

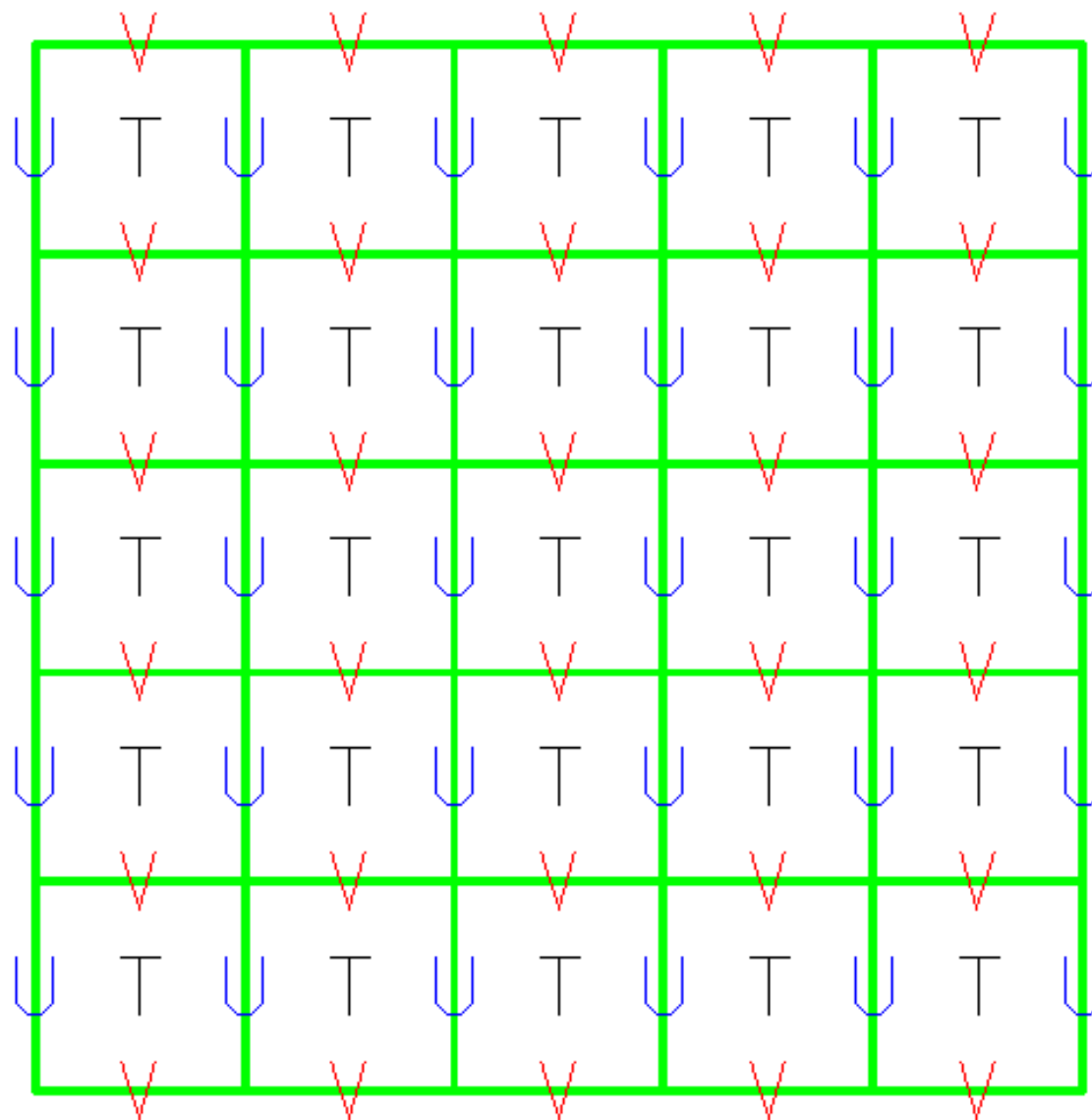
**How to set up and
run WRF for one-way
and
two-way nesting**

Dave Gill gill@ucar.edu

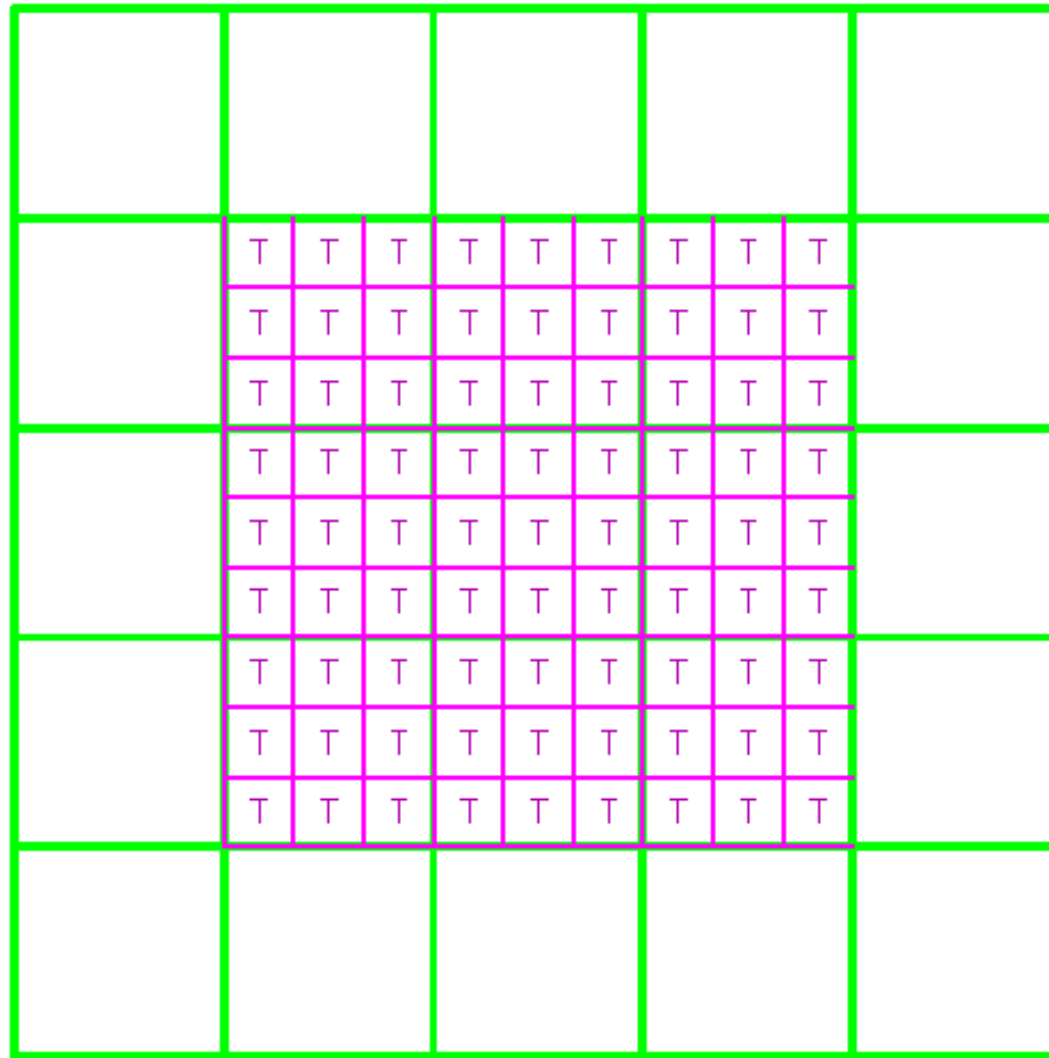
1-way vs. 2-way Nesting

- 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
- 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

