HIGH-DIMENSIONAL DATA ASSIMILATION AND MORPHING ENSEMBLE KALMAN FILTERS WITH APPLICATIONS IN WILDFIRE MODELING

by

Jonathan D. Beezley

Bachelor of Science, University of Nebraska Lincoln, 2001
Master of Science, University of Colorado Denver, 2006

A thesis submitted to the
University of Colorado Denver
in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Applied Mathematics
2009
This thesis for the Doctor of Philosophy degree by Jonathan D. Beezley has been approved by:

Jan Mandel

Janice Coen

Andrew Knyazev

Doug Nychka

Tolya Puhalskii

Date
In this dissertation, we examine practical aspects of incorporating data into high-dimensional and nonlinear models. The Kalman filter is discussed from a theoretical prospective and is shown to solve the Bayesian data assimilation problem under certain linearity and normality conditions. The ensemble Kalman filter (EnKF) and its variants are presented as an approximation to the optimal filtering distributions using statistical sampling that are used even when the assumptions of the Kalman filter are violated. We present a series of numerical experiments that explore the behavior of the EnKF as the model state size becomes large. Finally, we present an application, wildfire modeling, in which the standard EnKF techniques fail. The behavior of the EnKF applied to this highly nonlinear problem inspires the development of two new types of ensemble filters: predictor-corrector and morphing.

Predictor-corrector filters use a proposal ensemble (obtained by some method, called the predictor) with assignment of importance weights to recover the correct statistic of the posterior (the corrector). The proposal ensemble comes from an arbitrary unknown distribution and only needs concentrated coverage of the support of the posterior. The ratio of the prior and the proposal densities for calculating the
importance weights is obtained by density estimation. The prediction created by the ensemble Kalman formulas and correction formed by nonparametric density estimation based on the distance in Sobolev spaces are considered. Numerical experiments show that the new predictor-corrector filters combine the advantages of ensemble Kalman filters and particle filters for highly nonlinear systems and that they are suitable for high dimensional states that are discretizations of solutions of partial differential equations.

The morphing filter combines the EnKF with the ideas of morphing and registration from image processing. This results in filters suitable for nonlinear problems in which the solutions exhibit moving coherent features, such as thin interfaces in wildfire modeling. The ensemble members are represented as the composition of one common state with a spatial transformation, called registration mapping, plus a residual. A fully automatic registration method is used that requires only gridded data, so the features in the model state do not need to be identified by the user. The morphing EnKF operates on a transformed state consisting of the mapping obtained through registration and the residual image. The EnKF is shown to perform far better when applied to this transformed state than to the original model state for a wildfire model containing spatial errors.

This abstract accurately represents the content of the candidate’s thesis. I recommend its publication.

Signed  ________________________________  
                Jan Mandel
This thesis would not have been possible without the support of the many educators who have gone out of their way to help me. In particular, I wish to thank my high school teacher, Mike Smith, and undergraduate advisors, Gordon Woodward and Edward Jones. Mostly, I thank my graduate advisor, Jan Mandel, for his endless patience and encouragement. I would like to thank my sister, Jennifer, for supporting me through the most difficult times in my life and pushing me to apply for graduate school; my parents for their encouragement and financial support; and my sister, Angela, for her companionship. Finally, I thank the staff of the Mathematics department for guiding me through this process and my fellow students for their friendship over the last several years.
## CONTENTS

Figures . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . ix
Table . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . xxv

Chapter

1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 1
2. The Ensemble Kalman Filter . . . . . . . . . . . . . . . . . . . . . . . . 7
   2.1 Preliminaries . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
   2.2 The Kalman Filter . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
   2.3 The Ensemble Kalman Filter . . . . . . . . . . . . . . . . . . . . . . . . 14
   2.4 Variants of the EnKF . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
   2.4.1 Square Root Filters . . . . . . . . . . . . . . . . . . . . . . . . . . . . 17
   2.4.2 Localized Ensemble Filters . . . . . . . . . . . . . . . . . . . . . . . . 18
3. Predictor-Corrector Filter . . . . . . . . . . . . . . . . . . . . . . . . . . 23
   3.1 Preliminaries . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
   3.1.1 Weighted ensembles . . . . . . . . . . . . . . . . . . . . . . . . . . . . 24
   3.1.2 Particle filters . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 26
   3.1.3 Density estimation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
   3.2 Derivation of predictor-corrector filters . . . . . . . . . . . . . . . . . . . 30
   3.2.1 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30
   3.2.2 High-dimensional systems . . . . . . . . . . . . . . . . . . . . . . . . . 32
   3.2.3 Formulation of the method . . . . . . . . . . . . . . . . . . . . . . . . . 34
FIGURES

Figure

2.1 The function described in (2.25) is displayed as a blue line, and Gaussian function with bandwidth $\frac{\sqrt{3}}{10}$, scaled to maximum of 1, is displayed as a dotted black line.

3.1 A simple example showing degeneracy of the particle filter. The SIS is applied to a prior ensemble of size 10 sampled out of $\mathcal{N}(0, 0.16)$ with equal weights. The data likelihood is centered away from the prior mean with distribution $\mathcal{N}(2.5, 0.0625)$. In (a), the exact prior, likelihood, and posterior densities are given by the blue, red, and black lines, respectively. The prior and posterior ensemble members and weights are given by blue $\ast$'s and black $\times$'s. The value of the ensemble members are given by their position on the $x$ axis and the weights are the vertical positions. In addition, the scaled likelihood is given by red $\circ$'s, but because the prior ensemble had equal weights, the likelihood is proportional to the posterior weight. In (b), the posterior ensemble is shown on a semi-log plot. The weight of the largest ensemble member is almost 1, while weights of the rest are essentially 0. In (c), the exact mean and covariance is given along with the computed ensemble mean and covariance from (3.6) and (3.8) for both the prior and posterior. In this case, the particle filter fails to accurately represent these statistics of the posterior.
3.2 The problem shown in Figure 3.1 is repeated with the same prior ensemble but using the predictor-corrector filter. With the new method, the proposal ensemble is much closer to the data, and the posterior ensemble exhibits far less degeneracy and its sample mean and covariance is far closer to the exact mean and covariance compared with the SIS method.

3.3 In (a), the prior and data distributions are given by blue and red lines, respectively. The prior ensemble, with size 10, is displayed with blue \( \star \)'s, where the position of each ensemble member on the \( x \) axis is its value and the position on the \( y \) axis is its weight. The data likelihood given each ensemble member is displayed as red \( \circ \)'s, scaled so the likelihoods sum to 1. The prior distribution has two modes at \( \pm 1.5 \) and is constructed by sampling each ensemble member, \( x_k \sim \mathcal{N}(0, 2.5) \) and assigning weights as the pdf of the sum of two Gaussian distributions, \( \mathcal{N}(\pm 1.5, 0.1) \) evaluated at \( x_k \). The data distribution is \( \mathcal{N}(0, 0.1) \). In (b)-(d), the posterior ensemble is shown with black \( \times \)'s for EnKF, SIS, and EnKF-SIS, respectively. The exact posterior density is shown as a dotted line, while the estimated posterior density from the ensemble calculated from a gaussian kernel with bandwidth 0.5 is shown as a solid black line. The EnKF alone cannot account for the bimodal nature of the posterior, while the EnKF-SIS method corrects this by applying weights which increase the accuracy of the posterior estimation. In this scalar problem, the SIS method is able to perform reasonably well.

3.4 (a) The deterministic potential curve, and (b) A solution of (3.22) switching between stable points.
3.5 In (a), an example of the applying the EnKF, SIS, and EnKF+SIS on (3.22) with the data given by circles. Displayed is the ensemble mean of a single realization and the mean of the optimal solution from (3.23) using an ensemble of size 100. In (b) and (c), the estimated mean square error in the ensemble mean and covariance are given for each method by averaging over 100 simulations. 

3.6 In (a), a non-Gaussian prior distribution is applied to a data distribution with small likelihood. The exact posterior distribution is given in (b). The EnKF, SIS, and EnKF-SIS was applied to an ensemble of size 100. In (c), the EnKF does not approximate the exact posterior distribution well because it ignores the bimodal prior. The SIS given in (d) does approximate the posterior reasonably well because the data likelihood is large. In (e), the EnKF-SIS resembles the features of (d) by reducing the weights of the proposal ensemble members that pass through the center of the figure.

3.7 In (a), a Gaussian prior distribution is applied to a data distribution with small likelihood. The exact posterior distribution is given in (b). The EnKF, SIS, and EnKF-SIS was applied to an ensemble of size 100. In (c), the EnKF approximates the exact solution well because the prior was Gaussian; however, the SIS given in (d) was fairly degenerate and was not able to correct for the large shift in the distribution. In (e), the EnKF-SIS approximates the shape of the exact posterior well, while exhibiting far less degeneracy than (d).
4.1 For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 10 ensemble members and 25 data points with a random observation matrix.

4.2 For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 10 ensemble members and 250 data points with a random observation matrix.

4.3 For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 100 ensemble members and 25 data points with a random observation matrix.

4.4 For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 100 ensemble members and 250 data points with a random observation matrix.

4.5 For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 10 ensemble members and 25 data points with observation matrix, \( H_{ij} = \delta_{ij} \).

4.6 For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 10 ensemble members and 250 data points with observation matrix, \( H_{ij} = \delta_{ij} \).
4.7 For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 100 ensemble members and 25 data points with observation matrix, $H_{ij} = \delta_{ij}$.  

4.8 For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 100 ensemble members and 250 data points with observation matrix, $H_{ij} = \delta_{ij}$.  

4.9 For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 10 ensemble members and an identity observation matrix.  

4.10 For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 100 ensemble members and an identity observation matrix.  

4.11 The effect of creating a smooth random field using the procedure outlined above. An i.i.d. complex vector of length 100 is sampled out of $\mathcal{N}(0, 1)$. The components of this vector are scaled according to 4.3, with $\alpha = 0, 1, 2$. A larger $\alpha$ reduces the influence of higher frequency components, and the resulting function is smoother.
4.12 In (a), the fireline and level set function for a fire initialized by two thin lines and a small circle propagating in constant wind. The fire is in the area where the level set function is negative. The black contour is the fireline. In (b), the heat flux generated by the fire in (a). 

4.13 From Mandel et al. [2008], the temperature profile of a combustion wave traveling over a fixed point. The modeled temperature profile given by the black line was optimized to approximate the dotted line, representing real data obtained from Kremens et al. [2003].

4.14 The estimated point-wise probability density of the temperature near the fire line for an ensemble generated by perturbing a known temperature profile in space by a Gaussian random variable. The density has peaks centered around burning and non-burning regions and is highly non-Gaussian.

4.15 In (a), a forecast ensemble of size 10 containing temperature profiles shifted in space with a variance of 200 m. The data profile given as a dotted red line is located in the center of the ensemble members. In (b), the point-wise ensemble variance calculated from this ensemble.

4.16 The analysis ensemble resulted in applying the EnKF to the forecast ensemble and data in Figure 4.15. The data variance used was 100 K^2. The EnKF was unable to track the data accurately, and the analysis ensemble has essential degenerated to a single solution, with variance reduced by several orders of magnitude.
4.17 The analysis ensemble resulted in applying the $B$-localized EnKF to the forecast ensemble and data in Figure 4.15. The data variance used was $100\, K^2$ and localization radius was $20\, m$. The localized EnKF was able to create a posterior ensemble which tracks the data accurately on the reaction front; however, there are many spurious features outside of this region. In addition, the localization has reduced the degeneracy of the analysis ensemble compared to the EnKF without localization.

4.18 In (a), a forecast ensemble of size $10$ containing temperature profiles shifted in space with a variance of $200\, m$. The data profile given as a dotted red line is located $400\, m$ to the left of the center of the ensemble members. In (b), the point-wise ensemble variance calculated from this ensemble.

4.19 The analysis ensemble resulted in applying the EnKF to the forecast ensemble and data in Figure 4.18. The data variance used was $100\, K^2$. In this case, the analysis ensemble does not resemble the data at all because the forecast ensemble contained no members nearby.

4.20 The analysis ensemble resulted in applying the $B$-localized EnKF to the forecast ensemble and data in Figure 4.18. The data variance used was $100\, K^2$ and localization radius was $20\, m$. The results here are similar to that of the non-localized EnKF; the analysis ensemble was unable to track the data because no forecast ensemble members were nearby.

5.1 Shown in (a) and (b) are contour plots of the $x$ and $y$ components of an example morphing function, $T$. Figure (c) shows a quiver plot of $T$ indicating how an image would be warped when $T$ is applied to it.
5.2 Shown in (a) is an image $u$ made up of concentric circles. The morphing function given in Fig. 5.1 applied to this image, $u \circ (I + T)$. The resulting warped image is given in (b).

5.3 The example given in Remark 5.1 is illustrated for $\varphi(x) = u(x)$, a hat function centered at 0, displayed in blue, and $v(x) = 2\varphi(x + 1)$ is displayed in red. In (a) and (b), intermediate images are formed following (5.10), which uses the inverse morphing function, and (5.11), which is commonly used in applications, respectively, are displayed in black. The common method produces two features: $\varphi(x + \lambda)$, which is a translation of $u$ centered at $-\lambda$, and $\lambda \varphi(x + 1)$, which grows with $\lambda$, but remains centered at $-1$. The method requiring the inverse morphing function produces a single peak that translates and grows simultaneously. This is a result of the form of the residual in (5.5) which moves in conjunction with $u$. 
5.4 An example of the intermediate states of the morphing transformation applied to the heat flux emitted from two different fires. The morphing transformation is taken for both of the images and linear combinations of the two are formed, representing intermediate states in the morphing representation. The resulting states are transformed back into the standard representation. The two alternative formulations for the residual image yield different intermediate states. In the given figures, the top and bottom images are the two original fire lines. The intermediate states for the standard form of the residual (5.8) are given on the right and those of our representation (5.5) are given on the left. The standard method produces spurious features that remain centered on the location of the bottom fire. In our method, the large components of the residual move with the morphed image, so no spurious features are present.

5.5 The physical domain (a) with grid resolutions $dx$ and $dy$ is scaled to the normalized domain (b) on a square grid, $D = [0, 1] \times [0, 1]$, with grid resolutions $dx$ and $dy$. The change in aspect ratio from $dx:dy$ to $dx:dy$ must be accounted for in the algorithm using the parameter $\rho = \frac{dx}{dy}$, which represents the anisotropic scaling of each axis.

5.6 The original domain $D$ is split into 9 different sub-domains of size $\frac{1}{2} \times \frac{1}{2}$ at refinement level $\ell = 1$. The sub-domains at this level are indexed by $D_{i,j}^1$ with $i, j \in \{1, \frac{3}{2}, 2\}$. Fractional indices indicate that the sub-domain is offset from the global domain’s edge by a distance of half of the sub-domain’s width in its respective axis. Further refinement levels are decomposed similarly.
5.7 Displayed in (a) is a function defined over $[0, 1]$ containing a thin feature similar to the reaction front of a wildfire. This function is smoothed using equation 5.18 with various smoothing bandwidths which would be used at increasing refinement levels, $\ell$, where the bandwidth decreases as $2^{-\ell}$. The larger bandwidth at lower levels increases the ability of the registration algorithm to find the global minimum of the objective function. Decreasing the bandwidth on the same scale of the refinement enables the algorithm to retain accuracy as the algorithm attempts to align increasingly fine scale detail. 

5.8 (a) The morphing shape function $S(x)$ used in (5.19) defined as a cubic spline with $S(x) = 0$ and $S'(x) = 0$ on the boundary and $S(x) = 1$ and $S'(x) = 0$ at the center, $x = 0$. (b) The tensor product of the shape functions that make up the components of $B_1$ and $B_2$. The gradient of this tensor product is zero along the boundary, and it attains a maximal value of 1 at the center, $(0, 0)$. (c) and (d) Quiver plots showing the deformation of the domain performed by each shape function scaled by 0.2.
5.9 These figures illustrate the effect of various methods of interpolating a morphing function onto the pixel grid. (a) shows the original image with four arrows indicating the specified perturbations of the image. The remaining images show the morphed image with the specified perturbations interpolated to the pixel grid with (b) nearest neighbor, (c) bilinear, and (d) tensor product B-spline interpolants. Figure (b) exhibits clear discontinuities, while Figure (c) creates kinks in what were originally smooth curves.

5.10 (a) The derivative of $S(x)$ from Fig. 5.8a has maximal and minimal values at $-0.5$ and $0.5$. The partial derivatives of Fig. 5.8b in the $x$ and $y$ directions are given in (b) and (c), respectively. These partial derivatives have extreme values at $(\pm 0.5, 0)$ and $(0, \pm 0.5)$, where they attain slopes of $\pm 1.5$. Equation (5.16) is used to provide approximate conditions for the invertibility of $I + T + c_1 P^\ell_{ij} B_1$ by finding bounds for $c_1$ such that

$$\frac{\partial}{\partial x} \left( I + T + c_1 P^\ell_{ij} B_1 \right) > -1 \quad \text{at} \quad (\pm 0.5, 0).$$

The resulting bounds give rise to the conditions appearing in (5.25).
5.11 The black dots represent the sampling grid defined in (5.27) as a uniform grid with resolution $\kappa_x \times \kappa_y$. The region outlined in red, $C$, is defined as the feasibility region of the local minimization problem (5.24). The points marked with a black square are determined by solving (5.26) and represent the intersection of $C$ with the $x$ and $y$ axes. The distance of these points from the origin are multiplied by the contraction parameters, $\alpha_x$ and $\alpha_y$, to obtain the blue dots located at $(c_x, 0)$, $(0, c_y)$, $(c_x, 0)$, and $(0, c_y)$. The coordinates for these dots are explicitly given in (5.25) and define the vertices of the quadrilateral, $\tilde{C}$, outlined in blue. The sampling points, $S$, circled in blue are determined as the intersection of the sampling grid with the region $\tilde{C}$. The dashed blue line represents what the sampling region would be if we did not include the contraction parameters. This region is not a subset of $C$, and, in particular, contains the sampling point circled in red, which is not in the feasible region of the problem.

6.1 Data assimilation by the morphing ensemble filter. The forecast ensemble (b) was created by smooth random morphing of the initial temperature profile located near the center of the domain. The analysis ensemble (c) was obtained by the EnKF applied to the transformed state. The data for the EnKF was the morphing transformation of the simulated data (a), and the observation function was the identity mapping. Contours are at $800 \, K$, indicating the location of the fireline. The reaction zone is approximately between the two curves.
6.2 After five analysis cycles, the ensemble shows less spread and follows the data reliably. Ensemble members were registered using the initial state, advanced in time without data assimilation. The forecast ensemble (b) is closer to the simulated data (a) because of preceding analysis steps that have attracted the ensemble to the truth. The analysis ensemble (c) has a little less spread than the forecast, and the change between the forecast and the analysis is well within the ensemble spread. Contours are at 800 $K$, indicating the location of the fireline. The reaction zone is approximately between the two curves.

6.3 Probability densities estimated by a Gaussian kernel with bandwidths 37 $K$, 19 $K$, and 30 $m$. Data was collected from the ensemble shown in Fig. 6.1b. Displayed are typical marginal probability densities at a point near the reaction area of the original temperature (a), the registration residual of temperature (b), and the registration mapping component in the $x$ direction (c). The transformation has vastly improved Gaussian nature of the densities.

6.4 The $p$–value from the Anderson-Darling test of the data from the ensemble after five morphing EnKF analysis cycles shows the ensemble transformed into its registration representation, the registration residual of the temperature (b) and the registration mapping (c), has distribution much closer to Gaussian than the original ensemble (a). The shading at a point indicates where the marginal distribution of the ensemble at that point is highly Gaussian (white) or highly non-Gaussian (black).
6.5 The morphing EnKF applied to the reaction-diffusion model. The false color is generated from the temperature with shading for depth perception. The reference solution (a) is the simulated data. The initial ensemble is created by a random perturbation of the comparison solution (b), where the fire is ignited at an intentionally incorrect location. The standard ENKF panel (c) is the result of data assimilation of the temperature field after running the model for 500 seconds. The morphing EnKF panel (d) is the ensemble with the image registration against the temperature field at the time of ignition, and the morphing applied to both the temperature and the fuel supply. The ensembles have 25 members each, and are visualized as the superposition of transparent images of their members. The observation function is the temperature field on the whole grid. The standard EnKF ensemble diverges from the data, while the morphing EnKF ensemble remains closer to the data.
6.6 The morphing EnKF applied to the fireline propagation model. The false color is the output heat flux with shading for depth perception. The reference solution (a) is the simulated data. The initial ensemble is created by a random perturbation of the comparison solution (b), where the fire is ignited at an intentionally incorrect location. The standard ENKF panel (c) is the result of data assimilation of the time from ignition after running the model for 1000 seconds. The morphing EnKF panel (d) is the result with the image registration determined from the time from ignition and the morphing applied to all of the model variables. The ensembles have 25 members each, and are visualized as the superposition of transparent images of heat fluxes of their members. The registration is done on the atmospheric grid with the fire heat flux as the observation function, but the atmospheric model is not used. The standard EnKF ensemble diverges from the data, while the morphing EnKF ensemble remains closer to the data.
The morphing EnKF applied to the fireline propagation model coupled with WRF. The false color of the contour on the horizontal plane is the fire output heat flux. The superposition of transparent ensemble members is shown. The volume shading is the vorticity of the atmosphere in the ensemble mean, where red and blue shades represent positive and negative vorticity, respectively. The reference solution (a) is the simulated data. The initial ensemble is created by a random perturbation of the comparison solution (b), where the fire is ignited at an intentionally incorrect location. The standard ENKF (c) and the morphing EnKF (d) are applied after 15 minutes. The ensembles have 25 members each. The registration is done on the atmospheric grid with the fire heat flux as the observation function. The standard EnKF ensemble diverges from the data, while the morphing EnKF ensemble remains closer to the data.
Table 3.1 Data used in assimilation into (3.22) ........................................ 41

6.1 Analysis ensemble statistics for the EnKF with and without morphing. An ensemble of size 25 is generated of a 2-D fire shifted in space with mean 0 and standard deviation, $\sigma$. Four data points are tested with standard deviation $75\, m$. The exact analysis mean is given next to the sample mean resulting from the EnKF and the morphing EnKF, as well as the relative analysis standard deviation (6.4). The ensemble statistics were averaged over 100 simulations. While both methods exhibit large errors, the results indicate that the EnKF is unable to move the location of the fire, while the morphing EnKF can. Also, the EnKF alone produces far more degeneracy in the analysis ensemble compared to the morphing EnKF. ................................................................. 119
1. Introduction

Modelling physical systems with computational simulations has become a ubiquitous part of modern science. The purpose of the simulation can be to predict the outcome of a future event or to gain insight in the details of a prior event. The model itself can be described abstractly as a function that maps a vector of physical quantities (the model state) from some initial time into a forecasted state at some future time. The model is often given as the solution of a partial differential equation (PDE), but can also be an empirical algorithm, or a combination of the two. Usually, the model will depend on a set of parameters, such as the coefficients of the PDE. A simple model could describe the position of ball rolling down an incline, with parameters containing the mass of the ball, gravitational acceleration, and the coefficient of friction.

Clearly, we wish to create numerical models which forecast real life phenomena exactly; however, this is generally impossible. Often, either the precise mechanics of the problem are unknown or we lack the computational resources to model them. Neither the parameters of the model nor the initial conditions are known exactly. In addition, floating point error prevents any sort of numerical scheme from exactly solving the model equation. All of these factors combined create intrinsic errors in the forecasted solution to the system. Many nonlinear systems, such as weather models, exhibit chaotic effects where these errors grow exponentially over time. It is, therefore, essential to estimate these errors and their behavior when making numerical
forecasts of physical systems. These errors will be modelled in a statistical framework with the mathematical notation introduced in Section 2.1.

