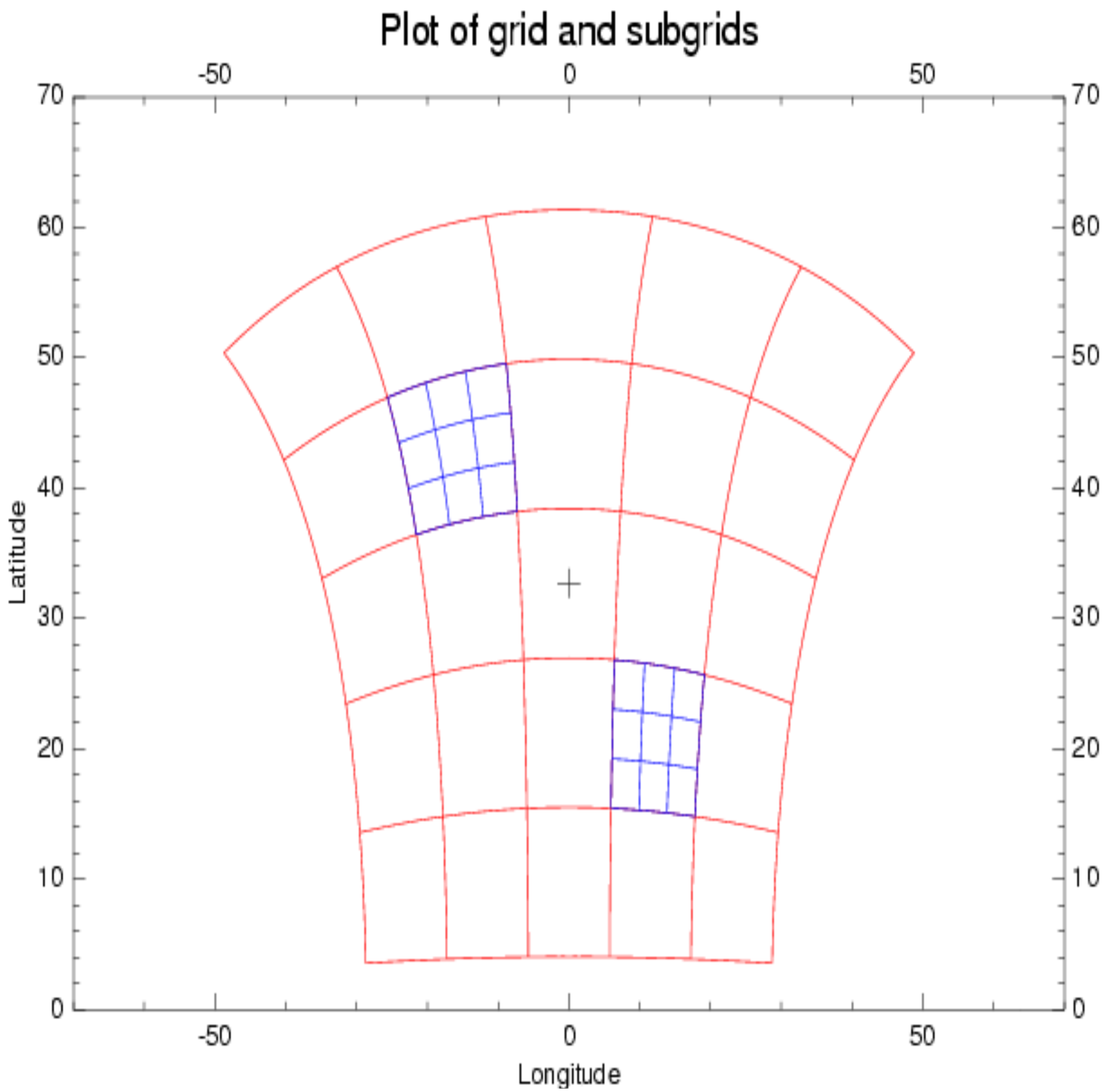
ARW Coarse Grid Staggering 3:1 Ratio

Feedback:
U : column
V : row
T : cell

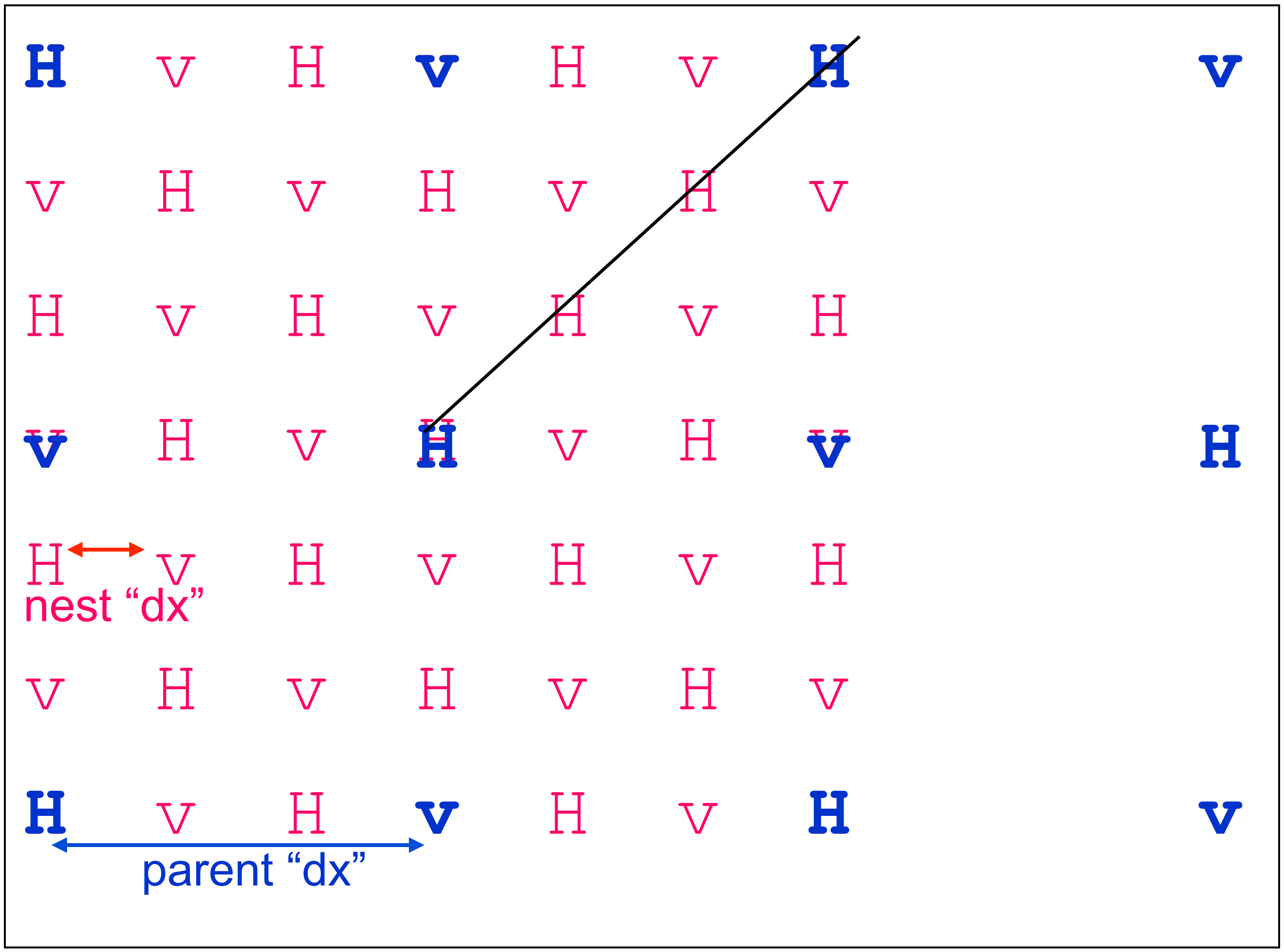




NMM Coarse/Fine Overlay

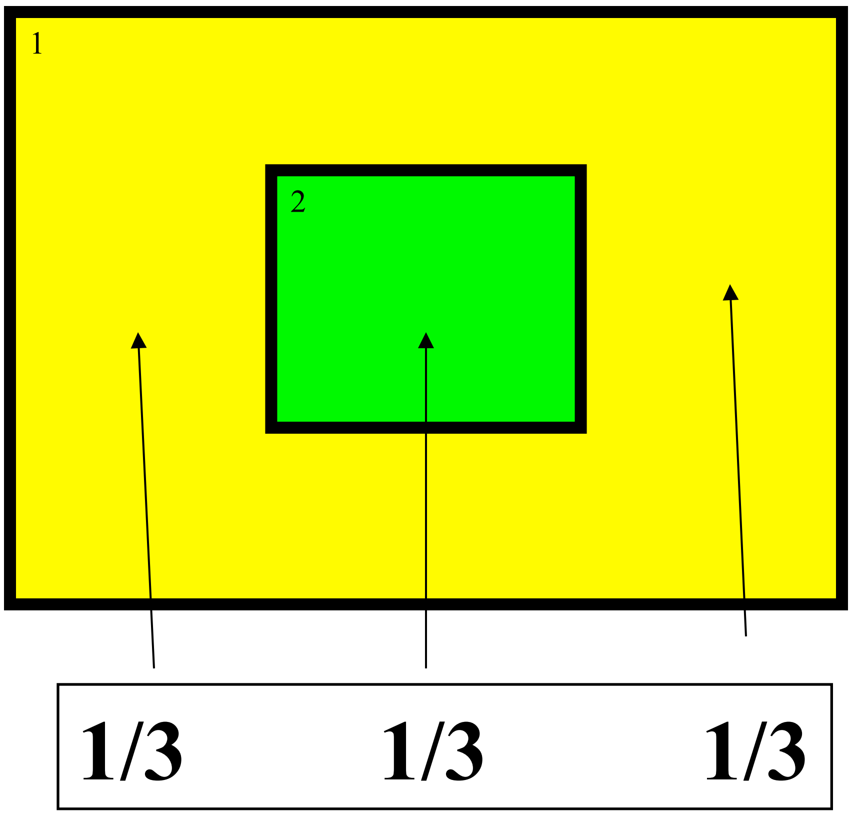


An odd grid ratio introduces parent/nest points being coincident, and a 3:1 ratio is preferred as it has been extensively tested.



Nest Location

- The minimum distance between the nest boundary and the parent boundary is FOUR grid cells
- You should have a **MUCH** larger buffer zone
- It is not unreasonable to have approximately 1/3 of your coarse-grid domain surrounding each side of your nest domain



Nesting Performance

- The size of the nested domain may need to be chosen with computing performance in mind.
- Assuming a 3:1 ratio and the same number of grid cells in the parent and nest domains, the fine grid will require 3x as many time steps to keep pace with the coarse domain.
- A simple nested domain forecast is approximately 4x the cost of just the coarse domain.
- Don't be *cheap* on the coarse grid, doubling the CG points results in only a 25% nested forecast time increase.

Nesting Performance

- Example: assume 3:1 nest ratio

