Data Assimilation for Dispersion Models

K. V. Umamaheswara Reddy
Dept. of Mechanical and Aerospace Engg.
State University of New York at Buffalo
Buffalo, NY, U.S.A.
venkatar@buffalo.edu

Tarunraj Singh
Dept. of Mechanical and Aerospace Engg.
State University of New York at Buffalo
Buffalo, NY, U.S.A.
tsingh@eng.buffalo.edu

Yang Cheng
Dept. of Mechanical and Aerospace Engg.
State University of New York at Buffalo
Buffalo, NY, U.S.A.
cheng3@buffalo.edu

Peter D. Scott
Dept. of Computer Science and Engg.
State University of New York at Buffalo
Buffalo, NY, U.S.A.
peter@buffalo.edu

Abstract - *The design of an effective data assimilation environment for dispersion models is studied. These models are usually described by partial differential equations which lead to large scale state space models. The linear Kalman filter theory fails to meet the requirements of this application due to high dimensionality, strong non-linearities, non-Gaussian driving disturbances and model parameter uncertainties. Application of Kalman filter to these large scale models is computationally expensive and real time estimation is not possible with the present resources. Various Monte Carlo filtering techniques are studied for implementation in the case of dispersion models, with a particular focus on Ensemble filtering approaches and particle filtering approaches. The filters are compared with the full Kalman filter estimates on a one dimensional spherical diffusion model.*

Keywords: Chem-Bio Dispersion, Data Assimilation, Ensemble Kalman filter, Ensemble Square Root filter, Particle Filter.

1 Introduction

Chem-Bio dispersion is a complex nonlinear physical (and chemical) process with numerous uncertainties in model order, model parameters, meteorological inputs, and initial and boundary conditions. The application of various empirical or mathematical models is limited by the lack of complete knowledge about the physical process and the various uncertainties. Even the best models are of limited forecast capabilities without measurements to correct the forecast errors as they accumulate over time. Simple interpolation of measurements also fails to describe the process to the satisfactory extent, because the measurements are often asynchronous, incomplete, imperfect, unevenly distributed, and spatially and temporally sparse. Dynamic data assimilation improves the knowledge of the process by combining the measurements with the model.

When a stochastic description of the process is available, the measurements may be incorporated into the model in an optimal way, using the sequential Bayesian data assimilation methods. The ultimate objective of Bayesian estimation is to reconstruct the posterior distribution of the states. For linear stochastic models whose uncertainty is modeled as Gaussian noise, the posterior distribution is also Gaussian, which is fully parameterized by its mean and covariance (the first two moments), and the optimal Bayesian estimator for this case is the well-known Kalman filter [1]. The Kalman filter may also be derived as an optimal linear least-squares estimator, without introducing probability distributions. Most data assimilation problems involve nonlinear models, however. In general, exact solution for the state posterior distribution of a nonlinear model is intractable, even if the nonlinear model is small and simple. Various approximate nonlinear filtering approaches are resorted in practice, many of which aim to recursively estimate the mean and covariance of the states instead of the much more complex posterior distribution.

The extended Kalman filter[1] is one of the simplest nonlinear filters. Of the infinite moments that describe the posterior distribution only the mean and the error covariance are computed. The model is linearized around the most recent estimate; the first derivatives associated with the linear error model have to be computed. The extended Kalman filter is sufficient for many applications. However, when the state estimates are poorly initialized, the time between measurements increases, the noise is large, and/or the model is highly nonlinear, the method may fail. In these cases it may be worthwhile to use filters with higher order truncations, which require second or higher order derivatives[1]. The Unscented Filter[2] avoids computation of any derivatives while still attaining second order accuracy in mean and covariance estimates, but is also computationally more expensive than the extended Kalman filter. The particle filters[3] approximate the posterior distribution of the states with weighted particles, being able to provide higher moment estimates and not requiring the posterior distribution to be Gaussian. The particle filters are most

suitable for highly nonlinear non-Gaussian models, but are among the computationally most expensive nonlinear filters. The computational complexity of a particle filter is closely related to the importance function employed. Daum recently showed that a carefully designed particle filter should mitigate the curse of dimensionality for certain filtering problems, but the particle filter does not avoid the curse of dimensionality in general[4].

A common characteristic of numerical atmospheric dispersion models is high dimensionality and high complexity, directly resulting from discretization of the governing partial differential equations. For grid-based models, the size n of the state vector of an operational model can be of the order of millions or even larger. In the extended Kalman filter, the computational complexity of the update of the error covariance alone is of the order of $\mathcal{O}(n^3)$[5]. Given the excessive or prohibitively large computational requirements, the extended Kalman filter and the more advanced methods cannot be applied directly[5].

Reduced-rank filters, which employ reduced-rank approximation to the full-rank covariance matrix, are becoming popular for large-scale data assimilation in atmospheric dispersion, weather forecast, and so on[5]. The ensemble Kalman filters[6, 7] are a typical example, with the error covariance matrix approximated by the ensemble covariance around the ensemble mean. The main operations of the ensemble Kalman filters are the dynamical propagation and transformation of the ensemble members. The transformation at measurement times may be preformed stochastically by treating observations as random variables, or deterministically by requiring that the updated analysis perturbations satisfy the Kalman filter analysis error covariance equation[8]. Compared with the extended Kalman filter, these methods are easy to implement and computationally inexpensive. Besides, the ensemble Kalman filters need no linearization or computation of the Jacobian matrices. Since the ensemble members are propagated with the fully nonlinear forecast model, the ensemble Kalman filter may do better in forecast than the extended Kalman filter[9].

