



WRF: *More Runtime Options*

Wei Wang

NCAR/NESL/MMM

Jan 2011



Fortran namelist

- Fortran 90 namelist is used to configure a run without recompiling the code
- Fortran 90 namelist has very specific format, so edit with care:

`&namelist-record` - start

/ - end

- As a general rule:

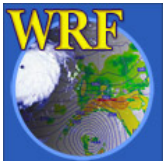
Multiple columns: domain dependent

Single column: value valid for all domains



More options

- Have covered basic, physics / diffusion options, and nudging options..
- More are introduced here:
 - IO options
 - Vertical interpolation options
 - options for long simulations
 - Adaptive-time step
 - Digital filter
 - Global runs
 - Moving nest
 - TC options
 - tracer
 - IO quilting
- Time series output



namelist.input

general namelist

records:

`&time_control`
`&domains`
`&physics`
`&dynamics`
`&bdy_control`
`&namelist_quilt`

specialized namelist
records:

`&dfi_control`
`&fdda`
`&grib2`
`&scm`
`&tc`



IO Control (1)

History output control in `&time_control`

`history_interval`: used often, unit in minutes

`history_interval_h`: history output interval in hours

`history_interval_s`: history output interval in seconds

`history_begin_h`: history output beginning time in hours

`history_begin_d`: history output beginning time in days

Complete listing in `Registry/`
`registry.io_boilerplate`



IO Control (2)

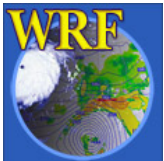
Optional history output in `&time_control`

1. Change Registry.EM and **recompile**:

```
state integer rainc ij misc 1 - h03 "RAINC" ""  
  "ACCUMULATED TOTAL CUMULUS PRECIPITATION"  
state integer rainnc ij misc 1 - h03 "RAINC" ""  
  "ACCUMULATED TOTAL GRID SCALE PRECIPITATION"
```

2. Edit namelist.input to output these variables:

```
auxhist3_outname = "rainfall_d<domain>"  
auxhist3_interval = 10  
frames_per_auxhist3 = 1000  
io_form_auxhist3 = 2
```



IO Control (3)

Starting in V3.2, there is an alternative to add additional output at **runtime** (state var in Registry only)

1. new namelists in &time_control:

```
iofields_filename(max_dom) = 'my_output.txt',  
ignore_iofields_warning
```

2. prepare text file ('my_output.txt') to define new io fields:

```
+:h:3:rainc,rainnc
```

3. set other namelists under &time_control:

```
auxhist3_outname = "rainfall_d<domain>"  
auxhist3_interval = 10  
frames_per_auxhist3 = 1000  
io_form_auxhist3 = 2
```



Vertical interpolation options (1)

Program **real** for ARW only, optional, &domains:

use_surface: whether to use surface observations

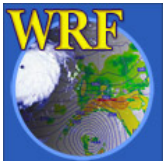
use_levels_below_ground: whether to use data below the ground

lowest_lev_from_sfc: logical, whether surface data is used to fill the lowest model level values

force_sfc_in_vinterp: number of levels to use surface data, default is 1

extrap_type: how to do extrapolation: 1 - use 2 lowest levels; 2 - constant

t_extrap_type : extrapolation option for temperature: 1 - isothermal; 2 - 6.5 K/km; 3 - adiabatic



Vertical interpolation options (2)

