Abstract

NGAMINI, MELISSA GUEMO. Nonlinear Filtering Problems for Systems Governed by PDEs. (Under the direction of Kazufumi Ito.)

In this thesis, we study how to efficiently solve nonlinear filtering and smoothing problems and joint state and parameters estimation problems. Data assimilation is one of the type of processes that we will be addressing as well since our approach encompasses a method for combining observations of variables into models. Also, our proposed methods can be applied for uncertainty quantification [20, 31]. Our objective is to develop smoothing and filtering algorithm for a large scale model dynamics, especially system governed by large scale partial differential equations dynamics. These large scale dynamics are commonly used in mathematical modeling, data assimilation and uncertainty quantification and have a lot of real life applications. Developing effective and efficient algorithm is essential due to the large scale and complex dynamics. This is why we must develop and analyze efficient but effective filtering algorithm to perform the task at hand.

Our approach is based on the optimal filtering theory; i.e., the optimal filter based on the Bayes' formula for discrete time dynamics and the Zakai equation for continuous time. After understanding the relationship between the discrete time and continuous time filter, we conclude that the discrete time filter with properly determined one step solution map can be applied directly to the continuous time filtering.

Well known result of Bayes' optimal filter is the Kalman filter for linear and Gaussian system. And our objective is to use Gaussian filter for nonlinear (significantly) system to improve the Kalman filter (extended). That is, we develop the filtering update via the assumed Gaussian density filter. A key step is that we update the covariance in the square root factors form and thus we update the square root factors of the Gaussian covariance. This evolves into the reduced Gaussian filter based on the reduced factor updates.

For dissipative system, we also develop an alternative to the reduced Gaussian filter, by the assumed covariance filter. For systems that are time reversible, we use the time reversal filter. As a result we obtain the forward and backward filter for time reversible systems. We also focus on the joint state-parameters estimate for parameters dependent problems such as media identification.

Nonlinear Filtering Problems for Systems Governed by PDEs

by
Melissa Guemo Ngamini

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Applied Mathematics

Raleigh, North Carolina

2015

APPROVED BY:

Zhilin Li	Hien Tran
Ralph Smith	Kazufumi Ito Chair of Advisory Committee

Dedication

To my parents Jean-Blaise and Therese Ngamini.

Biography

Melissa Guemo Ngamini was born in Miami, Florida on November 3rd 1981. She earned her Bachelor of Science in Mathematics in 2009 from Morgan State University in Baltimore Maryland. She began her graduate work at North Carolina State University in 2009 and earned her Masters of Science in Applied Mathematics in 2014.

Acknowledgements

First and foremost I would like to express my gratitude to my advisor, Dr. Kazufumi Ito, for his continued support and guidance. His dedication and insight has encouraged me to continually improve as a mathematician.

I would also like to thank my committee members especially Dr. Zhilin Li, Dr. Hien Tran and Dr. Ralph Smith for their professional support.

I am grateful to my family Jean-Blaise, Therese, Natasha, Mervyn and Darryl Ngamini for their support, encouragement, and patience.

Finally I would like to thank my friends and fellow graduate students. The late nights and the early mornings are memories I will always cherish.

Table of Contents

List of Figures

Chapter 1

Introduction

In this thesis we consider the nonlinear filtering problems, parameter estimation for a general class of stochastic dynamics. The nonlinear filtering and parameter estimation are very important aspects of mathematical modeling and have many significant applications in all sciences and engineerings, especially in fields of geosciences, weather forecasting, fluid dynamics and wave scattering. For example in weather forecasting, quantitative data is collected about the current state of the atmosphere at a given place and are used with a model to predict future state or to project how the atmosphere will change. A commonly used model dynamics to predict the physics and dynamics of the atmosphere are called primitive equations [27]. Therefore, the data collected is used to predict and estimate the state function and the parameters based on the model dynamics and the filtering algorithms. This suggests that the filtering methods require a good mathematical model, good algorithm and rich observations on which we can have a good estimate of the state [15, 19]. Developing good algorithms is what we are focused on throughout the thesis.

The Kalman filter is one of the very essential discovery for the last century and provides estimate of state and parameters. It has been used extensively in engineering and now extends further to other branch of sciences. It works well if the dynamics are nearly linear but it requires improvements for significantly nonlinear and non-Gaussian system. Contributions of the thesis includes an improved variation of Kalman filter for large scale nonlinear stochastic systems based on the Gaussian filter [2, 4, 8] and a noble implementation of Gaussian filters. The Gaussian filter is based on the optimal filtering theory; using the Bayesian formula [2] for discrete stochastic dynamics and the Zakai equation [5] for the continuous time stochastic differential dynamics. We develop a clear relationship between the discrete and continuous time filtering theory. It states that if we develop an accurate and effective one-step map in the discrete dynamics that approximates the solution of the equation driven by the drift term in the Itô stochastic differential equation for the continuous time, then the continuous time filtering problem can be well represented by the corresponding discrete time filtering problem, e.g. see the details in Chapter 2.

A large class of physical processes can be modeled by partial differential equations (PDE) stochastic differential equations, such as random wave equations, reaction diffusion equations and conservation laws. In such applications it is very essential to estimate the medium (function) parameters in terms of coefficients and source distributions. Our approach is to discretize in time and space in order to obtain an accurate and stable discrete time model that captures the original PDE dynamics. On this discretized model, we apply the proposed filter for the PDE governed system to perform the original filtering problem. The resulting system may be a very large system and could have many function parameters. Thus it is necessary to develop an efficient filter algorithm, since the covariance update for Gaussian (Kalman) filter is too expensive and requires a lot of storage. In order, to improve the performance of our proposed Gaussian filter we update the square root factor of the covariance and then develop an effective reduced order Gaussian filtering method. In this way we are able to reduce the storage need and the complexity of the system, moreover improve the performance.

In many applications the model dynamics is time reversible. There is a time reversal smoothing method for wave propagations [9] and we extend it and develop an assumed covariance filter. For example, for energy conservative systems we use the so-called dissipative filter and develop forward and backward filtering/smoothing algorithm. Since in general the diffusion systems are not time reversible, we use the quasi-reversible method to construct smoothing algorithms for systems with mild diffusion (or, the advection and convection dominant dynamics).

The other contribution of the thesis is the joint estimate of the state and parameters for parameter-dependent systems. As we mentioned parameter-dependent dynamics are very important in mathematical modeling and we develop the joint state-parameter estimate algorithm based on our proposed filters. Thus, we can use our proposed method for data assimilation and uncertainty quantification problems.

Now, we list some key topics which will be examined throughout the thesis.

- In Chapter 2: Nonlinear Filtering Problem and Gaussian Filter. We present the optimal filter based on the Baye's formula for discrete time system. We develop our Gaussian filter algorithm, which uses a Gaussian density assumption for nonlinear and non-Gaussian systems. The key step of the Gaussian filter is to represent the covariance in square root form and thus result in a square root filter. This Gaussian filter uses the square root filter to represent the covariance in square root form and uses a sampling and directional method. Realizing the Gaussian filter we use the CDF (central difference filter) which is a form of the directional sampling. The Lorenz 96 model and the wave equation model are the examples we tested our proposed Gaussian filter with. Finally, we develop the mixed Gaussian filter since even when we start with a single Gaussian initial condition, the resulting probability density function is not always Gaussian.

- In Chapter 3: Continuous Time Optimal Filter by Zakai Equation. We present the optimal filter which is described by the Zakai equation for nonlinear stochastic dynamics. To motivate the importance of an accurate and stable one step map from the discrete time to the continuous time dynamics, we present the relationship between the discrete time Bayes' formula and the Zakai equation for the continuous time filter. We develop the relationship between the discrete time and the continuous time filter and conclude that a proper one step solution map is necessary from the discrete time to the the continuous time filter for our proposed Gaussian filter algorithm. The parabolic stochastic system and conservation law are the PDE examples we examine. Finally we derive the Gaussian filter for the continuous time case.

- In Chapter 4: The Reduced Order Method. We develop the reduced order filter which reduces the sampling directions and thus the complexity of our proposed Gaussian filter. Singular value decomposition is used to reduced the square root factors for the Gaussian covariance update. To make our proposed filter effective and efficient, we truncate the square root factors by singular value decomposition and thus we reduce the directional sampling in CDF for the reduced Gaussian filter. We test our proposed reduced filter on the Lorenz 96 model, the wave speed and the wave with potential equations.

- In Chapter 5: Assumed Covariance and Time Reversal Filters. Some system have inherent dissipative property, we develop the assumed covariance filter for system that have a dissipative property. We develop the forward and backward smoothing method for system that are time reversible, which we used for the time reversal filter for estimating the initial condition. We used the Quasi reversible method for mildly diffusive system and thus not time reversible. We test our backward and smoothing method with the Burgers' equation and the time reversal filter on the advection equation. We also examine the Quasi reversible method with conduction equation.

- In Chapter 6: Parameter Dependent Case. Mathematical modeling can not be completed without the estimation of the parameters. We develop the joint state and parameters estimation algorithms for parameters dependent systems. We develop algorithms for joint state and parameters estimation for conservation laws and diffusion models. We examine these algorithms for the Lorenz equation, and PDE model dynamics. For the wave equation the function parameters the wave speed and the potential will be the function parameters estimated jointly with the states.

We also conclude our contribution in developing effective filtering/smoothing algorithms. Especially for PDEs model dynamics and parameters dependent models. We test our proposed algorithms (case by case) for concrete examples dynamics and demonstrate its capability.

Chapter 2

Nonlinear Filtering Problems and Gaussian Filter

In this chapter we discuss the discrete time stochastic dynamics and the optimal filter for nonlinear systems which is based on the Bayes' formula update of the probability density function. The optimal filter based on the Bayes' formula consists of sequentially updating the probability density function in two steps which are the predictor and corrector steps. Since the computation of the probability density function is not always feasible, next we discuss the Gaussian filter which updates the probability density function as a Gaussian distribution. That is, to use the Bayes' update formula for the predictor and corrector steps we employ an assumed density filter via Gaussian distribution and project the conditional probability density function onto the Gaussian distributional space. The key step of the Gaussian filter is to represent the covariance in square root form and update the covariance in the square root factor form [13, 22]. Even when we start with a single Gaussian initial distribution, the resulting probability density function is possibly no longer a Gaussian. That is why we develop the mixed Gaussian filter to approximate the probability density function by a sum of Gaussian distribution. The Lorenz 96 and the wave equation models are concrete examples we are going to use to test our Gaussian filter.

2.1 Discrete time filtering problem

In this section we introduce the discrete time and nonlinear filtering problems and their optimal filtering theory. To this end, we first introduce the discrete time stochastic process and present an example of one such system. If we suppose that our model dynamics is of the form

$$\begin{cases} x_k = f(x_{k-1}, a_{k-1}) + w_k, \\ a_k = a_{k-1} \\ y_k = h(x_k) + v_k \end{cases} \tag{2.1.1}$$

Here, $\{x_k\}$, represents the random process of the state, a_k are parameters in the model and $\{y_k\}$ is observation process. $f(x, a)$ is the one step map of the model dynamics and may depend on the parameter a and finally $h(x)$ is the observation map of the state x_k. Here, $w_k \in N(0, Q)$ and $v_k \in N(0, R)$ are the noise in the system and the observation respectively, they are often assumed to be independent and identically distributed (*i.i.d.*) random variables. For example if we consider the stochastic (Itô) DE model,

$$dX_t = b(X_t, a)\, dt + \sigma(X_t) dB_t \tag{2.1.2}$$

(2.1.2) is an SDE with drift $b(X_t, a)$ and diffusion coefficient $\sigma(X_t)$, while B_t is the Brownian motion process. The corresponding one step map in this case is

$$f(x_{k-1}, a_{k-1}) = x_{k-1} + \Delta t\, b(x_{k-1}, a_{k-1}), \quad w_k = \sigma(x_{k-1}) \Delta B_k \tag{2.1.3}$$

where $\Delta t > 0$ is the time step size and $\Delta B_k = B_{t_k} - B_{t_{k-1}}$ is the Brownian motion increment. For the discrete time system in (2.1.1), the nonlinear filtering problem is to determine (estimate) the (x_T, a) from observation $\{y_k\}$, $1 \le k \le T$ where $T > 0$ is the current time. Assume model dynamics f and observation map h are known and system and observation $\{w_k\}$ and $\{v_k\}$ are i.i.d. Gaussian random variables with covariance Q and R, respectively and x_0 is a Gaussian random variable with covariance P_0, which is independent with $\{w_k\}$ and $\{v_k\}$. The optimal nonlinear filtering theory is to determine the conditional expectation $E[x_k|Y_k]$ of the state process x_k, given the observations data $Y_k = \sigma\{y_j, 1 \le j \le k\}$ [22, 25, 30]. The conditional expectation $E[x_k|Y_k]$ is given by

$$E\left[f(x_k)|Y_k\right] = \int_{R^n} f(x) p_{k|k}(x)\, dx \text{ for all } f \in C(R^n) \tag{2.1.4}$$

where $p_{k|k}(x)$ is the conditional probability density function of x_k with respect to Y_k. This means that in order to solve discrete time nonlinear filtering problems, we need to have a good understanding on how to compute $p_{k|k}(x)$.

2.2 Bayes' formula for the discrete time optimal filtering

Now, we discuss the optimal filtering theory for the discrete time system (2.1.1) based on the Bayes' update for the conditional probability density in $p_{k|k}$ in (2.1.4) [11, 13]. The conditional probability density is updated sequentially in two steps; first the prediction step which is based on the model dynamics, and the correction step which uses the new observation y_k. In the predictor step, we have the Bayes' update for the conditional density by

$$p_{k|k-1}(x) = \int_{\mathbb{R}^n} \frac{1}{\sqrt{(2\pi)^n \det Q}} \exp\left[-\frac{1}{2}(x - f(t))^T Q^{-1}(x - f(t))\right] p_{k-1|k-1}(t)dt. \tag{2.2.1}$$

It is based on the model dynamics and the one step map $f(x)$ (the first equation of (2.1.1)). The corrector step is

$$p_{k|k}(x) = c \exp\left[-\frac{1}{2}(y_k - h(x))^T R^{-1}(y_k - h(x))\right] p_{k|k-1}(x) \tag{2.2.2}$$

where $c > 0$ is the normalization constant, $p_{k|k-1}$ is the one step prediction and is the probability density function of x_k conditioned at Y_{k-1}. The corrector update is based on the observation map $h(x)$ (the third equation of (2.1.1)), and uses the innovation process which will be explained in depth in Section 2.3.

In summary, the formulae (2.2.1)–(2.2.2) is the recursive filter that consists of the predictor (2.2.1) and the corrector (2.2.2) steps. It is an optimal filter and provides an exact solution in the case of linear Gaussian systems [19].

2.3 Kalman filter for linear Gaussian system

An example of the analytic form of the optimal filter that is linear and Gaussian is the Kalman filter [15, 19]. It operates recursively on noisy input data to produce a statistically optimal estimate of the underlying system state. The Kalman filter is also the most well know case where one can derive the exact form of the optimal filter based on (2.2.1)-(2.2.2) for linear Gaussian system. Suppose we consider the linear dynamics:

$$\begin{cases} x_k = A_k x_{k-1} + b_{k-1} + w_k, \quad x_0 \in N(m, \Sigma_0) \\ \\ y_k = H_k x_k + v_k, \quad k \geq 1 \end{cases} \tag{2.3.1}$$

where the system matrix $A_k \in \mathbb{R}^{n\times n}$ and the observation matrix $H_k \in \mathbb{R}^{p\times n}$ are given. If x_0 is a Gaussian (independent with (w_k, v_k)), then $(x_k,\, y_k)$ will turn out to be linear and jointly

Gaussian. Since the sum of Gaussian random variables is Gaussian, thus all the conditional densities are Gaussian and thus we assume that $p_{k|k} = N(\hat{x}_{k|k}, \Sigma_{k|k})$ is a Normal distribution with mean $\hat{x}_{k|k}$ and covariance $P_{k|k}$. From the predictor step (2.2.1) we obtain

$$\begin{aligned} p_{k|k-1}(x) = \textstyle\int_{\mathbb{R}^n} & \tfrac{1}{\sqrt{(2\pi)^n \det \Sigma_{k-1|k-1}}} \\ & \times \exp\left[-\tfrac{1}{2}(t - \hat{x}_{k-1|k-1})^T \Sigma_{k-1|k-1}^{-1} (t - \hat{x}_{k-1|k-1})\right] p_{k-1|k-1}(t) dt \end{aligned} \tag{2.3.2}$$

It is equivalent to the predictor step of the Kalman filter: the mean $\hat{x}_{k|k-1}$ is

$$\begin{aligned} \hat{x}_{k|k-1} &= E[x_k | Y_{k-1}] \\ &= E_{k|k-1}[A_k x_{k-1} + b_k + v_k | Y_{k-1}] \\ &= A_k \hat{x}_{k-1|k-1} + b_k \end{aligned}$$

and the covariance update $\Sigma_{k|k-1}$ is

$$\begin{aligned} \Sigma_{k|k-1} &= E_{k|k-1}[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^T] \\ &= E_{k|k-1}[(A_k x_{k-1} + v_k - A_k \hat{x}_{k-1|k-1})(A_k x_{k-1} + v_k - A_k \hat{x}_{k-1|k-1})^T | Y_{k-1}] \\ &= E_{k|k-1}[(A_k(x_{k-1} - \hat{x}_{k-1|k-1}))(A_k(x_{k-1} - \hat{x}_{k-1|k-1}))^T | Y_{k-1}] + Q \\ &= E_{k|k-1}[A_k(x_{k-1} - \hat{x}_{k-1|k-1})(x_{k-1} - \hat{x}_{k-1|k-1})^T A_k^T | Y_{k-1}] + Q \\ &= A_k \Sigma_{k-1|k-1} A_k^T + Q \end{aligned}$$

where

$$\Sigma_{k|k-1} = E_{k|k-1}[(x_{k-1} - \hat{x}_{k-1|k-1})(x_{k-1} - \hat{x}_{k-1|k-1})^T | Y_{k-1}].$$