It is not unusual to find in the data assimilation literature that 100 ensemble members are sufficient for data assimilation on systems of thousands of or more state variables[7], which is in sharp contrast with the typical nonlinear particle filtering applications in which hundreds of or thousands of particles are used with as low as one- or three-dimensional nonlinear systems. However, that should not be interpreted as that the ensemble Kalman filters are able to beat the curse of dimensionality. The justification for using such a limited number of ensemble members lies not in the magic of the ensemble Kalman filters, which are inferior to the full extended Kalman filter in estimation performance in many cases, but in the fact that the degrees of freedom of the sophisticated, elaborate dynamical models for atmospheric dispersion are much smaller than the dimensionality of the system. As a result, working in a subspace of much lower dimension does not cause severe performance degradation.

In this paper, the efficacy of the ensemble filtering and particle filtering schemes is studied using the example of a one-dimensional diffusion model. Both filtering methods are based on Monte Carlo simulations of an ensemble. For the moment, we are focused on the dimensionality issue. The paper is organized as follows: In section 2, the diffusion model is introduced as a partial differential equation which is solved using the finite-difference approach. In section 3, the various filtering schemes including the full Kalman filter are described in detail. These methods are implemented on the diffusion model and the results are compared in section 4. The conclusions and further research are discussed in section 5.

2 Diffusion Model

The model used is a one dimensional spherical diffusion model to simulate the diffusion of a species expelled at a radial velocity u_R from a spherical source of radius R.

The equation for conservation of a diffusing species for any control volume can be written as

$$\begin{aligned}\text{mass accumulation rate} &= (\text{mass in} - \text{mass out})_{advection} \\ &\quad +(\text{mass in} - \text{mass out})_{diffusion} \\ &\quad +\text{generation rate} \end{aligned} \tag{1}$$

Consider a thin spherical shell of thickness dr as the control volume. Assuming the generation rate in this volume to be zero, equation 1 reduces to the following partial differential equation,

$$\frac{\partial}{\partial t}(\rho f) + \frac{1}{r^2}\frac{\partial}{\partial r}(r^2 \rho u_r f) = \frac{1}{r^2}\frac{\partial}{\partial r}(r^2 D_m \frac{\partial(\rho f)}{\partial r}) \tag{2}$$

where f represents the mass fraction of the species and D_m is Diffusivity (from Fick's law). If ρ is constant, then it can easily be seen that $\rho r^2 u_r$ is a constant and is given by $\frac{\dot{m}}{4\pi}$. The equation for conservation of species is now given by,

$$\rho\frac{\partial f}{\partial t} + \frac{\dot{m}}{4\pi r^2}\frac{\partial f}{\partial r} = \frac{1}{r^2}\frac{\partial}{\partial r}(\rho r^2 D_m \frac{\partial f}{\partial r}) \tag{3}$$

Initially, $f = 0$ everywhere. The boundary conditions are stated below. At $r = R$ (surface of emission), we have

$$\frac{\dot{m}}{4\pi R^2} = \frac{\dot{m}}{4\pi R^2} f_s - \rho D_m \frac{\partial f}{\partial r}_{r=R} \tag{4}$$

where, f_s is the value of f at the surface $r = R$. This equation can be rewritten as follows:

$$\frac{\dot{m}}{4\pi R^2 \rho D_m} = \frac{1}{f_s - 1}\frac{\partial f}{\partial r}_{r=R} \tag{5}$$

Also, as $r \to \infty$, $f \to 0$.

Note that equation 2 is singular as $r \to 0$. This causes singularity problems when solving numerically.

To avoid this, the following variable transformations and standard notation are introduced.

$$\eta = \left(\frac{r}{R}\right)^3 \quad \text{and} \quad t^* = t/\left(\frac{R}{u_R}\right) \tag{6}$$

$$Sc = \frac{\nu}{D_m} \quad \text{and} \quad Re = \frac{u_R R}{\nu} \tag{7}$$

This simplifies the PDE 2 to the following:

$$\frac{\partial f}{\partial t} + 3\frac{\partial f}{\partial \eta} = \frac{9}{Sc \cdot Re}\frac{\partial}{\partial \eta}(\eta^{4/3}\frac{\partial f}{\partial \eta}) \tag{8}$$

which can be further rewritten as follows, which is useful for developing numerical solutions.

$$\frac{\partial f}{\partial t} + 3\frac{\partial f}{\partial \eta}\left[1 - \frac{4}{Sc \cdot Re}\eta^{1/3}\right] = \frac{9}{Sc \cdot Re}\eta^{4/3}\frac{\partial^2 f}{\partial \eta^2} \tag{9}$$

The boundary conditions in terms of the new variables are as follows.

$$\text{As} \quad \eta \to \infty, \qquad f \to 0 \tag{10}$$

$$\text{At} \quad \eta = 1, \qquad \frac{\partial f}{\partial \eta}_{\eta=1} = \frac{Sc \cdot Re}{3}(f_s - 1) \tag{11}$$