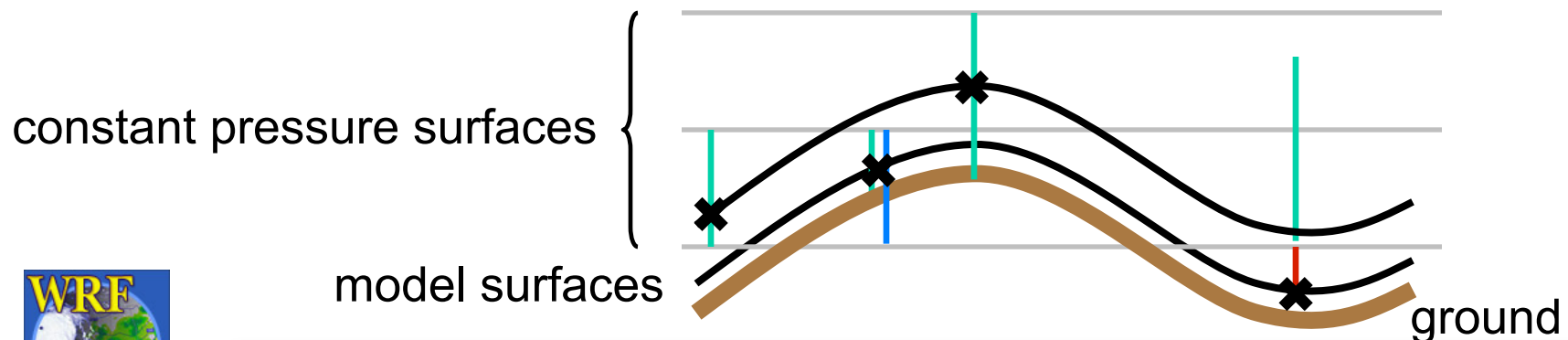
Program **real** for ARW only, optional:

interp_type: in pressure or log pressure

lagrange_order: linear or quadratic

zap_close_levels: delta p where a non-surface pressure level is removed in vertical interpolation

related namelists: **examples.namelist**



Options for long simulations (1)

Lower boundary update control: allow SST, seaice, monthly vegetation fraction and albedo to be updated regularly during a model run:

sst_update: 0 – no SST update
1 – update SST

Set before running **real**, and this will create additional output files: **wrflowinp_d01**, **wrflowinp_d02**, ..

Other namelists required in **&time_control**:

auxinput4_inname = "wrflowinp_d<domain>"

auxinput4_interval = 360

io_form_auxinput4 = 2 (netCDF)



Options for long simulations (2)

<code>sst_skin</code>	diurnal water temp update
<code>tmn_update</code>	deep soil temp update, used with lagday
<code>lagday</code>	averaging time
<code>bucket_mm</code>	bucket reset value for rainfall
<code>bucket_j</code>	bucket reset value for radiation fluxes
<code>spec_exp</code>	exponential multiplier for boundary zone ramping

