Ensemble Filter Data Assimilation

Jeffrey L. Anderson
NCAR Data Assimilation Initiative
17 June, 2004

I. Overview and some methods
   Big problems require clever simplification

II. Challenges
   A. Model bias
   B. Balances and attractors
   C. Assimilation and discrete distributions

III. The Data Assimilation Research Testbed
   A. A flexible, powerful, easy-to-use data assimilation facility
   B. Results from a realistic application: CAM 2.0 with real data
   C. Multivariate methods and (un)observed tracer assimilation

The Data Assimilation Problem

Given:

1. A physical system (atmosphere, ocean...)

---

2. Observations of the physical system

   Usually sparse and irregular in time and space
   Instruments have error of which we have a (poor) estimate
   Observations may be of ‘non-state’ quantities
   Many observations may have very low information content

---

3. A model of the physical system

   Usually thought of as approximating time evolution
   Could also be just a model of balance (attractor) relations
   Truncated representation of ‘continuous’ physical system
   Often quasi-regular discretization in space and/or time
   Generally characterized by ‘large’ systematic errors
   May be ergodic with some sort of ‘attractor’

We want to increase our information about all three pieces:

---

1. Get an improved estimate of state of physical system

Includes time evolution and ‘balances’
Initial conditions for forecasts
High quality analyses (re-analyses)

---

2. Get better estimates of observing system error characteristics

Estimate value of existing observations
Design observing systems that provide increased information

---

3. Improve model of physical system

Evaluate model systematic errors
Select appropriate values for model parameters
Evaluate relative characteristics of different models

Examples:

1. Numerical Weather Prediction

Model: Global troposphere / stratosphere O(1 degree by 50 levels)

Observations: radiosondes twice daily, surface observations, satellite winds, aircraft reports, satellite radiances, etc.

2. Tropical Upper Ocean State Estimation (ENSO prediction)

Model: Global (or Pacific Basin) Ocean O(1 degree by 50 levels)

Observations: Surface winds (possibly from atmospheric assimilation), TAO buoys, XBTs, satellite sea surface altimetry

3. Mesoscale simulation and prediction

Model: Regional mesoscale model (WRF), O(1km resolution)

Observations: Radial velocity from Doppler radar returns

4. Global Carbon Sources and Sinks

Nonlinear Filtering (A Bayesian Perspective)

Dynamical system governed by (stochastic) DE:

$$dx_t = f(x_t, t) + G(x_t, t)d\beta_t, \qquad t \geq 0 \tag{1}$$

Observations at discrete times:

$$y_k = h(x_k, t_k) + v_k; \;\; k = 1, 2, \ldots; \qquad t_{k+1} > t_k \geq t_0 \tag{2}$$

Observational error is white in time and Gaussian (nice, not essential)

$$v_k \rightarrow N(0, R_k) \tag{3}$$

Complete history of observations is:

$$Y_\tau = \{y_l; \;\; t_l \leq \tau\} \tag{4}$$

Goal: Find probability distribution for state at time t:

$$p(x, t | Y_t) \tag{5}$$

Nonlinear Filtering (cont.)

State between observation times obtained from DE

Need to update state given new observation:

$$p\left(x, t_k \middle| Y_{t_k}\right) = p\left(x, t_k \middle| y_k, Y_{t_{k-1}}\right) \tag{6}$$

Apply Bayes' rule:

$$p\left(x, t_k \middle| Y_{t_k}\right) = \frac{p\left(y_k \middle| x_k, Y_{t_{k-1}}\right) p\left(x, t_k \middle| Y_{t_{k-1}}\right)}{p\left(y_k \middle| Y_{t_{k-1}}\right)} \tag{7}$$

Noise is white in time (3) so:

$$p\left(y_k \middle| x_k, Y_{t_{k-1}}\right) = p(y_k \middle| x_k) \tag{8}$$

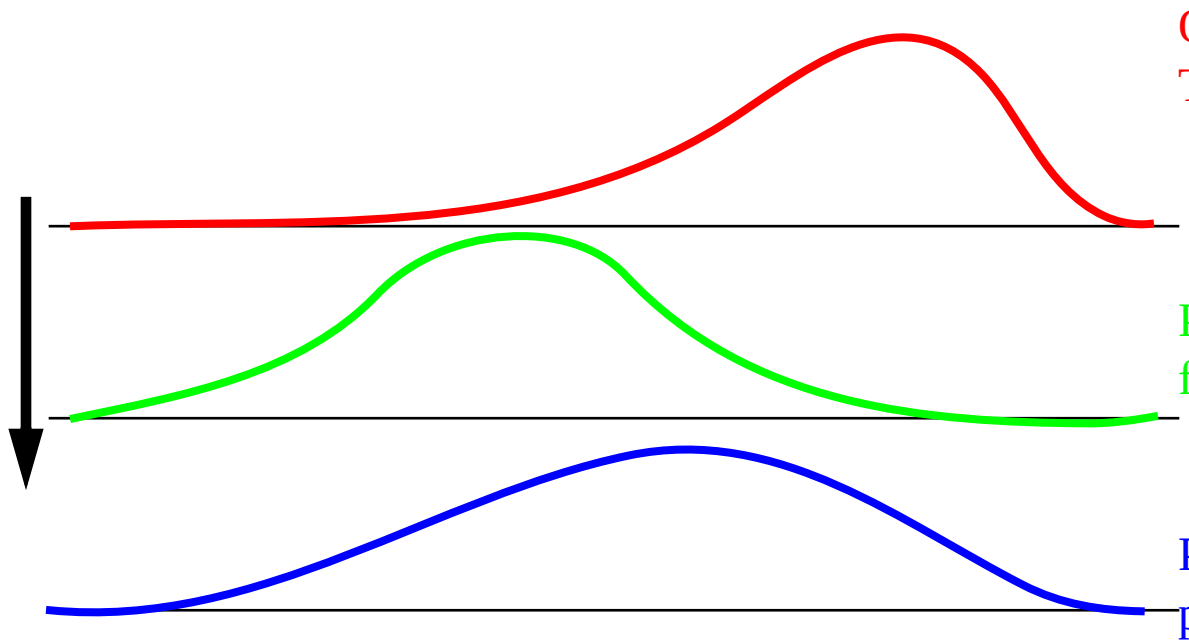
Also have:

$$p\left(y_k \middle| Y_{t_{k-1}}\right) = \int p(y_k \middle| x) p\left(x, t_k \middle| Y_{t_{k-1}}\right) dx \tag{9}$$

Nonlinear Filtering (cont.)

Probability after new observation:

$$p\left(x, t_k \middle| Y_{t_k}\right) = \frac{p(y_k|x)\, p\left(x, t_k \middle| Y_{t_{k-1}}\right)}{\int p(y_k|\xi)\, p\left(\xi, t_k \middle| Y_{t_{k-1}}\right) d\xi} \qquad (10)$$



Observational Error Term

Prior State Estimate, from Model

Posterior (normalized product)

General methods for solving the filter equations are known:

1. Advancing state estimate in time

2. Taking product of two distributions

But, these methods are far too expensive for problems of interest

1. Huge model state spaces (10 is big!), NWP models at O(10 million)

2. Need truncated representations of probabilistic state to avoid exponential solution time and storage

The ART of Data Assimilation:

Find heuristic simplifications that make approximate solution affordable

1. Localization (spatial or other truncated basis)

2. Linearization of models, represent time evolution as linear
   (around a control non-linear trajectory)

3. Represent distributions as Gaussian (or sum of Gaussians)

4. Monte Carlo methods

5. Application of simple balance relations

6. Many others...

Kalman Filter

Simplifications:

1. Linearization of model around non-linear control trajectory

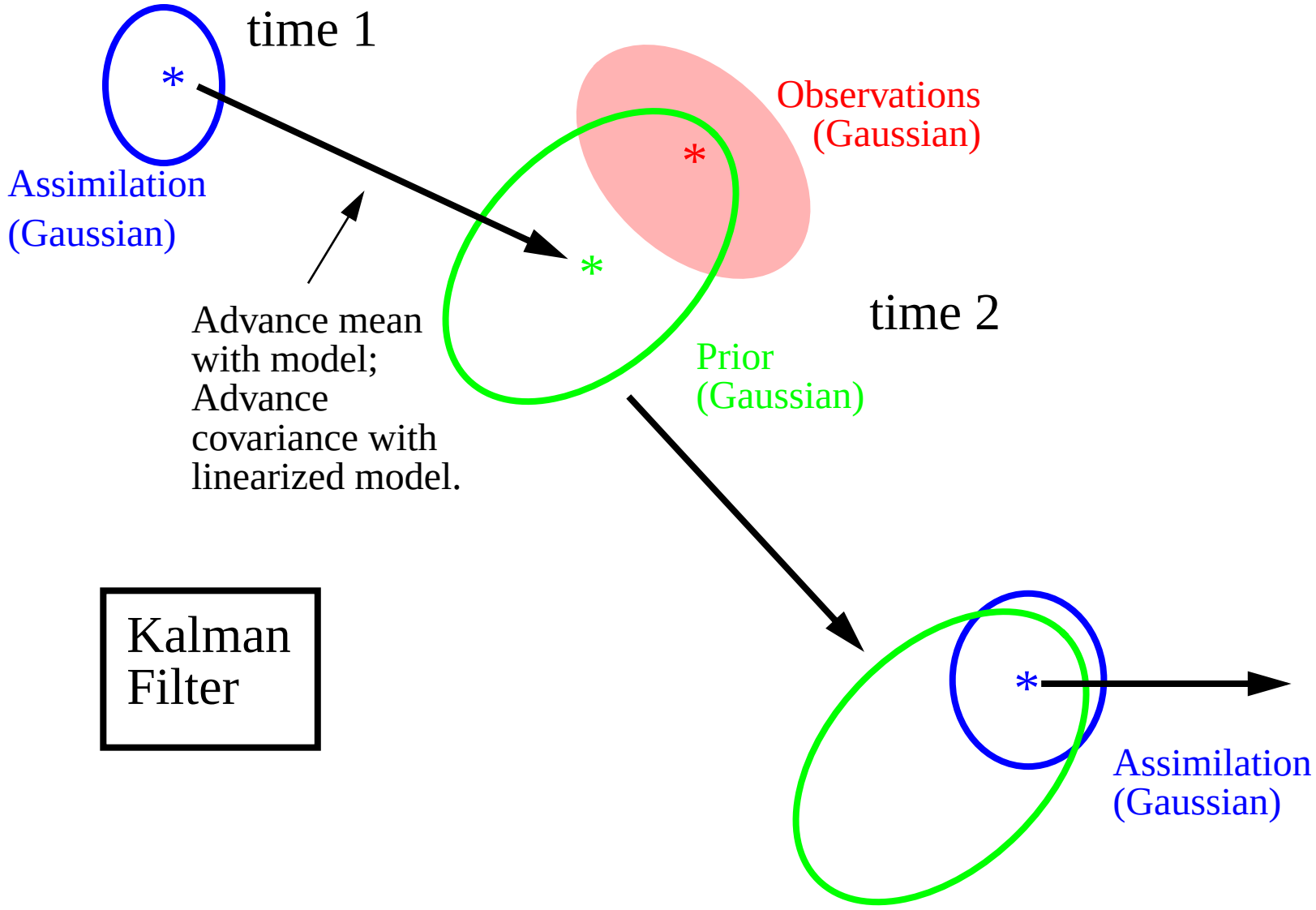
2. Error distributions assumed Gaussian

Fundamental Problem:

