Data assimilation by morphing ensemble Kalman filters

with application to wildland fires

Jan Mandel and Jonathan Beezley

Department of Mathematical and Statistical Sciences
University of Colorado Denver

Supported by NSF 0325314, 0623983, and 0719641

Modeling and Sensing Environmental Systems, LNCC, Petrópolis, Brazil

August 15, 2008

Selected Collaborators

Part of the wildfires DDDAS project

- Janice Coen (NCAR) - fire science, the fire model we started from, meteorology
- Craig Douglas (Wyoming) - computer science
- Tony Vodacek (RIT) - sensors and airborne images

Data assimilation

- Model
  - must support the assimilation cycle: export, modify, and import state
  - the state must be described: what, when, where
  - changes to the state must be meaningful: no discrete datastructures (such as tracers)
- Data
  - must have error estimate
  - must have metadata: what, when, where
- Observation function
  - connects the data and the model
  - creates synthetic data from model state to compare
- Data assimilation algorithm
  - adjusts the state to match the data
  - balances the uncertainty in the data and in the state

DDAS

Data assimilation is a part of a wider concept:
Dynamic Data Driven Application Systems

- Mathematical side:
  - data assimilation
  - measurement location to minimize uncertainty
  - control
  - models that survive meddling
  - updating of math objects (e.g. matrix decomposition, . . .)
- Computer Science side:
  - real-time and embedded systems
  - support ever changing dynamic datastructures
  - integrate into one flexible system
    - data assimilation and the model
    - sensors and other data sources
    - visualization and human interface
    - continuous, stochastic, and discrete modeling

The Ensemble Kalman Filter (EnKF)

- uses the model as a black box
- adjusts the state by making linear combinations of ensemble members (OK, locally in local versions of the filter, but still only linear combinations)
- if it cannot match the data by making the linear combinations, it cannot track the data
- probability distributions close to normal needed for proper operation

The Ensemble Kalman Filter (EnKF)

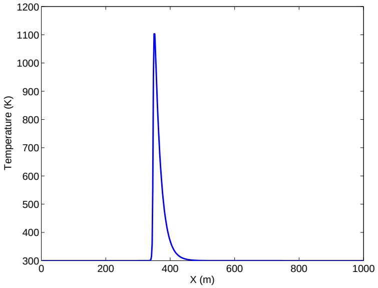
$$X^a = X^f + K\left(Y - HX^f\right), \qquad K = P^f H^T (HP^f H^T + R)^{-1}$$

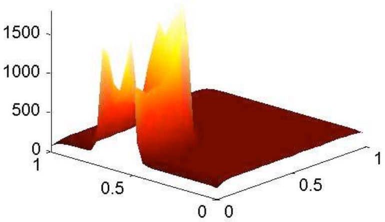
- X^a: Analysis/Posterior ensemble
- X^f: Forecast/Prior ensemble
- Y: Data
- K: Kalman gain
- H: Observation function
- P^f: Forecast sample covariance
- R: Data covariance

Basic assumptions:

- Model and observation function are linear
- Forecast and data distributions are independent and Gaussian

A simple wildfire model





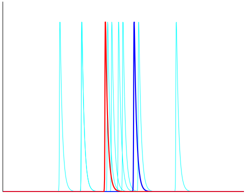
1D temperature profile

2D temperature profile

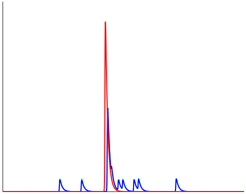
Solutions produce non-linear traveling waves and thin reaction fronts.

A simple example in 1D: filter degeneracy

Forecast ensemble



Analysis ensemble



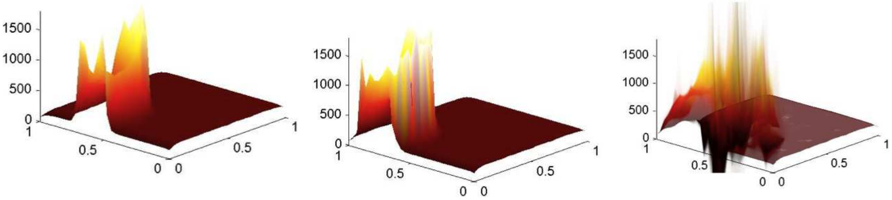
- Ensemble size, $N = 10$
- Identity observation function, $H = I$
- Data covariance, $R = 10\text{tr}(P^f)I$

