WRF:
More Runtime Options

Wei Wang

Jan 2024

Mesoscale and Microscale Meteorology Laboratory, NCAR

This material is based upon work supported by the National Center for Atmospheric Research, which is a major facility sponsored by the National Science Foundation under Cooperative Agreement No. 1852977.

More options

- Some useful *runtime* options:
  - Vertical interpolation options (program *real.exe,* &domains)
  - Base state parameters (*real.exe,* &dynamics)
  - Defining vertical model levels (*real.exe,* &domains*)*
  - Physics suites (&physics)
  - IO options (&time_control)
  - Options for long simulations (&physics)
  - Adaptive-time step (&domains)
  - Digital filter (&dfi_control)
  - Stochastic parameterization schemes (&stoch)
  - Tracer (&dynamics) / trajectory (&physics, &domains)
  - Optional output (various)
  - IO quilting (&namelist_quilt)
- Time series output (surface and profile)

More options

- Some useful *runtime* options:
  - Vertical interpolation options (program *real.exe,* &domains)
  - Base state parameters (*real.exe,* &dynamics)
  - Defining vertical model levels (*real.exe,* &domains*)*
  - Physics suites (&physics)
  - IO options (&time_control)
  - Options for long simulations (&physics)
  - Adaptive-time step (&domains)
  - Digital filter (&dfi_control)
  - Stochastic parameterization schemes (&stoch)
  - Tracer (&dynamics) / trajectory (&physics, &domains)
  - Optional output (various)
  - IO quilting (&namelist_quilt)
- Time series output (surface and profile)

namelist.input

general namelist records:

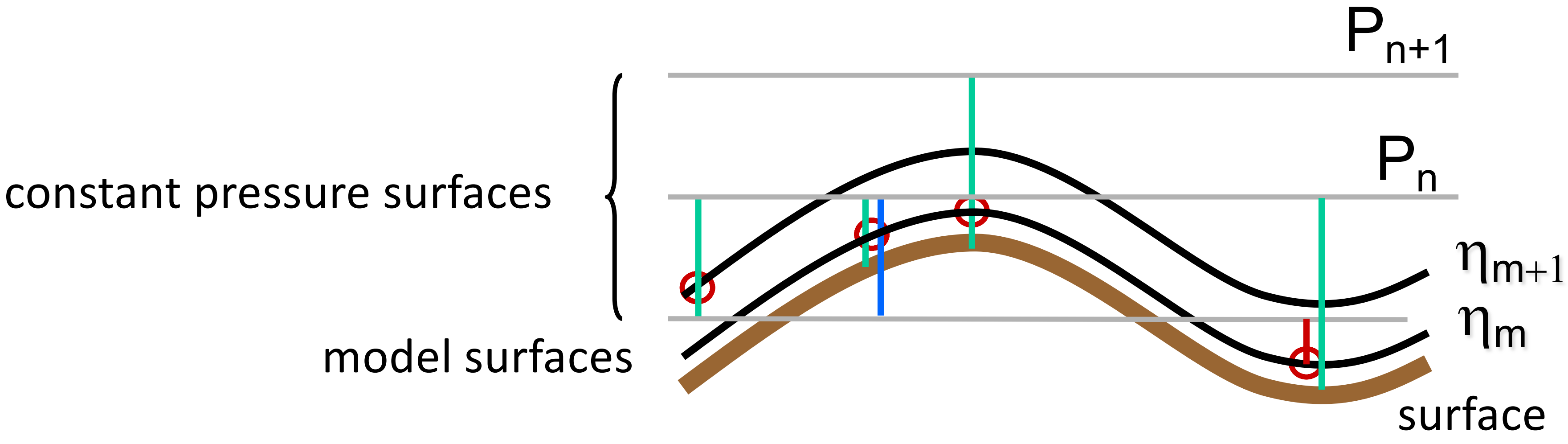
- **`&time_control`**
- **`&domains`**
- **`&physics`**
- **`&dynamics`**
- **`&bdy_control`**
- **`&namelist_quilt`**

specialized namelist records:

- **`&dfi_control`**
- **`&fdda`**
- **`&stoch`**
- **`&diags`**
- **`&scm`**
- **`&tc`**
- **`&noah_mp`**

Look for these in test/em_real/**`examples.namelist`**

Vertical interpolation options (1)



Program **`real.exe`**, **`&domains`**:

- **`use_surface`**: whether to use surface level data (default is true)
- **`lagrange_order`**: linear, quadratic (default) or cubic

Vertical interpolation options (2)

Program **`real`** only, **`&domains`**:

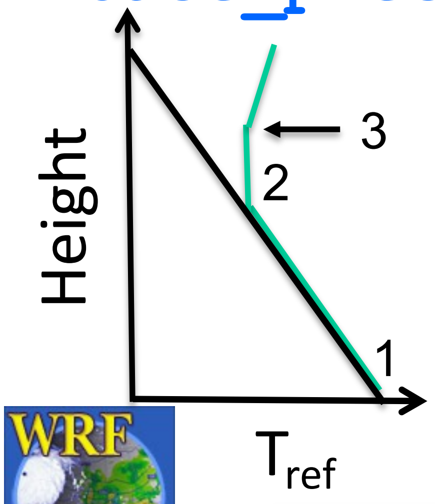
- **`use_levels_below_ground`**: whether to use data below the ground, T/F (default T)
- **`lowest_lev_from_sfc`**: whether surface data is used to fill the lowest model level values, T/F (default F)
- **`force_sfc_in_vinterp`**: number of levels to use surface data, default is 1
- **`extrap_type`**: how to do extrapolation: 1 - use 2 lowest levels; 2 – constant (default)
- **`t_extrap_type`** : extrapolation option for temperature: 1 - isothermal; 2 - 6.5 K/km (default); 3 - adiabatic