2.1 Numerical Solution

The Crank-Nicholson method is used for developing the numerical solution. Introducing the notation $\beta = 3\left[1 - \frac{4}{Sc \cdot Re}\eta^{1/3}\right]$ and $\alpha = \frac{9}{Sc \cdot Re}\eta^{4/3}$, the finite difference approximation of the model is given below:

$$\frac{f_m^{k+1} - f_m^k}{\Delta t} = \frac{1}{2}\left(-\beta\frac{f_{m+1}^k - f_{m-1}^k}{2\Delta x} - \beta\frac{f_{m+1}^{k+1} - f_{m-1}^{k+1}}{2\Delta x}\right)$$
$$+\frac{1}{2}\left(\alpha\frac{f_{m+1}^k - 2f_m^k + f_{m-1}^k}{\Delta x^2} + \alpha\frac{f_{m+1}^{k+1} - 2f_m^{k+1} + f_{m-1}^{k+1}}{\Delta x^2}\right) \tag{12}$$

The above system of equations can be solved by LU decomposition, i.e. by forming a tridiagonal system of equations which is solved using the Thomas algorithm.

3 Filtering Algorithms

A general nonlinear stochastic description of the discrete time model and the measurements can be shown as below:

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{w}_k) \tag{13}$$

$$\mathbf{y}_k = \mathbf{g}(\mathbf{x}_k, \mathbf{v}_k) \tag{14}$$

When a stochastic description of model and measurements of the form

$$\mathbf{x}_{k+1} = \mathbf{A}_k\mathbf{x}_k + \mathbf{B}_k\mathbf{u}_k + \mathbf{F}_k\mathbf{w}_k \tag{15}$$

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x}_k + \mathbf{v}_k \tag{16}$$

is available, it is possible to incorporate the measurements into the model to obtain an optimal estimate of the system. For this linear model with the assumption of Gaussian noise, the Kalman filter provides the optimal estimate. The main advantage of using a linear model driven by Gaussian noise is that the state and output will also be Gaussian. Furthermore, a Gaussian distribution is characterized by its mean and covariance only. The Kalman filter therefore has only first moment (mean) and second moment (error covariance) equations. The Kalman filter equations are given by:

$$\hat{\mathbf{x}}_{k+1|k} = \mathbf{A}_k\hat{\mathbf{x}}_{k|k} + \mathbf{B}_k\mathbf{u}_k \tag{17}$$

$$\mathbf{P}_{k+1|k} = \mathbf{A}_k\mathbf{P}_{k|k}\mathbf{A}_k^T + \mathbf{F}_k\mathbf{Q}_k\mathbf{F}_k^T \tag{18}$$

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}(\mathbf{y}_{k+1} - \mathbf{C}_{k+1}\hat{\mathbf{x}}_{k+1|k}) \tag{19}$$

$$\mathbf{K}_{k+1} = \mathbf{P}_{k+1|k}\mathbf{C}_{k+1}^T(\mathbf{C}_{k+1}\mathbf{P}_{k+1|k}\mathbf{C}_{k+1}^T + R_{k+1})^{-1} \tag{20}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}\mathbf{P}_{k+1|k} \tag{21}$$

$$\hat{\mathbf{x}}_{0|0} = \mathbf{x}_0 \tag{22}$$

$$\mathbf{P}_{0|0} = \mathbf{P}_0 \tag{23}$$

An equivalent but more numerically stable substitute for covariance update is as follows:

$$\mathbf{P}_{k+1|k+1} = [\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}]\,\mathbf{P}_{k+1|k}\,[\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}_{k+1}]^T + \mathbf{K}_{k+1}R_{k+1}\mathbf{K}_{k+1}^T \tag{24}$$

Although these Kalman filter equations can in principle be used to solve many data assimilation problems the actual implementation for real life problems is far from easy.

3.1 Ensemble Kalman Filter

The ensemble Kalman filter [6] is a data assimilation method that approximates the conditional density with a Monte Carlo method. The data is assimilated using the analysis step of the Kalman filter, whereas the error covariance matrix is replaced by the sample error covariance.

In this filter, the forecast error covariance is computed by integrating an ensemble of randomly perturbed initial analysis states in time with random perturbations added to the forcing. This Monte Carlo type approach based on the full nonlinear model allows for consistent statistics in the case of nonlinear dynamics. The analysis of the perturbed states, known as ensemble members, is carried out with perturbed observations.

The propagation equations are given by:

$$\xi_{ik+1|k} = \mathbf{f}\left(\xi_{ik|k}, \mathbf{u}_k, \mathbf{w}_{ik}\right) \tag{25}$$

$$\hat{\mathbf{x}}_{k+1|k} = \frac{1}{N}\sum_{i=1}^{N}\xi_{ik+1|k} \tag{26}$$

Let $\mathbf{X}$ be the matrix holding the ensemble members $\xi_{ik+1|k} \in \Re^{n \times N}$,

$$\mathbf{X}_{k+1|k} = (\xi_{1k+1|k}, \xi_{2k+1|k}, \ldots, \xi_{Nk+1|k}) \tag{27}$$

where N is the number of ensemble members and n is the size of the model state vector. The ensemble mean $\hat{\mathbf{x}}_{k+1|k}$ is stored in each column of $\overline{\mathbf{X}}_{k+1|k}$, which is

of size $n \times N$. The ensemble covariance matrix $P_e \in \Re^{n \times n}$ is defined as

$$\mathbf{P}_e = \frac{\mathbf{X}'\mathbf{X}'^T}{N-1} \quad (28)$$

$$\text{where,} \quad \mathbf{X}' = \mathbf{X}_{k+1|k} - \overline{\mathbf{X}}_{k+1|k} \quad (29)$$

