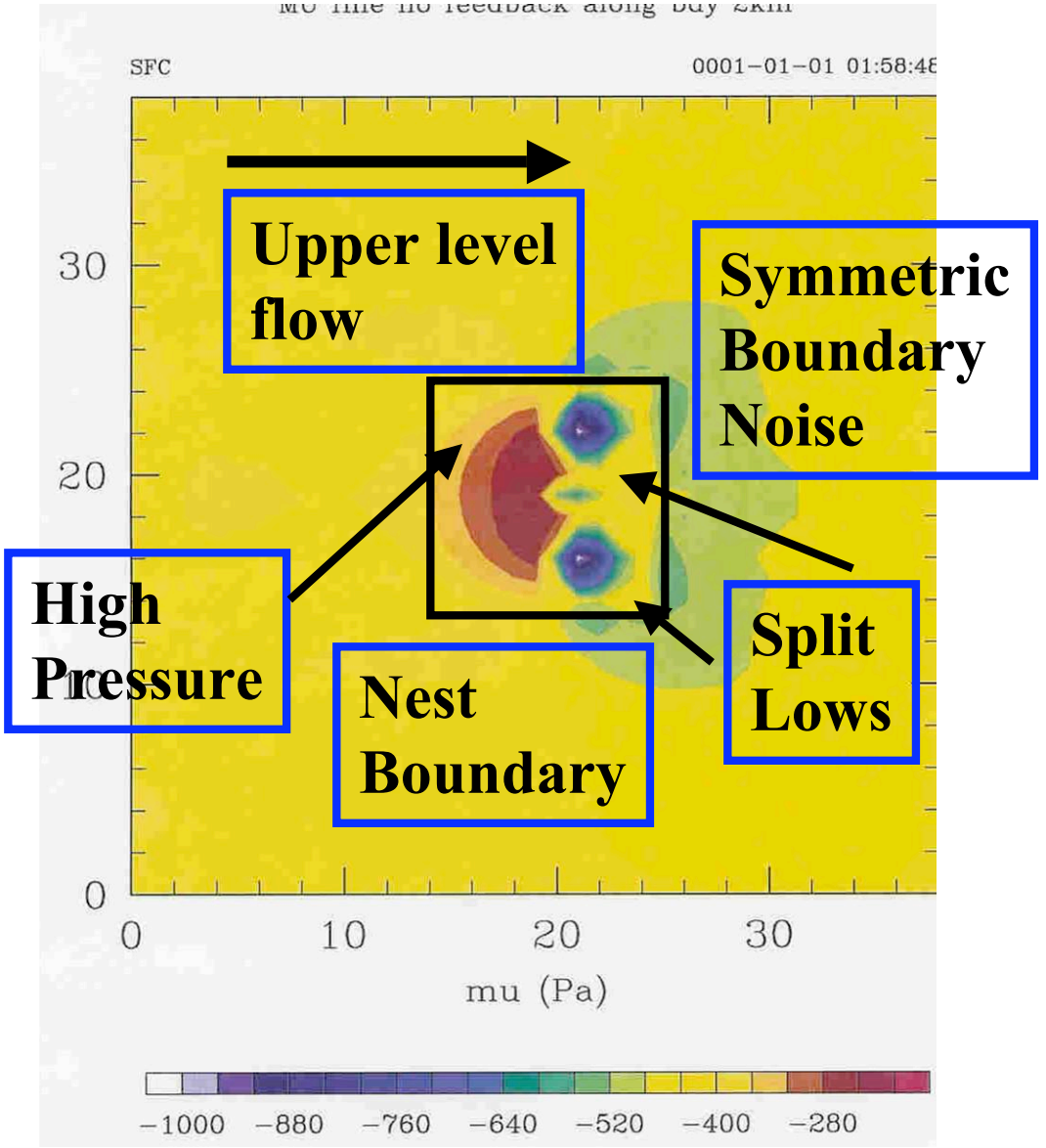
ARW Nesting

Dave Gill
gill@ucar.edu

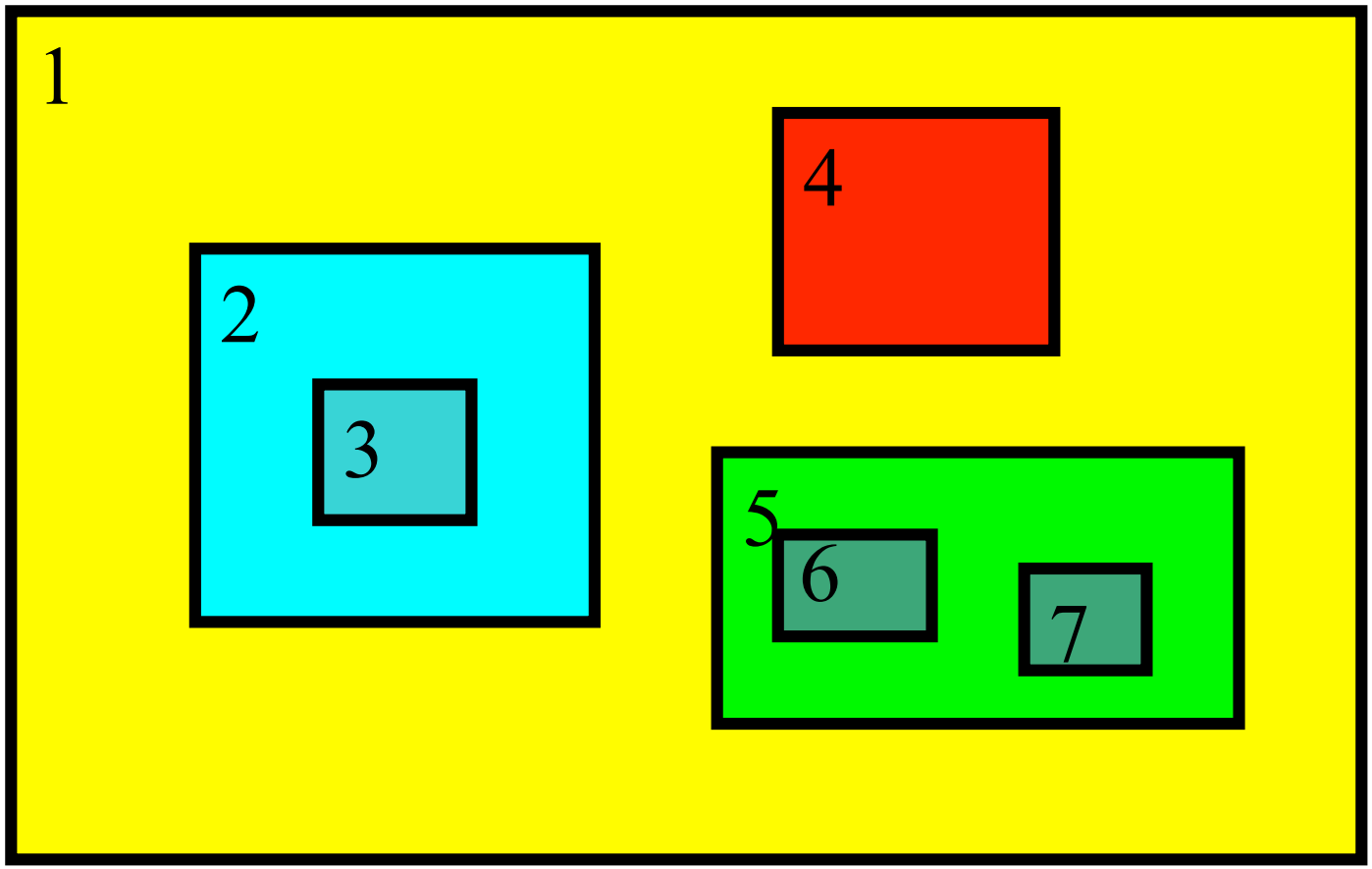


Some Nesting Hints

- Allowable domain specifications
- Defining a starting point
- Illegal domain specifications
- 1-way *vs* 2-way nesting