In summary $p_{k|k-1}$ is a Gaussian with mean

$$\hat{x}_{k|k-1} = A_k \hat{x}_{k-1|k-1} + b_k \tag{2.3.3}$$

and covariance

$$\Sigma_{k|k-1} = A_k \Sigma_{k-1|k-1} A_k^T + Q. \tag{2.3.4}$$

The corrector step (2.2.2) for the linear case is equivalent to the innovation process approach by Kailath [18]. First, we define the innovation process as

$$I_k = y_k - E_{k|k-1}[y_k] = y_k - E_{k|k-1}[H_k x_k] = H x_k - H_k \hat{x}_{k|k-1} + v_k. \tag{2.3.5}$$

One can prove that $\{I_k\}$ are independent Gaussian random variables and is orthogonal to Y_{k-1} [18]. Thus, we have

$$\hat{x}_{k|k} = E[x_k|Y_k] = \hat{x}_{k|k-1} + GI_k$$

for some $G \in \mathbb{R}^{n\times p}$. We find G as follows. First,

$$GE[I_k I_k^*] = E[(x_{k|k} - \hat{x}_{k|k-1})I_k^*] = E[x_k I_k^*]$$

and we have

$$\begin{aligned} E[x_k I_k^*] &= E[x_k(w_k + H_k x_k - \hat{x}_{k|k-1})^*)] \\ &= E[x_k(x_k - \hat{x}_{k|k-1})^*]H_k^* \\ &= E[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^*]H_k^* \\ &= \Sigma_{k|k-1}H_k^* \end{aligned}$$

and

$$E[I_k I_k^*] = R_k + H_k \Sigma_{k|k-1} H_k^*.$$

Thus we obtain

$$G = G_k = \Sigma_{k|k-1} H_k^* \left(R_k + H_k \Sigma_{k|k-1} H_k^*\right)^{-1}.$$

Next we define the error covariance update

$$\begin{aligned} \Sigma_{k|k} = E[(x_k - \hat{x}_{k|k})(x_k - \hat{x}_{k|k})^*] &= E[(x_k - \hat{x}_{k|k-1} - GI_k)(x_k - \hat{x}_{k|k-1} - GI_k)^*] \\ &= E[(x_k - \hat{x}_{k|k-1})(x_k - \hat{x}_{k|k-1})^*] - GE[I_k I_k^*]G^* \\ &= \Sigma_{k|k-1} - \Sigma_{k|k-1}H_k^*(R_k + H_k\Sigma_{k|k-1}H_k^*)^{-1}H_k\Sigma_{k|k-1} \\ &= \Sigma_{k|k-1} - G_k H_k \Sigma_{k|k-1}. \end{aligned}$$

In summary the corrector step of (2.2.2) for the Kalman filter ($p_{k|k} = N(\hat{x}_{k|k}, \Sigma_{k|k})$) is a Gaussian with mean

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + G_k(y_k - H_k \hat{x}_{k|k-1}) \tag{2.3.6}$$

and covariance update

$$\Sigma_{k|k} = \Sigma_{k|k-1} - G_k H_k \Sigma_{k|k-1}, \tag{2.3.7}$$

where G_k is the Kalman gain and is defined by

$$G_k = \Sigma_{k|k-1} H_k^T (H_k \Sigma_{k|k-1} H_k^T + R)^{-1}. \tag{2.3.8}$$

As we have stated previously the Kalman filter derives and exact form of the optimal filter based (2.2.1)-(2.2.2) for linear Gaussian system. While the optimal filter based on (2.2.1)-(2.2.2) is optimal, in general it may not be possible to use the steps (2.2.1)–(2.2.2) directly in computations. We develop a Gaussian filter to perform the updates of the conditional densities using assumed Gaussian density for nonlinear cases.

2.4 Gaussian filter

In this section we describe the Gaussian filter which employs the Bayes' update for nonlinear and non Gaussian system using an assumed Gaussian distribution for conditional probability $p_{k|k}(x)$. In order to practically achieve the Bayes' update (2.2.1)-(2.2.2) we use an assumed density filter with a single Gaussian distribution. Since we have assumed the probability density function to be Gaussian, this means that we will have to update the mean and covariance. This involves setting $p_{k-1|k-1} = N(x_{k-1|k-1}, P_{k-1|k-1})$ and project onto the update formula (2.2.1). That is, to obtain the mean we calculate the conditional expectation by

$$
\begin{aligned}
E_{k|k-1}[x] &= \int_{\mathbb{R}^n} x p_{k|k-1}(x) dx \\
&= \int_{\mathbb{R}^n} \left(\int_{\mathbb{R}^n} \frac{1}{\sqrt{(2\pi)^n \det Q}} \exp\left[-\frac{1}{2}(x - f(t))^T Q^{-1} (x - f(t)) \right] p_{k-1|k-1}(t) dt \right) x dx \\
&= \int_{\mathbb{R}^n} \left(\int_{\mathbb{R}^n} \frac{1}{\sqrt{(2\pi)^n \det Q}} \exp\left[-\frac{1}{2}(x - f(t))^T Q^{-1} (x - f(t)) \right] x dx \right) p_{k-1|k-1}(t) dt \\
&= \int_{\mathbb{R}^n} f(t) p_{k-1|k-1}(t) dt.
\end{aligned}
$$

For the predictor step, to get the mean it turns out that we evaluate the one step map $f(t)$ against the Gaussian density function. Thus, the mean is

$$
\begin{aligned}
x_{k|k-1} = \int_{\mathbb{R}^n} f(t) & \frac{1}{\sqrt{(2\pi)^n \det P_{k-1|k-1}}} \\
& \times \exp\left[-\frac{1}{2}(t - x_{k-1|k-1})^T P^{-1}_{k-1|k-1} (t - x_{k-1|k-1}) \right] dt.
\end{aligned} \tag{2.4.1}
$$

Next, we obtain the covariance by

$$\begin{aligned}
E_{k|k-1}[xx^T] &= \int_{\mathbb{R}^n} xx^T p_{k|k-1}(x)dx \\
&= \int_{\mathbb{R}^n} \left(\int_{\mathbb{R}^n} \frac{1}{\sqrt{(2\pi)^n \det Q}} \exp\left[-\frac{1}{2}(x - f(t))^T Q^{-1}(x - f(t)) \right] p_{k-1|k-1}(t)dt \right) xx^T dx \\
&= \int_{\mathbb{R}^n} \left(\int_{\mathbb{R}^n} \frac{1}{\sqrt{(2\pi)^n \det Q}} \exp\left[-\frac{1}{2}(x - f(t))^T Q^{-1}(x - f(t)) \right] xx^T dx \right) p_{k-1|k-1}(t)dt \\
&= Q + \int_{\mathbb{R}^n} f(t)f(t)^T p_{k-1|k-1}(t)dt.
\end{aligned}$$

Thus for the covariance we evaluate $f(t)f(t)^T$ against the assumed Gaussian distribution. Thus the covariance is

$$\begin{aligned}
P_{k|k-1} = Q + \int_{\mathbb{R}^n} (f(t) - x_{k|k-1})(f(t) - x_{k|k-1})^T \frac{1}{\sqrt{(2\pi)^n \det P_{k-1|k-1}}} \\
\times \exp\left[-\frac{1}{2}(t - x_{k-1|k-1})^T P_{k-1|k-1}^{-1}(t - x_{k-1|k-1}) \right] dt.
\end{aligned} \tag{2.4.2}$$

For the corrector step for the Gaussian filter from (2.2.2), we have to approximate $E_{k|k-1}[H_k x_k]$ by its Gaussian approximation z where the mean $\tilde{z}$ is obtain by evaluating the observation map $h(t)$ against $p_{k|k-1}$

$$\begin{aligned}
\tilde{z} &= \int_{\mathbb{R}^n} h(t)p_{k|k-1}(t)dt \\
&= \int_{\mathbb{R}^n} h(t) \frac{1}{\sqrt{(2\pi)^n \det P_{k|k-1}}} \exp\left[-\tfrac{1}{2}(t - x_{k|k-1})^T P_{k|k-1}^{-1}(t - x_{k|k-1}) \right] dt
\end{aligned} \tag{2.4.3}$$

and the covariance P_{zz} of z is obtain by evaluating $h(t)h(t)^T$ against $p_{k|k-1}$

$$\begin{aligned}
P_{zz} &= \int_{\mathbb{R}^n} h(t)h(t)^T p_{k|k-1}(t)dt \\
&= \int_{\mathbb{R}^n} (h(t) - \tilde{z})(h(t) - \tilde{z})^T \frac{1}{\sqrt{(2\pi)^n \det P_{k|k-1}}} \\
&\quad \times \exp\left[-\frac{1}{2}(t - x_{k|k-1})^T P_{k|k-1}^{-1}(t - x_{k|k-1}) \right] dt.
\end{aligned} \tag{2.4.4}$$

Now using the innovation process as in Section 2.3, we can construct the Gaussian update of $p_{k|k}$ with mean $x_{k|k}$ and covariance $P_{k|k}$ defined by

$$x_{k|k} = x_{k|k-1} + L_k(y_k - \tilde{z}) \tag{2.4.5}$$

and

$$P_{k|k} = P_{k|k-1} + L_k P_{xz}^T. \tag{2.4.6}$$

L_k is the Gaussian filter gain defined by

$$L_k = P_{xz}\left(R + P_{zz}\right)^{-1} \tag{2.4.7}$$

and the covariance P_{xz} is defined by

$$\begin{aligned} P_{xz} = \int_{\mathbb{R}^n} &(t - x_{k|k-1})(h(t) - \tilde{z})^T \frac{1}{\sqrt{(2\pi)^n \det P_{k|k-1}}} \\ &\times \exp\left[-\frac{1}{2}(t - x_{k|k-1})^T P_{k|k-1}^{-1}(t - x_{k|k-1})\right] dt. \end{aligned} \tag{2.4.8}$$

In order to realize the Gaussian filter we need to develop a method to evaluate integrals (2.4.1) to (2.4.8). The first thing we will need to do is to change the covariance matrices that are in our formulae to transform the integrals into something easier to compute.

2.5 Directional sampling method and Central difference filter (CDF)

In this section, we discuss how we evaluate the Gaussian integrals (2.4.1)–(2.4.8) for the Gaussian filter in Section 2.4. We use the change of coordinate transform (2.5.2) by square root factors of the covariance Σ to transform the Gaussian integrals into normalized Gaussian integrals. Next, we will discuss how to evaluate the normalized Gaussian integral by a sampling method via central difference. Alternatives are the Gauss-Hermite quadrature rule [13] and the Julier-Uhlmann [13, 16, 17]. Thus, we develop our proposed Central difference filter (CDF) that use a directional samplings of functions.

Now, we define the general form of Gaussian integrals:

$$I = \int_{\mathbb{R}^n} F(t) \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} \exp\left[-\frac{1}{2}(t - \bar{x})^T \Sigma^{-1}(t - \bar{x})\right] dt \tag{2.5.1}$$

where $F(t)$ is a given function. The first (key) step for evaluating I in (2.5.1) is to take the coordinate change by [13]

$$t = S^T s + \bar{x} \tag{2.5.2}$$

where $\Sigma = S^T S$ and we obtain

$$I = \int_{\mathbb{R}^n} F(S^T s + \bar{x}) \frac{1}{\sqrt{(2\pi)^n}} e^{-\frac{1}{2}|s|^2} \, ds \tag{2.5.3}$$

which is the standard weighted integral with the Gaussian weight. One of the most used method to calculate this integral is the Gauss-Hermite quadrature rule. It is given by

$$\int_{-\infty}^{\infty} g(x) \frac{1}{(2\pi)^{1/2}} e^{-x^2} \, dx = \sum_{i=1}^{m} w_i g(x_i)$$

where equality holds for all polynomials of degree up to $2m-1$. The sample points x_i and weights w_i are determined as followed.
If we let J be a symmetric tridiagonal matrix with zeros diagonals and $J_{i,i+1} = \sqrt{\frac{i}{2}}$ for $1 \leq i \leq m-1$. Then $\{x_i\}$ are the eigenvalues of the matrix J and $w_i = |(v_i)_1|^2$ where $(v_i)_1$ is the first element of the i-th normalized eigenvector of J. Thus, I is approximated by

$$I_m = \sum_{i_1=1}^{m} \cdots \sum_{i_n=1}^{m} \tilde{F}(q_{i_1} \cdots q_{i_n}) w_{i_1} \cdots w_{i_n} \tag{2.5.4}$$

where $q_i = x_i \times \sqrt{2}, \ 1 \leq i \leq m$ and $\tilde{F}(s) = F(S^T s + \bar{x})$. If we use m samples in each direction we have d^m number of evaluations where $s \in R^d$, which grows very rapidly with respect to d (dimension of space) and m (number of evaluation points).

Now, we introduce the evaluation method using the polynomial interpolation of $\tilde{F}(s)$. We use the quadratic function P for interpolation of $\tilde{F}$:

$$\tilde{F}(\epsilon_i) \sim P(s) = \tilde{F}(0) + \sum_{i=1}^{d} \left(a_i s_i + \frac{1}{2} H_{ii} |s_i|^2 \right) \tag{2.5.5}$$

where s_i is the i-th coordinate of a point $s \in R^d$, $a = a_i \in R^d$ is a central difference vector and $H = H_{ii}$ is the second order central difference of the diagonal elements of $\tilde{F}$.

$$a_i = \frac{\tilde{F}(he_i) - \tilde{F}(-he_i)}{2h}, \qquad 1 \leq i \leq d \tag{2.5.6}$$

$$H_{ii} = \frac{\tilde{F}(he_i) - 2\tilde{F}(0) + \tilde{F}(-he_i)}{h^2}, \qquad 1 \leq i \leq d \tag{2.5.7}$$

Therefore we have the central difference rule:

$$\begin{aligned} I \sim \hat{I} &= \textstyle\int_{R^d} P(s) \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2}|s|^2} \, ds \\ &= \tilde{F}(0) + \sum_{i=1}^{d} \frac{1}{2} H_{ii} \end{aligned} \tag{2.5.8}$$

and the integral

$$J = \int_{R^d} \tilde{F}_1(s) \tilde{F}_2(s) \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2}|s|^2} \, ds$$

by

$$\begin{aligned} J \sim \hat{J} &= \int_{R^d} (P^{(1)}(s) - \hat{I}^{(1)})(P^{(2)}(s) - \hat{I}^{(2)}) \frac{1}{(2\pi)^{d/2}} e^{-\frac{1}{2}|s|^2} \, ds \\ &= \sum_{i=1}^{d} \left(a_i^{(1)} a_i^{(2)} + \frac{1}{2} H_{ii}^{(1)} H_{ii}^{(2)} \right). \end{aligned} \tag{2.5.9}$$

In summary the predictor step of the central difference filter (CDF) gives us $p_{k|k-1} = N(x_{k|k-1}, P_{k|k-1})$ with mean

$$x_{k|k-1} = f(x_{k-1|k-1}) + \frac{1}{2} \sum_{i=1}^{d} H_{ii} \tag{2.5.10}$$

and covariance update

$$P_{k|k-1} = Q_k + \sum_{i=1}^{d} \left(a_i a_i^T + \frac{1}{2} H_{ii} H_{ii}^T \right) \tag{2.5.11}$$

For the corrector step we update $p_{k|k} = N(x_{k|k}, P_{k|k})$ with mean update

$$x_{k|k} = x_{k|k-1} + L_k(y_k - z_k) \tag{2.5.12}$$

and the covariance update is

$$P_{k|k} = P_{k|k-1} - L_k P_{xz}^T \tag{2.5.13}$$

where

$$z_k = H(x_{k|k-1}) + \sum_{i=1}^{d} \frac{1}{2} G_{ii}$$

$$P_{xz} = \tilde{S}^T (b_1, \cdots, b_d)^T$$

$$P_{zz} = \sum_{i=1}^{d} b_i b_i^T + \sum_{i=1}^{d} \frac{1}{2} G_{ii} G_{ii}^T$$

$$L_k = P_{xz}(R + P_{zz})^{-1}$$

and

$$b_i = \frac{\tilde{h}(he_i) - \tilde{h}(-he_i)}{2h}, \qquad 1 \leq i \leq d \tag{2.5.14}$$

$$G_{ii} = \frac{\tilde{h}(he_i) - 2\tilde{h}(0) + \tilde{h}(-he_i)}{h^2}, \qquad 1 \leq i \leq d \tag{2.5.15}$$

The CDF (2.5.8)-(2.5.9) are only based on the values $\tilde{F}(\pm he_i)$.

Remarks:

(1) The square factor S of Σ can be obtained by the Cholesky decomposition.

(2) We only use $2d + 1$ number of the sample points; which grows linearly in dimension d.

(3) The step size $h > 0$ can be adjusted, for example we can sample from the constrained set (e.g., nonnegative $(R^+)^d$).

(4) If we use the singular value decomposition $\Sigma = V\tilde{\Sigma}V^T$ one can truncate $\tilde{\Sigma}$ by the singular values (we only use the dominant singular values of $\tilde{\Sigma}$). In practice it offers a significant reduction in number of function evaluations. See the details in Chapter 4: Reduced Order Method.

(5) If we remove the second-order correction term H_{ii} in (2.5.10) and (2.5.11) then the CDF coincides with the extended Kalman filter with the Jacobian of f computed by its central difference (2.5.6). That means that for the predictor step we obtain

$$x_{k|k-1} = f(x_{k-1|k-1})$$

$$P_{k|k-1} = Q_k + \sum_{i=1}^{d} a_i a_i^T.$$

We use the square root factors for the covariance, thus it is an extended Kalman filter in a stable and robust manner and it works better. Without the square root factor, the Kalman filter may not work. This method will be used in a similar manner for the corrector step.