If the nest has the same number of grid cells, then the **amount of CPU** to do a single time step for a coarse grid (CG) and a fine grid step (FG) is **approximately the same**.

Since the fine grid (3:1 ratio) has 1/3 the grid distance, it requires 1/3 the model time step. Therefore, the FG requires 3x the CPU to catch up with the CG domain.

Nesting Performance

- Example: assume 3:1 nest ratio

If you try to cover the SAME area with a FG domain as a CG domain, you need **(ratio)^2 grid points**.

With the associated FG time step ratio, you require a **(ratio)^3**.

With a 3:1 ratio, a FG domain covering the same area as a CG domain **requires 27x CPU**.

Nesting Performance

- Example: assume **10:1 nest ratio**

To change your test case from 50-km resolution to a finer 5-km resolution would be **1000x more** expensive.

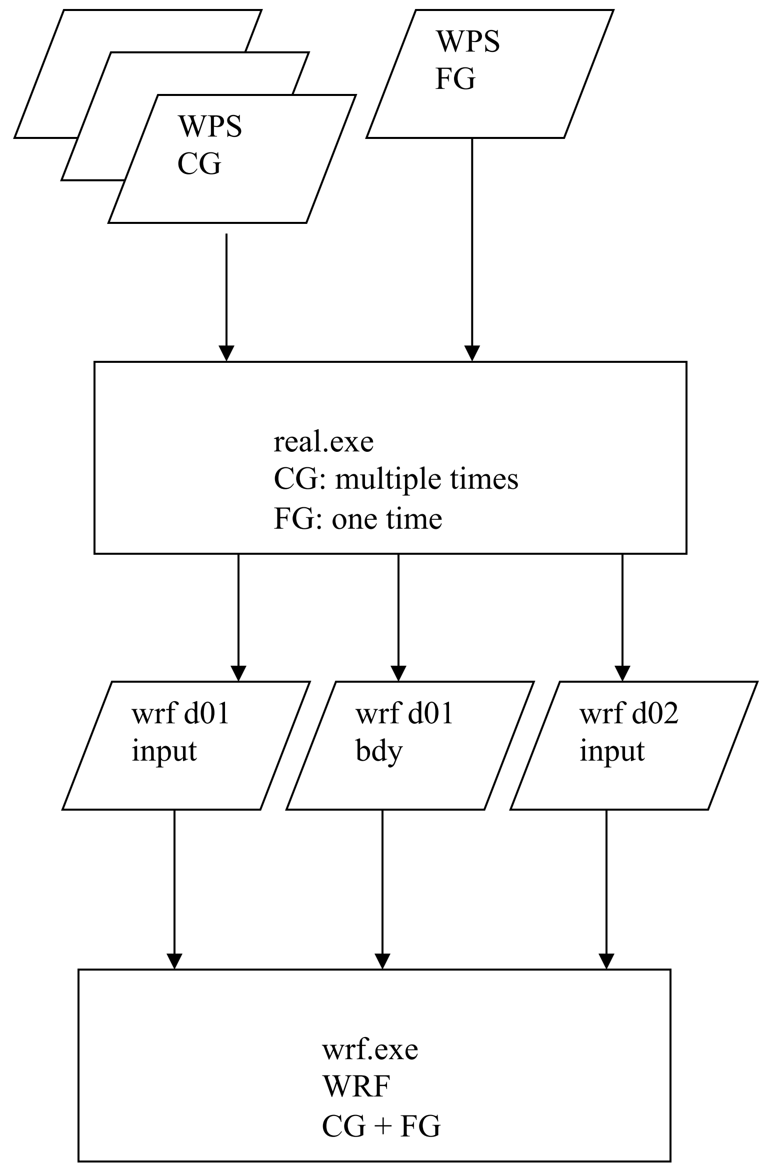
ARW: 2-Way Nest with 2 Inputs

Coarse and fine grid domains must start at the same time, fine domain may end at any time

Feedback may be shut off to produce a 1-way nest (cell face and cell average)

Any integer ratio for coarse to fine is permitted, odd is usually chosen for real-data cases

Options are available to ingest only the static fields from the fine grid, with the coarse grid data horizontally interpolated to the nest