The model errors present in computational simulations can often be reduced by the inclusion of data retrieved from the physical system. The instruments that collect this data have their own associated error. We generally assume the data error from these instruments is either known or can be estimated somehow. There should also be a known relationship between the data and the model state vector. This relationship is given in terms of a function known as the observation function, which acts on the model state to produce synthetic data. For each realization of the model state, the synthetic data represents what the data would be if the model state represented the “truth”. While methods of including continuous time data sets into a numerical model exist (such as the Kalman-Bucy filter [Kalman and Bucy, 1961]), we will limit our discussion to problems involving discrete data sets.

The process of including real data in numerical models is known as sequential statistical estimation or, in meteorology and oceanography, as data assimilation [Bennett, 1992, p. 67]. Data assimilation aims to find the best estimate from the model state and the data by balancing the uncertainties. This problem is often posed in one of two ways. Variational methods, such as 3DVAR and 4DVAR, construct least-square estimates using two norms weighted by the inverse covariances. The square error produced by the deviation from the original model state and data is minimized using an iterative Quasi-Newton method. Other data assimilation methods such as the ensemble Kalman and particle filters are posed as statistical estimation of the filtering distribution defined by the model state conditioned on the data. We show in Section 2.2 that the Kalman filter results in the exact filtering distribution when the
model is linear and the error distributions are Gaussian. In this case, the mean of the filtering distribution can be shown to be the least-square solution obtained from the variational methods, [Kalnay, 2003]. The Kalman filter works by explicitly updating the covariance of the model state, which is usuitable for large problems. In Section 2.3, we present the ensemble Kalman filter (EnKF) and several variants. These filters are Monte-Carlo methods that evolve in time a collection of simulations, called an ensemble, and replace the exact covariance by the sample covariance computed from the ensemble. In Section 4.3, we present a formulation of this method that can be computed efficiently on large problems.

While the methods based on the Kalman filter are, theoretically, only justified for linear models and Gaussian distributions, they are often used when these assumptions are violated. However, when the model is nonlinear, the distributions can become highly non-Gaussian. Kalman filter based methods represent the errors exclusively from their covariances. While Gaussian distributions are given uniquely by their mean and covariance, this is not true for general distributions. The higher order characteristics are implicitly ignored by the Kalman filter, which can lead to significant biases in the filtered distribution. In Section 3, we introduce particle filters, which implement the data assimilation without relying on the Gaussian assumption, but the required ensemble size grows quickly with the dimension of the state space for many problems [Bengtsson et al., 2008, Snyder et al., 2008].

Existing data assimilation techniques are incapable of producing acceptable estimations of filtering distributions for high-dimensional and nonlinear models. Thus, the design of more efficient filters for these problems has been the subject of significant recent interest. We will discuss two extensions to standard techniques in data
assimilation, which attempt to improve prediction capability when applied to such models. The development of these extensions will be motivated by numerical experiments intended to explore the problems encountered. Finally, we provide numerical evidence that the new methods improve the estimation of the filtering distributions.

This thesis is organized as follows. In chapter 2, we will derive the Kalman filter directly from Bayes’ rule under a set of assumptions based on a linear model. Next, the ensemble Kalman filter (EnKF) will be introduced as an approximation to the Kalman filter by replacing the distribution’s covariance with a sample covariance. Finally, several variants of the EnKF commonly used in practice will be discussed. In chapter 3, a hybrid filter will be presented that uses density estimation to combine EnKF and particle filter to retain the advantages of both. This method is based on the technical report [Mandel and Beezley, 2006] and the extended abstract [Mandel and Beezley, January 2007]. In chapter 4, we will discuss practical issues of the EnKF including generating initial ensembles and efficient implementation. A series of numerical experiments will be performed which are designed to explore the behavior of the filtering error for both the EnKF and the particle filter as the size of the model increases. Finally, a model for prediction of wildfire growth, based on the paper Mandel et al. [2009a], will be discussed along with the complications that arise when applying the EnKF to it. The EnKF alone will be shown to be insufficient when applied to this highly nonlinear model due to the natural evolution of errors in the location of sharp reaction fronts. This will motivate the development of a nonlinear transformation of the model state called the morphing transformation based on the paper, Beezley and Mandel [2008]. This transformation relies on a nonlinear optimization problem called image registration which attempts to align coherent features
of images by composition with spatial deformations known as morphing functions.
In chapter 5, we will present an efficient, automated procedure capable of registering images originating from fire modeling. In chapter 6, we will define the morphing transformation which will be applied to model states prior to their application to the EnKF. This morphing EnKF is an extension to the traditional algorithm that transforms the original state variables to a form intended to separate spatial errors from intensity errors. The transformed model states are shown to be much closer to Gaussian, and when applied to the EnKF, produce far superior results in comparison to the standard approach.

My personal contribution to the work described in this thesis is as follows. In chapter 2: the proofs of Lemma 2.1 and Theorem 2.4 along with the descriptions of the square root and localized filters. In chapter 3: the Matlab implementation and numerical results dealing with bimodal prior densities in Figures 3.1-3.5. In chapter 4: a Matlab implementation of the standard and localized EnKF, which were used to produce the Figures 4.1-4.10. In addition, I was a contributor to a joint project creating a parallel Fortran implementation of the EnKF, which has been used to collect numerical results for large problems on parallel supercomputers. My contributions to this project include: parallelizing the implementation, maintaining the main driver as model and data interfaces were revised, and programming subroutines capable of importing model states from binary NetCDF files. I also assisted in the development of the integrated weather/wildfire model used for the numerical results in this thesis. My contributions included: integrating the fire code into NCAR’s weather model (WRF), maintaining the software infrastructure, porting the changes to the development version of WRF, and modifying the WRF data preprocessor (WPS) to support refined
grids necessary for the inclusion of high resolution fuel data. These modifications to the standard WRF code have been merged with the development repository and are scheduled to be included in a future release cycle. In chapter 5: the design of the automated image registration algorithm and an efficient Fortran implementation, which was used in the relevant numerical results. In chapter 6: the design of the morphing transformation and its Fortran implementation, with a contribution from Jan Mandel, providing for non-gridded interpolation used for the inversion of morphing functions.
2. The Ensemble Kalman Filter

2.1 Preliminaries

We will deal with model states which at time $t$ are represented as real vectors of size $n$, $x_t \in \mathbb{R}^n$. These vectors are evolved in time using a model equation, which will be represented as the function $M : (x_t, t, \delta t) \mapsto x_{t+\delta t}$. We will assume that data is available at discrete time steps $t_1, \cdots, t_K$, which we will denote as $d_{t_i}$, a real vector of size $m_{t_i}$. Similarly, we assume observation function for each data vector is known and denote it by $h_{t_i}$. When we are discussing only a single assimilation cycle, we will often drop the subscripts $t_i$, and the data will be given as $d \in \mathbb{R}^m$ with observation function $h$.

The model and data vectors will be assumed to be random vectors whose distributions have a probability density and bounded second moment. In this context, the data assimilation is posed in terms of Bayes rule,

$$p(x_a^t) = p(x_f^t | d) \propto p(d_t | x_f^t) p(x_f^t),$$

where $\propto$ means proportionality. The probability distribution of the model state at a fixed time, called the prior or forecast distribution $(x_f^t)$, is combined with the probability distribution of the data error as a function of the model state, called data likelihood $(d_t | x_f^t)$, to produce the analysis or posterior $(x_a^t)$. Eq. (2.1) determines the posterior density completely because $\int_{\Omega} p(x_a^t) \, d\lambda(\omega) = 1$. The model is then advanced in time starting from the analysis solution to the next available data set, and the so-called analysis cycle repeats. The data assimilation problem is then to estimate the model
state at some time in the future, \( t_K \), using all data available in the time horizon of the simulation, \( d_{t_1}, \ldots, d_{t_{K-1}} \), with \( t_0 \leq t_1 < \cdots < t_{K-1} \leq t_K \). The optimal solution to this problem is the random variable \( x_{t_K} \) given \( x_{t_0}, \ldots, x_{t_{K-1}} \), and \( d_{t_1}, \ldots, d_{t_{K-1}} \). If the model satisfies the Markov equality,

\[
p(x_{t_i} | x_{t_0}, \ldots, x_{t_{i-1}}) = p(x_{t_i} | x_{t_{i-1}}),
\]

then using Bayes rule and the independence of model and data errors, we obtain the standard sequential estimation formula,

\[
p(x_{t_i} | d_{t_1}, \ldots, d_{t_{i-1}}) \propto \int p(x_{t_i} | x_{t_{i-1}}) p(x_{t_{i-1}} | d_{t_1}, \ldots, d_{t_{i-1}}) \, dx_{t_{i-1}}
\]

\[
p(x_{t_i} | d_{t_1}, \ldots, d_{t_{i-1}}) \propto p(d_{t_i} | x_{t_i}) p(x_{t_i} | d_{t_1}, \ldots, d_{t_{i-1}}).
\]

The first part of this formula is the model update or forecasting phase; the second is the data assimilation or analysis phase. In the case of a deterministic and continuous model, the Markov equality is trivial and reduces to

\[
p(x_{t_i} | x_{t_0}, \ldots, x_{t_{i-1}}) = p(x_{t_i}).
\]

### 2.2 The Kalman Filter

First described by Kalman [1960] and Kalman and Bucy [1961], the Kalman filter is a simple recursive formula that implements the sequential estimation formula (2.3) when the initial model state and data distributions are independent and Gaussian and the model is linear. When these conditions are satisfied, all forecast and analysis distributions remain Gaussian over a finite number of assimilation cycles, and the distributions can be represented uniquely by their means and covariances. The Kalman filter formula operates directly on the mean and covariance of the model state to produce the exact filtering distribution. For completeness, major points in the development of the Kalman filter are derived here.
We will assume that the model operator is linear and deterministic within each analysis cycle, but may change from one cycle to the next. Under this assumption, we will represent the model propagation operator for each assimilation cycle, in the following form,

\[ \mathcal{M}(x_{t_i}, t_i, t_{i+1} - t_i) = M_{t_i} x_{t_i} + b_{t_i} = x_{t_{i+1}}, \]

where \( M_{t_i} \) is an \( n \times n \) matrix and \( b_{t_i} \) is an \( n \) dimensional vector. We will also restrict the observation function to be a linear operator,

\[ h_{t_i}(x) = H_{t_i} x + h_{t_i}^0, \]

where the observation being assimilated has Gaussian error density,

\[ H_{t_i} x_{t_i} - d_{t_i} \sim \mathcal{N}(0, R_{t_i}) \quad (2.4) \]

for some matrix \( H \). In this case, the data likelihood is

\[ p(d_{t_i} | x_{t_i}) \propto \exp \left( -\frac{1}{2} (H_{t_i} x_{t_i} - d_{t_i}) R_{t_i}^{-1} (H_{t_i} x_{t_i} - d_{t_i}) \right). \]

When we are dealing only with a single assimilation cycle, the subscript \( t_i \) will be dropped for brevity; however, it is to be understood that the model and observation function can change from one cycle to the next. This is desirable because in most applications these operators will not be linear, but can be approximated as such. In this case, the matrix \( M_{t_i} \) is the Jacobian of the model evaluated at time \( t_i \), and the vector \( b_{t_i} \) is the model evaluated at zero, \( \mathcal{M}(0, t_i, t_{i+1} - t_i) \), as in a first order Taylor approximation. The observation function is approximated in a similar fashion. The matrix \( M_{t_i} \) is called the tangent linear model, and the Kalman filter applied to the approximated model is called the Extended Kalman filter.
Assume that at a given analysis cycle, the forecast has normal distribution with mean, \( E(x_{t_i}^f) = \mu_{x_{t_i}} \), and non-singular covariance, \( Q_{t_i}^f \), and the data likelihood is normal with mean \( H_t x_{t_i}^f \) and non-singular covariance, \( R_{t_i} \),

\[
p(x_{t_i}^f) \propto \exp \left( -\frac{1}{2} (x_{t_i}^f - \mu_{x_{t_i}})^T \left( Q_{t_i}^f \right)^{-1} (x_{t_i}^f - \mu_{x_{t_i}}) \right), \tag{2.5}
\]

\[
p \left( d_{t_i} | x_{t_i}^f \right) \propto \exp \left( -\frac{1}{2} (d - H x_{t_i}^f)^T R_{t_i}^{-1} (d - H x_{t_i}^f) \right), \tag{2.6}
\]

**Lemma 2.1.** Given normal forecast (2.5) and likelihood distributions (2.6), the analysis distribution given by Bayes theorem (2.1) is also normal. Further, the mean and covariance of the analysis distribution is

\[
\mu_{x_a} = \mu_{x_f} + K (d - H \mu_{x_f}), \quad Q_a = (I - KH) Q_f, \tag{2.7}
\]

where

\[
K = Q_f H^T (HQ_f H^T + R)^{-1} \tag{2.8}
\]

is called the Kalman gain matrix.

**Proof.** Taking the logarithm of Bayes rule (2.1) gives us the following expression for the posterior density

\[
-2 \log \left( p \left( x^f | d \right) \right) = (x^f - \mu_{x_f})^T (Q_f)^{-1} (x^f - \mu_{x_f})
+ (d - H x^f)^T R^{-1} (d - H x^f) + \text{const}
= x^{fT} \tilde{Q} x^f - 2x^{fT} \tilde{q} + \text{const} \tag{2.9}
\]

where const is some constant which does not depend on \( x_f \),

\[
\tilde{Q} = (Q_f)^{-1} + H^T R^{-1} H, \quad \text{and} \quad \tilde{q} = (Q_f)^{-1} \mu_{x_f} + H^T R^{-1} d.
\]
Because $\tilde{Q}$ is symmetric and positive definite, we can complete the square in (2.9) and conclude that the posterior distribution is Gaussian with some covariance, $Q^a$, and mean, $\mu_x^a$. The analysis density then has the following form

$$-2 \log (p(x^f|d)) = (x^f - \mu_x^a)^T (Q^a)^{-1} (x^f - \mu_x^a) + \text{const}$$

$$= x^f \! T (Q^a)^{-1} x^f - 2 x^f \! T (Q^a)^{-1} \mu_x^a + \text{const}. \quad (2.10)$$

Comparing the quadratic term in (2.10) with that in (2.9) yields

$$Q^a = \left((Q^f)^{-1} + H^T R^{-1} H\right)^{-1}.$$

By the Sherman-Morrison-Woodbury formula [Hager, 1989], $Q^a$ can be written as

$$Q^a = Q^f - Q^f H^T (HQ^f H^T + R)^{-1} HQ^f$$

$$= \left(I - Q^f H^T (HQ^f H^T + R)^{-1} H\right) Q^f$$

$$= (I - KH)Q^f.$$  

Finally, comparing the linear terms, we have

$$\mu_x^a = Q^a \left( ((Q^f)^{-1} \mu_x^f + H^T R^{-1} d) \right)$$

$$= (I - KH)\mu_x^f + (I - KH)Q^f H^T R^{-1} d$$

$$= (I - KH)\mu_x^f + \left(I - Q^f H^T (HQ^f H^T + R)^{-1} H\right) Q^f H^T R^{-1} d$$

$$= (I - KH)\mu_x^f +$$

$$Q^f H^T \left( HQ^f H^T + R \right)^{-1} \left((HQ^f H^T + R) - HQ^f H^T \right) R^{-1} d$$

$$= (I - KH)\mu_x^f + Q^f H^T \left( HQ^f H^T + R \right)^{-1} d$$

$$= (I - KH)\mu_x^f + Kd$$

$$= \mu_x^f + K (d - H\mu_x^f).$$
The following Lemma interprets the Kalman filter as a least-square method as interpreted by 3DVAR and shows that, under these assumptions, the mean of the analysis distributions produced by the Kalman filter is the same as the least-square estimate produced by 3DVAR.

**Lemma 2.2.** If $\mu_{xa}$ is defined by (2.7) and (2.8), then $\mu_{xa}$ is the solution of the least-squares problem

$$J(x) = (x - \mu_{xf})^T(Q^f)^{-1}(x - \mu_{xf}) + (d - Hx)^TR^{-1}(d - Hx) \to \min_x.$$  \hspace{1cm} (2.11)

**Proof.** At the minimum,

$$\nabla J(x) = 2(Q^f)^{-1}(x - \mu_{xf}) - 2H^TR^{-1}(d - Hx) = 0,$$

which gives

$$x = \tilde{Q}((Q^f)^{-1}\mu_{xf} + H^TR^{-1}d),$$

where

$$\tilde{Q} = (((Q^f)^{-1} + H^TR^{-1}H)^{-1},$$

$$= Q^f - Q^fH^T(HQ^fH^T + R)^{-1}HQ^f,$$

$$= (I - KH)Q^f = Q^a.$$

Consequently,

$$x = Q^a((Q^f)^{-1}\mu_{xf} + H^TR^{-1}d) = \mu_{xa}.$$  \hspace{1cm} \blacksquare$$

Because the posterior density is proportional to $\exp \left(-\frac{1}{2}J(x)\right)$, the solution $x = \mu_{xa}$, which minimizes the least-square estimate also maximizes the analysis density and is called the Bayesian estimate of the filtering problem.

The Kalman filter proceeds by computing the forecast distribution for the next Bayesian update at time $t_{i+1}$ using the following lemma, which is an elementary consequence of the linearity of $E(\cdot)$. 

12
Lemma 2.3. Assume \( x \) is a normal random variable with mean, \( \mu_x \), and covariance, \( Q_x \). If \( M \) and \( b \) are deterministic and of the appropriate size, then the random variable \( y = Mx + b \) is also normal with mean, \( \mu_y = M\mu_x + b \), and covariance, \( Q_y = M^TQ_xM \).

The Kalman filter can then be expressed as the following two step recursion:

\[
\begin{align*}
\mu_{x|t_i}^a &= \mu_{x|t_i}^f + K_{t_i} \left( d_{t_i} - H_{t_i}\mu_{x|t_i}^f \right) \quad \text{analysis step; (2.12)} \\
Q_{t_i}^a &= (I - K_{t_i}H_{t_i}) Q_{t_i}^f \\
\mu_{x|t_i+1}^f &= M_{t_i} \mu_{x|t_i}^a + b_{t_i} \quad \text{forecast step. (2.13)} \\
Q_{t_i+1}^f &= M_{t_i}^T Q_{t_i}^a M_{t_i}
\end{align*}
\]

Equations (2.12) and (2.13) can be seen as solving the sequential data assimilation problem from (2.3) for the special case involving step-wise linear model and observation functions, and Gaussian distributions. This is a consequence of the following theorem, which gives the optimal Bayesian solution to this problem.

Theorem 2.4. At time \( t_i \), \( \mu_{x|t_i}^a \), as defined in (2.12), maximizes the conditional probability density \( p(x_{t_i}^f | d_{t_1}, \ldots, d_{t_i}) \).

Proof. By induction, we have, at time \( t_{i-1} \), \( \mu_{t_{i-1}} \) maximizes the density \( p(x_{t_{i-1}}^f | d_{t_1}, \ldots, d_{t_{i-1}}) \). By Lemma 2.3, the density of \( p(x_{t_{i-1}}^f | d_{t_1}, \ldots, d_{t_{i-1}}) \) is Gaussian with mean \( \mu_{x|t_{i-1}}^f = M_{t_{i-1}} \mu_{x|t_{i-1}}^a + b \), and covariance, \( Q_{x|t_{i-1}}^f = M_{t_{i-1}}^T Q_{x|t_{i-1}}^a M_{t_{i-1}} \). By Lemma 2.2, \( \mu_{x|t_{i-1}}^a \) maximizes the density \( p(x_{t_{i-1}}^f | d_{t_{i-1}}) \). But \( x_{t_{i-1}}^f \) is a continuous function of \( x_{t_{i-1}}^f | d_{t_1}, \ldots, d_{t_{i-1}} \), therefore it is measurable with respect to \( d_{t_1}, \ldots, d_{t_{i-1}} \) and \( p(x_{t_{i-1}}^f | d_{t_{i-1}}) = p(x_{t_{i-1}}^f | d_{t_1}, \ldots, d_{t_{i-1}}) \). \( \square \)
2.3 The Ensemble Kalman Filter

While the Kalman filter provides optimal algebraic formulas for the change of the mean and covariance by the Bayesian update and a formula for advancing the covariance matrix in time provided the system is linear, it is not possible computationally for high-dimensional systems. For this reason, the EnKF was developed in Evensen [1994], Houtekamer and Mitchell [1998]. The EnKF represents the distribution of the system state using an ensemble of simulations, and replaces the covariance matrix by the sample covariance matrix of the ensemble. One advantage of the EnKF is that advancing the probability distribution in time is achieved by simply advancing each member of the ensemble. The EnKF, however, still relies on the Gaussian assumption, though it can still be used in practice for nonlinear problems, where the Gaussian assumption is not satisfied. Related filters attempting to relax the Gaussian assumption in EnKF include Anderson and Anderson [1999], Beezley and Mandel [2008], Bengtsson et al. [2003], Mandel and Beezley [2006], van Leeuwen [2003].

We use the EnKF following Burgers et al. [1998] and Evensen [2003], with only some minor differences. This filter involves randomization of data. For filters without randomization of data, see Anderson [1999], Evensen [2004], and Tippett et al. [2003]. The data assimilation uses a collection of independent simulations, called an ensemble. The ensemble filter consists of

1. generating an initial ensemble by random perturbations, by taking some forecasted solution and adding to it, for each ensemble member, independent gaussian random vectors
2. advancing each ensemble member in time until the time of the data, which gives the so-called forecast ensemble

3. modifying the ensemble members by injecting the data (the analysis step), which results in the so-called analysis ensemble

4. continuing with step 2 to advance the ensemble in time again.

We now consider the analysis step in more detail. We have the forecast ensemble

\[
X^f = [x^f_1, \ldots, x^f_N] = [x^f_i]
\]

where each \(x^f_i\) is a random, column vector of dimension \(n\), which contains the whole model state. Thus, \(X^f\) is a matrix of dimension \(n \times N\). As in the Kalman filter, the data is given as a measurement vector \(d\) of dimension \(m\) and data error covariance matrix \(R\) of dimension \(m \times m\), with a corresponding \(m \times n\) observation matrix, \(H\).

The forecast ensemble \(X^f\) is considered a sample from the prior distribution \(p(x)\), and the EnKF strives to create an analysis ensemble that is a sample from the posterior distribution \(p(x|d)\).

The EnKF is based on applying a version of the Kalman update (2.12) to each forecast ensemble member \(x^f_i\) to yield the analysis ensemble member \(x^a_i\). For this update, the data vector \(d\) in (2.12) is replaced by a randomly perturbed vector with distribution equal to that of the data error,

\[
d_j = d + v_j, \quad v_j \sim N(0, R),
\]

where each \(v_j\) is independent of both each other and of the forecast ensemble. These data vectors are combined into a perturbed data matrix, \(D = [d + v_1, \ldots, d + v_N]\).
This is done to ensure proper covariance statistics of the analysis ensemble. Let $X_f$ be the sample, or ensemble, mean of the forecast ensemble,

$$X_f = \frac{1}{N} \sum_{i=1}^{N} x_f^i.$$  \hfill (2.15)

The unknown covariance matrix $Q_f$ in (2.8) is replaced by the sample covariance matrix $C(X_f)$ of the forecast ensemble $X_f$, which we will define as

$$C(X_f) = (X_f) (X_f)^T - (X_f) (X_f)^T.$$  \hfill (2.16)

Note that $X_f$ and $C(X_f)$ are random quantities as opposed to the deterministic mean and covariance used in the Kalman filter. Define

$$X^a = [x^a_1, \ldots, x^a_N] = [x^a_i]$$

as the analysis ensemble. This gives the EnKF formula,

$$X^a = X_f + C(X_f) H^T (HC(X_f) H^T + R)^{-1} (D - H X_f).$$  \hfill (2.17)

Since $R$ is in practice positive definite, there is no difficulty with the inverse in (2.17).