(6) The accuracy of the predictor step of the CDF is of second order with the term $\frac{1}{2}\sum_i H_{ii}$ and the covariance update contains higher moments.

2.6 Mixed Gaussian filter

In general even if we start from a single Gaussian initial condition the (conditional) probability density $p_{k|k}(x)$ is no longer Gaussian. In this section we will discuss the mixed Gaussian filter in which we approximate the conditional density $p_{k|k}(x)$ by a linear combination of a sum of Gaussians. Since the mean and covariance for the predictor and corrector steps can be updated using the CDF formulae (2.5.10) to (2.5.13), we will discuss how to update the weights for the predictor and corrector steps.

For example consider

$$dx_t = x_t(1 - x_t^2)\, dt + \sigma\, dB_t$$

Then, the density function $p(t, x)$ of x_t satisfies the Fokker-Planck equation

$$\frac{\partial p}{\partial t} - (b(x)p)_x = \frac{\sigma^2}{2}\, \frac{\partial u}{\partial x^2} \tag{2.6.1}$$

where $b(x) = x(1 - x^2)$. In this example we start with $p(0, x) = N(0, 1)$ the solution $p(1, x)$ is not Gaussian but a bi-modal distribution.

As with this example one can approximate the probability density function by a sum of Gaussian distributions of the form:

$$p_{k|k}(x) \sim \sum_{i=1}^{m} \alpha_k^{(j)}\, N\left(m_k^{(j)}, P_k^{(j)}\right). \tag{2.6.2}$$

We will address how to update $(m_k^{(j)}, P_k^{(j)})$ for the predictor and corrector steps and also the update formula for the weight $\alpha_k^{(j)}$. Then, one can have the mixed Gaussian filter as a method to improve the performance of the Gaussian filter.

For the predictor step, we assume that the conditional probability density is a sum of Gaussian distributions, i.e., for the mixed Gaussian filter, we approximate $p_{k-1|k-1}$ by the linear combination of multiple Gaussian distributions, of the form

$$\begin{aligned} p_{k-1|k-1}(x) = \sum_{i=1}^{m} \alpha_{k-1|k-1}^{(i)} & \frac{1}{\sqrt{(2\pi)^n \det P_{k-1|k-1}^{(i)}}} \\ & \times \exp\left[-\frac{1}{2}\left(x - x_{k-1|k-1}^{(i)}\right)^T \left(P_{k-1|k-1}^{(i)}\right)^{-1} \left(x - x_{k-1|k-1}^{(i)}\right)\right] \end{aligned} \tag{2.6.3}$$

with weight $\alpha_{k-1|k-1}^{(i)}$ for each i-th Gaussian distribution at the time $k-1$.

The update of the mean and covariance is done by applying (2.5.10) for each individual Gaussian distribution $N\left(x_{k-1|k-1}^{(i)}, P_{k-1|k-1}^{(i)}\right)$ separately. Now, we describe the update method for weights $\alpha_{k|k-1}^{(j)}$ in

$$p_{k|k-1}(x) = \sum_{i=1}^{m} \alpha_{k|k-1}^{(i)} N\left(x_{k|k-1}^{(i)},\ P_{k|k-1}^{(i)}\right).$$

Note that

$$\begin{aligned} \tilde{\alpha}_k^{(j)} = p_{k|k-1}(x_{k|k-1}^{(j)}) \quad = & \quad \sum_{i=1}^{m} \alpha_{k-1|k-1}^{(i)} \frac{1}{\sqrt{(2\pi)^n P_{k|k-1}^{(i)}}} \\ & \times \exp\left[\ -\frac{1}{2}\left(x_{k|k-1}^{(j)} - x_{k|k-1}^{(i)}\right)^T \left(P_{k|k-1}^{(i)}\right)^{-1} \left((x_{k|k-1}^{(j)} - x_{k|k-1}^{(i)}\right)\right] \end{aligned}$$

where $\alpha_{k-1|k-1}^{(i)}$ is the i-th weight at the time $k-1$. Now we will update the weight $\alpha_{k|k-1}^{(j)}$ for the predictor step. by normalizing $\tilde{\alpha}_k^{(j)}$ by

$$\alpha_{k|k-1}^{(j)} = \frac{\tilde{\alpha}_k^{(j)}}{\sum_{i=1}^{m} \alpha_k^{(i)}}. \tag{2.6.4}$$

For the corrector step, once again, we update mean $x_{x|k}^{(i)}$ and covariance $P_{k|k}^{(i)}$, separately based on (2.5.12) and (2.5.13) respectively. Each update is independent from the other and can be performed in a parallel manner. Thus, we obtain

$$\begin{aligned} p_{k|k}(x) = \sum_{i=1}^{m} \alpha_k^{(i)} & \frac{1}{\sqrt{(2\pi)^n \det P_{k|k-1}^{(i)}}} \\ & \times \exp\left[-\frac{1}{2}\left(x - x_{k|k-1}^{(i)}\right)^T \left(P_{k|k-1}^{(i)}\right)^{-1} \left(x - x_{k|k-1}^{(i)}\right)\right]. \end{aligned} \tag{2.6.5}$$

Now, we update the weight $\alpha_k^{(i)}$ is based on the Baye's formula

$$p_{k|k}(x) \sim \exp\left[-\frac{1}{2}(y-h(x))^T R^{-1}(y-h(x))p_{k|k-1}(x)\right] \tag{2.6.6}$$

with assumed mixed Gaussian density

$$\begin{aligned} p_{k|k-1}(x) = \sum_{i=1}^{m} \alpha_{k|k-1}^{(i)} & \frac{1}{\sqrt{(2\pi)^n P_{k|k-1}^{(i)}}} \\ \times \exp\Bigg[& -\frac{1}{2}\left(x - x_{k|k-1}^{(i)}\right)^T \left(P_{k|k-1}^{(i)}\right)^{-1} \left((x - x_{k|k-1}^{(i)}\right)\Bigg]. \end{aligned} \tag{2.6.7}$$

There are three types of approaches to update the weights [13].

Moment Approach

For the Moment Approach, first we equate the zero moment of each Gaussian distribution and obtain

$$\begin{aligned} &\alpha_k^{(i)} \int_{\mathbb{R}^n} \frac{1}{\sqrt{(2\pi)^n \det P_{k|k}^{(i)}}} \exp\left[\frac{1}{2}\left(x - x_{k|k}^{(i)}\right)^T \left(P_{k|k}^{(i)}\right)^{-1} \left(x - x_{k|k}^{(i)}\right)\right] dx \\ &= \alpha_{k-1}^{(i)} \int_{\mathbb{R}^n} \frac{1}{\sqrt{(2\pi)^n \det P_{k|k-1}^{(i)}}} \exp\Bigg[-\frac{1}{2}\left(y - h(x)\right)^T R^{-1}\left(y - h(x)\right) \\ &+ \left(x - x_{k|k-1}^{(i)}\right)^T \left(P_{k|k-1}^{(i)}\right)^{-1} \left(x - x_{k|k-1}^{(i)}\right)\Bigg] dx. \end{aligned}$$

Now we approximate the right hand side by the method as in (2.4.1)-(2.4.8) and we get

$$\alpha_k^{(i)} = \alpha_{k-1}^{(i)} \frac{1}{\sqrt{(2\pi)^n \det\left(R + P_{zz}\right)}} \exp\left[-\frac{1}{2}(y-\tilde{z})^T (R + P_{zz})^{-1}(y - \tilde{z})\right] \tag{2.6.8}$$

where $\tilde{z}$ and P_{zz} are defined by (2.4.3) and (2.4.4).

Collocation Approach

For the Collocation Approach, we apply the collocation condition at $x = x_{k|k}^{(i)}$, thus we have

$$\begin{aligned} p_{k|k}(x_{k|k}^{(i)}) &= \alpha_k^{(i)} \frac{1}{\sqrt{(2\pi)^n \det P_{k|k}^{(i)}}} \exp\left[-\frac{1}{2}\left(x_{k|k}^{(i)} - x_{k|k}^{(i)}\right)^T \left(P_{k|k}^{(i)}\right)^{-1} \left(x_{k|k}^{(i)} - x_{k|k}^{(i)}\right)\right] \\ &= \alpha_k^{(i)} \frac{1}{\sqrt{(2\pi)^n \det P_{k|k}^{(i)}}}. \end{aligned}$$

On the other hand from (2.2.2) we have

$$\begin{aligned} p_{k|k}(x_{k|k}^{(i)}) = \alpha_{k-1}^{(i)} \frac{1}{\sqrt{(2\pi)^n \det P_{k|k-1}^{(i)}}} \exp\Bigg[-\frac{1}{2}\Bigg(\left(y - h\left(x_{k|k}^{(i)}\right)\right)^T R^{-1} \left(y - h\left(x_{k|k}^{(i)}\right)\right) \\ + \left(x_{k|k}^{(i)} - x_{k|k-1}^{(i)}\right)^T \left(P_{k|k-1}^{(i)}\right)^{-1} \left(x_{k|k}^{(i)} - x_{k|k-1}^{(i)}\right)\Bigg)\Bigg]. \end{aligned}$$

Thus we obtain

$$\begin{aligned} \alpha_k^{(i)} = \alpha_{k-1}^{(i)} \sqrt{\frac{\det P_{k|k}^{(i)}}{\det P_{k|k-1}^{(i)}}} \exp\Bigg[-\frac{1}{2}\Bigg(\left(y - h\left(x_{k|k}^{(i)}\right)\right)^T R^{-1} \left(y - h\left(x_{k|k}^{(i)}\right)\right) \\ + \left(x_{k|k}^{(i)} - x_{k|k-1}^{(i)}\right)^T \left(P_{k|k-1}^{(i)}\right)^{-1} \left(x_{k|k}^{(i)} - x_{k|k-1}^{(i)}\right)\Bigg)\Bigg] \end{aligned} \tag{2.6.9}$$

Least Square Approach

The Least square approach is the simultaneous update of the weights. This means that we will determine the weights $\alpha_k^{(i)}$ by the L^2-projection, that is $\alpha_k^{(i)},\ 1 \le i \le m$ minimizes

$$\begin{aligned} \int_{\mathbb{R}^n} \Bigg| p_{k|k}(x) - \sum_{i=1}^{m} \frac{1}{\sqrt{(2\pi)^n \det P_{k|k}^{(i)}}} \\ \times \exp\left[-\frac{1}{2}\left(x - x_{k-1|k-1}^{(i)}\right)^T \left(P_{k-1|k-1}^{(i)}\right)^{-1} \left(x - x_{k-1|k-1}^{(i)}\right)\right]\Bigg|^2 dx \end{aligned} \tag{2.6.10}$$

over $(\mathbb{R}^+)^m$, where $p_{k|k}$ is defined as in equation (2.2.2) with $p_{k|k-1}$ defined as in (2.6.7).

In order to perform this minimization (2.6.10) we need to evaluate integral of the form

$$\int_{\mathbb{R}^n} \exp\left[-\frac{1}{2}(y-h(x))^T R^{-1}(y-h(x))\right] \frac{1}{\sqrt{(2\pi)^n \det \Sigma}} \exp\left[-\frac{1}{2}(x-\bar{x})^T \Sigma^{-1}(x-\bar{x})\right]$$

this is relatively expensive.
In order to do this minimization we propose

$$\begin{aligned}\sum_{i=1}^{m} \Bigg| p_{k|k}\left(x_{k|k}^{(i)}\right) - \sum_{j=1}^{m} \alpha_k^{(j)} \frac{1}{\sqrt{(2\pi)^n \det P_{k|k}^{(j)}}} \\ \times \exp\left[-\frac{1}{2}\left(x_{k|k}^{(i)} - x_{k|k}^{(j)}\right)^T \left(P_{k|k}^{(j)}\right)^{-1} \left(x_{k|k}^{(i)} - x_{k|k}^{(j)}\right)\right] \Bigg|^2\end{aligned} \tag{2.6.11}$$

since in L^2 minimization is relaxed by the sum of the collocation distances over $\alpha \in \mathbb{R}^m$, satisfying $\alpha \geq \alpha_0 \geq 0$. The positive constant α_0 is chosen so that the likelihood of each Gaussian distribution is nonzero ($e.g : \alpha_0 = 0.001(1, \cdots, 1)^T \in \mathbb{R}^m$). (2.6.11) is formulated as the constraint quadratic programing

$$\min \frac{1}{2}\alpha^T A^T A\alpha - A^T b + \frac{\delta}{2}|\alpha|^2 \qquad \text{subject to } \alpha \geq \alpha_0 > 0 \tag{2.6.12}$$

where $\delta > 0$ is chosen so that the singularity of the matrix $A^T A$ is avoided and the matrices (A, b) are defined by

$$A_{i,j} = \frac{1}{\sqrt{(2\pi)^n \det P_{k|k}^{(i)}}} \times \exp\left[-\frac{1}{2}\left(x_{k|k}^{(i)} - x_{k|k}^{(j)}\right)^T \left(P_{k|k}^{(j)}\right)^{-1} \left(x_{k|k}^{(i)} - x_{k|k}^{(j)}\right)\right]$$

$$\begin{aligned}b_i = \sum_{i=1}^{m} \frac{1}{(2\pi)^n \sqrt{\det R \det P_{k|k-1}^{(j)}}} \exp\Bigg[-\frac{1}{2}\Bigg(\left(y - h\left(x_{k|k}^{(i)}\right)\right)^T R^{-1} \left(y - h\left(x_{k|k}^{(i)}\right)\right) \\ + \left(x_{k|k}^{(i)} - x_{k|k-1}^{(i)}\right)^T \left(P_{k|k-1}^{(i)}\right)^{-1} \left(x_{k|k}^{(i)} - x_{k|k-1}^{(i)}\right)\Bigg)\Bigg].\end{aligned}$$

Thus, we solve (2.6.12) to obtain $\alpha_k^{(j)}$ at each corrector step by using the existing numerical optimization method.

The theoretical foundation of the Gaussian sum approximation as above is that any probability density function can be approximated as closely as desired by a Gaussian sum [13].

2.7 Lorenz 96 Example

In this section we introduce examples for the discrete optimal filter (2.2.1). First, we consider the Lorenz 96 system [26]

$$\frac{dx_i}{dt} = x_{i-1}\left(-x_{i-2} + x_{i+1}\right) - x_i + F \tag{2.7.1}$$

where $i = 1, 2, \cdots, N$ with periodic boundary conditions i.e., $x_0 = x_N$, $x_{-1} = x_{N-1}$, $x_{N+1} = x_1$. It is a one dimensional atmospheric model, even though its equations are not much like those of the atmosphere but instead models the structure property of the atmosphere. The nonlinear (convective) term conserves the total energy $|x|^2$. It has been used as a benchmark study for system with chaotic-dynamics, filtering problem and data assimilation. With partially noisy observations of the state at p uniformly distributed locations, we estimate the behavior of the state at the other coordinates. Here x_i are values of some atmospheric quantity in N sectors of a latitude circle and F is a forcing constant. The most common model that has chaotic behavior is with $N = 40$ and $F = 8$. The physics of the atmosphere is present only to the extent that there are external forcing and internal dissipation, simulated by the constant and linear terms, while the quadratic terms, simulating advection, together conserve the total energy $(x_1^2 + \cdots + x_N^2)/2$ [26].

To develop the discrete time dynamics (2.1.1) for the Lorenz 96 system we define a proper one step map by applying the fixed point iterates defined by

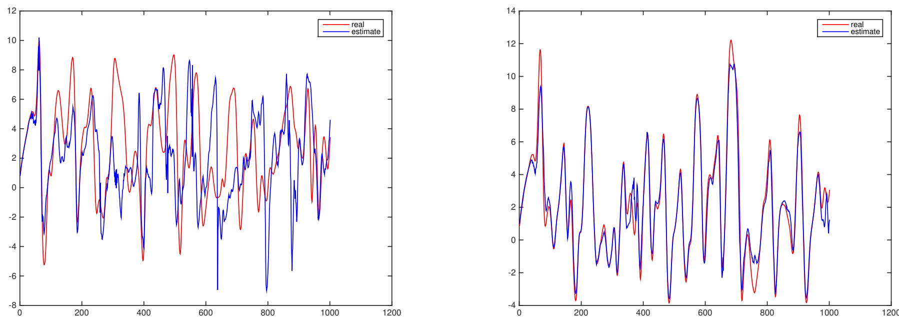
$$x_i^+ = \frac{x_0^n + \Delta t x_{i-1}^n\left(-x_{i-2}^n + x_{i+1}^n\right) + \Delta t F}{1 + \Delta t} \tag{2.7.2}$$

for x_i^{n+1}, where $x_i^{n-1} = x_0^n$, $i = 0$. The resulting discrete dynamics in written in matlab code:

```
for kk=1:5; xx=[x(N-1);x(N);x;x(1)];
x=(x0+dt*F+dt*(xx(i2).*(xx(i4)-xx(i1))))/(1+dt); end
x=x+0.1*sqrt(dt)*randn(N,1); xxx=[xxx x];
y(:,k)=x([1 11 21 31])+sqrt(dt*.01)*randn(4,1); end
```

where and the fixed point iterate (second line) is iterated 5-times and we add noise to x_k and $y_k \in R^4$ with $Q = .01$ and $R = .01$. Then we use the Gaussian filter based on CDF. We set the initial condition $x_0 = \sin(\pi x)$, $N = 40$, and $F = 8$. We test it for $p = 4, 5, 6$ and 8 observations cases. In Figure 2.1 we show the estimate of states at $x_i = 27$ for $p = 4$ and $p = 5$ cases and In Figure 2.2 we show the estimate of states for $p = 6$ and $p = 8$ cases

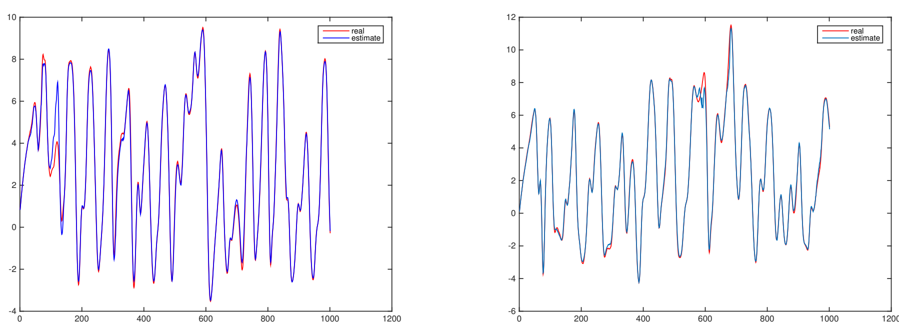
First we observe $p = 4$ is not responsible and does not give a good estimate but $p = 5$ is acceptablle and gives a reasonable estimate. Incerasing in number of observation the Gaussian filter performs excellently.