ARW: 2-Way Nest with 2 Inputs

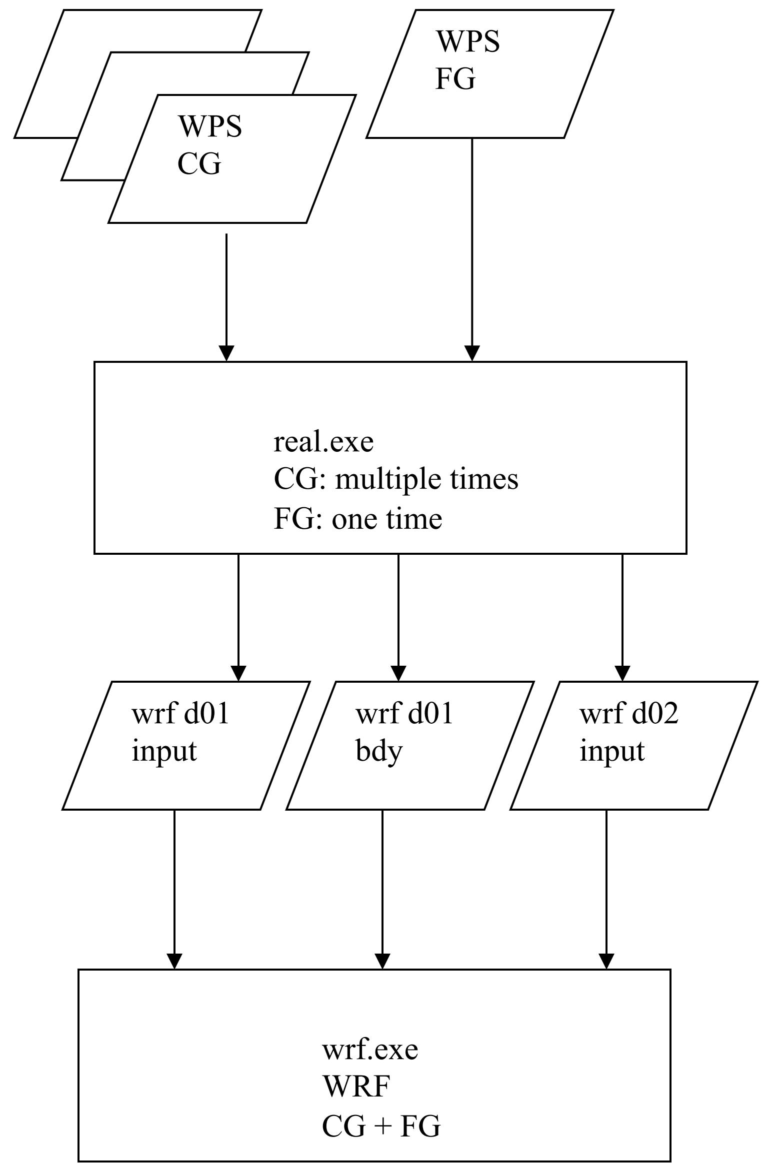
No vertical nesting

Usually the same physics are run on all of the domains (excepting cumulus)

The grid distance ratio is not strictly tied to the time step ratio

Topography smoothly ramps from coarse grid to the fine grid along the interface along the nest boundary