Given a vector of measurements $\boldsymbol{d} \in \Re^m$ at the $(k+1)^{th}$ time step, with m being the number of measurements, we can define the N vectors of perturbed observations as

$$\boldsymbol{d}_j = \boldsymbol{d} + \epsilon_j, \qquad j = 1, \ldots, N, \quad (30)$$

where $\boldsymbol{d}$ is the actual measurement vector and ϵ_j is the measurement error vector that is randomly generated from a predefined distribution with zero mean and covariance matrix $\mathbf{R}$. These perturbed observations can be stored in the columns of a matrix

$$\mathbf{D} = (\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_N) \in \Re^{m \times N}, \quad (31)$$

while the ensemble of perturbations, with ensemble mean equal to zero, can be stored in the matrix

$$\Upsilon = (\epsilon_1, \epsilon_2, \ldots, \epsilon_N) \in \Re^{m \times N}, \quad (32)$$

from which we can construct the ensemble representation of the measurement error covariance matrix

$$\mathbf{R}_e = \frac{\Upsilon\Upsilon^T}{N-1} \quad (33)$$

The ensemble of innovations vectors defined as:

$$\mathbf{D}' = \mathbf{D} - \mathbf{Y}_{k+1} \quad (34)$$

$$\text{where,} \quad \mathbf{Y}_{k+1} = \mathbf{g}\left(\mathbf{X}_{k+1|k}, \mathbf{v}_k\right) \quad (35)$$

and the mean $\overline{\mathbf{y}}_{k+1}$ is stored in each column of $\overline{\mathbf{Y}}_{k+1}$, which is of size $m \times N$.

$$\mathbf{Y}' = \mathbf{Y}_{k+1} - \overline{\mathbf{Y}}_{k+1} \quad (36)$$

Using the above notations, the standard analysis equation can be expressed as:

$$\mathbf{X}_{k+1|k+1} = \mathbf{X}_{k+1|k} + \mathbf{X}'\mathbf{Y}'^T(\mathbf{Y}'\mathbf{Y}'^T + \Upsilon\Upsilon^T)^{-1}\mathbf{D}' \quad (37)$$

The potential singularity of the inverse computation requires the use of a pseudoinverse.

3.2 Ensemble Square Root Kalman Filter

The Ensemble Kalman Filter (EnKF) as described above uses pure Monte Carlo sampling when generating the initial ensemble, the model noise and the measurement perturbations. By selecting the initial ensemble, the model noise and the measurement perturbations wisely, it is possible to achieve a significant improvement in the EnKF results, using the same number of ensemble members. The measurement perturbations introduce sampling errors which can be fully eliminated when the square root analysis algorithm is used. Potential loss of rank may occur in the case when random measurement perturbations are used to represent the measurement error covariance matrix. This can be avoided by a proper sampling of measurement perturbations or avoiding the perturbations as such. The algorithm described here solves for the analysis avoiding the perturbation of measurements, and without imposing any additional approximations, such as the assumption of uncorrelated measurement errors. This is discussed in detail in Evensen[7].

The algorithm is used to update the ensemble perturbations and is derived starting from the traditional analysis equation for the covariance update 21 in the Kalman Filter. When using the ensemble representation 28 this can be rewritten as

$$\mathbf{X}^{a'}\mathbf{X}^{a'T} = \mathbf{X}'(\mathbf{I} - \mathbf{Y}'^T\mathbf{G}^{-1}\mathbf{Y}')\mathbf{X}'^T \quad (38)$$

$$\text{where,} \quad \mathbf{X}^{a'} = \mathbf{X}_{k+1|k+1} - \overline{\mathbf{X}}_{k+1|k+1} \quad (39)$$

$$\mathbf{G} = \mathbf{Y}'\mathbf{Y}'^T + \Upsilon\Upsilon^T \quad (40)$$

The analyzed ensemble mean is computed from the standard analysis 37, as follows:

$$\hat{\mathbf{x}}_{k+1|k+1} = \hat{\mathbf{x}}_{k+1|k} + \mathbf{X}'\mathbf{Y}'^T\mathbf{G}^{-1}(\boldsymbol{d} - \bar{\mathbf{y}}_{k+1}) \quad (41)$$

The equation for the ensemble analysis is derived by defining a factorization of the covariance 38, where $\mathbf{G}$ is rewritten as

$$\mathbf{G} = \mathbf{Y}'\mathbf{Y}'^T + (N-1)\mathbf{R} \quad (42)$$

so that no reference is made to the measurements or measurement perturbations.

Assume that $\mathbf{G}$ is of full rank such that $\mathbf{G}^{-1}$ exists. Compute the eigenvalue decomposition and obtain

$$\mathbf{G}^{-1} = \mathbf{Z}\Lambda^{-1}\mathbf{Z}^T \quad (43)$$

Substituting this in 38, we obtain

$$\begin{aligned} \mathbf{X}^{a'}\mathbf{X}^{a'T} &= \mathbf{X}'(\mathbf{I} - \mathbf{Y}'^T\mathbf{Z}\Lambda^{-1}\mathbf{Z}^T\mathbf{Y}')\mathbf{X}'^T \\ &= \mathbf{X}'\left[\mathbf{I} - (\Lambda^{-\frac{1}{2}}\mathbf{Z}^T\mathbf{Y}')^T(\Lambda^{-\frac{1}{2}}\mathbf{Z}^T\mathbf{Y}')\right]\mathbf{X}'^T \\ &= \mathbf{X}'\left(\mathbf{I} - \mathbf{X}_1^T\mathbf{X}_1\right)\mathbf{X}'^T \quad (44) \end{aligned}$$

$$\text{where,} \quad \mathbf{X}_1 = \Lambda^{-\frac{1}{2}}\mathbf{Z}^T\mathbf{Y}' \quad (45)$$