The EnKF can be written analogously to the Kalman filter’s recursion formula (2.12 and 2.13), by defining a sample Kalman gain matrix,

$$L = C(X_f) H^T (HC(X_f) H^T + R)^{-1}.$$  \hfill (2.18)

Then, the EnKF can be expressed as

$$X^a_{t_i} = X^f_{t_i} + L_{t_i} (D_{t_i} - H_{t_i} X^f_{t_i}), \quad \text{(analysis step);} \hfill (2.19)$$

$$X^f_{t_{i+1}} = M_{t_i} X^a_{t_i} + b_{t_i} e_{1 \times N} \quad \text{(forecast step),} \hfill (2.20)$$

where $e_{1 \times N}$ is a $1 \times N$ vector of 1’s.
2.4 Variants of the EnKF

2.4.1 Square Root Filters

The form of the EnKF described here (2.19) recreates the proper posterior covariance by making use of randomly perturbed data. It has been shown by Whitaker and Hamill [2002] that the use of perturbed data in certain non-Gaussian problems can cause systematic errors in the posterior covariance. As a result, a variant of the EnKF, called the Ensemble Square Root Filter (EnSRF) has been developed that eliminates the necessity of data perturbation. The EnSRF is given as a two step method where the ensemble mean is updated as

\[
\bar{X}^a = \bar{X}^f + L(d - H\bar{X}^f).
\]

Then the analysis ensemble is created out of deviations from this mean

\[
X^a_k = \bar{X}^a + (X^f_k - \bar{X}^f)\tilde{L}_k,
\]

where \(\tilde{L}\) is determined by solving

\[
C(X^a) = (I - LH)C(X^f) \tag{2.21}
\]

ensuring the analysis ensemble has the correct covariance. If we define \(A = X^f - \bar{X}^f1_{1\times N}\), then 2.21 can be rewritten as

\[
A\tilde{L}L^TA^T = (I - LH)AA^T \tag{2.22}
\]

\[
= \left(I - AA^TH^T(HAA^TH^T + R)^{-1}H\right)AA^T
\]

\[
= A\left(I - A^TH^T(HAA^TH^T + R)^{-1}HA\right)A^T
\]

\[
= A\left(I - ZZ^T\right)A^T,
\]
where \( Z \) is a matrix square root, \( Z^T Z = A^T H^T (H A A^T H^T + R)^{-1} H A \). Following Andrews [1968], if we define the singular value decomposition (SVD), \( Z = U \Sigma V^T \), then the solutions to this equation are of the form

\[
\tilde{L} = V \sqrt{I - \Sigma^T \Sigma} \Theta,
\]

where \( \Theta \) is an arbitrary unitary matrix. Evensen [2004] shows that the analysis ensemble can be expressed as \( X^a = X^f \tilde{A} \) for some matrix \( \tilde{A} \). Therefore, as with the EnKF, the analysis ensemble is made up of linear combinations of the forecast ensemble.

The Ensemble Transform Kalman Filter [Bishop et al., 2001] and the Ensemble Adjustment Kalman Filter [Anderson, 1999] are both variations of the EnKF based on the EnSRF, which have been shown to provide more accurate representations of the posterior covariance for many problems. However, both methods require computing the eigenvalue decomposition of dense \( m \times m \) matrices at a cost of \( O(m^3) \).

When there are a small number of data points or data errors are independent and the assimilation can be done sequentially, the cost of this SVD negligible; however, when there is a large amount of correlated data (such as a high resolution image), computing the SVD can become prohibitive; however, Anderson [2003] describes a method in which the ensemble is expressed in terms of a rotated basis, where the data covariance becomes diagonal.

2.4.2 Localized Ensemble Filters

Because the EnKF and the EnSRF arise by replacing the covariance in the optimal Kalman filter with an ensemble covariance, the error in the ensemble methods is necessarily dependent on the sampling error, \( C(X_{N_f}^f) - \text{Cov}(X_{N+1}^f) \). In practice, we
have $N \ll n$. In this case, the sample covariance consists of the sum of $N$ rank 1 matrices,

$$\frac{1}{N-1}(X_{Nk}^f - \bar{X}_N^f)(X_{Nk}^f - \bar{X}_N^f)^T.$$  

Because $\sum_{k=1}^N X_{Nk}^f - \bar{X}_N^f = 0$, the terms are linearly dependent, and the rank of $C(X_f)$ is at most $N - 1$. Further, the perturbation of the forecast ensemble, $X_N^a - X_N^f$, is always in the span of the columns of $C(X_N^f)$. Any uncertainty outside of this subspace will simply be ignored by the classical Kalman filter causing the filter to diverge from the optimal solution over several analysis cycles. In addition, spurious correlations between distant model nodes tend to appear [Houtekamer and Mitchell, 2001, Anderson, 2007]. Furrer and Bengtsson [2007] show that the mean trace of the sampled covariance is equal to the trace of the true covariance; the fact that the sampled covariance has only $N - 1$ positive eigenvalues implies that these eigenvalues are heavily biased. In order to deal with this problem without increasing the size of the ensemble, localized methods attempt to modify the covariances artificially so that it can more accurately reflect the true uncertainty of the system.

Because we do not know the true covariances, we cannot solve this problem precisely; however, we can make certain assumptions about its general structure. Model states are usually made up of variables discretized over a spatial grid. Typically, the correlation between two grid points decays with distance, so that nearby points have a correlation of close to one and distant points are almost uncorrelated. This can be understood mathematically by observing that the state vectors tend to be discretizations of smooth functions. For global weather models, this localized covariance structure has been demonstrated by Patil et al. [1997] and Houtekamer and Mitchell [1998].

While EnKF localization techniques can take a variety of forms, we will consider here only a simple method called $B$-localization. The forecast ensemble covariance
(often called the background error covariance \( B \) in the literature) is directly modified by tapering off correlations between distant grid points. Specifically, the localized covariance is created taking the Schur product with a spatial correlation matrix, \( \rho \),

\[
C_\rho(X^f) = \rho \bullet C(X^f).
\]  (2.23)

Where the Schur product, \( D = A \bullet B \), is an element-wise operation on two matrices of the same size, \( D_{ij} = A_{ij} B_{ij} \). The correlation matrix represents the assumed correlation between grid points, which requires some amount of heuristic analysis. A detailed discussion of modeling correlation over spatial fields can be found in Gaspari and Cohn [1999].

Typically the correlation matrix is generated by a radial shape function, \( S(r) \), which is monotonically decreasing, compactly supported, and \( f_\rho(0) = 1 \). We will denote the “distance” between the \( i^{th} \) and \( j^{th} \) components of the state vector, located at \( r_i \) and \( r_j \) as \( d(r_i, r_j) \). For example, if \( X_i \) and \( X_j \) are the values of variables \( v_{k_1} \) and \( v_{k_2} \) at the locations \( r_{\ell_1} \) and \( r_{\ell_2} \) on the physical domain, then the distance between these components might be the Euclidean distance between their locations, \( d(r_i, r_j) = |r_{\ell_1} - r_{\ell_2}| \). The correlation matrix is then given by

\[
\rho_{ij} = S(d(r_i, r_j)).
\]  (2.24)

If \( d(\cdot, \cdot) \) is a metric over \( \mathbb{R}^n \), then \( \rho \) is symmetric and \( C_\rho(X^f) \) is a covariance [Houtekamer and Mitchell, 2001]. A simple example of this sort of tapering when \( X^f \) contains only a single variable defined on a 1-dimensional domain is

\[
\rho_{ij} = \begin{cases} 
1, & |i - j| \leq 1; \\
0, & |i - j| > 1.
\end{cases}
\]
In this case, $C_{\rho}(X_f)$ is sparse with a bandwidth of 3. In general, these correlation matrices will be sparse with bands of non-zeros near the diagonal. Under traditional uniform discretizations, multi-dimensional domains will have additional bands of non-zeros centered some distance from the main diagonal depending on the grid sizes.

The generator function for correlation matrices typically used in practice is described in Gaspari and Cohn [1999] as a piecewise fifth order rational function, which is similar to a Gaussian function, with bandwidth $\sqrt{3/10}$ and scaled to a maximum value of 1, in shape but compactly supported. This function can be written as

$$S(r) = \begin{cases} 
-\frac{1}{4}r^5 + \frac{1}{2}r^4 + \frac{5}{8}r^3 - \frac{5}{3}r^2 + 1, & 0 \leq r \leq 1; \\
\frac{1}{12}r^5 - \frac{1}{2}r^4 + \frac{5}{8}r^3 + \frac{5}{3}r^2 - 5r + 4 + 2\frac{2}{3}r^{-1}, & 1 < r \leq 2; \\
0, & r > 2. 
\end{cases} \quad (2.25)$$

If we define a length scale, $l$, as the minimum distance over which correlation reduces to zero, then the correlation matrix is defined by

$$\rho_{ij}^l = S\left(\frac{2}{l}d(r_i, r_j)\right). \quad (2.26)$$

The tapered forecast covariance is then used in place of the ensemble covariance in the ensemble update formula. Hamill et al. [2001] describe an algorithm based on this approach; however, instead of assimilating the ensemble globally, each data component is assimilated sequentially on a subset of the state vector. In this sequential approach, a local ensemble is defined for each data component $d_i$ located in on the domain at $r_{d_i}$. A local state vector, $\hat{X}_k^f$, contains only the components of the global state vector, $X_k^f$, that are correlated with the data component, $\{j: d(r_{d_i}, r_j) < l\}$. 

21
The EnSRF is then applied to the local ensemble, $\hat{X}^f$, with the tapered ensemble covariance. This gives a local perturbation to the forecast ensemble for each data component. This procedure is repeated for every data component, where the sum of all the local perturbations is applied to the global forecast ensemble.

For the sequential algorithm, the tapered covariance is a much smaller matrix in which the uncorrelated components of the global state vector are filtered out. This can be seen splitting a large sparse problem into many smaller dense problems; however, it is only rigorously justifiable when the data errors are independent. Anderson [2003] describes a method that allows the use of correlated data by changing the basis of the ensemble states so that the data covariance becomes diagonal. Hunt et al. [2007] note that performing data assimilation sequentially or in small batches has computational advantages over the standard EnSRF because they can be split into many independent problems computed in parallel.
3. Predictor-Corrector Filter

In this section, we focus on an ensemble data assimilation technique known as the particle filter. Unlike the EnKF, the particle filter is capable of representing non-Gaussian distributions. This fact allows it to perform far more accurately in situations involving highly non-Gaussian distributions, such as those with multiple modes. In practice, however, the particle filter requires many more ensemble members than the dimension of the model space, which limits its usefulness in modern applications. In this section, we introduce a modification to the standard particle filter, called the predictor-corrector filter, which seeks to combine the usability of the EnKF in high dimension with the non-Gaussian capabilities of the particle filter. The method discussed here is based on, Mandel and Beezley [2006] and Mandel and Beezley [2009].

3.1 Preliminaries

Because in this section we are concerned with the behavior of filters in high dimension, we will consider model states that are elements of some Hilbert space, $V$, rather than restricting ourselves to $\mathbb{R}^n$. We will denote the inner product of $V$ as $\langle \cdot, \cdot \rangle$. Ensemble members will be random elements, $X_i$, that are $(\Omega, \mathcal{S}, P) \rightarrow (V, \mathcal{B}(V))$ measurable. The forecast distribution of the ensemble is written as $\mu^f(B) = P(X_i^f \in B)$ defined over all $B \in \mathcal{B}(V)$. The analysis distribution, $\mu^a$, is defined similarly. These distributions are allowed to be non-Gaussian, but are assumed to have bounded second moments. Also, the observation function $h$ is now allowed to be nonlinear, but is assumed to be continuous on its domain, and it is assumed that the data likelihood $p(d|X^f)$ is a continuous function of $X^f$. 

23
In this more general setting, we will use the following form of Bayes rule [Kleijn and van der Vaart, 2006]:

\[
\mu^a(B) = \frac{\int_B p(d|X^f) d\mu^f(X^f)}{\int_V p(d|X^f) d\mu^f(X^f)}.\tag{3.1}
\]

for any set \(B \in \mathcal{B}\). Clearly, \(\mu^a\) is again a probability measure defined on \(\mathcal{B}\). The mean of a distribution \(\mu\) is defined as the element \(\overline{\mu} \in V\) such that

\[
\langle x, \overline{\mu} \rangle = \int_V \langle x, u \rangle d\mu(u), \quad \forall x \in V.
\tag{3.2}
\]

The covariance of \(\mu\) is the operator \(Q_\mu : V \to V\) given by

\[
\langle x, Q_\mu y \rangle = \int_V \langle x, u - \overline{\mu} \rangle \langle y, u - \overline{\mu} \rangle d\mu(u), \quad \forall x, y \in V.
\tag{3.3}
\]

Notice that when \(V = \mathbb{R}^n\), we can take \(x\) and \(y\) in these definitions to be coordinate unit vectors, and we recover the usual definition of the mean and covariance.

3.1.1 Weighted ensembles

Weighted ensemble \((X_{Nk}, w_{Nk})_{k=1}^N\) is a collection of members \(X_{Nk} \in V\) and weights \(w_{Nk} \in \mathbb{R}, k = 1, \ldots, N\), such that

\[
w_{Nk} \geq 0, \quad \sum_{k=1}^N w_{Nk} = 1.
\tag{3.4}
\]

We will work with weighted ensembles where the ensemble members \(X_{Nk}, k = 1, \ldots, N\), are identically distributed random variables with some marginal distribution \(\pi\) and the weights are given as proportional to some \(f_{Nk}\). A weighted ensemble defines a measure \(\mu_N\) as the linear combination of Dirac deltas,

\[
\mu_N = \sum_{k=1}^N w_{Nk} \delta(X_{Nk}), \quad w_{Nk} \propto f_{Nk}, \quad k = 1, \ldots, N.
\tag{3.5}
\]
The proportionality constant is easily determined from (3.4). Note that for all sets $B \in \mathcal{B}(V)$ and all real $\mathcal{B}$-measurable functions $f$,

$$\mu_N(B) = \sum_{k=1}^{N} w_{Nk}, \quad \int_B f d\mu_N = \sum_{k=1}^{N} w_{Nk} f(X_{Nk}).$$

From (3.2), the mean of the weighted ensemble is the mean of the measure $\mu_N$, given by

$$\langle x, \mu_N \rangle = \int \langle x, u \rangle \, d\mu_N(u) = \sum_{k=1}^{N} w_{Nk} \langle x, X_{Nk} \rangle, \quad \forall x \in V,$$

giving the usual formula for the weighted sample mean,

$$\mu_N = \sum_{k=1}^{N} w_{Nk} X_{Nk}. \quad (3.6)$$

Likewise from (3.3), the covariance of the weighted ensemble is the covariance operator $Q_{\mu_N} : V \to V$ of the measure $\mu_N$, given by

$$\langle x, Q_{\mu_N} y \rangle = \int \langle x, u - \mu_N \rangle \langle y, u - \mu_N \rangle \, d\mu_N(u)$$

$$= \sum_{k=1}^{N} w_{Nk} \langle x, X_{Nk} - \mu_N \rangle \langle y, X_{Nk} - \mu_N \rangle, \quad \forall x, y \in V. \quad (3.7)$$

In the case when $V = \mathbb{R}^n$, with $x = e_i, y = e_j$ the unit vectors and $X_{Nk,i}$ the entries of the vector $X_{Nk}$, (3.7) yields the weighted sample covariance matrix

$$Q_{\mu_N} = [Q_{\mu_N,ij}], \quad Q_{\mu_N,ij} = \sum_{k=1}^{N} w_{Nk} (X_{Nk,i} - \mu_{N,i})(X_{Nk,j} - \mu_{N,j}). \quad (3.8)$$

Using these definitions for the mean and covariance of weighted ensembles, we can apply the EnKF formula (2.17) to weighted ensembles. In this case, the EnKF update formula only updates the ensemble members, while the weights remain the same.
We say that a weighted ensemble represents a probability distribution $\mu$ when the weights are chosen as the importance weights,

$$X_{Nk} \sim \pi, \quad w_{Nk} \propto \frac{d\mu}{d\pi}(X_{Nk}), \quad k = 1, \ldots, N.$$  \hspace{1cm} (3.9)

where $d\mu/d\pi$ is the Radon-Nikodym derivative, i.e., the density of $\mu$ with respect to $\pi$. (If both $\mu$ and $\pi$ have densities with respect to some reference measure, the Radon-Nikodym derivative is the ratio of their densities, which is how the importance weights are often written.) A standard Monte-Carlo quadrature argument, e.g., [Doucet et al., 2001], shows that if $X_{Nk}, k = 1, \ldots, N,$ are in addition independent, i.e., $X_{Nk}, k = 1, \ldots, N,$ is a sample from $\pi$, then the random measures $\mu_N$ converge to $\mu$ in the sense that

$$E \left( \left| \int f d\mu_N - \int f d\mu \right|^2 \right) = O \left( \frac{1}{N} \right)$$  \hspace{1cm} (3.10)

for every continuous function $f$ on $V$ such that $\int |f d\mu/d\pi|^2 d\pi < \infty$.

The subscript $N$ was needed only to formulate properly the assumptions of the stochastic convergence (3.10). Since the theory of convergence as $N \to \infty$ is outside of the scope of this section, the subscript $N$ in the ensemble members and in the weights is not needed any more and it will be dropped from now on.

### 3.1.2 Particle filters

We first derive the particle filter [Doucet et al., 2001, Ch. 1] by considering the measure $\mu^f_N$, defined by the forecast weighted ensemble $(X^f_k, w^f_k)^N_{k=1}$

$$\mu^f_N = \sum_{k=1}^N w^f_k \delta \left( X^f_k \right), \quad X^f_k \sim \nu, \quad w^f_k \propto \frac{d\mu^f}{d\nu}(X^f_k), \quad k = 1, \ldots, N.$$  \hspace{1cm} (3.11)
as the forecast distribution itself. Application of the Bayes’ rule (3.1) then gives the analysis distribution

\[
\mu_a^N (B) = \frac{\int_B p(d|u) \, d\mu^f_N (u)}{\int_V p(d|u) \, d\mu^f_N (u)} = \frac{\sum_{k=1}^{N} p(d|X^f_k) \, w^f_k}{\sum_{k=1}^{N} p(d|X^f_k) \, w^f_k} = \sum_{k=1}^{N} w^a_k,
\]

where the analysis weights are given by

\[
w^a_k \propto p(d|X^f_k) \, w^f_k, \quad k = 1, \ldots, N,
\]

and the analysis ensemble members are unchanged from the forecast, \(X^a_k = X^f_k\). This is the Sequential Importance Sampling (SIS) method.

Alternatively, one arrives at the same SIS update formula (3.12) by considering weighted ensembles as Monte-Carlo representations of probability distributions. Suppose that \(\mu^f\) is the forecast distribution and, following (3.9), the forecast ensemble \((X^f_k, w^f_k)_{k=1}^{N}\) represents \(\mu^f\),

\[
X^f_k \sim \nu, \quad w^f_k \propto \frac{d\mu^f}{d\nu} (X^f_k), \quad k = 1, \ldots, N.
\]

Consider an analysis ensemble \((X^a_k, w^a_k)_{k=1}^{N}\) such that the measure \(\mu^a_N\) approximates the analysis distribution \(\mu^a\), given by (3.1), and suppose that the marginal distribution of the analysis ensemble is some probability distribution \(\pi\), called a proposal distribution. Then

\[
X^a_k \sim \pi, \quad w^a_k \propto \frac{d\mu^a}{d\pi} (X^a_k) = p(d|X^a_k) \, \frac{d\mu^f}{d\pi} (X^a_k), \quad k = 1, \ldots, N.
\]
While the data likelihood $p(d|X^a_k)$ is assumed to be known and can be readily evaluated, the density of the forecast distribution with respect to the proposal distribution is in general not available, except in the particular case when the members of the analysis ensemble is taken the same as in the forecast ensemble. In that case, $\pi = \nu$, which gives

$$\frac{d\mu_f}{d\pi}(X^a_k) = \frac{d\mu_f}{d\nu}(X^f_k) \propto w^f_k, \quad k = 1, \ldots, N,$$

and, comparing with (3.11), we recover the SIS update (3.12).

Since the regions where the forecast and the analysis distributions are concentrated in general differ, many of the analysis weights tend to be very small or even degenerate to numerical zeros. Since only members with large weights contribute to the accuracy of the representation of the probability distribution, the effective size of the ensemble shrinks. Over several analysis cycle, all the weight tends to become concentrated on a single member. This effect is known as filter degeneracy. To avoid filter degeneracy, the SIS update is followed by a resampling. In the resulting method, called SIR or the bootstrap filter [Gordon and Smith, 1993], a new ensemble member $X^a_k$ is obtained by selecting $X^f_\ell$ with probability $w^a_\ell$. This results in an ensemble with repeated members and all weights equal. Advancing in time by a stochastic model is then relied upon to spread the ensemble again. In [Kim et al., 2003, Xiong et al., 2006], the resampling is done by first estimating the density $p_a$, and then generating the new ensemble by random sampling from the estimated density. In [van Leeuwen, 2003], the use of Cauchy distribution, which has thicker tails that the normal distribution, is suggested to alleviate the degeneracy problem.
3.1.3 Density estimation

If \((X_k)_{k=1}^N\) is a sample from a probability distribution \(\mu\), which has density \(f\) with respect to the measure \(\pi\) and \(f\) is continuous at \(x\), then the density can be approximated by the nonparametric kernel estimate

\[
f(x) \approx \frac{1}{N} \left| \left\{ k : X_k \in B_U(x, r) \right\} \right| \pi(B_U(x, r)),
\]

(3.14)

where \(|\cdot|\) is the counting measure and \(B_U(x, r)\) is the closed ball

\[
B_U(x, r) = \{ u \in V : u - x \in U, \|u - x\|_U \leq r \}
\]

(3.15)

with respect to a subspace \(U \subset V\). The radius \(r\) is called the bandwidth. In finite dimension, Loftsgaarden and Quesenberry [1965] have shown that the density estimate (3.14) converges in probability when \(\pi\) is the Lebesgue measure, \(\|\cdot\|_U\) is the Euclidean norm, \(U = V\), \(r\) is chosen as the distance to the \(k(N)\)-th nearest sample point to \(x\), and

\[
\lim_{N \to +\infty} k(N) = +\infty, \quad \lim_{N \to +\infty} \frac{k(N)}{N} = 0.
\]

(3.16)

For probability distributions on a Banach space \(V\), the estimate (3.14) converges in mean square when

\[
\lim_{N \to +\infty} r = 0, \quad \lim_{N \to +\infty} N \pi(B_U(x, r)) = +\infty,
\]

(3.17)

\(U = V\), and some other technical conditions are satisfied [Dabo-Niang, 2004].
3.2 Derivation of predictor-corrector filters

3.2.1 Motivation

The Kalman filter is capable of making an arbitrarily large change in the state (called innovation in geosciences) in response to the data, because the KF computes the exact analysis probability distribution from Bayes’ rule for the Gaussian case. The EnKF shares this advantage, because it is based on the same algebraic update formulas. In addition, the EnKF formulas can be used for arbitrary probability distributions. However, if the distributions are no longer Gaussian, the underlying assumptions of EnKF are no longer valid and there is no reason to expect that the EnKF should deliver ensembles that would describe the correct analysis distribution. Instead, the EnKF tends to replace the analysis distribution by a distribution closer to Gaussian and concentrated approximately in the same region of the state space.