Look for these in examples.namelist

Base State Parameters

The following could be varied (program **`real`**, **`&dynamics`**):

`base_temp`[1]	Base state surface temperature (default 290 K)
`iso_temp`[2]	Base state stratosphere temperature (default 200 K)
`base_pres_strat`[3]	Pressure at which the stratosphere temperature lapse rate changes (default 0 hPa)



The purpose is to minimize perturbation fields to improve solution accuracy when discretized.

Defining Vertical Levels

Program **`real.exe`**:

Aim to produce a set of vertical levels so that the *thickness* of the layers varies smoothly with height.

Method 1: Specifying the levels

Coordinate Definition: $$\eta = \frac{p_d - p_t}{p_s - p_t}$$

```
&domains
 e_vert     = 56
 eta_levels = 1.,0.9945,0.9882,0.9810,0.9727,0.9632, 0.9524,
                 0.9402,0.9264,0.9109,0.8936,0.8745,0.8534,
                 0.8303,0.8054,0.7785,0.7500,0.7200,0.6886,
                 ... 0.0
```

Defining Vertical Levels

Program **real.exe**: *Method 2: Computing levels*

```
&domains
auto_levels_opt = 2  (default since v4.0)
dzstretch_s = 1.1 ~ 1.3
dzstretch_z = 1.02 ~ 1.1
dzbot = 20 ~ 50
max_dz = < 1000
```

```
kness =   30.0 m
kness =   35.7 m
kness =   42.5 m
kness =   50.4 m
kness =   59.7 m
kness =   70.5 m
kness =   83.1 m
kness =   97.6 m
kness =  114.2 m
kness =  133.0 m
```

```
……
Full level index =   35     Height = 11815.4 m      Thickness =  748.8 m
Full level index =   36     Height = 12585.8 m      Thickness =  770.4 m
Full level index =   37     Height = 13378.7 m      Thickness =  792.9 m
Full level index =   38     Height = 14191.4 m      Thickness =  812.6 m
Full level index =   39     Height = 15004.0 m      Thickness =  812.6 m
Full level index =   40     Height = 15816.6 m      Thickness =  812.6 m
Full level index =   41     Height = 16629.3 m      Thickness =  812.6 m
Full level index =   42     Height = 17441.9 m      Thickness =  812.6 m
Full level index =   43     Height = 18254.5 m      Thickness =  812.6 m
Full level index =   44     Height = 19067.1 m      Thickness =  812.6 m
```

Defining Vertical Levels

Pro

&d

au

dz

dz

dz

ma

```
Full level index =    1      Height =     0.0 m
Full level index =    2      Height =    30.0 m        Thickness =   30.0 m
Full level index =    3      Height =    65.7 m        Thickness =   35.7 m
Full level index =    4      Height =   108.2 m        Thickness =   42.5 m
Full level index =    5      Height =   158.6 m        Thickness =   50.4 m
Full level index =    6      Height =   218.3 m        Thickness =   59.7 m
Full level index =    7      Height =   288.8 m        Thickness =   70.5 m
Full level index =    8      Height =   371.9 m        Thickness =   83.1 m
Full level index =    9      Height =   469.6 m        Thickness =   97.6 m
Full level index =   10      Height =   583.8 m        Thickness =  114.2 m
Full level index =   11      Height =   716.8 m        Thickness =  133.0 m
......
Full level index =   35      Height = 11815.4 m        Thickness =  748.8 m
Full level index =   36      Height = 12585.8 m        Thickness =  770.4 m
Full level index =   37      Height = 13378.7 m        Thickness =  792.9 m
Full level index =   38      Height = 14191.4 m        Thickness =  812.6 m
Full level index =   39      Height = 15004.0 m        Thickness =  812.6 m
Full level index =   40      Height = 15816.6 m        Thickness =  812.6 m
Full level index =   41      Height = 16629.3 m        Thickness =  812.6 m
Full level index =   42      Height = 17441.9 m        Thickness =  812.6 m
Full level index =   43      Height = 18254.5 m        Thickness =  812.6 m
Full level index =   44      Height = 19067.1 m        Thickness =  812.6 m
```