Compute the singular value decomposition (SVD) of $\mathbf{X}_1$ and obtain

$$\mathbf{X}_1 = \mathbf{U}_1\Sigma_1\mathbf{V}_1^T \quad (46)$$

Substituting back, we get

$$\begin{aligned} \mathbf{X}^{a'}\mathbf{X}^{a'T} &= \mathbf{X}'\left(\mathbf{I} - \left[\mathbf{U}_1\Sigma_1\mathbf{V}_1^T\right]^T\left[\mathbf{U}_1\Sigma_1\mathbf{V}_1^T\right]\right)\mathbf{X}'^T \\ &= \mathbf{X}'\left(\mathbf{I} - \mathbf{V}_1\Sigma_1^T\Sigma_1\mathbf{V}_1^T\right)\mathbf{X}'^T \\ &= \mathbf{X}'\mathbf{V}_1\left(\mathbf{I} - \Sigma_1^T\Sigma_1\right)\mathbf{V}_1^T\mathbf{X}'^T \\ &= \left(\mathbf{X}'\mathbf{V}_1\sqrt{\mathbf{I} - \Sigma_1^T\Sigma_1}\right)\left(\mathbf{X}'\mathbf{V}_1\sqrt{\mathbf{I} - \Sigma_1^T\Sigma_1}\right)^T \quad (47) \end{aligned}$$

Thus, a solution for the analysis ensemble perturbations is given by,

$$\mathbf{X}^{a'} = \mathbf{X}'\mathbf{V}_1\sqrt{\mathbf{I} - \Sigma_1^T\Sigma_1}\Theta^T \quad (48)$$

This is added to the updated ensemble mean to get the ensemble update $\mathbf{X}(k+1|k+1)$ (39).

The additional multiplication with a random orthogonal matrix Θ^T also results in a valid solution. Such a random redistribution of the variance reduction among the ensemble members is in some cases necessary and is used by default. The matrix Θ^T is easily constructed, e.g., by using the right singular vectors from an SVD of a random $N \times N$ matrix. However, note that by definition $\mathbf{X}' \cdot \mathbf{1} = \mathbf{X}^{a'} \cdot \mathbf{1} = \mathbf{0}$ with $\mathbf{1}$ a column vector of 1's. In this sense, the choice of Θ^T should not be arbitrary.

3.3 Particle Filters

In particle filters, the posterior distribution $p(\mathbf{x}_k|\mathbf{Y}_k)$ is approximated with N weighted particles $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$, given by

$$P_N(d\mathbf{x}_k|\mathbf{Y}_k) \approx \sum_{i=1}^{N} w_k^{(i)} \delta_{\mathbf{x}_k^{(i)}}(d\mathbf{x}_k) \tag{49}$$

where $\mathbf{x}_k^{(i)}$ are the particles drawn from the importance function or proposal distribution, $w_k^{(i)}$ are the normalized importance weights, satisfying $\sum_{i=1}^N w_k^{(i)} = 1$, and $\delta_{\mathbf{x}_k^{(i)}}(d\mathbf{x}_k)$ denotes the Dirac-delta mass located in $\mathbf{x}_k^{(i)}$. We use $\mathbf{X}_k$ and $\mathbf{Y}_k$ to denote the state trajectory $\{\mathbf{x}_j\}_{j=0}^k$ and measurement history $\{\mathbf{y}_j\}_{j=1}^k$, respectively. The expectation of a known function $\mathbf{f}(\mathbf{x}_k)$ with respect to $p(\mathbf{x}_k|\mathbf{Y}_k)$ is then approximated by

$$\int \mathbf{f}(\mathbf{x}_k) p(\mathbf{x}_k) d\mathbf{x}_k \approx \sum_{i=1}^{N} w_k^{(i)} \mathbf{f}(\mathbf{x}_k^{(i)}) \tag{50}$$

For example, the approximation to the arithmetic mean of $\mathbf{x}_k$ is $\sum_{i=1}^N w^{(i)} \mathbf{x}^{(i)}$.

A particle filter updates the particle representation $\{\mathbf{x}_k^{(i)}, w_k^{(i)}\}_{i=1}^N$ in a recursive manner. A cycle of a generic particle filter includes[3]

- Sequential Importance Sampling
  - For $i = 1, \ldots N$, sample $\mathbf{x}_{k+1}^{(i)}$ from the importance function $q(\mathbf{x}_{k+1}|\mathbf{X}_k^{(i)}, \mathbf{Y}_{k+1})$
  - For $i = 1, \ldots N$, evaluate and normalize the importance weights

$$w_{k+1}^{(i)} \propto w_k^{(i)} \frac{p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}^{(i)}) p(\mathbf{x}_{k+1}^{(i)}|\mathbf{x}_k^{(i)})}{q(\mathbf{x}_{k+1}^{(i)}|\mathbf{X}_k^{(i)}, \mathbf{Y}_{k+1})} \tag{51}$$

- Reampling: Multiple/Discard particles $\{\mathbf{x}_{k+1}^{(i)}\}_{i=1}^N$ with respect to high/low importance weights $w_{k+1}^{(i)}$ to obtain N new particles $\{\mathbf{x}_{k+1}^{(i)}\}_{i=1}^N$ with equal weights.