Still too expensive for large models

Advancing covariance in linearized model is at least:

O(model_size * model_size)


time 1
Assimilation
(Gaussian)
Observations
(Gaussian)
Advance mean
with model;
Advance
covariance with
linearized model.
time 2
Prior
(Gaussian)
Kalman
Filter
Assimilation
(Gaussian)

Reduced Space Kalman Filters:

Additional simplification:

Assume that covariance projects only on small subspace of model state

Evolving covariance in linearized model projected on subspace may be cheap

Subspace selection:

1. Dynamical: use simplified model based on some sort of scaling

2. Statistical: use long record of model (or physical system) to find reduced basis in which most variance occurs (EOF most common to date)

Problems:

1. Dynamics constrained to subspace may provide inaccurate covariance evolution

2. Observations may not ‘project strongly’ on subspace

3. Errors orthogonal to subspace unconstrained, model bias in these directions can quickly prove fatal

Ensemble ~~Kalman~~ Filters:

Simplifications:

1. Monte Carlo approximation to probability distributions

2. Localization in space, avoids degeneracy from samples smaller than state space and reduces sampling noise

3. Gaussian representation of probability distributions generally used for computing update

Problems:

1. Selecting initial samples for ensembles (Monte Carlo samples)

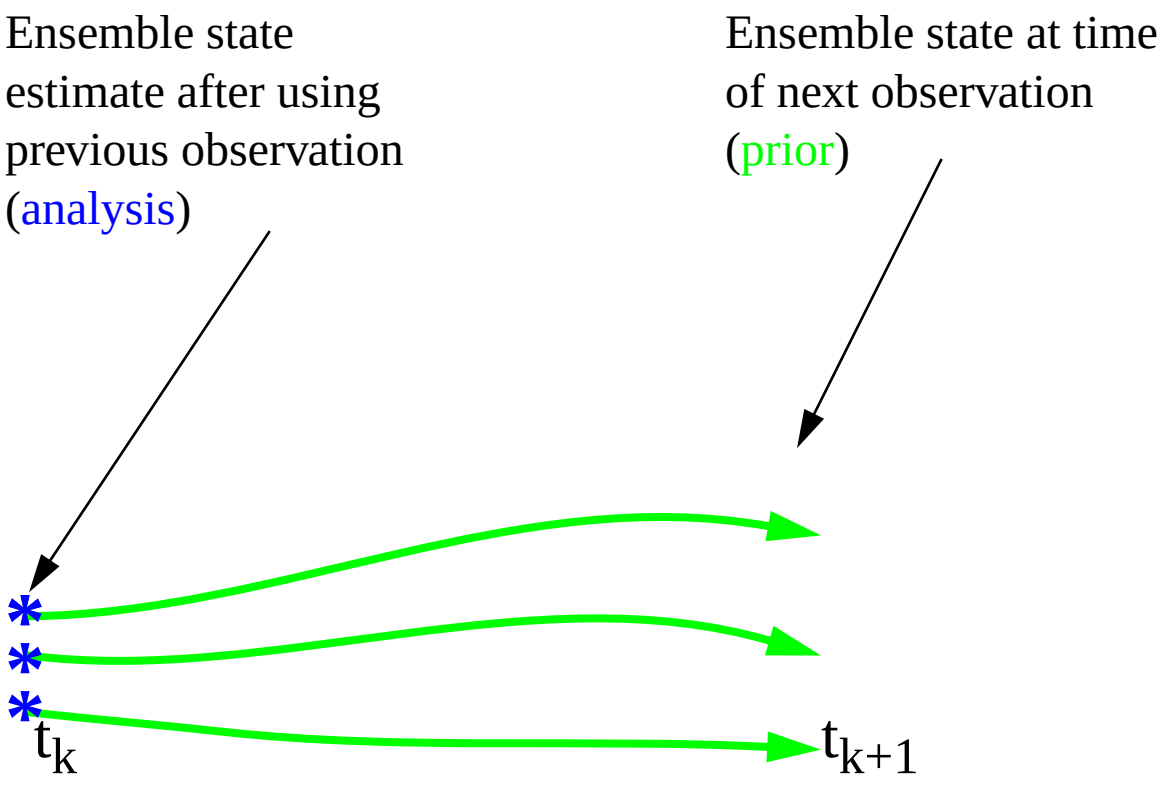
2. Determining degree of spatial localization; sampling error

3. Maintaining appropriate model ‘balances’ in ensemble members

BUT, UNPRECEDENTED EASE OF INITIAL APPLICATION

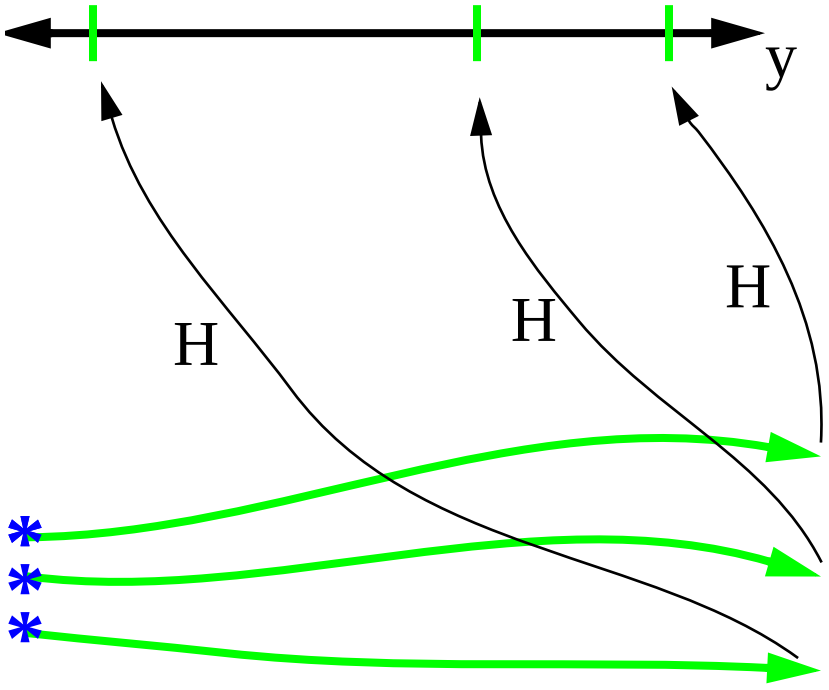
How an Ensemble Filter Works

1. Use model to advance ensemble (3 members here) to time at which next observation becomes available



How an Ensemble Filter Works

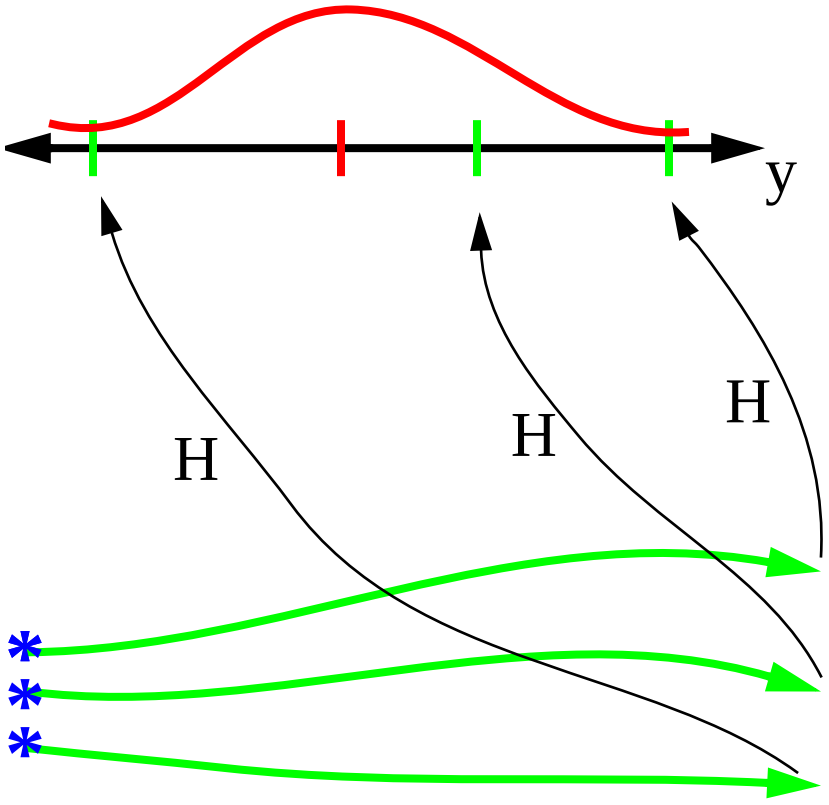
2. Get prior ensemble sample of observation, y=H(x), by applying forward operator H to each ensemble member



Theory: observations from instruments with uncorrelated errors can be done sequentially.