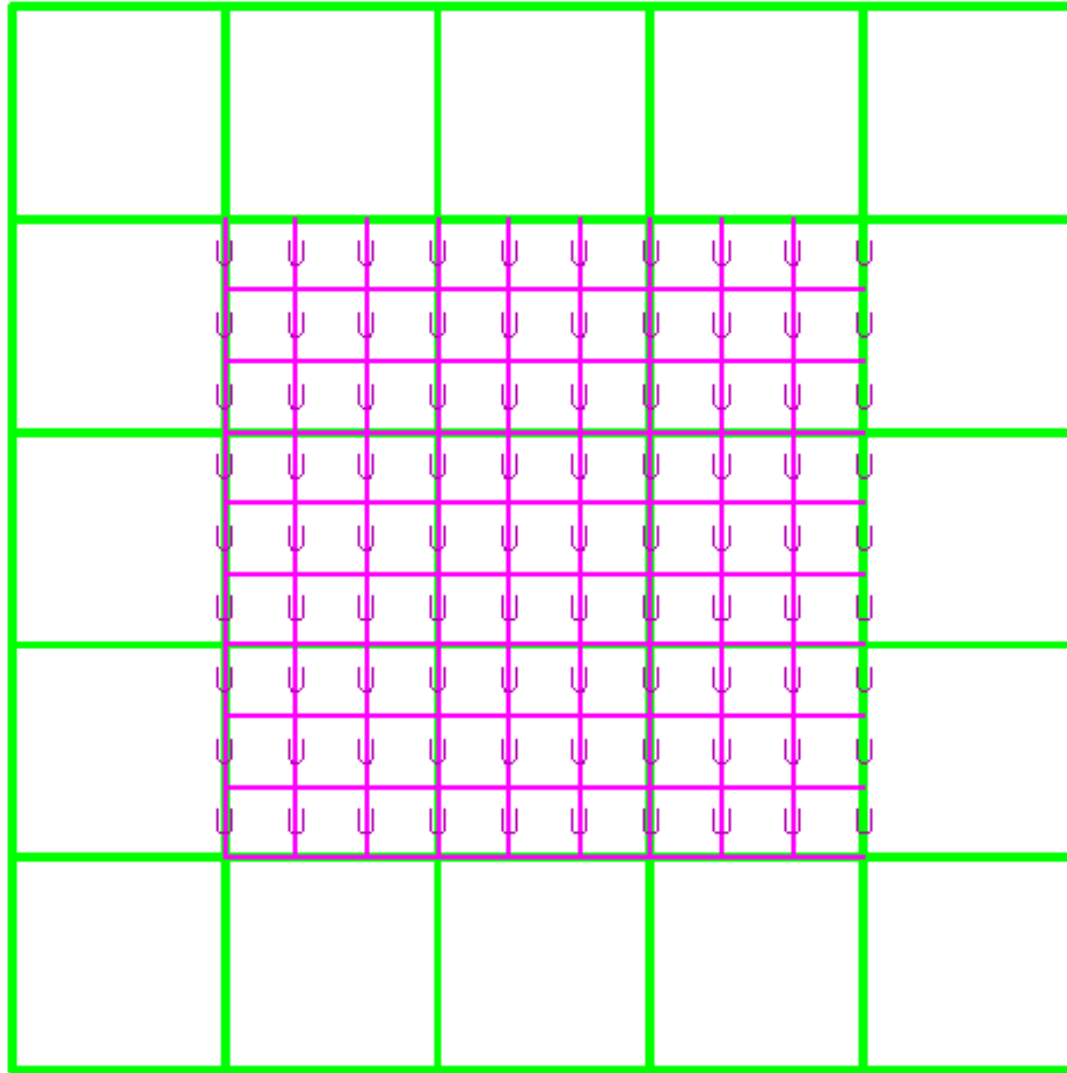
Arakawa-C Staggering



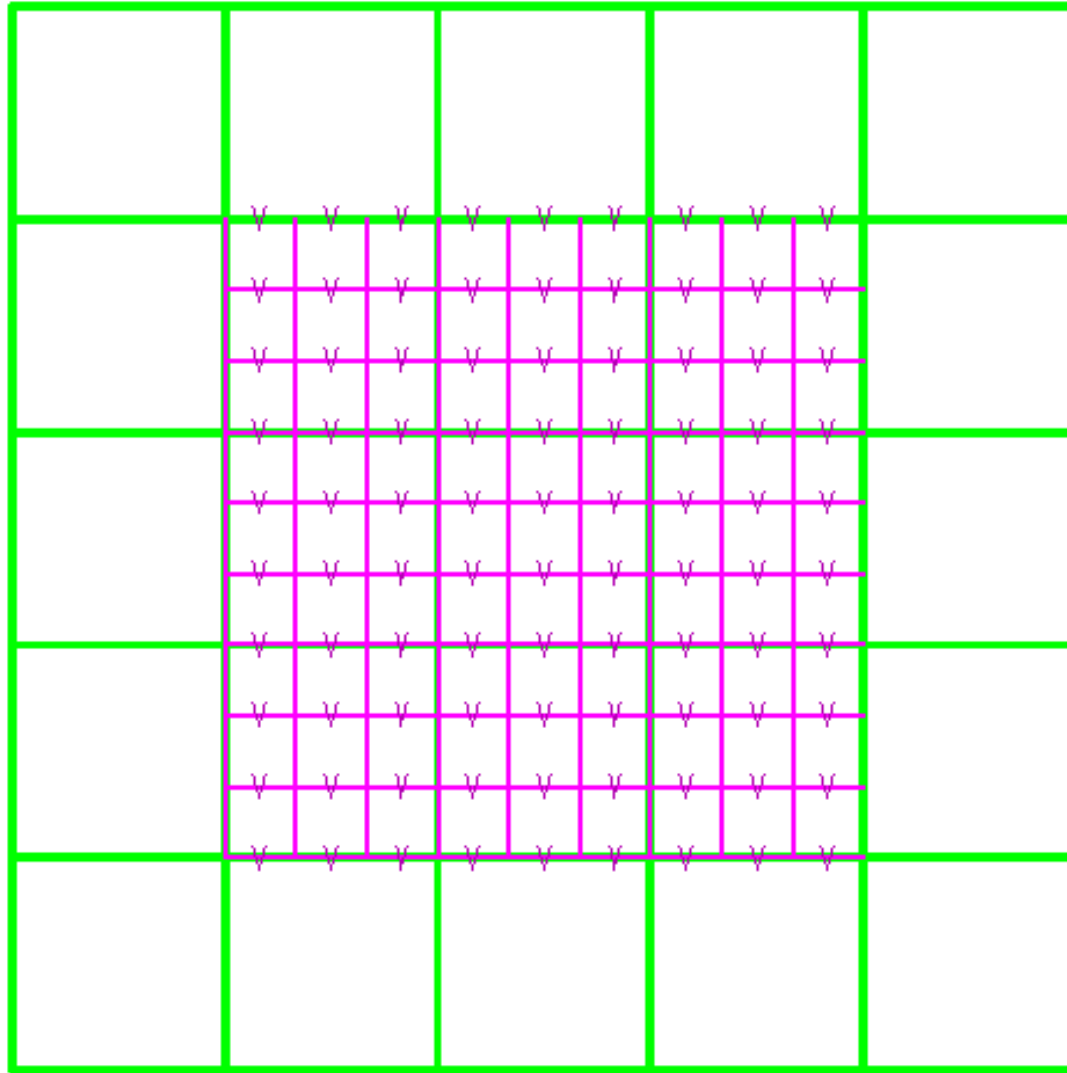
Arakawa-C Staggering – Mass Points 3::1 Ratio



