WRF/DART – Basic building blocks

Glen Romine

romine@ucar.edu

Outline

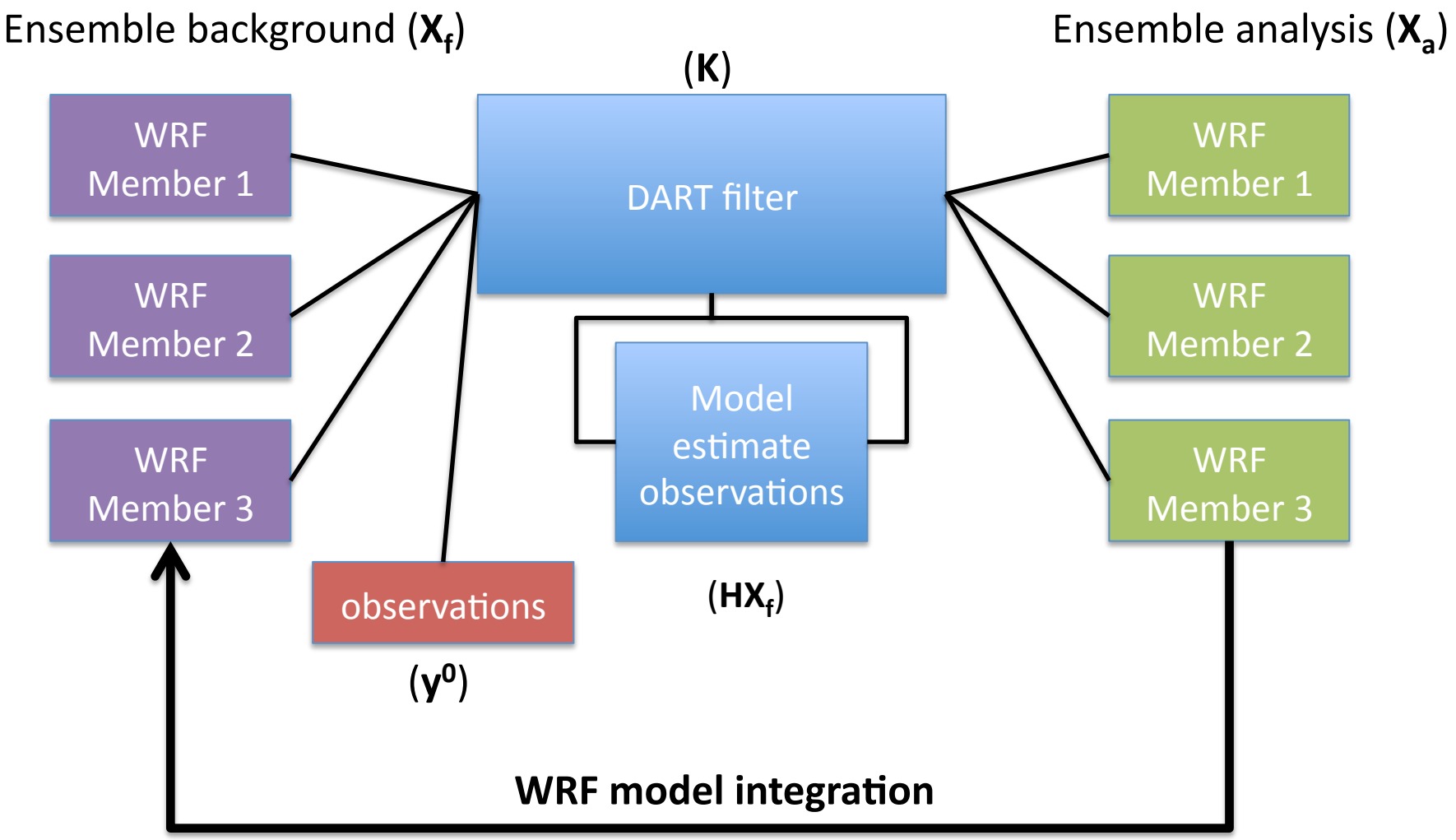
- Idealized experiment setup and components
  - Radar data assimilation
  - Single column model
- 'Real data' experiment setup and components
  - Full physics, terrain, requires lateral boundary conditions
- Observing system simulation experiments (OSSE)

Before we are ready to run filter….