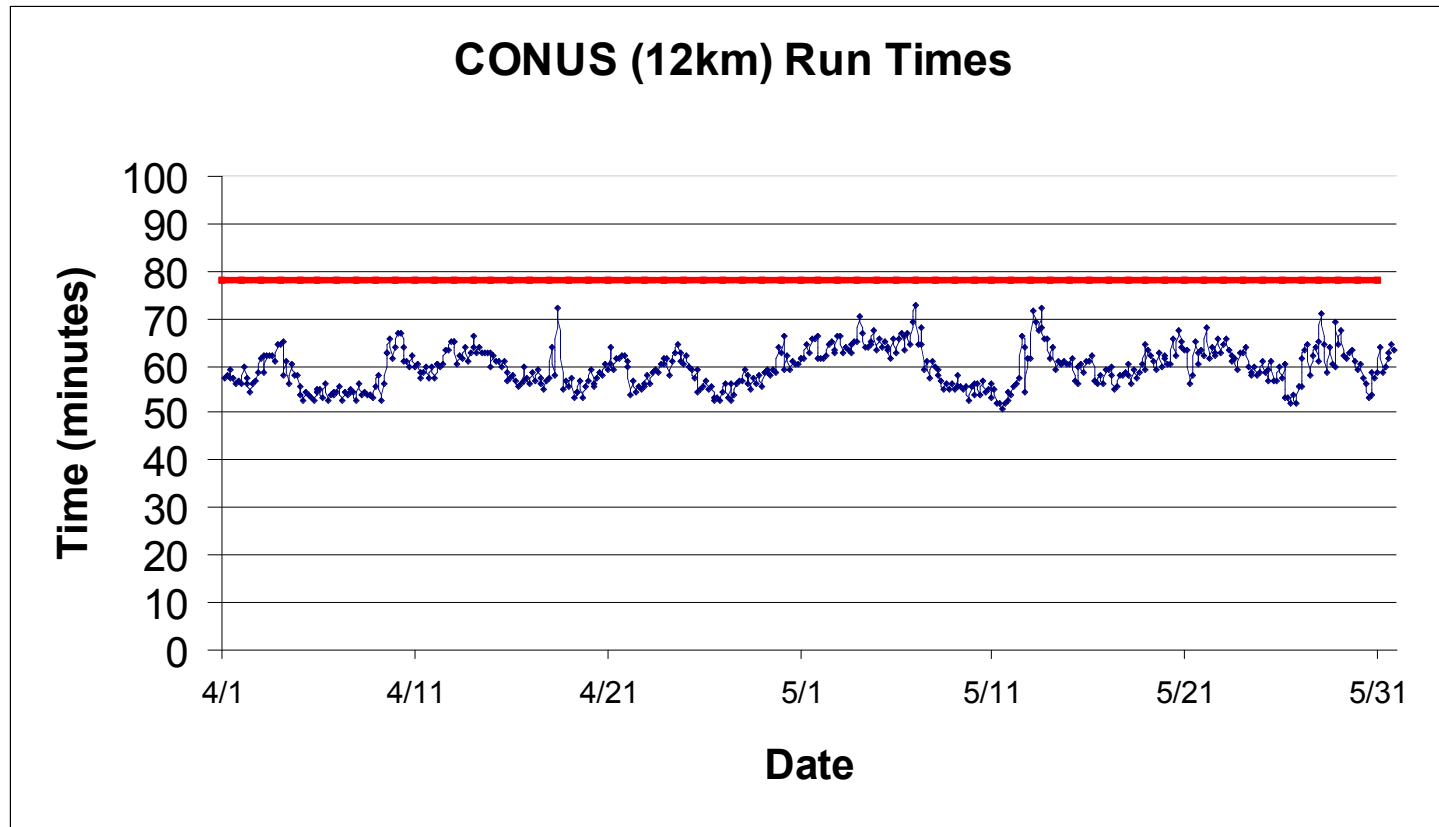


Adaptive time steps (1)

- Adaptive-time-step is a way to maximize the model time step while keeping the model numerically stable
- New in V3. Good to use for real-time run



Adaptive time steps (2): an example



On average, forecasts finish in 60 min (50-73min)
as compared to 79 min standard runtime

Adaptive time steps (3)

Namelist control: `&domains` * **USE WITH CARE**

<code>use_adaptive_time_step</code>	logical switch
<code>step_to_output_time</code>	whether to write at exact history output times
<code>target_cfl</code>	maximum cfl allowed (1.2)
<code>max_step_increase_pct</code>	percentage of time step increase each time; set to 5, 51, 51 (larger value for nest)
<code>starting_time_step</code>	in seconds; e.g. set to 4*DX
<code>max_time_step</code>	in seconds; e.g. set to 8*DX
<code>min_time_step</code>	in seconds; e.g. set to 4*DX



Digital filter initialization (1)

Digital filter initialization is a simple way to remove initial model imbalance:

- May be introduced by simple interpolation, different topography, or by objective analysis, or data assimilation
- It may generate spurious gravity waves in the early simulation hours, which could cause erroneous precipitation, numerical instability and degrade subsequent data assimilation



Digital filter initialization (2)