The SIS is able to approximate arbitrary distributions and converges to the correct distribution in the limit for large ensembles [Crisan, 2001], but it cannot relocate the ensemble members and thus make larger innovation than the spread of the ensemble. When a large change of state is required in the analysis step, SIS with a modestly sized ensemble fails, because it has few or no ensemble members where the posterior is concentrated. This fact is illustrated in Figure 3.1.

The idea of the predictor-corrector filters is that it is possible to get proposal ensembles that can be expected to cover the posterior better by other means, such as the EnKF. The predictor phase is intended to take the forecast ensemble \((X_f^k, w_f^k)\) and produce the proposal ensemble, \((X^a_k, w^f_k)\). In the corrector phase, the weights of the proposal must then be corrected by (3.12) to obtain the analysis ensemble
Figure 3.1: A simple example showing degeneracy of the particle filter. The SIS is applied to a prior ensemble of size 10 sampled out of $\mathcal{N}(0, 0.16)$ with equal weights. The data likelihood is centered away from the prior mean with distribution $\mathcal{N}(2.5, 0.0625)$. In (a), the exact prior, likelihood, and posterior densities are given by the blue, red, and black lines, respectively. The prior and posterior ensemble members and weights are given by blue *’s and black ×’s. The value of the ensemble members are given by their position on the $x$ axis and the weights are the vertical positions. In addition, the scaled likelihood is given by red ◦’s, but because the prior ensemble had equal weights, the likelihood is proportional to the posterior weight. In (b), the posterior ensemble is shown on a semi-log plot. The weight of the largest ensemble member is almost 1, while weights of the rest are essentially 0. In (c), the exact mean and covariance is given along with the computed ensemble mean and covariance from (3.6) and (3.8) for both the prior and posterior. In this case, the particle filter fails to accurately represent these statistics of the posterior.
However, the Radon-Nikodym density \(d\mu^f/d\pi\) is not directly available, so we replace it by density estimation. Recall that from section 3.1.2 that the proposal distribution \(\pi\) is given as the marginal distribution of the members of the analysis ensemble \(X^a_k\). Motivated by the kernel estimate (3.14), we replace the ratio of the densities in (3.13) by the nonparametric estimate

\[
\frac{d\mu^f (X^a_k)}{d\pi (X^a_k)} \approx \frac{1}{N} \sum_\ell \frac{\sum_{\|X^f_\ell - X^a_k\|_U \leq r_k} w^f_\ell}{\pi (B_U(X^a_k, r_k))} = \frac{1}{N} \sum_\ell \frac{\sum_{\|X^f_\ell - X^a_k\|_U \leq r_k} w^f_\ell}{\pi (B_U(X^a_k, r_k))} \quad (3.18)
\]

This estimate uses only the concept of distance and it is inherently dimension independent. Motivated by (3.16), we choose the bandwidth \(r_k\) as the distance to the \(\lfloor N^{1/2} \rfloor\)-th nearest member \(X^a_\ell\) from \(X^a_k\), measured in the \(\|\cdot\|_U\) norm.

### 3.2.2 High-dimensional systems

It remains to choose the norm \(\|\cdot\|_U\). In the scalar case, every norm is simply a multiple of the absolute value. We are particularly interested in the case when the ensemble consists of discrete representations of random smooth functions, such as solutions of partial differential equations. Since the dimension of the state space can be very large when the mesh is fine, looking at the infinitely dimensional case provides a guidance to the choice of the norm \(\|\cdot\|_U\) in the density estimate kernel and brings an insight into the performance of EnKF and SIS as well.

So, consider filters operating on an infinitely dimensional, separable Hilbert space \(V\). The ensemble is initialized as a sample from an appropriate distribution of random functions. A well-known way [Evensen, 1994, Ruan and McLaughlin, 1998] to construct smooth random functions for the initial ensemble is from a complete
orthonormal basis \( \{ \varphi_n \} \) in \( V \) and coefficients \( \lambda_n > 0 \), \( \sum_{n=1}^{\infty} \lambda_n^2 < +\infty \), by

\[
    u = \sum_{n=1}^{\infty} d_n \lambda_n \varphi_n, \quad d_n \sim \mathcal{N}(0,1), \quad \{d_n\} \text{ independent.} \tag{3.19}
\]

The sum (3.19) defines a Gaussian random variable with values in \( V \). In computations, of course, the sum is only finite and the space \( V \) is finite dimensional. Possible choices of \( \{ \varphi_n \} \) include a Fourier basis, such as the sine or cosine functions, or bred vectors [Kalnay, 2003]. In any case, with increasing \( n \), the basis functions \( \varphi_n \) are more oscillatory, and they contribute less to the sum since \( \lambda_n \to 0 \). If \( \lambda_n \to 0 \) sufficiently fast, the sum (3.19) defines a random smooth function \( u \). For example, for \( \varphi_n(x) = \sin nx \), it is easy to see that \( u \in C^k \) a.s. if \( \lambda_n = O(n^{-\ell}) \), \( \ell > k + 1 \).

The distribution of \( u \) is a Gaussian measure on \( V \) and \( \lambda_n^2 \) are the eigenvalues of its covariance.

We choose as the norm in the density estimate (3.18)

\[
    \|u\|_U^2 = \sum_{n=1}^{\infty} \frac{1}{\kappa_n^2} c_n^2, \quad u = \sum_{n=1}^{\infty} c_n \varphi_n, \quad U = \{ u \in V : \|u\|_U^2 < +\infty \}, \tag{3.20}
\]

where \( 0 < \kappa_n \leq 1 \) and \( \lim_{n \to \infty} \lambda_n/\kappa_n = 0 \). The reason for this choice is that for \( \kappa_n = \lambda_n \), \( U \) becomes the Cameron-Martin space of the Gaussian measure, \( \vartheta \), the initial ensemble was sampled from, and \( \vartheta(U) = 0 \) [Da Prato, 2006, Prop. 1.27]. A ball of measure zero does not bode well for the density estimate (3.14) and thus for (3.18); however, a ball in the \( V \) norm (all \( \kappa_n = 1 \)) is guaranteed to have a positive measure [Da Prato, 2006, Prop. 1.25]. Choosing intermediate norms may allow norms more adapted to the probability distribution (and thus to the smoothness of the state), with a smaller ball possibly yielding a more accurate estimate.

An important consequence of the expansion (3.19) is that the Gaussian initial state can be approximated by only a finite sum of the basis functions with arbitrary
accuracy. In fact the sum of eigenvalues of the covariance of any probability measure on a separable Hilbert converges, as long as the covariance exists. For non-Gaussian distributions, the Karhunen-Loève expansion [Loève, 1977] provides a more general approximation. So, one can expect that for a state space of large finite dimension that approximates an infinitely dimensional separable space (such as the spaces where the solutions of partial differential equations are), the behavior of stochastic algorithms such the filters considered here will depend only on the nature of the infinitely dimensional probability distribution and it will not deteriorate in the limit for a large dimension of the state space. This is also the reason why we do not consider distributions where the eigenvalues of the covariance are bounded away from zero. The convergence of particle filters is known to deteriorate [Bengtsson et al., 2008, Snyder et al., 2008] for such state distributions.

3.2.3 Formulation of the method

Combining EnKF, SIS, and density estimation yields the following predictor-corrector algorithm. EnKF is used to make a potentially large change in state, followed by SIS, which changes the weights to account for non-Gaussian distributions.

**Algorithm 3.1.** Given forecast weighted ensemble \( \left( X_{f_k}^f, w_{f_k}^f \right)_{k=1}^N \) and observation \( Hx - d \sim \vartheta \), with \( f \) the density of \( \vartheta \), \( R = \text{Cov} \vartheta \), and \( E(\vartheta) = 0 \):

1. (Predictor) Compute the members \( X_{a_k}^a \) of the analysis ensemble by the EnKF,

\[
X_{a_k}^a = X_{f_k}^f + K(d_k - HX_{f_k}^f), \quad d_k - d \sim \vartheta,
\]

\[
K = C(X_f^f, w_f^f)H^T \left( HC(X_f^f, w_f^f)H^T + R \right)^{-1}
\]

where \( C(X_f^f, w_f^f) \) is the covariance of \( \left( X_{f_k}^f, w_{f_k}^f \right)_{k=1}^N \) following (3.8).
2. (Corrector) Compute the analysis weights $w_k^a$ from the SIS update with density estimation,

$$w_k^a \propto f(HX_k^a - d) \frac{\sum \ell \|x_\ell^f - x_k^a\|_U \leq r_k w_k^f}{\sum \ell \|x_\ell^a - x_k^a\|_U \leq r_k w_k^f},$$

optionally with resampling: Choose $Y_k$ as $X_\ell^a$ with probability $w_\ell^a$, $k = 1, \ldots, N$, then set all $X_k^a = Y_k$ and all $w_k^a = 1/N$.

The bandwidth $r_k$ is the distance to the $\lfloor N^{1/2} \rfloor$-th nearest member $X_\ell^a$ from $X_k^a$, measured in the $\| \cdot \|_U$ norm, which is chosen following (3.20).

Note that if resampling is used, then the ensemble weights are all same, so standard EnKF applies and the extension to weighted EnKF is not needed. The resampling generally duplicates some ensemble members. However, even for a deterministic model, the random selection of the data vectors $d_k$ in the next analysis step will spread the ensemble again. Finally, the predictor corrector filter is not restricted to the present version of EnKF; other version of EnKF can be used just as well.

3.3 Numerical Results

To get an idea of why are predictor-corrector filters beneficial, we will consider some situations where standard filtering techniques are unsuitable. Such conditions are frequently encountered when considering nonlinear problems, or when it is technically unfeasible to use a sufficiently large ensemble to approximate the distributions. We will use predictor by weighted EnKF, and the resulting predictor-corrector filter will be called EnKF-SIS.
Figure 3.2: The problem shown in Figure 3.1 is repeated with the same prior ensemble but using the predictor-corrector filter. With the new method, the proposal ensemble is much closer to the data, and the posterior ensemble exhibits far less degeneracy and its sample mean and covariance is far closer to the exact mean and covariance compared with the SIS method.

<table>
<thead>
<tr>
<th></th>
<th>exact prior</th>
<th>ensemble prior</th>
<th>exact posterior</th>
<th>ensemble posterior</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean</td>
<td>0</td>
<td>-0.13228</td>
<td>1.7978</td>
<td>2.2193</td>
</tr>
<tr>
<td>variance</td>
<td>0.16</td>
<td>0.27636</td>
<td>0.044944</td>
<td>0.0091291</td>
</tr>
</tbody>
</table>
3.3.1 Bimodal Prior

The first such situation we will consider is that of a bimodal prior. We construct a bimodal prior by first sampling from $\mathcal{N}(0, 2.5)$. These samples are then weighted by the function

$$w_f(x_i) = e^{-5(1.5-x_i)^2} + e^{-5(-1.5-x_i)^2}$$

representing the sum of two Gaussian distributions with mean $\pm 1.5$ and variance 0.1. The resulting weighted ensemble can then be considered a weighted sample from the prior probability distribution function shown in Fig. 3.3 (a). The data likelihood is Gaussian, with the mean shifted slightly to the right.

Each filter (EnKF, SIS, and EnKF-SIS) was applied to this problem with an ensemble size of 100. The density ratio estimate (3.18) was then applied to the resulting posterior ensemble to obtain an approximate posterior probability distribution. The results for each method are given in Fig. 3.3.

Because the EnKF technique assumes that all distributions are Gaussian, it is no surprise that it would fail to capture the non-Gaussian features of the posterior. Both SIS and EnKF-SIS were able to represent the nature of the posterior reasonably well.
Figure 3.3: In (a), the prior and data distributions are given by blue and red lines, respectively. The prior ensemble, with size 10, is displayed with blue *’s, where the position of each ensemble member on the x axis is its value and the position on the y axis is its weight. The data likelihood given each ensemble member is displayed as red ◦’s, scaled so the likelihoods sum to 1. The prior distribution has two modes at ±1.5 and is constructed by sampling each ensemble member, $x_k \sim \mathcal{N}(0, 2.5)$ and assigning weights as the pdf of the sum of two Gaussian distributions, $\mathcal{N}(±1.5, 0.1)$ evaluated at $x_k$. The data distribution is $\mathcal{N}(0, 0.1)$. In (b)-(d), the posterior ensemble is shown with black ×’s for EnKF, SIS, and EnKF-SIS, respectively. The exact posterior density is shown as a dotted line, while the estimated posterior density from the ensemble calculated from a gaussian kernel with bandwidth 0.5 is shown as a solid black line. The EnKF alone cannot account for the bimodal nature of the posterior, while the EnKF-SIS method corrects this by applying weights which increase the accuracy of the posterior estimation. In this scalar problem, the SIS method is able to perform reasonably well.
3.3.2 Filtering for a stochastic ODE

The results given above describe how each filtering technique applies to certain carefully designed synthetic distributions. It remains to be seen how these filters work when applied to an actual model. We have implemented a stochastic differential equation as used by Kim et al. [2003],

\[
\frac{du}{dt} = 4u - 4u^3 + \kappa \eta,
\]

(3.22)

where \( \eta(t) \) is white noise, namely an element with Gaussian distribution of zero mean and covariance \( E[\eta(t)\eta(t')] = \delta(t - t') \). The parameter \( \kappa \) controls the magnitude of the stochastic term.

The deterministic part of this differential equation is of the form \( \frac{du}{dt} = -f'(u) \), where the potential \( f(u) = -2u^2 + u^4 \), cf., Fig 3.4a. The equilibria are given by \( f'(u) = 0 \); there are two stable equilibria at \( u = \pm 1 \) and an unstable equilibrium at \( u = 0 \). The stochastic term of the differential equation makes it possible for the state to flip from one stable equilibrium point to another; however, a sufficiently small \( \kappa \) insures that such an event is rare (Fig 3.4b).

The equation (3.22) is a simple model problem related to fire simulation that includes spatial diffusion; the two equilibria in (3.22) are related to the burning and not burning states.

A suitable test for an ensemble filter will be determining if it can properly track the model as it transitions from one stable fixed point to the other. From the previous results, it is clear that EnKF will not be capable of describing the bimodal nature of
Figure 3.4: (a) The deterministic potential curve, and (b) A solution of (3.22) switching between stable points.

the distributions involved so it will not perform well when tracking the transition. When the initial ensemble is centered around one stable point, it is unlikely that some ensemble members would be close to the other stable point, so SIS will be even slower in tracking the transition [Kim et al., 2003].

The solution $u$ of (3.22) is a random variable dependent on time, with probability density $p(t, u)$. The evolution of the density in time is given by the Fokker-Planck equation [Miller et al., 1999],

$$
\frac{\partial p}{\partial t} = \frac{\partial}{\partial u} [4u(u^2 - 1)p] + \frac{\kappa^2}{2} \frac{\partial^2 p}{\partial u^2}. 
$$

(3.23)

To obtain the exact (also called optimal) solution to the Bayesian filtering problem, up to a numerical truncation error, we have advanced the probability density of $u$ between the Bayesian updates by solving the Fokker-Planck equation (3.23) numerically on a uniform mesh from $u = -3$ to $u = 3$ with the step $\Delta u = 0.01$, using the
MATLAB function `pdepe`. At the analysis time, we have multiplied the probability density of $u$ by the data likelihood as in (2.1) and then scaled so that again $\int pdu = 1$, using numerical quadrature by the trapezoidal rule.

The data points (Table 3.1) were taken from one solution of this model, called a reference solution, which exhibits a switch at time $t \approx 1.3$. The data error distribution was normal with the variance taken to be 0.1 at each point. To advance the ensemble members and the reference solution, we have solved (3.22) by the explicit Euler method with a random perturbation from $N(0, (\Delta t)^{1/2})$ added to the right hand side in every step [Higham, 2001]. The simulation was run for each method with ensemble size 100, and assimilation performed for each data point. There was no resampling in EnKF-SIS.

Examining the results in Fig. 3.5, EnKF-SIS was able to approximate the mean of the optimal solution better than either SIS or EnKF alone in the three cycles following the data switching to the lower mode; however, the new method was unable to track the variance as accurately as the EnKF alone during these cycles. The EnKF-SIS appears to under-estimate the true covariance; this behavior could possibly be ameliorated this problem. EnKF provided the smoothest approximation; however, it is unable to track the data quickly as it switches. SIS suffers from a lot of noise as only a small number of ensemble members contribute to the posterior. In addition, SIS

<table>
<thead>
<tr>
<th>$t$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u$</td>
<td>1.2</td>
<td>1.3</td>
<td>-0.1</td>
<td>-0.6</td>
<td>-1.4</td>
<td>-1.2</td>
</tr>
</tbody>
</table>

Table 3.1: Data used in assimilation into (3.22)
Figure 3.5: In (a), an example of the applying the EnKF, SIS, and EnKF+SIS on (3.22) with the data given by circles. Displayed is the ensemble mean of a single realization and the mean of the optimal solution from (3.23) using an ensemble of size 100. In (b) and (c), the estimated mean square error in the ensemble mean and covariance are given for each method by averaging over 100 simulations.
provides even worse tracking of the data than EnKF. The hybrid EnKF-SIS method provides the best tracking in this case, but it exhibits some noise in the solution, because the proposal and forecast distributions are far apart. This noise is caused by the ensemble degenerating to only a few solutions and is similar to that seen with SIS, but less severe.
3.3.3 Filtering in high dimension

Typical results for filtering in the space of functions on \([0, \pi]\) of the form

\[ u = \sum_{n=1}^{d} c_n \sin(nx) \]  \hspace{1cm} (3.24)

are in Figs. 3.6 and 3.7. The horizontal axis is the spatial coordinate \(x \in [0, \pi]\). The vertical axis is the value of \(u\). The level of shading on each vertical line is the marginal density of \(u\) at a fixed \(x\), computed from a histogram with 50 bins. The ensemble size was \(N = 100\) and the dimension of the state space was \(d = 500\). The Fourier coefficients were chosen \(\lambda_n = n^{-3}\) to generate the initial ensemble from (3.19) (i.e., eigenvalues of the covariance are \(n^{-6}\), and \(\kappa_n = n^{-2}\) for the norm in the density estimation.

Fig. 3.6 again shows that EnKF cannot handle bimodal distribution. The prior was constructed by assimilating the data likelihood

\[ p(d|u) = \begin{cases} 
1/2 \text{ if } u(\pi/4) \text{ and } u(3\pi/4) \in (-2, -1) \cup (1, 2) \\
0 \text{ otherwise}
\end{cases} \]

into a large initial ensemble (size 50000) and resampling to obtain the forecast ensemble size 100 with a non-Gaussian density. Then the data likelihood

\[ u(\pi/2) - 0.1 \sim N(0, 1) \]

was assimilated to obtain the analysis ensemble.
Fig. 3.7 shows a failure of SIS. The prior ensemble sampled from Gaussian distribution with coefficients \( \lambda_n = n^{-3} \) using (3.19) and (3.24), and the data likelihood was
\[
u(\pi/2) - 7 \sim N(0, 1).
\]
We observe that while EnKF and EnKF-SIS create ensembles that are attracted to the point \((\pi/2, 7)\), SIS cannot reach so far because there are no such members in this relatively small ensemble of size 100.

We have demonstrated the potential of predictor-corrector filters to perform a successful Bayesian update in the presence of non-Gaussian distributions, large number of degrees of freedom, and large change of the state distribution. The new filter performs as well as EnKF on Gaussian distributions.

We have also observed that on spaces of smooth functions, the particle filter itself often works quite well already, as long as the updates are not too large, even in very high dimension. This appears due to the fact that the smooth functions can be approximated well by a low-dimensional space regardless of the number of the state dimension. We have noted that due to general properties of probability distributions, this also holds for a large class of probability distribution on infinite dimensional vector spaces, in particular for models by partial differential equations.

Open questions include convergence of the filters when applied to multiple updates over time, mathematical convergence proofs for the density estimation and for the Bayesian update, and performance of the filters when applied to systems with a large number of different physical variables and modes, as is common in atmospheric models.
Figure 3.6: In (a), a non-Gaussian prior distribution is applied to a data distribution with small likelihood. The exact posterior distribution is given in (b). The EnKF, SIS, and EnKF-SIS was applied to an ensemble of size 100. In (c), the EnKF does not approximate the exact posterior distribution well because it ignores the bimodal prior. The SIS given in (d) does approximate the posterior reasonably well because the data likelihood is large. In (e), the EnKF-SIS resembles the features of (d) by reducing the weights of the proposal ensemble members that pass though the center of the figure.
Figure 3.7: In (a), a Gaussian prior distribution is applied to a data distribution with small likelihood. The exact posterior distribution is given in (b). The EnKF, SIS, and EnKF-SIS was applied to an ensemble of size 100. In (c), the EnKF approximates the exact solution well because the prior was Gaussian; however, the SIS given in (d) was fairly degenerate and was not able to correct for the large shift in the distribution. In (e), the EnKF-SIS approximates the shape of the exact posterior well, while exhibiting far less degeneracy than (d).
4. The EnKF in Practice

4.1 The curse of dimensionality

The “curse of dimensionality” is the well known issue involving Monte Carlo sampling, where the number of samples necessary to estimate a high dimensional probability density must grow exponentially with the dimension of the system. The problems with the particle filter in high dimension discussed in Section 3 are directly related to this issue. In Bengtsson et al. [2008], the authors formalize this concept in a Gaussian setting by showing that, when the forecast covariance has a spectrum bounded away from zero, filter collapse will occur unless the size of the ensemble grows exponentially with the size of the model. The assumptions of this statement, however, are likely too strong to be of practical relevance. For instance, it is known that the covariance of a Gaussian measure on a Hilbert space has a bounded trace [Kuo, 1975]. Therefore, a Gaussian measure on an infinitely dimensional space cannot have a covariance satisfying these conditions. In its limit, this problem can no longer be considered as Monte Carlo sampling of a Gaussian distribution.

We would like to examine the behavior filtering error as the size of the model, \( n \), increases; however, this requires some care because the distributions necessarily vary with \( n \). For this, we will consider an infinitely dimensional, separable, Hilbert space, \( V \), with a complete, orthonormal sequence, \( \{v_n\} \). Denote \( V_n \subset V \) as the \( n \)-dimensional subspace spanned by \( v_1, \ldots, v_n \). If we assume that \( \mu \) is a measure on \((V, \mathcal{B}(V))\), such that \( \mu(\text{span}\{v_n\}) > 0 \) for each \( n \). We will consider “\( n \)-dimensional projections” of this measure, defined on \((V_n, \mathcal{B}(V_n))\), as \( \mu_n(A) := \mu(A)/\mu(V_n) \) for
every $A \in \mathcal{B}(V_n)$. The question the dependence of filtering error on model dimension will be posed as follows: We are given $\mu^f$, the true forecast distribution on $(V, \mathcal{B}(V))$ and a given data set with known error density. For each $n$, $\mu^a_n$ is the true analysis distribution resulting from the $n$-dimensional projection, $\mu^f_n$, of the forecast applied to the same data likelihood. We can apply a data assimilation algorithm to each $\mu^f_n$, which results in an approximation, $\tilde{\mu}^a_n$, of the true analysis distribution, $\mu^a_n$. Following (3.7) we will denote, $Q_\mu$, the covariance of the measure, $\mu$. The error of the data assimilation algorithm for each $n$ will be $||Q_\tilde{\mu}^a_n - Q_{\mu^a_n}||_2 = E(||Q_\tilde{\mu}^a_n - Q_{\mu^a_n}||^2)$, where $|| \cdot ||$ is the matrix 2-norm. In the numerical experiments that follow, we will study the behavior of this error as a function of $n$.