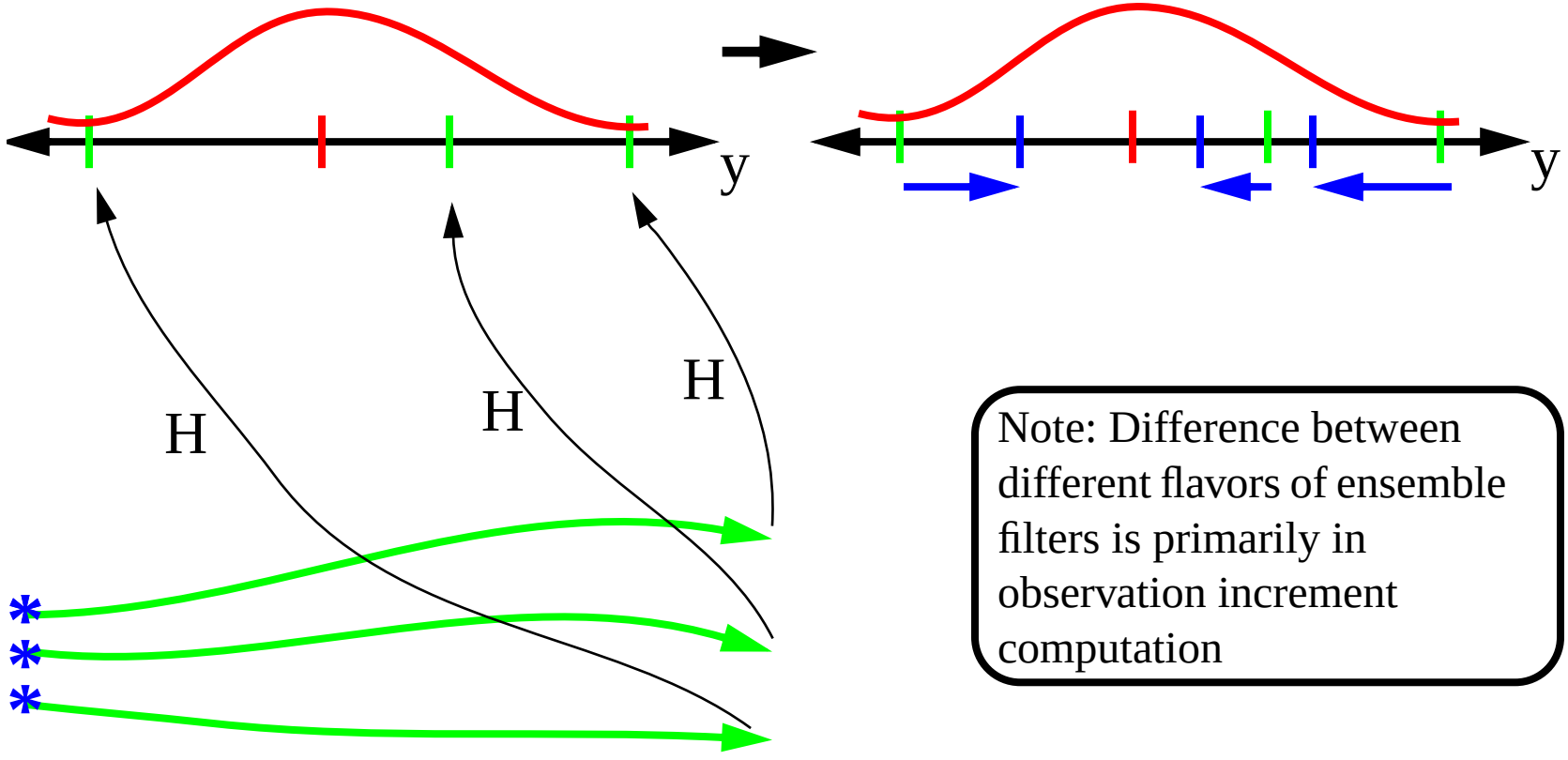
How an Ensemble Filter Works

3. Get observed value and observational error distribution from observing system



How an Ensemble Filter Works

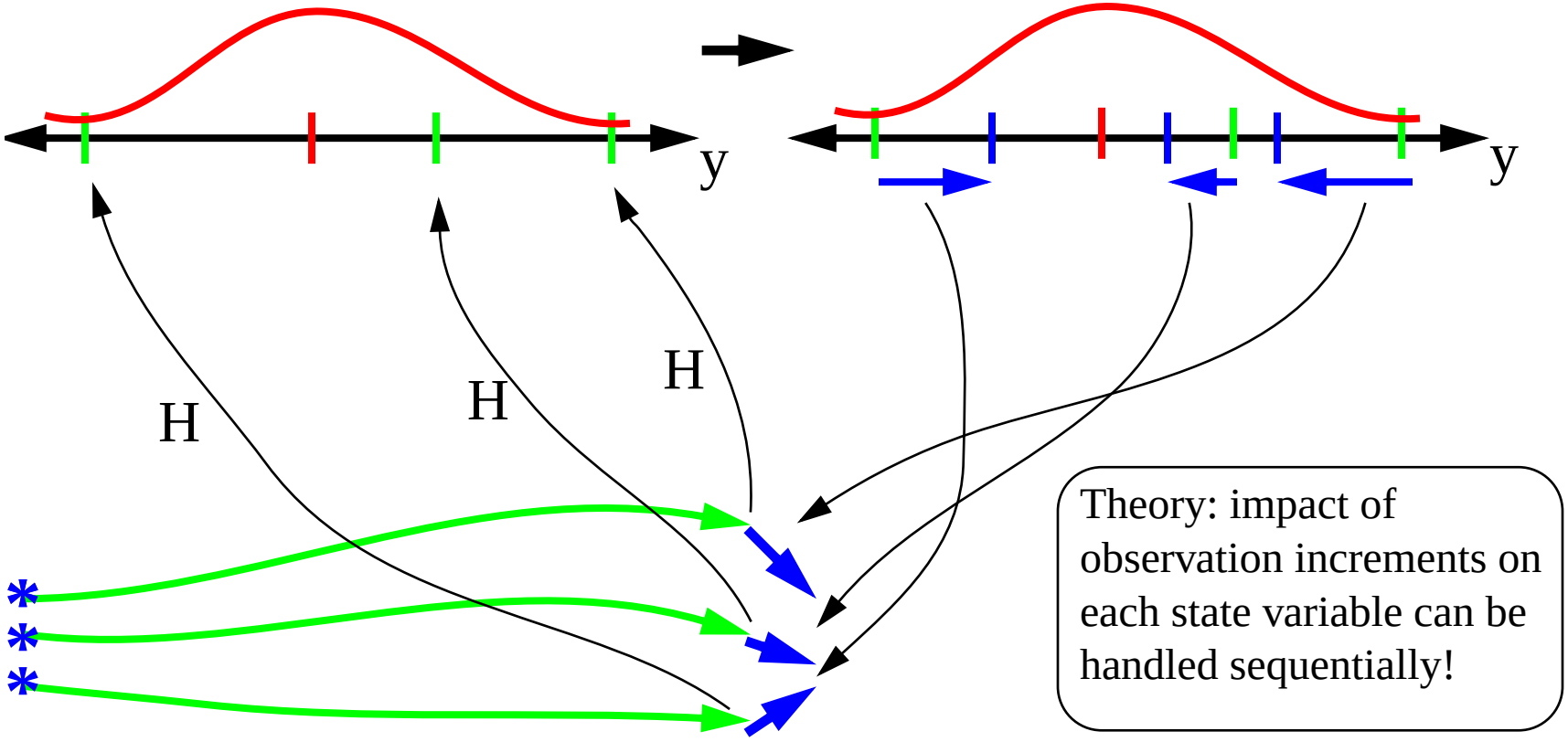
4. Find increment for each prior observation ensemble
(this is a scalar problem for uncorrelated observation errors)



Note: Difference between different flavors of ensemble filters is primarily in observation increment computation

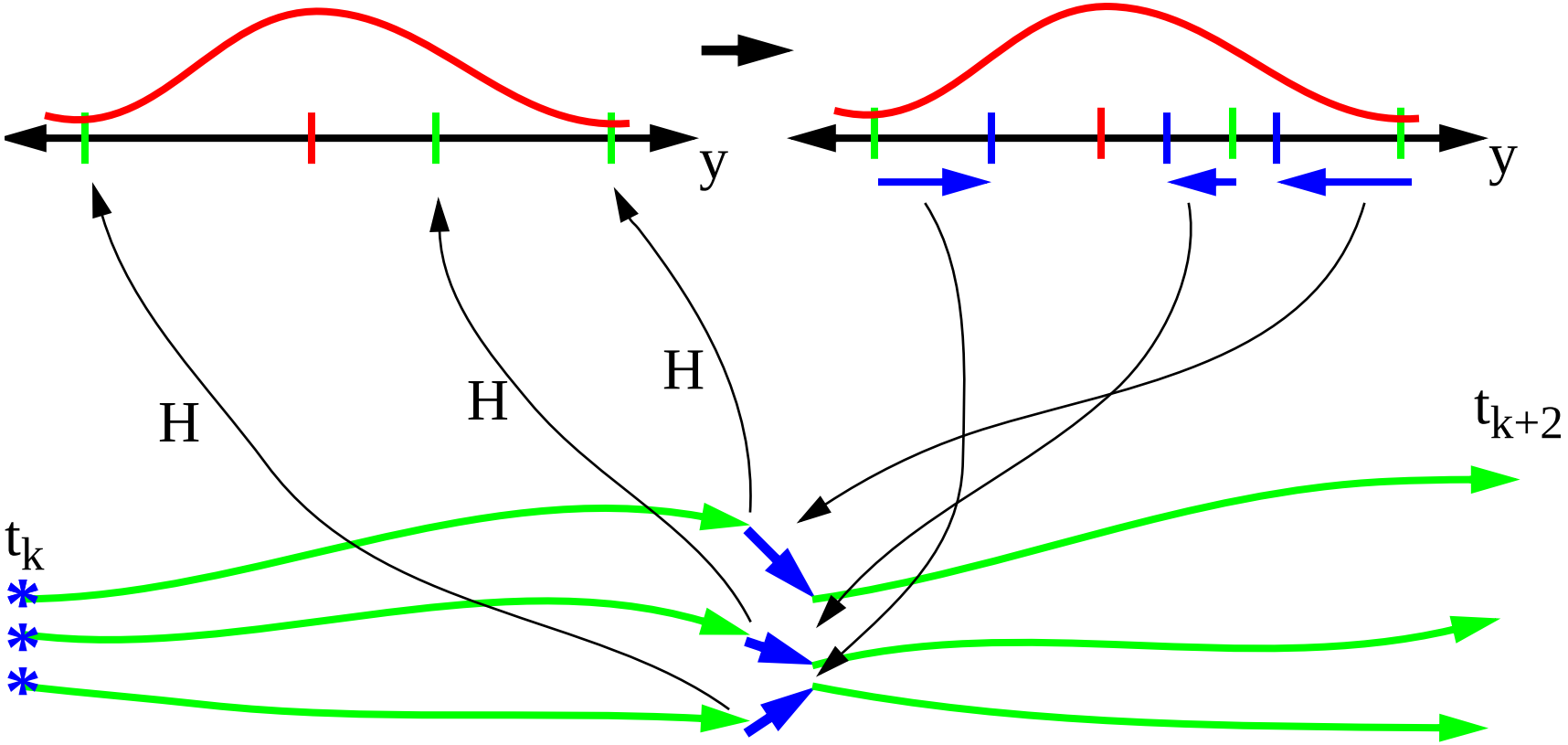
How an Ensemble Filter Works

5. Use ensemble samples of y and each state variable to linearly regress observation increments onto state variable increments



How an Ensemble Filter Works

6. When all ensemble members for each state variable are updated, have a new analysis. Integrate to time of next observation...