These are all OK

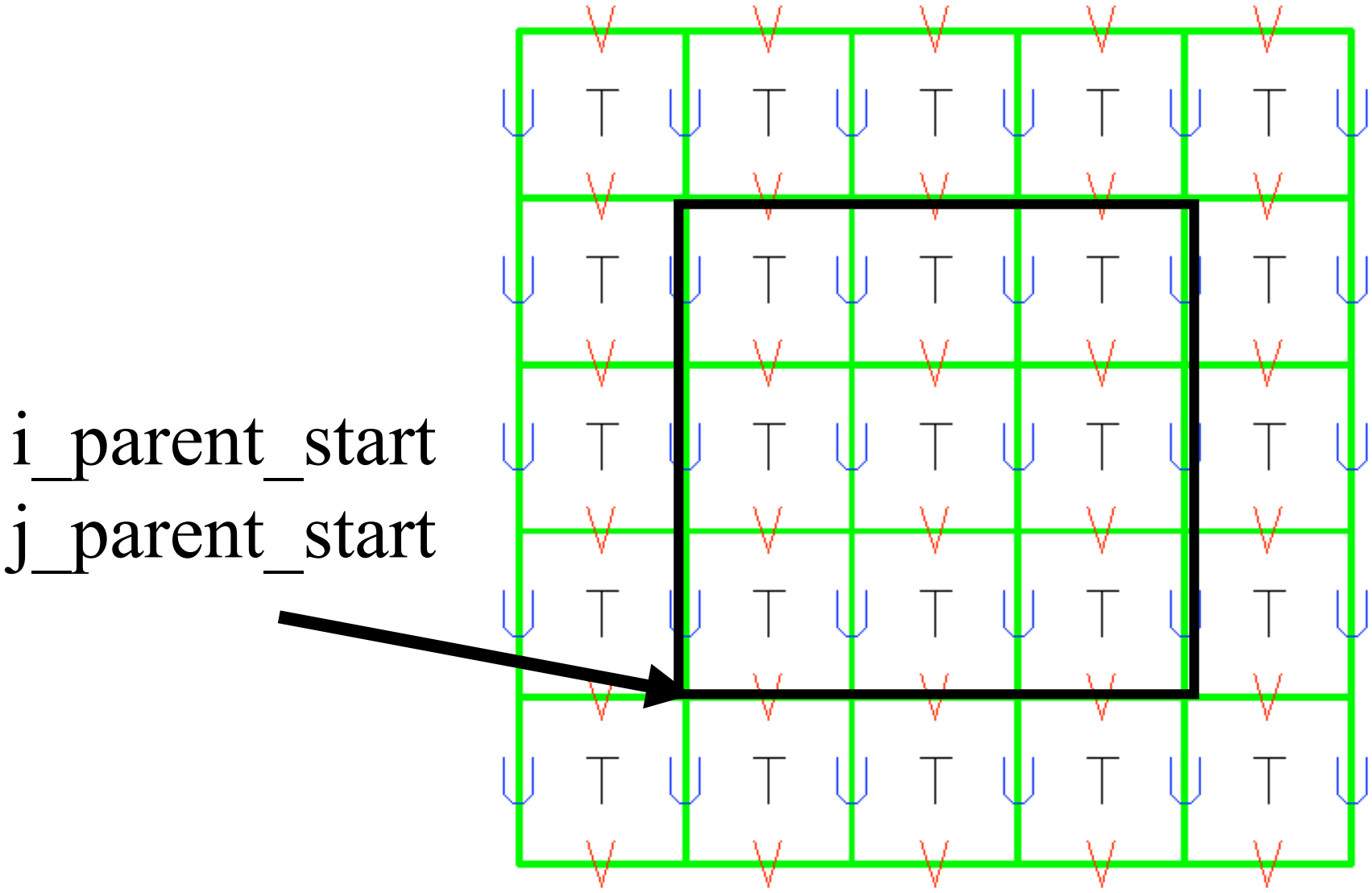
Telescoped to any depth
Any number of siblings



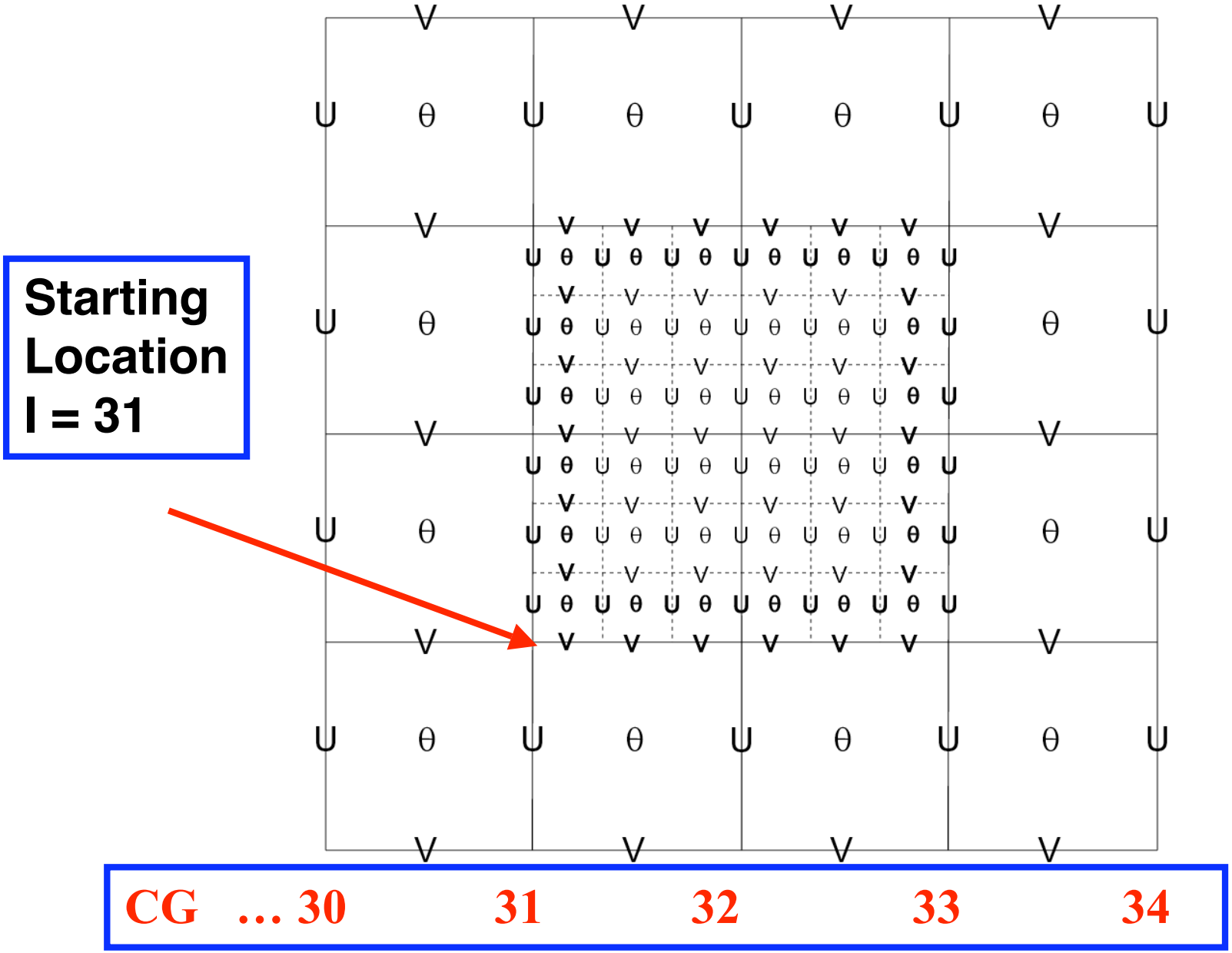
Some Nesting Hints

- Allowable domain specifications
- Defining a starting point
- Illegal domain specifications
- 1-way *vs* 2-way nesting