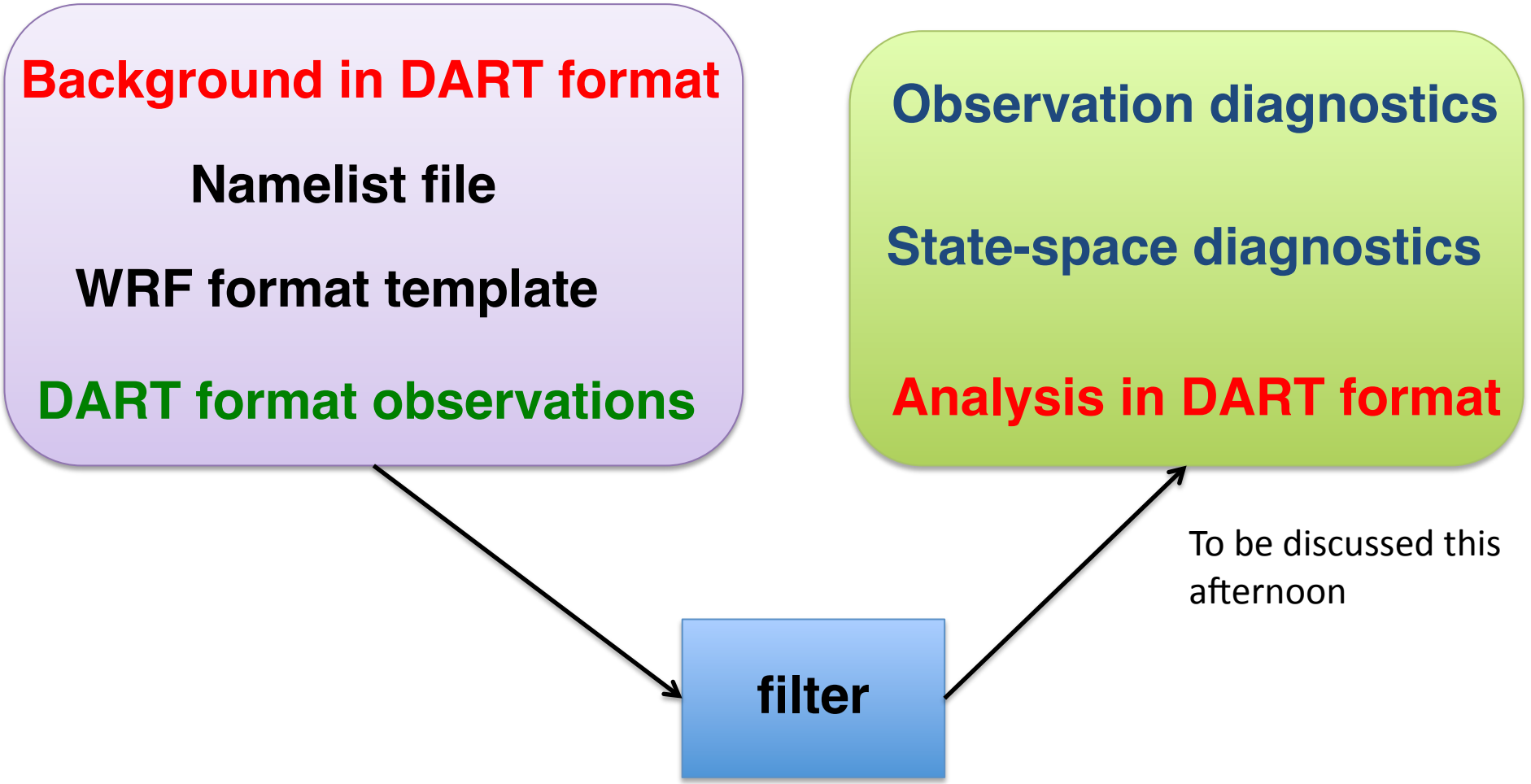
- Determine WRF model domain and configuration
  - May need to modify WRF:
    - Forward operators
    - May need to adjust Registry to cleanly restart model from history files
    - Surface ‘w’, lateral boundaries
  - Pre-processing (WPS, IDEAL or REAL)
- Obtain and process all observations
- Ensemble initial state (perturbations, integration)
- Lateral boundary files

