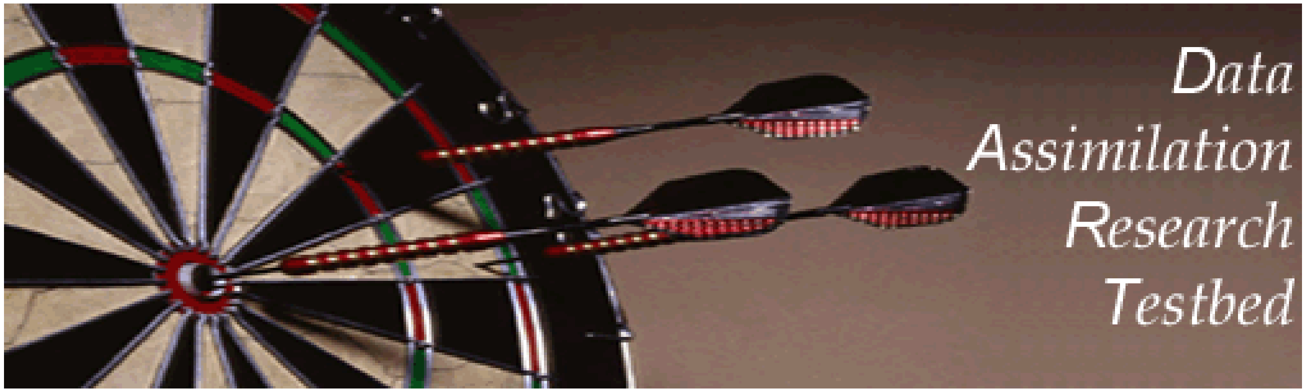
Data Assimilation Research Testbed Tutorial



Section 20: Model Parameter Estimation

Version 1.0: June, 2005

Suppose model is governed by (stochastic) Difference Equation:

$$dx_t = f(x_t, t;u) + G(x_t, t;w)d\beta_t, \quad t \geq 0 \qquad (1)$$

where u and w are vectors of parameters.

Also, suppose we really don’t know values of parameters (very well).

Can use observations with assimilation to help constrain these values.

Rewrite (1) as:

$$dx_t^A = f^A(x_t^A, t) + G^A(x_t^A, t)d\beta_t, \quad t \geq 0 \qquad (2)$$

where the augmented state vector includes x_t, u, and w.

Model is modified so values of u and w can be changed by assimilation.

Model might also introduce some time tendency for u and w.

From ensemble filter perspective:

Just add any parameters of interest to the model state vector;
Proceed to assimilate as before.

Possible difficulties:

1. Where are parameters ‘located’ for localization?
2. Parameters won’t have any error growth in time
   (unless we add some): could lead to filter divergence.
3. Parameters may not be strongly correlated with any observations.

Testing Parameter Estimation in DART:

DART includes *models/forced_lorenz_96* directory.

Each state variable has corresponding forcing variable, F_i.

$$dX_i / dt = (X_{i+1} - X_{i-2})X_{i-1} - X_i + F_i \qquad (3)$$

$$dF_i/dt = N(0, \sigma_{noise}) \qquad (4)$$

Can observations of some function of state variables constrain F?

Additional namelist control aspects required for experimentation:

1. *reset_forcing*,
   If true, F_i = *forcing* (also from namelist) for all i, t.
2. *random_forcing_amplitude*
   σ_{noise} for F_i time tendency,
   not used if reset_forcing is true.

Using these, can create OSSE sets with fixed, global F value.

Assimilate these with filter, estimate state and forcing.

Get an ensemble sample of F_i at each time.

Random noise can be useful for avoiding filter divergence.

Assimilation in the forced Lorenz-96 model:

cd *models/forced_lorenz_96/work.*
Execute *csh workshop_setup.csh.*

Use matlab, etc. to examine output.

Same 40 randomly located observations as in lorenz_96 cases.
Forcing was fixed at 8.0 in the perfect_model run.
Values of Fi are modified in assimilation.
There was some noise (amplitude 0.1) added to the time tendency.

Amazing Fact: Best assimilations of state come when Fi varies,
even better than when Fi is set to exact value, 8!

Contest: Given an observation set, what was the value of F?

You are given an observation sequence file, *obs_seq.out.CONTEST*.

Copy this to *obs_seq.out* and run the *filter*.

Question: What was the value of the forcing in the perfect_model run?

You can try anything (ethical) you want.

Feel free to ask for help to try experiments you don’t know how to do.

Consistent with the theme of the workshop...

In event of tie, random number generator will be used.

Honor, fame, and fabulous prizes go to the winning team!!!