- Cyan: ensemble
- Blue: last ensemble member
- Red: data

- $\text{tr}(P^f) = 4.7 \times 10^6,\ \text{tr}(P^a) = 4.8 \times 10^2$
- the ensemble spread decreased severely
- the ensemble members are non-physical

An example in 2D: non-physical results

Forecast ensemble | Data | Analysis ensemble



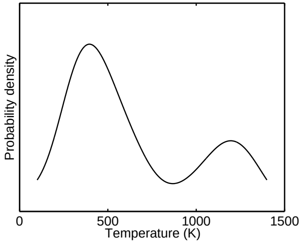
- Forecast ensemble generated by random spatial perturbations of the displayed image
- Analysis ensemble displayed as a superposition of semi-transparent images of each ensemble member
- Identity observation function, $H = I$
- Data variance, 100 K

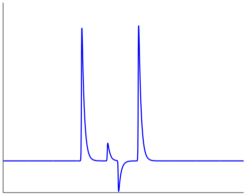
What went wrong?

The Kalman update formula can be expressed as

$$X^a = A(X^f)^T,$$

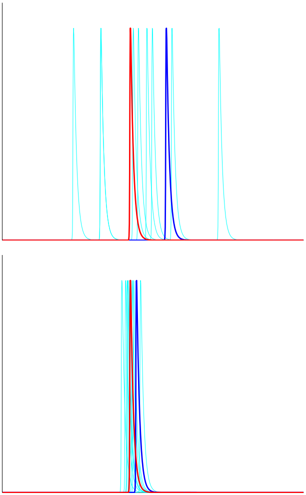
so $X_i^a \in \text{span}\{X^f\}$, where the analysis ensemble is made of linear combinations of the forecast.





Spatial perturbations yield forecast distributions with two modes centered around burning and non-burning regions.

Representing spatial error, in 1D



Define a non-linear transformation

$$\mathcal{T}(u_i) = \text{argmax}\{u_0\} - \text{argmax}\{u_i\} = t_i$$
$$\mathcal{T}^{-1}(t_i) = u_0(x + t_i) = u_i$$

- t_i, translation of ensemble member i from a “reference” state u_0
- run the EnKF with scalar ensemble t_i and data $\mathcal{T}(Y)$
- recover analysis ensemble by applying the inverse transformation
- $t_i \sim \mathcal{N}(m, \sigma)$, by original construction of forecast ensemble

But what about 2D?

Morphing functions

- A morphing function, $T : \Omega \to \Omega$ defines a spatial perturbation of an image, u.
- It is invertible when $(I + T)^{-1}$ exists.
- An image u "morphed" by T is defined as $\tilde{u} = u(x + Tx) = u \circ (I + T)(x)$.

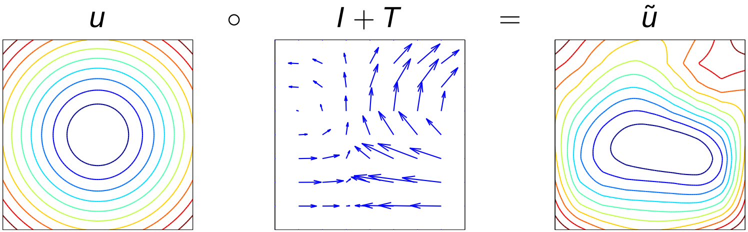


Image registration

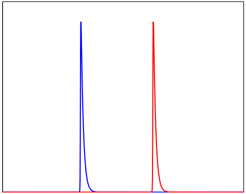
Goal: Given two images u and v, find an invertible morphing function, T, which makes $u \circ (I + T) \approx v$, while ensuring that T is "small" as possible.

Image registration problem

$$J_{u \to v}(T) = ||u \circ (I + T) - v||_{\mathcal{R}} + ||T||_{\mathcal{T}} \to \min_T$$

- $||r||_{\mathcal{R}} = c_{\mathcal{R}}||r||_2$
- $||T||_{\mathcal{T}} = c_{\mathcal{T}}||T||_2 + c_{\nabla}||\nabla T||_2$
- $c_{\mathcal{R}}$, $c_{\mathcal{T}}$, and c_{∇} are treated as optimization parameters

Minimizing the objective function



$$\nabla J_{u \to v}(0) = 0!!!$$

Problems with minimization:

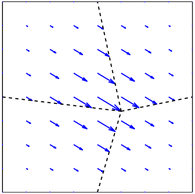
- Highly nonlinear
- Many local minima
- Need an automated procedure
- Needs to be done quickly

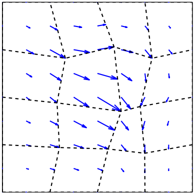
Steepest descent methods do not work in general.