Block diagram of ensemble cycled analysis

$$X_a = X_f + K[y^0 - HX_f]$$



Analysis step:
Practical Inputs and outputs of DART filter program for WRF



Create initial ensemble state

- Need an ensemble of varying initial states (40+)
  - IDEAL: e.g., build em_quarter_ss configuration (cloud model)
    - Create initial horizontally homogeneous states by perturbing the input sounding (temperature, horizontal winds, moisture)
    - Add 'noise' patch in the vicinity where storm will be triggered
    - Integrate ensemble state to allow perturbations to grow
    - Convert ensemble state to DART format

Prepare ideal background

Before the first analysis, need flow dependent background error covariance:

(1) IDEAL – perturbed soundings for horizontally homogeneous background states, local Gaussian noise in area where observations will be added, short integration to develop flow dependent structure

Unique base state and local perturbations added to each member

Scripts provided to generate initial state

Initial spread ~ obs error

N-S wind component

Prepare real background

Before the first analysis, need flow dependent background error covariance:

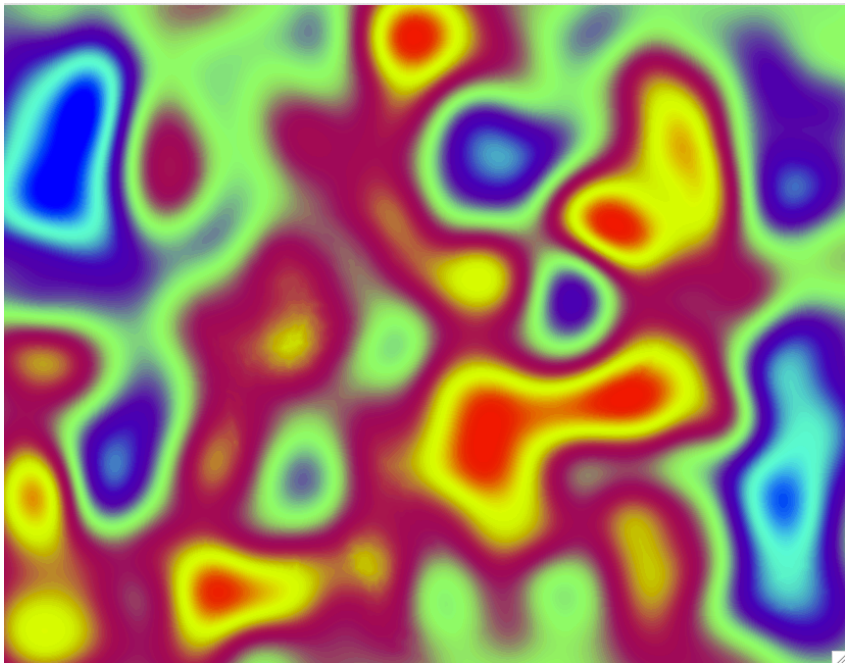
(2) REAL – initial state from external analysis (e.g., GDAS), add random perturbations (e.g., WRFDA random cv), integrate

Sample random CV perturbations: Drawn from climatological differences in 24 and 12 h forecasts

Same perturbation method often used for lateral boundary perturbations

namelist options control spatial scales and perturbation magnitude

Unique set added to mean state for each member



Prepare background

After WRF initial state is ready – must convert to DART format

DART converters wrf_to_dart, dart_to_wrf
Extract state variables from wrfinput file(s) and writes to DART format binary using wrf_to_dart, or vice versa (dart_to_wrf), repeat for each ensemble member

Recommendation: Use, or adapt from, provided scripts to prepare ensemble initial state:
Ideal radar example:

https://proxy.subversion.ucar.edu/DAReS/DART/trunk/models/wrf/experiments/Radar/IC/

Real data example:
https://proxy.subversion.ucar.edu/DAReS/DART/trunk/models/wrf/shell_scripts/init_ensemble_var.csh

DART Namelist file:

Input.nml – fortran namelist file, controls behavior of DART executables, information about the WRF state variables, observation treatment, etc…

WRF format template:

Wrfinput_d0? file(s), used for grid information, base state, static field updates – must match wrfinput files

Core namelist components:

filter_nml – async, outlier threshold, inflation, debugging flag, ensemble size, diagnostics output

https://proxy.subversion.ucar.edu/DAReS/DART/trunk/filter/filter.html

> *async* – more later

> *ens_size* – number of ensemble members, 40 or more recommended

> *outlier_threshold* – Bad obs occasionally get through QC. Control distance from observation to background state before observation is rejected. Value of 3 recommended.