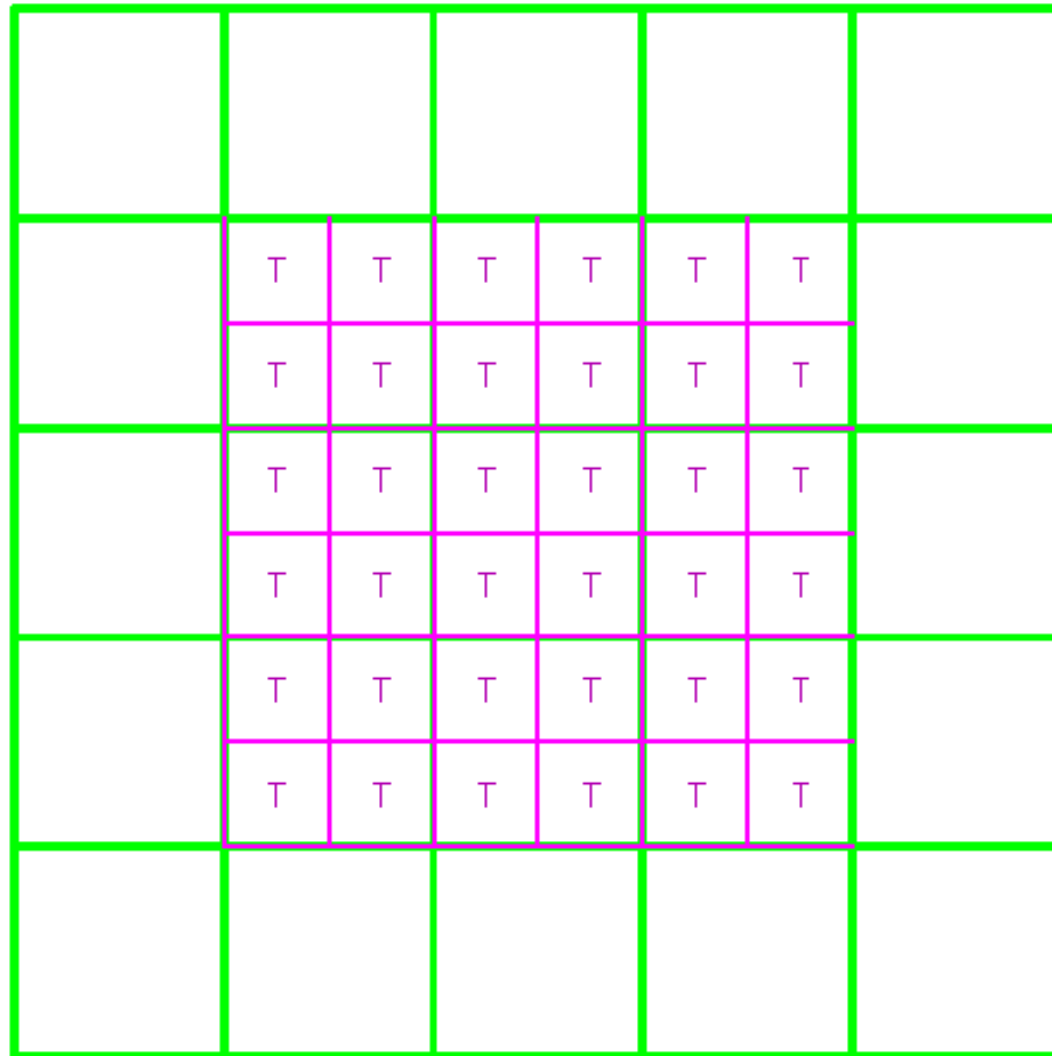
Arakawa-C Staggering – U Velocity Points 3::1 Ratio



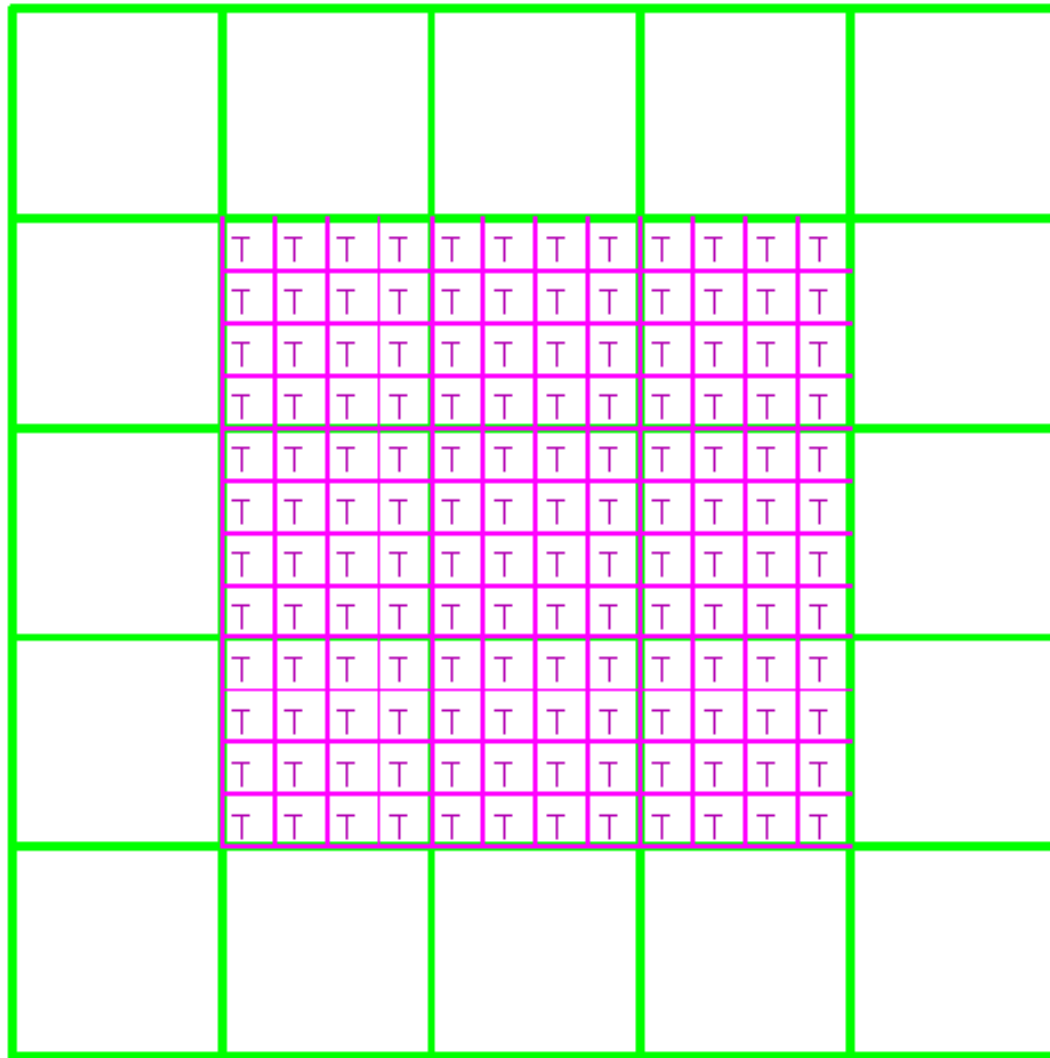
Arakawa-C Staggering – V Velocity Points 3::1 Ratio