(a) $x_i = 27$, $p = 4$

(b) $x_i = 27$, $p = 5$

Figure 2.1: Lorenz 96 model, $N = 40$.



(a) $x_i = 27$, $p = 6$

(b) $x_i = 38$, $p = 8$

Figure 2.2: Lorenz 96 model, $N = 40$.

From our tests we conclude that at least 5 observations are needed to make a reasonable estimate of the state.

2.8 Wave equation examples

In this section we discuss a PDE model example to illustrate an application of Gaussian filter. More detailed discussion of PDE models will be given in Section 4.2. We consider the wave equation

$$\frac{\partial^2}{\partial t^2} u = c^2(x)\,\Delta u - q(x)u \text{ in } x \in \Omega \tag{2.8.1}$$

where $u = u(t,x)$ represents the acoustic pressure, elastic deformation and electrical or magnetic field. $c(x)$ is the propagation speed in the medium Ω and $q(x)$ is the potential in the medium. We assume the nominal values of the wave speed $\bar{c}(x)$ and potential $\bar{q}(x)$. Our interests are to determine the state $u = u(t,x)$ from the observation

$$y_k = u(t_k, x_i) + v_k, \quad x_i \in \Omega = [0,1] \tag{2.8.2}$$

for the one-dimensional case. Or, we generate a solution for (2.8.1) with randomly perturbed medium with $c(x) = \bar{c}(x) + "noise"$ and $q(x) = \bar{q}(x) + "noise"$ and it results in a noisy observation y_k. For multiple dimensional case we will have

$$y_k(s) = u(t_k, s), \quad s \in \Gamma = \text{a surface } \Gamma \text{ in } \Omega.$$

First, we develop the first order form of the wave equation and we then develop a stable and accurate discretization in time and space for the one step map. The first order form is

$$\frac{d}{dt}(u,v) = \mathcal{A}(u,v),$$

i.e.,

$$u_t = v, \quad v_t = c^2(x)\,\Delta u.$$

where v is the velocity. We use the second order central difference in space and the Leapfrog finite difference in time:

$$\frac{u_k^{n+1} - 2u_k^n + u_k^{n-1}}{\Delta t^2} = c_k^2\,\frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\Delta x^2} - q_k\,u_k^n \tag{2.8.3}$$

where $\Omega = [0,1]$. $x_k = \frac{k}{N}$ is the uniform grid and u_k^n represents the value of $u(t,x)$ at $(n\,\Delta t, x_k)$ with $\Delta t > 0$ the time step size. It can be proven that if we assume the CFL condition $\Delta t \leq \max(c_k)\Delta x$, then (2.8.3) is stable in the sense of Neumann criterion. Our discrete dynamics in the first order form is written as

$$\begin{aligned}
&\tfrac{v_k^{n+\frac{1}{2}} - v_k^{n-\frac{1}{2}}}{\triangle t} = c_k^2\,\tfrac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\triangle x^2} - q_k u_k^n, \quad u_0^n = 0, \;\; u_N^n = 0 \\
&\tfrac{u_k^{n+1} - u_k^n}{\triangle t} = v_k^{n+\frac{1}{2}}.
\end{aligned}$$

where we assume zero boundary condition $u(t,0) = 0$ and $u(t,1) = 0$. So first we will have to find how to update v, knowing u and the previous value of v. This means that

$$v_k^{n+\frac{1}{2}} = v_k^{n-\frac{1}{2}} + \frac{\triangle t}{\triangle x^2} c_k^2 (u_{k+1}^n - 2u_k^n + u_{k-1}^n) - \triangle t q_k u_k^n.$$

Thus, we obtain the one step map in the form $f(u, v)$ where

$$f(u,v) = \begin{cases} v_k^{n-\frac{1}{2}} + \triangle t \left(c_k^2 \frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\triangle x^2} - q_k u_k^n \right) \\ \\ u_k^n + \triangle t v_k^{n+\frac{1}{2}} + \triangle t^2 \left(c_k^2 \frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\triangle x^2} - q_k u_k^n \right). \end{cases} \tag{2.8.4}$$

This is represented in the matlab file

```
for k=1:100;
tmp=-dt*(1+.05*randn)*u(ip).*(h0*u(1:m));
u(m+[1:m])=u(m+[1:m])+tmp;
u(1:m)=u(1:m)+dt*u(m+[1:m]); xx=[xx u]; end
y=xx([30 80],:); y0=y; y=y0+.01*randn(size(y));
```

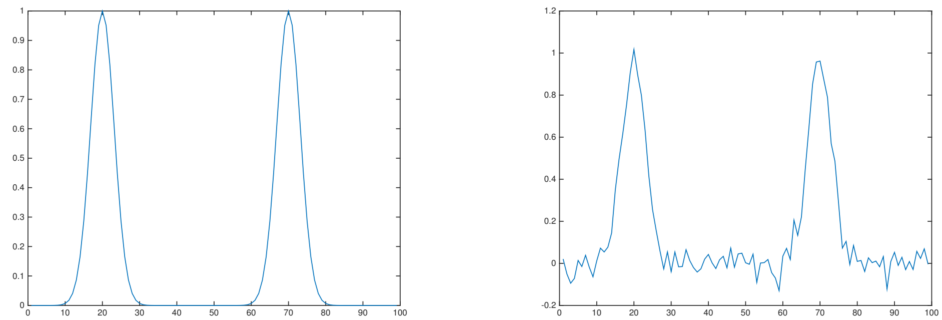
In our first numerical tests we pick $n = 100$, and $m = n - 1$ $\Delta x = .01$ and

$$c_k^2 = .9 + .1 \sin(2\pi x_k)) + .05 randn(1, n-1) \tag{2.8.5}$$

and $q_k = 0$ (no potential term) and $CFL = 1$ (i.e., $\Delta t = \Delta x = .01$). In the third line in the matlab we update the velocity v and in the fourth line we update the state u according to (2.8.4). Here we used two point measurements y_k at $x_1 = .3$ and $x_2 = .8$ in (2.8.2). We tested our algorithm, we set the initial condition

$$u_0(x) = u_0(x) + .05 * randn(x)$$

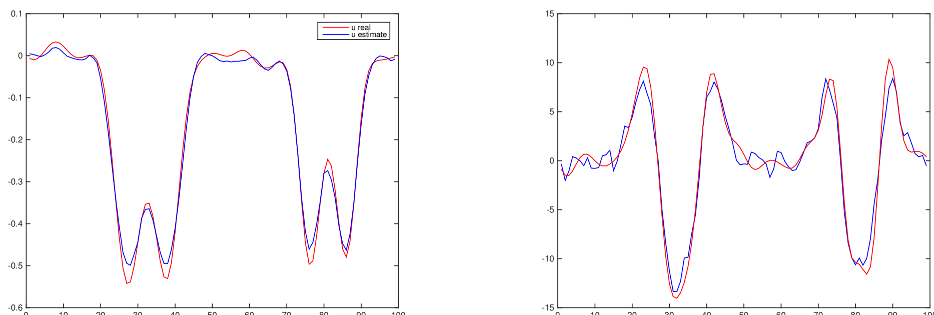
as depicted in Figure 2.3 and test the Gaussian filter (2.5.8)-(2.5.9).



(a) u_0 (b) $\hat{u}_0$ with noise added

Figure 2.3: u_0 vs noisy $\hat{u}_0$

We obtain the estimated $\hat{u}$ and $\hat{v}$ at time $T = 1$ as depicted in Figure 2.4.



(a) displacement $\hat{u}$ at $T = 1$ (b) velocity $\hat{v}$ at $T = 1$

Figure 2.4: estimation of $\hat{u}$ and $\hat{v}$ with Noisy initial u

First the results are shown for no noise in the measurements for Figure 2.3 and Figure 2.4. We also added the regularization step after the predictor and corrector steps as

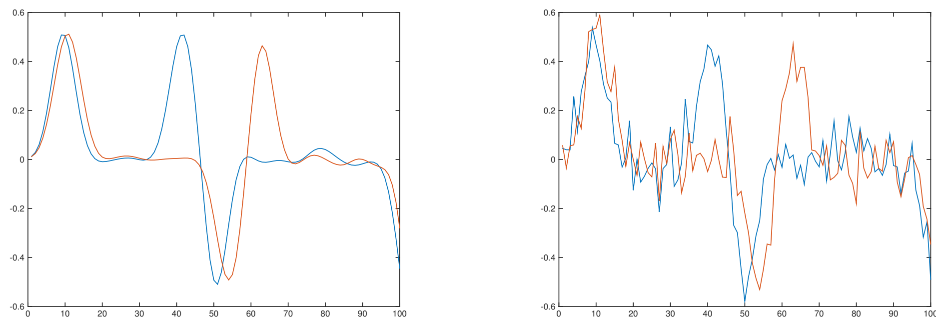
$$u^n \to (I + \beta H)^{-1} u^n$$

$$v^n \to (I + \beta H)^{-1} v^n$$

with $\beta = 1e - 6$ (we need to adjust in general). This regularization step turns out to be very essential to obtain good estimates (u, v). If we use the same method, this time with noisy measurements

$$y(x) = y(x) + .01 * randn(y)$$

which is shown in Figure 2.5,

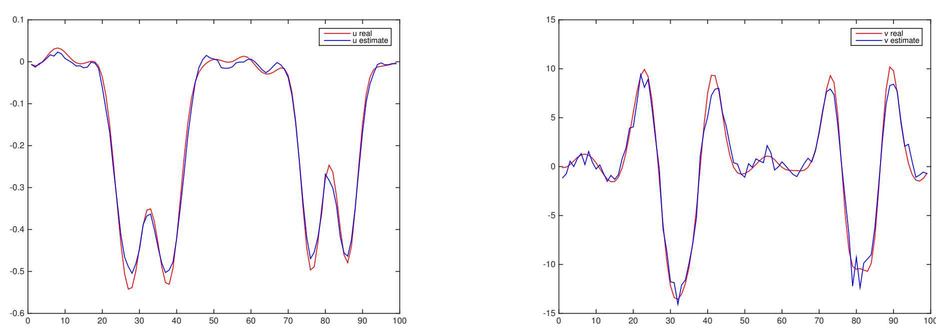


(a) y with no noise

(b) y with noise

Figure 2.5: y with no noise vs y with noise added

then we show the estimated u and v at time $T = 1$ in Figure 2.6.



(a) $\hat{v}$ displacement

(b) $\hat{u}$ velocity

Figure 2.6: estimation of u and v with noisy observations y

Based on our tests we can conclude that the Gaussian filter works very well and thus gives a good estimate of the state given noisy initial conditions or noisy observations.

Chapter 3

Continuous Time Optimal Filter by Zakai Equation

In this chapter we discuss the continuous time stochastic dynamics governed by Itô stochastic differential equations (SDEs) and the optimal filtering problem with continuous time observations. The optimal filter equation for the conditional probability density is governed by the Zakai equation. As will be discussed, the first term of the Zakai equation corresponds to the predictor step while the second term corresponds to the corrector step in comparison to Bayes' updates. Next, we discuss the optimal filter for continuous time with discrete-time observation case then compare and relate it to the time splitting method of the Zakai equation for continuos-time. They have the common predictor step by solving the Fokker-Plank equation. The corrector steps are directly related if we make the correspondence $y_k \sim \frac{y(t_k)-y(t_{k-1})}{\Delta t}$ using the innovation process. That is, the discrete time filter is a good representation of the continuous time filter. Here, of course, we must have a good one step map in (2.1.1) from the continuous time (3.1.1) to the discrete time dynamics. Thus, we use the filtering methods for the resulting discrete time model to obtain a filtering algorithm for the continuous time case.

3.1 Nonlinear stochastic system

We consider a continuous time stochastic process x_t by the Itô's stochastic differential equation, i.e., the continuous-time stochastic process x_t satisfies

$$\begin{cases} dx_t = f(x_t)\,dt + \sigma(x_t)\,dB_t \\ \\ dy_t = h(x_t)\,dt + dW_t \end{cases} \tag{3.1.1}$$

where f and h are nonlinear in general and $f(x_t)$, $\sigma(x_t)$ represents the drift and the intensity of the diffusion, respectively. Also, h is the observation map and y_t is the observation process. Here (B_t, W_t) are Brownian motions. Even if f is linear and $\sigma = \sigma(t)$ is constant, if the initial condition x_0 is not Gaussian then x_t is not a Gaussian process.

The stochastic differential equations (3.1.1) should be understood in the integral form as

$$
\begin{aligned}
x_t &= x_0 + \int_s^t f(x_s)\, ds + \int_0^t \sigma(x_s)\, dB_s \\
y_t &= \int_0^t h(x_s)\, ds + W_t.
\end{aligned}
\tag{3.1.2}
$$

The path-wise unique solution to SDEs is established under assumption (Itô's) condition, i.e., f and σ are Lipschitz and linear growth on σ ($|\sigma(x)| \leq C\,(1+|x|)$) [11]. Also, the probability density function $p = p(t,x)$ of the state process x_t satisfies the Fokker-Plank equation also known as Kolmogorov forward equation

$$
\frac{\partial}{\partial t} p = \mathcal{A}^* p(t,x), \quad p(0,x) = p_0(x), \tag{3.1.3}
$$

where $\mathcal{A}$ is the generator of the Itô diffusion process x_t and $\mathcal{A}$ is defined by

$$
\mathcal{A}\phi = \sum_{j=1}^{n} f_j(x) \frac{\partial \phi}{\partial x_j} + \frac{1}{2} \sum_{i,j=1}^{n} a_{ij}(x) \frac{\partial^2 \phi}{\partial x_i \partial x_j}, \quad \phi \in C^2(R^d) \tag{3.1.4}
$$

where the symmetric positive matrix $a = \sigma\sigma^t$. Thus, by Green's formula the (formal) adjoint $\mathcal{A}^*$ of the generator $\mathcal{A}$ is given by

$$
\mathcal{A}^* p = -\sum_{j=1}^{n} \frac{\partial}{\partial x_j}(b_j(x), p) + \frac{1}{2} \sum_{i,j=1}^{n} \frac{\partial^2}{\partial x_i \partial x_j}(a_{ij}(x) p). \tag{3.1.5}
$$

Next, the unnormalized conditional density p of the process x_t, given σ-field $\mathcal{Y} = \sigma(y_s, s \leq t)$, generated by observation process, i.e.,

$$
\hat{x}_t = E[x_t|\mathcal{Y}_t] = \frac{\int_{R^n} x p(t,x)\, dx}{\int_{R^n} p(t,x)\, dx}
$$

is governed by the Zakai equation [5, 14, 36]

$$
dp(t,x) = \mathcal{A}^* p(t,x)\, dt + R^{-1} h(x)\, dy_t. \tag{3.1.6}
$$

That is, the Zakai equation gives the optimal filter but for unnormalized density, i.e., for any measurable function f we have the formula for conditional expectation of f by

$$\widehat{f(x_t)} = E[f(X_t)|\mathcal{Y}_t] = \frac{\int_{R^n} f(x)p(t,x)\,dx}{\int_{R^n} p(t,x)\,dx}.$$

Also, the normalized density satisfies the Kushner's equation [12, 30]. Note that $p(t,x)$ is a random process and Zakai equation (3.1.6) is a linear stochastic differential equation in $L^2(R^n)$ [19, 30]. The Zakai theory also states that y_t is a Brownian motion on the transformed triple $(\Omega, \mathcal{Y}_t, \tilde{P})$ with Randon-Nikodym derivative

$$\frac{d\tilde{P}}{dP} \sim \exp\left(\int_0^t (\widehat{h(x_s)})dy_s - \frac{1}{2}|(\widehat{h(x_s)})|\,ds\right).$$

3.2 Time-Splitting method and Relation to Discrete-time optimal filter

In this section we first describe the case of continuous time process x_t governed by (3.1.1) but with a discrete time observation

$$y_k = h(x_{t_k}) + w_k.$$

In this case the optimal filter is given as follows. First given $p(t_{k-1}^+)$ we solve the Fokker-Plank equation

$$\frac{\partial p(t,x)}{\partial t} = \mathcal{A}^* p(t,x) \tag{3.2.1}$$

on the interval $[t_{k-1}, t_k]$. Then, we incorporate the new observation y_k at t_k and update the conditional probability density $p(t_k^+)$ by the corrector step:

$$p(t_k^+) \sim \exp\left[-\frac{1}{2}(h(x) - y_k)^T R^{-1}(h(x) - y_k)\right] p(t_k). \tag{3.2.2}$$

Now, we introduce the time-splitting method for the Zakai equation (3.1.6). The first step of the time-splitting method for Zakai equation equals to the Fokker Plank step (3.2.1) and the second step is to solve for $p^+(t_k)$

$$dp^+(t,x) = R^{-1}h(x)p^+(t,x)dy_t, \quad p^+(t_{k-1}) = p(t_k), \tag{3.2.3}$$

where y_t is the continuous time observation on $[t_{k-1}, t_k]$. The time-splitting method splits deterministic part (Fokker-Plank) and stochastic part (as innovation). (3.2.3) is equivalent to

$$p(t_k^+) \sim \exp\left[h(x)\left(\frac{y_{t_k} - y_{t_{k-1}}}{\Delta t} - \frac{h(x)^2}{2}\right)\Delta t\right] p(t_k). \tag{3.2.4}$$

Now, comparing the formulae (3.2.2) (discrete-time observation) and (3.2.3) (continuous-time observation) clearly they are related by the correspondence

$$y_k \sim y_{t_k} - y_{t_{k-1}}.$$

This relation suggests that the continuous time filtering (3.2.3) can be performed by the corresponding discrete time filtering (3.2.2). Recall that the Bayes' formula for the predictor step (2.2.1)

$$p_{k|k-1}(x) = \int_{R^n} \frac{1}{\sqrt{(2\pi)^n \det Q}} \exp\left[-\frac{1}{2}(x - f(t))^T Q^{-1}(x - f(t))\right] p_{k-1|k-1}(t)dt \tag{3.2.5}$$

where f is the one-step map of $\dfrac{dx}{dt} = b(x)$ on $[t_{k-1}, t_k]$. (3.2.5) gives an approximation formula to the Fokker-Plank equation (3.2.1). So, one has a very clear relationship between the optimal filter (3.2.1)-(3.2.2) for continuous time with discrete time observations, the time splitting (3.2.1)-(3.2.3) of the Zakai equation as well as the Bayes' updates (2.2.1)-(2.2.2) for discrete time optimal filter.