> Inflation (*inf_flavor*) – helps avoid filter divergence (collapse of spread and model trajectory differs from observation trajectory). Recommend spatially and temporally adaptive inflation of the prior state (2,0).

filter_nml (cont.)

Inflation (cont.)
Also must manage adaptive inflation restart files (*inf_XX_file_name*)

Adaptive inflation settings from default:

inf_initial = 1.0
inf_sd_initial = 0.6
inf_damping = 0.9

> Debugging (*trace_execution*) – set true to enable helpful output when filter is not doing what you expect

> Diagnostics (*num_output_XX_members*) – set to ens_size or less, recommend ens_size for obs space, 1 for state space

> *sampling_error_correction* – optional flag, particularly useful for small ensembles, if used, copy in the appropriate sampling error table

assim_tools_nml – filter kind, cutoff

https://proxy.subversion.ucar.edu/DAReS/DART/trunk/assim_tools/assim_tools_mod.html

> *Filter_kind* – most WRF users use the ensemble adjustment Kalman filter (EAKF), option 1

> *Cutoff* – controls the localization distance, in radians. For typical WRF applications, convert horizontal distance as:

cutoff*40,000/(2xpi) = **half width** of GC in km

Ideal radar assimilation (0.002) ~ 12 km

Small ensembles with conventional observations (0.05) ~ 320 km

Large ensembles with conventional observations (0.12) ~ 765 km

model_nml – WRF state variables, bounds, cycling frequency

https://proxy.subversion.ucar.edu/DAReS/DART/trunk/models/wrf/model_mod.html

Default_state_variables = false (specify the variables you want to update)

wrf_state_variables = 'U', 'KIND_U_WIND_COMPONENT', 'TYPE_U', 'UPDATE','999',
'V', 'KIND_V_WIND_COMPONENT', 'TYPE_V', 'UPDATE','999',
'PH', 'KIND_GEOPOTENTIAL_HEIGHT', 'TYPE_GZ','UPDATE','999',

List all the state variables that you want updated, as well as any diagnostic fields needed for forward operators that are in WRF state (e.g. radar reflectivity)

wrf_state_bounds – list variables that need to have restricted range upon returning to WRF, e.g. positive definite fields like mixing ratios

Cycling frequency (*assimilation_period_seconds*), e.g. 21600 (6 h)

Vert_localization_coord – controls coordinate of vertical localization

obs_kind_nml – Assimilate/evaluate these obs

https://proxy.subversion.ucar.edu/DAReS/DART/trunk/obs_kind/obs_kind_mod.html

assimilate_these_obs_types = 'RADIOSONDE_TEMPERATURE',

List the observation types you want to assimilate. Similar list of 'evaluate', which computes the observation space diagnostics, but these do not impact the model state

location_nml – vertical localization

https://proxy.subversion.ucar.edu/DAReS/DART/trunk/location/threed_sphere/location_mod.html

horiz_dist_only - set to false to allow for vertical localization

vert_normalization_height - value in meters that will be the cutoff distance on the vertical (multiply by the cuttoff). Similar namelist options for other coordinate systems. For cutoff of 0.12, setting of 80,000. gives 9.6 km half width cutoff in vertical

Specialty namelists:

obs_def_radar_mod_nml (only for radar observation assimilation)