Arakawa-C Staggering – Mass Points 2::1 Ratio

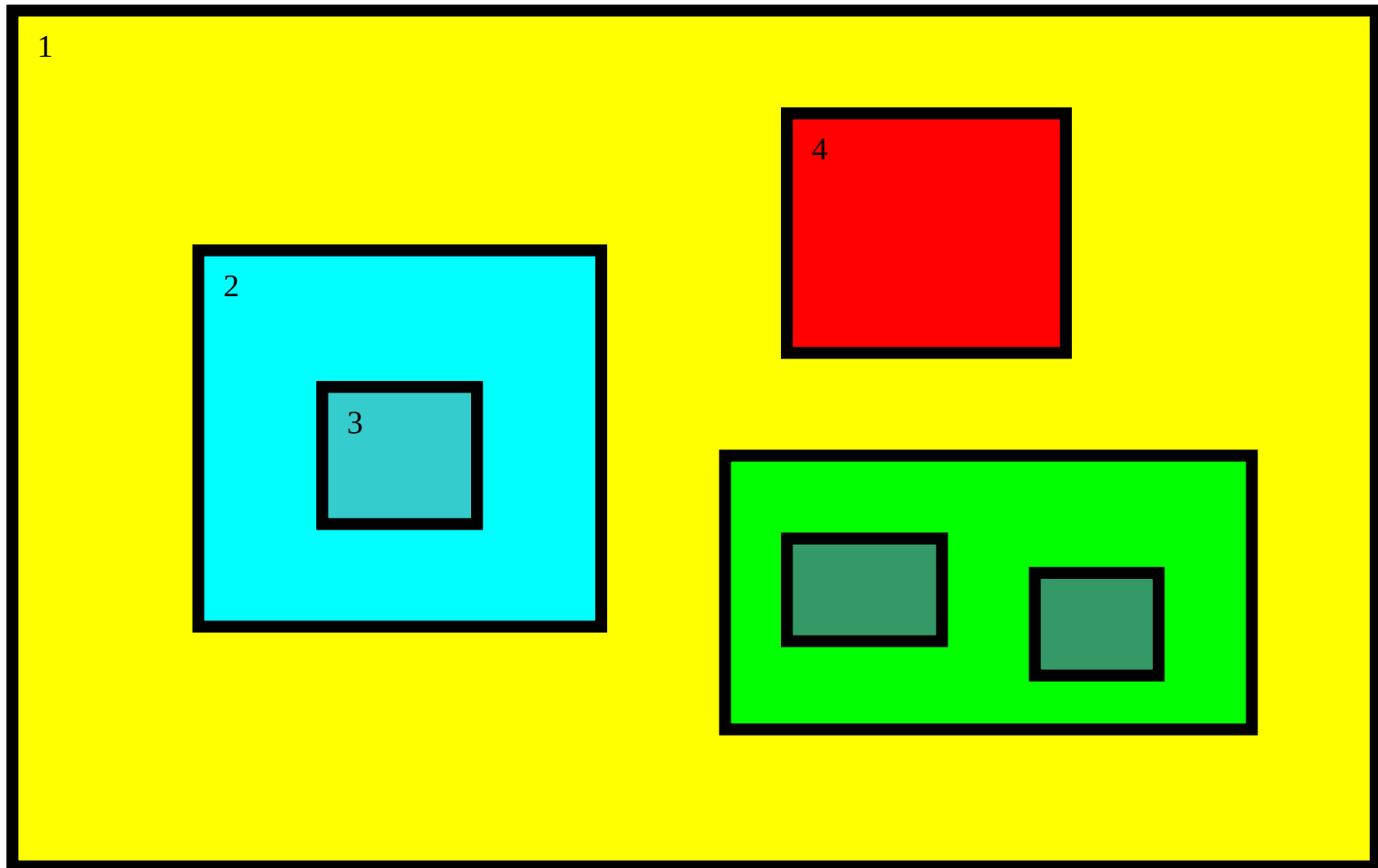


Arakawa-C Staggering – Mass Points 4::1 Ratio



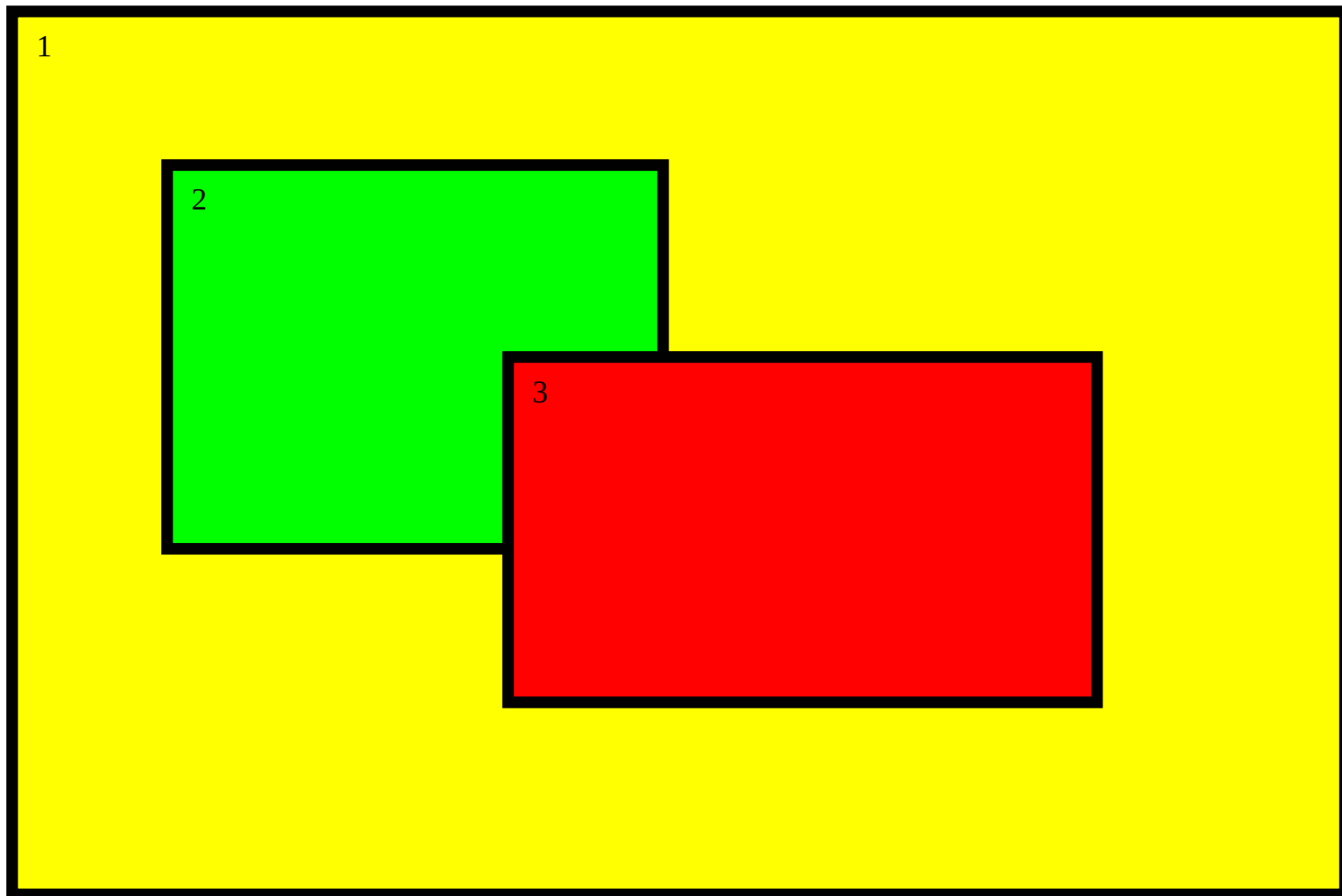
This is OK

Telescoped to any depth
Any number of siblings



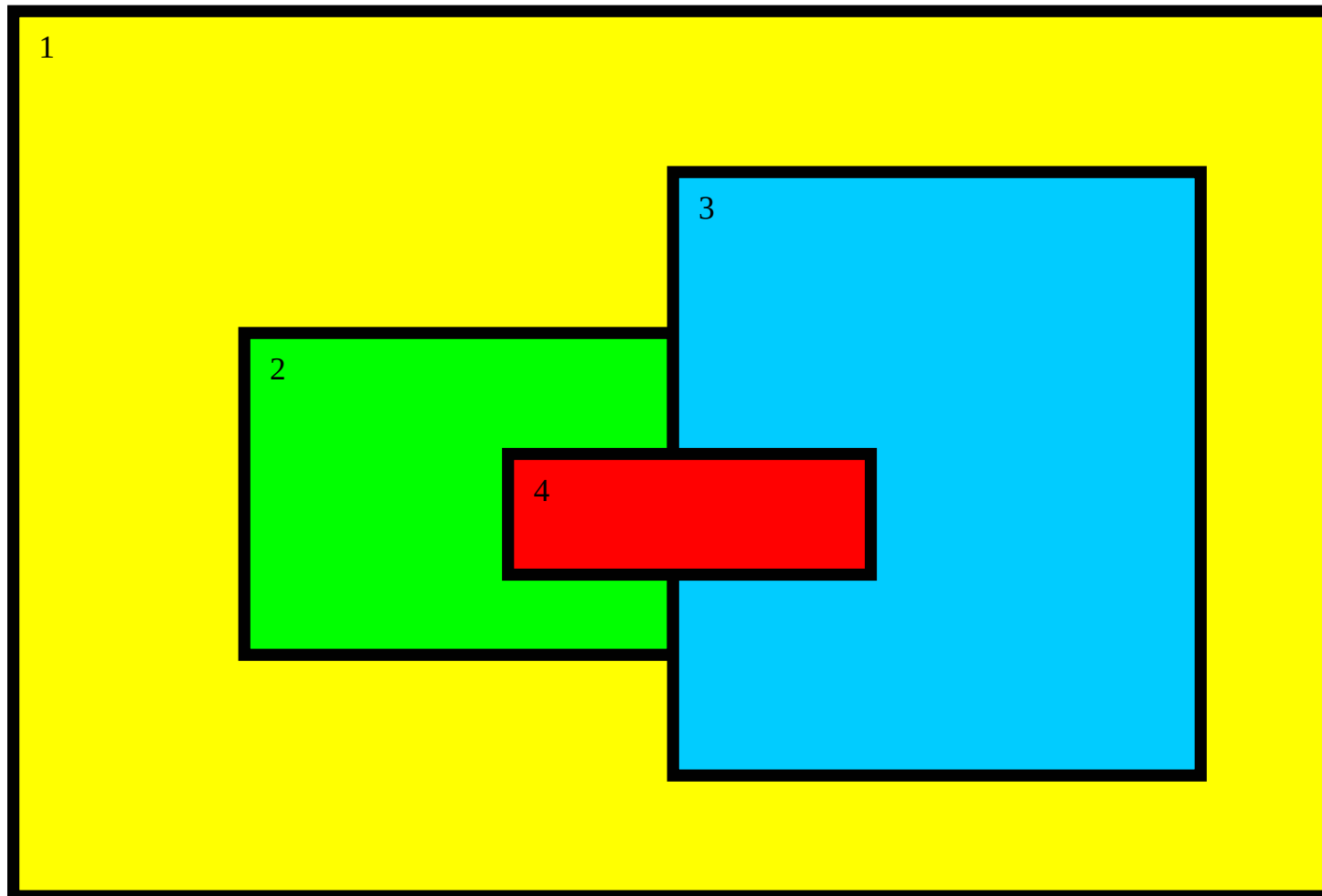
Not OK

Domains may not have overlapping points in the CG



Not OK either

Domains have 1 and only 1 parent



Run-Time Changes for Nesting

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		

Run-Time Changes for Nesting

max_dom	= 2,		
s_we	= 1,	1,	1,
e_we	= 74,	112 ,	94,
s_sn	= 1,	1,	1,
e_sn	= 61,	97 ,	91,
s_vert	= 1,	1,	1,
e_vert	= 28,	28,	28,
dx	= 30000,	10000 ,	3333,
dy	= 30000,	10000 ,	3333,