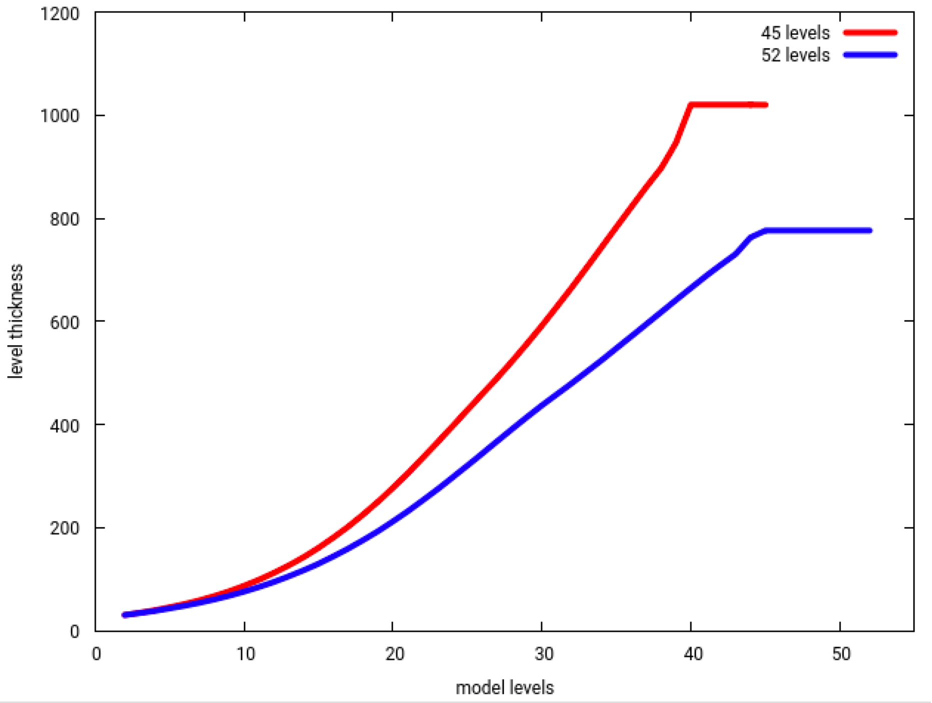
Defining Vertical Levels

Program **`real.exe:`** *Method 2*

Example on the right:

`p_top_requested = 5000`
`e_vert = 52 (45)`

See WRF Initialization/Setting Model Vertical Levels in User's Guide



Use of physics suite

Since 3.9, physics can be selected as a suite. These represent well-tested physics (**`&physics`**).

```
physics_suite = ‘tropical’

mp_physics = 6, 6,
cu_physics = 16, 16,
ra_lw_physics = 4, 4,
ra_sw_physics = 4, 4,
bl_pbl_physics = 1, 1,
sf_sfclay_physics = 91, 91,
sf_surface_physics = 2, 2,
```

```
physics_suite = ‘CONUS’

mp_physics = 8, 8,
cu_physics = 6, 6,
ra_lw_physics = 4, 4,
ra_sw_physics = 4, 4,
bl_pbl_physics = 2, 2,
sf_sfclay_physics = 2, 2,
sf_surface_physics = 2, 2,
```

* *Note* other configuration choices can have an impact on model results

Use of physics suite

To turn an option off for a particular domain:

```
physics_suite = 'tropical'
```

+

```
cu_physics = -1, 0,
```

-1: using suite option

To overwrite one or more with other options:

```
physics_suite = 'CONUS'
```

+

```
cu_physics = 16, 16,
bl_pbl_physics = 1, 1,
sf_sfclay_physics = 1, 1,
```

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

Look for the list in

`Registry/registry.io_boilerplate`

IO Control (2)

Specify input and output files explicitly in
`&time_control`

```
auxinput1_inname =“/mydirectory/met_em.d<domain>.<date>”
```

- explicitly specify input file (its name and directory)

```
history_outname = “/mydirectory/wrfout_d<domain>_<date>”
```

- explicitly specify history output file (its name and directory)

Look for these in
`Registry/registry.io_boilerplate`

Additional Output Option

- **`wrfout`** files: default, data goes to *stream* 0
- Additional output goes to auxiliary output streams
- WRF has 24 *streams* by default
- Require some knowledge of registry
- To output to any of the auxiliary streams, these output specifications need to be added to the namelist section *&domains*

`auxhistN_outname, io_form_auxhistN,`
`auxhistN_interval, frames_per_auxhistN`

Here N is the stream number

IO Control (3)

Optional output in **`&time_control`**

1. Change **`Registry.EM_COMMON`** and recompile:

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

2. Edit namelist.input to output these variables:

```
auxhist3_outname = “rainfall_d<domain>”
auxhist3_interval = 10, 10,
frames_per_auxhist3 = 1000, 1000,
io_form_auxhist3 = 2
```

Good for production runs

IO Control (4)

There is an alternative way to add/remove output fields at **runtime** (state variables in Registry only)

1. namelists in **`&time_control`**:

```
iofields_filename(max_dom) = ‘my_output.txt’ ,
ignore_iofields_warning = .true.
```

2. prepare a text file (‘`my_output.txt`’) to select io fields:

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

← syntax in the file

3. set other namelists under **`&time_control`**:

```
auxhist3_outname = “rainfall_d<domain>”
auxhist3_interval = 10, 10,
frames_per_auxhist3 = 1000, 1000,
io_form_auxhist3 = 2
```

Good for development runs

See ‘WRF Output/Run-Time IO’ section in User’s Guide

Options for long simulations (1)

Update control for lower boundary fields: allow SST, seaice, monthly vegetation fraction and albedo to be updated regularly during a model run:

`sst_update:` 0 – no update

1 – update all above fields (in **`&physics`**)

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

Other namelists required in **`&time_control`**:

`auxinput4_inname` = “wrflowinp_d<domain>”