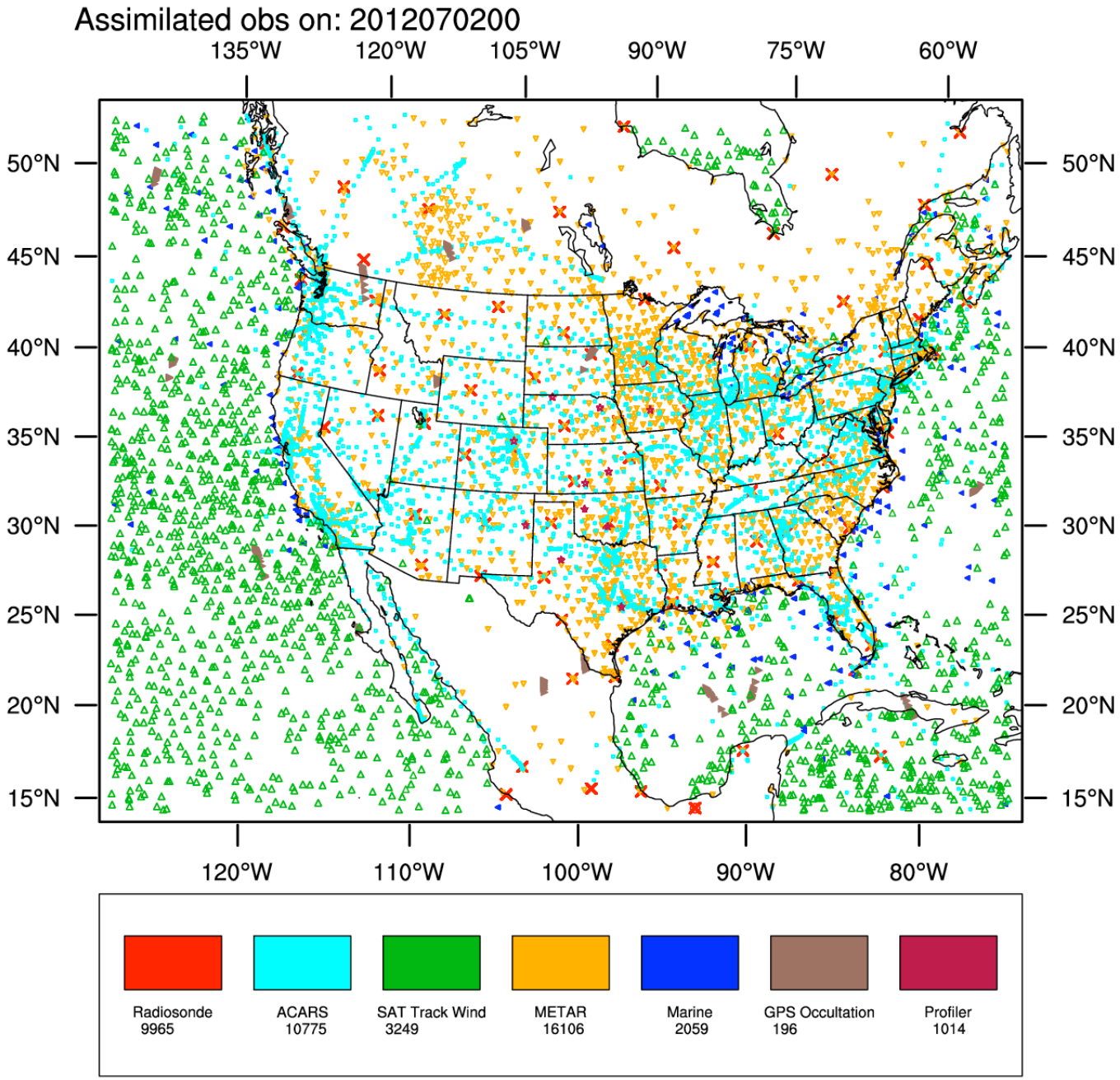
diagnostics namelists (more on diagnostics later)

restart file tool (needed when filter is no longer handling model advances)

replace_wrf_fields (used to update static fields in WRF member states)

