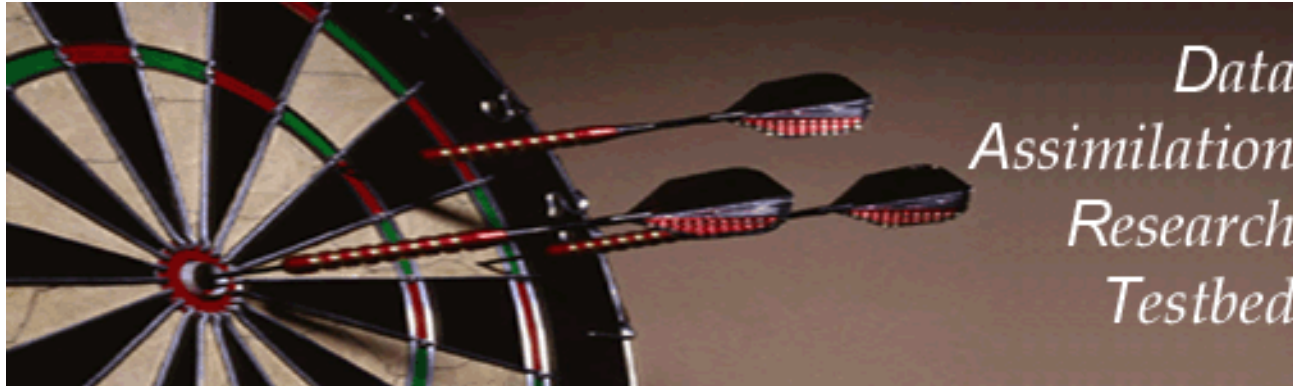


Data Assimilation Research Testbed Tutorial

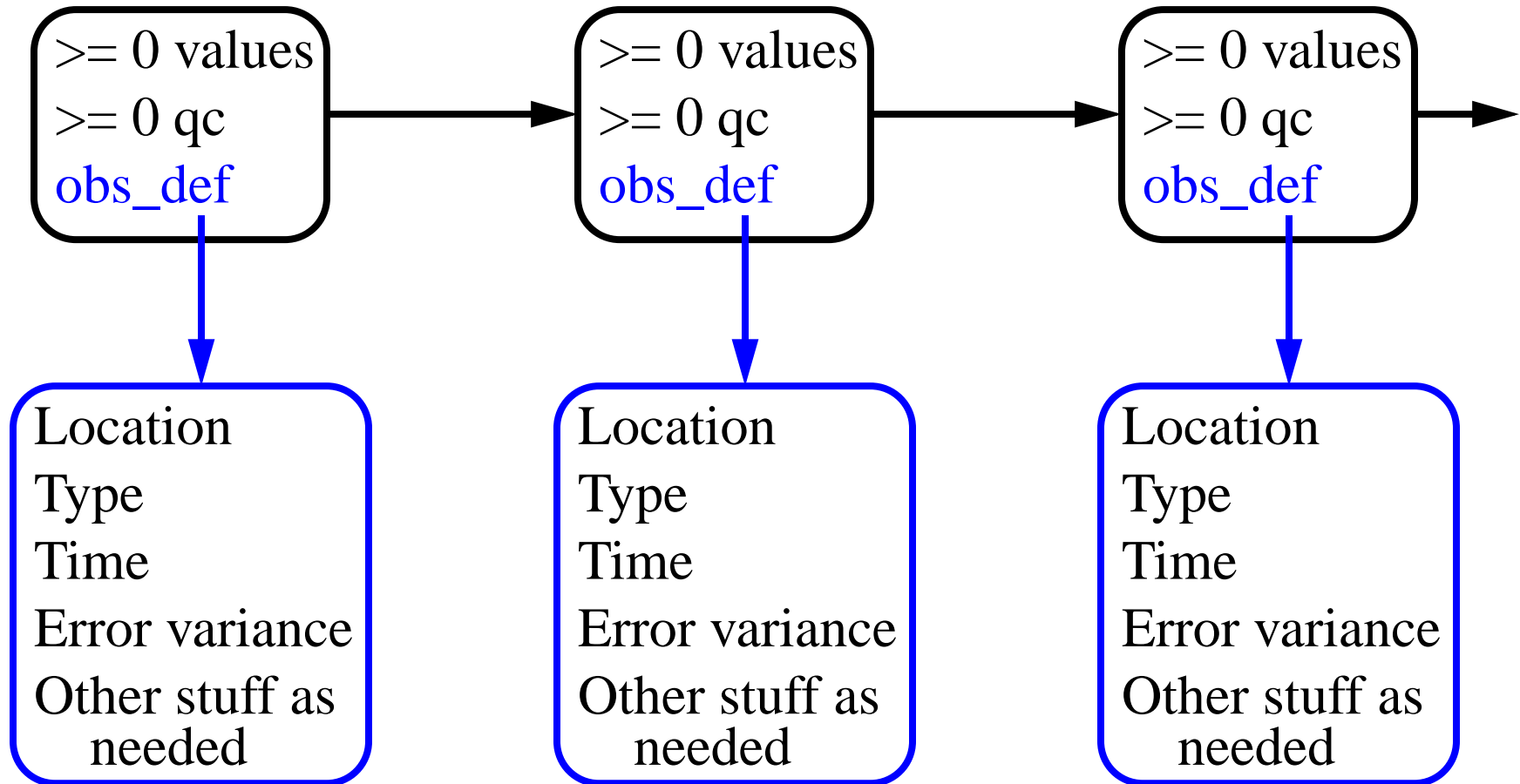


Section 17: Creating Observation Sequences

Version 2.0: September, 2006

Structure of an obs_sequence file:

Sequence with non-decreasing times in definitions.



Building Real observation sequences:

1. Interactive direct construction: program *create_obs_sequence*.

Queries for information for each observation in turn.

Enter type, location, time, error variance, value, qc value(s).

Often convenient to create an input file.

Then redirect this file to standard input for *create_obs_sequence*.

2. Creating your own program.

The *obs_sequence* module provides full set of interfaces to create.