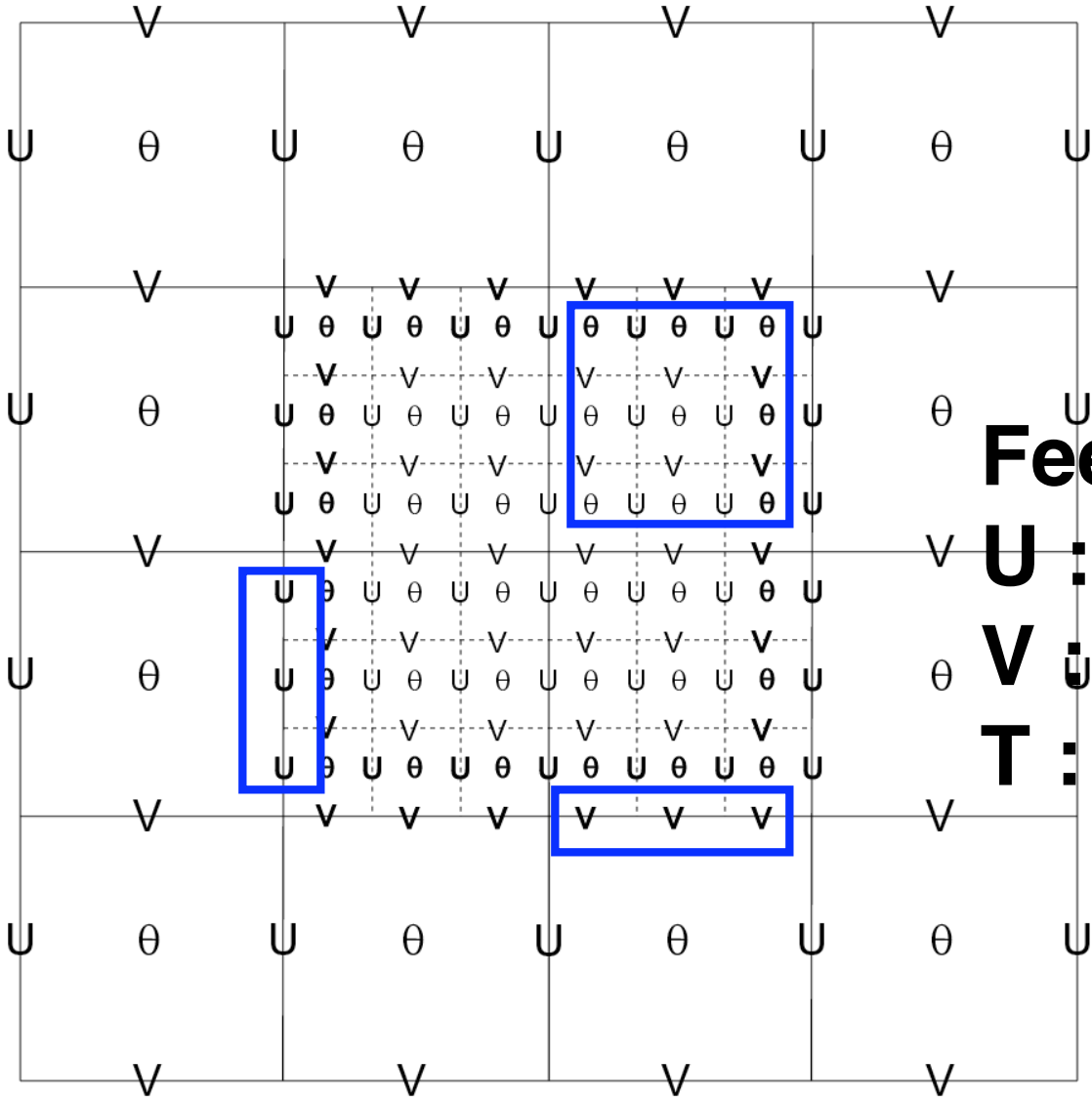
Coarse Grid Staggering



Coarse Grid Staggering 3:1 Ratio



Coarse Grid Staggering 3:1 Ratio



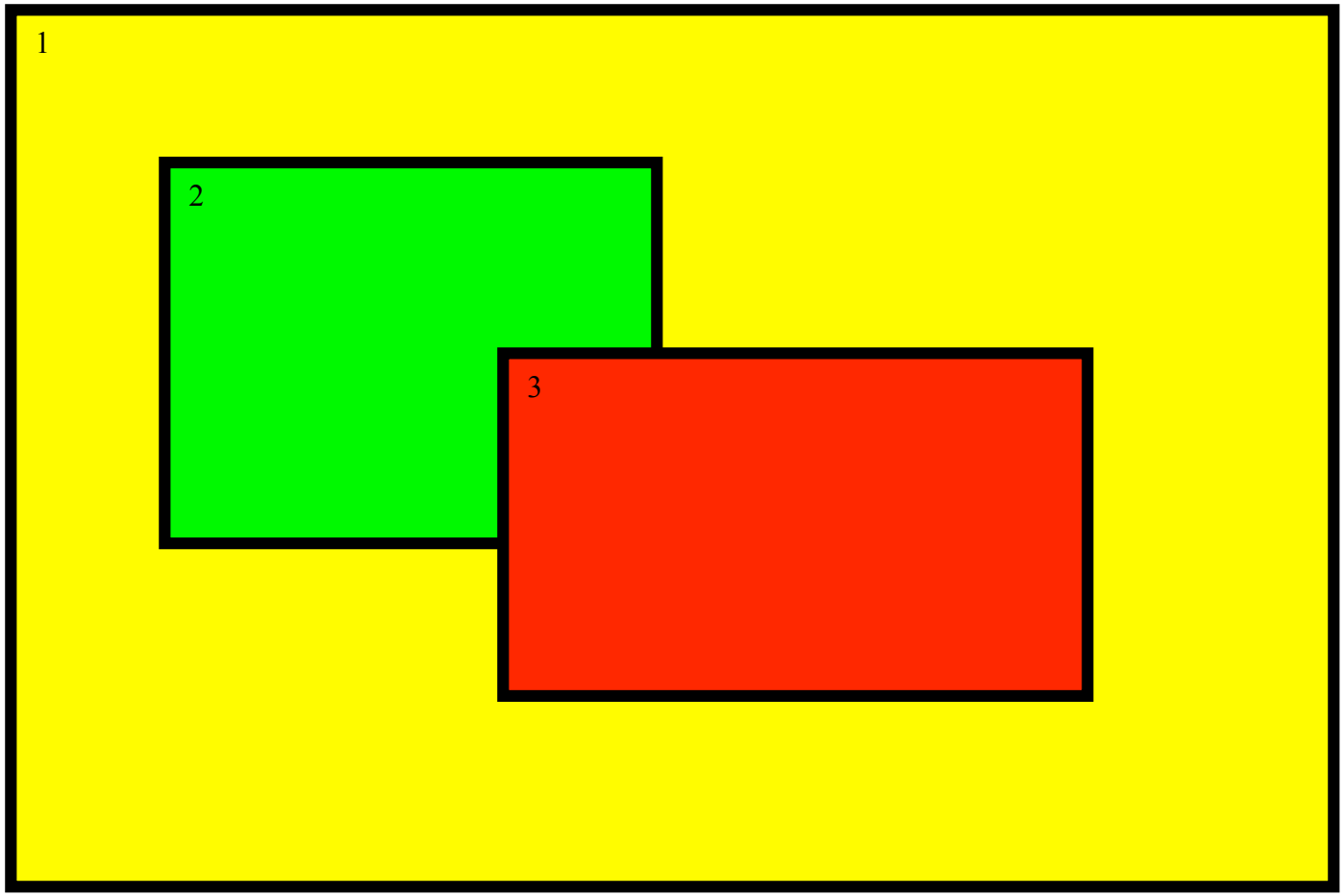
Feedback:
U : column
V : row
T : cell

Some Nesting Hints

- Allowable domain specifications
- Defining a starting point
- Illegal domain specifications
- 1-way *vs* 2-way nesting

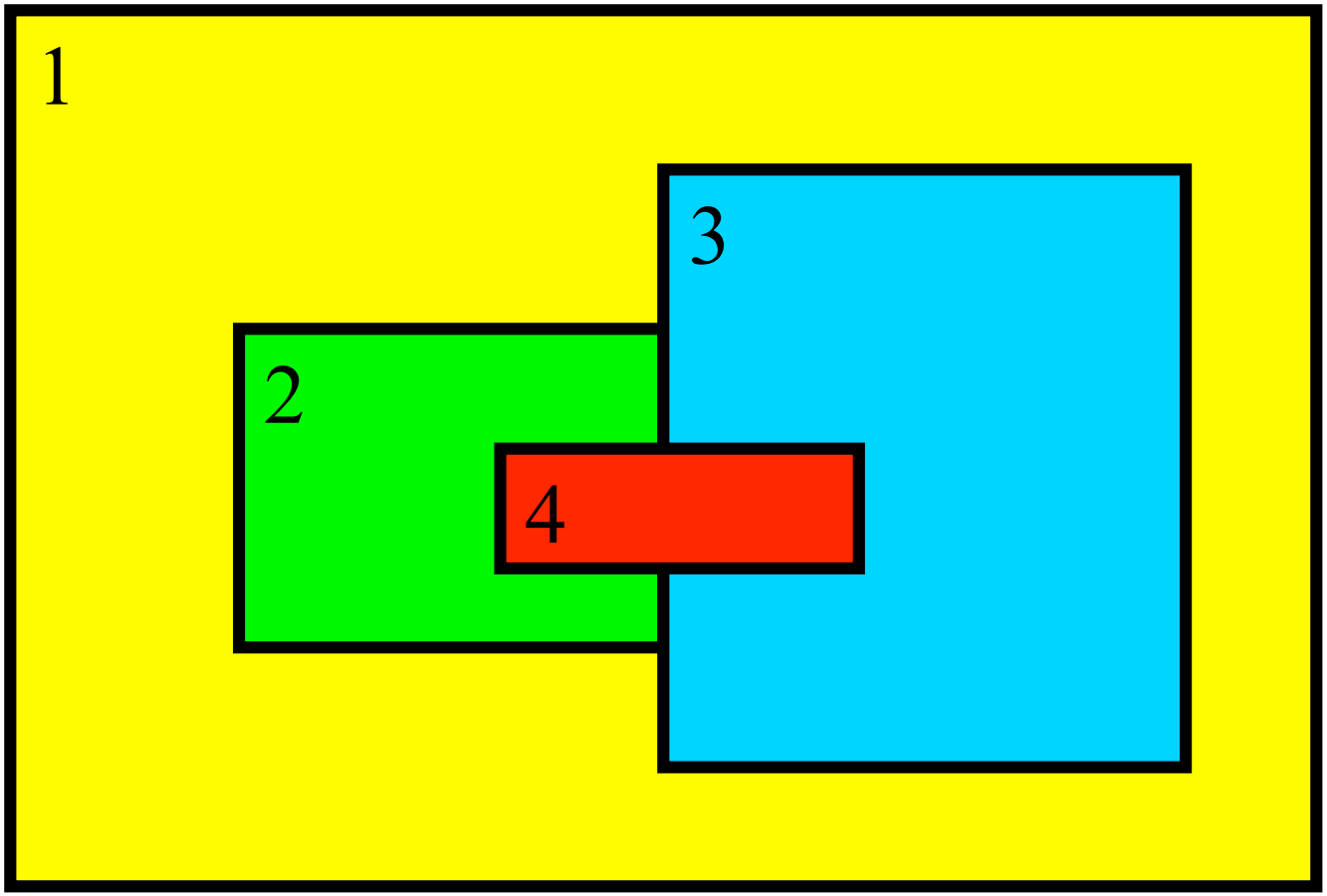
Not OK

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)



Some Nesting Hints

- Allowable domain specifications
- Defining a starting point
- Illegal domain specifications
- 1-way *vs* 2-way nesting

ndown vs real

- wrf integrates 1 domain at a time
- CG forces FG through lateral boundaries
- No FG to CG feedback
- ndown run between CG wrf and FG wrf (or shut off feedback)

- wrf integrates 2 domains at a time
- CG forces FG at every FG timestep
- FG to CG feedback at every CG timestep
- ndown not required