Thus, we must have an accurate stable one step map from the continuous time dynamics to the discrete time dynamics. Thus, the different examples that we test need a careful construction of the one step map case-by-case.

3.3 Important aspects of the filtering problems

In this section we state and summarize the related and important aspects of the filtering problems in addition to what we described.

Remarks:

(1) As we stated before the continuous time SDE (2.1.2) can be approximated by

$$x_{t_k} \sim x_{t_{k-1}} + \Delta t_k\, b(x_{t_{k-1}}) + \sigma(x_{t_{k-1}})(B_{t_k} - B_{t_{k-1}}).$$

That is, it reduces to the one-step map (2.1.3) as

$$f(x_{t_{k-1}}) = x_{t_{k-1}} + \Delta t\, b(x_{t_{k-1}}).$$

But, if we have a stiff ODE system $\dfrac{dx}{dt} = b(x)$, we must have a stable one-step map based on an implicit scheme. Especially, time and space discretization for PDE system and like we did for the Lorenz 96 model (2.7.1) with the one step map (2.7.2).

(2) It is very essential to estimate unknown parameters a in models in discrete time model

$$x_k = f(x_k, a_k) + v_k$$

and in continuous time model

$$dx_t = b(x_t, a)\, dt + \sigma(x_t, a) dB_t$$

as well as in PDEs models (e.g., wave equation with speed and potential medium parameter (2.8.1) in Chapter 2, conductivity (3.4.3) in Section 3.4 and 3 by 3 Lorenz model in Chapter 6). We have an augmented model:

$$\begin{aligned} x_k &= f(x_{k-1}, a_k) + v_k \\ a_k &= a_{k-1} + \text{"}noise\text{"} \end{aligned} \tag{3.3.1}$$

in which a is a random coefficient. We also consider the mean return model

$$\begin{aligned} x_k &= f(x_{k-1}, a_k) + v_k \\ a_k &= \gamma\,(\bar{a} - a_{k-1}) + \text{"}noise\text{"} \end{aligned} \tag{3.3.2}$$

where $\gamma > 0$ is a return rate to an equilibrium $\bar{a}$. We apply the proposed filter for the joint estimation the state x_k and parameter a_k in Chapter 6.

(3) As we sated in Section 2.5 where we introduce the CDF one can introduce reducing $d = n$ (full rank) to m by a sampling method, i.e., in the transformation $t = S^T s + \bar{x}$ (2.5.2) with $\Sigma = S^T S$ (square factor) we use the singular value decomposition $\Sigma = U\tilde{\Sigma}U^T$ and use the reduced order transform $t = S_m^T s + \bar{x}$ (4.1.1). But, this reduction in directions works for any linearization of f and h just not only for the CDF.

(4) Smoothing problem is to estimate the initial condition x_0 from discrete time $\{y_k, k \leq T\}$ and solving backward. It can be performed by using time reversal method that will be discussed in depth in Chapter 5.

(5) We will introduce the assumed covariance filter of the form

$$x_{k|k-1} = f(x_{k-1|k-1}), \quad x_{k|k} = x_{k|k-1} + G(y_k - h(x_{k|k-1}))$$

where $G = \Sigma H^T$ is the assumed gain. In Gaussian filter the covariance $\Sigma_{k|k}$ is updated by the covariance updates (2.5.10)–(2.5.13). For example for the assumed covariance filter, we use the so-call dissipative gain $G = \gamma H^T$ (equivalently, we assume $\Sigma = \gamma\, I$ for the time-reversal filter/smoothing (see details in Chapter 5).

3.4 PDEs example

In this section we discuss PDE (partial differential equation) models for the stochastic dynamics. Recall we have introduce a wave equation model in Chapter 2. Now we discuss different PDE models.

First, consider parabolic stochastic system of the form

$$\frac{\partial}{\partial t}u + \nabla\cdot(F(u)) = \mu\,\Delta u + \sigma(x)\frac{dB_t}{dt} \text{ in } x \in \Omega \tag{3.4.1}$$

with boundary condition

$$\alpha\,\frac{\partial u}{\partial \nu} + \beta u = \text{"}noise\text{" at boundary } \partial\Omega$$

This represents stochastic model for conduction, diffusion and advection processes for $u = u(t,x)$, $x \in \Omega$ where Ω is a domain in R^d. It a system for vector filed u and $F(u)$ is a flux vector in general. A wide class of PDE models including model (3.4.1) is described by semi linear dynamics of the form

$$du_t = (Au(t,x)\,dt + F(u_t))\,dt + \sigma(x_t)\,dB_t \tag{3.4.2}$$

where A generates an analytic semi group on $X =$ the state space for u and F is a nonlinear part of the dynamics. Existence of solutions to (3.4.2) has been discussed in literature [32, 33] based on the C_0 semigroup theory. That is, the (mild) solution $u(t)$ satisfies

$$u(t) = S(t)u(0) + \int_0^t S(t-s)F(x(s))\,ds + \int_0^t S(t-s)\sigma\,dB_s.$$

A specific case is the conservative law when the viscosity $\mu = 0$, e.g., a linear diffusion advection equation:

$$\frac{\partial}{\partial t}u + \frac{\partial}{\partial x_1}(a(x_1,x_2))u) + \frac{\partial}{\partial x_2}(b(x_1,x_2)u) = \mu\,\Delta u \tag{3.4.3}$$

where $\mu > 0$ is the viscosity. Also, nonlinear case such as the Burgers' equation (see Section 5.3) and Shallow water model [34]. In general (3.4.2) encompasses models for heat diffusion, meteorology, oceanography, acoustic wave propagation, physical or mathematical systems. And it also contains compressible fluid equations and primitive equations.

Once again as stated in Section 3.2, it is very essential to develop an accurate and stable discretization in time and space so that we have the corresponding discrete time dynamics with an accurate one step map (2.1.1). In this manner we develop the filtering method for PDE models, by the corresponding discrete time dynamics (2.1.1).

3.5 Gaussian filter for continuous-time case

In this section we derive the Gaussian filter for the continuous time case as (3.2.1)–(3.2.2). It is shown in Section 3.1 that the unnormalized conditional probability function $p(t,x) \in L^2(R^d)$. The nonlinear filtering theory [21] is summarized as The best estimate is given by the conditional probability density

$$\hat{x}(t) = E[x(t)|\mathcal{Y}_t] = \int_{\mathbb{R}^n} x\pi(t,x)\,dx$$

where $\pi(t,x)$ is the conditional density of $x(t)$ given $\mathcal{Y}_t$ and the optimal filtering equation is given by

$$d\hat{x}(t) = \widehat{f(x(t))}\,dt + G(t)(dy_t - \widehat{h(x(t))})$$

where

$$E[\phi(x)|\mathcal{Y}_t] = \int_{\mathbb{R}^n} \phi(x)\pi(t,x)\,dx$$

and the filter gain is defined by

$$G(t) = x(t)\widehat{h(x(t))}^T - \widehat{x(t)}\widehat{h(x(t))}^T.$$

In the case $h(x) = Hx$ we have

$$G(t) = E[(x(t) - \hat{x}(t))(x(t) - \hat{x}(t))^t]H^t = \Sigma(t)H^t.$$

The covariance Σ is updated by

$$\frac{d\Sigma}{dt} = F(\hat{x}, \Sigma(t)) - G(t)G(t)^T + Q = 0.$$

where $GG^T = \Sigma H^t R^{-1} H\Sigma(t)$ and F is

$$F(t,\hat{x},\Sigma) = \int (f(x) - \widehat{f(x(t))})(x - \hat{x}(t))^T N(t,x)\,dx + \int (x - \hat{x}(t))^T (f(x) - \widehat{f(x(t))})^T N(t,x)\,dx.$$

For example, (bi-linear case)

$$f(x,a) = ax$$

thus we have

$$\widehat{f} = \hat{x}\hat{x}E[\hat{a}(x-\hat{x}) + \hat{x}(a-\hat{a}) + (x-\hat{x})(a-\hat{a})]N(t,x)\,dx$$

$$= \hat{a}\hat{x} + \Sigma_{\hat{x},\hat{a}}$$

and

$$F = 2E[\hat{a}(x-\hat{x}) + \hat{x}(a-\hat{a}) + (x-\hat{x})(a-\hat{a})(x-\hat{x})]N(t,x)\,dx$$

$$= 2\left(\hat{a}\Sigma_{\hat{x},\hat{a}} + \hat{x}\Sigma_{\hat{x},\hat{a}}\right).$$

We need to evaluate

$$\widehat{f} = \int f(x)dN(t,\hat{x},\Sigma)$$

$$= f(\hat{x}) + \int (f(x) - f(\hat{x}))N(t,\hat{x},\Sigma)\,dx$$

$$= f(\hat{x}) + \int (f'(\hat{x}(x-\hat{x}) + \tfrac{1}{2}f''(\hat{x})(x-\hat{x})^2 + h.o.d.)N(t,\hat{x},\Sigma)\,dx$$

$$\sim f(\hat{x}) + \tfrac{1}{2}f''(\hat{x})\Sigma$$

and

$$\int (f(x) - \widehat{f(x)})(x-\hat{x})^t N(t,\hat{x},\Sigma)\,dx$$

$$\sim \int (f'(\hat{x}(x-\hat{x}) + \tfrac{1}{2}f''(\hat{x})(x-\hat{x})^2)(x-\hat{x})^t N(x,\hat{x},\Sigma)\,dx = f'(\hat{x})\Sigma.$$

Chapter 4

Reduced Order Method

As we have noted in Chapter 2, if the dimension $n = d$ of a stochastic system is large, the implementation of the Gaussian filter is too time consuming, requires too large of a storage and thus may not have real time applications. Problem arises due to a large number of sampling directions (full rank) and the corresponding covariance updates (2.5.10)–(2.5.13) is too expensive. In this chapter we discuss how to remove these difficulties and actually improve the performance of our filter by using the reduced order filter approach. The reduced order approach consists of first compute the singular value decomposition of a nominal matrix, reduce its rank by choosing the dominant singular values and use the coordinate change $t = S_m^T s + \bar{x}$ (4.1.1) see remark (4) in Section 3.3. That is, we reduce S to S_m based on the dominant singular values. Thus, in the reduced filter we don't store the covariance matrix Σ but update the square root factors. Because we have reduced the unnecessary and redundant part by the reduced order filter, it should improve the performance of our filter. We will demonstrate how good this reduced order filter is by testing it on the Lorenz 96 and the wave equations models.

4.1 Reduced order filter

Recall the transformation step (2.5.2) for the Gaussian filter:

$$t = S^T s + \bar{x} < \quad S = U^T \tilde{\Sigma}^{\frac{1}{2}}$$

with singular value decomposition $\Sigma = U\tilde{\Sigma}U^T$. We use the reduced order transformation of S_m that corresponds to the m dominant singular values of Σ, i.e. $S_m = \tilde{\Sigma}_m^{\frac{1}{2}} U_m^T$. The selection of m depends on the speed, storage and performance requirements as we discuss using the concrete examples we tested. It results in the coordinate transform

$$t = U_m \Sigma_m^{\frac{1}{2}} s + \hat{x}_{k-1|k-1} \tag{4.1.1}$$

in the predictor step of the Gaussian filter. Thus, we only need to use $m << d$ directions for the central difference operations, i.e., evaluate

$$\begin{aligned} a_i &= \frac{\tilde{F}(he_i) - \tilde{F}(-he_i)}{2h} \quad 1 \le i \le m \\ H_{ii} &= \frac{\tilde{F}(he_i) - 2\tilde{F}(0) + \tilde{F}(-he_i)}{h^2} \quad 1 \le i \le m \end{aligned} \tag{4.1.2}$$

Now, the predictor step (reduced) becomes

$$\hat{x}_{k|k-1} = \hat{x}_{k-1|k-1} + \frac{1}{2}\sum_{i=1}^{m} H_{ii} \tag{4.1.3}$$

and we have the square root form of the covariance (we don't form $\Sigma_{k|k-1}$)

$$\Sigma_{k|k-1} = Q_k + \sum_{i=1}^{m} a_i a_i^T + \frac{1}{2}\sum_{i=1}^{m} H_{ii} H_{ii}^T. \tag{4.1.4}$$

Next the corrector step becomes (for simplicity for the linear observation we assume that $h(x) = Hx$).

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + G_k(y_k - H_k \hat{x}_{k|k-1}) \tag{4.1.5}$$

where the gain G_k is computed as follows. Let's recall the gain formula:

$$G_k = \Sigma_{k|k-1} H_k^T (R + H_k \Sigma_{k|k-1} H_k^T)^{-1}.$$

First, we evaluate

$$\Sigma_{k|k-1} H_k^T = Q_k H_k + \sum_{i=1}^{m} a_i (H_k a_i)^T + \frac{1}{2}\sum_{i=1}^{m} H_{ii}(H_{ii} H_k)^T.$$

Then, we can compute the gain using the gain formula (2.3.8). For the purpose of reducing the complexity of the filter in terms of the covariance update, there is the option of dropping the H_{ii} part which is the 2nd order variation term. The most technical part is how to update $\Sigma_{k|k}$ without ever forming it. We use the Householder transformation of $A = [a_k, b_j]$ to obtain $PA = [R_1, R_2]$, where R_1, R_2 correspond to columns of a_k and columns of b_j. Thus, we have

$$P\left(\sum_{k=1}^{m} a_k a_k^T - \sum_{j=1}^{p} b_j \tilde{R} b_j^T\right) P^T = \begin{bmatrix} R_1 R_1^T - R_2 \tilde{R} R_2^T & 0 \\ & \\ 0 & 0 \end{bmatrix}$$

where

$$b_j = S_j = \left(Q - \sum_{k}^{p} a_k a_k^T \right) H_{j,:}^T$$

and

$$\tilde{R} = R + H_k \Sigma_{k|k-1} H_k^T.$$

Assuming that $Q = \alpha I$ we can take the singular value decomposition of

$$\alpha I_{m+p} + R_1 R_1^T - R_2 \tilde{R} R_2^T.$$

Thus the Householder transform helps us to keep the square root form of the covariance update. After this step, we can now proceed to the next time step for the predictor and corrector steps.

In summary to perform the reduced order Gaussian filter, we first initialize the SVD decomposition as $P_{0|0} = U\tilde{\Sigma}U^T$.

Predictor step:

- From the decomposition $U\Sigma U^T$ reduce the rank by $U_m\tilde{\Sigma}_m U_m^T$.
- Define $S_m = \tilde{\Sigma}_m^{\frac{1}{2}} U_m^T$.
- Change Coordinate with $t = S_m^T s + \hat{x}_{k-1|k-1}$.
- Set $\tilde{F}(s) = f(S_m^T s + \hat{x}_{k-1|k-1})$.
- Compute CDF components (4.1.2):

$$a_i = \frac{\tilde{F}(he_i) - \tilde{F}(-he_i)}{2h} \qquad 1 \le i \le m,$$

$$H_{ii} = \frac{\tilde{F}(he_i) - 2\tilde{F}(0) + \tilde{F}(-he_i)}{h^2} \qquad 1 \le i \le m.$$

- Update the mean and covariance by

$$\hat{x}_{k|k-1} = x_{k-1|k-1} + \frac{1}{2}\sum_{i=1}^{m} H_{ii}$$

$$\Sigma_{k|k-1} = Q_k + \sum_{i=1}^{m} a_i a_i^T + \frac{1}{2}\sum_{i=1}^{m} H_{ii} H_{ii}^T.$$

Corrector step:

- Compute the gain G_k by

$$S_k = QH^T + \sum_{i=1}^{m} a_i (Ha_i)^T$$

and

$$G_k = S_k (HS_k + R)^{-1} S_k^T.$$

- Thus the mean is updated by

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + G_k (y_k - H_k \hat{x}_{k|k-1}).$$

Householder step:

- Use the Householder transforms for A and obtain

$$P \left(\sum_{k=1}^{m} a_k a_k^T - \sum_{j=1}^{p} b_j \tilde{R} b_j^T \right) P^T = \begin{bmatrix} R_1 R_1^T - R_2 \tilde{R} R_2^T & 0 \\ & \\ 0 & 0 \end{bmatrix}.$$

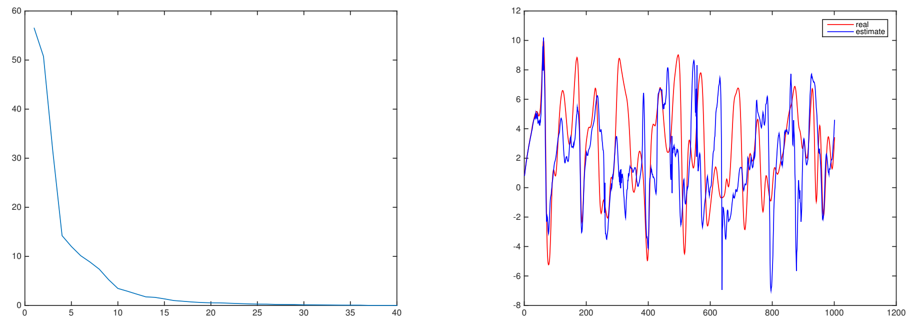
- Proceed to the prediction step without having to form $\Sigma_{k|k}$.