All fine grids must use the nested lateral boundary condition



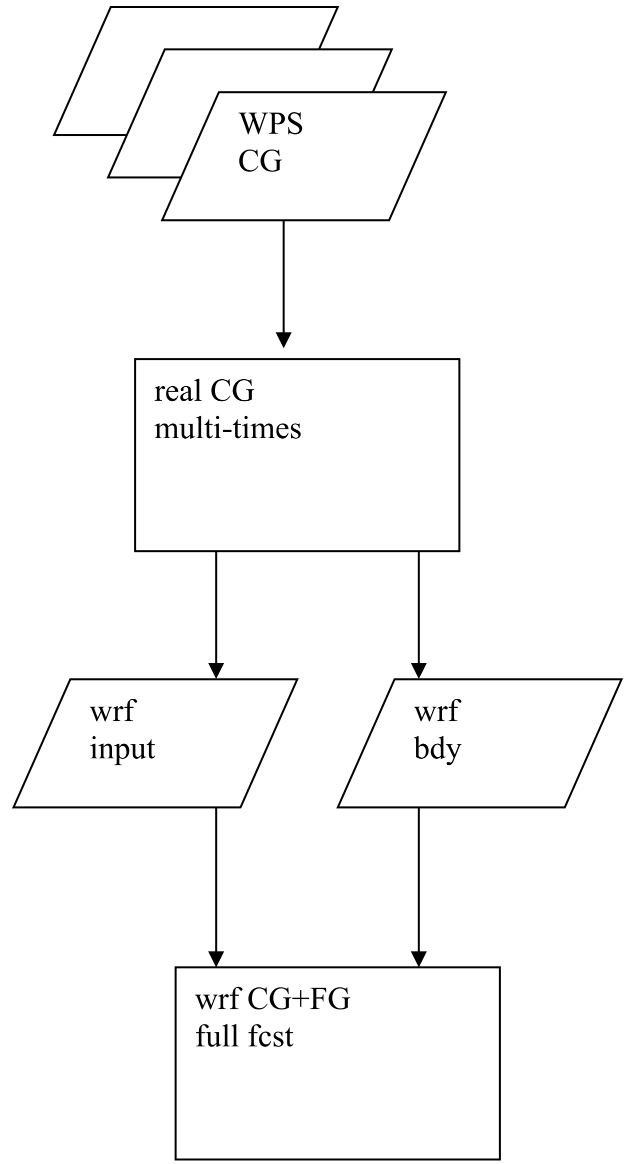
ARW: 2-Way Nest with 1 Input

A single namelist column entry is tied to each domain

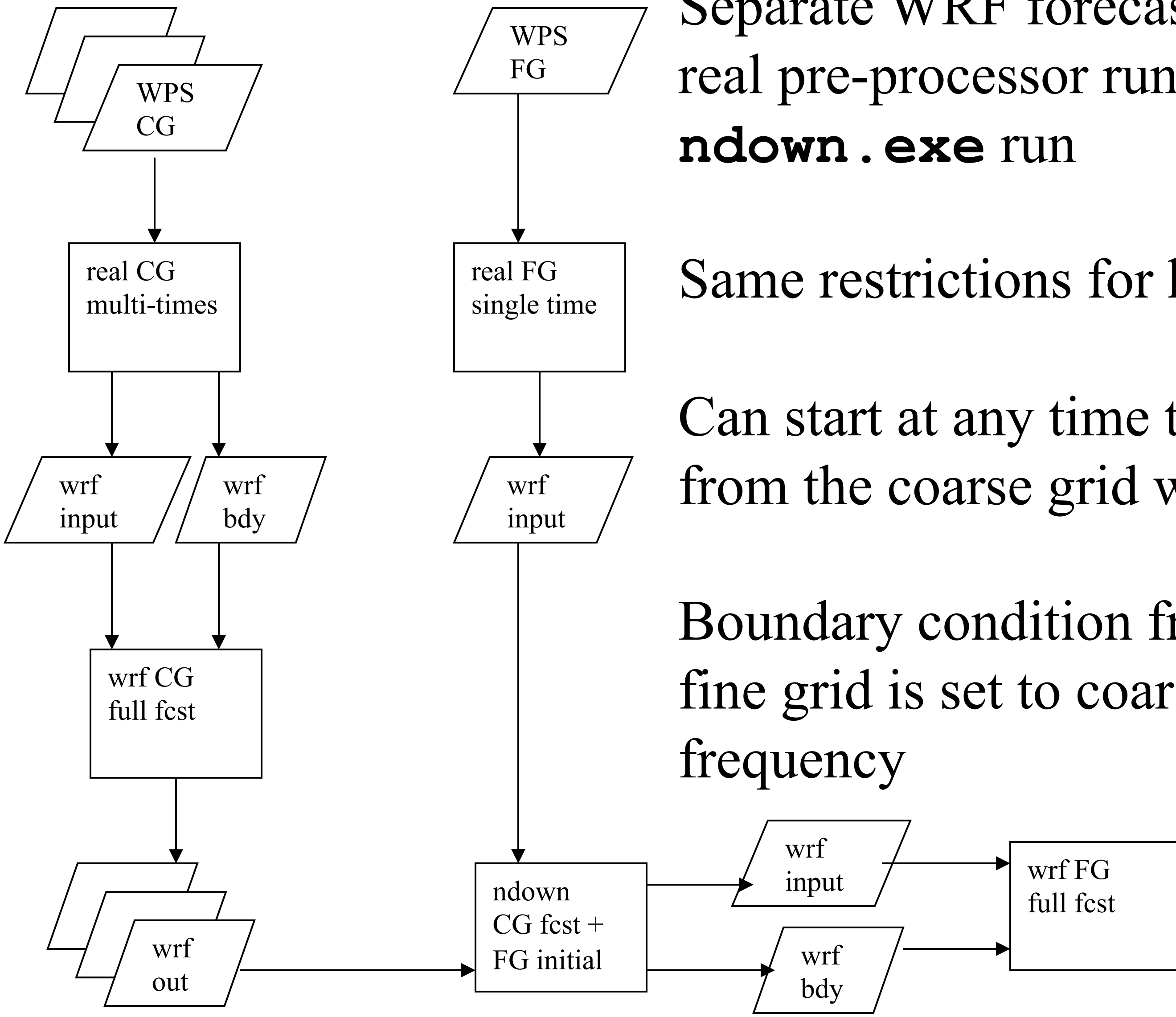
The horizontal interpolation method, feedback, and smoothing are largely controlled through the Registry file

For a 3:1 time step ratio, after the coarse grid is advanced, the lateral boundaries for the fine grid are computed, the fine grid is advanced three time steps, then the fine grid is fed back to the coarse grid (recursively, depth first)

Helpful run*.tar files are located in the **`./WRFV3/test/em_real`** directory



ndown: 1-Way Nest with 2 Inputs



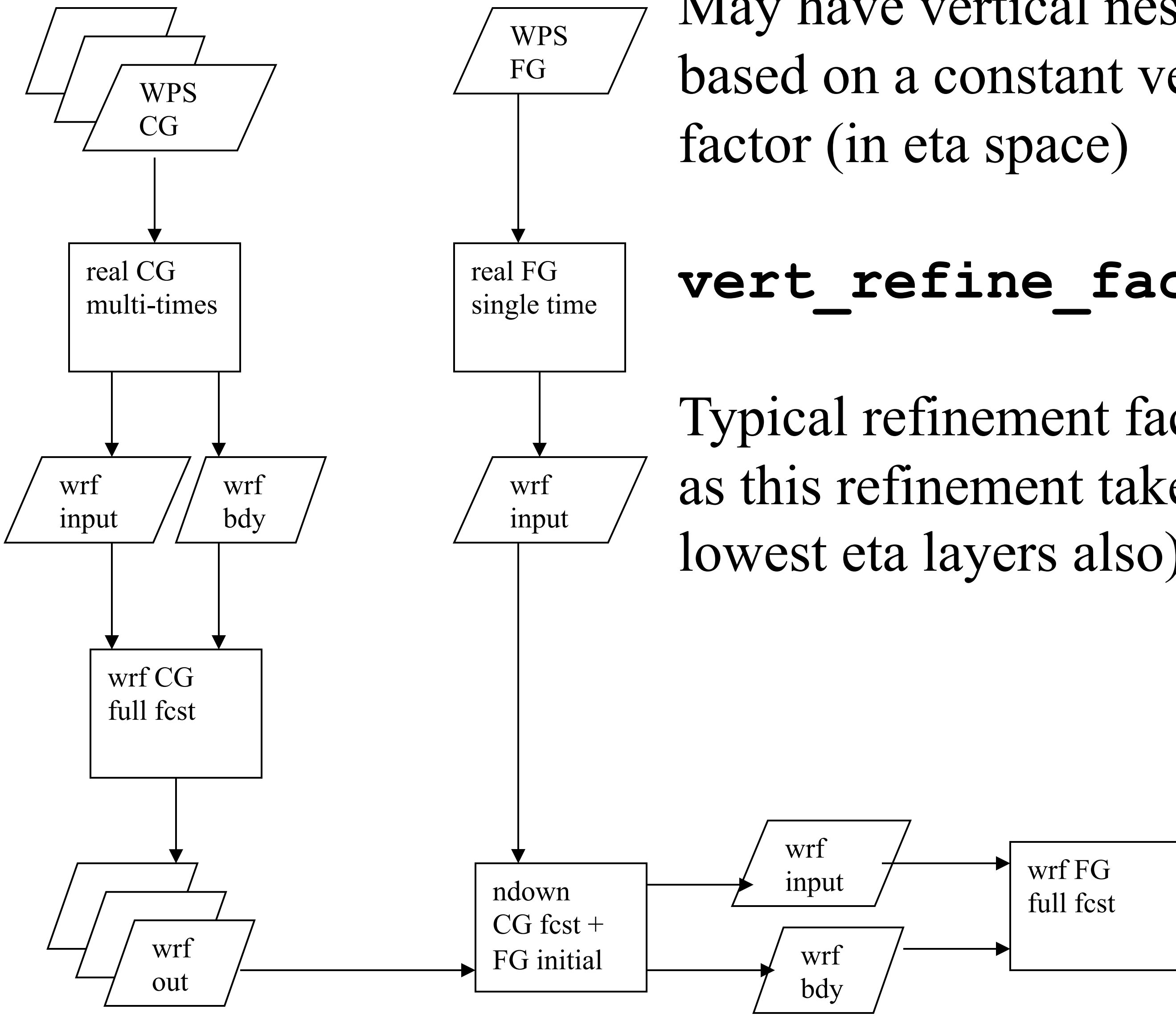
Separate WRF forecast runs, separate real pre-processor runs, intervening `ndown.exe` run

Same restrictions for horizontal nest ratios

Can start at any time that an output time from the coarse grid was created

Boundary condition frequency for the fine grid is set to coarse grid output frequency