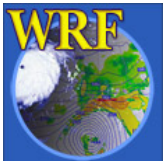
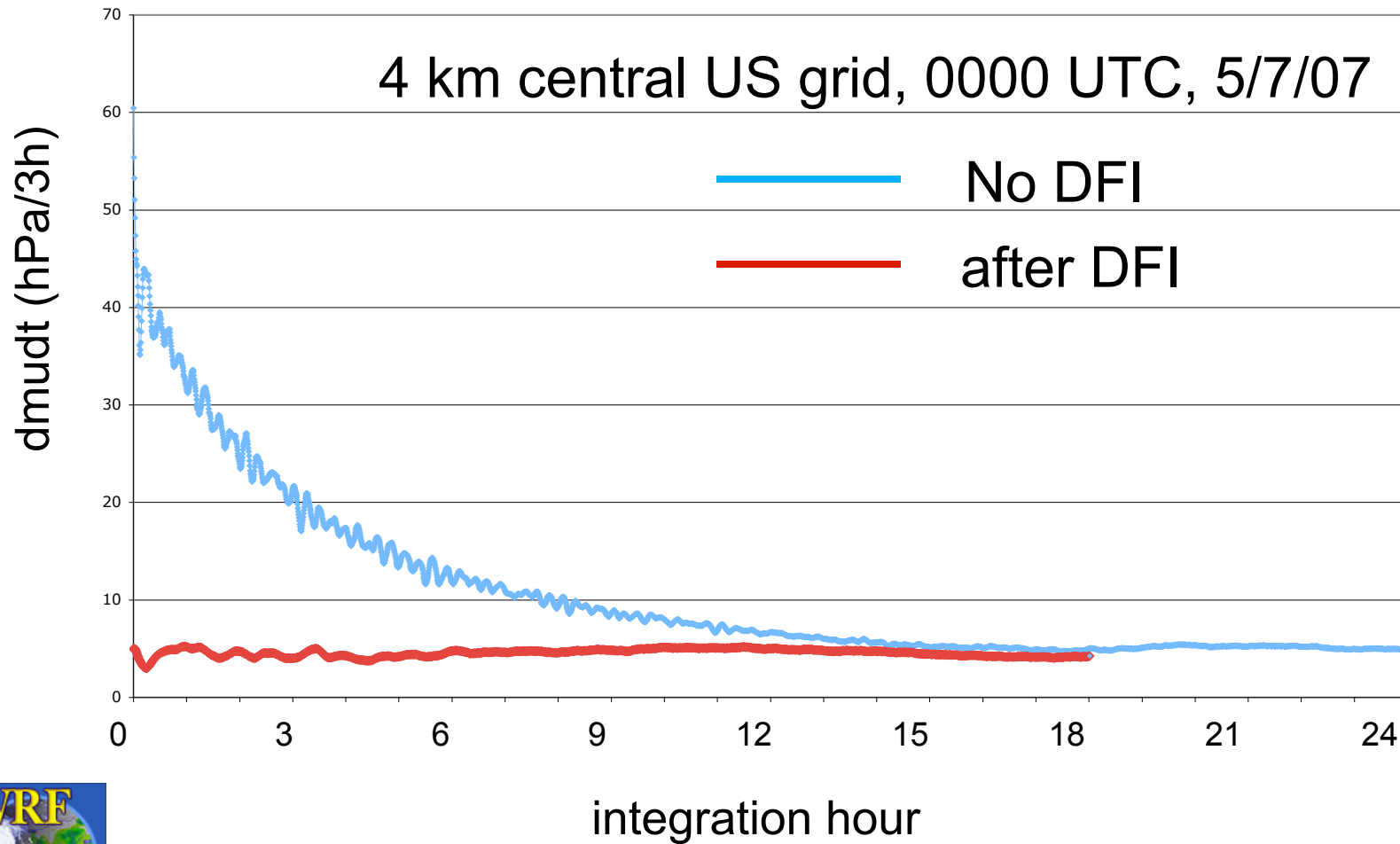
Using DFI

- can construct consistent model fields which do not exist in the initial conditions, e.g. vertical motion, cloud variables
- may reduce the spin-up problem in early simulation hours

DFI is done after program **real**, or data-assimilation step, just before model integration



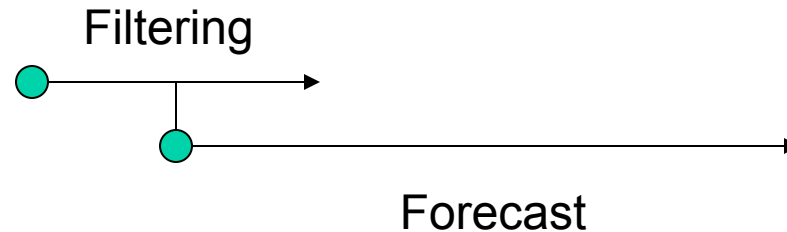
Digital filter initialization (3)



Digital filter initialization (4)