Run-Time Changes for Nesting

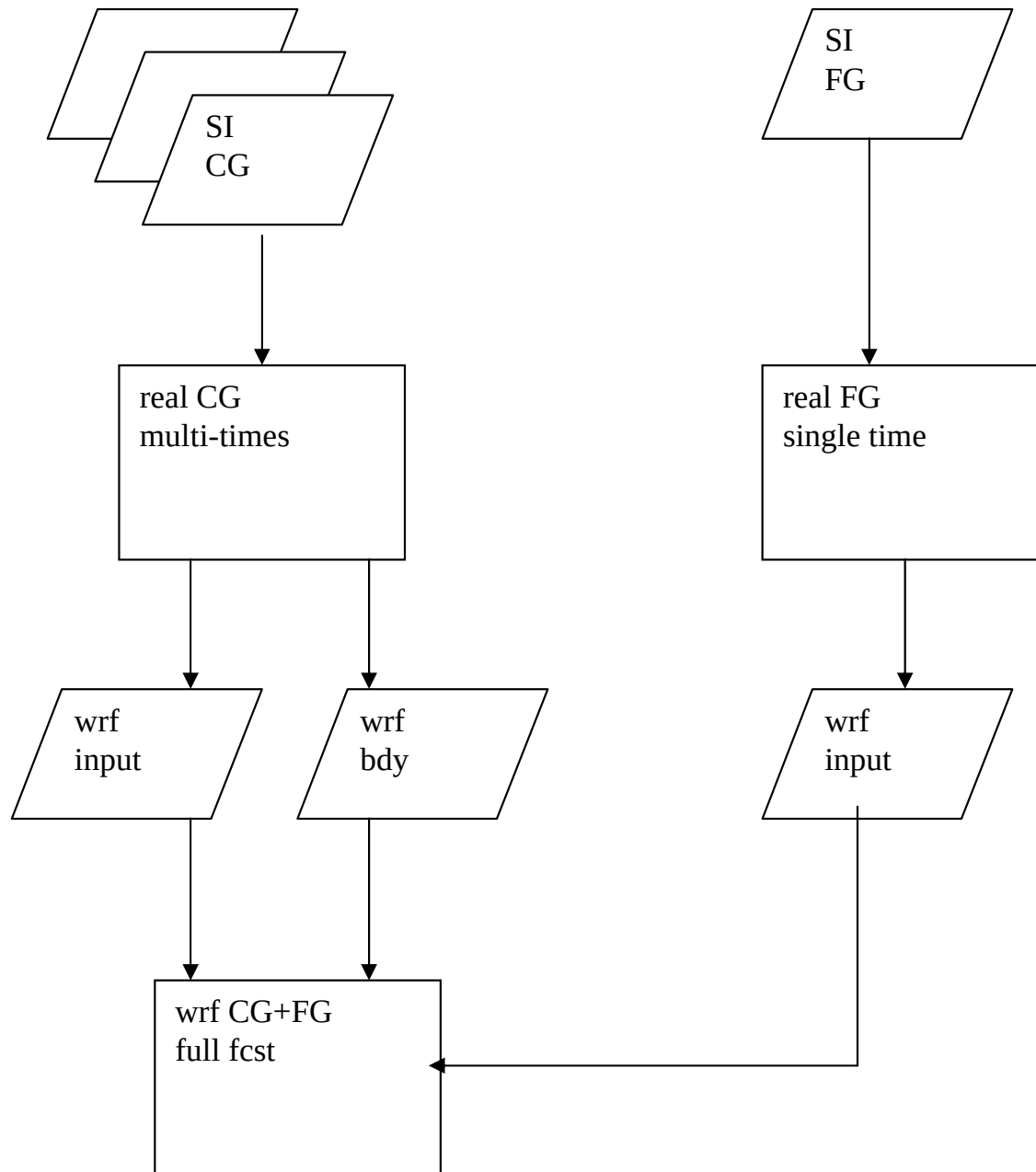
grid_id	= 1,	2,	3,
parent_id	= 0,	1,	2,
i_parent_start	= 0,	31,	30,
j_parent_start	= 0,	17,	30,
parent_grid_ratio	= 1,	3 ,	3,
parent_time_step_ratio	= 1,	3 ,	3,
feedback	= 1,		
smooth_option	= 0		

Run-Time Changes for Nesting

num_moves	= 4	
move_id	= 2 , 2 , 2 , 2	DOMAIN ID
move_interval	= 7, 22 , 37 , 52	TIME (min)
move_cd_x	= 1 , 1 , 1 , 1	Must be
move_cd_y	= 1 , 1 , 1 , 1	{ -1, 0, 1 }

Moving nests require `-DMOVE_NESTS` flag on the `ARCHFLAGS` in the `configure.wrf` file at build

2-Way Nest with 2 Inputs



2-Way Nest with 2 Inputs

real.exe

- There is file renaming required, so it is easier to start with the fine grid during the processing
- Probably should start the coarse and fine grids at the same time

2-Way Nest with 2 Inputs

real.exe

- Run single time period of fine grid SI output into real
- The namelist.input first column needs to have the fine grid dimensions
- The dx/dy values should be consistent with the fine grid
- The start and end date/times should reflect the single time for the fine grid

2-Way Nest with 2 Inputs

real.exe

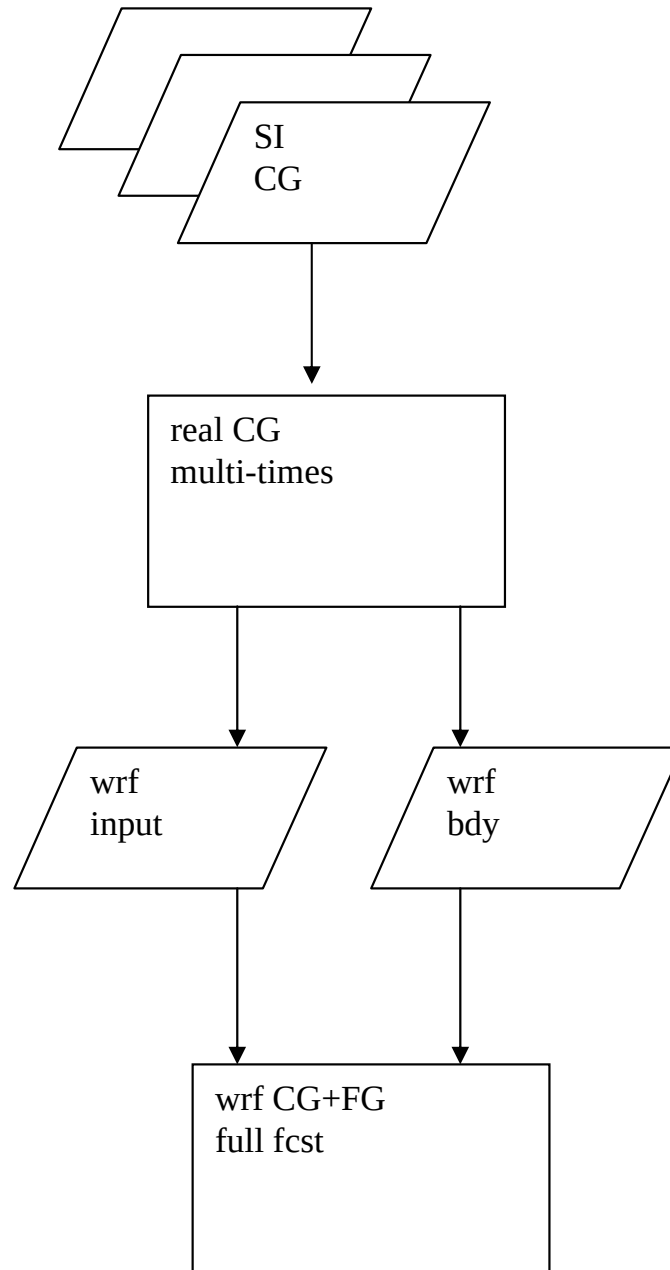
- Link the SI output file for the fine grid into the domain, it needs to be “d01” for real to recognize it as input, run real, rename the output to “d02”
- **ln -s **
**\$MOAD_DATAROOT/siprd/wrf*d02* **
wrf_real_d01_2000_09_24-12:00:00
- **./real.exe**
- **mv wrfinput_d01 wrfinput_d02**