It should be noted that the computation of the mean and covariance is not required for the process of the particle filter.

The importance function plays a significant role in the particle filter. One of the simplest importance function is given by $q(\mathbf{x}_{k+1}|\mathbf{X}_k^{(i)}, \mathbf{Y}_{k+1}) = p(\mathbf{x}_{k+1}|\mathbf{x}_k^{(i)})$. The corresponding importance weights are $w_{k+1}^{(i)} \propto w_k^{(i)} p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}^{(i)})$. Sampling $\mathbf{x}_{k+1}^{(i)}$ from $p(\mathbf{x}_{k+1}|\mathbf{x}_k^{(i)})$ is equivalent to the dynamic propagation of $\mathbf{x}_k^{(i)}$ to time t_{k+1}. The optimal importance function that minimizes the variance of the importance weight $w_{k+1}^{(i)}$ conditional upon $\mathbf{x}_k^{(i)}$ and $\mathbf{y}_{k+1}$ is given by $q(\mathbf{x}_{k+1}|\mathbf{X}_k^{(i)}, \mathbf{Y}_k) = p(\mathbf{x}_{k+1}|\mathbf{x}_k^{(i)}, \mathbf{y}_{k+1})$. The corresponding importance weights are $w_{k+1}^{(i)} \propto w_k^{(i)} p(\mathbf{y}_{k+1}|\mathbf{x}_k^{(i)})$. The optimal particle filter gives the limiting performance of particle filters.

When the state model is linear Gaussian as given in the previous section, all the above-mentioned probability density functions are Gaussian. In the simple particle filter, the mean and covariance of $p(\mathbf{x}_{k+1}|\mathbf{x}_k^{(i)})$ are given by $\hat{\mathbf{x}}_{k+1|k} = \mathbf{A}_k \mathbf{x}_k^{(i)} + \mathbf{B}_k \mathbf{u}_k$ and $\mathbf{P}_{k+1|k} = \mathbf{F}_k \mathbf{Q}_k \mathbf{F}_k^T$ respectively; the mean and covariance of $p(\mathbf{y}_{k+1}|\mathbf{x}_{k+1}^{(i)})$ are given by $\mathbf{C}_{k+1}\mathbf{x}_{k+1}^{(i)}$ and $\mathbf{R}_{k+1}$ respectively. In the optimal particle filter, the mean and covariance of $p(\mathbf{x}_{k+1}|\mathbf{x}_k^{(i)}, \mathbf{y}_{k+1})$ are given by $\hat{\mathbf{x}}_{k+1|k} + \mathbf{K}_{k+1}(\mathbf{y}_{k+1} - \mathbf{C}_{k+1}\hat{\mathbf{x}}_{k+1|k})$ and $(\mathbf{I} - \mathbf{K}_{k+1}\mathbf{C}_{k+1})\mathbf{P}_{k+1|k}$ respectively; the mean and covariance of $p(\mathbf{y}_{k+1}|\mathbf{x}_k^{(i)})$ are given by $\mathbf{C}_{k+1}\hat{\mathbf{x}}_{k+1|k}$ and $\mathbf{C}_{k+1}\mathbf{P}_{k+1|k}\mathbf{C}_{k+1}^T + \mathbf{R}_{k+1}$ respectively.

It is well known that the simple particle filter with the prior $p(\mathbf{x}_{k+1}|\mathbf{x}_k^{(i)})$ as the importance function does not work well when the overlap between the prior and the likelihood is small, for example, when the measurement is very accurate[10]. A simple trick is applied in the simple particle filter. Basically, the measurement update of the particle weights are done twice per filter cycle. First, a large measurement noise variance $25R$ is used to calculate the Gaussian likelihood and update the weights. Then, the particles are resampled. Finally, $25/24R$ is used to do the same update again. That is based on the factorization $e^{-1/R} = e^{-1/25/R} \cdot e^{-24/25/R}$. The procedure will not change the weight associated with a particle but will have more particles be selected in the resampling steps. The more systematic, adpative treatment is known as "progressive corrections[10]."

4 Comparison of Various Filters

For the purpose of data assimilation using the various filtering schemes, the numerical model 12 is formulated as a state-space model. The state variables are the mass fractions f of the species at all grid points, the number of grid points n being the number of states. A grid with $n = 31$ is used for the model used for testing the various filtering schemes. A finer grid ($n = 301$) is used to simulate the measurements at sensor locations. The uncertainties in the process model and the measurement model are modeled as Gaussian white noise processes.

In the simulations, the distance from the source and

the simulation time are given by the non-dimensional variables η and t^* respectively, as described in equation 6. The value of η varies from 1 to 100 in these simulations, while the simulation time t^* is 20. The model is simulated for the fine grid ($n = 301$) and the values of f are taken as the truth at each time-step. The values of the measurement at the simulated sensor locations $\eta = [20\ 40\ 60]$ is obtained by linear interpolation of the values of f at each grid point of the fine grid, and adding measurement noise with a covariance R.