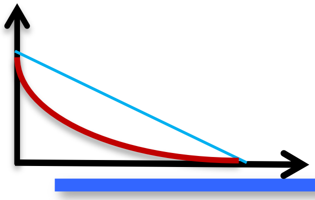
`auxinput4_interval` = 360, 360,

`io_form_auxinput4` = 2 (netCDF)

See ‘Run-time Capabilities/SST Update’ in User’s Guide

Options for long simulations (2) (`&physics`)

`sst_skin`	diurnal water temp update
`tmn_update`	deep soil temp update, used with lagday
`lagday`	averaging time in days
`bucket_mm`	bucket reset value for rainfall (e.g. `rainc=i_rainc*bucket_mm+rainc`)
`bucket_j`	bucket reset value for radiation fluxes
`spec_exp`	exponential multiplier for boundary zone ramping (set in real, &bdy_control). Usually used with wider boundary zone

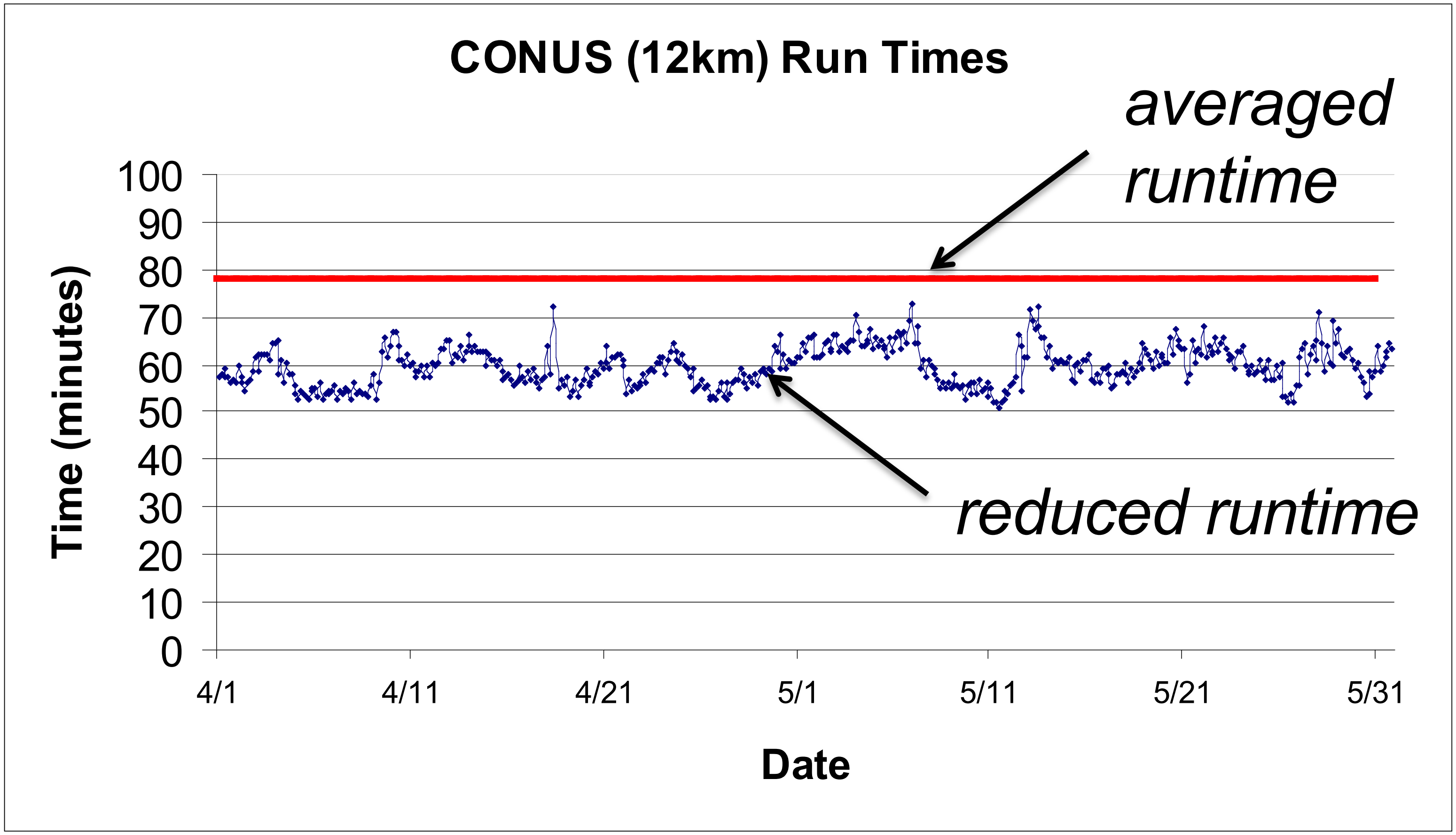


Adaptive time steps (1)

- Adaptive-time-step is a way to maximize the model time step while keeping the model numerically stable.
- Good to use for real-time run.
- May not work in combination with other options.