2-Way Nest with 2 Inputs

real.exe

- Test [script](#) in `./WRFV2/test/em_real`
- Run test case

2-Way Nest with 1 Input



2-Way Nest with 1 Input

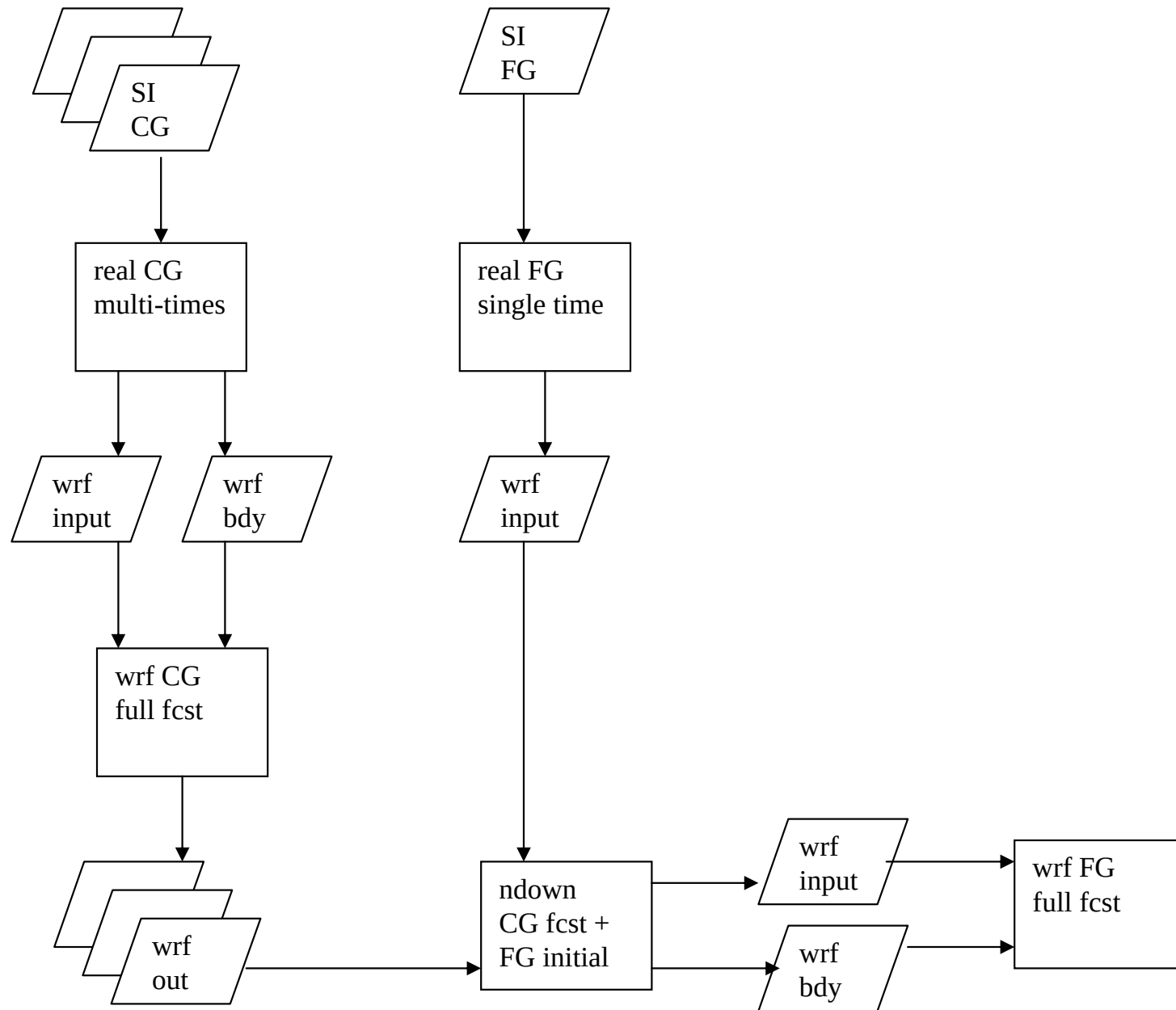
real.exe

- When running a 2-way nest with only a single input domain, the fine grid is entirely generated by the coarse grid
- No separate fine grid pre-processing is required
- The real.exe program is run identically to a single domain run (as if the coarse grid was running alone)

2-Way Nest with 1 Input real.exe

- Can do moving inner nest
- Test [script](#) in ./WRFV2/test/em_real
- (and moving [script](#), too)
- Run test cases

1-Way Nest with 2 Inputs



1-Way Nest with 2 Inputs

- Similar to 2-way nesting with 2 input files
- [Test script](#) in `./WRFV2/test/em_real`

1-Way Nest with 2 Inputs

- Run test case

```

#!/bin/csh

#      Run the WRF 2-way using 2 input files.

#      This shell assumes that you have already built the WRF code and that the
#      real.exe, ndown.exe, and wrf.exe files exist. It assumes that the input
#      data for the Jan00 case is in a "normal" location. This code is executed
#      from the ./WRFV2/test/em_real directory.

#      Steps in the shell:
#      1. make fine grid ic, only a single time period
#      2. make coarse+fine grid forecast - 24 h Jan 00, 30 km

#####

#      1.
#      Make the fine grid namelist for a 1-time run for real

echo 1.1
sed -e '/^ end_day/s/25/24/' \
    -e '/^ sf_surface_physics/s/1/2/' \
    -e '/^ num_soil_layers/s/5/4/' \
    -e '/^ e_we/s/74,/' \
    -e '/^ e_sn/s/61,/' \
    -e '/^ dx/s/30000/10000/' \
    -e '/^ dy/s/30000/10000/' \
    namelist.input.jan00 >! namelist.input

#      Get the fine grid input data from the SI

echo 1.2
rm -rf wrf_real*d01*
rm -rf wrfi* wrfb*
if ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    rm rsl*
    ln -sf /mmm/users/gill/WRF-data-EM/jan00/wrf_real*d02*24_12:00:00 wrf_real_input_em.d01.2000-01-
24_12:00:00
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    ln -sf /users/gill/WRF-data-EM/jan00/wrf_real*d02*24_12:00:00 wrf_real_input_em.d01.2000-01-
24_12:00:00
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then

```

```
ln -sf /users/gill/WRF-data-EM/jan00/wrf_real*d02*24_12:00:00 wrf_real_input_em.d01.2000-01-
24_12:00:00
endif
```

```
# Run fine grid real
```

```
echo 1.3
```

```
if ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    setenv MP_PROCS 1
    setenv MP_RMPPOOL 1
    real.exe
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    real.exe
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    mpirun -np 1 -machinefile ./machfile real.exe
endif
```

```
# Well, did the fine grid ic (remember, 1-time, no bc) work?
```

```
echo 1.4
```

```
set ic_size = `ls -ls wrfininput_d01 | awk ' { print $6 } '`
if ( $ic_size > 10000 ) then
    rm rsl*
    mv wrfininput_d01 wrfininput_d02
else
    echo fine grid ic 1 time wrong size
    exit ( 12 )
endif
```

```
#####
```

```
# 2.
```

```
# Make the coarse grid namelist: 1 h, output every hour, and use the Noah LSM
```

```
echo 2.1
```

```
sed -e '/^ run_hours/s/12/1/' \
    -e '/^ history_interval/s/180/60/' \
    -e '/^ sf_surface_physics/s/1/2/' \
    -e '/^ sf_surface_physics/s/1/2/' \
    -e '/^ num_soil_layers/s/5/4/' \
    -e '/^ max_dom/s/1/2/' \
    -e '/^ input_from_file/s/false/true/' \
```

```

namelist.input.jan00 >! namelist.input

#      Get the coarse grid input data from the SI

if      ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    ln -sf /mmm/users/gill/WRF-data-EM/jan00/wrf_real*d01* .
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    ln -sf /users/gill/WRF-data-EM/jan00/wrf_real*d01* .
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    ln -sf /users/gill/WRF-data-EM/jan00/wrf_real*d01* .
endif

#      Run coarse grid real

echo 2.2
if      ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    setenv MP_PROCS 1
    setenv MP_RMPPOOL 1
    real.exe
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    real.exe
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    cat >! machfile << EOF
`hostname`
`hostname`
`hostname`
`hostname`
EOF
    mpirun -np 1 -machinefile ./machfile real.exe
endif

#      Well, did the coarse grid ic/bc work?

echo 2.3
set ic_size = `ls -ls wrfinput_d01 | awk ' { print $6 } '`
set bc_times = `ncdump -h wrfbdy_d01 | grep "Time = UNLIMITED" | cut -d"(" -f2 | cut -dc -f1`
if ( ( $ic_size > 10000 ) && ( $bc_times == 4 ) ) then
    rm rsl*
else
    echo coarse grid ic bc wrong size
    exit ( 2 )
endif