The coarse model ($n = 31$) is taken as an approximate model to simulate the above true model and the filtering schemes are applied to estimate the states using the measurements obtained as described before. The ensemble Kalman filter, the ensemble square root filter and the simple particle filter are applied to the spherical diffusion problem, and the plots are compared with the truth and the full Kalman estimates. Note that the number of states in this model is $n = 31$. The measurement noise standard deviation at all the three sensor locations is 0.1. The number of ensemble members used is 20 in both the ensemble filtering schemes and the number of particles used is also 20 in the particle filter. Since Monte Carlo filtering schemes are used for estimation, the results plotted are the values averaged over 50 Monte Carlo simulations.

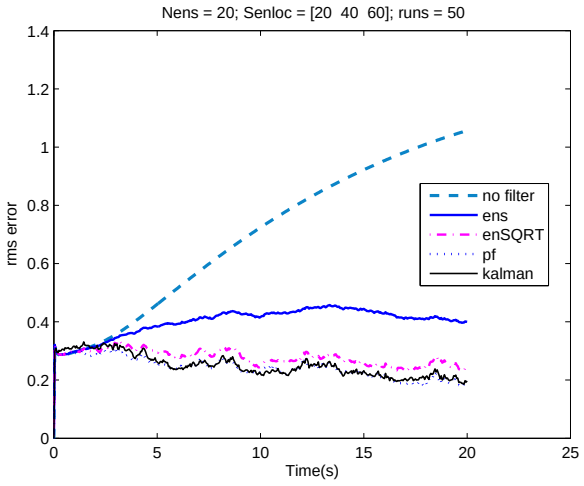

Figure 1: RMS error in all states

In figure 1, the root mean square error in all the states is calculated and is plotted vs. time, for the ensemble filters and the particle filter. These are compared against the error calculated using just the model propagation and using the full kalman filter. It can be observed that the full Kalman filter gives the best results as expected, while the ensemble square root filter has a significant advantage over the ensemble filter. The simple particle filter gives results similar to the full Kalman filter. Also the model error grows in time (to a steady state) as the error keeps propagating through the coarse model, while the estimates in all the schemes improve with time as the filters acquire more information. In figure 2, the results of the simple particle filter are compared with that of the optimal particle filter.

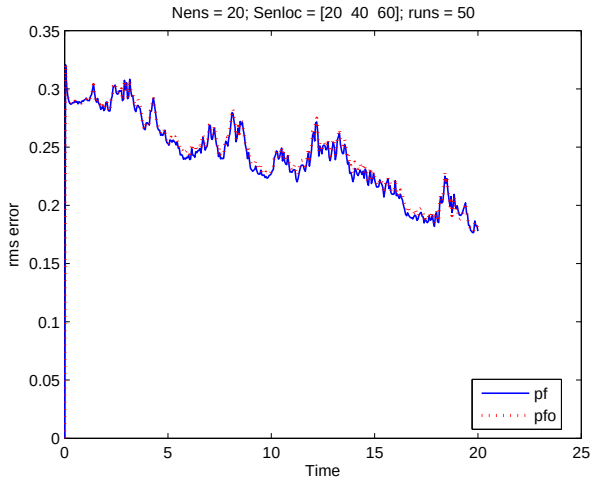

Figure 2: RMS error in all states for PF schemes

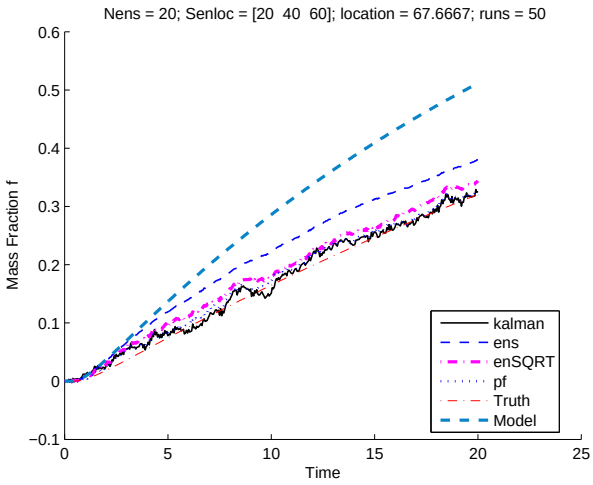

Figure 3: Estimates of the Mass fraction

It can be noticed that in our case, both the results are comparable in terms of accuracy while the optimal filter takes a very long time ($7519.6s$) compared to the simple particle filter ($445.3s$).

In figure 3, the estimates of the mass fraction of the species at two-thirds ($\eta = 66.67$) of the diffusion domain are plotted for various schemes, for the purpose of illustration. These estimates are compared with the truth as obtained from the simulation of the fine grid model, the full Kalman filter estimates and the values as predicted by coarse model. It can be observed that the model propagated state differs considerably from the truth, which is corrected using the various filtering techniques. Also, the particle filter estimate is as good as the Kalman estimate. The square root filter estimate is comparable to the Kalman estimate and shows a very good improvement over the standard ensemble filter. In figure 4 and figure 5, the one standard deviation bands are plotted for both the ensemble filtering methods. The same is plotted for the simple particle filtering scheme in figure 6.