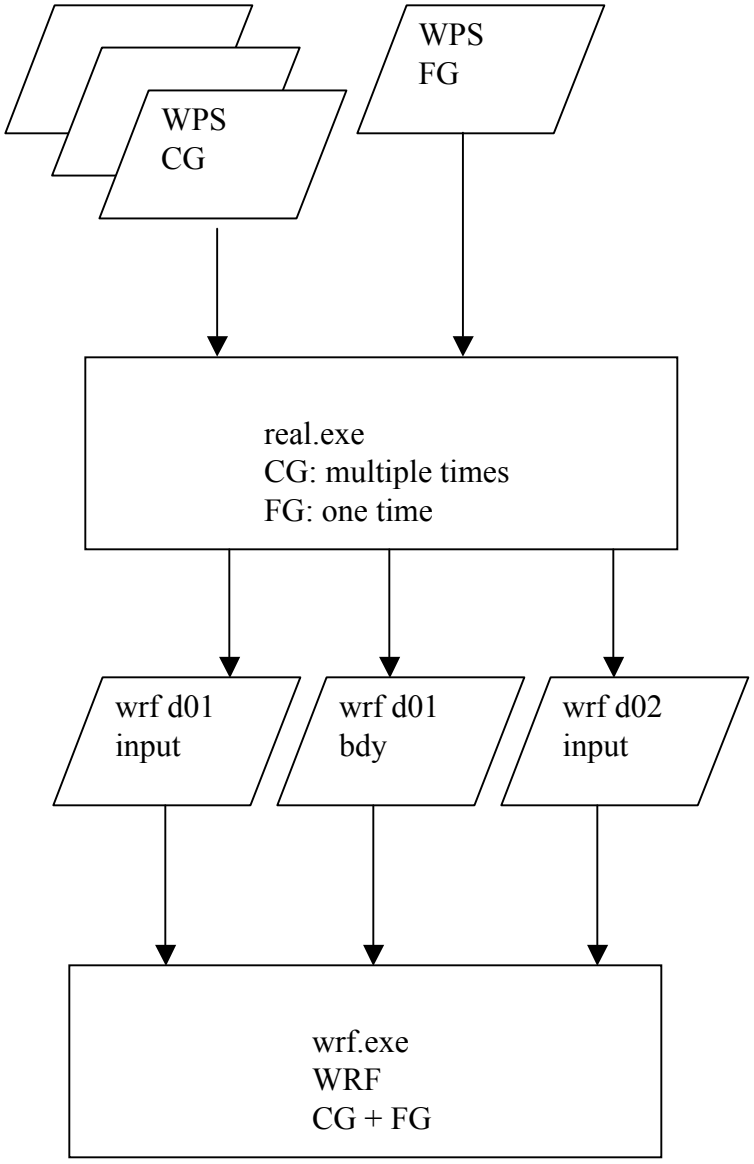
2-Way Nest with 2 Inputs

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



2-Way Nest with 2 Inputs

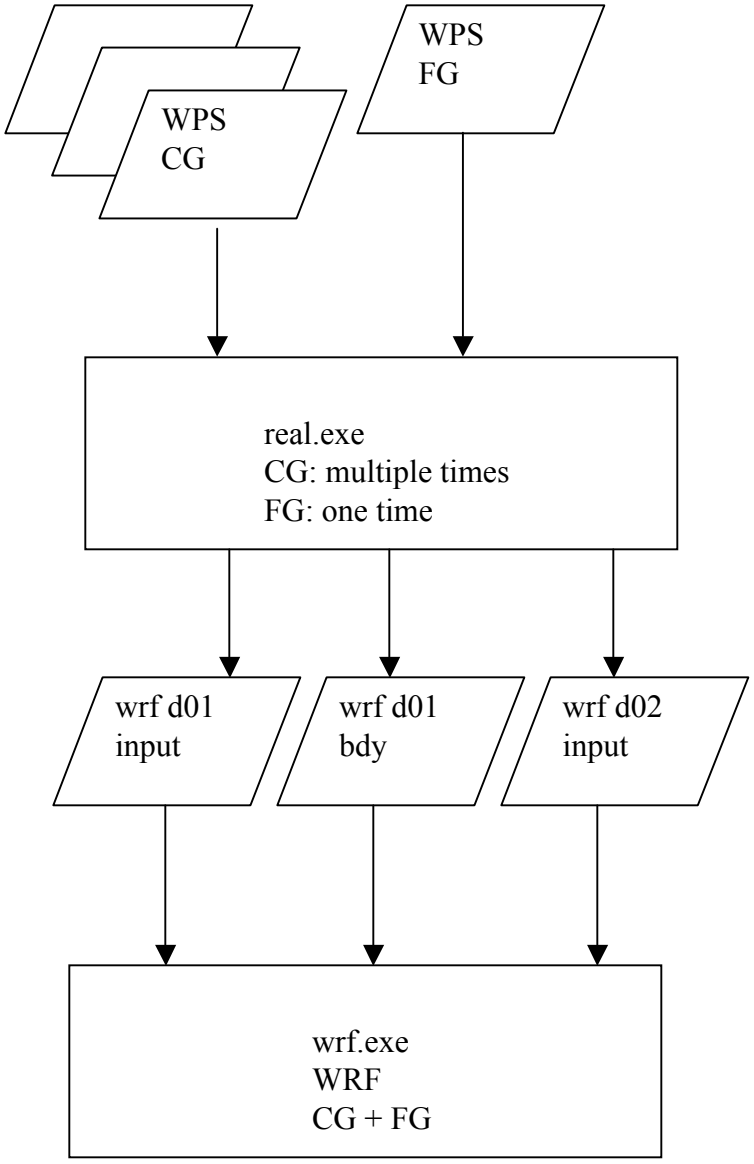
No vertical nesting

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

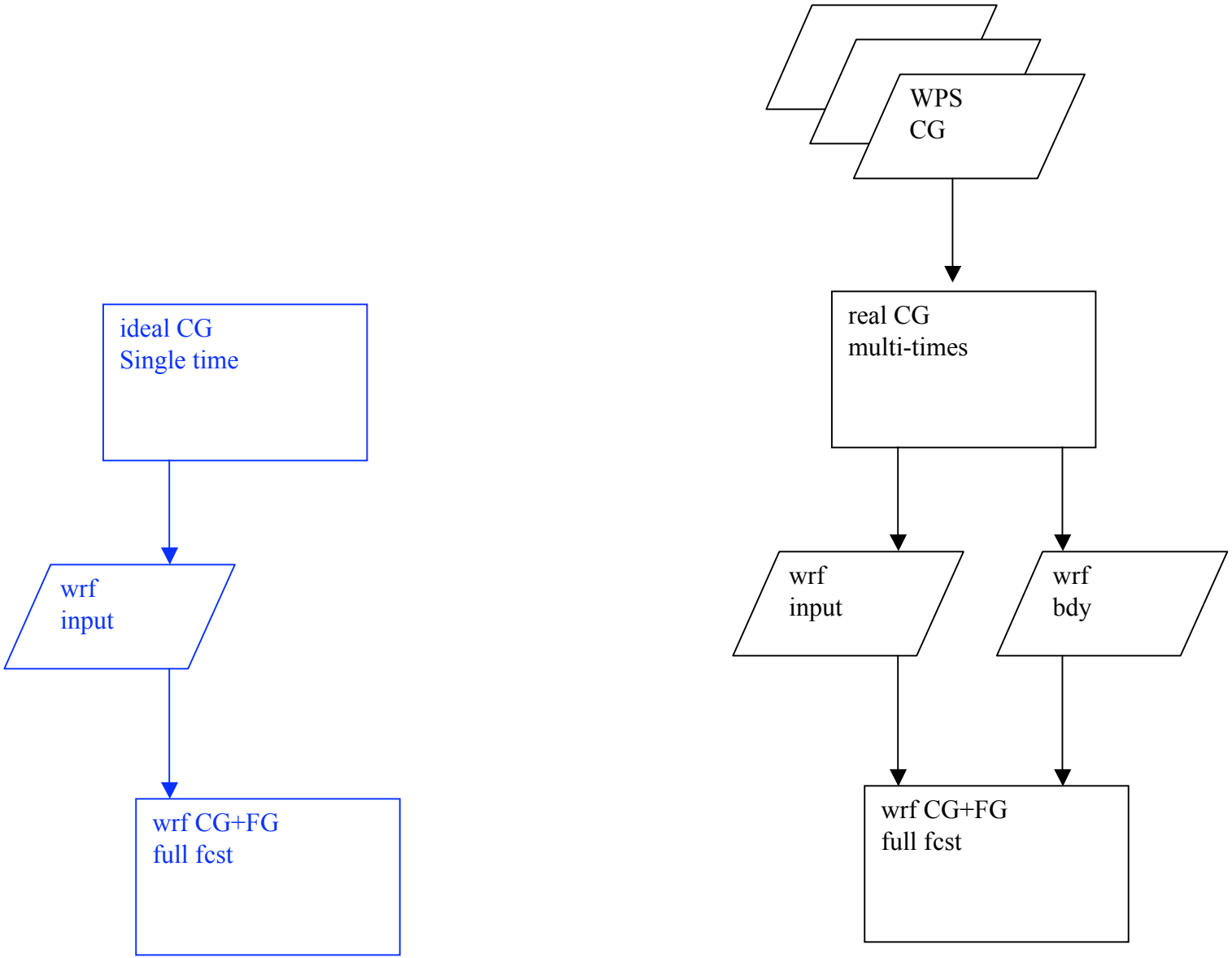
The grid distance ratio is not strictly tied to the time step ratio