Problems to overcome

- The global minima can be found by steepest descent if one starts with a morphing function close enough to this solution.
- Run steepest descent many times starting from a suitably dense sample of morphing functions.
- The number of samples needed grows exponentially with the number of grid points!!!
- Need a heuristic simplification.

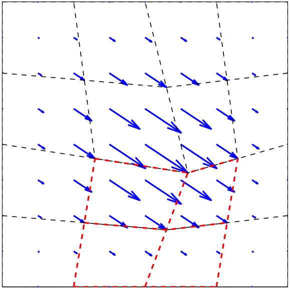
Minimization by sampling

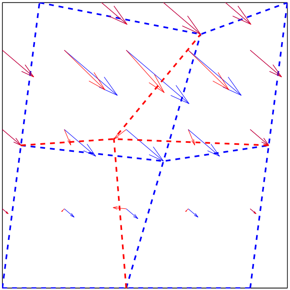




- Probe the solution space by moving the center to sample points and evaluating the objective function and taking the minimum.
- Morphing function on grid points determined by some sort of interpolation.
- Refine the grid and repeat until desired accuracy is reached.
- When using bilinear interpolation, invertibility is achieved when all grid quadrilaterals are convex.

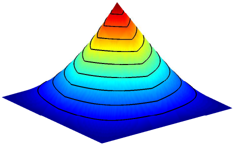
Grid refinement





The objective function need only be calculated locally, within the subgrid, allowing acceptable computational complexity, $O(n \log n)$.

Interpolation methods



Bilinear interpolation:

- Not differentiable
- Easy to enforce invertibility
- Commonly used in the literature

Tensor product cubic splines

- 0 value on the boundary
- 0 gradient on the boundary
- Globally differentiable
- More difficult to maintain invertibility

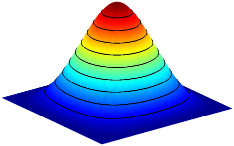
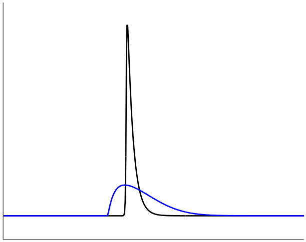


Image smoothing



A smoothed temperature profile (in blue) with bandwidth 200 m.

Gaussian kernel with bandwidth h

$$G_h(x) = c_h \exp\left\{-\frac{x^T x}{2h}\right\}$$

Smoothing by convolution with $G_h(x)$ improves performance of steepest descent methods applied to $J_{u\to v}(T)$.

Multilevel registration algorithm

```
Input:  T_0, u, v
Output:  T
Parameters:  nlev, h_i, c_R, c_T, c_∇, N_i
for i ← 1 to nlev
    ũ ← u * G_{h_i} and ṽ ← v * G_{h_i} // smooth images
    foreach subgrid at level i
        choose {δT_j}_{N_i} // sampled corrections
        T_0 ← argmin{J_{ũ→ṽ}(T_0 + δT_j)}
        T ← steepest_descent(T_0)
        if stopping_condition(T), return
```

Input: T_0, u, v
Output: T
Parameters: nlev, $h_i, c_{\mathcal{R}}, c_{\mathcal{T}}, c_{\nabla}, N_i$
for $i \leftarrow 1$ to nlev
  $\tilde{u} \leftarrow u * G_{h_i}$ and $\tilde{v} \leftarrow v * G_{h_i}$ // smooth images
  foreach subgrid at level i
    choose $\{\delta T_j\}_{N_i}$ // sampled corrections
    $T_0 \leftarrow \text{argmin}\{J_{\tilde{u}\to\tilde{v}}(T_0 + \delta T_j)\}$
    $T \leftarrow$ *steepest_descent*(T_0)
    if *stopping_condition*(T), return