Consider a Gaussian measure $\mu$ with mean 0 and positive definite covariance $Q_\mu$. If we define $\{v_i\}$, an orthonormal basis of eigenvectors of $Q_\mu$ with $\lambda_i$ their corresponding eigenvalues, then a random element, $X \sim \mu$, can be expressed in terms of this basis as a random sequence, $\{x_i\}$, with diagonal covariance:

$$\lambda_i \delta_{ij} = v_i^T Q_\mu v_j = [\text{Cov}(X)]_{ij} = E(<x_i v_i, x_j v_j>) = E(x_i^2) \delta_{ij}. $$

Because a covariance is Gaussian if and only if it has a bounded trace, we can construct a measure with Gaussian covariance over $V$ out of an i.i.d. sequence defined by

$$x_i = \sqrt{\lambda_i} \varepsilon_i, \quad \lambda_i \in t^1, \quad \varepsilon_i \sim \mathcal{N}(0, 1). \tag{4.1}$$

The following figures explore the error of the EnKF and SIR applied to forecast ensembles of $n$-dimensional random vectors sampled out of a Gaussian distribution with zero mean and diagonal covariance. We will explore this error for various eigenvalue structures: constant ($\lambda_i = 1$), inverse ($\lambda_i = \frac{1}{i}$), and inverse square ($\lambda_i = \frac{1}{i^2}$).
Notice that only in the last eigenvalue structure can the limiting case possibly be a Gaussian distribution, and only the first conforms to the assumptions discussed in Bengtsson et al. [2008]. The data used will be sampled out of an $m$-dimensional Gaussian distribution with zero mean and covariance $0.1I$. We will consider a linear observation operator made up of independent random components sampled out of $\mathcal{N}(0, 1)$. The true analysis covariance will be calculated from (2.12). In the $y$-axis is the error in the analysis covariance, $E(|Q_{\tilde{\mu}} - Q_{\mu}|^2)$, is approximated by averaging over 100 simulations.

While the use of a random observation function may seem counter-intuitive, the intention is to describe the average filtering error for an arbitrary data assimilation satisfying the assumptions of the EnKF. Naturally, as the size of the system state changes, the observation operator must change as well. The use of a random $H$ with independent components sampled out of $\mathcal{N}(0, 1)$ is an ad-hoc method for accomplishing the desired result. Therefore, there is little theoretical justification for this choice. For completeness, Figures 4.5-4.8 repeat these experiments using a deterministic observation function defined as an $m \times n$ matrix, $H_{ij} = \delta_{ij}$. This form of $H$ produces observations in the largest $m$ directions of uncertainty of the forecast distribution. Figures 4.9 and 4.10 show the filtering error where the data size changes with the model size $m = n$ and the observation operator is the identity, representing a direct observation of the system state.

The behavior of the error as a function of model size is fairly consistent in all of the figures for both filtering methods. In particular, the error for constant $\lambda_i$ tend to rapidly increase with the model size. This is consistent with the familiar curse of dimensionality. However, the error for $\lambda_i = \frac{1}{i^2}$ remain remarkably flat in all of the
Figure 4.1: For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 10 ensemble members and 25 data points with a random observation matrix.

Figure 4.2: For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 10 ensemble members and 250 data points with a random observation matrix.
Figure 4.3: For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 100 ensemble members and 25 data points with a random observation matrix.

Figure 4.4: For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 100 ensemble members and 250 data points with a random observation matrix.
Figure 4.5: For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 10 ensemble members and 25 data points with observation matrix, $H_{ij} = \delta_{ij}$.

Figure 4.6: For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 10 ensemble members and 250 data points with observation matrix, $H_{ij} = \delta_{ij}$. 
Figure 4.7: For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 100 ensemble members and 25 data points with observation matrix, $H_{ij} = \delta_{ij}$.

Figure 4.8: For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 100 ensemble members and 250 data points with observation matrix, $H_{ij} = \delta_{ij}$.
Figure 4.9: For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 10 ensemble members and an identity observation matrix.

Figure 4.10: For constant, inverse, and inverse square forecast covariance eigenvalue structures, the mean square covariance filtering error versus model size is displayed for SIR (a) and EnKF (b) using 100 ensemble members and an identity observation matrix.
figures, suggesting a certain amount of independence between filter error and model size. The behavior for $\lambda_i = \frac{1}{i}$ exhibits mixed results between that of the red and blue lines. As expected, increasing the number of samples tends to decrease the error. Also, the experiments present in all of the figures was repeated using the predictor corrector filter, EnKF-SIR, but the results were nearly identical to that of the EnKF alone and have been omitted. The analysis distributions are Gaussian by construction, so the corrections produced by SIR are of little consequence.

We observe that the results for $\lambda_i = \frac{1}{i^2}$ are the only problems where the distributions can remain Gaussian in the limit. The results for $\lambda_i = \frac{1}{i}$ lie on the boundary of such distributions in that covariances with eigenvalues decaying any faster are Gaussian covariances in their limit. The results of these experiments provide evidence that the curse of dimensionality in the analysis covariance error simply does not occur when the limiting covariances satisfy the conditions of the EnKF (i.e. they are Gaussian). The evidence presented here present opportunities for further theoretical research.
4.2 Generating initial ensembles

So far, we have assumed that an initial ensemble is given which is sampled out of a known forecast error distribution without mentioning how this is actually done. In atmospheric sciences, both the creation of such ensembles and producing initial distributions has been the focus of much research. Common techniques involve constructing covariances out of known spatial correlations and determining components with the largest variance directly from the computational model known as bred vectors. For our purposes, we will limit the construction of initial ensembles to Gaussian perturbations of a central model state, \( X_i = X_0 + \varepsilon_i, \varepsilon_i \sim \mathcal{N}(0, Q) \).

Because the model states in our application represent smooth functions obtained from solutions of PDE’s, we must take care to produce ensemble states that retain the properties of the model solutions. The most straightforward method of creating Gaussian perturbations would be to add a Gaussian random variable with independent components or equivalently a diagonal covariance. However, in the limit, such perturbations do not produce smooth functions.

For simplicity, assume the model states are comprised of a single variable, which belongs to a space of smooth functions over the bounded domain, \( D = [-\pi, \pi] \). For this, we will consider Sobolev spaces, \( H^\alpha(D) \) for \( \alpha \geq 0 \), which contain functions whose \( \alpha \)-order derivative is in the space \( L^2(D) \). If \( \{v_n\}_{n=0}^\infty \) is Fourier series of the element \( v \in H^\alpha(D) \), then the norm of this space can be expressed as

\[
||v||_{H^\alpha} = \sum_{n=0}^\infty (1 + n^2)^\alpha |v_n|^2.
\] (4.2)
This tells us that the higher frequency components of a smooth function must decay sufficiently rapidly for this sum to remain bounded. In particular, if the Fourier coefficients decay as $\frac{1}{n^{1+\alpha}}$, then the function is an element of $H^\alpha(D)$. We will use this as motivation for the following construction, similar to that presented in Evensen [1994], of what we will call “random fields of smoothness $\alpha$”.

If the domain, $D$, is discretized over a uniform mesh of size $n$ with resolution $h$, then we define a vector $z_k = c_k + id_k$, with $c_k, d_k \sim \mathcal{N}(0, 1)$. The random field will be given by

$$
v(x) = \Re \left\{ \sum_{k=1}^{n} \frac{1}{k^{1+\alpha}} z_k e^{i(k-1)x} \right\},
$$

which can be computed efficiently by taking the real part of the inverse fast Fourier transform of $\frac{1}{k^{1+\alpha}} u_k$.

In general, we can follow this procedure to produce perturbations to each discrete spatial variables making up the state vector. For a 3-dimensional variable discretized over a uniform grid of dimensions $n_x \times n_y \times n_z$ with resolution $h_x \times h_y \times h_z$, the process is similar. We create a random element in $\mathbb{C}^{n_x \times n_y \times n_z}$ with independent components sampled out of $\mathcal{N}(0, 1)$. Each component, $(i_x, i_y, i_z)$, of this random element is scaled by something proportional to

$$
\left( \frac{i_x}{h_x} \right)^2 + \left( \frac{i_y}{h_y} \right)^2 + \left( \frac{i_z}{h_z} \right)^2 \frac{1+\alpha}{2}.
$$

The proportionality constant serves to make the scaling independent of the spatial units of the domain and will be considered to be a problem dependent parameter. The real part of the 3-dimensional inverse Fast Fourier Transform (FFT) of the scaled array is multiplied by another proportionality constant to obtain the random perturba-
Figure 4.11: The effect of creating a smooth random field using the procedure outlined above. An i.i.d. complex vector of length 100 is sampled out of $N(0, 1)$. The components of this vector are scaled according to 4.3, with $\alpha = 0, 1, 2$. A larger $\alpha$ reduces the influence of higher frequency components, and the resulting function is smoother.

The effect of this procedure is illustrated in Figure 4.11 for various levels of smoothness, $\alpha = 0, 1, 2$. The inverse FFT constructs functions that have periodic boundary conditions. If this behavior is not desired, one can use this procedure on a domain twice as large in each dimension and restrict the output to the desired size. In some applications, it is desirable to create random functions with 0 Dirichlet boundary conditions. This is possible by modifying the array just before applying the inverse FFT in the following manner. Suppose $\tilde{z}_{i_x, i_y, i_z}$ is an array in the frequency domain. If we modify the first component in each dimension according to

\[
\tilde{z}_{1, i_y, i_z} = -\sum_{i_x=2}^{n_x} \tilde{z}_{i_x, i_y, i_z}, \quad \tilde{z}_{i_x, 1, i_z} = -\sum_{i_y=2}^{n_y} \tilde{z}_{i_x, i_y, i_z}, \quad \tilde{z}_{i_x, i_y, 1} = -\sum_{i_z=2}^{n_z} \tilde{z}_{i_x, i_y, i_z},
\]

then the inverse FFT of the resulting array will be 0 on the boundary.
The preceding analysis provides a means to construct random perturbations to a state vector using smooth functions. In addition, forecast ensembles produced in this way satisfy the conditions presented in 4.1. We can expect that the filtering error produced from such ensembles to behave well as the state size grows.

### 4.3 Implementation

The formulation of the EnKF as presented in (2.17) is written in form that can be easily understood; however, it is not efficient to compute it in this form. We have chosen to compute the ensemble update with the inverse formed by an application of the Sherman-Morrison-Woodbury formula [Hager, 1989]

\[
(\mathbf{H}^T \mathbf{C}_f \mathbf{H} + \mathbf{R})^{-1} = \left( \mathbf{R} + \frac{1}{N-1} (\mathbf{H} \mathbf{A}) (\mathbf{H} \mathbf{A})^T \right)^{-1} = \\
(\mathbf{R}^{-1} + \frac{1}{N-1} (\mathbf{H} \mathbf{A}) (\mathbf{I} + (\mathbf{H} \mathbf{A})^T \mathbf{R}^{-1} \frac{1}{N-1} (\mathbf{H} \mathbf{A}))^{-1} (\mathbf{H} \mathbf{A})^T \mathbf{R}^{-1})^{-1}
\]

(4.4)

\[
A = \mathbf{X}_f - \bar{\mathbf{X}} \mathbf{e}_1 \times N.
\]

This formula is advantageous when the data error covariance matrix \( \mathbf{R} \) is of a special form such that left and right multiplications by \( \mathbf{R}^{-1} \) can be computed inexpensively. In particular, when the data errors are uncorrelated, which is often the case in practice, the matrix \( \mathbf{R} \) is diagonal. In this form, the \( N \times N \) matrix,

\[
\mathbf{W} = \mathbf{I} + (\mathbf{H} \mathbf{A})^T \mathbf{R}^{-1} \frac{1}{N-1} (\mathbf{H} \mathbf{A}),
\]

is inverted using the LAPACK subroutine \texttt{dposv}, which takes its Cholesky decomposition \( \mathbf{W} = \mathbf{L} \mathbf{L}^T \) and solves \( \mathbf{V} \leftarrow \mathbf{W}^{-1} \mathbf{V} \) in the factored form using triangular back-substitution. In this case, the EnKF formula (2.17) with (4.4) costs \( O(N^3 + mN^2 + nN^2) \) operations, which is suitable both for a large number \( n \) of the degrees of freedom and a large number \( m \) of data points. Also, (2.17) can be
implemented without forming the linearized observation matrix $H$ explicitly by only evaluating the observation function $h$ using

$$[HA]_i = HX_i^f - H \frac{1}{N} \sum_{j=1}^{N} X_j^f = h(X_i^f) - \frac{1}{N} \sum_{j=1}^{N} h(X_j^f),$$

$$D_i - HX_i^f = D_i - h(X_i^f).$$

See Mandel et al. [2009a] for further details.

The ensemble filter formulas are operations on full matrices, and they were implemented in a distributed parallel environment using MPI and ScaLAPACK. EnKF is naturally parallel: each ensemble member can be advanced in time independently. The linear algebra in the Bayesian update step links the ensemble members together.

### 4.4 Wildfire data assimilation

A wildfire is a large and complex physical process driven by fine scale chemical reactions. The length scales involved range over several orders of magnitude, from millimeters for the smallest pieces of vegetation up to the entire burning region spanning many kilometers. From the heat generated, these fires are capable of producing large gusts of wind that can even affect the local weather patterns. In turn, the weather, from winds, moisture, and atmospheric temperature, are major components driving behavior of the fire. It is essential to take into account these effects when modeling the wildfire phenomena in the form of a two way coupled fire-atmospheric model.

Many wildfire models have been developed over time that range from complex geophysical models, which attempt to replicate small scale features, to simple, idealized models, designed only to approximate the large scale behavior. While the complex models tend produce more accurate simulations in reanalysis studies, they
are generally far too computationally expensive for use in real time forecasting. Because we are interested in real time forecasting with ensemble data assimilation, it is essential that the model is capable of running faster than real time on a workstation class computer.

In the research reported here, we will consider two idealized wildfire models designed to mimic the large scale behavior of real wildfires. The first model [Mandel et al., 2005] is given as a convection-reaction-diffusion PDE that describes the state in terms of the temperature of the burning and the fraction of unburned fuel. These variables, obtained from the solution of two coupled PDE’s, exhibit a nonlinear, traveling wave, the profile and speed of which are determined to match experimental data. The second model [Mandel et al., 2009a] is semi-empirical, based on the assumption that fire propagation speed is normal to the fireline as a function of wind and terrain slope and assumes an exponential decay of fuel from the time of ignition. The fire propagation speed and heat generated is given by experimental data generated for various fuels. It is distinguished by the lack of temperature or fire intensity model variables. Instead, the model state contains only information on the time of ignition, as well as the unburned fuel fraction. This model is designed to run on a relatively coarse grid in comparison to the PDE model, so it is more suitable for problem domains exceeding 1 km.

4.4.1 A PDE Wildfire Model

The PDE wildfire model expresses the model state in terms of the temperature of the fire, $U(x, y)$, and the fuel fraction, $F(x, y)$ defined on a surface. The evolution of the model is derived from a heat balance equation with terms describing the transporting of heat through diffusion and convection. The heat balance also contains
forcing terms representing cooling into the atmosphere and reactive heat input. The reaction rate, \( r(T) \), is a function of the temperature and is governed by a modified Arrhenius equation from chemical kinetics,

\[
r(U) = r_0 e^{-c_r (U - U_a)},
\]

where constants \( r_0 \) and \( c_r \) depend on the fuel properties and \( U_a \) is the atmospheric temperature. The model is governed by the following coupled equations,

\[
\begin{align*}
\left\{
\begin{array}{l}
\frac{c_u}{c_u} \frac{\partial U}{\partial t} = \nabla \cdot \left( k \nabla U \right) - c_v (\overrightarrow{v} + \gamma \nabla z) \cdot \nabla U + c_f Fr(U) - c_c (U - U_a) \\
\frac{dF}{dt} = -Fr(U),
\end{array}
\right.
\end{align*}
\]

with constants \( c_u \), \( k \), \( c_v \), \( c_f \), \( \gamma \), and \( c_c \) and terrain height, \( z \). The constants have been determined in Mandel et al. [2005] so that the temperature profile (Figure 4.13) approximates the shape of a real fire. The wind velocity, \( v \), is taken as a known quantity in the standalone model, or it is determined from the output of the atmospheric model in the coupled model. Although this model is simple, its solutions exhibit the qualitative behaviors of combustion; however, stability of a numerical solution requires mesh resolutions on the order of 1 m and a time step of about 0.01 s.

4.4.2 A Semi-empirical Wildfire Model

The semi-empirical fire propagation model discussed here imposes the fire spread rate directly from laboratory data, without consideration of the reaction intensity or shape of the fire profile. Instead, it assumes that the propagation direction is normal to the fireline and the speed is known. This model predicts the heat flux, \( \Phi(x, y) \) produced by the reaction, which is determined exclusively by the time passed since ignition, \( t_i(x, y) \) and various fuel properties. The equations governing this behavior
are given in the following formula:

\[ \Phi(x, y, t) = -A(x, y) \frac{\partial}{\partial t} F(x, y, t), \]  

(4.7)

where \( A(x, y) \) is a parameter determined from the fuel type and \( F(x, y, t) \) is the fuel remaining, which is assumed to decrease exponentially as it burns:

\[
F(x, y, t) = \begin{cases} 
  F_0(x, y) e^{-\left(t-t_i(x,y)/W(x,y)\right)}, & \text{if } (x, y) \in \Omega(t), \\
  F_0(x, y), & \text{otherwise},
\end{cases}
\]  

(4.8)

where \( F_0(x, y) \) is the initial fuel supply, \( W(x, y) \) is the time constant of the fuel, and \( \Omega \) defines the burning region.

The propagation of the burning region, \( \Omega \), is modeled as a level set function, \( \psi(x, y) \), which has the property that it is continuous and non-negative outside the burning region, \( \Omega = \{(x, y) : \psi(x, y) \leq 0\} \). The fireline is defined as the boundary of the burning region which coincides with the zero contour of \( \psi \). The evolution of the level set function is governed by the following differential equation:

\[
\frac{\partial \psi}{\partial t} + S(x, y) \| \nabla \psi \| = 0,
\]  

(4.9)

where the spread rate, \( S(x, y) \) of the fireline is modeled from the modified Rothermel’s formula [Clark et al., 2004, Rothermel, 1972]. This empirical formula takes into account the wind, terrain slope, and fuel properties to replicate the observed behavior of real wildfires. The level set equation 4.9 is initialized so that its magnitude increases with the distance from the fireline. For a point ignition centered at \((\tilde{x}, \tilde{y})\) with radius \( r \), the level function is initialized as

\[ \psi(x, y) = \| (x, y) - (\tilde{x}, \tilde{y}) \| - r. \]
Figure 4.12: In (a), the fireline and level set function for a fire initialized by two thin lines and a small circle propagating in constant wind. The fire is in the area where the level set function is negative. The black contour is the fireline. In (b), the heat flux generated by the fire in (a).

Figure 4.12 illustrates such a level set function for a fire propagating across the domain, and the corresponding heat flux. Finally, while the level set function does specify the burning region uniquely, the converse is not true; there is only a one to one relationship between $\Omega$ and the zero contour of $\psi$.

4.4.3 Coupling fire and weather models

Generally, a fire model is coupled with a computational fluid dynamics solver in order to approximate the complex relationship between the fire and the atmosphere. We have chosen to couple the level set propagation model with the Weather Research and Forecasting (WRF) code [WRF Working Group, 2005]. WRF, which is a standard for weather forecasting, supports distributed memory execution, and provides import and export of the state through binary files for data assimilation.
The fire model takes as input the horizontal wind velocity \( \vec{v} \), and it outputs the heat flux into the atmosphere, \( \Phi(x, y) \), given by (4.7). The heat flux is split into sensible heat flux (a transfer of heat between the surface and air due to the difference in temperature between them) and latent heat flux (the transfer of heat due to the phase change of water between liquid and gas) in the proportion given by the fuel type and its moisture.

Since the fire mesh is generally finer than the atmospheric mesh, the wind is interpolated to the nodes of the fire mesh, and the heat flux is aggregated over the cells of the fire mesh that make up one cell of the atmospheric mesh. The ratio of the fire mesh resolution to the atmospheric mesh resolution is called mesh refinement ratio. In the numerical results in Section 6.2, the atmospheric mesh has dimensions \( 42 \times 42 \times 41 \) over a \( 2.5 \times 2.5 \times 1.5 \, km^3 \) domain and the fire mesh as dimensions \( 420 \times 420 \), for a refinement ratio of \( 10 : 1 \). The time step used for these simulations was \( 250 \, ms \). The dimension of the combined state vector used for data assimilation was about \( 1.4 \times 10^8 \). For a problem of this scale, the model runs a 1 minute simulation in approximately 4 minutes on a 1.4 \( GHz \) Intel Xeon processor. Using the parallel processing capabilities, the model spread over 4 processes can be run in approximately real time. For the most part, the processing time is dominated by the atmospheric solver rather than the wildfire model. Increasing the refinement ratio will reduce the computational burden significantly at the cost of increased error in interpolating the wind onto the fire mesh.

4.4.4 Wildfire Data

We want to model the location of the burning region as it moves through the domain. The data considered here will be assumed to be an observation of the heat flux
Figure 4.13: From Mandel et al. [2008], the temperature profile of a combustion wave traveling over a fixed point. The modeled temperature profile given by the black line was optimized to approximate the dotted line, representing real data obtained from Kremens et al. [2003].

throughout the domain, as would be obtained by an arial infrared image. Such images can be processed to determine the approximate location of the fire front [Ononye et al., 2007]. For simplicity, we will assume that the data is a direct observation of one of the model state variables, such as the heat flux. While this is not a realistic assumption, the data can be preprocessed to interpolate the image data to a subset of the model grid. Extensions that allow the use of more complicated observation functions will be the focus of future research.

Wildfire models tend to develop into traveling waves with a sharp leading edge followed by an exponentially decaying cool-down [Mandel et al., 2008]. Under uniform fuel and wind conditions, the shape of the combustion wave (Figure 4.13) remains relatively constant; however, small errors in the speed of propagation will eventually result in large errors in its location over time. If we create an ensemble by moving a known temperature profile in space, where the magnitude of the pertur-
Figure 4.14: The estimated point-wise probability density of the temperature near the fire line for an ensemble generated by perturbing a known temperature profile in space by a Gaussian random variable. The density has peaks centered around burning and non-burning regions and is highly non-Gaussian.

...bation is a Gaussian random variable, then the point-wise error distributions of the model variables tends to develop highly non-Gaussian features. Figure 4.14, shows the estimated density of the temperature at single point in the domain for an ensemble constructed in this manner. Applying additive corrections to the state variables as discussed in Section 4.2 simply cannot replicate this highly non-Gaussian behavior.

4.4.5 Spatially Perturbed Ensembles

For the remainder of this section, we will limit our attention to a highly simplified problem, in which the fuel supply and wind are all uniform over the problem domain. The ensemble members will be initialized as a point ignition originating at a location determined from a Gaussian random variable. The ensemble members will then be advanced in time using the fire model described in Mandel et al. [2009a]. The resulting ensemble members will all have approximately the same shape shifted in space according to the perturbation of the ignition. The data will be a direct observa-
tion of one of the state variables taken from an independently initialized model state representing the “truth”. In this case, we want the analysis ensemble to consist of the same temperature profiles shifted toward data with a slightly reduced spatial spread amongst them.