```

```

#      Run coarse+fine grid wrf

echo 2.4
if      ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    setenv MP_PROCS 4
    wrf.exe
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    wrf.exe
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    mpirun -np 2 -machinefile ./machfile wrf.exe
endif

#      Well, did the coarse+fine grid wrf work?

echo 2.5
set c_times = `ncdump -h wrfout_d01_2000-01-24_12:00:00 | grep "Time = UNLIMITED" | cut -d"(" -f2 | cut -dc -
f1`
set f_times = `ncdump -h wrfout_d02_2000-01-24_12:00:00 | grep "Time = UNLIMITED" | cut -d"(" -f2 | cut -dc -
f1`
if ( ( $c_times == 2 ) && ( $f_times == 2 ) ) then
    echo 2w2i worked
    rm rsl*
else
    echo coarse grid wrf output wrong size
    exit ( 3 )
endif

```

```

#!/bin/csh

#      Run the WRF 2-way using the WRF program to generate the fine grid info
#      from the coarse grid.

#      This shell assumes that you have already built the WRF code and that the
#      real.exe, ndown.exe, and wrf.exe files exist. It assumes that the input
#      data for the Jan00 case is in a "normal" location. This code is executed
#      from the ./WRFV2/test/em_real directory.

#      Make the coarse+fine grid namelist: 24 h, output every hour, and use the Noah LSM

echo 1.1
sed -e '/^ run_hours/s/12/1/' \
    -e '/^ history_interval/s/180/60/' \
    -e '/^ sf_surface_physics/s/1/2/' \
    -e '/^ sf_surface_physics/s/1/2/' \
    -e '/^ num_soil_layers/s/5/4/' \
    -e '/^ max_dom/s/1/2/' \
    namelist.input.jan00 >! namelist.input

#      Get the coarse grid input data from the SI

if      ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    ln -sf /mmm/users/gill/WRF-data-EM/jan00/wrf_real*d01* .
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    ln -sf /users/gill/WRF-data-EM/jan00/wrf_real*d01* .
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    ln -sf /users/gill/WRF-data-EM/jan00/wrf_real*d01* .
endif

#      Run coarse grid real

echo 1.2
if      ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    setenv MP_PROCS 1
    setenv MP_RMPPOOL 1
    real.exe
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    real.exe
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    cat >! machfile << EOF
`hostname`
`hostname`

```

```

`hostname`
`hostname`
EOF
    mpirun -np 1 -machinefile ./machfile real.exe
endif

#      Well, did the coarse grid ic/bc work?

echo 1.3
set ic_size = `ls -ls wrfinp_d01 | awk ' { print $6 } '`
set bc_times = `ncdump -h wrfbdy_d01 | grep "Time = UNLIMITED" | cut -d "(" -f2 | cut -dc -f1`
if ( ( $ic_size > 10000 ) && ( $bc_times == 4 ) ) then
    rm rsl*
else
    echo coarse grid ic bc wrong size
    exit ( 2 )
endif

#      Run coarse+fine grid wrf

echo 1.4
if ( ( `uname` == AIX ) && ( `hostname | cut -c 1-2` == bs ) ) then
    setenv MP_PROCS 4
    wrf.exe
else if ( ( `uname` == OSF1 ) && ( `hostname | cut -c 1-6` == joshua ) ) then
    wrf.exe
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    mpirun -np 4 -machinefile ./machfile wrf.exe
endif

#      Well, did the coarse grid wrf work?

echo 1.5
set c_times = `ncdump -h wrfout_d01_2000-01-24_12:00:00 | grep "Time = UNLIMITED" | cut -d "(" -f2 | cut -dc -f1`
set f_times = `ncdump -h wrfout_d02_2000-01-24_12:00:00 | grep "Time = UNLIMITED" | cut -d "(" -f2 | cut -dc -f1`
if ( ( $c_times == 2 ) && ( $f_times == 2 ) ) then
    echo 2w1i worked
    rm rsl*
else
    echo coarse grid wrf output wrong size
    exit ( 3 )
endif

```



```

#!/bin/csh

#      Run the WRF 2-way using the WRF program to generate the fine grid info
#      from the coarse grid.

#      This shell assumes that you have already built the WRF code and that the
#      real.exe, and wrf.exe files exist. It assumes that the input
#      data for the Jan00 case is in a "normal" location. This code is executed
#      from the ./WRFV2/test/em_real directory.

#      Are we allowed to have a moving nest? Check the build from configure.wrf. There
#      should be a flag -DMOVE_NESTS on the ARCHFLAGS.

if ( -e ../../configure.wrf ) then
    grep DMOVE_NESTS ../../configure.wrf | grep -v "#" >& /dev/null
    set ok = $status
    if ( $ok == 0 ) then
        #echo things are ok to run moving nest
    else
        echo $ok
        echo "probably did not build for a moving nest ..."
        exit ( 1 )
    endif
endif

#      Make the coarse+fine grid namelist: 24 h, output every hour, and use the Noah LSM

echo 1.1

cat >! move_stuff << EOF
    num_moves                = 4
    move_id                   = 2 , 2 , 2 , 2
    move_interval              = 7, 22 , 37 , 52
    move_cd_x                  = 1 , 1 , 1 , 1
    move_cd_y                  = 1 , 1 , 1 , 1
EOF

sed -e '/^ run_hours/s/12/1/' \
    -e '/^ history_interval/s/180/60/' \
    -e '/^ sf_surface_physics/s/1/2/' \
    -e '/^ sf_surface_physics/s/1/2/' \

```

```

-e '/^ num_soil_layers/s/5/4/' \
-e '/^ max_dom/s/1/2/' \
-e '/^ smooth/r move_stuff' \
namelist.input.jan00 >! namelist.input
rm move_stuff

#      Get the coarse grid input data from the SI

if      ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    ln -sf /mmm/users/gill/WRF-data-EM/jan00/wrf_real*d01* .
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    ln -sf /users/gill/WRF-data-EM/jan00/wrf_real*d01* .
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    ln -sf /users/gill/WRF-data-EM/jan00/wrf_real*d01* .
endif

#      Run coarse grid real

echo 1.2
if      ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    setenv MP_PROCS 1
    setenv MP_RMPPOOL 1
    real.exe
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    cat >! machfile << EOF
`hostname`
`hostname`
`hostname`
`hostname`
EOF
    mpirun -np 1 -machinefile ./machfile real.exe
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    cat >! machfile << EOF
`hostname`
`hostname`
`hostname`
`hostname`
EOF
    mpirun -np 1 -machinefile ./machfile real.exe
endif

#      Well, did the coarse grid ic/bc work?