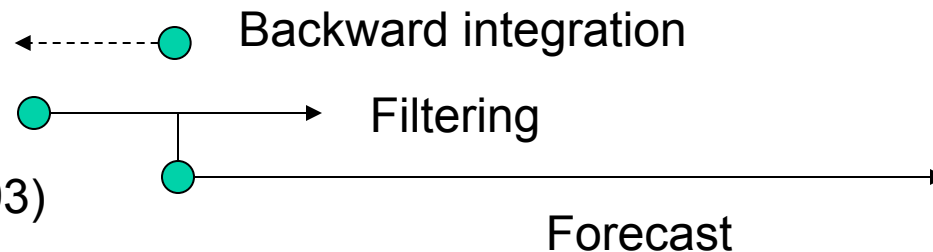
DFL:

(Lynch and Huang, 1994)



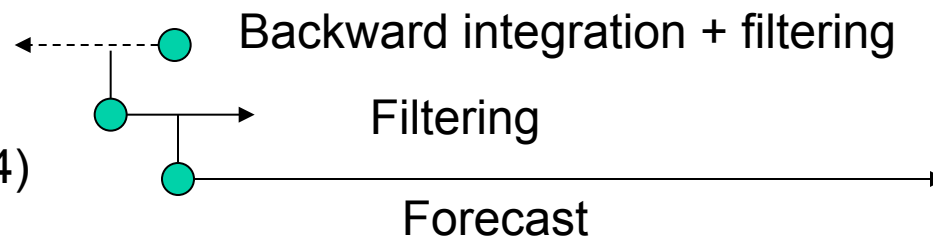
DDFI:

(Huang and Lynch, 1993)



TDFI:

(Lynch and Huang, 1994)



Digital filter initialization (5)

Namelist control: **&dfi**

dfi_opt: dfi options: 0: no DFI; 1: DFL; 2: DDFI; 3: TDDFI (recommended)

dfi_nfilter: filter options 0 - 8, recommended: 7

dfi_cutoff_seconds : cutoff period

dfi_write_filtered_input : whether to write filtered IC

dfi_bckstop_* : stop time for backward integration

dfi_fwdstop_* : stop time for forward integration

related namelists: **examples.namelist**



Global application

Setup mostly done in WPS:

`map_proj = 'lat-lon'`

`e_we, e_sn`: `geogrid` will compute dx, dy

See template `'namelist.wps.global'`

In the model stage:

`fft_filter_lat`: default value is 45 degrees

Caution: some options do not work, or have been tested with global domain. Start with template

`'namelist.input.global'`



Automatic moving nest options

Tropical cyclone / typhoon / hurricane applications:

vortex_interval: time interval when vortex location is estimated

max_vortex_speed: used to compute the search radius for vortex location

corral_dist: how far the vortex can move near the parent domain boundary (number of grids)