In Figures 4.15 and 4.18, forecast ensembles of size 10 are shown in black lines superimposed on each other. Each ensemble member is shifted in the domain with a variance of 200 m. The data is given as a dotted red line. In Figure 4.15a, the data is located in the mean position of the ensemble members, and in Figure 4.18a the data is shifted to the left 400 m. In Figures 4.15b and 4.18b, the point-wise ensemble variance is given as a black line. In each case, the EnKF as given in 4.5 is applied with diagonal data covariance containing 100 $K^2$ components. The resulting analysis ensemble and variance is given in Figures 4.16 and 4.19. In addition, the same ensemble is applied to a $B$-localized EnKF, where the forecast covariance is tapered by (2.25), with a localization radius of 20 m, chosen to be approximately half the width of the temperature profile. The resulting analysis ensembles are given in Figures 4.17 and 4.20.

The results of these simple tests show that the EnKF, with and without localization, do not in general produce analysis ensembles exhibiting the behavior that we want. The analysis ensemble which most closely resembles the data is obtained with the localized EnKF in Figure 4.17. However, the ensemble members contain spurious features outside of the main ignition region. In terms of the location of the reaction front, even this ensemble has essentially degenerated to a single location. When the data does not happen to be near an ensemble member, both methods fail to produce anything resembling a valid temperature profile.
Figure 4.15: In (a), a forecast ensemble of size 10 containing temperature profiles shifted in space with a variance of 200 m. The data profile given as a dotted red line is located in the center of the ensemble members. In (b), the point-wise ensemble variance calculated from this ensemble.

Figure 4.16: The analysis ensemble resulted in applying the EnKF to the forecast ensemble and data in Figure 4.15. The data variance used was 100 $K^2$. The EnKF was unable to track the data accurately, and the analysis ensemble has essential degenerated to a single solution, with variance reduced by several orders of magnitude.
Figure 4.17: The analysis ensemble resulted in applying the $B$-localized EnKF to the forecast ensemble and data in Figure 4.15. The data variance used was 100 $K^2$ and localization radius was 20 $m$. The localized EnKF was able to create a posterior ensemble which tracks the data accurately on the reaction front; however, there are many spurious features outside of this region. In addition, the localization has reduced the degeneracy of the analysis ensemble compared to the EnKF without localization.

Figure 4.18: In (a), a forecast ensemble of size 10 containing temperature profiles shifted in space with a variance of 200 $m$. The data profile given as a dotted red line is located 400 $m$ to the left of the center of the ensemble members. In (b), the point-wise ensemble variance calculated from this ensemble.
Figure 4.19: The analysis ensemble resulted in applying the EnKF to the forecast ensemble and data in Figure 4.18. The data variance used was $100 \, K^2$. In this case, the analysis ensemble does not resemble the data at all because the forecast ensemble contained no members nearby.

Figure 4.20: The analysis ensemble resulted in applying the $B$-localized EnKF to the forecast ensemble and data in Figure 4.18. The data variance used was $100 \, K^2$ and localization radius was $20 \, m$. The results here are similar to that of the non-localized EnKF; the analysis ensemble was unable to track the data because no forecast ensemble members were nearby.
These experiments were specifically designed to eliminate many of the complexities that would be involved in applying the EnKF to a real world problem. In a real 2-diminsional environment, the location of the fire cannot be described by a scalar position, and intensity and shape of the front can vary as well. Because the standard filters failed to produce the kind of results that we want, even with these simple tests, we need to examine the inherent difficulties involved. We can offer several plausible explanations for why the results were unsuccessful. First, the nonlinearity of the model itself produces highly non-Gaussian forecast distributions. The EnKF represents the forecast distribution only in terms of its mean and covariance, so it cannot "see" the higher order features of the true distribution. Second, the EnKF forms analysis ensembles out of linear combinations of the forecast ensemble. Taking a linear combination of two spatially disjoint reaction fronts results in two small reaction fronts rather than a single one between the original two as we might want. While localization eases this restriction, the results still exhibit the same behavior.

The issues presented here underscore a fundamental problem with employing a standard approach to this data assimilation problem. As it has been posed in this example, this is essentially a scalar problem. The only true unknown is the position of the fire. The representation of the problem in terms of a function (the temperature) casts it in an unnecessarily complicated high-dimensional fashion. A far more reasonable representation of the ensemble would contain only a single, scalar state variable: the position of the fire. If we also represented the data in this way, the standard assumptions of the EnKF are met: the model is linear because the position of the fire moves at a constant speed, the observation function is 1, and the ensemble was initialized from i.i.d. Gaussian random variables. Clearly, real world applica-
tions of wildfire data assimilation cannot be cast in such simplistic terms, but we will use the concept of transforming the ensemble into some sort of alternate form which describes its position in space as a motivation for the Morphing EnKF described in Section 6.
5. Image Morphing and Registration

In this section, we build the tools from image processing that we will use for the morphing EnKF later in Section 6.1. The registration problem in image processing is to find a spatial mapping that turns two given images into each other [Brown, 1992]. Classical registration requires the user to pick manually points in the two images that should be mapped onto each other, and the registration mapping is interpolated from that. Here we are interested in a fully automatic registration procedure that does not require any user input. The specific feature or objects, such as fire fronts or hurricane vortices, do not need to be specified. The method takes as input only the pixel values in an image (i.e., gridded arrays). Of course, for the method to work well, the images being registered should be sufficiently similar.

5.1 Preliminaries

5.1.1 Image Registration

Consider, for example, two grayscale images with intensities $u$ and $v$, given as functions on some domain $D$ (such as a rectangle in the plane, $\mathbb{R}^2$). For simplicity, assume that both $u$ and $v$ are equal to some constant at and near the boundary of the domain $D$, or at least can be extended beyond $D$ in a physically meaningful way. In our application, $u$ and $v$ will be heat flux fields from two states of our wildfire model, the fire will be inside the domain, and the heat flux near the boundary of the domain $D$ will be zero. In image processing, $u$ and $v$ can be the darkness levels of two photographs of objects with the same solid background. The functions $u$ and $v$ will be also referred to as images in the following description.
A mapping $T$ from $D \subset \mathbb{R}^2$ to $\mathbb{R}^2$

$$T : (x, y) \mapsto (T_x(x, y), T_y(x, y)),$$ \hfill (5.1)

such that

$$v \approx u \circ (I + T) \text{ on } D,$$

or, equivalently,

$$v - u \circ (I + T) \approx 0 \text{ on } D.$$ \hfill (5.2)

The mapping $I + T$ will be called the **registration mapping**, and the mapping $T$ will be called **warping**. The reason for writing the registration mapping as $I + T$ is that the zero warping $T = 0$ is the neutral element of the operation of addition, and so linear combinations of warpings have a meaningful interpretation as blends of the warpings. This will be important in the development of the morphing EnKF.

To avoid unnecessarily large and complicated warping, the warping $T$ should be as also close to zero and as smooth as possible,

$$T \approx 0, \quad \nabla T \approx 0,$$ \hfill (5.3)

where $\nabla T$ denotes the matrix of the first derivatives (the Jacobian matrix) of $T$,

$$\nabla T = \begin{pmatrix} \frac{\partial T_x}{\partial x} & \frac{\partial T_x}{\partial y} \\ \frac{\partial T_y}{\partial x} & \frac{\partial T_y}{\partial y} \end{pmatrix}.$$

In addition, we require that the registration mapping $I + T$ is one-to-one, so the inverse $(I + T)^{-1}$ exists. However, we do not require that the values of $I + T$ are always in $D$ or the inverse $(I + T)^{-1}$ is defined on all of $D$, so $u \circ (I + T)$ and $u \circ (I + T)^{-1}$ may not be defined on all of $D$. Therefore, we consider all functions $u,$
Figure 5.1: Shown in (a) and (b) are contour plots of the $x$ and $y$ components of an example morphing function, $T$. Figure (c) shows a quiver plot of $T$ indicating how an image would be warped when $T$ is applied to it.
Figure 5.2: Shown in (a) is an image $u$ made up of concentric circles. The morphing function given in Fig. 5.1 applied to this image, $u \circ (I + T)$. The resulting warped image is given in (b).

$v, u \circ (I + T), u \circ (I + T)^{-1}$, etc., extended on the whole $\mathbb{R}^2$ by the constant value of $u$ and $v$ on the boundary of $D$.

### 5.1.2 Morphing

Once the registration mapping $I + T$ is found, one can construct intermediate functions $u_\lambda$ between $u_0$ and $u_1$ by morphing (Fig. 5.4),

$$u_\lambda = (u + \lambda r) \circ (I + \lambda T), \quad 0 \leq \lambda \leq 1,$$

where

$$r = v \circ (I + T)^{-1} - u.$$  \hspace{1cm} (5.4)

The vector, $r$, will be called the registration residual; it is easy to see that $r$ is linked to the approximation in (5.2) by

$$r = (v - u \circ (I + T)) \circ (I + T)^{-1},$$
thus (5.2) also implies that $r \approx 0$.

The original functions $u$ and $v$ are recovered by choosing in (5.4) $\lambda = 0$ and $\lambda = 1$, respectively,

$$
u_0 = u \circ I = u, \quad (5.6)$$

$$
u_1 = (u + r) \circ (I + T) \quad (5.7)$$

$$
= (u + v \circ (I + T)^{-1} - u) \circ (I + T) \\
= v \circ (I + T)^{-1} \circ (I + T) = v.
$$

Remark 5.1. In the registration and morphing literature, the residual is often neglected. Then the morphed function is given simply by the transformation of the argument, $u \circ (I + \lambda T)$; however, this formula does not recover the image, $v$, when $\lambda = 1$. The simplest way to account for the residual might be to add a correction term to $u_\lambda$. This gives the morphing formula

$$
\tilde{u}_\lambda = u \circ (I + \lambda T) + \lambda (v - u \circ (I + T)), \quad 0 \leq \lambda \leq 1, \quad (5.8)
$$

which is much easier to use because does not require the inverse $(I + T)^{-1}$ like (5.4). The formula (5.8) also recovers $u = u_0$ and $v = u_1$, but, in our computations, we have found it unsatisfactory for tracking features and therefore we do not use it. The reason is that when the residual is not negligibly small, the intermediate functions $u_\lambda$ will have a spurious feature in the fixed location where the residual is large, cf. Figures 5.3 and 5.4. On the other hand, the more expensive morphing formula (5.4) moves the contribution to the change in amplitude along with the change of the position. To understand why this happens, consider a simple example in one dimension where we have some function $\varphi : \mathbb{R} \to \mathbb{R}$ so that $u(x) = \varphi(x)$, $v(x) = 2\varphi(x + 1)$, and
\( T(x) = 1 \). In this scenario, we have

\[
\begin{align*}
  r(x) &= v \left( (x + 1)^{-1} \right) - u(x) \\
  &= v(x - 1) - u(x) \\
  &= 2\varphi(x) - \varphi(x) = \varphi(x),
\end{align*}
\]

so that

\[
\begin{align*}
  u_\lambda(x) &= (u + \lambda r)(x + \lambda) \\
  &= (1 + \lambda)\varphi(x + \lambda) \\
  &= (5.9) \\
  &= (5.10)
\end{align*}
\]

from (5.5); however, if instead we used the form of (5.8), then

\[
\begin{align*}
  \tilde{u}_\lambda(x) &= u(x + \lambda) + \lambda(v(x) - u(x + 1)) \\
  &= \varphi(x + \lambda) + \lambda(2\varphi(x + 1) - \varphi(x + 1)) \\
  &= \varphi(x + \lambda) + \lambda\varphi(x + 1). \\
  &= (5.11)
\end{align*}
\]

Suppose that \( \varphi(x) \) is equal to 1 at 0 and 0 everywhere else (as in the Kronecker \( \delta_0(x) \)), then for \( \lambda \in (0, 1) \), \( u_\lambda(x) \) has one non-zero value at \( x = -\lambda \), but \( \tilde{u}_\lambda \) has two non-zero values at \( x = -\lambda \) and \( x = -1 \). Similarly, if \( \varphi \) contains a single peak, then \( \tilde{u}_\lambda \) will contain two peaks, one moving with \( \lambda \) and the other stationary. This example is illustrated in Figure 5.3. Figure 5.4 shows the same spurious feature appearing in our application, a 2-D fire simulation.

### 5.2 An Automatic Registration Procedure

The formulation of registration as (5.2) – (5.3) naturally leads to a construction of the mapping \( T \) by optimization. So, suppose we are given \( u \) and \( v \) and wish to find
Figure 5.3: The example given in Remark 5.1 is illustrated for $\varphi(x) = u(x)$, a hat function centered at 0, displayed in blue, and $v(x) = 2\varphi(x+1)$ is displayed in red. In (a) and (b), intermediate images are formed following (5.10), which uses the inverse morphing function, and (5.11), which is commonly used in applications, respectively, are displayed in black. The common method produces two features: $\varphi(x + \lambda)$, which is a translation of $u$ centered at $-\lambda$, and $\lambda\varphi(x + 1)$, which grows with $\lambda$, but remains centered at $-1$. The method requiring the inverse morphing function produces a single peak that translates and grows simultaneously. This is a result of the form of the residual in (5.5) which moves in conjunction with $u$. 
Figure 5.4: An example of the intermediate states of the morphing transformation applied to the heat flux emitted from two different fires. The morphing transformation is taken for both of the images and linear combinations of the two are formed, representing intermediate states in the morphing representation. The resulting states are transformed back into the standard representation. The two alternative formulations for the residual image yield different intermediate states. In the given figures, the top and bottom images are the two original fire lines. The intermediate states for the standard form of the residual (5.8) are given on the right and those of our representation (5.5) are given on the left. The standard method produces spurious features that remain centered on the location of the bottom fire. In our method, the large components of the residual move with the morphed image, so no spurious features are present.
a warping $T$ that is an approximate solution of

$$J(T, u, v) = \|v - u \circ (I + T)\| + C_1\|T\| + C_2\|\nabla T\| \rightarrow \min_T,$$  (5.12)

where the norms are chosen as

$$\|v - u \circ (I + T)\|^2 = \int_D |v - u \circ (I + T)|^2 dxdy,$$  (5.13)

$$\|T\|^2 = \int_D |T_x|^2 + |T_y|^2 dxdy,$$  (5.14)

$$\|\nabla T\|^2 = \int_D \left( \left| \frac{\partial T_x}{\partial x} \right|^2 + \left| \frac{\partial T_x}{\partial y} \right|^2 + \left| \frac{\partial T_y}{\partial x} \right|^2 + \left| \frac{\partial T_y}{\partial y} \right|^2 \right) dxdy.$$  (5.15)

The optimization formulation tries to balance the conflicting objectives of good approximation by the registered image, and as small and smooth warping as possible. The objective function $J(T, u, v)$ is in general not a convex function of $T$, and so there are many local minima. For example, a local minimum of may occur when some small part of $u$ and $v$ matches, while the overall match is still not very good. To solve this minimization problem, we have used an algorithm based on that described by Gao and Sederberg [1998] with several modifications. This algorithm differs from other automated image registration techniques in that $I + T$ is guaranteed to be invertible by construction. We use a similar construction here; however, because the invertibility explicitly relies on their use of bilinear interpolation, we can no longer make that guarantee. Instead, we attempt to force invertibility through a weak constraint in the objective function (5.12) by way of the norm on the gradient of $T$. We will also impose a explicit constraint based on the following condition on the partial derivatives of the morphing function, $T$:

$$\frac{\partial}{\partial x} T_x(x, y) > -1, \quad \frac{\partial}{\partial y} T_y(x, y) > -1,$$  (5.16)
for each \((x, y) \in D\). When this is satisfied, \(I + T\) is strictly monotonically increasing in each component, and, thus, invertible.

### 5.2.1 Grids and Sub-domains

In our application, the domain \(D\) is a rectangle, discretized by a uniform \(n_x \times n_y\) pixel grid with \(n = n_x n_y\) nodes. For simplicity, the domain will be represented in a normalized coordinate system, where \(D\) is mapped by an affine transformation into \([0, 1] \times [0, 1]\). In this coordinate system, the \((i, j)^{th}\) node has coordinates \(\left(\frac{i-1}{n_x-1}, \frac{j-1}{n_y-1}\right)\) (Figure 5.5b). Unless otherwise noted, the domain and coordinate system used will be assumed to be in normalized form. We will also split the normalized domain into increasingly refined sub-domains \(D_{i,j}^\ell\). This will be called the \((i, j)^{th}\) sub-domain at
level \( \ell \). The indices \( i \) and \( j \) will take on the fractional values \( \{1, 3/2, 2, \cdots, 2^\ell - 1/2, 2^\ell\} \). The refinement level \( \ell \) will range from 0 to some predetermined maximum \( L \). The 0\(^{th}\) level contains only one sub-domain, namely \( D_{0,1}^0 = D \). At the first level, \( D \) will be split into 4 standard sub-domains with integer indices:

\[
D_{1,1}^1 = \left[ 0, \frac{1}{2} \right] \times \left[ 0, \frac{1}{2} \right], \quad D_{2,1}^1 = \left[ \frac{1}{2}, 1 \right] \times \left[ 0, \frac{1}{2} \right],
\]

\[
D_{1,2}^1 = \left[ 0, \frac{1}{2} \right] \times \left[ \frac{1}{2}, 1 \right], \quad D_{2,2}^1 = \left[ \frac{1}{2}, 1 \right] \times \left[ \frac{1}{2}, 1 \right],
\]

and 5 offset sub-domains with fractional indices:

\[
D_{1,3}^1 = \left[ 0, \frac{1}{2} \right] \times \left[ \frac{1}{4}, \frac{3}{4} \right], \quad D_{2,3}^1 = \left[ \frac{1}{2}, 1 \right] \times \left[ \frac{1}{4}, \frac{3}{4} \right],
\]

\[
D_{1,4}^1 = \left[ \frac{1}{4}, \frac{3}{4} \right] \times \left[ 0, \frac{1}{2} \right], \quad D_{2,4}^1 = \left[ \frac{1}{4}, \frac{3}{4} \right] \times \left[ \frac{1}{2}, 1 \right],
\]

\[
D_{1,5}^1 = \left[ \frac{1}{4}, \frac{3}{4} \right] \times \left[ \frac{1}{4}, \frac{3}{4} \right].
\]

These are illustrated in Figure 5.6. The definition of each sub-domain can be expressed in the following explicit formula:

\[
D_{ij}^\ell = \left[ \frac{i - 1}{2^\ell}, \frac{i}{2^\ell} \right] \times \left[ \frac{j - 1}{2^\ell}, \frac{j}{2^\ell} \right],
\]

for each \( i, j = 1, \frac{3}{2}, \cdots, 2^\ell - \frac{1}{2}, 2^\ell \). So that each level contains \( (2^{\ell+1} - 1) \times (2^{\ell+1} - 1) \) sub-domains in total.

The sub-domains themselves do not have their own grid. Instead, they will share the pixel grid. When we refer to an image or morphing function restricted to a sub-domain, it is defined on the subset of pixel nodes located inside the sub-domain. In general, the pixel grid will not be aligned with the sub-domains nor will each sub-domain at a given level contain the same number of pixel nodes. Also, while the
Figure 5.6: The original domain $D$ is split into 9 different sub-domains of size $\frac{1}{2} \times \frac{1}{2}$ at refinement level $\ell = 1$. The sub-domains at this level are indexed by $D_{i,j}^1$ with $i,j \in \{1, \frac{3}{2}, 2\}$. Fractional indices indicate that the sub-domain is offset from the global domain’s edge by a distance of half of the sub-domain’s width in its respective axis. Further refinement levels are decomposed similarly.

The number of sub-domains grow exponentially at each level as $4^\ell$, the number of pixel nodes in each sub-domain decays exponentially as $4^{-\ell}$. Denote by $T|_{D_{ij}^\ell}$ the gridded array $T$ restricted to the domain $D_{ij}^\ell$. We will also use the notation $(x_{ij}^\ell, y_{ij}^\ell)$ and taken to mean the center point of the sub-domain $D_{ij}^\ell$.

When we evaluate a morphed image $u' = u \circ T$, we will take the value of the morphing function at each pixel grid point $T(x_i, y_j)$ and evaluate the original image by interpolation at the morphed location $u'(x_i, y_j) = u \circ T(x_i, y_j)$. In this way, the original image and the morphed image are defined on the same grid. Similarly, $u \circ T|_{D_{ij}^\ell}$ will be the morphed image restricted to the sub-domain $D_{ij}^\ell$. When evaluating a morphed image, it is necessary to obtain its values away from the pixel grid by interpolation. Two specific methods of interpolating the images will be considered.
here: bicubic spline and bilinear. While bilinear interpolation is significantly faster, morphed images created this way can exhibit noticeable kinks due to the general loss of differentiability of the interpolant. Thus, we will evaluate the morphed images using bicubic spline interpolation.

In order to reduce the chance that the minimization gets in a local minimum, the method proceeds by building up $T$ on the nested hierarchy of sub-domains $D_{ij}^\ell$. It begins with some initial guess $T = \tilde{T}$ of the optimal solution, which could be $\tilde{T} = 0$ if none is known. This initial solution is then repeatedly refined on each sub-domain at each refinement level by correcting the global solution inside each sub-domain. The corrections to the global solution will come in the form of shape functions with support inside their respective sub-domains. The role of this refinement procedure will be to align increasingly smaller features of the images, while limiting to dimension of the minimization problem on each sub-domain to a small subspace. This allows the registration procedure to align fine details in the images, while avoiding the issues inherent in large nonlinear minimization problems. The refinement continues until either the maximum number of refinement levels has been reached, $\ell = L$, or a sufficiently accurate solution has been found. In the computation, (5.13), (5.14), and (5.15) are integrated numerically on each of the sub-domains $D_{ij}^\ell$, with the derivatives approximated by finite differences, and the integrals are approximated by piece-wise constant quadrature. Denote the objective function restricted to the grid $D_{ij}^\ell$ by $J_{ij}^\ell(T, u, v)$. On each sub-domain $D_{ij}^\ell$, the method proceeds as follows. In order not to overload the notation with many iteration indices, the values of $T$, and thus also $T_{D_{ij}^\ell}$, change during the computation just like a variable in a computer program.

1. Initialize $T = \tilde{T}$. 

87
2. Smooth \( u \) and \( v \) by convolution to get \( \tilde{u}_\ell \) and \( \tilde{v}_\ell \).

3. Minimize the objective function \( J_{ij}^\ell(T, \tilde{u}_i, \tilde{v}_i) \) in the span of some shape functions with the constraint that \((I + T)\) is invertible.

