WRF:
More Runtime Options

Wei Wang

NCAR/ESSL/MMM

July 2009

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
  - SST update and other options for long simulations
  - Adaptive-time step
  - Digital filter
  - Global runs
  - Moving nest
  - TC options
  - 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
   ```

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
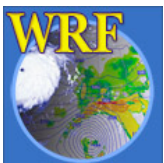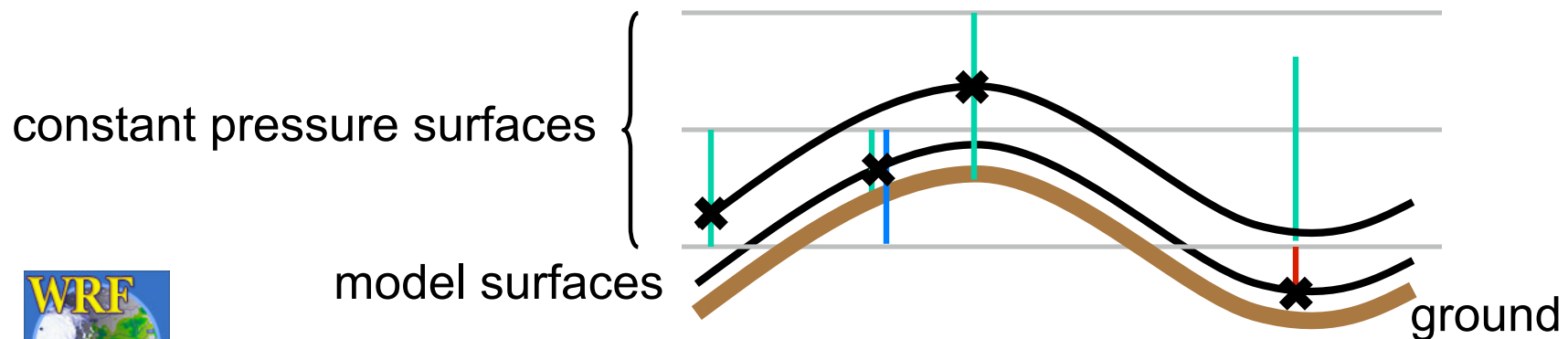
Vertical interpolation options (2)

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**

constant pressure surfaces

model surfaces

ground

SST update for long simulations (1)

Lower boundary update control: allow SST, seaice, monthly vegetation fraction and albedo to be updated 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, ..

To use the files in `wrf`, in `&time_control`, add

`auxinput4_inname` = "wrflowinp_d<domain>"

`auxinput4_interval` = 360

SST update for long simulations (2)

sst_skin   diurnal water temp update

tmn_update   deep soil temp update, used with lagday

lagday   averaging time

bucket_mm   bucket reset value for rainfall

bucket_j   bucket reset value for radiation fluxes

spec_exp   exponential multiplier for boundary zone ramping
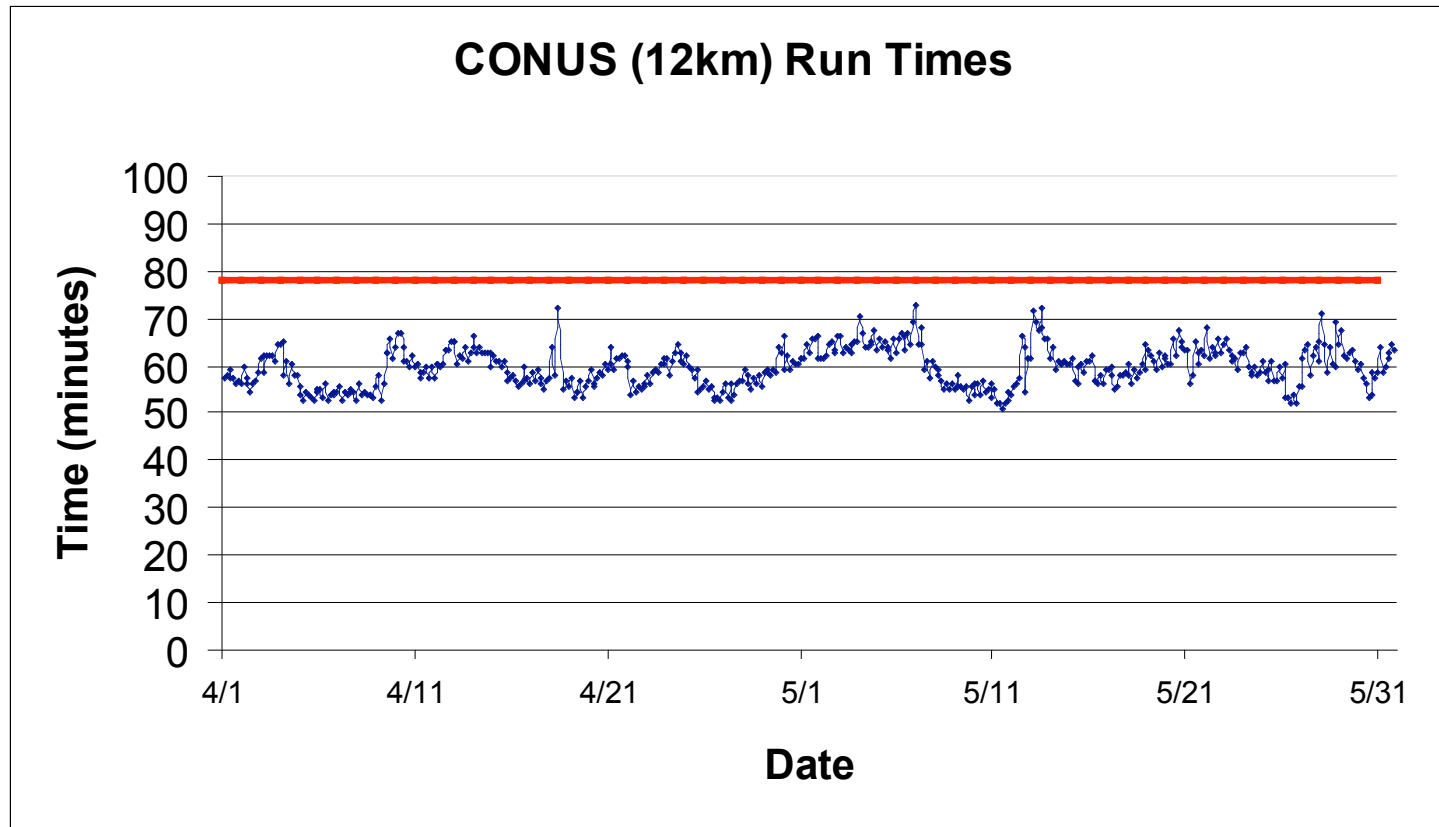