In summary, the reduced order filter uses the same one step map and the same observation, the initial estimate $\hat{u}_0$ but only difference is that we use a much reduced directional sampling (4.1.2) and update the square factor without forming the covariance matrices. Thus, for the reduced order filter, the complexity is very much reduced so that real time application may be possible. As shown in our examples (Sections 4.2, 4.3 and 4.4) the reduced order filter actually outperforms the full rank filter, which shows the effectiveness of the reduced ordered filter.

4.2 Reduced order for Lorenz 96

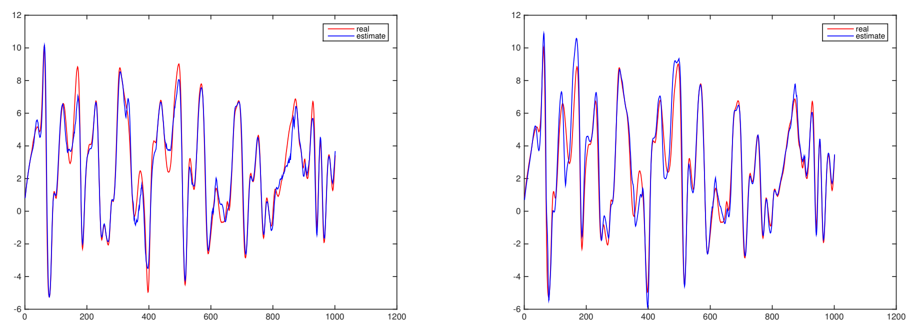
We tested the Gaussian filter for the Lorenz 96 model in Section 2.7. Now we test the reduced order Gaussian filter for the Lorenz 96 , i.e., we reduce $N = 40$ to a reduced rank m as in Section 4.1.

Specifically, we examine $p = 4$ observations cases as in Section 2.7, Figure 2.1 (a). Figures 4.1-4.2 show the results with different reduced order $m = 10,\ 15\ 20$. According to singular value plot Figure 4.1 (a) we tested $m = 10$ (starting) and $m = 15$ (middle) and $m = 20$ (maximum order).



(a) S, $p = 4$ (b) $x_i = 27$, $m = 10$

Figure 4.1: reduced order with Lorenz 96 $p = 4$



(a) S, $m = 15$ (b) $x_i = 27$, $m = 20$

Figure 4.2: reduced order with Lorenz 96 $p = 4$

The following is matlab we used to implement the reduced order filter (number of observation $p = 4$ and number of sampling direction $m = 15$).

```
MM=M;
for kk=1:nmax;
[U S  V]=svd(P+1e-7*eye(N)); S=U(:,1:m)*sqrt(S(1:m,1:m));
x=S*yy+M*ones(1,2*m+1);
for k=1:2*m+1; ff=x(:,k); x0=ff;
for ii=1:5; xx=[ff(N-1);ff(N);ff;ff(1)];
```

```
ff=(x0+dt*F+dt*(xx(i2).*(xx(i4)-xx(i1))))/(1+dt); end
f(:,k)=ff; end;
for k=1:m;
g1(:,k)=f(:,2*k+1)-f(:,2*k); g2(:,k)=f(:,2*k+1)-2*f(:,1)+f(:,2*k); end
g1=g1/2/h; g2=g2/h/h;
M=f(:,1); for k=1:m; M=M+.5*g2(:,k); end
P=c*speye(N);
for k=1:m; P=P+g1(:,k)*g1(:,k)'+.5*g2(:,k)*g2(:,k)'; end
Pxz=P*H'; Pzz=R+H*Pxz; L=Pxz/Pzz; z=H*M;
M=M+L*(y(:,kk)-z); P=P-L*Pxz'; MM=[MM M]; end
```

In summary, we conclude that the reduced order filter perform very well for our tested example, i.e. one can obtain a reasonable estimate with $p = 4$, number of observations for $N = 40$ (dimension of system) with reduced order filter with $m = 15$. Recall the full rank filter was reasonable for $p = 5$ observations but not for $p = 4$ observations. Thus, our claim of the reduced filter is demonstrated (i.e, reduce complexity but yet a better performance).

4.3 Reduced order for the wave speed

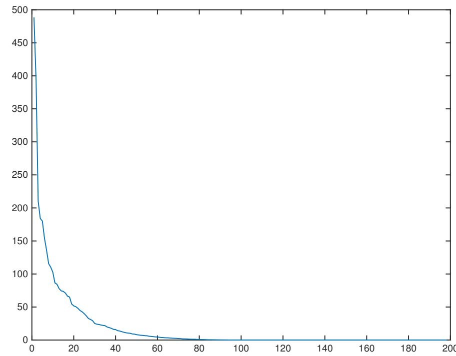
We test the reduced order filter in section 2.8 for the wave equation (2.8.1). For the first case we let

$$c_k^2 = .9 + .1\sin(2\pi x_k)) + .05randn(1, n-1)$$

as (2.8.5) and $q_k = 0$, i.e., the one-step map of our discrete time model is

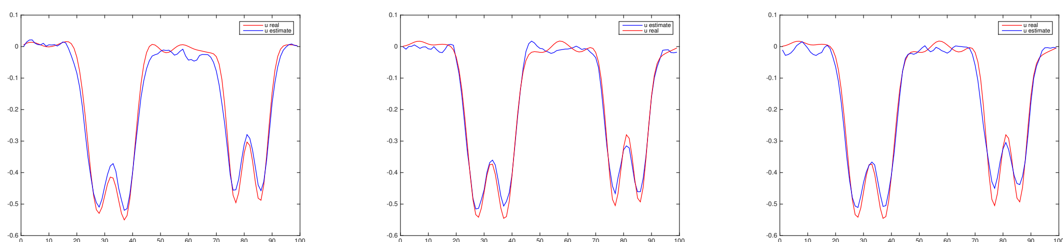
$$f(u,v) = \begin{cases} v_k^{n-\frac{1}{2}} + \triangle t\left(c_k^2 \frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\triangle x^2}\right) \\ \\ u_k^n + \triangle t v_k^{n+\frac{1}{2}} + \triangle t^2\left(c_k^2 \frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\triangle x^2}\right). \end{cases} \tag{4.3.1}$$

All set ups are same and the only difference in order m as in Section 2.8. Figure 4.3 shows the singular values. Around $m = 20$ the singular value curve has a lagest curvature. Our test is conduced with $m = 10, 15$ and 20 for the directional sampling (4.1.1).We show our estimate for the displacement in Figure 4.4, and the estimate for the velocity $hatv$ in Figure 4.5. The results are nearly equal performance comparing to the full rank case ($N = 198$) depicted in Figure 2.4, but m is 5–10% of N.



(a) S, $N = 198$

Figure 4.3: Singular values $N = 198$

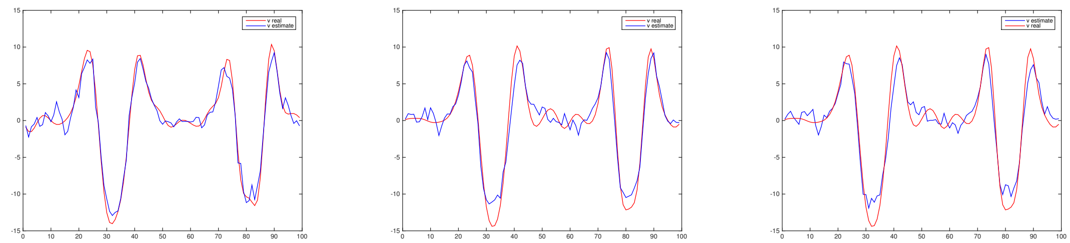


(a) $m = 10$, $\hat{u}$, $T = 1$

(b) $m = 15$, $\hat{u}$, $T = 1$

(c) $m = 20$, $\hat{u}$, $T = 1$

Figure 4.4: reduced order for wave equation u estimate



(a) $m = 10$, $\hat{v}$, $T = 1$ (b) $m = 15$, $\hat{v}$, $T = 1$ (c) $m = 20$, $\hat{v}$, $T = 1$

Figure 4.5: reduced order for wave equation v estimate

We can conclude that using the reduced order filter with $m = 20$, gives us a good estimate of the velocity. The matlab code below concludes this.

```
u=zeros(N,1); u(1:m)=u0+.01*randn(m,1); q=20;
P=1.e-5*eye(N); W=[];
for kk=1:100; P=P+1.e-5*eye(N);
[U S V]=svd(P); S=U(:,1:q)*sqrt(S(1:q,1:q));
J=zeros(N); J(1:m,m+[1:m])=speye(m);
J(m+[1:m],1:m)=-spdiags(c0,0,m,m)*h0;
J(1:m,:)=J(1:m,:)+dt*J(m+[1:m],:); JJ=speye(N)+dt*J; g1=JJ*S;
tmp=-dt*c0.*(h0*u(1:m));u(m+[1:m])=u(m+[1:m])+tmp;
u(1:m)=u(1:m)+dt*u(m+[1:m]);
P=0*speye(N); P=P+g1*g1';
Pxz=P*C'; Pzz=R+C*Pxz; L=Pxz/Pzz; z=C*u;
u=u+L*(y(:,kk)-z); u(1:m)=(speye(m)+al*hh)\u(1:m);
u(m+[1:m])=(speye(m)+al*hh)\u(m+[1:m]);
P=P-L*Pxz'; W=[W u]; end
```

In the first line, q represent the number m of sampling directions we are going to reduce from. This means we are going from $N = 198$ sampling directions to $m = 15, 20$ and 25. In the third line we calculate $S_m = \tilde{\Sigma}_m^{\frac{1}{2}} U_m^T$ that is obtained by reducing the rank by $U_m \tilde{\Sigma}_m U_m^T$ using singular value decomposition. In the 12th and 13th lines we have the regularization of the displacement and velocity.

In summary, we can conclude that the reduced order filter with $m = 20$ gives us a good enough estimate of the state $\hat{u}$ and the velocity $\hat{v}$ for the wave equation (2.8.1) as shown in Section 4.3.

4.4 Reduced order for the potential

In this section, we test the reduced filter for the state estimate (u, v) in wave equation (2.8.1) but with $c_k^2 = 1$ and random potential and

$$q_k = .5(1 + .5\sin(2\pi x_k)) + .05randn(1, n-1).$$

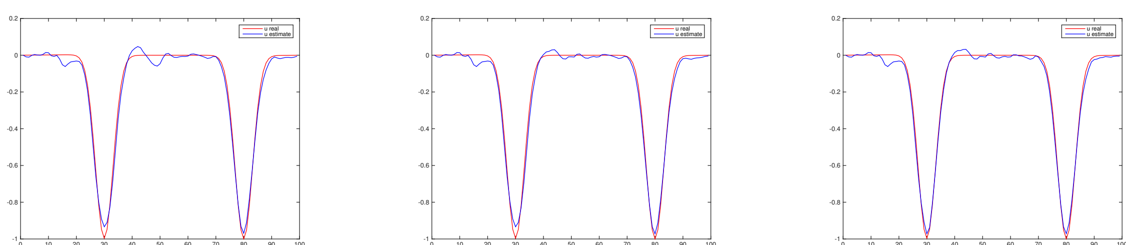
$$\frac{\partial^2}{\partial t^2}u = \Delta u - q(x)u. \text{ in } x \in \Omega.$$

Thus, our discrete time model (4.3.1) is

$$\begin{cases} v_k^{n+\frac{1}{2}} = v_k^{n-\frac{1}{2}} + \triangle t \left(\frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\triangle x^2} - q_k u_k^n \right) \\ u^{n+1} = u_k^n + \triangle t v_k^{n+\frac{1}{2}} + \triangle t^2 \left(\frac{u_{k+1}^n - 2u_k^n + u_{k-1}^n}{\triangle x^2} - q_k u_k^n \right). \end{cases} \tag{4.4.1}$$

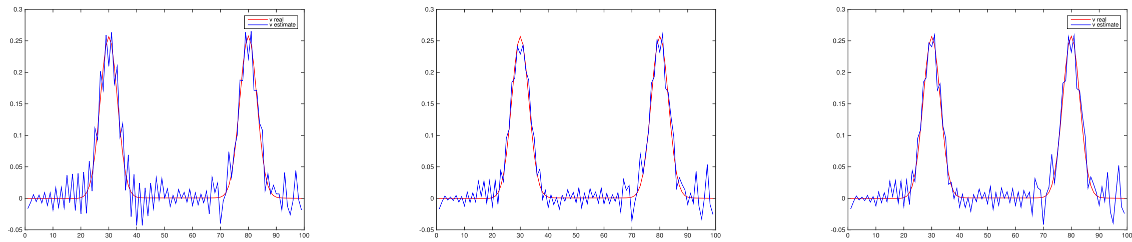
All set ups are same except the new model with c_k and q_k as above and We test the reduced order filter with reduced order direction m as in Section 4.3.

Our test is conduced with $m = 60$ for the directional sampling (4.1.1). In Figure 4.6 we show our estimate for the displacement, and in Figure 4.9 we show our estimate for the velocity. The result are a better performance comparing to full rank case ($N = 198$) depicted in Figure 4.7, here m is 30% of N.



(a) $m = 50$, $\hat{u}$, $T = 1$ (b) $m = 60$, $\hat{u}$, $T = 1$ (c) $m = 70$, $\hat{u}$, $T = 1$

Figure 4.6: reduced order for potential equation u estimate



(a) $m = 50$, $\hat{v}$, $T = 1$ (b) $m = 60$, $\hat{v}$, $T = 1$ (c) $m = 70$, $\hat{v}$, $T = 1$

Figure 4.7: reduced order for potential equation v estimate

We can conclude that using the reduced order filter with $m = 60$, gives us a good estimate of the velocity and displacement. The matlab code below concludes this.

```
u=zeros(N,1); u(1:m)=u0+.01*randn(m,1); q=60;
P=1.e-5*eye(N);  W=[];
for kk=1:100; P=P+1.e-5*eye(N);
[U S V]=svd(P); S=U(:,1:q)*sqrt(S(1:q,1:q));
J=zeros(N); J(1:m,m+[1:m])=speye(m);
J(m+[1:m],1:m)=-h0-spdiags(q2,0,m,m);;
J(1:m,:)=J(1:m,:)+dt*J(m+[1:m],:); JJ=speye(N)+dt*J; g1=JJ*S;
tmp=-dt*(h0*u(1:m)+ q2.*u(1:m));   u(m+[1:m])=u(m+[1:m])+tmp;
u(1:m)=u(1:m)+dt*u(m+[1:m]);
P=0*speye(N); P=P+g1*g1';
Pxz=P*C'; Pzz=R+C*Pxz; L=Pxz/Pzz; z=C*u;
u=u+L*(y(:,kk)-z); u(1:m)=(speye(m)+al*hh)\u(1:m);
u(m+[1:m])=(speye(m)+al*hh)\u(m+[1:m]);
 P=P-L*Pxz'; W=[W u]; end
```

The matlab illustrates that in the case of the potential (4.4.1), the reduced order filter is implemented in the same manner than the one for the wave speed (4.3.1), to reduce from $N = 198$ sampling directions to $m = 50, 60$ and 70. In the third line, once again we calculate $S_m = \tilde{\Sigma}_m^{\frac{1}{2}} U_m^T$ that is obtained by reducing the rank by $U_m \tilde{\Sigma}_m U_m^T$ using singular value decomposition.

In summary, we can conclude that the reduced order filter with $m = 60$ gives us a good estimate of the state u and the velocity v for the wave equation (2.8.1) as shown in Section 4.4.

4.5 Conductivity equation

The other examples of applications of the reduced order filter is the diffusion and advection equation. Let $u(t,x)$ be the temperature on the domain Ω. Consider the heat conduction equation:

$$u_t = \nabla \cdot (\sigma(x)\nabla u) + \nabla(\vec{b}(x)u) - c(x)u + \text{"}noise\text{" in } \Omega, \tag{4.5.1}$$

where $\sigma(x)$ is the conductivity and $\vec{b}$ is a current (field) and $c(x)$ absorption. For example here, "noise" may represent the statistical fluctuation of media, the uncertainty of media and an unmodeled components of the conduction process. Suppose that the observations are boundary measurements, i.e.

$$y(t) = u(t,x), x \in \Gamma_1 \supset \partial\Omega = \text{ boundary of } \Omega$$

where Γ_1 is a subset of the boundary $\partial\Omega$. We can also have observations as distributional measurement:

$$y_k(t) = \frac{1}{vol(\Omega_k)} \int_{\Omega_k} u(t,x)\, dx.$$

for given sub-domains Ω_k of Ω. We must specify the boundary condition for (4.5.1), for example

$$\sigma \tfrac{\partial u}{\partial \nu} + \alpha\, u = g \text{ at } \Gamma_1$$

$$u = 0 \text{ on } \Gamma_2 \supset \partial\Omega \setminus \Gamma_1.$$

Our objective is as follows:

(1) Filtering; where we need to estimate the state $u(t,x)$ from observations (all coefficients $\sigma(x)$, $\vec{b}(x)$ and $c(x)$ are known or perturbed randomly as we did for the wave equation in Section 2.8).

