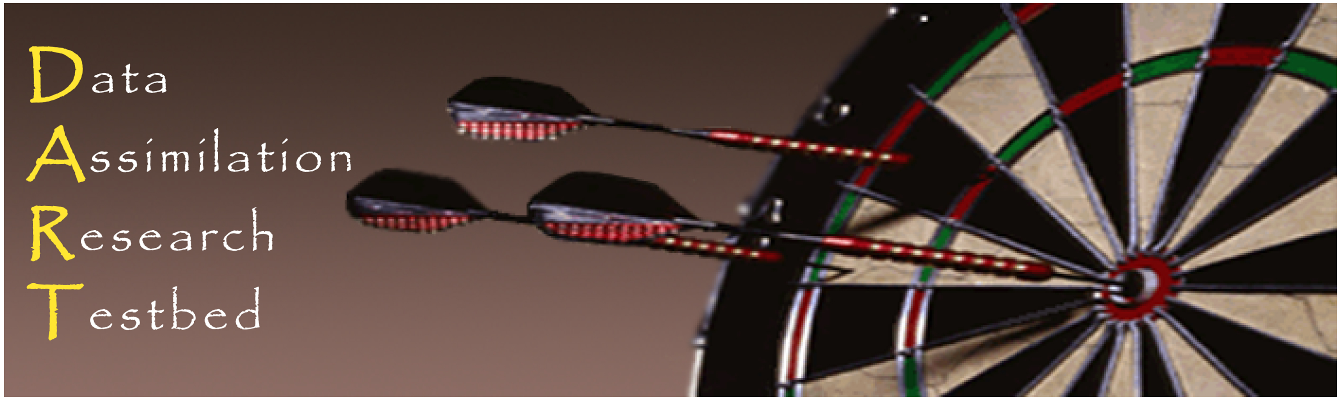
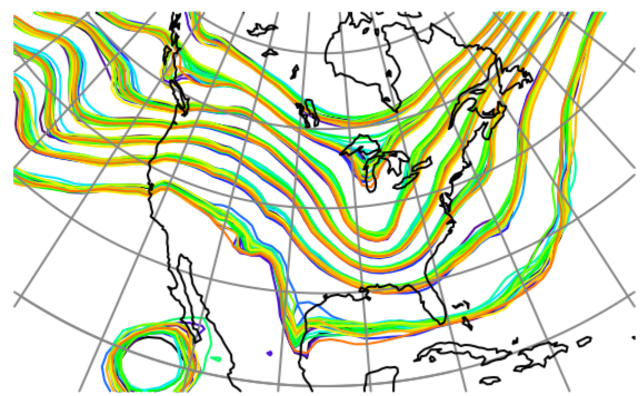


DART Tutorial Section 20: Model Parameter Estimation



The National Center for Atmospheric Research is sponsored by the National Science Foundation.
Any opinions, findings and conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

NCAR | National Center for
UCAR | Atmospheric Research

Model Parameter Estimation

Suppose a model is governed by a (stochastic) Difference Equation:

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

where u and w are vectors of parameters. Also, suppose we really don't know the parameter values (very well). **Can we use observations with assimilation to help constrain these values?**

Rewrite (1) as:

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

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

The model is modified so values of u and w can be changed by assimilation. The model might also introduce some time tendency for u and w.

Model Parameter Estimation

From the 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 a ***models/forced_lorenz_96*** directory.

Each state variable has a corresponding forcing variable, F_i.

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

Observational errors for obs. in set i independent of those in set j.

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

Can observations of some function of state variables constrain F?

Adding 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.

Adding namelist control aspects required for experimentation:

models/forced_lorenz_96/work/

```
&model_nml
   num_state_vars     = 40
   forcing            = 8.0
   delta_t            = 0.05
   time_step_days     = 0
   time_step_seconds  = 3600
   reset_forcing      = .false.
   random_forcing_amplitude = 0.10
   /
```

If `reset_forcing = .true.`, F_i = *forcing* (also from namelist) for all i,t.

σ_{noise} for F_i time tendency, not used if `reset_forcing = .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
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 F_i are modified in the assimilation.
There was some noise (amplitude of 0.1) added to the time tendency.

Amazing Fact: *Best assimilations of state come when* F_i *varies, even better than when* F_i *is set to exact known value of 8.0!*

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

In *models/forced_lorenz_96/work* edit *input.nml*

```
&filter_nml
   …
   obs_sequence_in_name = "obs_seq.out"
```

Change to
"obs_seq.out.CONTEST"

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.

Remember: The Truth is NO LONGER KNOWN!

Consistent with the theme of the workshop … in the event of a tie, a random number generator will be used to decide the winner.

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

DART Tutorial Index to Sections

1. Filtering For a One Variable System
2. The DART Directory Tree
3. DART Runtime Control and Documentation
4. How should observations of a state variable impact an unobserved state variable? Multivariate assimilation.
5. Comprehensive Filtering Theory: Non-Identity Observations and the Joint Phase Space
6. Other Updates for An Observed Variable
7. Some Additional Low-Order Models
8. Dealing with Sampling Error
9. More on Dealing with Error; Inflation
10. Regression and Nonlinear Effects
11. Creating DART Executables
12. Adaptive Inflation
13. Hierarchical Group Filters and Localization
14. Quality Control
15. DART Experiments: Control and Design
16. Diagnostic Output
17. Creating Observation Sequences
18. Lost in Phase Space: The Challenge of Not Knowing the Truth
19. DART-Compliant Models and Making Models Compliant
20. Model Parameter Estimation
21. **Observation Types and Observing System Design**
22. **Parallel Algorithm Implementation**
23. Location module design (not available)
24. Fixed lag smoother (not available)
25. **A simple 1D advection model: Tracer Data Assimilation**