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



2-Way Nest with 1 Input



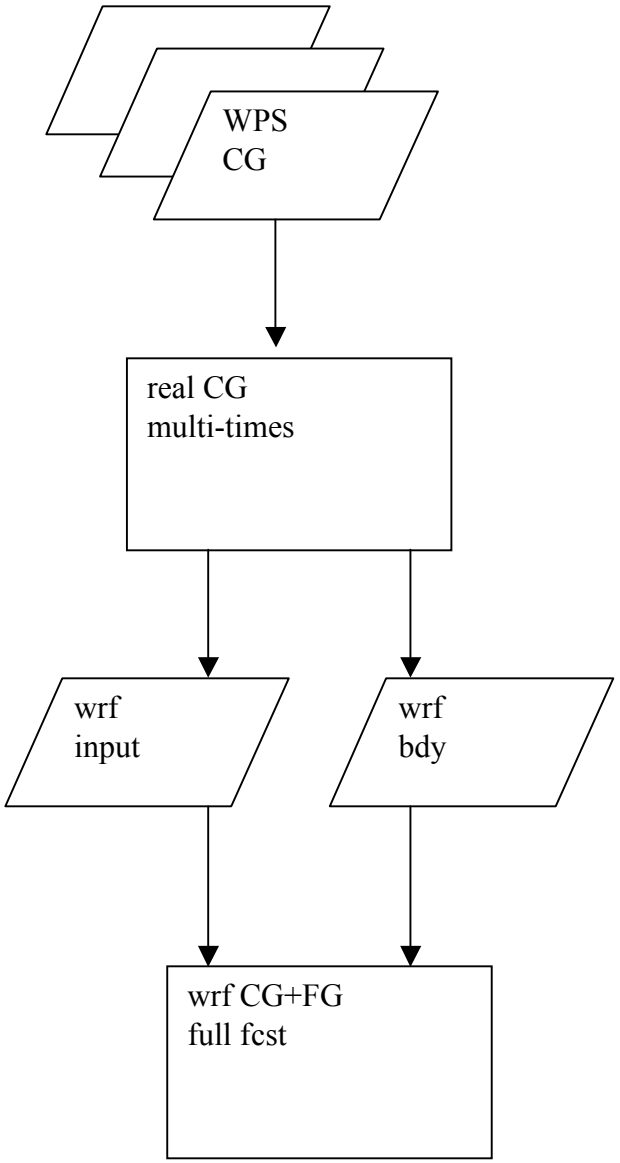
2-Way Nest with 1 Input

A single namelist column entry is tied to each domain

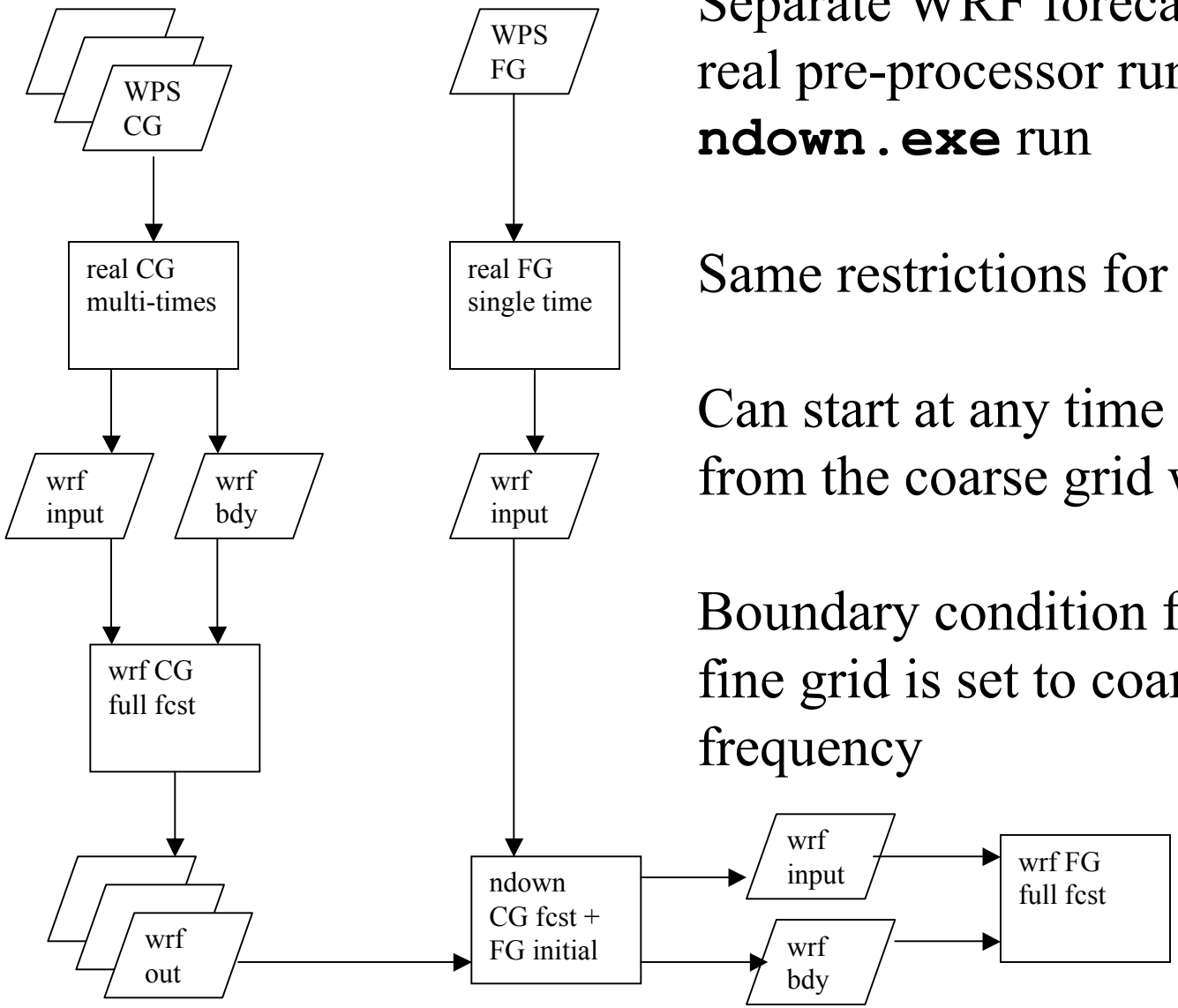
The horizontal interpolation method, feedback, and smoothing are largely controlled through the Registry file

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 **`./WRFV2/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 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

Some Nesting Hints

- Allowable domain specifications
- Defining a starting point
- Illegal domain specifications
- 1-way *vs* 2-way nesting
- Nest logic in WRF source code
- Nest information in the Registry

Allocate and Initialize a Nest

```
DO WHILE ( nests_to_open( grid , nestid , kid ) )
  a_nest_was_opened = .true.
  CALL med_pre_nest_initial ( grid , nestid , &
         config_flags )
  CALL alloc_and_configure_domain ( &
         domain_id  = nestid ,   &
         grid       = new_nest , &
         parent     = grid ,     &
         kid        = kid        )
   CALL Setup_Timekeeping (new_nest)
   CALL med_nest_initial ( grid , new_nest, &
         config_flags )
END DO
```

All Siblings get Processed

```
DO WHILE ( ASSOCIATED( grid_ptr ) )
  CALL set_current_grid_ptr( grid_ptr )
  CALL solve_interface ( grid_ptr )
  CALL domain_clockadvance ( grid_ptr )
  CALL domain_time_test( grid_ptr, &
  'domain_clockadvance' )
  grid_ptr => grid_ptr%sibling
END DO
```