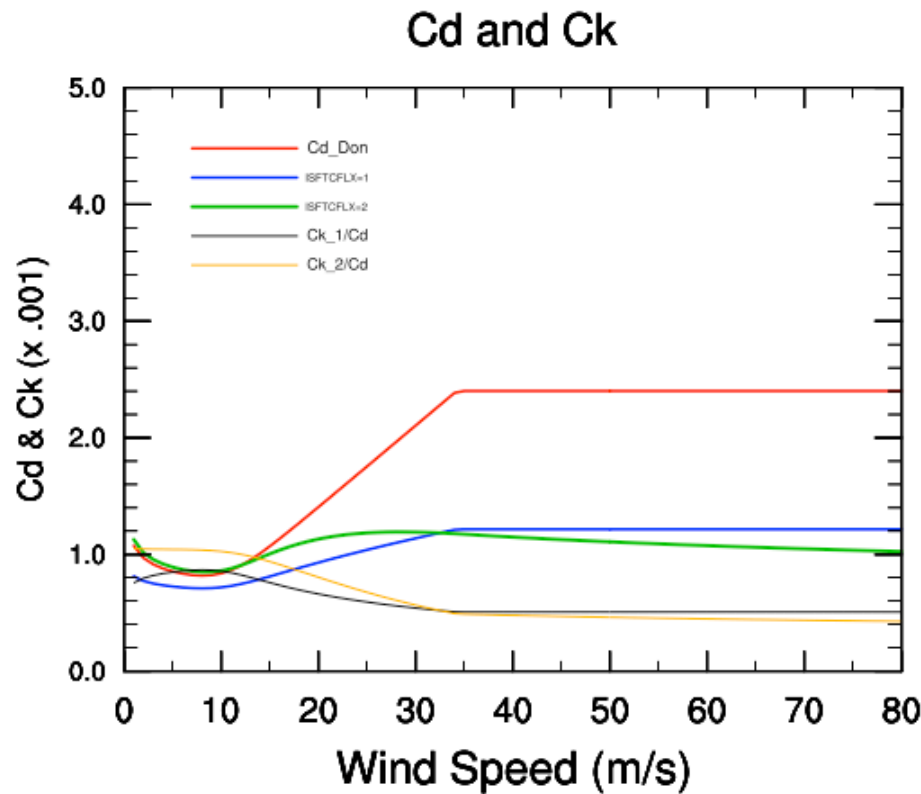
track_level: e.g. 700 or 500 mb

time_to_move: hold nests still until this time



TC options (1)

isftcflx: alternative C_d (Donelan) and C_k ($=1$, const z_0q ; $=2$, Garratt) formulation for TC application



TC options (2)

`omlcall`: simple ocean mixed layer

`oml_hml0`: initial ocean mixed layer depth

`oml_gamma`: lapse rate in deep water

The ocean mixed layer model can also be initialized with real-data, e.g. HYCOM

http://www.mmm.ucar.edu/wrf/users/hurricanes/wrf_ahw.html



tracer options

`tracer_opt:` = 2, activate tracers (default no. is 8)

One would need some way to initialize the tracer



IO quilting: &namelist_quilt

Parallel I/O control:

nio_tasks_per_group (>0) : allow IO to be done on separate processors. Performance improvement for large domain runs. A value of 2 to 4 works well.

io_groups (>1) : number of I/O streams that the quilting applies.



Time Series Output (1)

- It is a special output in text format with file name like
prefix.d<domain>.TS
- It outputs 14 surface variables at every time step:
e.g. 10 m u/v, 2 m T/qv, precipitation, radiation, surface fluxes
- One file per location/weather station



Time Series Output (2)

- Not a namelist option
- Depends the presence of a file called '**tslist**' (a sample of the file is available in **WRFV3/run/**)

```
#-----#  
# 24 characters for name | pfx | LAT | LON |  
#-----#  
Cape Hallett          hallt -72.330  170.250  
McMurdo Station      mcm    -77.851  166.713
```

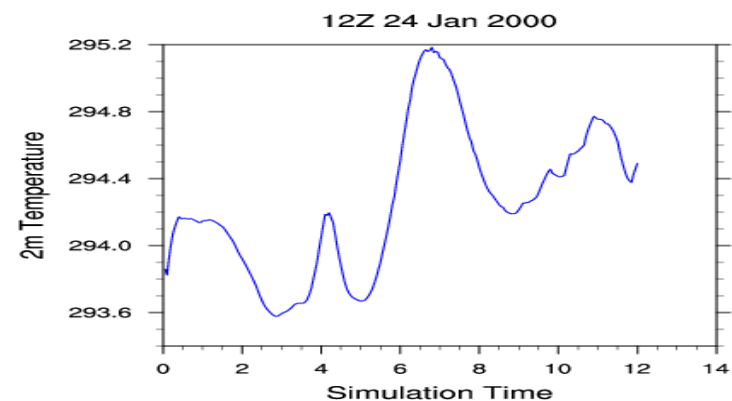
- This file provides a list of locations where you would like to output time series
- More information in **run/README.tslist** and User's Guide, Chapter 5



Time Series Output (3)

Content in hallt.d01.TS:

```
Cape Hallett          1  1 hallt ( 36.710, -79.000) ( 41, 38)
( 36.600, -79.142) 159.6 meters
1      0.050000      1  41  38      275.47397      0.00288
3.52110      -2.34275  99988.76563      244.81276
0.00000      -29.94841      4.09765      273.90295      278.20197
0.00000      0.00000      0.00000
1      0.100000      1  41  38      275.56287      0.00282
3.14414      -2.05875  99956.98438      244.81276
0.00000      -25.64095      4.18446      273.78323      278.18314
0.00000      0.00000      0.00000
```



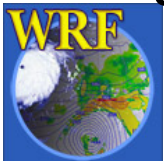
Recommended

Start with the **namelist template** in a particular test directory, and the options specified in them, and make modifications.