Details of Step 4: Finding Increments for Observation Variable Ensemble, y

Scalar Problem: Wide variety of options available and affordable. Examples:

1. Perturbed Observation Ensemble Kalman Filter (EnKF); stochastic

2. Ensemble Adjustment Kalman Filter (EAKF); deterministic

Key to Kalman Filters: Product of Gaussians is Gaussian

Prior ensemble sample mean $\bar{y}^p$ and variance Σ^p

Observation y^o with observational error variance Σ^o

Posterior Variance is:

$$\Sigma^u = \left[\left(\Sigma^p\right)^{-1} + \left(\Sigma^o\right)^{-1} \right]^{-1} \qquad (11)$$

and mean is:

$$\bar{y}^u = \Sigma^u \left[\bar{y}^p / \Sigma^p + y^o / \Sigma^o \right] \qquad (12)$$

Details of Step 4: Perturbed Observation Ensemble Kalman Filter (EnKF)

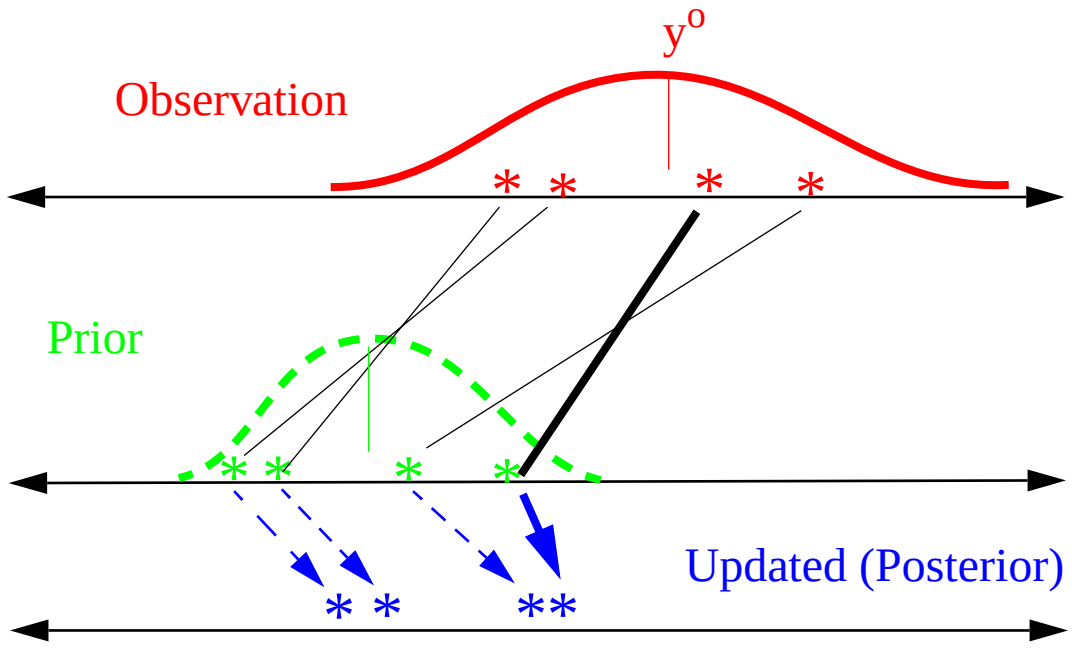
1. Apply (11) once to compute updated covariance Σ^u

2. Create N-member sample of observation dist. by adding samples of obs. error to y^o

3. Apply (12) N times to compute updated ensemble members, $\bar{y}^u_i$

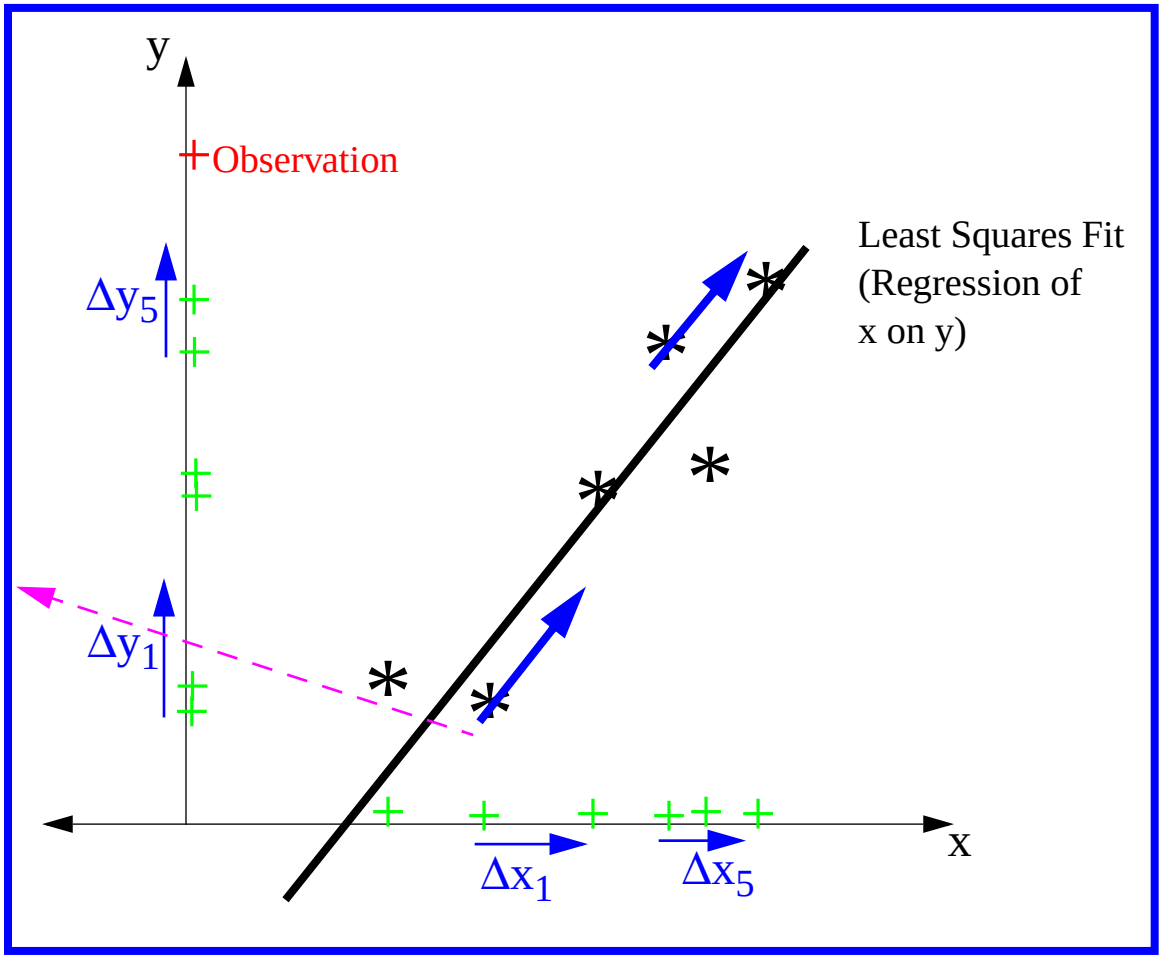
   Replace $\bar{y}^p$ with ith prior ensemble member, y^p_i

   Replace y^o with ith value from random sample, y^o_i



Details of Step 5: Compute state var. increments from obs. variable increments

Regression using joint sample statistics from ensembles: can be done sequentially!



Regression begins with least squares fit to sample, *

Increments for state variable, x, multiplied by |correl(x, y)|

Large sample size needed to filter ‘noise’

Trade-offs with ‘local’ linearization:
Precision vs. accuracy

Challenges Being Overcome!

Problem 1. Sampling error impacts estimates of increments

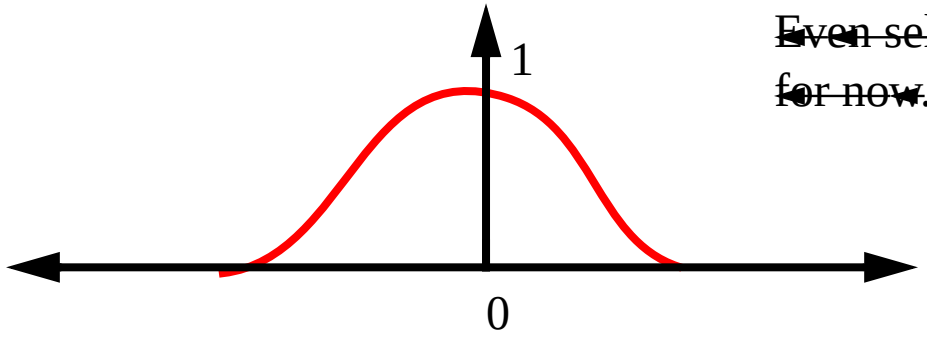
Key: estimates of regression coefficients have errors

Many obs. with small (or zero) expected correlations => error build-up

Solution: Reduce impact of observations as function of ensemble size, sample correlation, and expected distribution of correlation

But...need this prior estimate (may be mostly unknown?)