(2) Smoothing; where we estimate the initial condition u_0.

(3) Parameter estimation of all the coefficients, knowing the initial condition.

To do so, we need to describe how to construct a proper one step map in time and space of (4.5.1), i.e., a time and space discretization of equation (4.5.1). Our construction of the one step map is based on the time splitting method; First for the compute the intermediate $u^{n+1/2}$ by the advection equation:

$$u_t + \nabla(\vec{b}(x)u) = 0 \text{ on } [t_n, t_{n+1}] \tag{4.5.2}$$

and then

$$u^{n+1} = (I - \Delta t A_0)^{-1} u^{n+\frac{1}{2}}$$

where the linear operator A_0 is defined by

$$A_0 u = \nabla \cdot (\sigma(x)\nabla u) - c(x)u.$$

We use the central difference method for elliptic operator A_0 as

$$\begin{aligned} A_0^h u = &\frac{1}{h}\left(\sigma_{i+\frac{1}{2},j}\frac{u_{i+1,j}-u_{i,j}}{h} - \sigma_{i-\frac{1}{2},j}\frac{u_{i,j}-u_{i-1,j}}{h}\right) \\ &+\frac{1}{h}\left(\sigma_{i,j+\frac{1}{2}}\frac{u_{i,j+1}-u_{i,j}}{h} - \sigma_{i,j-\frac{1}{2}}\frac{u_{i,j}-u_{i,j-1}}{h}\right) - c_{i,j}u_{i,j}. \end{aligned}$$

where $\{u_{i,j}\}$ atat node $x_{i,j} = (i\,h, j\,h)$ with uniform step size $h > 0$. Also, see the details in Quasi-reversal section.

Chapter 5

Assumed Covariance and Time Reversal Filter

In Chapter 4, we developed the reduced order method of the Gaussian filter. Despite the reduction, it may still be expensive to apply the reduced order Gaussian filter directly to large scale problems. For example in the case of large scale PDE models like the Navier-Stokes, the three dimensional wave propagation, we may not be able to use the reduced order Gaussian filter algorithm directly because it is too expensive. In this Chapter we discuss an alternative to the reduced order Gaussian filter to a class of dissipative systems based on assumed covariance filter for filtering and smoothing, especially the time reversal method. The forward-backward filtering/smoothing algorithm is developed for time reversible dynamics. For the mildly diffusion case the filtering/smoothing is developed based on the quasi-reversible method. We test the assumed covariance and the time reversal filter and smoothing for the Burgers' equations and the diffusion advection equation.

5.1 The assumed covariance and dissipative filter

In this section, we discuss the filtering estimate $\hat{x}(t)$ for (3.1.1) from observations $\{y(s), s \leq t\}$. To this end, we consider the filter dynamics of the form:

$$d\hat{x}(t) = f(\hat{x}(t))dt - G(dy(t) - H\hat{x}(t)\, dt). \tag{5.1.1}$$

For example we select Σ and let $G = \Sigma H^T R^{-1}$, where we assume the following property:

$$\big(f(x_1) - f(x_2), \Sigma^{-1}(x_1 - x_2)\big) \leq \omega\, |x_1 - x_2|^2, \tag{5.1.2}$$

with some $\omega \le 0$. If we have the observation:

$$y(t) = Hx(t) + \text{"}noise\text{"}. \tag{5.1.3}$$

then, we obtain the Lyapunov estimate

$$\begin{aligned}\tfrac{1}{2}\tfrac{d}{dt}\left(x-\hat{x}, \Sigma^{-1}(x-\hat{x})\right) &= \left(f(x)-f(\hat{x}), \Sigma^{-1}(x-\hat{x})\right) \\ &\quad - \left(R^{-1}H(x-\hat{x}), H(x-\hat{x})\right) + \left(\text{"}noise\text{"}, \Sigma^{-1}(x-\hat{x})\right).\end{aligned} \tag{5.1.4}$$

Integration of this estimate in time t yields

$$\begin{aligned}&\frac{1}{2}\left(x(t)-\hat{x}(t), \Sigma^{-1}(x(t)-\hat{x}(t))\right) + \int_0^t -\omega\,|x-\hat{x}|^2 + |H(x(t)-\hat{x}(t))|^2_{R^{-1}} \\ &\quad \le \frac{1}{2}\left(x(0)-\hat{x}(0), \Sigma^{-1}(x(0)-\hat{x}(0))\right) + \int_0^t \left(\text{"}noise\text{"}, \Sigma^{-1}(x-\hat{x})\right)\,dt.\end{aligned} \tag{5.1.5}$$

The first term of the right hand side represents the weighted error of $\hat{x}(t)$ and the second term is the integration dissipation induced by the observations due to the assumed gain $G = \Sigma H^T R^{-1}$ ($\omega \le 0$). The left hand side contains the initial error and also depends on noises in the system and observations.

In the low dimension case we use the Riccati equation to determine Σ:

$$A\Sigma + \Sigma A^T - \Sigma H R^{-1} H^T \Sigma + Q = 0 \tag{5.1.6}$$

where $A = J_f(\bar{x})$ which is the Jacobian of f at a nominal state $\bar{x}$. In fact, Σ is pre-determined and we don't update for filtering (5.1.1). For the linear system (continuous time) $G = \Sigma H^T R^{-1}$ is the stationary Kalman filter gain. Equivalently for (5.1.1) we have

$$\frac{d}{dt}x(t) = f(t, x(t)) + \text{"}noise\text{"}, \tag{5.1.7}$$

which has the dissipative property (5.1.2). In particular if $\Sigma = I$, .i.e, $G = H^T R^{-1}$ if we assume the dissipativity:

$$(f(x_1) - f(x_2), x_1 - x_2) \le \omega\,|x_1 - x_2|^2$$

for ($\omega \le 0$), we obtain the Lyapunov estimate

$$\frac{1}{2}\frac{d}{dt}|x-\hat{x}|^2 \le \omega\,|x-\hat{x}|^2 - \left(R^{-1}H(x-\hat{x}), H(x-\hat{x})\right) + (noise, (x-\hat{x}))$$

and the corresponding error estimate is

$$\frac{1}{2}|x(t)-\hat{x}(t)|^2+\int_0^t \big(-\omega\,|x-\hat{x}|^2+R^{-1}H(x(t)-\hat{x}(t)),H(x(t)-\hat{x}(t))\big)\;dt$$

$$=\frac{1}{2}|(x(0)-\hat{x}(0)|^2+\int_0^t (\text{"}noise\text{"},(x-\hat{x}))\;dt.$$

5.1.1 Examples of dissipative system

In the case of the Lorenz-96 system (2.7.1)

$$\frac{d}{dt}x_i=x_{i-1}\left(-x_{i-2}+x_{i+1}\right)-x_i+F$$

in which

$$f(x)=x_{i-1}\left(-x_{i-2}+x_{i+1}\right)-x_i+F$$

and $f(x)$ satisfies the dissipative property (5.1.2) with $\Sigma=I$, with periodic boundary conditions with $x_0=x_N$, $x_{-1}=x_{N-1}$, $x_{N+1}=x_1$.

Next, as we discussed in Section 2.8 the wave equation (2.8.1)

$$\frac{\partial^2}{\partial t^2}u=c^2(x)\,\Delta u-q(x)\,u \text{ in } x\in\Omega$$

has the first order form

$$\mathcal{A}(u,v)=\begin{pmatrix} v \\ \\ c(x)^2\,\Delta u-q(x)\,u \end{pmatrix}$$

for the state (u,v). If we define the norm $(u,v)\in X=H^1(\Omega)\times L^2(\Omega)$ by

$$|(u,v)|_X^2=\int_\Omega\left(\frac{1}{c^2}|v(x)|^2+|\nabla u(x)|^2\right)\,dx$$

Then, it can be shown that

$$\langle\mathcal{A}(u,v),(u,v)\rangle_X=0 \quad \text{ for } (u,v)\in X$$

where for simplicity we assume $q(x)=0$. Thus, the system is conservative.

Other examples are the diffusion-advection equation (4.5.1). In general we consider the diffusion-advection equation of the form

$$\frac{d}{dt}u+A_1u=-\kappa\,A_0u \tag{5.1.8}$$

where A_0 is a non-negative self-adjoint operator on a Hilbert space and $(A_1\, u, u) \geq 0$ as in Section 5.4. In the case of (5.1.8)

$$(A_1 u + A_0 u, u) \leq -\kappa\,(A_0\, u, u)$$

That is, we have a dissipation due to the diffusion term $\kappa\, A_0$.

5.2 The time reversal filter and forward and backward method

In this section, we discuss the time reversal method for the smoothing, i.e. estimation of the initial condition $x(0)$ form observations $\{y_s,\ s \leq [0,T]\}$ for (5.1.7). First, we define the time reversal function z by

$$z(t) = x(T-t), \quad 0 \in [0,T]$$

and thus $z(t)$ satisfies

$$-\frac{d}{dt} z(t) = f(z(t)) + \text{"}noise\text{"}, \quad z(0) = x(T) \tag{5.2.1}$$

with the time reversal observation

$$y(T-t) = Hz(t) + \text{"}noise\text{"}. \tag{5.2.2}$$

Therefore we will apply the assumed gain filter for the time reversal system (5.2.1)-(5.2.2) to obtain

$$-\frac{d}{dt}\hat{z}(t) = f(\hat{z}) + \tilde{G}(y(T-t) - H\hat{z}(t)) \tag{5.2.3}$$

with

$$\hat{z}(0) = \hat{x}(T).$$

Thus, the observation is used in the time reversal manner. Here assume the system (5.1.7) is time reversible, i.e., the system (5.2.1) is well-posed as an ODE for $z(t)$. For example, if we assume the system is conservative,

$$(f(x_1) - f(x_2), x_1 - x_2) = 0 \text{ and } f(0) = 0. \tag{5.2.4}$$

Then, it is time reversible and the same G can be used for the time reversal smoothing (5.2.3) and the Lyapunov stability (5.1.2) and the performance estimate (5.1.5) apply. Combining the forward step (5.1.1) for filtering and applying the smoothing by the time reversal (5.2.3), we obtain the forward and backward smoothing method.

5.2.1 Conservation law example

An example of conservative system with conservative property (5.2.4) that is time reversible, we discuss system of conservation laws of the form:

$$u_t + (F(u))_x + q(x)\,u = 0 \text{ in } R^d, \tag{5.2.5}$$

where $q(x)\,u$ is the potential term and $F(u) \subset R^d$ is a nonlinear flux. With flux function $F(u) = \frac{1}{2}u^2$ and when $q(x) = 0$, (5.2.5) becomes the inviscid Burgers' equation in R :

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0.$$

We consider the periodic boundary condition on $\Omega = [0, 1]$, i.e., $u(0) = u(1)$. In this case we have $u \in X = L^1(0, 1)$ space formulation, i.e, we have

$$I = \int_0^1 (F(u_1)_x - F(u_2)_x, sign(u_1 - u_2))\; dx = 0.$$

In fact for $u_1,\ u_2 \in C^1(0, 1)$

$$I = \textstyle\int_{u_1-u_2>0}(F(u_1)_x - F(u_2)_x) - \int_{u_1-u_2<0}(F(u_1)_x - F(u_2)_x)$$

$$= \textstyle\sum_j 2(\pm 1)F(u_1)(x_j) - F(u_2)(x_j) = 0$$

where $u_1(x_j) - u_2(x_j) = 0$ for a finitely points $x_j \in (0, 1)$. Thus, we have

$$|u(t) - \hat{u}(t)|_{L^1(0,1)} + \int_0^t \sum_i^p |u(t, x_i) - \hat{u}(t, x_i)|\, dt \le |u(0) - \hat{u}(0)|_{L^1(0,1)} + \int_0^t (\text{"}noise\text{"}, sign(u_1(s)))\, ds.$$

Then, we have

$$\int_0^1 (F(u_1)_x - F(u_2)_x, sign(u_1 - u_2))\; dx = 0$$

and we have the L^1 error estimate

$$|u_1(t) - u_2(t)|_{L^1(0,1)} \le |u_1(0) - u_2(0)|_{L^1(0,1)} + \int_0^t (\text{"}noise\text{"}, sign(s))\, ds. \tag{5.2.6}$$

That is, the Lyapunov estimate (5.1.5) is replaced by L^1 estimate (5.2.6).
For multidimensional the system of conservation laws such as Euler gas dynamics and Shallow water equations [34] is written as

$$u_t + div(\vec{F}(u)) = 0, \;\; u(0, x) = u_0(x) \text{ in } \mathbb{R}^d \tag{5.2.7}$$

where $\vec{F}$ is the vector valued flux function and u_0 is the initial condition. Equations must be understood in the distribution sense. For example if we take the volume of (5.2.7) we obtain the following conservation

$$\frac{d}{dt}\int_\Omega u(t,x)\,dx = \int_\Omega \frac{\partial}{\partial t}u\,dx = -\int_{\partial\Omega} \vec{n}\cdot\vec{F}(u)\,ds, \tag{5.2.8}$$

for all subdomain Ω. There is the PDE theory [7] for showing the existence of weak (distribution) solution satisfying

$$\int_{\mathbb{R}}\int_{\mathbb{R}^d}(-u\psi_t - \vec{F}(u(t,x))\cdot\nabla\psi)\,dx\,dt + \int_{\mathbb{R}^d} u(t,x)\psi(t,x)\,dx - \int_{\mathbb{R}^d} u_0(x)\psi(0,x)\,dx,$$

for all $\psi \in C_1^0(\mathbb{R}\times\mathbb{R}^d)$.
We apply the proposed filtering and smoothing methods for conservation equation (5.2.7). We assume the observation process $y(t)$

$$y(t) = Hu(t) + \text{"noise"}$$

For example point-wise observation of u at the observation locations $x_i \in \Omega$:

$$Hu = u(t,x_i), \quad x_i \in \Omega$$

for $1 \le i \le p$. The time-reversal filtering and smoothing algorithm consists of the filtering step

$$\frac{d}{dt}\hat{u}(t) + div(f(\hat{u})) + \gamma\, H^*(y(t) - H\hat{u}(t)) = 0 \tag{5.2.9}$$

forward in time and the smoothing step

$$\frac{d}{dt}z(t) + div(f(z)) + \gamma\, H^*(y(t) - Hz(t)), \quad z(T) = \hat{u}(T) \tag{5.2.10}$$

backward in time. Here we used the assumed covariance $\Sigma = \gamma\, I$.

5.2.2 Numerical test for the forward and backward method using conservation law

In this section we examine the one dimensional Burgers' equation $\Omega = (0,1)$

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0. \tag{5.2.11}$$

to test the forward-backward method (5.2.9) and (5.2.10) We use point-wise observations of the form

$$y_i(t) = u(t, x_i) + \text{ ''noise''},$$

at observation points $x_i \in \Omega = (0, 1)$. Next, we describe how to construct the one step map f in (2.1.1) for conservation law (5.2.5) with $q = 0$. We use the two steps Lax-Wendroff scheme (Richtmeyer) [24]

$$\begin{aligned} v_{k+\frac{1}{2}}^{n+\frac{1}{2}} &= v_{k+\frac{1}{2}}^{n-\frac{1}{2}} - \lambda \left(F\left(u_{k+1}^n\right) - F\left(u_k^n\right) \right) \\ u_{k+1}^{n+1} &= u_{k+1}^n - \lambda \left(F\left(v_{k+\frac{3}{2}}^{n+\frac{1}{2}}\right) - F\left(v_{k+\frac{1}{2}}^{n+\frac{1}{2}}\right) \right) \end{aligned} \tag{5.2.12}$$

where $\lambda = \frac{\Delta t}{\Delta x}$ is the CFL number and $F(u) = \frac{1}{2}u^2$ is the flux vector. In term of matlab code (5.2.12) is concluded in the following matlab.

```
for k=1:kmax; uu=[u(end);u];
u=u-.5*(uu(2:end).^2-uu(1:end-1).^2); vv(:,k)=u(:); end
```

Here the second line represent the forward filtering (5.2.9). Here, $v_{k+\frac{1}{2}}^{n+\frac{1}{2}}$ represents an intermediate value of u_k^n in time and space (n means time and k means space).