### 5.2.2 Smoothing

The purpose of doing the registration on the large sub-domains first is to capture coarse similarities between the images \( u \) and \( v \). In order to force this, on each grid, we first smooth the images by convolution with a Gaussian kernel. The bandwidth of the kernel, \( h_\ell \) is reduced at each refinement level allowing the registration to track large scale perturbations on coarse levels even for a thin feature such as a fireline, while maintaining small scale accuracy at fine levels. The precise values one should use for \( h_\ell \) is in general a parameter that depends on the problem at hand. However, a good rule of thumb is to have \( h^0 \) scale with the ratio of the feature size over the global domain size. For all other refinement levels, we set \( h_\ell = 2^{-\ell}h^0 \) so that fine scale detail present in each sub-domain scales with its size. If the original image domain is not square, it is helpful to use different bandwidths in the \( x \) and \( y \) axes so the smoothing is consistent with the true spatial scaling due to the numerical use of a normalized, square grid. In this case, we will use bandwidth in the \( x \) and \( y \) axes denoted \( h^\ell_x \) and \( h^\ell_y \) respectively, where \( h^\ell_x = \frac{D_y}{D_x} h^\ell_y \) and \( D_x \) and \( D_y \) are the physical sizes of the domain in each axis. The convolution is defined by the discrete Fourier transform:

\[
\tilde{u}_\ell(x_i, y_j) = \sum_{i' = 1}^{n_x} \sum_{j' = 1}^{n_y} \hat{\phi}_i(x_i') u(x_i', y_j') \hat{\psi}_j(y_j'),
\]  

(5.17)
A direct implementation of (5.17) is, however, very inefficient because it requires $O \left( (n_x n_y)^2 \right)$ floating point operations to compute the convolution $u \star (\hat{\phi} \otimes \hat{\psi})$. The same operation can be performed much faster in the frequency domain using the relation
\[
u \star (\hat{\phi} \otimes \hat{\psi}) = F^{-1} \left( F(u) \cdot F(\hat{\phi} \otimes \hat{\psi}) \right),
\] (5.18)
where $F(u)$ denotes the Fourier transform of $u$. Using FFT to compute the forward and inverse Fourier transforms, the convolution can be computed in only $O \left( (n_x n_y) \log (n_x n_y) \right)$ operations.

5.2.3 Local minimization

Finding a global minimum of $J_{ij}^\ell(T, \tilde{u}_\ell, \tilde{v}_\ell)$ is generally too complicated to achieve using traditional minimization techniques directly. The problem is highly nonlinear, has many unknowns (i.e. twice the number of grid nodes inside $D_{ij}^\ell$), and has a complicated feasibility region (in which $I + T$ is invertible). Because we are working with a multi-level algorithm, the solution will be refined at higher levels, and we can seek to solve the problem in a subset of all feasible $T|_{D_{ij}^\ell}$, which is of low dimension and vastly simpler constraints. The subset in which we seek to minimize the local objective function will be of the form
\[
u + \text{span}_{k=1,K} \left\{ B_k P^\ell_{ij} \right\},
\]
Figure 5.7: Displayed in (a) is a function defined over \([0, 1]\) containing a thin feature similar to the reaction front of a wildfire. This function is smoothed using equation 5.18 with various smoothing bandwidths which would be used at increasing refinement levels, \(\ell\), where the bandwidth decreases as \(2^{-\ell}\). The larger bandwidth at lower levels increases the ability of the registration algorithm to find the global minimum of the objective function. Decreasing the bandwidth on the same scale of the refinement enables the algorithm to retain accuracy as the algorithm attempts to align increasingly fine scale detail.
where each $B_k$ is a shape function defined over $D$ and $P^\ell_{ij}$ is the affine transformation that maps $D^\ell_{ij}$ to $[-1, 1] \times [-1, 1]$. A simple example of this would be constant shape functions: $B_1 : (x, y) \mapsto (1, 0)$ and $B_2 : (x, y) \mapsto (0, 1)$. Because these shape functions are non-zero on the boundary of $D$, they will not preserve the global continuity of $T$. An important example of shape functions that will preserve continuity are cones formed by bilinear interpolation of $3 \times 3$ uniform grid with values $0$ on the boundary, $(1, 0)$ and $(0, 1)$ at the center. If the pixel grid has dimensions $n_x = n_y = 2^{L+1} + 1$, then each sub-domain will contain a pixel node at the side, corners, and center. In this case, the algorithm presented here without image smoothing is equivalent to that presented in Gao and Sederberg [1998]. This method of interpolation has two major benefits. Bilinear interpolation is very fast. There is also a simple $O(1)$ method of maintaining invertibility of $T$ by assuring that $T \circ P^\ell_{ij}(0, 0)$ is strictly inside the quadrilateral formed by the image of $T \circ P^\ell_{ij}$ at $(-1, 0), (1, 0), (0, -1)$, and $(0, 1)$. When this condition is satisfied, then the morphing function maps each rectangular grid element into a convex quadrilateral, which is a necessary and sufficient condition for its invertibility [Frey et al., 1978]. However, because these shape functions have non-zero normal derivative on the boundary, global differentiability will not be preserved.

We have chosen to use the interpretation of the corrections on each sub-domain as a deformation of the center point, $P^\ell_{ij}(0, 0)$, which has two degrees of freedom to be represented in $B_1$ and $B_2$. Because we to maintain both the continuity and the differentiability of $T$, we will fix the value and gradient of $T$, on the boundary, to $0$. A simple set of functions that meet these restrictions are created by a tensor product
cubic spline,
\[ S(x) = 2|x|^3 - 3x^2 + 1, \]
\[ B_1(x, y) = (S(x)S(y), 0), \]
\[ B_2(x, y) = (0, S(x)S(y)). \]
This form of shape function allows us to create a continuously differentiable morphing function that moves the center point of an arbitrary sub-domain, remaining zero outside of that sub-domain. For example, \( T = (c_1B_1 + c_2B_2) P_{ij}^\ell \) defines such a morphing function that moves the center of sub-domain \( D_{ij}^\ell \) by \( (c_1, c_2) \). Figure 5.9 illustrates the application of various methods of interpolation to a standard test image.

5.2.4 The Levenberg-Marquardt Algorithm

In this section, we introduce an algorithm used for minimizing more general classes of nonlinear functions known as the Levenberg-Marquardt Algorithm (LMA) originally described by Levenberg [1944] and Marquardt [1963]. Given a function \( f : \mathbb{R}^n \to \mathbb{R}^p \) with \( p \geq n \), which has continuous second partial derivatives, we seek to find an \( x^* \) which is a local minimum of
\[ F(x) = \frac{1}{2} |f(x)|^2 = \frac{1}{2} f(x)^T f(x). \] (5.20)

We will denote the gradient of \( F \) at \( x \) as \( F'(x) \), which is a row vector of size \( n \) with components
\[ (F'(x))_j = \frac{\partial F}{\partial x_j}(x), \]
and the Hessian of \( F \) at \( x \), \( F''(x) \), a matrix of size \( n \times n \) with components
\[ (F''(x))_{ij} = \frac{\partial^2 F}{\partial x_i \partial x_j}(x). \]
Sufficient conditions that \( x^* \) is a local minimizer are \( F'(x^*) = 0 \) and \( F''(x^*) \) is positive definite [Frandsen et al., 1999].
Figure 5.8: (a) The morphing shape function $S(x)$ used in (5.19) defined as a cubic spline with $S(x) = 0$ and $S'(x) = 0$ on the boundary and $S(x) = 1$ and $S'(x) = 0$ at the center, $x = 0$. (b) The tensor product of the shape functions that make up the components of $B_1$ and $B_2$. The gradient of this tensor product is zero along the boundary, and it attains a maximal value of 1 at the center, $(0, 0)$. (c) and (d) Quiver plots showing the deformation of the domain performed by each shape function scaled by 0.2.
Figure 5.9: These figures illustrate the effect of various methods of interpolating a morphing function onto the pixel grid. (a) shows the original image with four arrows indicating the specified perturbations of the image. The remaining images show the morphed image with the specified perturbations interpolated to the pixel grid with (b) nearest neighbor, (c) bilinear, and (d) tensor product B-spline interpolants. Figure (b) exhibits clear discontinuities, while Figure (c) creates kinks in what were originally smooth curves.
The Gauss-Newton method is an iterative procedure, where the current guess of minimum, \( x \), is improved by the increment, \( \delta x \). This increment is determined by approximating the solution to

\[
F'(x + \delta x) = 0. \quad (5.21)
\]

The gradient in this expression is approximated using the Taylor expansion of \( f \),

\[
f(x + \delta x) = f(x) + f'(x)\delta x + O(|\delta x|^2).
\]

Omitting the second order term, we have

\[
F'(x + \delta x) = f'(x + \delta x)^T f(x + \delta x) \approx f'(x)^T f(x) + f'(x)^T f'(x)\delta x.
\]

Substituting this approximation in (5.21) gives us the equation for the increment of the Gauss-Newton method

\[
f'(x)^T f'(x)\delta x = -f'(x)^T f(x). \quad (5.22)
\]

When the current iterate is sufficiently close to the minimum, one can expect at least linear convergence; however, in general, this method may fail to converge at all [Björck, 1996]. In addition, when \( f'(x)^T f'(x) \) is singular, a solution to this equation may not exist.

Damped Gauss-Newton methods attempt to solve these problems by modifying the increment equation (5.22) so that

\[
(f'(x)^T f'(x) + \alpha I)\delta x = -f'(x)^T f(x), \quad (5.23)
\]

for \( \alpha > 0 \). In this case, \( f'(x)^T f'(x) + \alpha I \) is symmetric positive definite, and a unique solution exists. In addition, when \( \alpha \) is small, the increment becomes approximately
equal to that of the undamped Gauss-Newton method. When \( \alpha \) is large,
\[
\delta x \approx -\frac{1}{\alpha} f'(x)^T f(x) = -\frac{1}{\alpha} F'(x),
\]
and the increment is a small step in the direction of steepest descent.

The LMA uses the damped increment equation (5.23), where \( \alpha \) is determined adaptively so that it is large when the iterate is far away from the minimum and small when it is close. For this, the LMA is often considered a combination between the Gauss-Newton method, with fast local convergence, and steepest descent method, with more reliable global convergence [Frandsen et al., 1999]. This property makes the LMA popular in many nonlinear optimization applications including image registration [Zitová and Flusser, 2003].

While the details vary among implementations, generally a single iteration consists of computing the increment \( \delta x \) and
\[
\rho \leftarrow \frac{F(x) - F(x + \delta x)}{\frac{1}{2} \delta x^T (\alpha \delta x - F'(x))}.
\]
The numerator of this expression is the improvement in the value of \( F \) from the increment, and when it is negative, indicates that the increment has degraded the solution. The denominator is the estimated improvement using the linearized version of \( f \) and is always positive. When \( \rho \) is negative, the iteration is rejected and \( \alpha \) is increased for the next iteration. When \( \rho \) is positive, the iteration is accepted and the damping parameter is reduced.

**5.2.5 Optimization**

Because the shape functions were designed to be zero outside of their respective sub-domains, we have the relation
\[
J(T, \bar{u}^T, \bar{v}^T) - J(T + c_k B_{k} P_{ij}, \bar{u}^T, \bar{v}^T) = J_{ij}^i (T, \bar{u}^T, \bar{v}^T) - J_{ij}^i (T + c_k B_{k} P_{ij}, \bar{u}^T, \bar{v}^T).
\]
Therefore on each subdomain, minimizing the global objective function in the span
of the shape functions is equivalent to minimizing the local objective function over
the same span. Within each sub-domain, the registration problem is reduced to mini-
mizing
\[ g(c_1, c_2) = J_{ij}^\ell \left( T + c_1 B_1 P_{ij}^\ell + c_2 B_2 P_{ij}^\ell, \tilde{u}^\ell, \tilde{v}^\ell \right) \]
constrained in a region \((c_1, c_2) \in C \subset \mathbb{R}^2\) such that \(I + T + c_1 B_1 P_{ij}^\ell + c_2 B_2 P_{ij}^\ell\)
is invertible. We will seek to perform this minimization using two complementary
methods. The first involves sampling in the region \(C\) similar to the method described
in Gao and Sederberg [1998]. The second method uses an implementation of the
Levenberg-Marquardt algorithm (LMA) (described in Section 5.2.4) released as part
of MINPACK [Moré et al., 1980].

The sampling method described here is intended to prevent the LMA minimiza-
tion procedure from converging to a local minimum. This is particularly relevant at
cosmic refinement levels. Rather than initializing the LMA at just one value, we run it
multiple times starting from a uniform grid of values inside \(C\). However, the region
\(C\) can be quite complex in general, and determining it would require looping over all
nodes inside the local region. To avoid the additional complexity in determining this
region explicitly, we will approximate it by a quadrilateral \(\tilde{C} \subset C\) defined by the
following vertices on the principal axes \((c^x, 0), (c_x, 0), (0, c^y),\) and \((0, c_y),\) with
\[ c^x = \alpha \frac{2^{-\ell}}{3} \frac{\partial}{\partial x} T \left( x_{ij}^\ell + \frac{1}{2^{\ell+2}}, y_{ij}^\ell \right), \quad c^y = \alpha \frac{2^{-\ell}}{3} \frac{\partial}{\partial y} T \left( x_{ij}^\ell, y_{ij}^\ell + \frac{1}{2^{\ell+2}} \right), \]
\[ c_x = -\alpha \frac{2^{-\ell}}{3} \frac{\partial}{\partial x} T \left( x_{ij}^\ell - \frac{1}{2^{\ell+2}}, y_{ij}^\ell \right), c_y = -\alpha \frac{2^{-\ell}}{3} \frac{\partial}{\partial x} T \left( x_{ij}^\ell, y_{ij}^\ell - \frac{1}{2^{\ell+2}} \right), \]
(5.25)
where \(\alpha < 1\) is a contraction parameter. The definitions for the other vertices follow
in a similar manner. These ad-hoc definitions come about from the constraint given
in (5.16) and by the fact that \(|\frac{\partial}{\partial x} B_1|\) is maximal on \(y = 0\) at \(x = -\frac{1}{2}\) and \(x = \frac{1}{2}\).
Then, solving

$$\frac{\partial}{\partial x} \left( I + T + e^x B_1 P_{ij} \right) \left( x_{ij} \ell + \frac{1}{2^{\ell+1}}, y_{ij} \ell \right) = -1$$

(5.26)

for $c^x$ lead to the given definition. However, depending on $T$, the quadrilateral $\tilde{\mathcal{C}}$ may fail to be a subset of $\mathcal{C}$. For this reason, the contraction parameter $\alpha$ is included. Experimentally, we have determined that $\alpha = \frac{3}{4}$ is small enough to ensure that $\tilde{\mathcal{C}} \subset \mathcal{C}$, but there is no reason that this will be true in general (cf. Figure 5.11).

Once the region $\tilde{\mathcal{C}}$ has been determined, a set of sampling points is defined as

$$\mathcal{S} = \{(\kappa_{x i_x}, \kappa_{y i_y})\}_{i_x, i_y \in \mathbb{Z}} \cap \tilde{\mathcal{C}}.$$

(5.27)

Since we assume that the initial morphing function $T$ is invertible, then $(0, 0) \in \mathcal{S}$. The sampling grid resolutions $\kappa_{x}$ and $\kappa_{y}$ are global optimization parameters that will generally be problem dependent. As a general rule, these parameters should be set to approximately the ratio of the feature width over the global domain size in each axis. For instance, in our application, the global domain will be about $1 \text{ km}$ and the width of an ignition region is about $10 \text{ m}$, so we would use a sampling grid resolution of $0.01$. (In practice, we have found that a resolution of $0.1$ is suitable because the image smoothing reduces the chance of converging to a local minimum.) In this case, the global domain may have several thousand sample points; however, the number of sample points in $\mathcal{S}$ will decay on the order of $4^{-\ell}$ as a function of the refinement level $\ell$. This is due to the fact that the area of $\mathcal{C}$ must be strictly smaller than the area of the local sub-domain, whose area decays as $4^{-\ell}$. This procedure is displayed graphically in Figure 5.11.
Figure 5.10: (a) The derivative of $S(x)$ from Fig. 5.8a has maximal and minimal values at $-0.5$ and $0.5$. The partial derivatives of Fig. 5.8b in the $x$ and $y$ directions are given in (b) and (c), respectively. These partial derivatives have extreme values at $(\pm 0.5, 0)$ and $(0, \pm 0.5)$, where they attain slopes of $\pm 1.5$. Equation (5.16) is used to provide approximate conditions for the invertibility of $I + T + c_1 P_{ij}^T B_1$ by finding bounds for $c_1$ such that $\frac{\partial}{\partial x} (I + T + c_1 P_{ij}^T B_1) > -1$ at $(\pm 0.5, 0)$. The resulting bounds give rise to the conditions appearing in (5.25).
Figure 5.11: The black dots represent the sampling grid defined in (5.27) as a uniform grid with resolution $\kappa_x \times \kappa_y$. The region outlined in red, $C$, is defined as the feasibility region of the local minimization problem (5.24). The points marked with a black square are determined by solving (5.26) and represent the intersection of $C$ with the $x$ and $y$ axes. The distance of these points from the origin are multiplied by the contraction parameters, $\alpha_x$ and $\alpha_y$, to obtain the blue dots located at $(c_x, 0)$, $(0, c_y)$, $(c_x, 0)$, and $(0, c_y)$. The coordinates for these dots are explicitly given in (5.25) and define the vertices of the quadrilateral, $\tilde{C}$, outlined in blue. The sampling points, $S$, circled in blue are determined as the intersection of the sampling grid with the region $\tilde{C}$. The dashed blue line represents what the sampling region would be if we did not include the contraction parameters. This region is not a subset of $\tilde{C}$, and, in particular, contains the sampling point circled in red, which is not in the feasible region of the problem.
Finally, the LMA is performed starting at each point in $S$. The function, $f$ in (5.20), that is minimized in the least-square sense is defined as

$$f(c_1, c_2) = \begin{bmatrix} v - u \circ (I + \tilde{T}) \\ \sqrt{C_1 \tilde{T}} \\ \sqrt{C_2 \nabla \tilde{T}} \end{bmatrix}, \quad \tilde{T} = T + c_1 B_1 P_{ij}^e + c_2 B_2 P_{ij}^e,$$

where range is a single, long vector made up of all components inside the brackets. This function is chosen so that the LMA minimizes $F = f^T f$, which is similar to the square of the registration local objective function (5.24). In fact, one can view these to norms as being approximately equivalent because we have assumed a uniform mesh and the vector two norm can be viewed as proportional to piecewise constant numerical integration with error on the order of the square of the mesh resolution. We expect that the minima of both (5.20) and (5.12) to be approximately aligned.

This procedure results in a set of local minima for the local optimization problem. The solution $(c_1, c_2)$ that results in the minimum value of $g$ is then used to update the global morphing function,

$$T \leftarrow T + c_1 B_1 P_{ij}^e + c_2 B_2 P_{ij}^e.$$

The LMA itself takes several parameters including a tolerance $\tau$. This tolerance is treated as another global parameter for the registration algorithm. As the refinement level increases, the size of the sub-domain decreases, and thus the number of iterations of the LMA performed also decreases since the tolerance is kept constant. The algorithm then moves on to the next sub-domain.

In practice, the order in which the sub-domains are optimized tends to affect the registered morphing function. The local optimization is limited to correcting the
morphing function in the span of the shape functions, which are large in the center and small near the boundary. This structure tends to force the local optimization to over estimate the perturbation of $T$ in the center of the sub-domain while under estimating the perturbation near the edges. The staggering of the sub-domains at each refinement level means that a corner of one sub-domain (in the interior of $D$) will actually be the center of another. So the staggering tends to stabilize the registration somewhat, but the over estimation will still occur in the center of the sub-domains that were optimized first. This effect can be significantly reduced by looping over all sub-domains twice before moving on to the next refinement level.

5.2.6 Computational Complexity

The registration algorithm is summarized here for a complexity analysis.

1. Loop over all refinement levels, $\ell = 1, \cdots, L$.

   (a) Smooth the images by convolution using FFT.

   (b) Loop over all sub-grids $D_{ij}^\ell$ and $\hat{D}_{ij}^\ell$ at the refinement level $\ell$.

      i. Perform LMA at each sample point in $S$ for the local subgrid.

   (c) Update $T$ using the correction found in the local optimization procedure.

The algorithm then requires $L$ FFT’s each of size $O(n \log n)$ operations, where $n = n_x n_y$. In addition, there are $O(4^\ell)$ sub-domains at each refinement level $\ell$. The local optimization procedure requires several evaluations of the local objective function, each of which require $O(n 4^{-\ell})$ operations. The number of evaluations of the local objective function is some multiple of the number of sample points depending on number of iterations of the LMA performed. Therefore each refinement level requires
$O(n)$ operations for the local optimization procedure over all sub-domains. While the total number of refinement levels necessary for an accurate registration will depend on the resolution of the images, we require that each sub-domain contain at least one image pixel, which necessarily bounds the maximum number of refinement levels by $n \lg n$. Therefore the algorithm as described requires $O(n \lg^2 n)$ operations in total.

In the production of the numerical results in Chapter 6, we have found that an implementation of this algorithm in Fortran is capable producing acceptable registrations for our wildfire application, even when the fires in the images are separated by as much 1 km. The residual images produced in these experiments have large components only near the fireline of the moving image $u$. The morphed images lack any noticeable spurious features. In addition, the computational time for producing these registrations is acceptable for use in a real time application. The images used in these examples have a resolution of $411 \times 411$ defined on an approximately 2.5 km square physical domain, with 5 subdomain refinement levels, and a 250 m sampling grid resolution. The registrations using these parameters were produced in about 2 minutes when starting from $T = 0$. When a good initial guess of the initial morphing function is provided, the registration is performed in about 30 s.
6. The Morphing Ensemble Kalman Filter

In Section 6, we observed that standard EnKF approach fails for our wildfire model when the forecast ensemble involves spatial errors. This behavior is consistent with other applications involving sharp features which evolve in time, such as rain fronts or vortices [Chen and Snyder, 2007]. This can be ameliorated to some degree by penalization of nonphysical states [Johns and Mandel, 2008], localization of EnKF [Anderson, 2003, Ott et al., 2004], and employing the location of the feature as an observation function [Chen and Snyder, 2007], but the basic problem remains: EnKF works only when the increment in the location of the feature is small [Chen and Snyder, 2007].

While the location and the size of the feature may have an error distribution that is approximately Gaussian, this is not necessarily the case for the value of the state at a given point. There is clearly a need to adjust the simulation state by distorting the simulation state in space rather than employing an additive correction to the state. Therefore, alternative error models that include the position of features were considered in the literature [Hoffman et al., 1995, Davis et al., 2006] and a number of works emerged that achieve more efficient movement of features by using a spatial transformation as the field to which additive corrections are made, such as a transformation of the space by a global low order polynomial mapping to achieve alignment [Alexander et al., 1998], and two-step models to use alignment as preprocessing to an additive correction [Lawson and Hansen, 2005, Ravela et al., 2007].
The essence of the new method described here is to replace the linear combinations of states in an ensemble filter by intermediate states obtained by morphing. This method provides additive and position correction in a single step. For the analysis step (the Bayesian update), the state is transformed into an extended state consisting of additive and position components. After the analysis step, the state is converted back and advanced in time.

6.1 The Morphing Transformation

For simplicity, we will assume that the state of the model consists of several gridded variables, \( X = (w, z, \ldots) \), all arrays are defined over the same grid, and the registration is applied only to the first variable, \( w \); this will be the case in the model application in Section 6.2. The general case will be discussed in Section 6.3.

Let \( [X_k] = \{X_1, \ldots, X_N\} \) be an ensemble of states, with the ensemble member \( X_k \) consisting of the gridded arrays

\[
X_k = (w_k, z_k, \ldots).
\]

Given one fixed state \( X_0 = (w_0, z_0, \ldots) \), the automatic registration (5.12) of the first array \( w \) defines the *registration representations* \([R_k, T_k]\) of the ensemble members as morphs of \( X_0 \), with the registration residual

\[
R_k = (r_{w_k}, r_{z_k}, \ldots),
\]

\[
r_{w_k} = w_k \circ (I + T_k)^{-1} - w_0,
\]

\[
r_{z_k} = z_k \circ (I + T_k)^{-1} - z_0,
\]

\[
\vdots
\]
and warpings $T_k$ determined as approximate solutions of independent optimization problems based on the state array $w$,

$$
\|w_k - w_0 \circ (I + T_k)\| + \|T_k\| + \|\nabla T_k\| \rightarrow \min_{T_k}.
$$

The mapping $T_k$ from the previous analysis cycle is used as the initial $\tilde{T}_k$ in the automatic registration. In our tests, this all but guarantees good registration for the ensemble members and a significant speedup compared to starting from zero.