Example: translation from NCEP BUFR file format.

Reads BUFR files, writes *obs_sequence*.

Creating Synthetic Observation Sequences (OSSEs):

Step 1: Create an observation sequence with no values.

A. Direct use of *create_obs_sequence*: no need to specify value for obs.

OR...

B. Synthetic observing network fixed in time:

1. First, use *create_obs_sequence* to specify observations in fixed network, all with time 0 days, 0 seconds.
2. Use *create_fixed_network_seq* to specify times at which fixed network is observed.
3. Times can be regularly or irregularly spaced.

Creating Synthetic Observation Sequences (OSSEs):

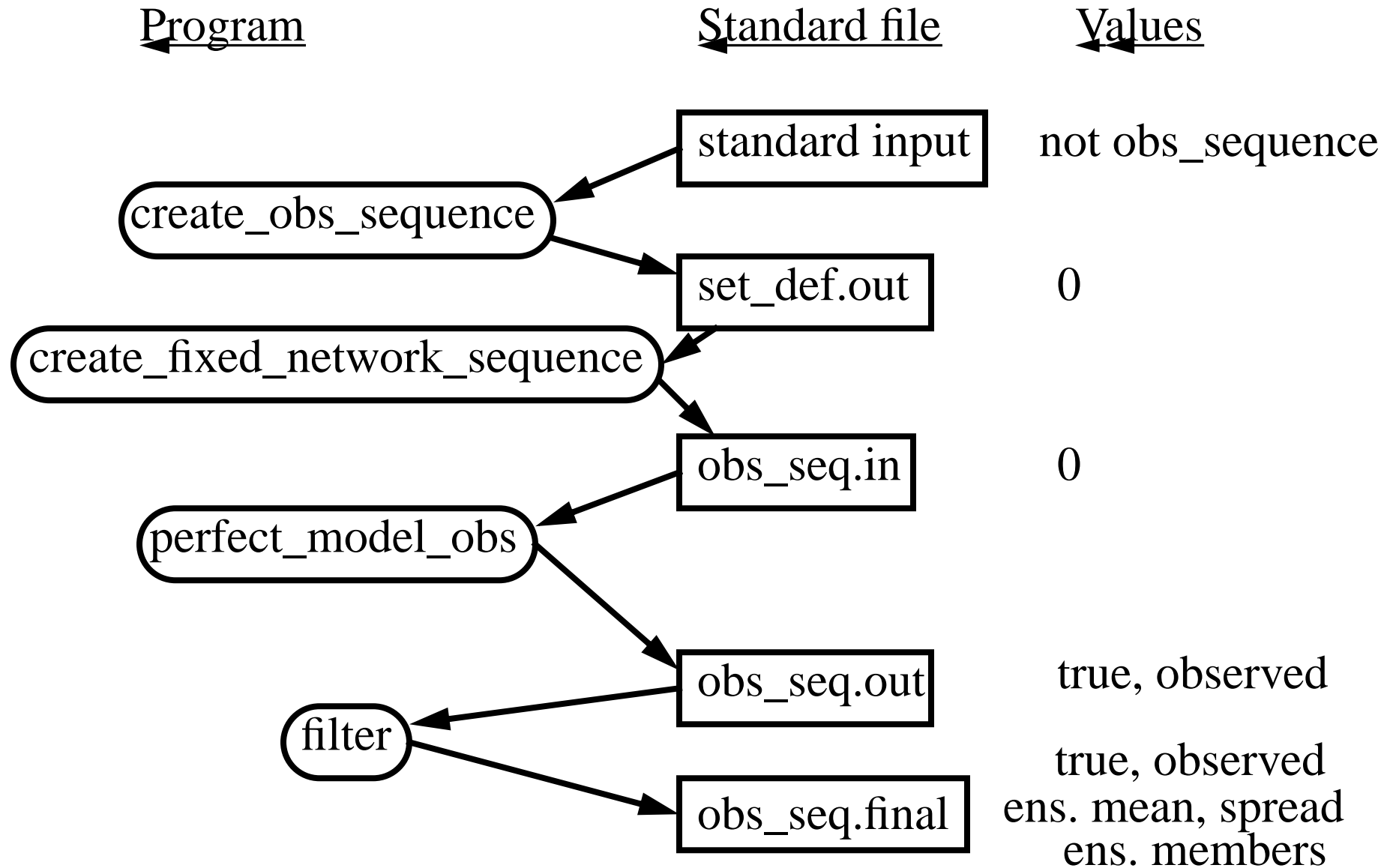
Step 2: Use perfect model_obs to add observed values:

1. Integrates model.
2. Applies forward operators to get ‘true’ observed values.
3. Adds sample from observational error to get observed value.
(Output obs_sequence has 2 values for each observation).

Step 3: Run the filter:

Ensemble mean, spread, and individual ensemble members are added as values if requested (*filter_nml*, see section 16).

Creating Synthetic Observation Sequences (OSSEs):



Example: Localized observation set for Lorenz-96

1. Run *create_obs_sequence*

Select 5 observations as upper bound

Select 0 copies of data (we'll let perfect_model_obs fill these in)

Also 0 quality control fields

Never input a -1 to terminate (we'll do all 5)

Select raw state variable for each observation

Pick a location (try grouping them close to 0.5)

Select time as 0 days, 0 seconds

Error variance of 1.0 for first try

Repeat for all 5 observations just varying location

Example: Localized observation set for Lorenz-96 (cont.)

2. Run `create_fixed_network_seq` to observe these 5 obs. repeatedly
 - Select a regularly repeating sequence
 - Select 1000 times
 - Initial time as 0 days, 0 seconds
 - Observation period as 1 hour (0 days, 3600 seconds)
 - Resulting `obs_seq.in` observes once an hour for 1000 hours
3. Run *perfect_model_obs* to generate synthetic observations (OSSE)
4. Run *filter* with some adaptive inflation, and 80 members in 4 groups

Use matlab diagnostics to examine results

Try `plot_ens_time_series`

Select a variable close to the observations and one far away.

Designing localized observation set for bgrid model

1. Run *create_obs_sequence*

Enter only 1 observation, 0 values and qc fields

Select Radiosonde temperature

Vertical coordinate Pressure, 500 hPa

Try longitude and latitude 270, 45

Time is 0 days, 0 seconds

Error variance is 1.0

2. Run *create_fixed_network_sequence*

Select regularly repeating, 2 times

0 days 0 seconds for initial time, 0 days 3600 seconds for period

3. Run *perfect_model_obs*

4. Run *filter*

5. Create innovations: *ncdiff Prior_Diag.nc Posterior_Diag.nc Innov.nc*

6. Use *ncview* to look at copies 1 (mean) and 2 (spread) for fields

Interesting to see how T observation impacts u, v, surface pressure

Selecting set of observation definitions:

Need to specify via name (character string) in namelist:

- Type of all observations to be assimilated;

- Type of all observations to be evaluated but not assimilated.

(Forward operators are computed and stored in `obs_sequence`).

List of available observation types found in `obs_kind_mod.f90`.

(see declaration for `obs_kind_info`).

Specify in `obs_kind_nml` using names:

- `&obs_kind_nml`

 - `assimilate_these_obs_types = 'RAW_STATE_VARIABLE'`

 - `evaluate_these_obs_types = 'RAW_STATE_1D_INTEGRAL'/`

Selecting set of observation definitions:

Have to preprocess using program *preprocess*.

DEFAULT_obs_def_mod.F90 and DEFAULT_obs_kind_mod.F90 are merged with additional special observation definition files to create obs_def_mod.f90 and obs_kind_mod.f90

See section 21 for more details.