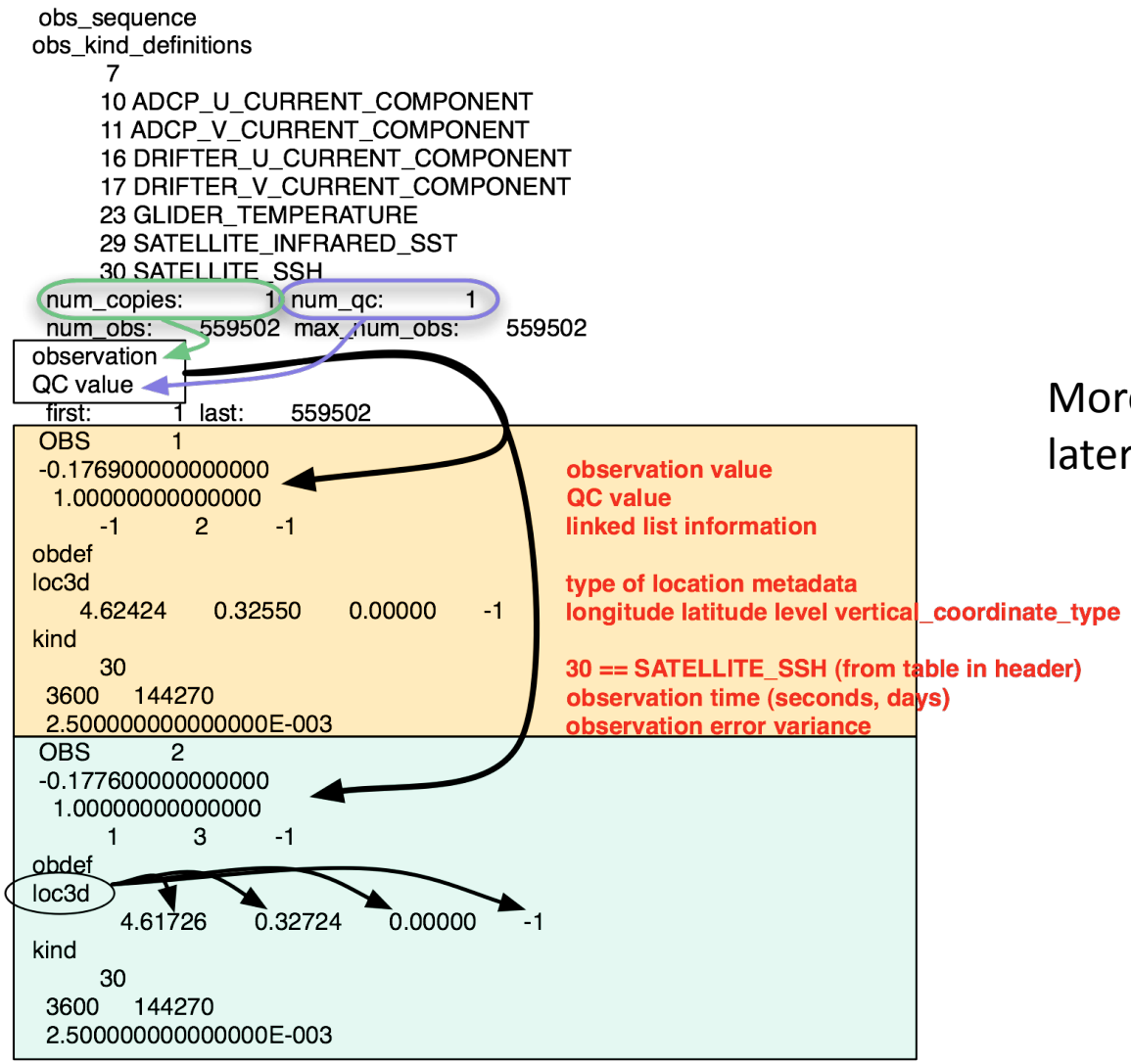
observation processing namelists (more on observation processing later)

Inputs - OBS


Assimilated obs on: 2012070200
135°W
120°W
105°W
90°W
75°W
60°W
50°N
45°N
40°N
35°N
30°N
25°N
20°N
15°N
120°W
110°W
100°W
90°W
80°W
Radiosonde 9965
ACARS 10775
SAT Track Wind 3249
METAR 16106
Marine 2059
GPS Occultation 196
Profiler 1014

DART format observations

Observation sequence files – text format, provided utilities for conversion and processing



More on conversion programs later today

Within the directory you will run filter, you need:

- Wrfinput_d0$ file(s) – template for grid information, static state
- Executables (filter, + support files as needed)
  - Dart_to_wrf, wrf_to_dart, advance_time, update_wrf_bc, etc…
- Namelist file (input.nml, namelist.input)
- Advance_model.csh, or equivalent
- WRF executables and support files (in directory WRF_RUN)
- advance_temp* directories containing wrfinput files
- Linked directory 'WRF' where boundary condition files are kept
- Support files (bc_pert_scale, final_full.*)