Recursive Nest Depth

```
DO kid = 1, max_nests
  IF ( ASSOCIATED( grid_ptr%nests(kid)%ptr ) ) THEN
    CALL set_current_grid_ptr( grid_ptr%nests(kid)%ptr )
    CALL med_nest_force ( grid_ptr , &
         grid_ptr%nests(kid)%ptr , config_flags )
    grid_ptr%nests(kid)%ptr%start_subtime = &
         domain_get_current_time(grid) - &
         domain_get_time_step(grid)
    grid_ptr%nests(kid)%ptr%stop_subtime = &
         domain_get_current_time(grid)
    CALL integrate ( grid_ptr%nests(kid)%ptr )
    CALL med_nest_feedback ( grid_ptr , &
         grid_ptr%nests(kid)%ptr , config_flags )
  END IF
END DO
```

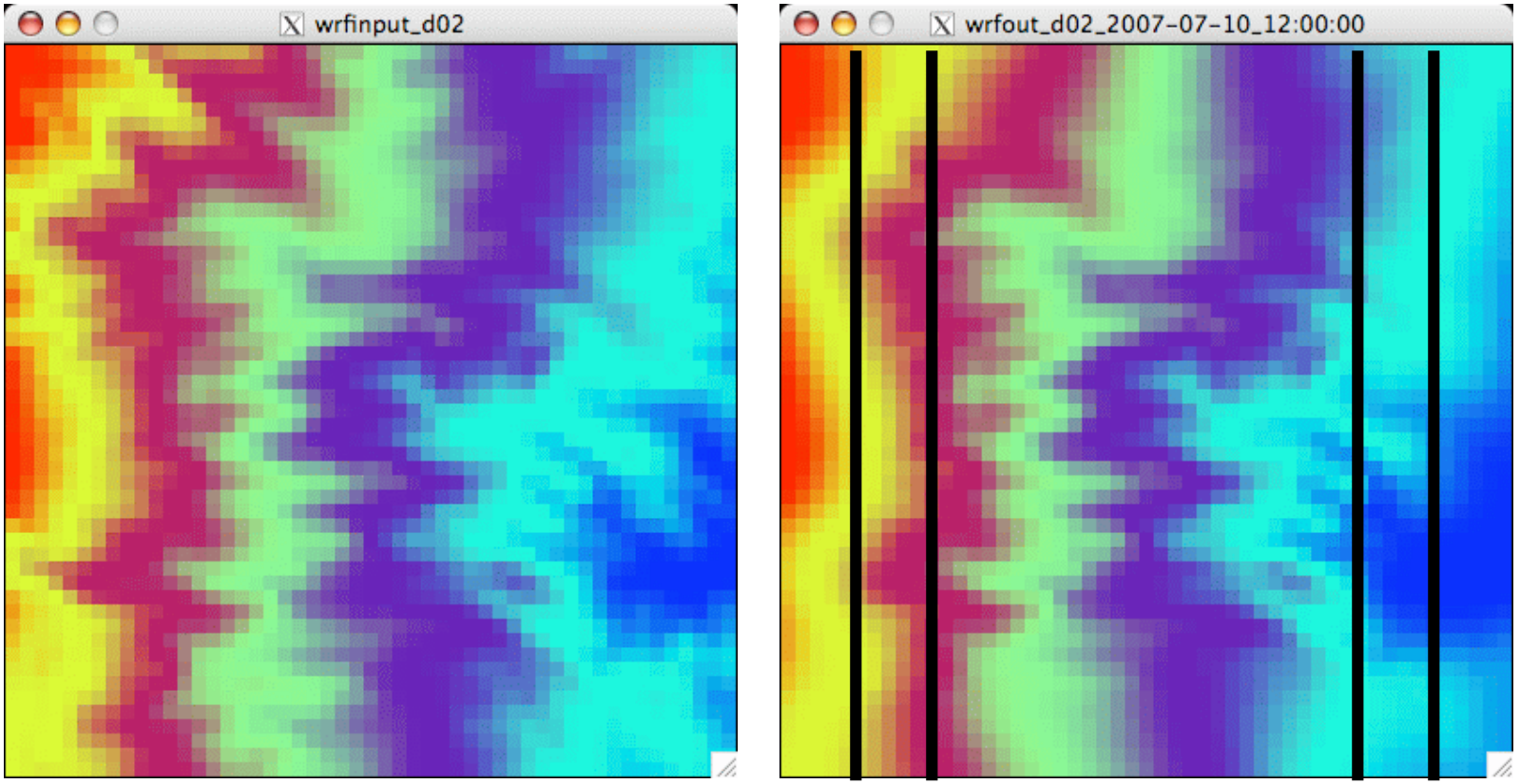
Input vs Interpolating

```
CALL med_interp_domain( parent, nest )

CALL init_domain_constants ( parent, nest )

IF ( nest_config_flags%input_from_file ) THEN

   IF ( nest_config_flags%input_from_file ) THEN
      CALL med_initialdata_input_ptr( nest , &
         nest_config_flags )
   ENDIF
```

Feedback and Base State

```
CALL med_nest_feedback ( parent , nest , &
   config_flags )

CALL start_domain ( nest , .TRUE. )

CALL start_domain ( parent , .TRUE. )
```

Lateral Smoothing



Intermediate Domains - Part 1

```
grid => nested_grid%intermediate_grid
CALL alloc_space_field ( grid, grid%id , 1 , 2 , .TRUE. ,
grid => parent_grid
CALL model_to_grid_config_rec ( grid%id , &
   model_config_rec , config_flags )
CALL couple_or_uncouple_em ( grid , config_flags ,.true. &
#          include "em_actual_new_args.inc" )
grid => nested_grid
CALL model_to_grid_config_rec ( grid%id , &
   model_config_rec , config_flags )
CALL couple_or_uncouple_em ( grid , config_flags ,.true. &
#          include "em_actual_new_args.inc" )
```