Newly developed hierarchical ensemble filters have solved this problem!



~~Even selecting envelope is tricky for now.~~

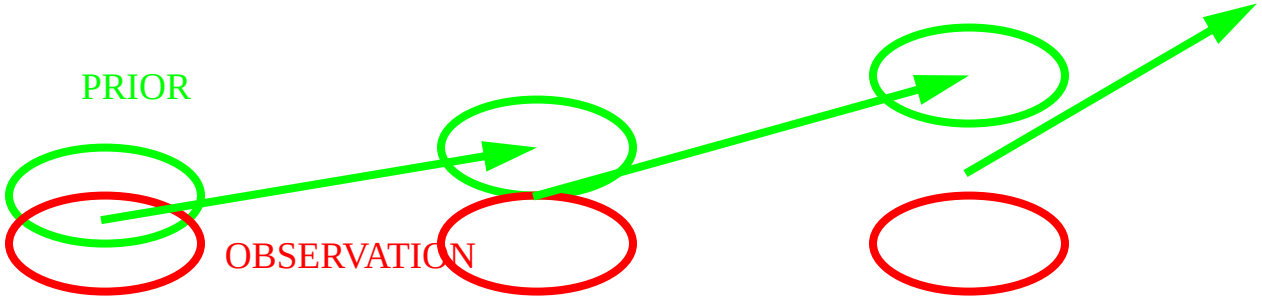
Problem 2: Model Bias (Systematic Error)

Filter equations assume prior estimate (and observations) are unbiased

Questionable for Observations, ridiculous for Models

Biased prior estimate will cause observations to be given too little weight

Repeated applications lead to progressively less weight, estimate can diverge



Dealing with model bias is most serious remaining challenge

1. Can reduce confidence in model prior estimates by some constant factor

2. Explicitly model the model bias as an extended state vector and assimilate coefficients of this bias model

   Model: $dx/dt = F(x)$

   Model plus bias model: $dx/dt = F(x) + \varepsilon(t)$; $d\varepsilon/dt = 0$

   where ε is a vector of the same length as x

Very tricky: if we knew much about modeling the bias, we could remove it

3. Adaptive filters: Use observations to tell us if we're drifting from obs. (Promising!)

Problem 3. Initial conditions for ensembles

Key: Bayesian, assumes initial ensembles are magically available

Solution: For ergodic models spin-up by running ensemble a very long time from arbitrary initial perturbations, slowly ‘turn on’ observations

But... this may be impossible for some models (WRF regional applications)

Given prior knowledge of expected correlations (see problem 1) should be able to generate appropriate ensemble ICs

Is this a problem for ‘chemical’ assimilation?

Problem 4. Assimilation of variables with discrete distributions

Key: ensemble prior may indicate zero probability of an event that is occurring

i.e. All ensemble members say no rain but rain is observed

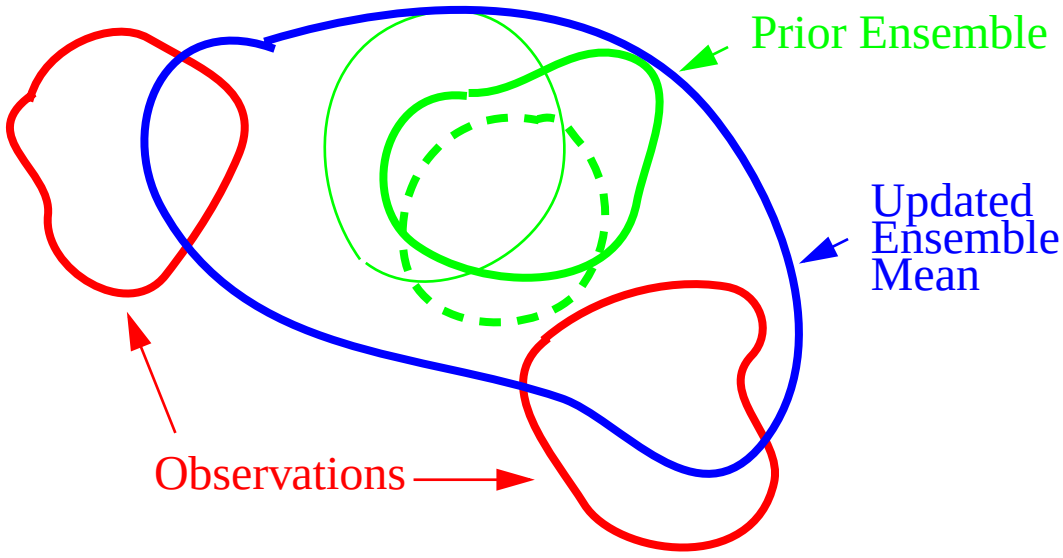
Directly related to existence of discrete convective cells

Solutions: Apply methods for accounting for model error

Redefine state variables to avoid discrete probability densities

Research on this problem is in its infancy

Problem 4: Assimilation of Discrete Distributions



Example: assimilation of convective elements

Prior is 'certain' that there are no convective cells outside the green areas

Observations indicate discrete areas outside the green

This is indicative of highly non-linear problem

Ensemble techniques, at best, tend to smear out prior discrete structures

4D-Var is likely to have non-global local minima

But, we think we know what we want to do

- Keep information from prior on larger scale 'background'
- Introduce cells where observed

Requires new norms or ways to deal with model bias as function of scale

Using Data Assimilation to Constrain Model Parameters

Example from another low-order model: Lorenz-96 Model

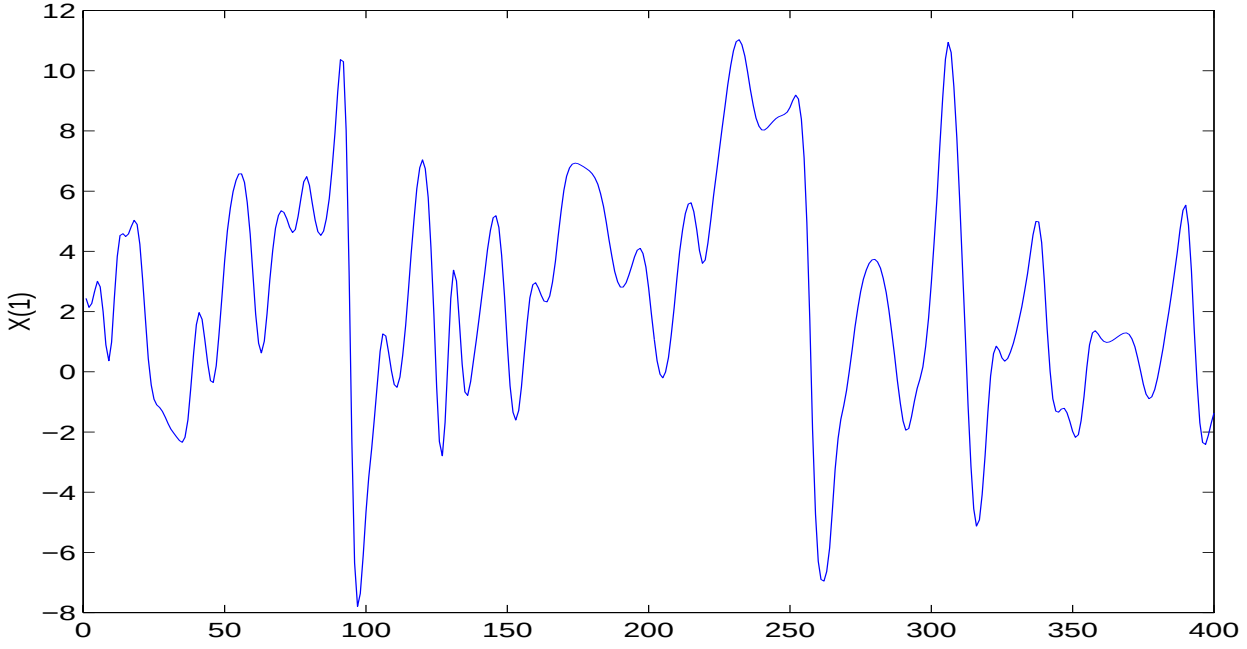
Variable size low-order dynamical system

N variables, $X_1, X_2, ..., X_N$

$dX_i / dt = (X_{i+1} - X_{i-2})X_{i-1} - X_i + F$

$i = 1, ..., N$ with cyclic indices

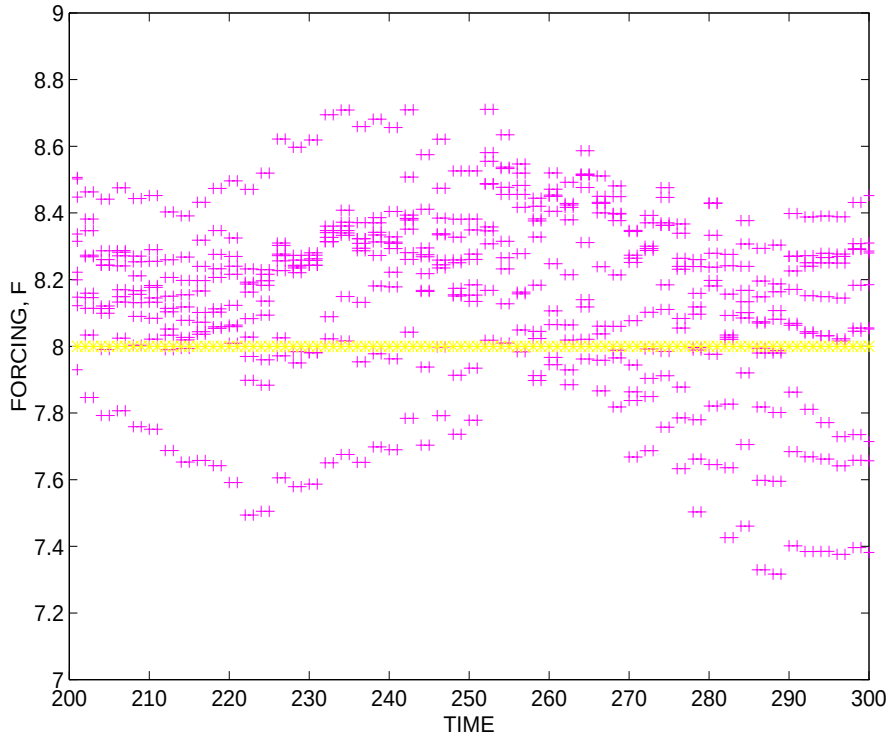
Use N = 40, F = 8.0, 4th-order Runge-Kutta with dt=0.0