```

```

echo 1.3
set ic_size = `ls -ls wrfinput_d01 | awk ' { print $6 } '`
set bc_times = `ncdump -h wrfbdy_d01 | grep "Time = UNLIMITED" | cut -d "(" -f2 | cut -dc -f1`
if ( ( $ic_size > 10000 ) && ( $bc_times == 4 ) ) then
    rm rsl*
else
    echo coarse grid ic bc wrong size
    exit ( 2 )
endif

#      Run coarse+fine grid wrf

echo 1.4
if ( ( `uname` == AIX ) && ( `hostname | cut -c 1-2` == bs ) ) then
    setenv MP_PROCS 4
    wrf.exe
else if ( ( `uname` == OSF1 ) && ( `hostname | cut -c 1-6` == joshua ) ) then
    #wrf.exe
    mpirun -np 4 -machinefile ./machfile wrf.exe
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    mpirun -np 4 -machinefile ./machfile wrf.exe
endif

#      Well, did the coarse grid wrf work?

echo 1.5
set c_times = `ncdump -h wrfout_d01_2000-01-24_12:00:00 | grep "Time = UNLIMITED" | cut -d "(" -f2 | cut -dc -f1`
set f_times = `ncdump -h wrfout_d02_2000-01-24_12:00:00 | grep "Time = UNLIMITED" | cut -d "(" -f2 | cut -dc -f1`
if ( ( $c_times == 2 ) && ( $f_times == 2 ) ) then
    echo 2w1i_move worked
    rm rsl*
else
    echo coarse grid wrf output wrong size
    exit ( 3 )
endif

```

```

#!/bin/csh

#      Run the WRF 1-way using the ndown program to generate a nested forecast.
#      This shell assumes that you have already built the WRF code and that the
#      real.exe, ndown.exe, and wrf.exe files exist.  It assumes that the input
#      data for the Jan00 case is in a "normal" location.  This code is executed
#      from the ./WRFV2/test/em_real directory.

#      Steps in the shell:
#      1. make coarse grid forecast - 24 h Jan 00, 30 km
#      2. make fine grid ic, only a single time period
#      3. use ndown to make fine grid input, use hourly BC
#      4. make fine grid forecast, 24 h, 10 km nested domain
#      5. a last check of whether it all worked

#goto number_one
#goto number_two
#goto number_three
#goto number_four
#goto number_five

number_one:
#      1.
#      Make the coarse grid namelist: 24 h, output every hour, and use the Noah LSM

echo 1.1
sed -e '/^ run_hours/s/12/24/' \
    -e '/^ history_interval/s/180/60/' \
    -e '/^ sf_surface_physics/s/1/2/' \
    -e '/^ num_soil_layers/s/5/4/' \
    namelist.input.jan00 >| namelist.input

#      Get the coarse grid input data from the SI

if      ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    ln -sf /mmm/users/gill/WRF-data-EM/jan00/wrf_real*d01* .
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    ln -sf /users/gill/WRF-data-EM/jan00/wrf_real*d01* .
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    ln -sf /users/gill/WRF-data-EM/jan00/wrf_real*d01* .
endif

```

```

#           Run coarse grid real

echo 1.2
if          ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    setenv MP_PROCS 1
    setenv MP_RMPPOOL 1
    real.exe
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    real.exe
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    cat >! machfile << EOF
`hostname`
`hostname`
`hostname`
`hostname`
EOF
    mpirun -np 1 -machinefile ./machfile real.exe
endif

#           Well, did the coarse grid ic/bc work?

echo 1.3
set ic_size = `ls -ls wrfinput_d01 | awk ' { print $6 } '`
set bc_times = `ncdump -h wrfbdy_d01 | grep "Time = UNLIMITED" | cut -d"(" -f2 | cut -dc -f1`
if ( ( $ic_size > 10000 ) && ( $bc_times == 4 ) ) then
    rm rsl*
else
    echo coarse grid ic bc wrong size
    exit ( 2 )
endif

#           Run coarse grid wrf

echo 1.4
if          ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    setenv MP_PROCS 2
    wrf.exe
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    wrf.exe
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    mpirun -np 2 -machinefile ./machfile wrf.exe

```

```

endif

#      Well, did the coarse grid wrf work?

echo 1.5
set m_times = `ncdump -h wrfout_d01_2000-01-24_12:00:00 | grep "Time = UNLIMITED" | cut -d"(" -f2 | cut -dc -
f1`
if ( $m_times == 25 ) then
    rm rsl*
else
    echo coarse grid wrf output wrong size
    exit ( 3 )
endif

#####

number_two:
#      2.
#      Make the fine grid namelist for a 1-time run for real

echo 2.1
sed -e '/^ end_day/s/25/24/' \
    -e '/^ sf_surface_physics/s/1/2/' \
    -e '/^ num_soil_layers/s/5/4/' \
    -e '/^ e_we/s/74,/' \
    -e '/^ e_sn/s/61,/' \
    -e '/^ dx/s/30000/10000/' \
    -e '/^ dy/s/30000/10000/' \
    namelist.input.jan00 >! namelist.input

#      Get the fine grid input data from the SI

echo 2.2
rm -rf wrf_real*d01*
rm -rf wrfi* wrfb*
if ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    rm rsl*
    ln -sf /mmm/users/gill/WRF-data-EM/jan00/wrf_real*d02*24_12:00:00 wrf_real_input_em.d01.2000-01-
24_12:00:00
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    ln -sf /users/gill/WRF-data-EM/jan00/wrf_real*d02*24_12:00:00 wrf_real_input_em.d01.2000-01-
24_12:00:00

```

```

else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    ln -sf /users/gill/WRF-data-EM/jan00/wrf_real*d02*24_12:00:00 wrf_real_input_em.d01.2000-01-
24_12:00:00
endif

#      Run fine grid real

echo 2.3
if      ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    setenv MP_PROCS 1
    setenv MP_RMPPOOL 1
    real.exe
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    real.exe
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    mpirun -np 1 -machinefile ./machfile real.exe
endif

#      Well, did the fine grid ic (remember, 1-time, no bc) work?

echo 2.4
set ic_size = `ls -ls wrfinput_d01 | awk ' { print $6 } '`
if ( $ic_size > 10000 ) then
    rm rsl*
    rm wrf_real_input_em.d01.2000-01-24_12:00:00
    mv wrfinput_d01 wrf_real_input_em.d02.2000-01-24_12:00:00 # tricky, I know
else
    echo fine grid ic 1 time wrong size
    exit ( 12 )
endif

#####

number_three:
#      3.
#      Make the ndown namelist: 24 h, output every hour, and use the Noah LSM

echo 3.1
sed -e '/^ interval_seconds/s/21600/3600/' \
    -e '/^ sf_surface_physics/s/1/2/' \
    -e '/^ sf_surface_physics/s/1/2/' \

```

```

        -e '/^ num_soil_layers/s/5/4/' \
        namelist.input.jan00 >| namelist.input

#           Run ndown

echo 3.2
if          ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    setenv MP_PROCS 1
    setenv MP_RMPPOOL 1
    ndown.exe
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    ndown.exe
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    mpirun -np 1 -machinefile ./machfile ndown.exe
endif

#           Well, did ndown work?

echo 3.3
set ic_size = `ls -ls wrfinput_d02 | awk ' { print $6 } '`
set bc_times = `ncdump -h wrfbdy_d02 | grep "Time = UNLIMITED" | cut -d"(" -f2 | cut -dc -f1`
if ( ( $ic_size > 10000 ) && ( $bc_times == 24 ) ) then
    rm rsl*
    mv wrfinput_d02 wrfinput_d01
    mv wrfbdy_d02 wrfbdy_d01
    if ( ! -d hold ) mkdir hold
    mv wrfo* wrf_real_input_em.d02.2000-01-24_12:00:00 hold
else
    echo ndown output wrong size
    exit ( 22 )
endif

#####

number_four:
#           4.
#           Make the fine grid namelist for the full forecast run

echo 4.1
sed -e '/^ run_hours/s/12/24/' \
    -e '/^ interval_seconds/s/21600/3600/' \
    -e '/^ history_interval/s/180/60/' \

```



```

-e '/^ time_step/s/180/60/' \
-e '/^ e_we/s/74,/' \
-e '/^ e_sn/s/61,/' \
-e '/^ dx/s/30000,/' \
-e '/^ dy/s/30000,/' \
-e '/^ sf_surface_physics/s/1/2/' \
-e '/^ num_soil_layers/s/5/4/' \
namelist.input.jan00 >! namelist.input

#      Run fine grid wrf

echo 4.2
if      ( ( `uname` == AIX ) && ( `hostname` | cut -c 1-2` == bs ) ) then
    setenv MP_PROCS 2
    #wrf.exe
    echo submitting wrf job in batch queue
    m4 -DPWD=`pwd` run.csh.template >! run.csh
    llsubmit run.csh
    echo need to manually see if it worked
    exit ( 0 )
else if ( ( `uname` == OSF1 ) && ( `hostname` | cut -c 1-6` == joshua ) ) then
    wrf.exe
else if ( ( `uname` == Linux ) && ( `hostname` == bay-mmm ) ) then
    mpirun -np 2 -machinefile ./machfile wrf.exe
endif

#####

number_five:
#      5.
#      Well, did the fine grid wrf work?

echo 5.1
set m_times = `ncdump -h wrfout_d01_2000-01-24_12:00:00 | grep "Time = UNLIMITED" | cut -d"(" -f2 | cut -dc -
f1`
if ( $m_times == 25 ) then
    rm rsl*
else
    echo fine grid wrf output wrong size
    exit ( 43 )
endif

```