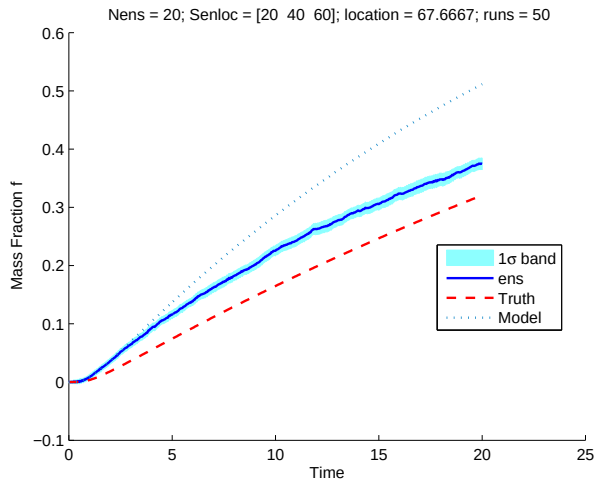

Figure 4: 1σ band of the Ensemble estimate

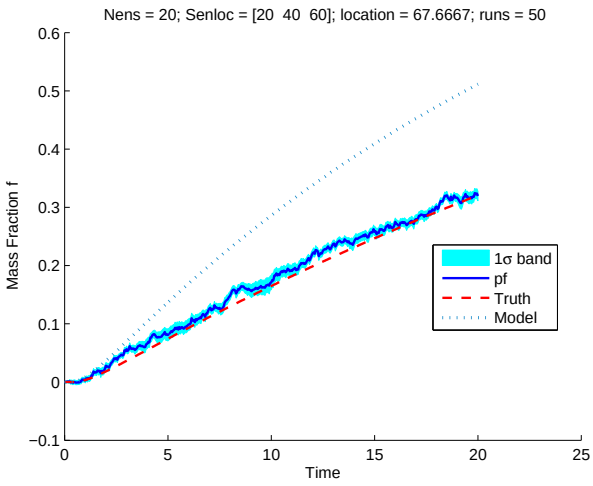

Figure 6: 1σ band of the PF estimate

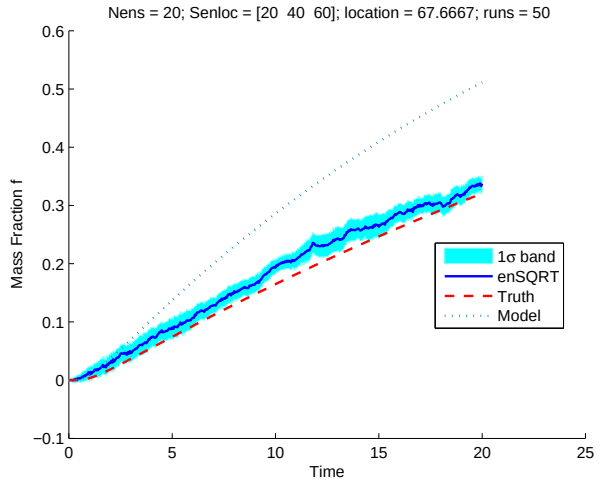

Figure 5: 1σ band of the SQRT estimate

For the same number of representations of the states ($N = 20$), it has been shown that the simple particle filter gives better results in terms of accuracy than the two ensemble filters, for the model considered. However, it has been noticed that the computational time required for the simple particle filter is also the highest among the three sub-optimal filters considered. The CPU time required for the simple particle filter for 50 Monte Carlo runs is $445.2813s$ whereas the same for the ensemble and the square root filters are $342.4688s$ and $364.3125s$ respectively. Note that the full Kalman filter takes $713.1563s$ for the same. Thus, it can be observed that the sub-optimal filters considered, offer significant improvement over the full Kalman filter in terms of computational time, while giving comparable estimates.

5 Conclusion

Various filtering schemes are implemented on the diffusion model and the results are compared with the full Kalman estimates. The ensemble and particle filters offer much advantage in terms of estimation time and computational power used, as they avoid the storage and propagation of the state covariance matrix. The Ensemble Square root filter offers significant improvement over the standard Ensemble Kalman filter in terms of the estimation error. While the simple particle filter gives the best results in terms of accuracy, it is also computationally expensive and takes longer than the two ensemble filters.

References

[1] Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory*, chapter 4-7. Academic Press, San Diego, CA, 1970.

[2] Simon J. Julier, Jeffrey K. Uhlmann, and Hugh F. Durrant-Whyte. A new approach for filtering nonlinear systems. In *Proceedings of the American Control Conference*, pages 1628–1632, Seattle, WA, June 1995.

[3] B. Ristic, S. Arulampalam, and N. Gordon. *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, chapter 3. Artech House, Boston, MA, 2004.

[4] F. Daum and J. Huang. Curse of dimensionality and particle filters. In *Proceedings of the IEEE Aerospace Conference*, volume 4, pages 1979–1993, Big Sky, MT, March 2003.

[5] John Lewis, S. Lakshmivarahan, and Sudarshan Dhall. *Dynamic Data Assimilation: A Least Squares Approach*. Cambridge University Press, Cambridge, UK, 2006.

[6] Geir Evensen. The ensemble kalman filter: Theoritical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003.

[7] Geir Evensen. Sampling strategies and square root analysis schemes for the enkf. *Ocean Dynamics*, 54:539–560, 2004.

[8] M. K. Tippett, J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker. Ensemble square-root filters. *Monthly Weather Review*, 131:1485–1490, 2002.

[9] Thomas M. Hamill, Jeffrey S. Whitaker, and Chris Snyder. Distance-dependent filtering of background error covariance estimates in an ensemble kalman filter. *Monthly Weather Review*, 129:2776–2790, 2001.

[10] A. Doucet, N. de Freitas, and N. Gordan, editors. *Sequential Monte Carlo Methods in Practice*, chapter 1. Springer, New York, NY, 2001.