Lorenz-96 Free Forcing Model Filter

20 Member Ensemble (10 Plotted) Obs Every 2 Steps of State Variables only

Truth in Yellow (8.0) Ensemble



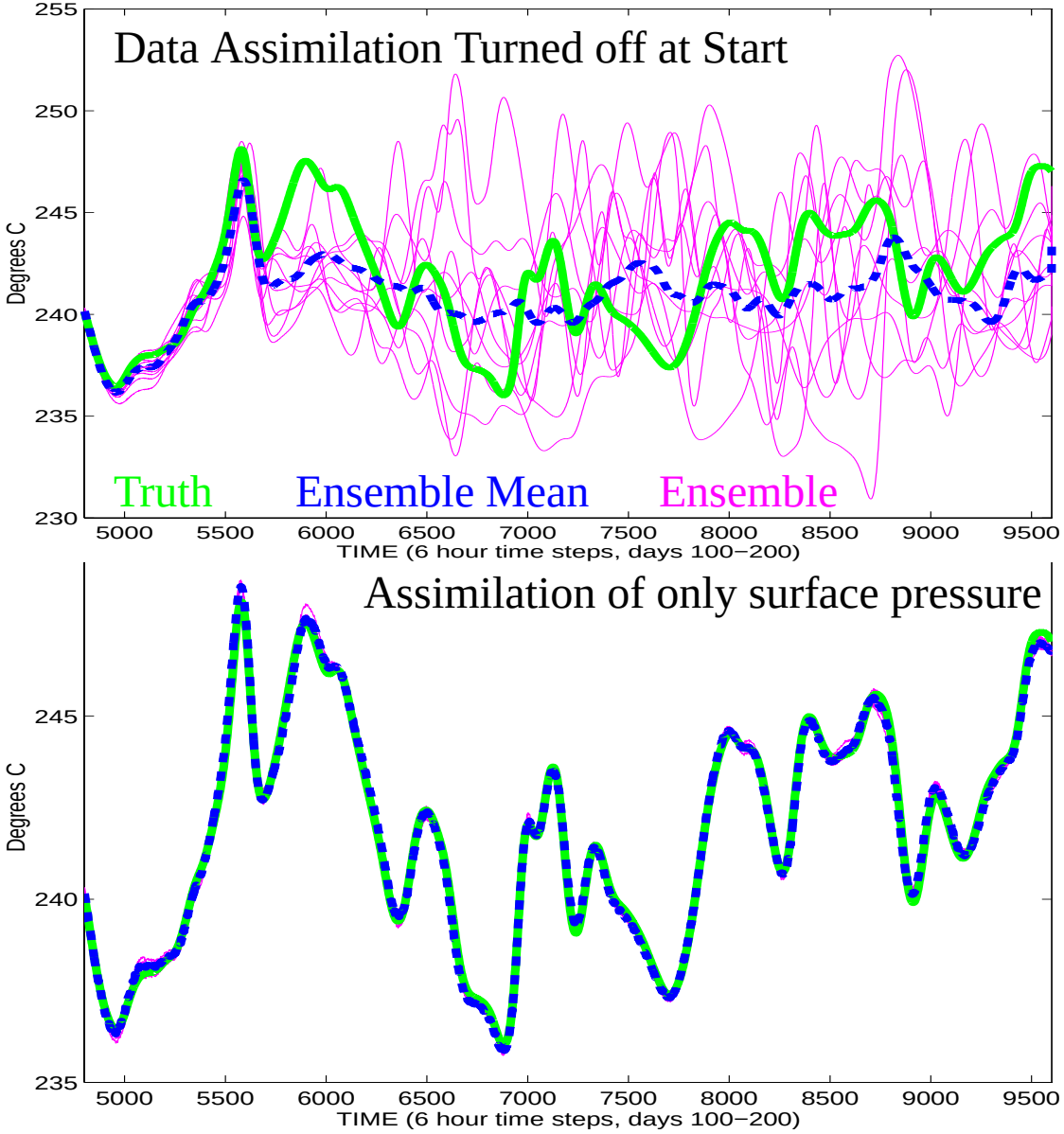
Can treat model parameters as free parameters

Here, the forcing F is assimilated along with the state

This is potential mechanism for dynamic adjustment of unknown parameters and for dealing with unknown model systematic error

Many models include a number of poorly know free parameters
May be able to improve models by using data to constrain these
Observation system parameters can also be constrained by data (obs. error for instance)

Evaluating and Designing Observing Systems: Information Content of Observations



Example: What is information content of surface pressure observations in an atmospheric GCM?

Observing System Simulation Experiment (OSSE) with an Ensemble Filter

Observations generated from a model run (truth in green)

Same model assimilates

Surface pressure is able to closely constrain entire atmosphere

Figure shows mid-latitude, mid-troposphere temperature

Towards a General, Flexible Ensemble Assimilation Facility

Goals:

1. Assimilation that works with variety of models and obs. types
2. Coding for system must be easy to implement (weeks max)
3. Must allow complicated forward operators
4. Must NOT require assimilation expertise for good performance
5. EXCELLENT performance with added expertise/development
6. Must run tolerably on variety of platforms with little effort
7. Must run efficiently on variety of platforms with expertise

Data Assimilation Research Testbed (DART)

Basic framework implemented
Primarily implementing ensemble (Kalman) filters
Variational for low-order models only
Plans MAY include a variational (4D-Var) capability

DART compliant models (largest collection ever with assim system)

CGD's CAM 2.0
GFDL FMS B-grid GCM
Many low-order models available
MMM's WRF model
NCEP MRF (GFS)
GFDL MOM ocean model partially incorporated in earlier version

It Really Works!

Model:

CAM 2.0 T42L26

U,V, T, Q and PS state variables impacted by observations

Land model (CLM 2.0) not impacted by observations

Observed SSTs

Assimilation / Prediction Experiments:

Uses observations used in reanalysis

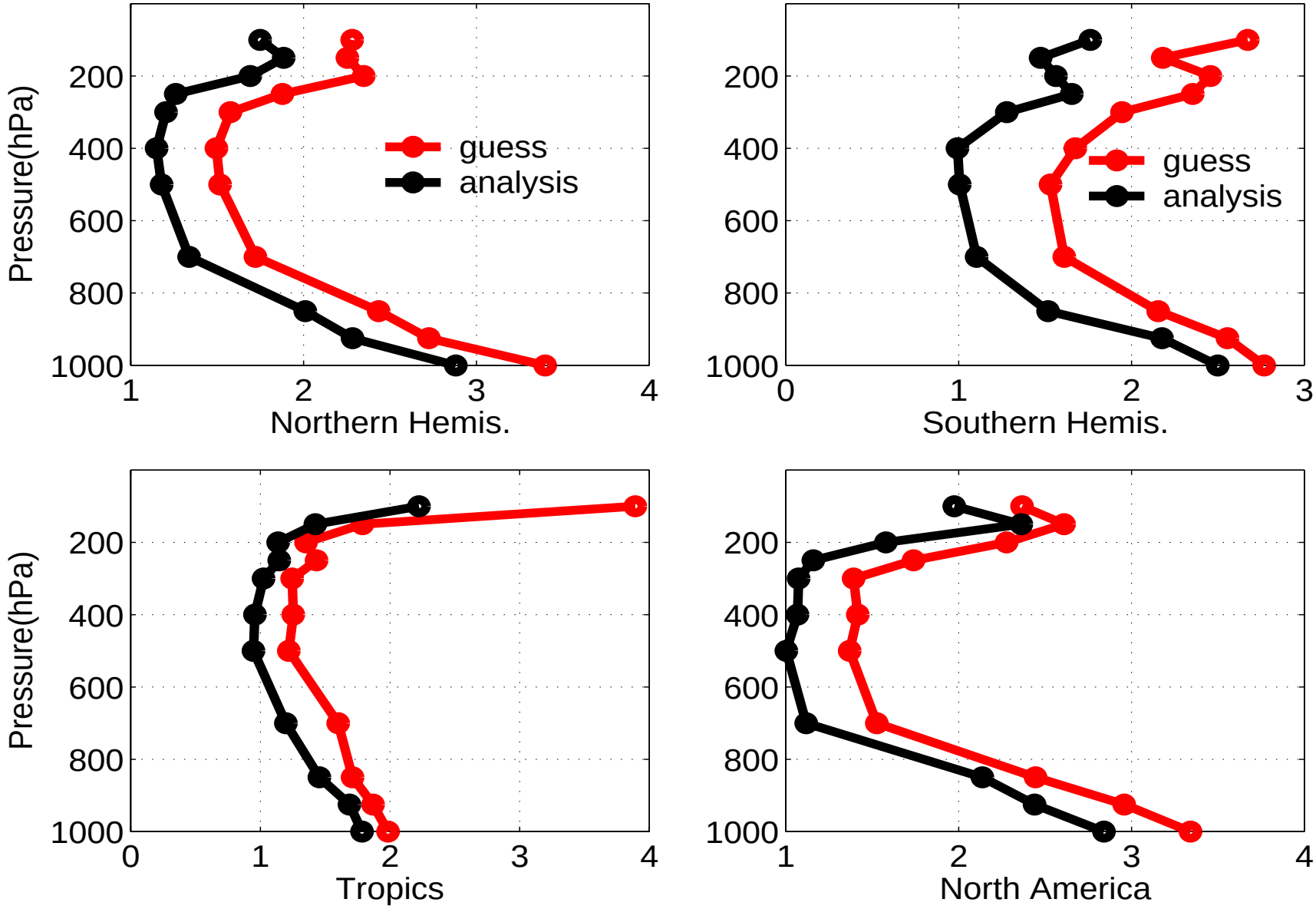
(Radiosondes, ACARS, Satellite Winds...)