Example run directory:

Directory

DART executables from $DART/models/wrf/work/ directory

Shell script from $DART/models/wrf/shell_scripts

```
add_pert_where_high_refl  advance_temp21  advance_temp35  advance_temp49  namelist.input
advance_model.csh         advance_temp22  advance_temp36  advance_temp5   obs_diag
advance_temp1             advance_temp23  advance_temp37  advance_temp50  obs_sequence_tool
advance_temp10            advance_temp24  advance_temp38  advance_temp6   pert_wrf_bc
advance_temp11            advance_temp25  advance_temp39  advance_temp7   replace_wrf_fields
advance_temp12            advance_temp26  advance_temp4   advance_temp8   restart_file_tool
advance_temp13            advance_temp27  advance_temp40  advance_temp9   update_wrf_bc
advance_temp14            advance_temp28  advance_temp41  advance_time    wakeup_filter
advance_temp15            advance_temp29  advance_temp42  bc_pert_scale   WRF
advance_temp16            advance_temp3   advance_temp43  dart_to_wrf     wrf_dart_obs_preprocess
advance_temp17            advance_temp30  advance_temp44  ensemble_init   WRF_RUN
advance_temp18            advance_temp31  advance_temp45  filter          wrf_to_dart
advance_temp19            advance_temp32  advance_temp46  final_full.50
advance_temp2             advance_temp33  advance_temp47  grid_refl_obs
advance_temp20            advance_temp34  advance_temp48  input.nml
```

WRF_RUN directory:

```
aerosol.formatted        da_wrfvar.exe      nup.exe                RRTMG_LW_DATA      tr67t85
aerosol_lat.formatted    ETAMPNEW_DATA      ozone.formatted        RRTMG_LW_DATA_DBL  URBPARM.TBL
aerosol_lon.formatted    ETAMPNEW_DATA_DBL  ozone_lat.formatted    RRTMG_SW_DATA      VEGPARM.TBL
aerosol_plev.formatted   GENPARM.TBL        ozone_plev.formatted   RRTMG_SW_DATA_DBL  wrf.exe
be.dat                   grib2map.tbl       real.exe               SOILPARM.TBL
CAM_ABS_DATA             gribmap.txt        real.serial.exe        tc.exe
CAM_AEROPT_DATA          LANDUSE.TBL        RRTM_DATA              tr49t67
co2_trans                ndown.exe          RRTM_DATA_DBL          tr49t85
```