Also see ‘Run-time Capabilities/Adaptive Time Stepping’ section in User’s Guide

Adaptive time steps (2): an example



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

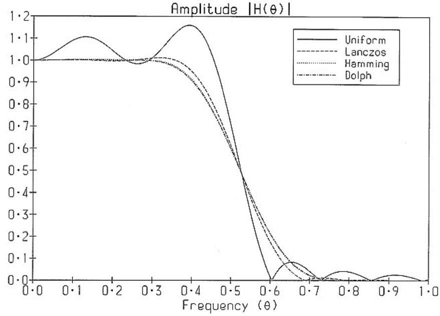
Adaptive time steps (3)

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

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

Digital Filter Initialization (DFI) (1)

- DFI is a way to use a low-pass filter to remove noise in model initial conditions
- Imbalances in model IC
  - May be introduced by interpolation, different topography, or by objective analysis, and data assimilation
  - May generate spurious gravity waves in the early simulation hours, which could cause erroneous vertical motion and precipitation, numerical instability and degrade subsequent data assimilation
- Useful for short-range model runs (1-6 hours)



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
- Useful for short-range (1-6 h) forecasts and cycling with data assimilation

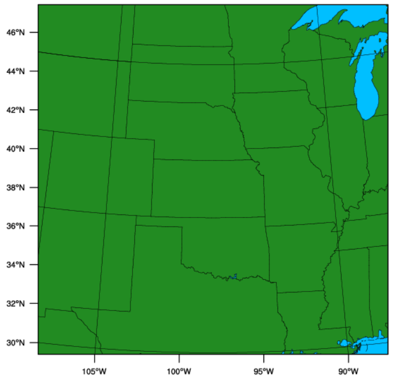
DFI is done after program **`real`**, or data-assimilation step

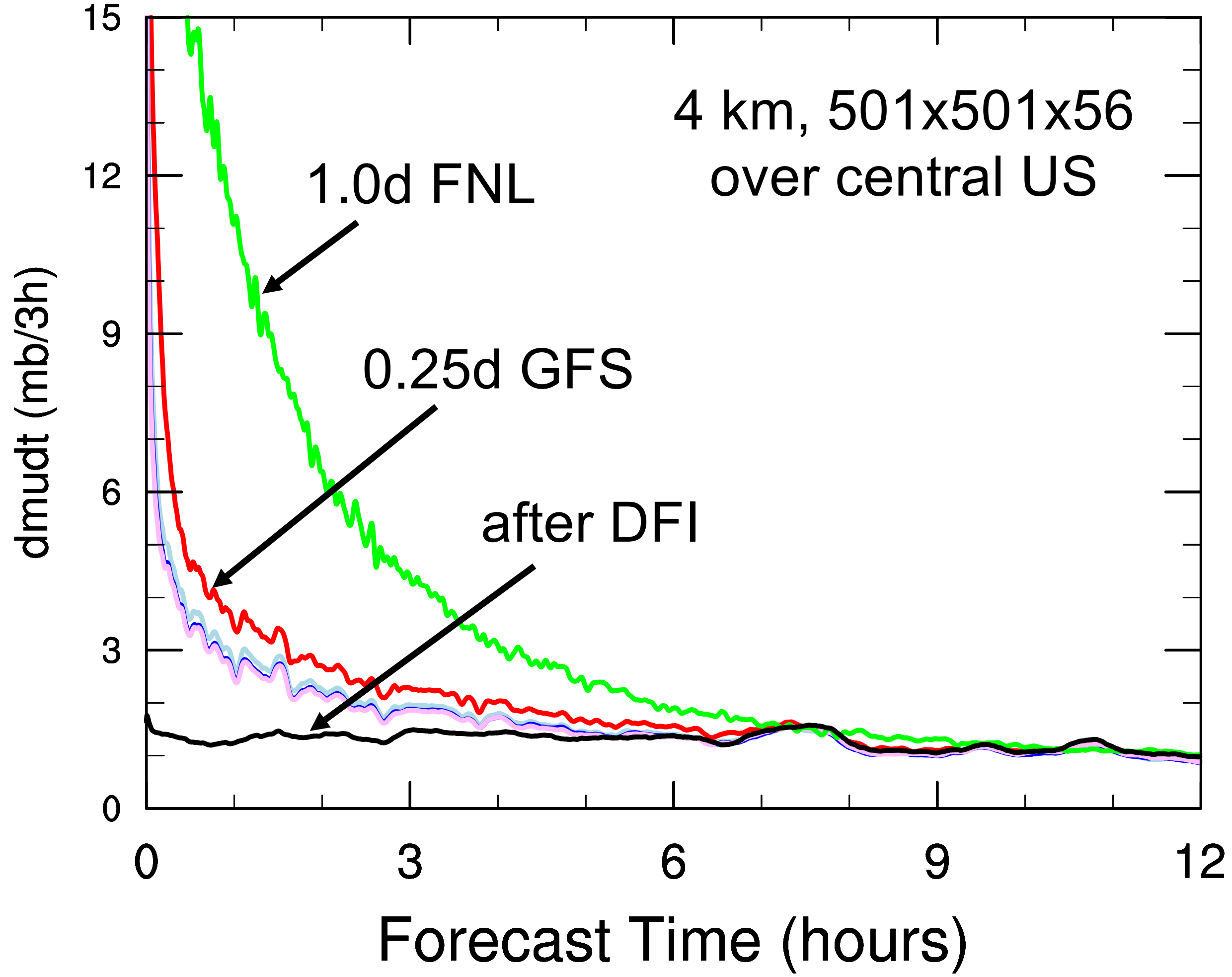
See ‘Run-time Capabilities/Digital Filter Initialization’ section of the Users’ Guide.