Initial tests for first week of January, 2003

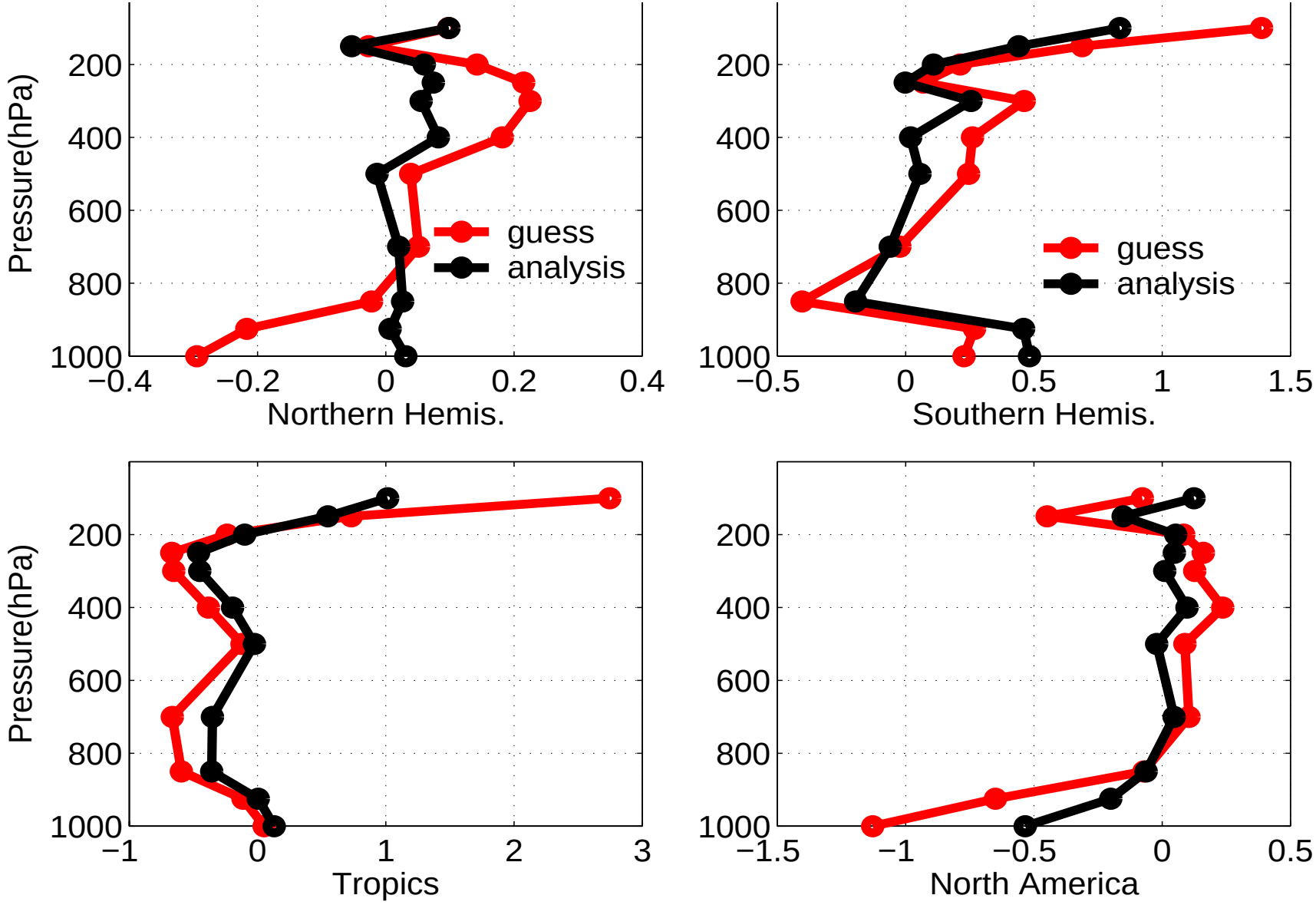
Assimilated every 6 hours

Run on CGD linux cluster Anchorage

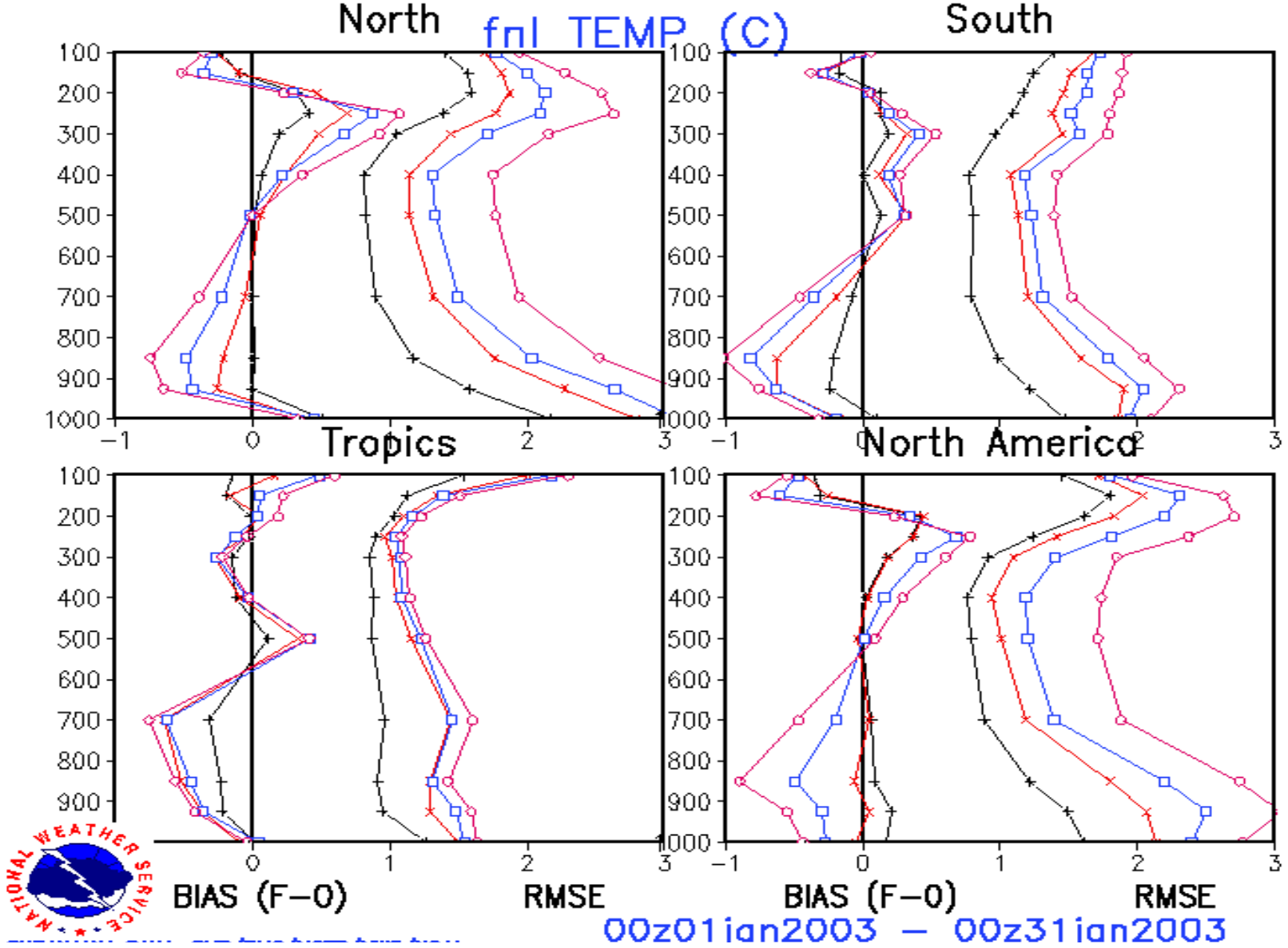
CAM RESULTS: ENSEMBLE MEAN RMS TEMP. ERROR



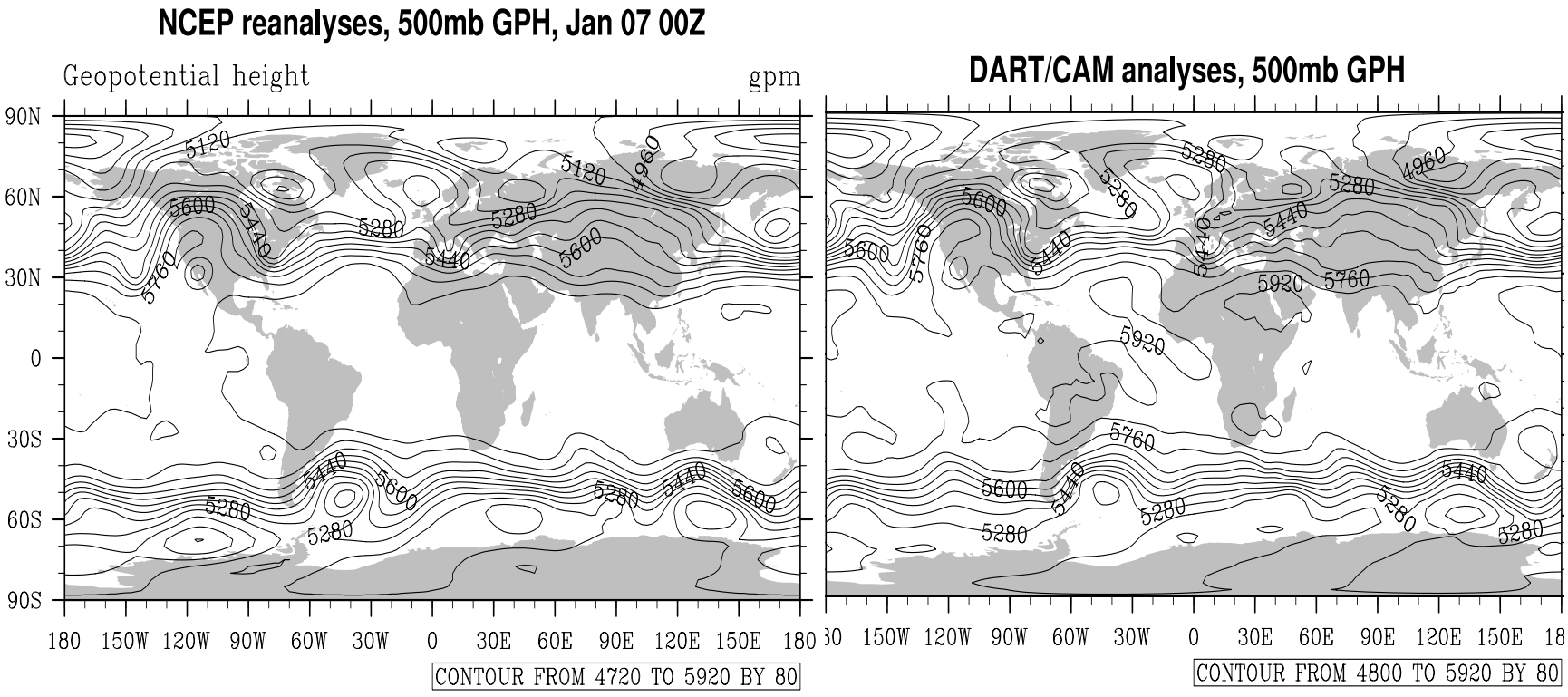
CAM RESULTS: Ensemble Mean Time Mean Temperature BIAS