Everything except WRF namelist

Advance_model.csh

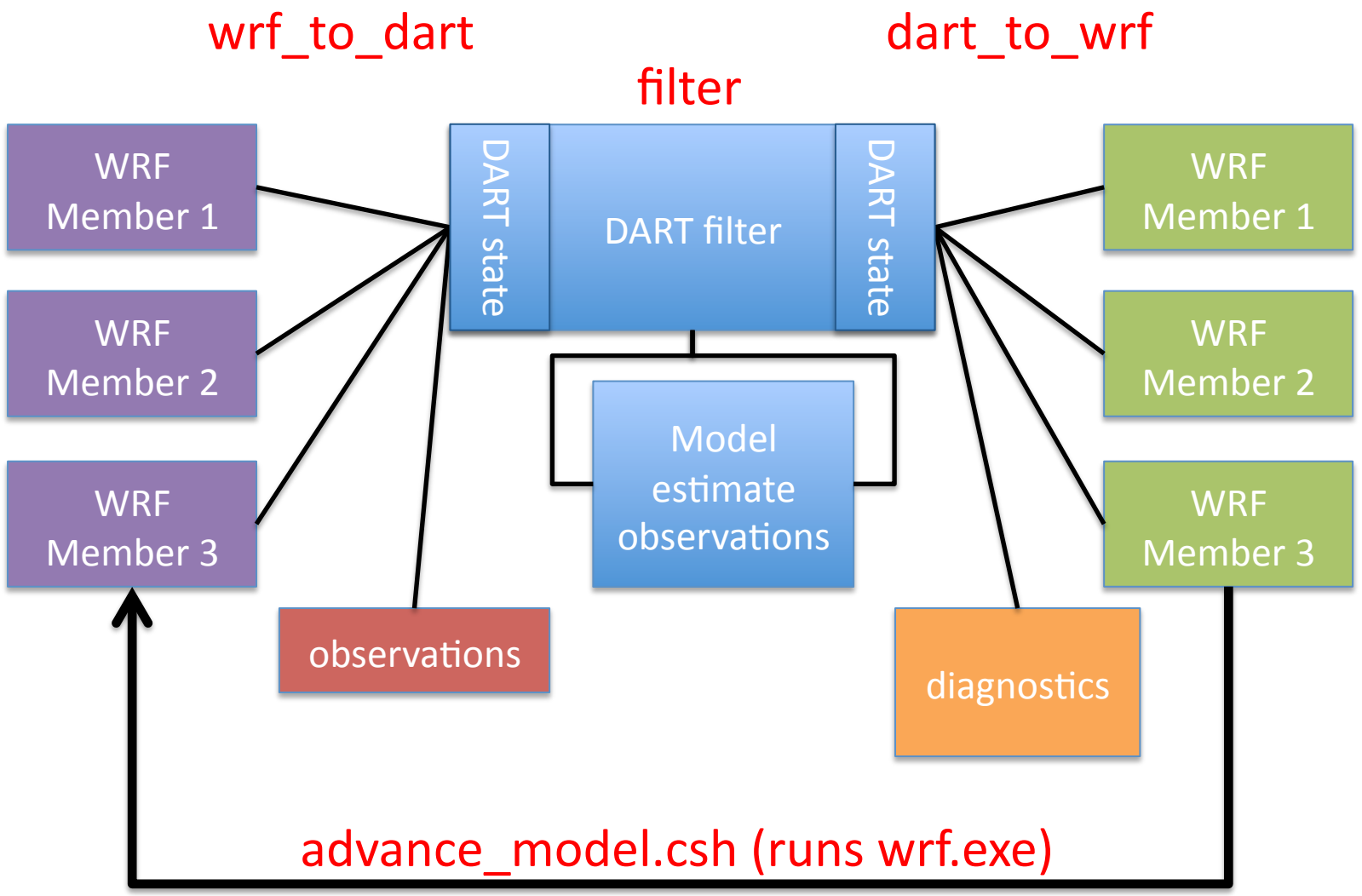
Shell script that *can* be called by filter, designed to complete a model integration of WRF to the next assimilation window. Do not recommend edits beyond setting select flags at the top – conform environment to fit the script to avoid dependency issues. See dependency list at top of script.

https://proxy.subversion.ucar.edu/DAReS/DART/trunk/models/wrf/shell_scripts/advance_model.html

Key settings:
Individual_members = true (allows fields such as soil state to be carried forward)
Perturbed Lateral boundary conditions, determined by executables found in work directory: ./WRF_RUN/da_wrfvar.exe; be.dat; pert_wrf_bc vs. update_wrf_bc

Pre-generated, on-the-fly (N/A for idealized configurations of WRF)

Block diagram of ensemble cycled analysis – key programs



Modes of cycled analysis system operation:

DART is an observation driven system

Most systems won't allow MPI tasks to call other MPI tasks

Typical async modes:

(a) Single observation sequence spanning many analysis times

2 – serial model executable, MPI filter

4 – MPI model, but ensemble forecasts are completed in serial order, MPI filter

(b) Individual observation sequence file for each analysis time

5 – unofficial, but most commonly used mode on big computers: MPI model, simultaneous model advances, MPI filter requires external, environment specific scripting to run tasks

http://www.image.ucar.edu/DAReS/DART/filter_async_modes.html

Observing System Simulation Experiments (OSSEs)

OSSEs, or perfect model experiments

Works the same as above, except the observations are drawn from a forecast model. Thus, these observations have the potential to be 'perfect', as in no error from the forward operator, no bias, etc.

https://proxy.subversion.ucar.edu/DAReS/DART/trunk/perfect_model_obs/perfect_model_obs.html

perfect_model_obs – define observing network ob types and locations, perturbations are drawn from specified observation error variance and added to drawn observations.

Can also be used for imperfect model experiments

Useful for testing new forward operators, observing networks

Questions?

dart@ucar.edu
romine@ucar.edu