ndown: 1-Way Nest with 2 Inputs



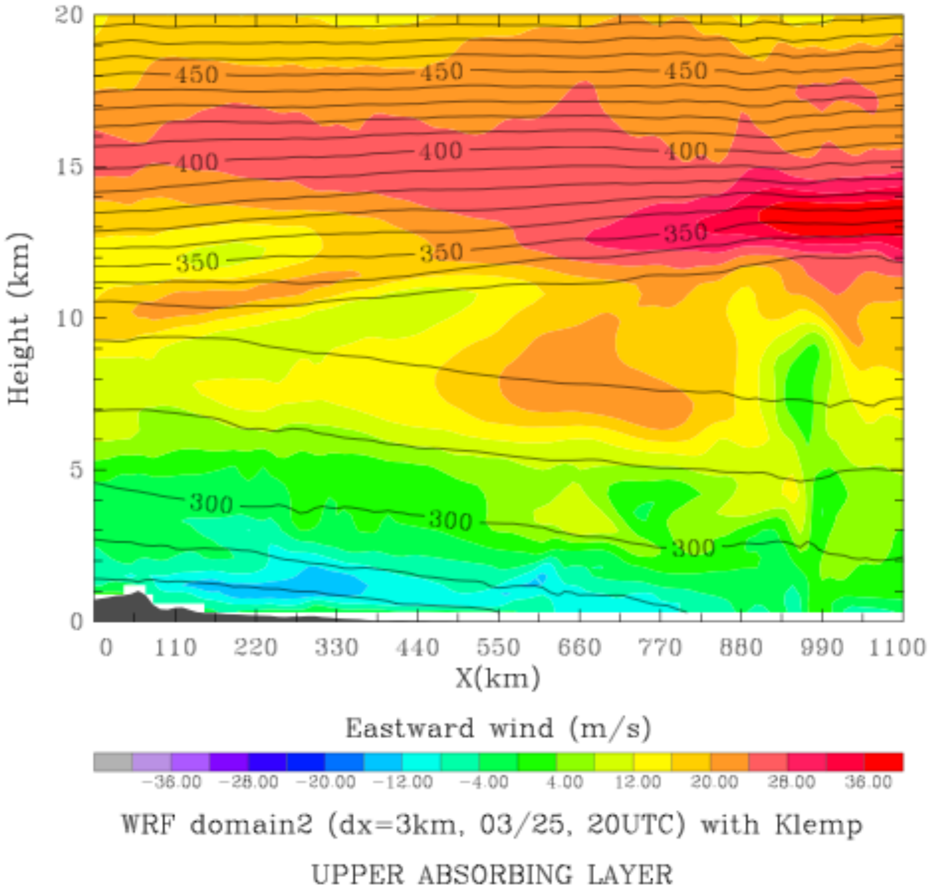
May have vertical nesting on the fine grid based on a constant vertical refinement factor (in eta space)

`vert_refine_fact`

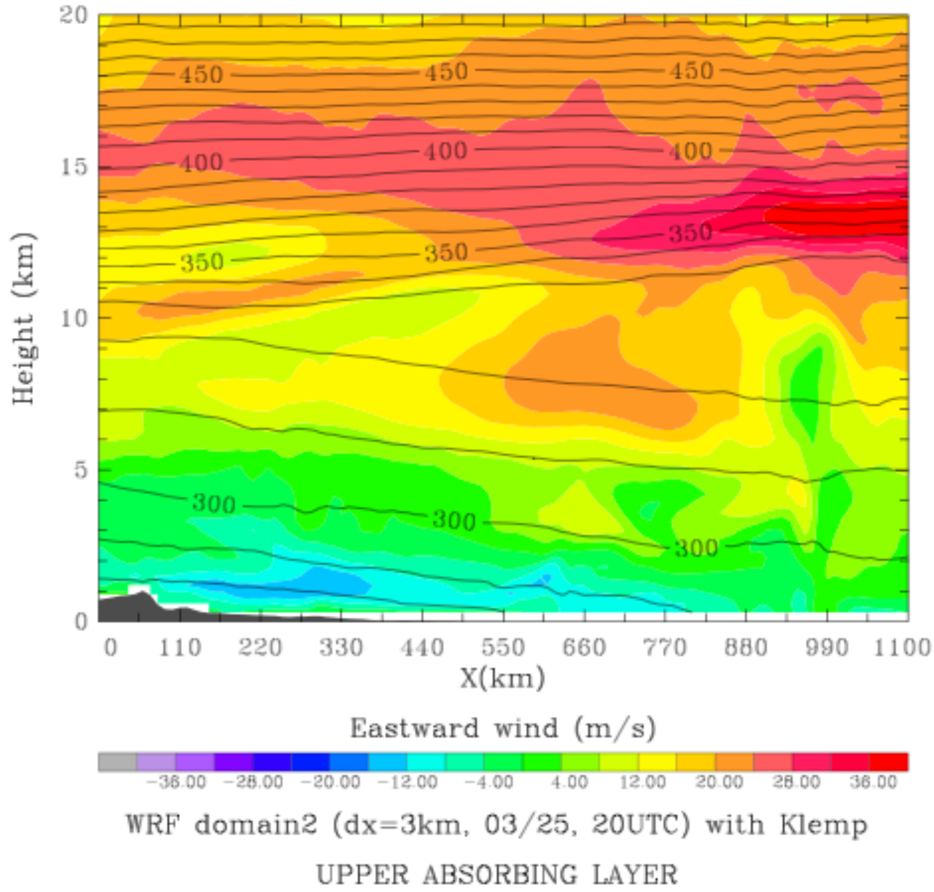
Typical refinement factors 2-5 (be careful, as this refinement takes place in the lowest eta layers also)

West East Cross section
Shaded: v; Contour: theta
6-h Forecast, from Mohamed Moustaoui