Adaptive time steps (1)

- Adaptive-time-step is a way to maximize the model time step while keeping the model numerically stable

- New in V3. Works well for single domain. Good to use for real-time run

Adaptive time steps (2): an example

CONUS (12km) Run Times

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

Adaptive time steps (3)

Namelist control: **&domains**

    use_adaptive_time_step : logical switch

    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

    * USE WITH CARE

Digital filter initialization (1)

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

– May be introduced by simple interpolation, 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
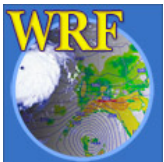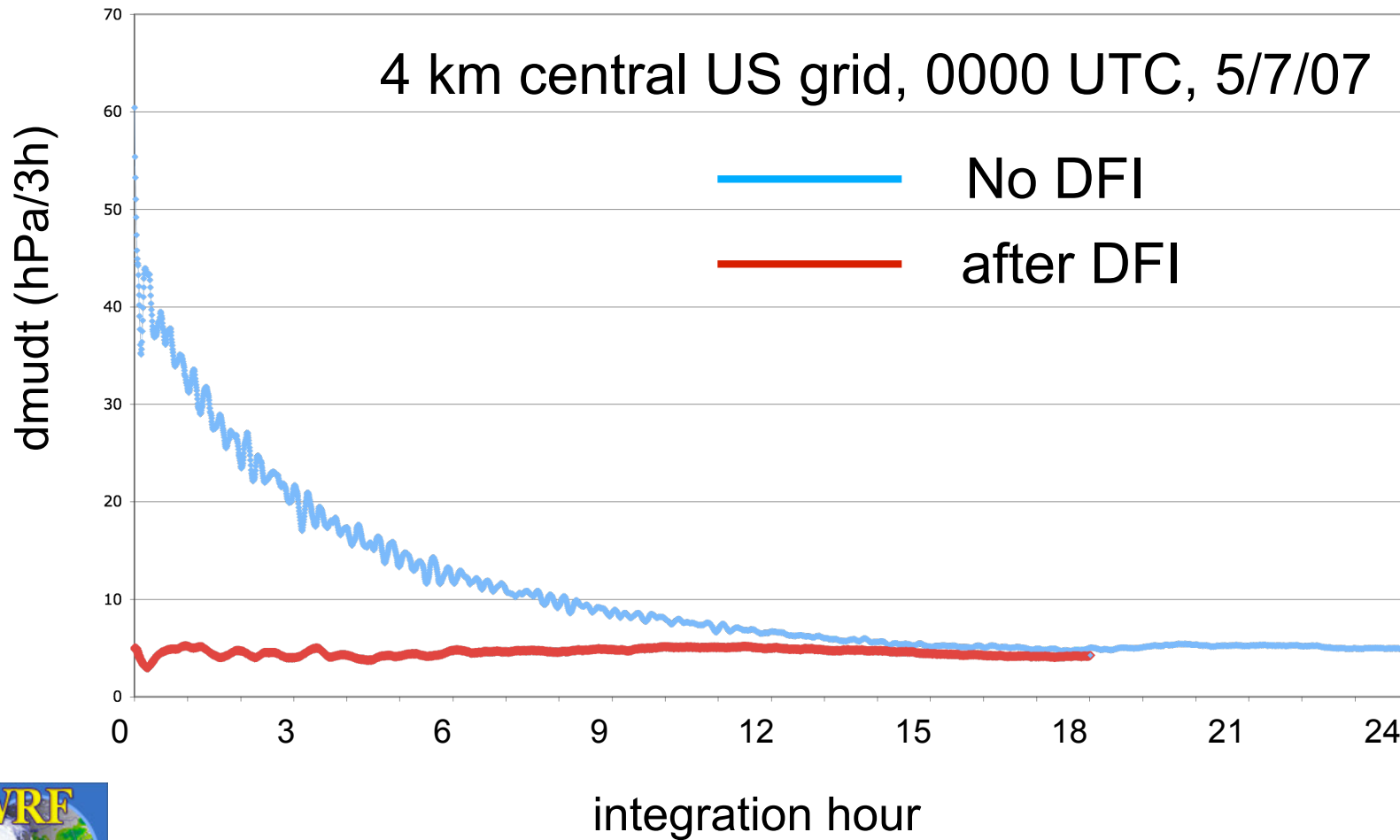
Digital filter initialization (2)

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

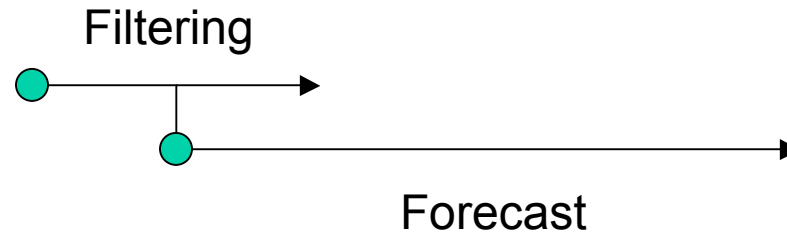DFI is done after program `real`, or data-assimilation step, just before model integration
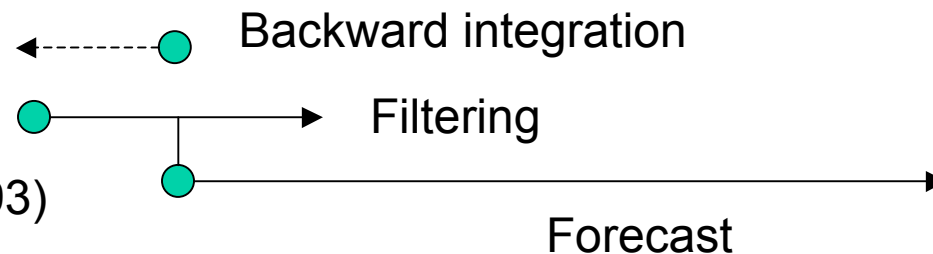
Digital filter initialization (3)

4 km central US grid, 0000 UTC, 5/7/07

No DFI
after DFI

dmudt (hPa/3h)

70
60
50
40
30
20
10
0

0   3   6   9   12   15   18   21   24

integration hour

Digital filter initialization (4)