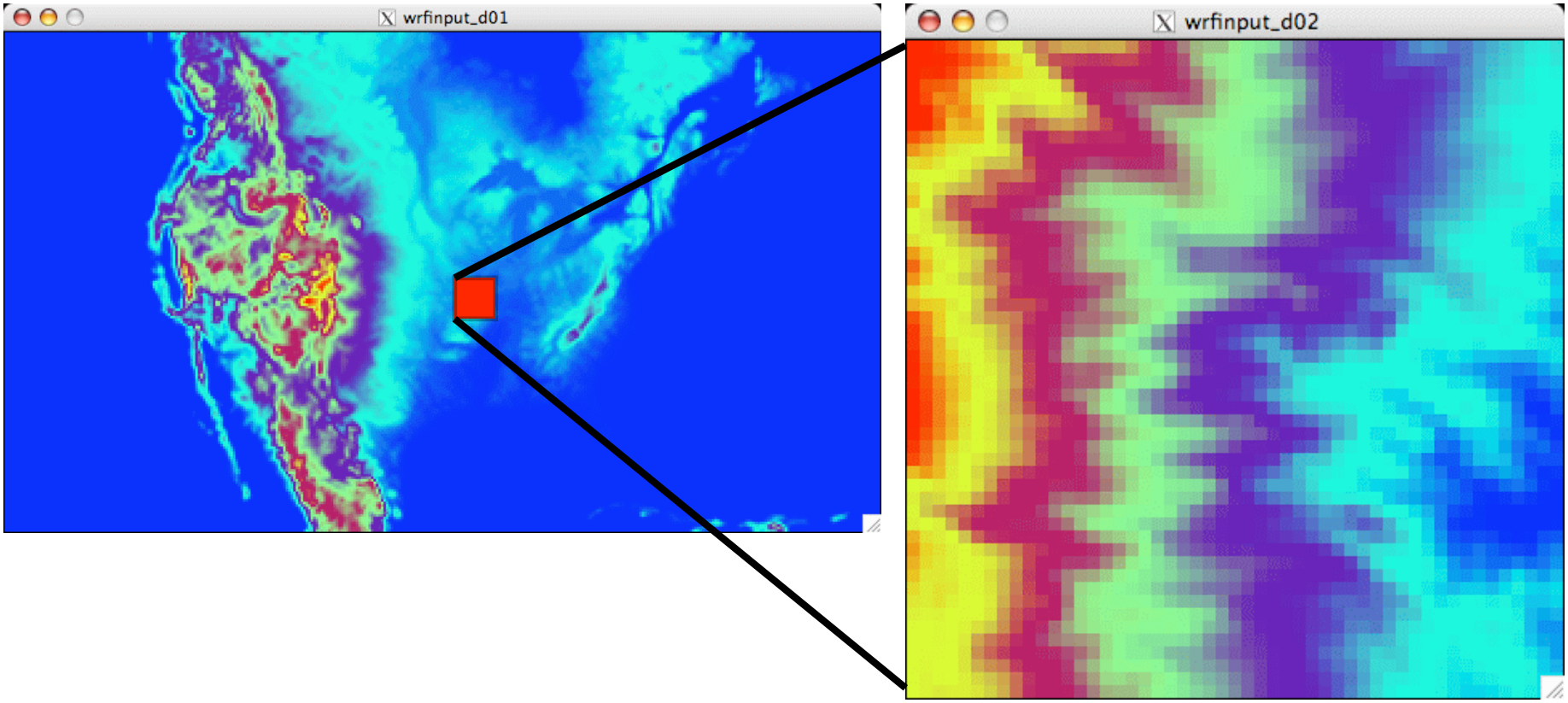
Intermediate Domains - Part 1

```
grid => parent_grid
CALL model_to_grid_config_rec ( grid%id ,
   model_config_rec , config_flags )
CALL interp_domain_em_part1 ( grid , &
   nested_grid%intermediate_grid, nested_grid, &
   config_flags   &
#         include "em_actual_new_args.inc" )
```

Intermediate Domains - Part 2

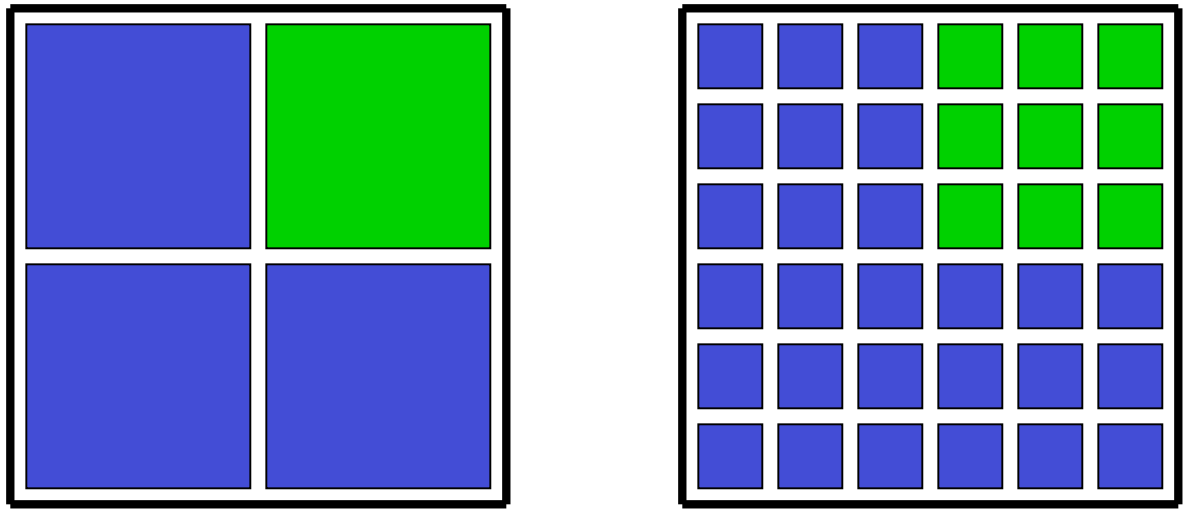
```
grid => nested_grid%intermediate_grid
CALL model_to_grid_config_rec ( nested_grid%id , &
   model_config_rec , config_flags )
CALL force_domain_em_part2 ( grid, nested_grid, &
   config_flags   &
#          include "em_actual_new_args.inc")

grid => nested_grid
CALL model_to_grid_config_rec ( grid%id , &
   model_config_rec , config_flags )
CALL couple_or_uncouple_em ( grid , config_flags ,.false. &
#          include "em_actual_new_args.inc" )

grid => parent_grid
CALL model_to_grid_config_rec ( grid%id , &
   model_config_rec , config_flags )
CALL couple_or_uncouple_em ( grid , config_flags ,.false. &
#          include "em_actual_new_args.inc" )
```

Intermediate Domains

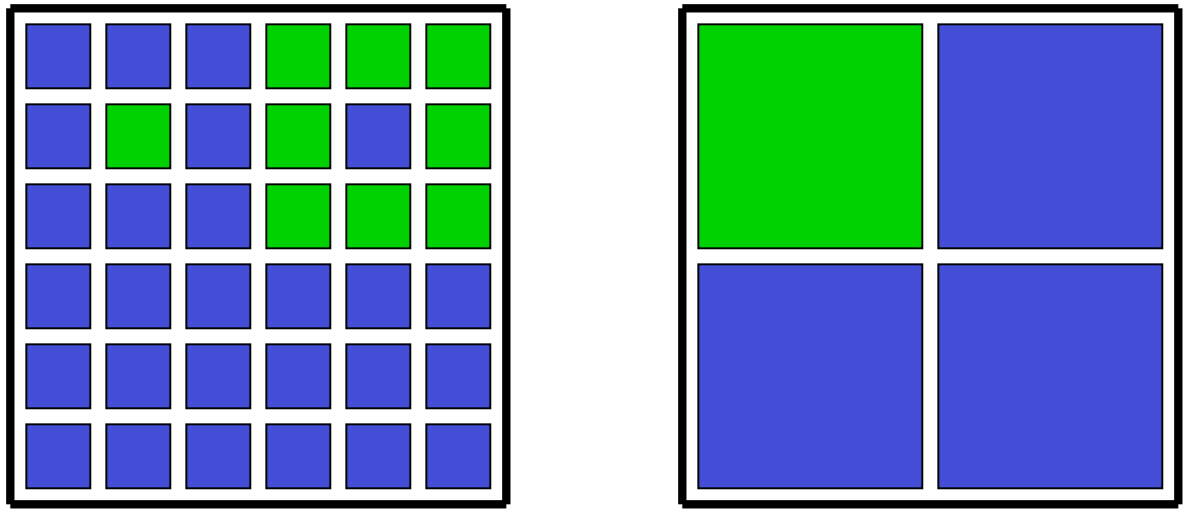


The intermediate domain between a parent and a child is the resolution of the coarse grid over the size of the fine grid. It allows the model to re-decompose the domain among all of the processors.

Masked Interpolation



Masked Feedback



Some Nesting Hints

- Allowable domain specifications
- Defining a starting point
- Illegal domain specifications
- 1-way *vs* 2-way nesting
- Nest logic in WRF source code
- Nest information in the Registry

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 (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" ""
```