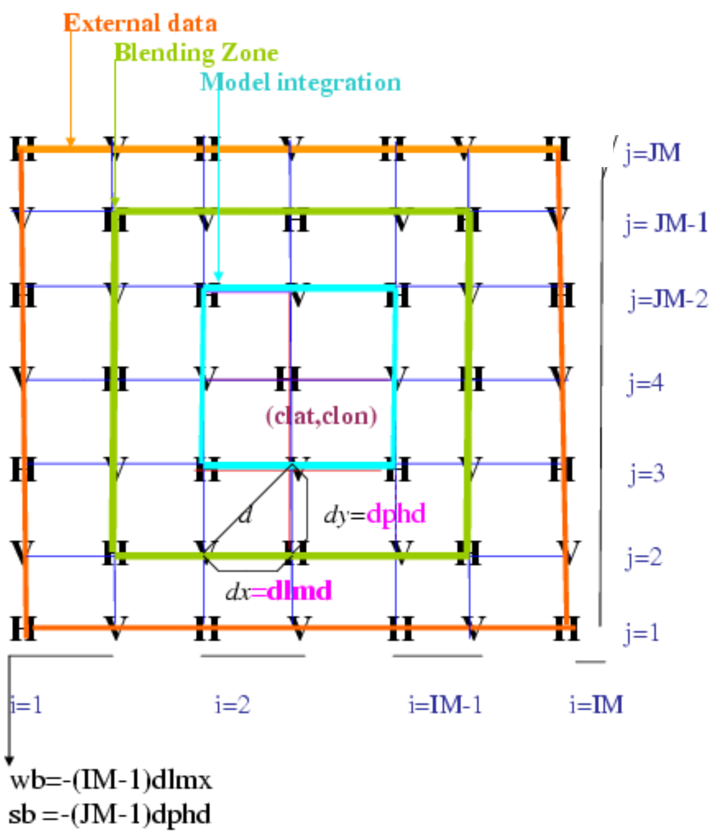
Standard Levels



3x Refinement



NMM Nested LBCs



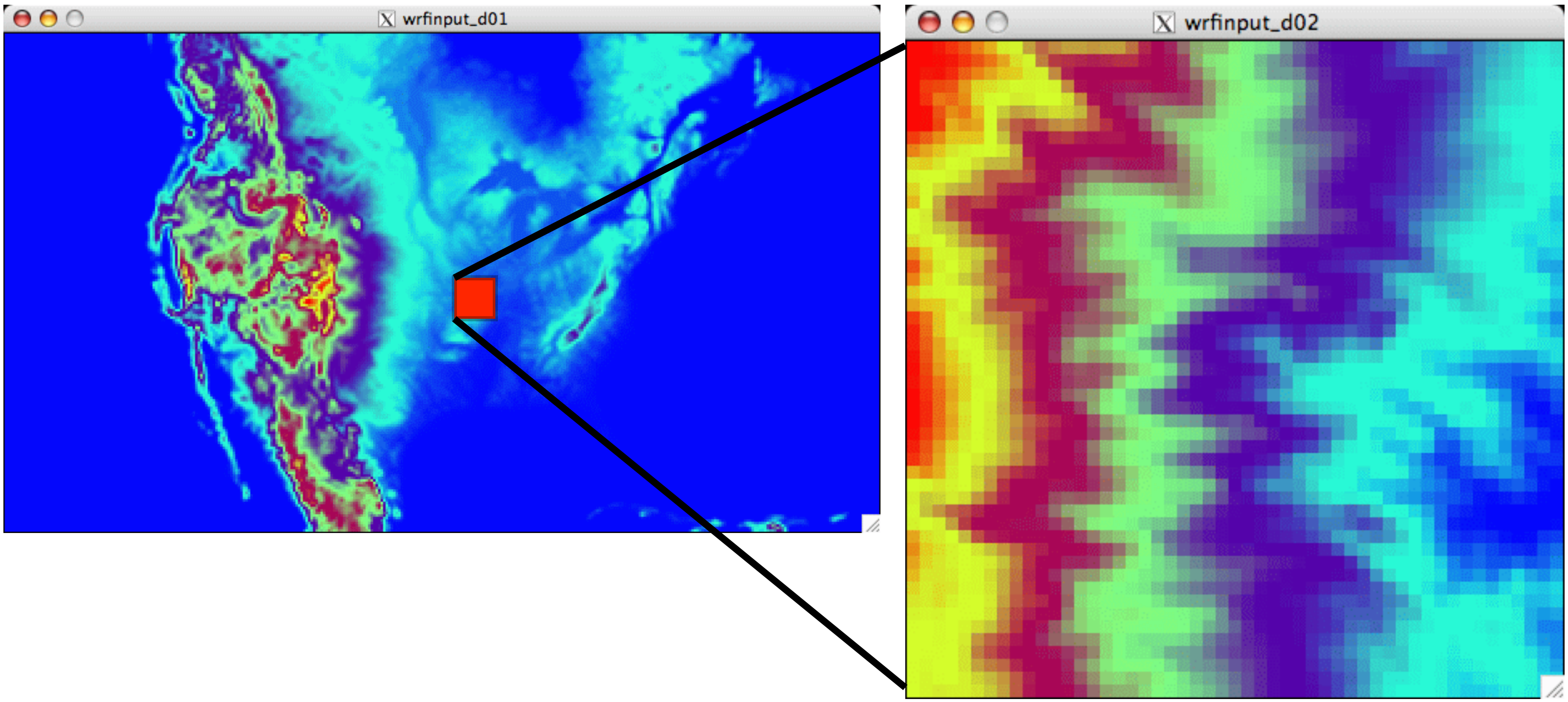
* Given wb,sb, clat and clon, the above rotated lat-lon grid system can be transformed to a lat-lon grid system.

- Nest boundaries generally are treated in the same way as the standard parent domain boundaries:
  - outermost row is prescribed
  - two rows in from boundary is freely integrating
  - in between is a blending zone (average of outermost and freely integrating points)
- The one key difference is frequency of boundary updates: *nested boundaries are updated at every time step of the parent domain.*

NMM Mass Balancing for LBCs

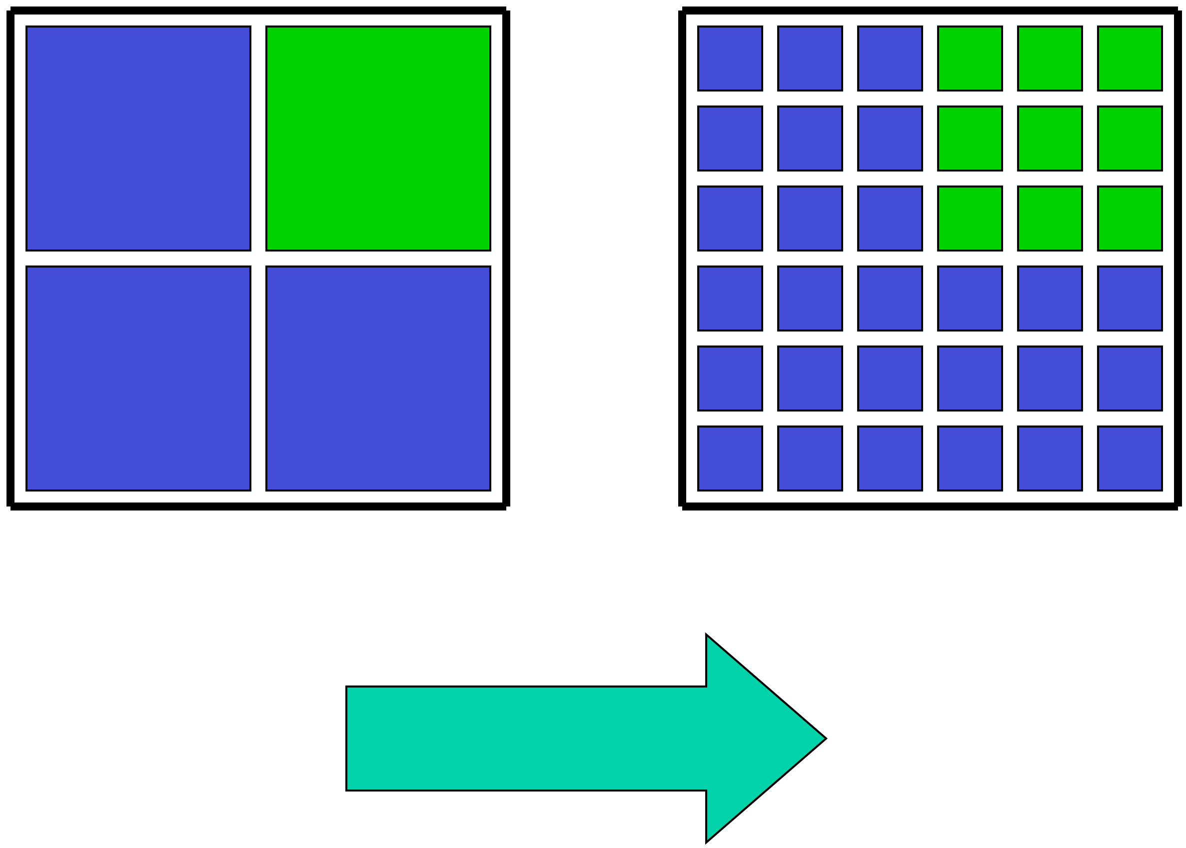
- The parent domain geopotential height, temperature, and moisture are all vertically interpolated (cubic splines) from the hybrid surfaces onto standard isobaric levels.
- Using horizontally interpolated information of the height field from the parent domain, and high-resolution topography from the nest level, mass is adjusted and revised hybrid surfaces are constructed.
- T and q: 1) horizontally interpolated to the nest domain on standard pressure levels, 2) vertically interpolated onto the new hybrid surfaces
- Approach produces an effective way of updating the nest interface without much distortion or noise