Digital filter initialization (3)

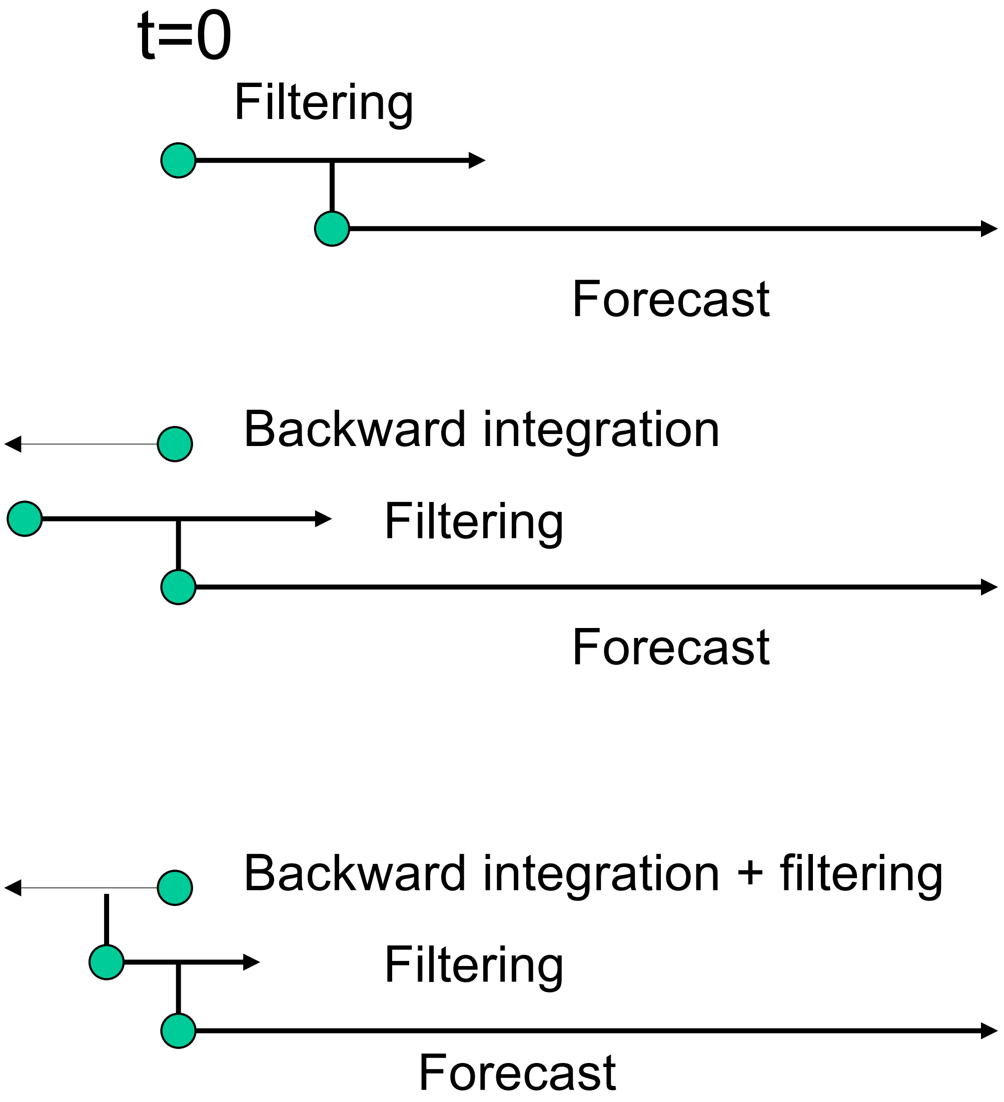
Use of DFI helps to damp high pressure tendencies in early forecast

4 km WRF domain





Digital filter initialization (4)



t=0

DFL:

(Lynch and Huang, 1994)

Filtering

Forecast

DDFI:

(Huang and Lynch, 1993)

Backward integration

Filtering

Forecast

TDFI:

(Lynch and Huang, 1994)

Backward integration + filtering

Filtering

Forecast

Digital filter inilialization (5)

Namelist control: **`&dfi_control`**

- **`dfi_opt`**: dfi options: 0: no DFI; 1: DFL; 2: DDFI; 3: TDFI (recommended)
- **`dfi_nfilter`**: filter options 0 - 8, recommended: 7
- **`dfi_cutoff_seconds`** : cutoff period
- **`dfi_write_filtered_input`** : whether to write filtered IC to a file
- **`dfi_bckstop_*`** : stop time for backward integration
- **`dfi_fwdstop_*`** : stop time for forward integration

Related namelists: examples.namelist

To get pressure tendency data, set **`diag_print=1`** or **`2`**

Stochastic parameterization schemes

These are the ways to stochastically perturb forecasts (`&stoch`)

`skebs:` = 1, activate SKEBS

`nens:` = N, an integer that controls the random number stream; a different integer will give a differently perturbed forecast