Instead of EnKF operating on the ensemble $\{X_k\}$ and making linear combinations of its members, the morphing EnKF applies the EnKF algorithm to the ensemble of registration representations $\{[R_k, T_k]\}$, resulting in the analysis ensemble in registration representation, $\{[R^a_k, T^a_k]\}$, with $R^a_k = (r^a_{w_k}, r^a_{z_k}, \ldots)$ containing all model fields, not just the registration variable. The analysis ensemble is then transformed back by (5.7), which here becomes

$$
w^a_k = (w_k + r^a_{w_k}) \circ (I + T^a_k),
$$

$$z^a_k = (z_k + r^a_{z_k}) \circ (I + T^a_k),
$$

...  

Given an observation function $h$, the transformed observation function for EnKF on the registration representations can be obtained directly by substituting from (6.1) into the observation function,

$$
\tilde{h} ([R, T]) = h((w + r_w) \circ (I + T), (z + r_z) \circ (I + T), \ldots).
$$

However, constructing the observation function this way may not be the best. Consider the case of one point observation, such as the temperature at some location.
Then the difference between the observed temperature and the value of the observation function gives little indication which way should the transformed state be adjusted. Suppose the temperature reading is high and the ensemble members have high temperature only in some small location (fireline). Then it is quite possible that the observation function (temperature at the given location) evaluated on ensemble members will miss the fire in the ensemble members completely. This is, however, a reflection of the inherent difficulty of localizing small features from point observations.

For data that is given as gridded arrays (e.g., images, or a dense array of measurements), there is a better way. Suppose the data $d$ is a measurement of the first array in the state, $w$. Then, transforming the data $d$ into its registration representation $[r_d, T_d]$ just like the registration of the state array $w$, the observation equation becomes the comparison between the registration representations of the data $d$ and the state array $w$,

$$\hat{h}([R, T]) = [r_d, T_d] \approx [r_w, T]. \quad (6.3)$$

Data given on a part of the domain can be registered and used in the same way. Note that no manual identification of the location of the feature either in the data or in the ensemble members is needed.

### 6.2 Numerical Results

We have applied the morphing EnKF to an ensemble from our PDE wildfire model Mandel et al. [2005]. The simulation state consists of the temperature and the fuel fraction remaining on a $500 \times 500$ $m$ domain with a $2$ $m$ uniform grid. The model has two state arrays, the temperature $w$ and fuel supply $z$. An initial fire solution $X_0$ was created by raising a small square in the center of the domain above the
ignition temperature and applying a small amount of ambient wind until a fire front
developed. The simulated data consisted of the whole temperature field of another so-
lution, started from $U_0$ and morphed so that the fire was moved to a different location
of the domain compared with the average location of the ensemble members. The
observation equation (6.3) was used, with Gaussian error in the registration residual
of the temperature and in the registration mapping. The standard deviation of the
data error was 50 $K$ and 5 $m$, respectively. This large discrepancy is used to show
that the morphing EnKF can be effective even when data is very different from the
ensemble mean. The image registration algorithm was applied with $L = 4$ refine-
ment levels. We have performed up to 5 optimization sweeps, stopping if the relative
improvement of the objective function for the last sweep was less than 0.001 or if the
infinity norm of the residual $r_w$ fell below 1 $K$. The optimization parameters used
for scaling the norms in the objective function (5.12) were $C_1 = 10000 \, mK^{-1}$ and
$C_2 = 1000 \, m^2K^{-1}$. For simplicity in the computation, the fuel supply variables were
not included in the data assimilation. Although the fuel supply was warped spatially
as in (6.2), the registration residual of the fuel supply, $r_z$, was taken to be zero.

The 50 member ensemble shown in Fig. 6.1 was generated by morphing the initial
state $X_0$ using smooth random fields $r_{wk}$ and $T_k$, with smoothness parameter $\alpha = 1,$
restricted to 0 on the boundary. Since it is not guaranteed that $(I + T)^{-1}$ exists for
a smooth random $T$, we have tested if $I + T$ is one to one and generated another
random $T$ if not. The resulting two $250 \times 250$ matrices are appended to form $2 \times 250^2$
element vectors representing an ensemble state $[R_k, T_k]$ for the EnKF. The same state
$U_0$ was advanced in time along with the ensemble. (Of course, other choices of $U_0$
are possible.)
Figure 6.1: Data assimilation by the morphing ensemble filter. The forecast ensemble (b) was created by smooth random morphing of the initial temperature profile located near the center of the domain. The analysis ensemble (c) was obtained by the EnKF applied to the transformed state. The data for the EnKF was the morphing transformation of the simulated data (a), and the observation function was the identity mapping. Contours are at 800 K, indicating the location of the fireline. The reaction zone is approximately between the two curves.

The ensemble was advanced in 3 minute analysis cycles. The new registration representation \([r_k, T_k]\) was then calculated using the previous analysis values as an initial guess and incremented by EnKF. The ensemble after the first analysis cycle is shown in Fig. 6.1. The results after five analysis cycles were similar, indicating no filter divergence (Fig. 6.2). Numerical results indicate that the error distribution of the registration representation is much closer to Gaussian than the error distribution of the temperature itself. This is demonstrated in Fig. 6.3, where the estimated probability density functions for the temperature, the registration residual of the temperature, and the registration mapping for Fig. 6.2 are computed at a single point in the domain using a Gaussian kernel with bandwidth 0.3 times the sample standard deviation. The point was chosen to be on the boundary of the fire shape, where the non-Gaussianity may be expected to be the strongest. In Fig. 6.4, the Anderson-Darling test for normality was applied to each point on the domain for the analysis step from Fig. 6.2c.
Figure 6.2: After five analysis cycles, the ensemble shows less spread and follows the data reliably. Ensemble members were registered using the initial state, advanced in time without data assimilation. The forecast ensemble (b) is closer to the simulated data (a) because of preceding analysis steps that have attracted the ensemble to the truth. The analysis ensemble (c) has a little less spread than the forecast, and the change between the forecast and the analysis is well within the ensemble spread. Contours are at 800 K, indicating the location of the fireline. The reaction zone is approximately between the two curves.

Figure 6.3: Probability densities estimated by a Gaussian kernel with bandwidths 37 K, 19 K, and 30 m. Data was collected from the ensemble shown in Fig. 6.1b. Displayed are typical marginal probability densities at a point near the reaction area of the original temperature (a), the registration residual of temperature (b), and the registration mapping component in the \( x \) direction (c). The transformation has vastly improved Gaussian nature of the densities.
Figure 6.4: The \( p \)-value from the Anderson-Darling test of the data from the ensemble after five morphing EnKF analysis cycles shows the ensemble transformed into its registration representation, the registration residual of the temperature (b) and the registration mapping (c), has distribution much closer to Gaussian than the original ensemble (a). The shading at a point indicates where the marginal distribution of the ensemble at that point is highly Gaussian (white) or highly non-Gaussian (black).

The resulting \( p \)-values were plotted on their corresponding locations in the domain with darkness determined on a log scale with black as \( 10^{-8} \) (highly non-Gaussian) and white \( 1 \) (highly Gaussian). While the Anderson-Darling test is intended to be a hypothesis test for normality, it is used here to visualize on a continuous scale the closeness to normality of the marginal probability distribution any point in the domain. Again, strongest departure from normality of the distribution is seen around the fire.

Figure 6.5 shows a 3-D representation of a similar experiment with the PDE model. The temperature profiles are displayed as a superposition of transparent images to give an idea of both the shape analysis ensemble members and the variance of their position. The same ensembles and data were used for the EnKF both with and without morphing. The standard EnKF produces a degenerate ensemble made up of
highly non-physical states, while the morphing EnKF ensemble members retain the approximate shape of a fire profile.

In Figure 6.6, this experiment was also applied to the stand alone level set fire model, where the sensible heat flux from the ground was used for registration and data. The results are similar to the PDE model where the EnKF produces a degenerate non-physical result and the morphing EnKF create something far more reasonable. Finally, the experiment was repeated using the coupled fire model in Figure 6.7. Again, the sensible heat flux is used as the registration variable. The atmospheric winds were included in the ensemble states. The figures contours of the heat fluxes for all ensemble members superimposed on the $x$-$y$ plane. In addition, the mean of the vertical vorticity, defined as the curl of the wind velocity in the atmosphere, is displayed in red (positive vorticity) and blue (negative vorticity). The heat flux contours in the EnKF analysis indicate similar non-physical results, while the morphing EnKF is able to move the burning areas near the data. In addition, the wind turbulence was also translated to better match the data.

While the experiments described so far show that the morphing EnKF yeilds a significant improvement in the creation of ensembles that retain spatial error and physical features, the conclusion is based on a subjective analysis of a single realization of the filter. It would be desirable to obtain some sort of a metric that describes these qualities and can be averaged over multiple realizations. Unfortunately, there is no easy method for obtaining such statistics. As we have seen, describing the error using the pointwise mean and covariance of the model variables is insufficient for describing the error distributions involved. Even using the model variables in the morphing representation to collect these statistics is unlikely to be effective because

112
Figure 6.5: The morphing EnKF applied to the reaction-diffusion model. The false color is generated from the temperature with shading for depth perception. The reference solution (a) is the simulated data. The initial ensemble is created by a random perturbation of the comparison solution (b), where the fire is ignited at an intentionally incorrect location. The standard EnKF panel (c) is the result of data assimilation of the temperature field after running the model for 500 seconds. The morphing EnKF panel (d) is the ensemble with the image registration against the temperature field at the time of ignition, and the morphing applied to both the temperature and the fuel supply. The ensembles have 25 members each, and are visualized as the superposition of transparent images of their members. The observation function is the temperature field on the whole grid. The standard EnKF ensemble diverges from the data, while the morphing EnKF ensemble remains closer to the data.
Figure 6.6: The morphing EnKF applied to the fireline propagation model. The false color is the output heat flux with shading for depth perception. The reference solution (a) is the simulated data. The initial ensemble is created by a random perturbation of the comparison solution (b), where the fire is ignited at an intentionally incorrect location. The standard ENKF panel (c) is the result of data assimilation of the time from ignition after running the model for 1000 seconds. The morphing EnKF panel (d) is the result with the image registration determined from the time from ignition and the morphing applied to all of the model variables. The ensembles have 25 members each, and are visualized as the superposition of transparent images of heat fluxes of their members. The registration is done on the atmospheric grid with the fire heat flux as the observation function, but the atmospheric model is not used. The standard EnKF ensemble diverges from the data, while the morphing EnKF ensemble remains closer to the data.
Figure 6.7: The morphing EnKF applied to the fireline propagation model coupled with WRF. The false color of the contour on the horizontal plane is the fire output heat flux. The superposition of transparent ensemble members is shown. The volume shading is the vorticity of the atmosphere in the ensemble mean, where red and blue shades represent positive and negative vorticity, respectively. The reference solution (a) is the simulated data. The initial ensemble is created by a random perturbation of the comparison solution (b), where the fire is ignited at an intentionally incorrect location. The standard ENKF (c) and the morphing EnKF (d) are applied after 15 minutes. The ensembles have 25 members each. The registration is done on the atmospheric grid with the fire heat flux as the observation function. The standard EnKF ensemble diverges from the data, while the morphing EnKF ensemble remains closer to the data.
the analysis ensemble members for the EnKF do not resemble true model states and
the registration most likely will be meaningless. In addition, characterizing error
statistics this way would be unfairly advantageous to the morphing EnKF, which uses
this representation inside the filter.

To get some idea of the true error statistics for this problem, we propose to gen-
erate a forecast ensemble which is made up of the heat flux from a single solution of
the level set fire model translated in space by a Gaussian random variable, with some
given variance. The data will be the same model state shifted by a known quantity.
In this case, we can characterize the spatial data assimilation problem in terms of a
2-D random variable, the spatial shift from the original model state. The problem
lies in how to determine this shift from the analysis ensembles generated by the filter
methods on the model states. We will solve this problem by approximating the center
of the burning region. As we know, the burning region, \( \Omega \) is defined as the subset
of the domain where the level function is non-positive. We will define center of this
region, \((x_0, y_0)\), by the location of its centroid:

\[
x_0 = \frac{\iint_{\Omega} x\, dx\, dy}{\iint_{\Omega} dx\, dy}, \quad y_0 = \frac{\iint_{\Omega} y\, dx\, dy}{\iint_{\Omega} dx\, dy}.
\]

With this definition, we can calculate the true analysis distribution exactly and com-
pare it to the averaged statistics of the centroids to see how well the filters reproduce
the expected results. This method doesn’t inherently favor the morphing EnKF be-
cause the registration cannot be done perfectly because it does not allow for transla-
tions. It can also be defined even when the model state is non-physical (assuming \( \Omega \)
is non-empty).

The process we will employ is as follows:
• Generate an ensemble by translating a model solution in space by a Gaussian random variable with mean \((0, 0)\) and covariance \(\sigma_f^2 I\).

• Pick a translation \((x_d, y_d)\) to use for the data and create a model state from it. Also pick a data variance \(\sigma_d^2\) for the spatial shift and \(\sigma_r^2\) for the model state.

• Create the morphing transformation of the ensemble and the data solution.

• Run the EnKF from both the original model state and the transformed model state using the data state, with variance \(\sigma_r^2\), for the EnKF and the transformed data state, with variance \(\sigma_d^2\), for the morphing EnKF.

• Perform the inverse morphing transformation for the morphing EnKF analysis ensemble.

• Determine the center of the burning region for each ensemble member for both analysis ensembles, and calculate the ensemble mean and covariance of these quantities.

We will repeat this process 100 times for each chosen value of \(\sigma_f^2, \sigma_d^2, \sigma_r^2, \) and \((x_d, y_d)\).

The ensemble mean from all repetitions will then be averaged and compared to the true analysis mean of the location of the center to detect any bias in the results. In addition, the degeneracy of the analysis ensembles will be estimated by calculating the relative analysis standard deviation defined by

\[
\text{Rel. Analysis } \sigma = \sqrt{\frac{|C(\{x_N^a, y_N^a\})|}{|\text{Cov}(\{x^a, y^a\})|^{1/2}}},
\]

where \(C(\{x_N^a, y_N^a\})\) is the sample covariance of the centers of the analysis ensemble and \(\text{Cov}(\{x^a, y^a\})\) is the exact covariance of the analysis distribution. Because ensemble data assimilation techniques tend to deflate the covariance artificially (filter
degeneracy), we expect this number to be less than one. In the absence of any filter
degeneracy, the relative analysis standard deviation will be one.

The results of these experiments are given in Table 6.1. The EnKF essentially
does not respond to the data, the center of the burning region remains close to the forecast mean and the analysis standard deviation is deflated by more than three orders of magnitude. While the morphing EnKF does not approximate the exact analysis distribution well, it does tend to move the center of the fire much closer to the expected location while exhibiting much less degeneracy than the EnKF alone. These results indicate that some modification, such as a larger ensemble or covariance localization, is necessary to improve the accuracy.
Table 6.1: Analysis ensemble statistics for the EnKF with and without morphing. An ensemble of size 25 is generated of a 2-D fire shifted in space with mean 0 and standard deviation, $\sigma$. Four data points are tested with standard deviation 75 m. The exact analysis mean is given next to the sample mean resulting from the EnKF and the morphing EnKF, as well as the relative analysis standard deviation (6.4). The ensemble statistics were averaged over 100 simulations. While both methods exhibit large errors, the results indicate that the EnKF is unable to move the location of the fire, while the morphing EnKF can. Also, the EnKF alone produces far more degeneracy in the analysis ensemble compared to the morphing EnKF.

<table>
<thead>
<tr>
<th>Fore. $\sigma$</th>
<th>Data Mean</th>
<th>Analysis Mean</th>
<th>Rel. Analysis $\sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Exact</td>
<td>EnKF</td>
</tr>
<tr>
<td>1</td>
<td>(0.5,1.1)$\cdot 10^1$</td>
<td>(0.6,1.1)$\cdot 10^{-3}$</td>
<td>(0.5,1.0)$\cdot 10^1$</td>
</tr>
<tr>
<td>10</td>
<td>(0.5,1.1)$\cdot 10^1$</td>
<td>(0.5,1.1)$\cdot 10^{-1}$</td>
<td>(8.3,0.0)$\cdot 10^6$</td>
</tr>
<tr>
<td>100</td>
<td>(0.5,1.1)$\cdot 10^1$</td>
<td>(2.7,5.4)$\cdot 10^9$</td>
<td>(5.6,1.6)$\cdot 10^9$</td>
</tr>
<tr>
<td>1</td>
<td>(2.1,4.3)$\cdot 10^1$</td>
<td>(2.2,4.4)$\cdot 10^{-3}$</td>
<td>(0.3,1.5)$\cdot 10^1$</td>
</tr>
<tr>
<td>10</td>
<td>(2.1,4.3)$\cdot 10^1$</td>
<td>(2.2,4.4)$\cdot 10^{-1}$</td>
<td>(0.1,1.0)$\cdot 10^1$</td>
</tr>
<tr>
<td>100</td>
<td>(2.1,4.3)$\cdot 10^1$</td>
<td>(1.1,2.2)$\cdot 10^1$</td>
<td>(1.3,4.8)$\cdot 10^1$</td>
</tr>
<tr>
<td>1</td>
<td>(0.9,1.7)$\cdot 10^2$</td>
<td>(0.9,1.8)$\cdot 10^{-2}$</td>
<td>(0.3,1.7)$\cdot 10^1$</td>
</tr>
<tr>
<td>10</td>
<td>(0.9,1.7)$\cdot 10^2$</td>
<td>(0.9,1.7)$\cdot 10^0$</td>
<td>(0.2,2.0)$\cdot 10^1$</td>
</tr>
<tr>
<td>100</td>
<td>(0.9,1.7)$\cdot 10^2$</td>
<td>(4.3,8.7)$\cdot 10^1$</td>
<td>(1.2,2.3)$\cdot 10^1$</td>
</tr>
<tr>
<td>1</td>
<td>(3.4,6.8)$\cdot 10^2$</td>
<td>(3.5,7.0)$\cdot 10^{-2}$</td>
<td>(0.3,1.7)$\cdot 10^1$</td>
</tr>
<tr>
<td>10</td>
<td>(3.4,6.8)$\cdot 10^2$</td>
<td>(3.5,7.0)$\cdot 10^0$</td>
<td>(0.2,2.0)$\cdot 10^1$</td>
</tr>
<tr>
<td>100</td>
<td>(3.4,6.8)$\cdot 10^2$</td>
<td>(1.7,3.5)$\cdot 10^2$</td>
<td>(1.3,2.0)$\cdot 10^1$</td>
</tr>
</tbody>
</table>
6.3 Conclusion

The numerical results show that the morphing EnKF is useful for a highly nonlinear problem (a model problem for wildfire simulation) with a coherent spatial feature of the solution (propagating fireline). In previous work [Mandel et al., 2005], we have used penalization of nonphysical solutions, but the location of the fire in the data could not be too far from the location in the ensemble, artificial perturbation had to be added to retain the spread of the ensemble, and the penalty constant, the amount of additional spread, and the data variance had to be finely tuned for acceptable results. This new method does not appear to have the same limitations. The registration works automatically from gridded data and no objects need to be specified. The difference between the feature location in the data and in the ensemble can be large and the data variance can be as small as necessary, without causing filter divergence. One essential limitation is that the registered images need to be sufficiently similar, and the registration mapping should be sufficiently similar to the initial guess. This will eventually impose a limitation on how long can the ensemble go without an analysis step. However, compared to previous results for the same problem [Johns and Mandel, 2008, Mandel et al., 2005], the convergence of the present filter is much better.

Several enhancements to the morphing EnKF are in development. These additional features are designed to extend the usability of the technique to more general models and data types. They include the following. Support for registration of an image consisting of only a subset of the model domain. This is important because a large wildfire may stretch across many kilometers, but an aerial image may only cap-
ture a small fragment of the fireline. Preliminary support for this feature is included in the current implementation by the addition of a masking array. The image is projected onto the model grid, and pixels which do not contain valid data are recorded in the masking array. The registration procedure continues as normal except in the evaluation of the residual norm in the objective function (5.13); residual pixels located where the masking array indicates that no data is present are set to zero. This method allows for the subimage to be registered while maintaining a smooth morphing function over the whole domain; however, it requires computations over the whole domain even when the image is much smaller. Further research is required to determine whether this method is feasible in general.

We have assumed in the registration that the data is a direct observation of one of the state variables. Even if subimages are allowed, we still cannot operate on data with more complicated observation functions that frequently occur in real applications. For image data, where each pixel is only locally dependent on the model variables, it may be possible to substitute $u$ and $v$ in the residual component of the objective function (5.13) with morphed synthetic data, such as $u \circ (I + T) = \tilde{h}(R, T)$. This approach necessarily requires that the observation function is not computationally expensive because it will need to be performed with each evaluation of the objective function.

There is also a need for similar filtering methods for point data, which might be obtained from weather stations. Applications such as these are more difficult because they don’t directly provide information about the location of the feature of interest, and the presence of such information is integral to the concept of the morphing transformation. However, assuming that the observations are made continuously in time, it could be possible to obtain information about when (or if) the feature, such as the
fire, passed over the location of the instrument. Future research will be conducted regarding the inclusion of this information into the morphing ensemble filter.

Other simple modifications that might improve the success of the algorithm described here are possible. For instance, one could register the ensemble states more often, whether there are new data or not. This would ensure that the initial guess for the registration algorithm is always sufficiently close, thus improving the quality of the registration and the time required to produce it. Also, the common state $X_0$, which the ensemble members are compared against has been evolved from one initial condition regardless of the data. Over time, such $X_0$ could diverge significantly from the ensemble (which tracks the data), resulting in more strenuous registration. If this becomes an issue, a better $X_0$ might be constructed from the analysis directly (perhaps as the mean of registration representations of the analysis ensemble members) and then evolved in time until the next analysis step.

While the numerical results indicate that the morphing EnKF provides a vast improvement when using synthetic data generated by the numerical model, no experiments have been attempted using real data. In addition to the complications presented by more realistic observation functions, real data will appear more noisy than synthetic data. The real data may not even resemble the synthetic data at all. Realistic fires could exhibit much more complicated behavior such as splitting into multiple separate fires. It is not clear how the registration and morphing algorithms will react to these situations. Future research will be needed to address these issues as they arise. The fact that existing methods behave so poorly, even with the simple experiments contained here, provides some optimism for the success of future incarnations of the morphing EnKF.
Bibliography


Robert Kremens, Jason Faulring, and Colin C. Hardy. Measurement of the time-temperature and emissivity history of the burn scar for remote sensing applications.


Jan Mandel, Lynn S. Bennethum, Mingshi Chen, Janice L. Coen, Craig C. Douglas, Leopoldo P. Franca, Craig J. Johns, Minjeong Kim, Andrew V. Knyazev, Robert Kremens, Vaibhav Kulkarni, Guan Qin, Anthony Vodacek, Jianjia Wu, Wei Zhao, and Adam Zornes. Towards a dynamic data driven application system for wildfire


Istvan Szunyogh, Eric J. Kostelich, Gyorgyi Gyarmati, Eugenia Kalnay, Brian R. Hunt, Edward Ott, Elizabeth Satterfield, and James A. Yorke. A local ensemble