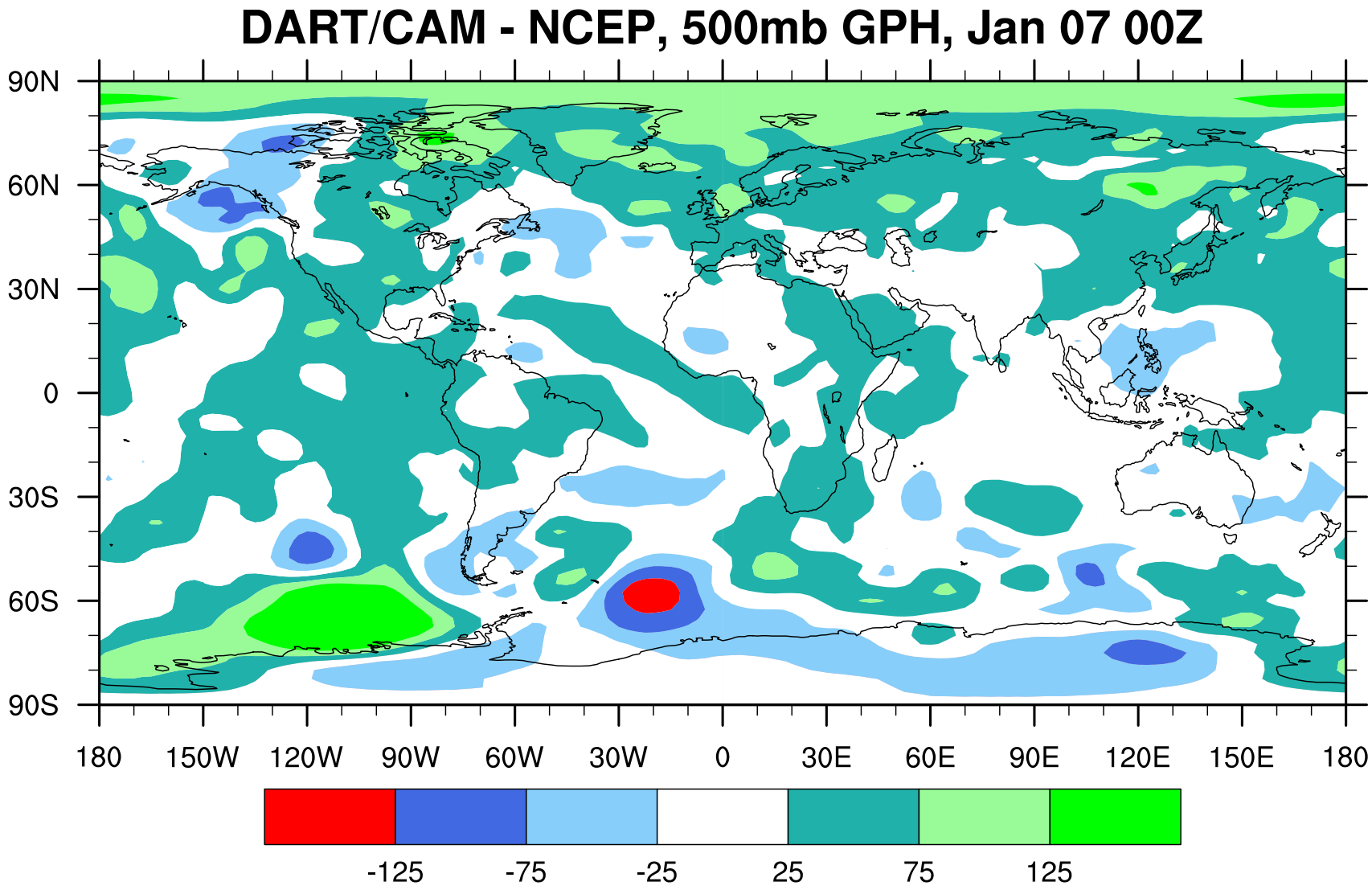


NCEP GFS BIAS (Left), RMS (right): Black Analysis, Red Guess

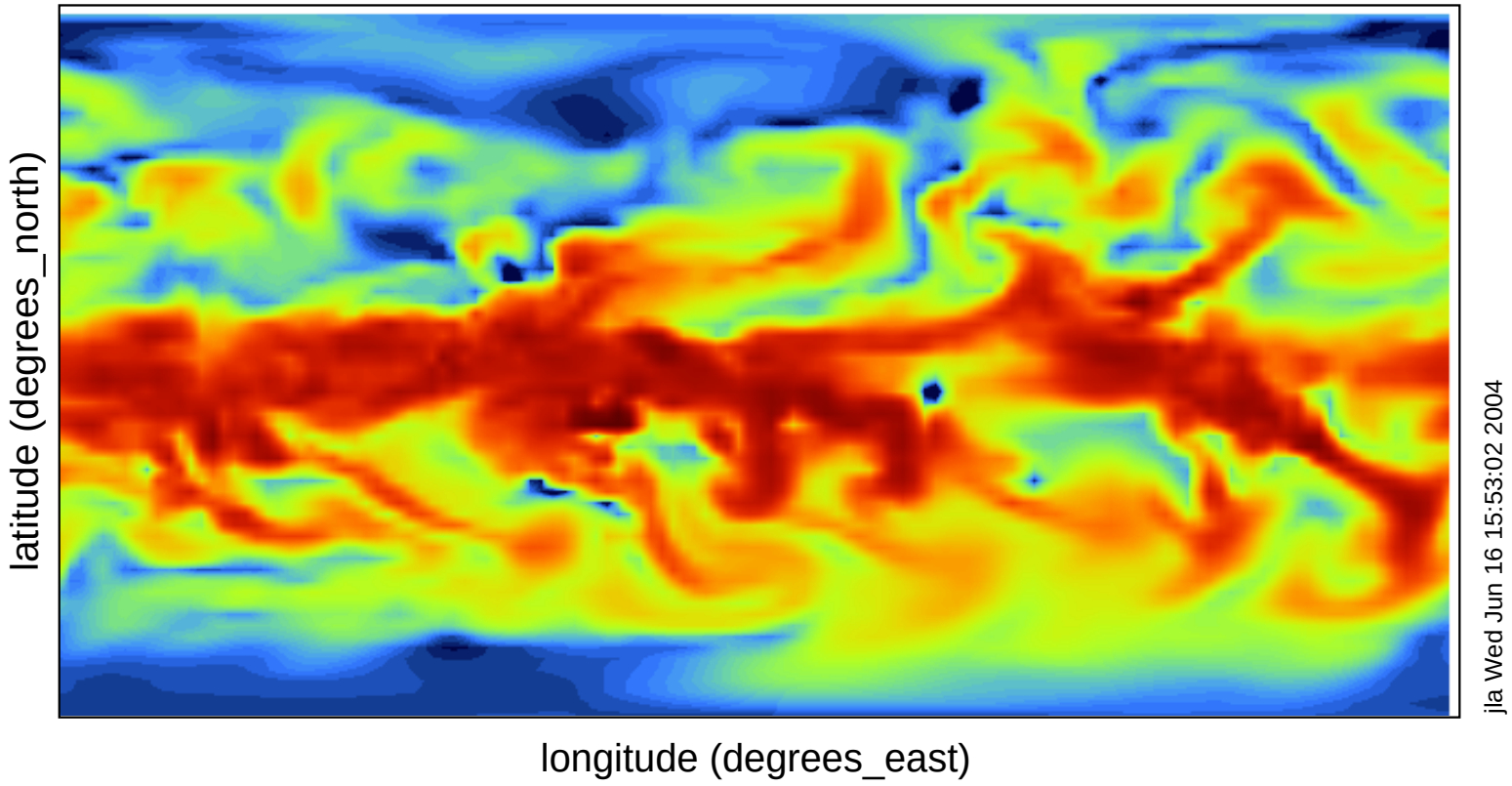


500mb Height Comparison to NCEP CDAS Analysis; Jan. 7, 2003




DART/CAM - NCEP, 500mb GPH, Jan 07 00Z
90N
60N
30N
0
30S
60S
90S
180
150W
120W
90W
60W
30W
0
30E
60E
90E
120E
150E
180
-125
-75
-25
25
75
125

Captures details of q without q obs; q increments from other obs!



Comparison to Adjoint: Computational Cost and Implementation

1. Model integrations
   a. Filter requires N forward integrations of model; O(10) sufficient?
   b. Adjoint requires K*L forward and backward integrations
      K - number of observation intervals over which optimization is performed
      L - average number of iterations of minimization solver
      K*L at least O(10) for any envisioned application

2. Assimilation algorithm cost
   a. Filter: O(αNnm) :N is ensemble size, n is model size, m is number of obs
      α related to what fraction of state variables are impacted by given ob.
      In certain scenarios this may reduce order of cost
   b. Adjoint: O(nm) in best of all possible cases
      Relation of constant factors not clear, depends on ensemble size

3. Ease of implementation
   a. Filter-needs no model specific software
   b. Adjoint-requires exact adjoint model plus linear tangent (can be a pain)

Conclusions

1. Ensemble filters can do complex, real-data assimilation problems

2. Implementing filters is extremely simple
   (compared to most assimilation techniques)

3. Filters are powerful in extracting multi-variate relations

4. Filters can deal with tracers, observed or unobserved

5. Assimilation is relatively cheap, but ensembles are required