Intermediate Domains

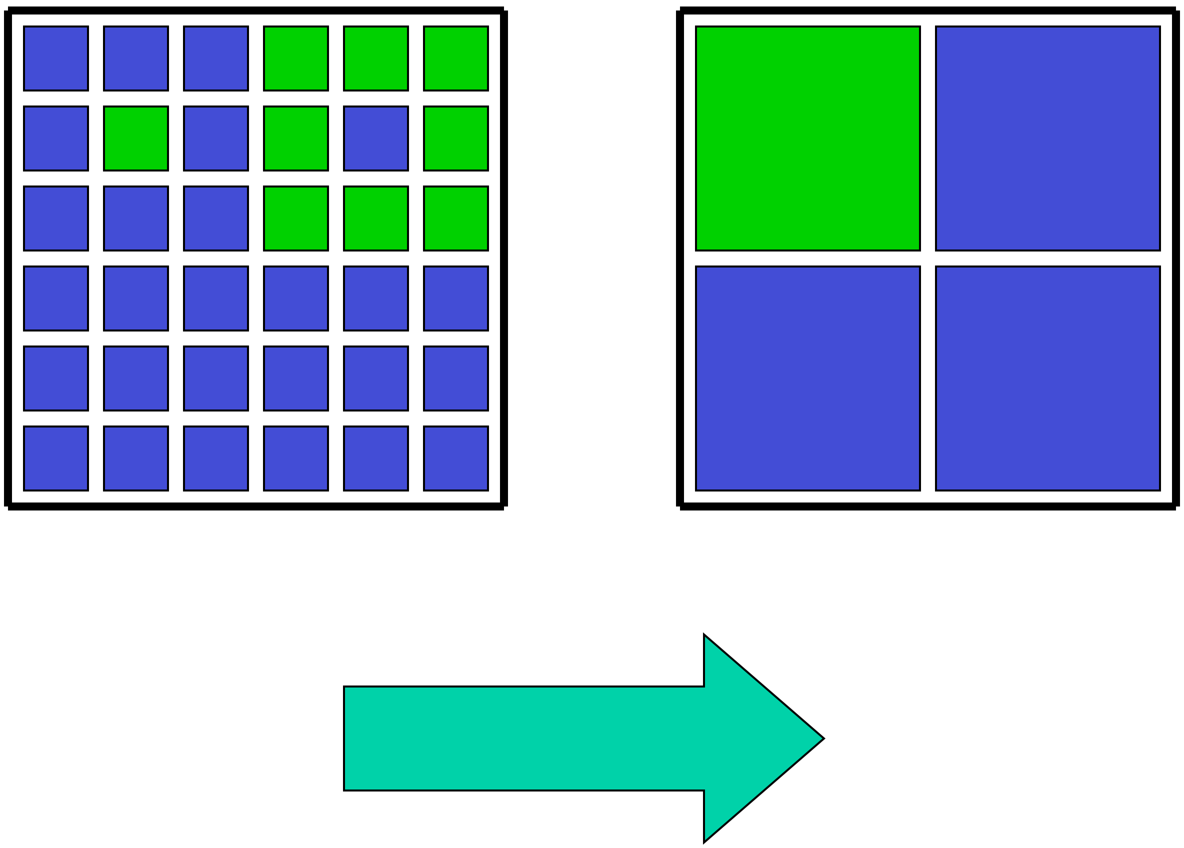


The intermediate domain between a parent and a child is the resolution of the coarse grid over the size of the fine grid. It allows the model to re-decompose the domain among all of the processors.

ARW Masked Interpolation



ARW Masked Feedback



What are those “usdf” Options

```
state real u ikjb dyn_em 2 X \
  i01rhusdf=(bdy_interp:dt) \
  "U" "x-wind component" "m s-1"
```

“f” defines what lateral boundary forcing routine (found in **share/interp_fcn.F**) is utilized, colon separates the additional fields that are required (fields must be previously defined in the Registry)

What are those “usdf” Options

```
state real landmask ij misc 1 -  \
  i012rhd=(interp_fcnm)u=(copy_fcnm)\
  "LANDMASK" "LAND MASK (1=LAND, 0=WATER)"
```

“u” and “d” define which feedback (up-scale) and horizontal interpolation (down-scale) routines (found in **`share/interp_fcn.F`**) are utilized

Default values (i.e. not a subroutine name listed in the parentheses) assume non-masked fields

At compile-time, users select options

What are those “usdf” Options

```
state real ht ij  misc  1  - i012rhdus  "HGT" \
  "Terrain Height"   "m"
```

“s” if the run-time option for smoothing is activated, this field is to be smoothed - only used for the parent of a nest domain, smoothing is in the area of the nest, excluding the outer row and column of the nest coverage

Whether or not smoothing is enabled is a run-time option from the namelist

Special IO Stream #2 Fields

```
state real msft ij  misc  1 -  \   i012rhdu=
   (copy_fcnm)  "MAPFAC_M"   \
  "Map scale factor on mass grid" ""

state real msfu ij  misc  1 X  \  i012rhdu=
   (copy_fcnm)  "MAPFAC_U"   \
  "Map scale factor on u-grid" ""

state real msfv ij  misc  1 Y  \   i012rhdu=
   (copy_fcnm)  "MAPFAC_V"   \
  "Map scale factor on v-grid" ""
```