`perturb_bdy:` = 1, use SKEB pattern; = 2, use user-provided pattern

`sppt:` = 1, activate stochastically pert parameterization tendencies

`spp:` = 1, activate stochastically perturbed parameterization physics

Note: sppt and spp only applicable for a subset of physics

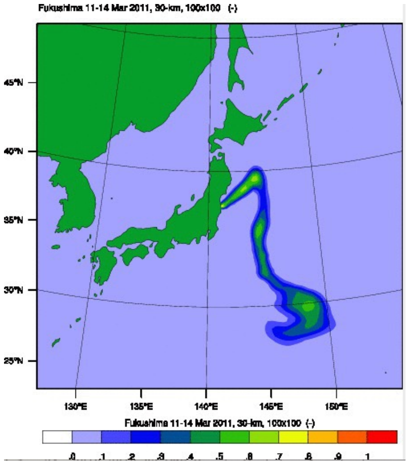
Also see ‘Run-time Capabilities/Stochastic Parameterization schemes’ section in User’s Guide

Also see http://www.cgd.ucar.edu/~berner/skebs.html

Tracer option

Add the following in **`&dynamics`** to activate tracer option (default no. is 8: with array names **`tr17_1, tr17_2, …, tr17_8`**):

`tracer_opt = 2,`



One would need some way to initialize the tracer. A simple initialization can be found in program real (**`dyn_em/module_initialize_real.F`**)

Trajectory option

Add the following in **`&physics`** to activate trajectory option:

`traj_opt` = 1,

And set the number of trajectories in **`&domains`**:

`num_traj` = 1000, (default value)

Since V3.9: it can output meteorological variables, as well as chemistry ones, along the trajectories. Full document at

https://www2.acom.ucar.edu/sites/default/files/wrfchem/Trajectory.desc_.pdf

Additional Output Option (1)

`prec_acc_dt = 60.:` in `&physics,` unit: minute

Output precipitation in a time interval (e.g. 60 min):

- `PREC_ACC_C,` for convective rain
- `PREC_ACC_NC,` for explicit rain
- `SNOW_ACC_NC,` for explicit snow

Data goes to history file. Interval should be the same as history intervals.

Additional Output Option (2a)

```
&diags
 p_lev_diag = 1.
 num_press_levels = 4,
 press_levels = 85000,70000,50000,20000
```

Output a few met fields on pressure levels :

`U_PL, V_PL, S_PL, T_PL, Q_PL, RH_PL, GHT_PL, TD_PL`

Output goes to auxiliary stream 23, so need to set

```
auxhist23_outname, io_form_auxhist23,
auxhist23_interval, frames_per_auxhist23
```

Additional Output Option (2b)

```
&diags
 z_lev_diag = 1.
 num_z_levels = 4,
 z_levels = 80,150,300,3000
```

Output a few met fields on pressure levels :

`U_ZL, V_ZL, S_ZL, T_ZL, Q_ZL, RH_ZL, P_ZL, TD_ZL`

Output goes to auxiliary stream 22, so need to set

```
auxhist22_outname, io_form_auxhist22,
auxhist22_interval, frames_per_auxhist22
```

Additional Output Option (3)

```
&diags
 diag_nwp2 = 1
```

Output a few met fields on model levels :

```
   sealevelp, temperature, geoheight,
   pressure, umet, vmet, speed, dir, U10,
   V10, Q2, T2, RAIN, LIQRAIN, TPW, RH
```

Output goes to auxiliary stream 1, so need to set

```
   auxhist1_outname, io_form_auxhist1,
   auxhist1_interval, frames_per_auxhist1
```

Additional Output Option (4)

`nwp_diagnostics = 1:` **`(&time_control)`**

Output max 10 m wind speed, max helicity in 2 – 5 km layer, max w in updraft and downdraft below 400 mb, mean w in 2 – 5 km layer, and max column graupel in a time window between history output times.

Data goes to history file.

Additional Output Option (5)

`do_radar_ref = 1: (&physics)`

Compute and output radar reflectivity using parameters in respective microphysics. Works for options mp_physics = 2,4,6,7,8,10,14,16. Option 9, NSSL mp also produce radar reflectivity output.

Data (**`refl_10cm`**) goes to history file.

Additional Output Option (6)

`afwa_*_opt = 1:` (**`&afwa`**, with sub-options)

Output over 60 diagnostic variables to history file (for example, MSLP, precipitable water, cloud cover, visibility, etc.)

See **`Registry/registry.afwa`** for full listing.

Data goes to history as well as **`auxhist2`** file. Set `auxhist2_outname`, `io_form_auxhist2`, `auxhist2_interval`, `frames_per_auxhist2`

Additional Output Option (7)

`output_diagnostics = 1:` **`(&time_control)`**

Output max, min, time of max and min, mean value, standard deviation of the mean for 8 surface variables (T2, Q2, TSK, U10, V10, 10 m wind speed, RAINCV, and RAINNCV [time step rain]) in history output interval

Data goes to auxiliary stream 3 or **`auxhist3`**. Set `auxhist3_outname`, `io_form_auxhist3`, `auxhist3_interval`, `frames_per_auxhist3`

Additional Output Option (8)

More climate output (from RASM, **`&time_control`**):

`mean_diag = 1:` (with interval options)

`diurnal_diag = 1`

Output time-step and diurnal averaging of a number of surface variables and radiative fluxes at surface and top of atmosphere (e.g. monthly averages)

See **`run/README.rasm_diag`** for details, and **`Registry/registry.rasm_diag`** for full listing.