DFL:
(Lynch and Huang, 1994)

Filtering

Forecast

DDFI:
(Huang and Lynch, 1993)

Backward integration

Filtering

Forecast

TDFI:
(Lynch and Huang, 1994)

Backward integration

Filtering

Forecast
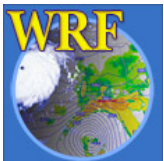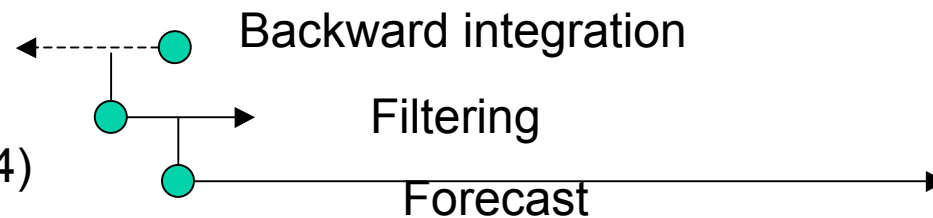
Digital filter inilialization (5)

Namelist control: **`&dfi`**

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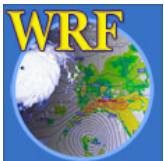- **`dfi_nfilter`**: filter options 0 - 8, recommended 7
- **`dfi_cutoff_seconds`** : cutoff period
- **`dfi_write_filtered_input`** : whether to write filtered IC
- **`dfi_bckstop_*`** : stop time for backward integration
- **`dfi_fwdstop_*`** : stop time for forward integration

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 the vortex can move near the parent domain boundary

`track_level:` e.g. 700 or 500 mb

`time_to_move:` hold nests still until this time

TC options

isftcflx: alternative C_d and C_k formulation for TC application

omlcall: simple ocean mixed layer

oml_hml0: initial ocean mixed layer depth

oml_gamma: lapse rate in deep water

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

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.

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