The morphing transformation

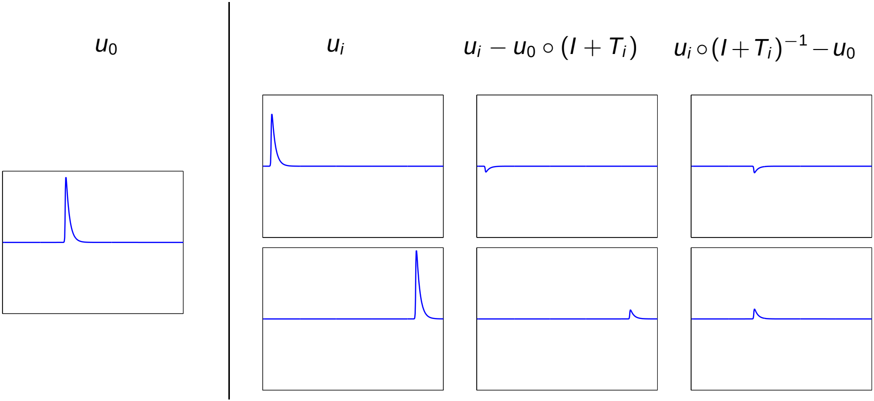
Morphing transformation

$$\mathcal{M}_{u_0} u_i = \begin{cases} T_i, & \text{by image registration} \\ r_i = u_i \circ (I + T_i)^{-1} - u_0, & \text{representation error} \end{cases}$$

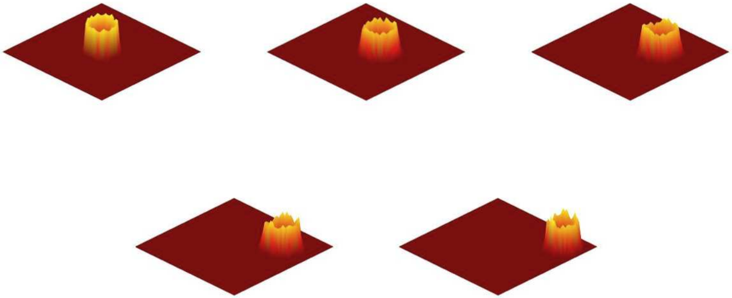
$$\mathcal{M}_{u_0}^{-1}[T_i, r_i] = u_i = (u_0 + r_i) \circ (I + T_i).$$

Just as in the 1D case, apply $\mathcal{M}_{u_0}$ to the ensemble and the data, run the EnKF on the transformed variables, and apply the inverse transformation to get the analysis ensemble.

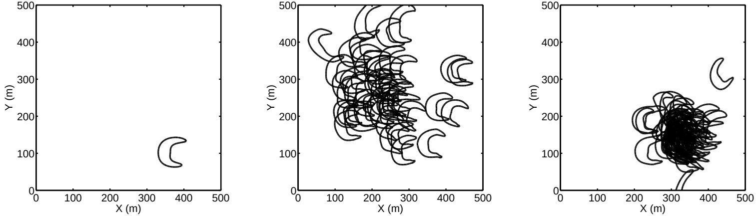
Why use $r_i = u_i \circ (I + T_i)^{-1} - u_0$ instead of $r_i = u_i - u_0 \circ (I + T_i)$?



Linear combinations of transformed states are now physically realistic.

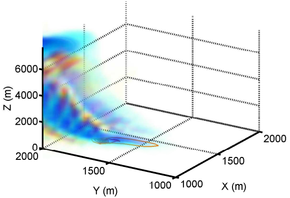


Assimilation of fireline position
into a reaction-diffusion PDE fire model Fireline propagation model by level sets coupled with WRF

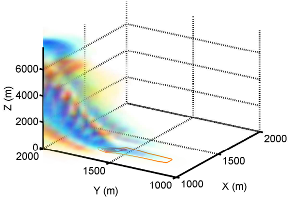


Data | Forecast | Analysis

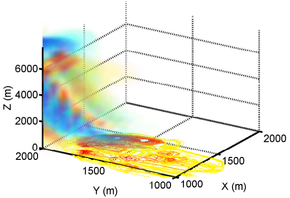
Assimilation of fireline position into a fireline propagation model by level sets, coupled with WRF



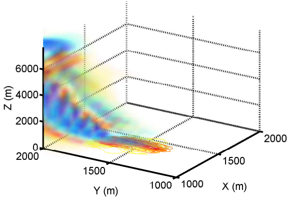
Data source



No assimilation



Standard EnKF



Morphing

Future plans

- Level set-based fire propagation model running continuously (1 month), web-accessible (3 months), distributed with WRF (6 months) (with Janice Coen)
- Morphing EnKF for station observations
- Observation functions for assimilation of realistic data (with Craig Douglas and Tony Vodacek)
- Continuously running web-based fire prediction, assimilating fire perimeters from the web
- Plug-in architecture for multiple models, video game-like visualization (with Chris Johnson and Claudio Silva)