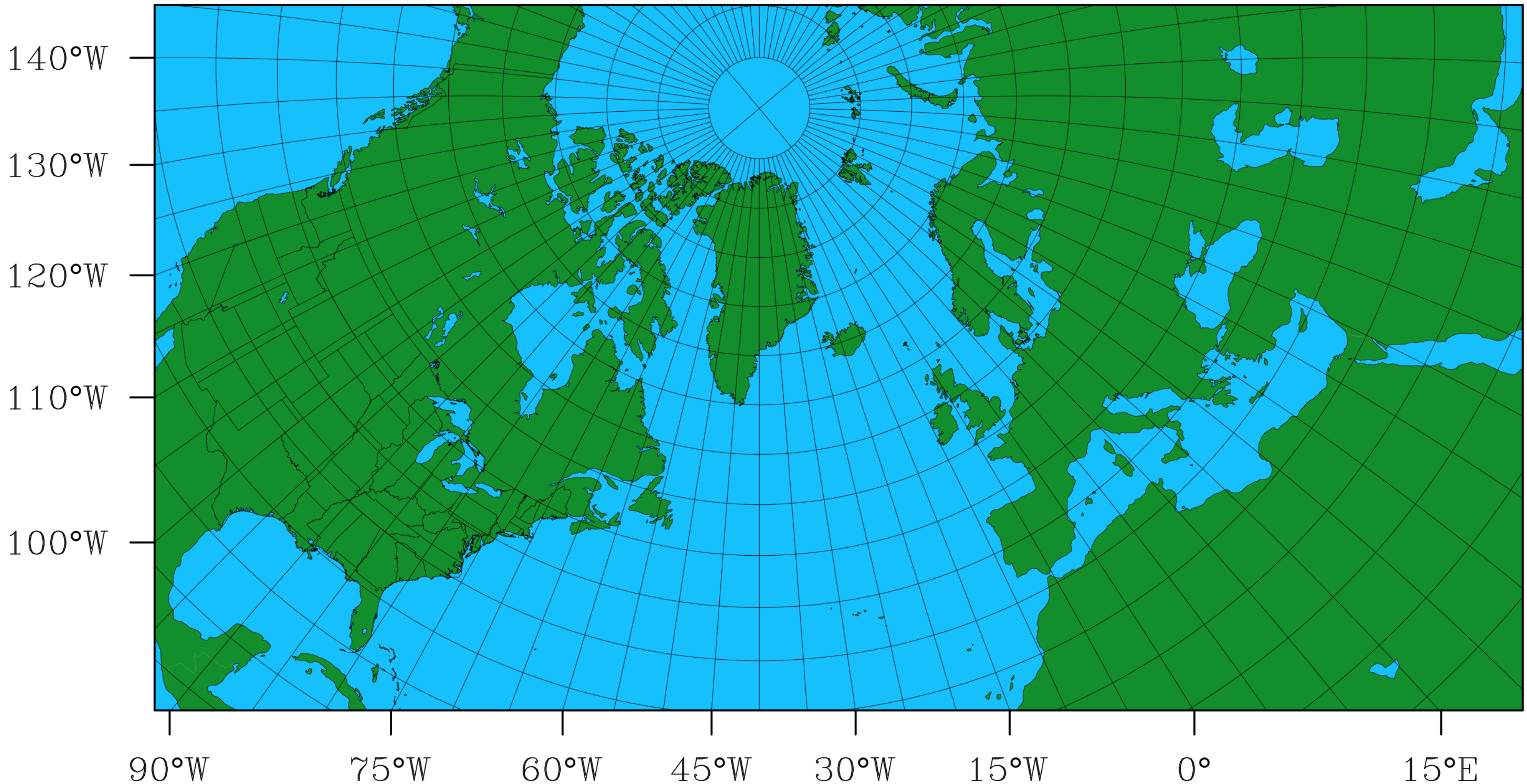
Nesting Suggestions

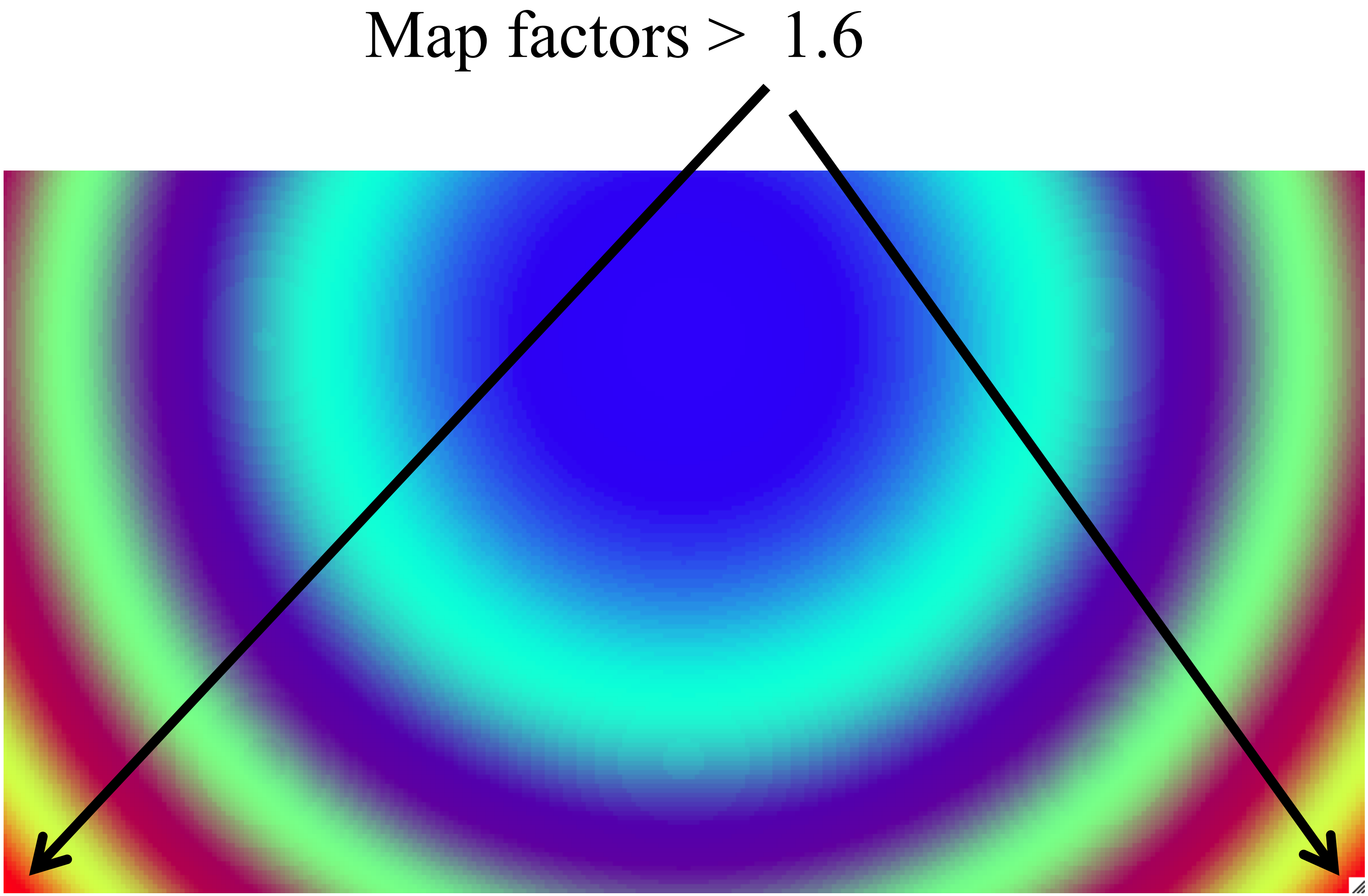
- Start with designing your inner-most domain. For a traditional forecast, you want everything important for that forecast to be entirely contained inside the domain.
- Then start adding parent domains at a 3:1 or 5:1 ratio. A parent should not have a smaller size (in grid points). Keep adding domains until the most coarse WRF grid has a no more than a 3:1 to 5:1 ratio to the external model (first guess) data.

Nesting Suggestions

- Larger domains tend to be better than smaller domains.
- A 60 m/s parcel moves at > 200 km/h. A 2-km resolution grid with 100x100 grid points could have most of the upper-level initial data swept out of the domain within a couple of hours.

Nesting Suggestions




Map factors > 1.6

Nesting Suggestions

- The most-coarse domain may have a geographic extent that causes large map factors.

```
time_step              = 300 (BLOWS UP)
dx                     = 45000,15000,5000
grid_id                = 1,    ,2     ,3
parent_id              = 0,    ,1     ,2
parent_grid_ratio      = 1,    ,3     ,3
parent_time_step_ratio = 1,    ,3     ,3
```

Nesting Suggestions

- Reducing the time step so that the coarse grid is stable makes the model too expensive. 1.6x

```
time_step              = 180 (STABLE, PRICEY)
dx                     = 45000,15000,5000
grid_id                = 1,    ,2     ,3
parent_id              = 0,    ,1     ,2
parent_grid_ratio      = 1,    ,3     ,3
parent_time_step_ratio = 1,    ,3     ,3
```

Nesting Suggestions

- Only reduce the time step on the coarse grid, and keep the fine grid time steps at their approx original values.

```
time_step                = 180 (STABLE, CHEAP)
dx                       = 45000,15000,5000
grid_id                  = 1,    ,2    ,3
parent_id                = 0,    ,1    ,2
parent_grid_ratio        = 1,    ,3    ,3
parent_time_step_ratio   = 1,    ,2    ,3
```

Nesting Suggestions

- Model time step is always proportional to the time step of the most coarse grid.
- The coarse grid is the only grid impacted with large map factors: dt(s) = 6*dx(km) but the nominal grid distance needs to be scaled: dt(s) = dx(km) / MAX (map factor)
- Reducing the coarse grid time step does not significantly reduce model performance if you can tweak the time step ratio.

Nesting Suggestions

- The time step ratio and grid distance ratio are not necessarily identical, and may used effectively when large map factors in the coarse grid domain force a time step reduction for stability.
- If map factors are causing stability troubles, it is usually only the most coarse grid that is impacted.

Nesting Suggestions

- Set up domain first to provide good valid forecast, then deal with efficiency
- Selecting a set of domains with the reason "it is all I can afford" gets you into trouble
- Numerically stable and computationally expedient do not imply scientifically or physically valid