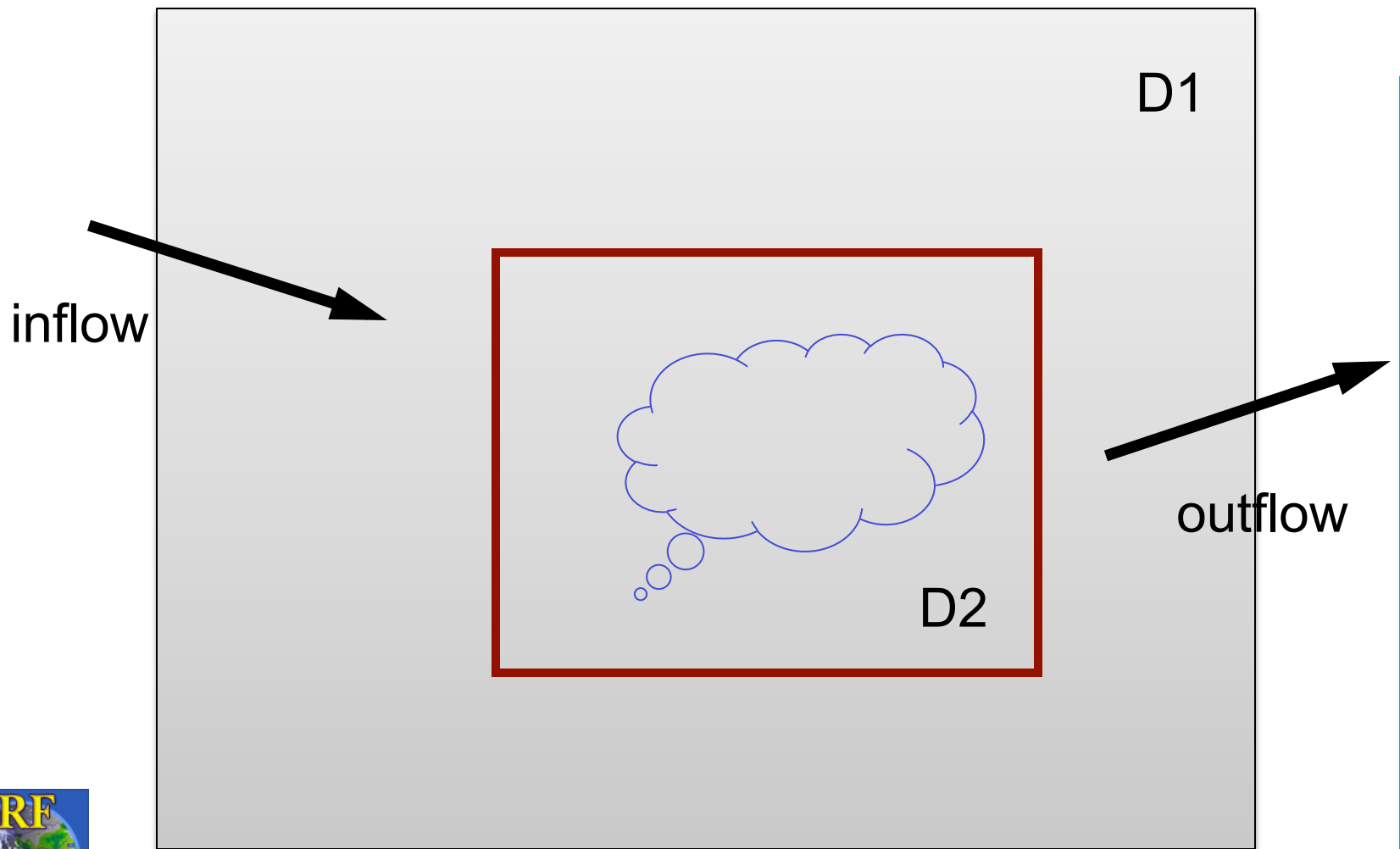
For special applications, look for related namelists in the file *examples.namelist* in *test/em_real/* directory.

Chapter 5 of ARW User's Guide, pages 5-23 – 5-25: examples for various applications.