Data goes to auxhist5 and auxhist6 files.

Additional Output Option (9)

`acc_phy_tend = 1:` **`(&physics)`**

Output accumulated physics tendencies for u, v, T, qv, etc. at history output interval:
ATHMPTEN, AQVMPTEN, ATHCUTEN, AQVCUTEN, AUCUTEN, AVCUTEN,ATHBLTEN, AQVBLTEN, AUBLTEN, AVBLTEN, ATHRATENLW, ATHRATENSW…

- Useful for model diagnostics.

- Data goes to standard history file.

Additional Output Option (10)

`do_avgflx_em = 1:` **`(&dynamics)`**

Output history-time-averaged, mass-coupled advective velocities u, v and w:
`AVGFLX_RUM, AVGFLX_RVM, AVGFLX_RWM`

Useful for driving downstream transport model.

Data goes to standard history file.

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 fluxes, surface fluxes
- One file per location (e.g. at weather station), per domain

Time Series Output (2)

- It also outputs profiles of U, V, Th, Qv, PH (levels set by `max_ts_level`, default 15, from surface upward):

  ***prefix*.d<domain>.UU**

  ***prefix*.d<domain>.VV**

  ***prefix*.d<domain>.TH**

  ***prefix*.d<domain>.QV**

  ***prefix*.d<domain>.PH**

- One file per variable, per location (e.g. at weather station), and per domain.

Time Series Output (3)

- Requires a file called **'tslist'** present in working directory (a sample of the file is available in **`WRF/run/`**
- This file provides a list of names and locations where you would like to output time series. A sample file looks like this:

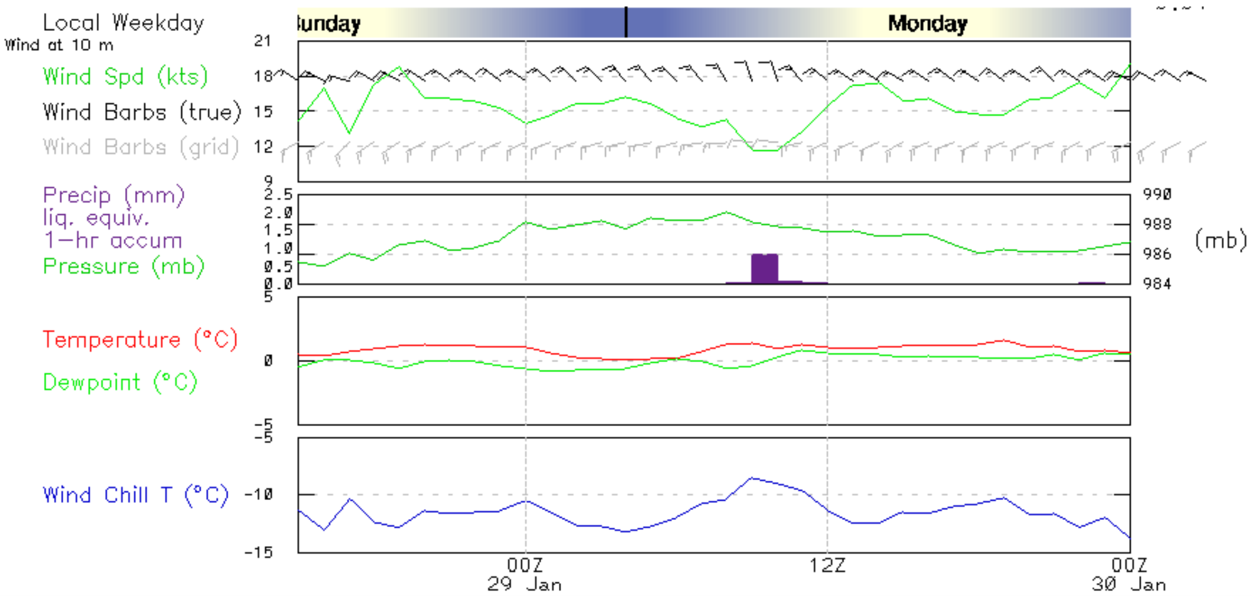
```
#-----------------------------------------------#
# 24 characters for name | pfx |  LAT  |   LON  |
#-----------------------------------------------#
Cape Hallett              hallt -72.330  170.250
McMurdo Station           mcm   -77.851  166.713
```

- If output more than 5 locations, use namelist **`max_ts_locs`** in **`&domains`**
- More information in **`run/README.tslist`** and 'WRF Output/Output Diagnostics/Time Series output' in UG

Time Series Output (4)

Sample data 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      -2.0
   25.64095       4.
   0.00000       0.0
   ......
```



IO quilting: &namelist_quilt

I/O quilting 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.

See 'Using IO Quilting' section, Chap 5, UG

Other ways to improve IO: 1) p-netCDF; 2) use netCDF4 compression option; 3) use io_form_history=102 to output patches of data

Recommended

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

WRF User's Guide, examples for various applications and physics suites.

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

To find which namelist record a namelist variable belongs, try `Registry/Registry.EM_COMMON` and other registry files.

For more information on DFI, adaptive time step and stochastic parameterizations, read WRF Tech Note and User's Guide.