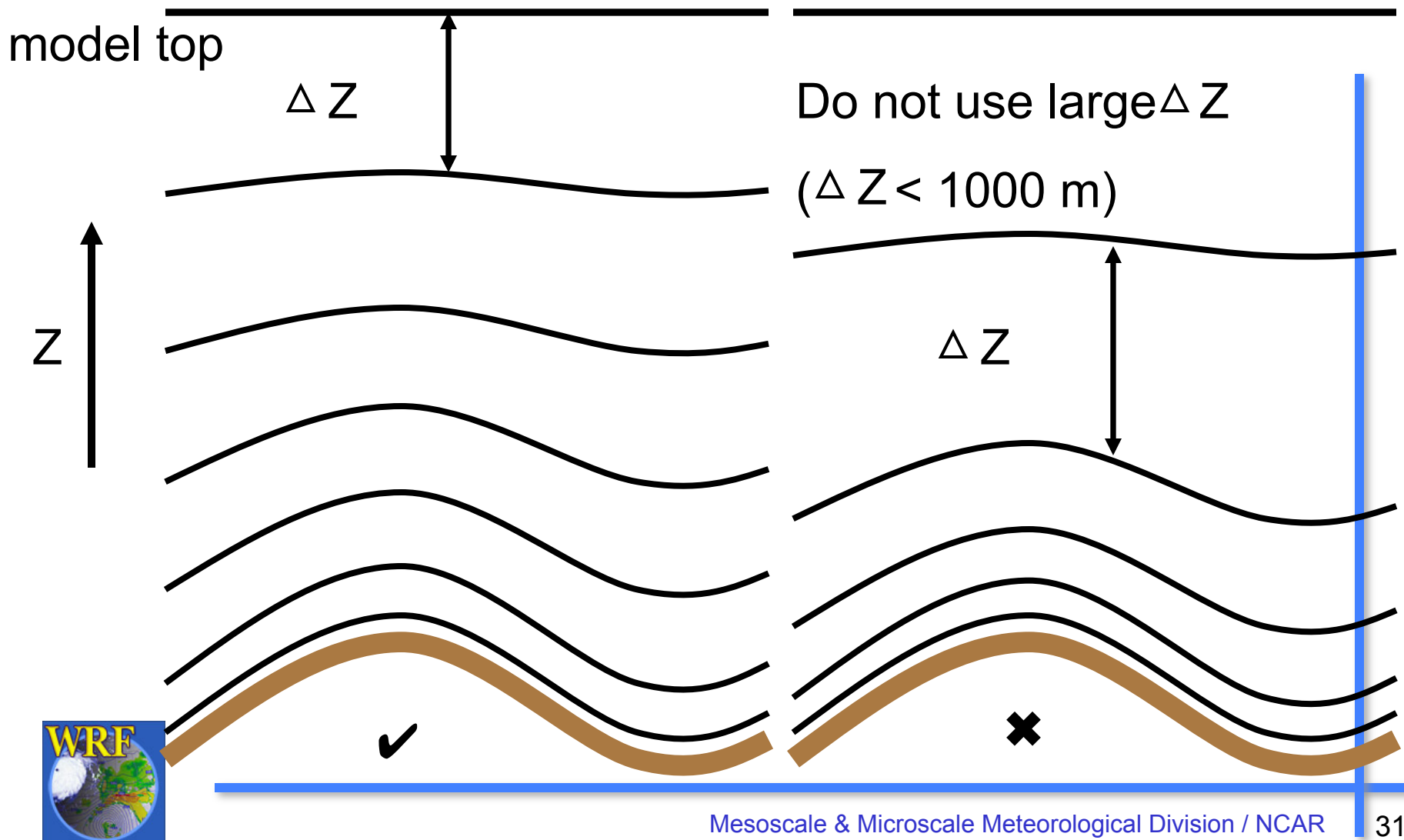
For more information on global extension, DFI and adaptive time step, read Tech Note, and User's Guide.



Note on Configuring Domains: Horizontal



Note on Configuring Domains: Vertical



Reference Book:

Numerical Weather and Climate Prediction,
2011. By Thomas Warner, Cambridge
University Press.