Now, we conduct the test and present our numerical results for the forward and backward algorithm (5.2.9)-(5.2.10). Here we pick the initial condition as

$$u_0 = .45(\cos(2pix) + 1) + .05$$

and the initial guess

$$\hat{u}_0(x) = 1 - .5x,$$

which are depicted in Figure 5.1.(a). We also have $\lambda = 1$ and the time and space step size $\Delta t = \Delta x = 0.01$. We use the measurements at 3 points

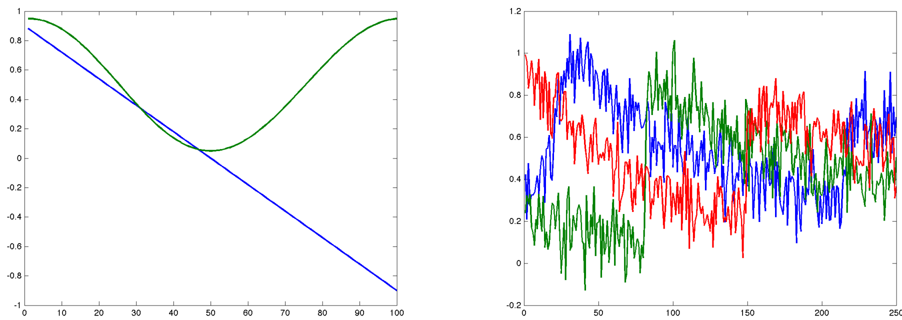
$$x_1 = .33,\ x_2 = .66 \text{ and } x_3 = 1,$$

and add noise to it over 200 points on the interval $(0, T = 2)$ so

$$y = y + .05\, randn(3, 200)$$

which is shown in Figure 5.1.(b). Our test is conducted by the forward filtering step (5.1.1) with initial guess $\hat{u}_0(x)$ and then apply the backward smoothing with

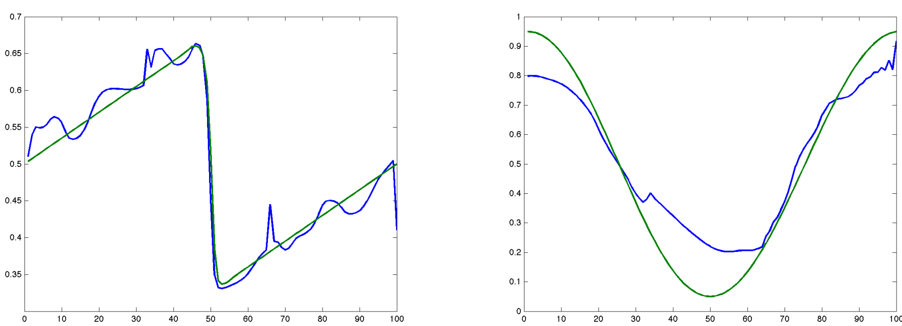
$$z(0) = \hat{u}(T) \quad (T = 2)$$



(a) initial estimate (b) noisy observations

Figure 5.1: initial condition and noisy observations for the Burgers' equation.

In Figure 5.2 (a) shows the filtering estimate $\hat{u}(T)$ at $T = 2$ and Figure 5.2 (b) shows the smoothing estimate $z(T)$ of initial condition u_0.



(a) foward smoothing (b) forward smoothing

Figure 5.2: forward-backward method for the Burgers' equation.

Despite the fact that $\hat{u}_0(x)$ was very away from $u_0(x)$ and we added significant noise asin Figure 5.1 (b) to the observations, the forward and backward smoothing method yield a very good estimate.

5.2.3 Numerical test for time reversal using the advection equation

Suppose we have the advection equation (3.4.3) of the form

$$\frac{\partial u}{\partial t} + (au)_x + (bu)_y = 0, \quad u(0,x) = u_0(x) \tag{5.2.13}$$

where we assume there is no diffusion.We consider that the line observation at Γ

$$y(t) = u(t,x) + \text{ "noise"}, \quad x \in \Gamma.$$

where $y(t)$ is observed at the observation line Γ depicted in Figure 5.3.

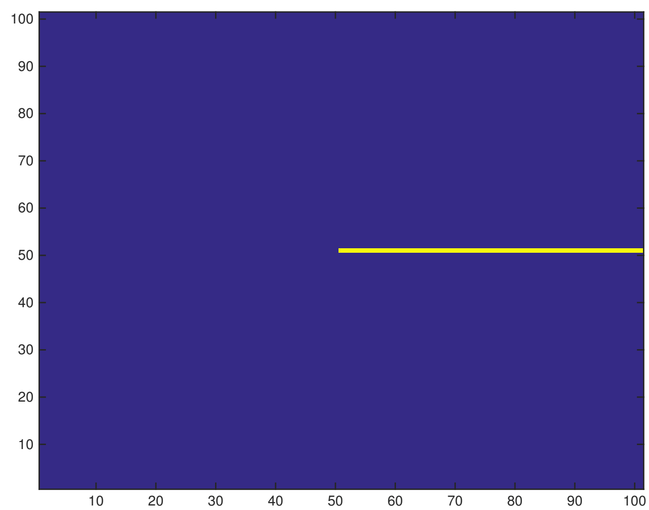


Figure 5.3: line observation in Γ

Next we define the one-step map (in time and space) $f(x_k)$ in (2.1.1) for system (5.2.13), We use the two steps Lax-Wendroff by Richtmyer [24, 29] :

$$
\begin{aligned}
v^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}} &= v^{n-\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}} - \frac{\lambda}{2}\Big[\Big((au)^n_{i+1,j+1} + (au)^n_{i+1,j}\Big) - \Big((au)^n_{i,j+1} + (au)^n_{i,j}\Big) \\
&\qquad + \Big((bu)^n_{i+1,j+1} + (bu)^n_{i,j+1}\Big) - \Big((bu)^n_{i+1,j} + (bu)^n_{i,j}\Big)\Big] \\
u^{n+1}_{i,j} &= u^n_{i,j} - \frac{\lambda}{2}\Big[\Big((av)^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}} + (av)^{n+\frac{1}{2}}_{i-\frac{1}{2},j+\frac{1}{2}}\Big) - \Big((av)^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2}} + (au)^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}}\Big) \\
&\qquad + \Big((bv)^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}} + (bv)^{n+\frac{1}{2}}_{i-\frac{1}{2},j+\frac{1}{2}}\Big) - \Big((bv)^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2}} + (bv)^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}}\Big)\Big]
\end{aligned}
\tag{5.2.14}
$$

where x_k stands for the velocity $u_{i,j}$ at uniformly nodes over $\Omega = (0,1) \times (0,1)$ and $v^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2}}$ is an intermediate value of $u_{i,j}$ in time and space.

Now, we present our numerical results for the time reversal algorithm (5.2.1)-(5.2.3) for (5.2.13). If we let he vector field $\vec{a}$ be defined by

$$a(x,y) = y - .5 \quad \text{and} \quad b(x,y) = .5 - x, \qquad \text{over } \Omega = (0,1) \times (0,1)$$

which results in the clockwise rotating transformation. Thus, the initial condition $u_0(x)$ will reach to the observation line Γ after some time (i.e., we set $T = 3$ and 270° degree of rotation). We pick u_0 as depicted in Figure 5.4

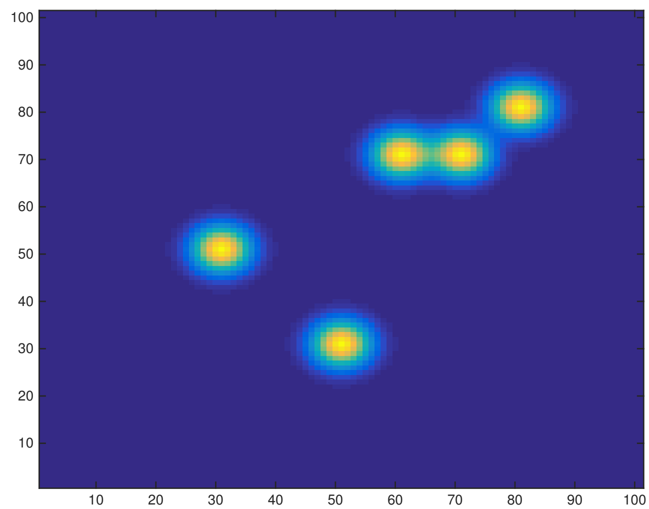


Figure 5.4: u_0, $T = 3$.

We conduct tests with $\Delta t = \Delta x = 0.01$ and the CFL number $\lambda = \dfrac{\Delta t}{\Delta x} = 1$. In this case the time-reversal implies rotating backward in time. That is, our method will demonstrate the capability of the time reversal filter (in a very transparent manner) by

$$\begin{aligned} &\tfrac{\partial z}{\partial t} - (az)_x - (bz)_y = 0, \quad z(0,x) = 0 \\ &z(t,x) = y(T-t). \end{aligned} \tag{5.2.15}$$

The following matlab shows the time integration of (5.2.14)

```
kmax=300;uu=[];for k=1:kmax;
u(j)=u(j)-lmd*(c1*(aa.*v)+c2*(bb.*v));
v=v-lmd*(b1*(a.*u)+b2*(b.*u)); uu=[uu u]; end
```

where the second and third lines represent the two time steps (5.2.14). We use 50 points on observation line Γ and 300 time steps $(0,3)$, $T = 3$ for observation y, which is shownFigure 5.5. The figure shows clearly the arrival of the initial condition $u_0(x)$ by the rotation,

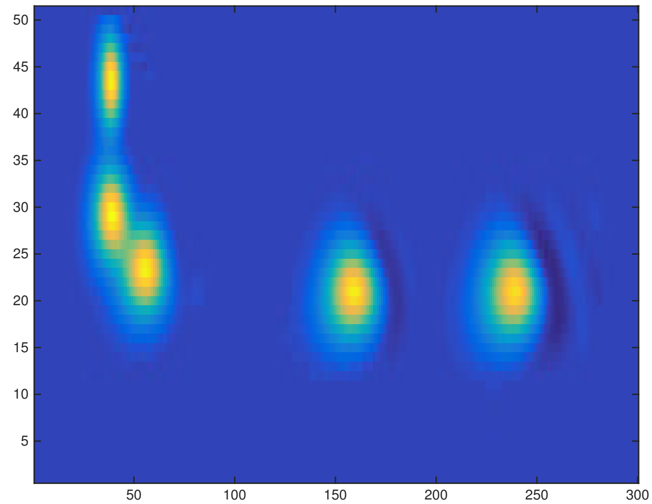
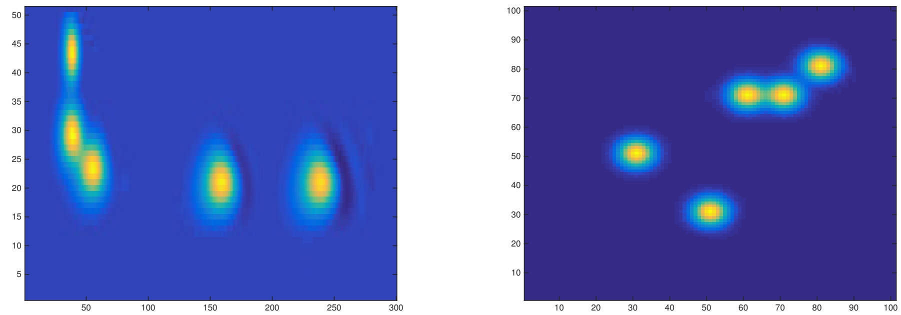


Figure 5.5: measurements obtain at line of observation Γ

Now we proceed to the time reversal method (5.2.1)-(5.2.2). The following matlab shows the time reversal method:

```
u1=zeros(m^2,1); v1=0*zeros(n^2,1);
uu=[];for k=kmax:-1:1;
u1(i)=yy(:,k);
v1=v1+lmd*(b1*(a.*u1)+b2*(b.*u1));
u1(j)=u1(j)+lmd*(c1*(aa.*v1)+c2*(bb.*v1));
u1(kk)=0;
uu=[uu u1]; end
```

which is for the time integration of (5.2.14) in backward smoothing to obtain the estimate for the initial $u(0) = u_0$.



(a) y with no noise

(b) $\hat{u}_0$ time reversal

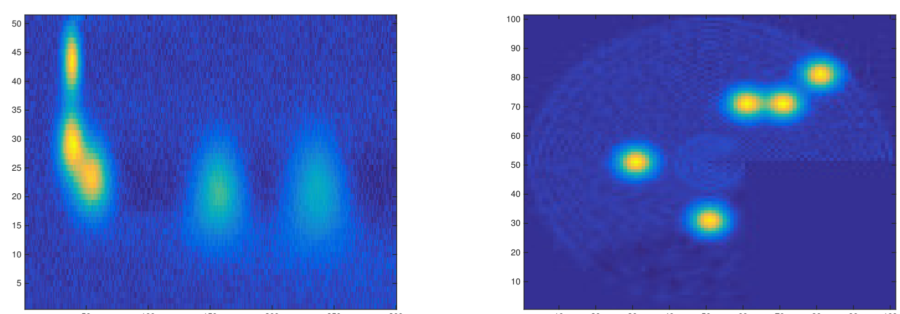
Figure 5.6: time Reversal for the advection equation no noise in measurements.

First, Figure 5.6.(b), shows what we obtain after applying the time reversal method without any noise added in the observations (No noise case). In this case we numerically integrated (5.2.15), which is the backward rotation.

Now we test the time reversal algorithm with noise in observations.

$$y = y + .1rand(size(y)).$$

Figure 5.8.(b) shows the time reversal result.



(a) y with noise

(b) $\hat{u}_0$ time Reversal

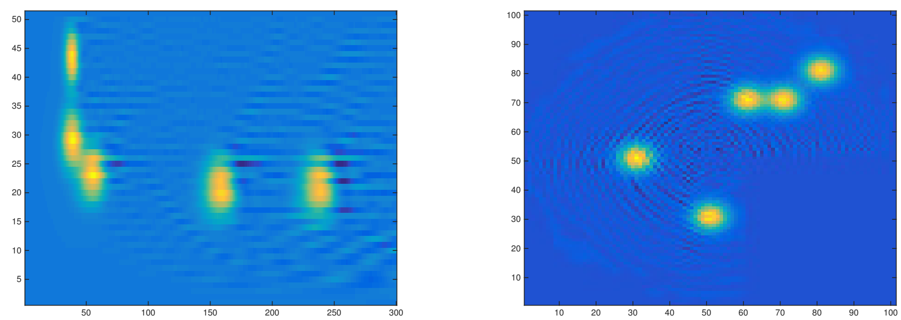
Figure 5.7: time reversal on advection with noisy measurements.

Now we test the time reversal method with added random perturbation in the vector field

$\vec{a}$:

$$a(x, y) = \bar{a} + .02\,(rand(size(a)) - .5) \text{ and } b(x, y) = \bar{a} + .02\,(rand(size(a)) - .5)$$

where $\bar{a}$ and $\bar{b}$ where defined above. Figure 5.8 (a) shows the effect of perturbation in the media in observations y. It show clearly the perturbation of the vector field in observations at Γ. Figure 5.8.(b) illustrates that even with the perturbation of the vector field, using the time reversal algorithm (5.1.1)-(5.2.3) for (5.2.13), we obtain an excellent estimate of initial condition u_0



(a) y, perturbed $\vec{a}$ (b) $\hat{u}_0$ time reversal

Figure 5.8: time reversal on advection with perturbed vector field $\vec{a}$.

In summary, we can conclude that time reversal algorithm (5.1.1)-(5.2.3) for the advection equation (5.2.13) works so well and very efficient and thus an effective algorithm.

5.3 Quasi-reversible

We cannot apply the time reversal (5.2.1)-(5.2.2) method for dynamics with diffusion. For example for diffusion-advection equation of the form

$$\frac{\partial u}{\partial t} + \nabla \cdot (a(x)u) = \kappa\,\Delta u \tag{5.3.1}$$

where $a(x)$ is a vector field and $\kappa > 0$ is a constant conductivity of media. Due to diffusion, the system (5.3.1) is not time reversible, i.e., backward integration in time is not possible [1]. So, we are interested in the case of mildly diffusion, i.e., κT is sufficiently small where $\kappa > 0$ is diffusion constant and $T > 0$ is the terminal time or the duration of the time reversal. To perform the back rad integration in time, we will use the quasi-reversal method, which has been used for the backward heat equation in [23, 28].

In general, if we consider system dynamics of the form

$$\frac{\partial u}{\partial t} + A_1 u = \kappa\, A_0 u \tag{5.3.2}$$

where A_1 is conservative, i.e., $(A_1 u, u) = 0$ and A_0 is a negative self-adjoint operator on a Hilbert space X. Then, we have the dissipative property

$$\frac{1}{2}\frac{d}{dt}|u|_X^2 = -(A_1 u, u) + \kappa\,(A_0 u, u) = \kappa\,(A_0 u, u). \tag{5.3.3}$$

Thus, we obtain the estimate

$$|u(T)|^2 = |u(0)|^2 + 2\kappa \int_0^T (A_0 u(t), u(t))\, dt. \tag{5.3.4}$$

This implies that $|u(T)| \le |u(0)|$ but, the converse is not true. This means that $|u(0)|$ cannot be estimated by $|u(T)|$, i.e., (5.3.2) is not time reversible in general. The quasi-reversal method is used to make (5.3.2) time reversible for $\kappa\, T$ sufficiently small. For example, we have a method that uses the following backward equation for (5.3.2):

$$u_t + A_1 u = \kappa\; A_0 u - \epsilon (A_0 u)_t, \quad u(T) = u_1 \tag{5.3.5}$$

for $\epsilon > 0$ chosen appropriately [1, 6]. That is, $z(t) = u(T-t)$ satisfies the time reversal equation

$$z_t - A_1 z = -\kappa\, A_0 z + \epsilon (A_0 z)_t, \quad z(0) = u_1. \tag{5.3.6}$$

5.3.1 Quasi-reversal algorithm

We consider the time-splitting integration for (5.3.6): first we solve

$$z_t - A_1 z = 0, \quad z(t_n) = z_n \text{ on } [t_n, t_{n+1}]. \tag{5.3.7}$$

and let $u_{n+1/2} = z(t_{n+1})$, since A_1 is conservative, $|z_{n+1/2}| = |z_n|$. Equivalently to (5.3.6)–(5.3.7) we update z_{n+1} using the implicit and explicit scheme by

$$\frac{z_{n+1} - z_{n+1/2}}{\Delta t} = \kappa\, H z_{n+1/2} - \epsilon H \frac{z_{n+1} - z_{n+1/2}}{\Delta t} \tag{5.3.8}$$

where $H = -A_0$ is positive i.e.,

$$\begin{aligned} z_{n+1} &= (I + \epsilon\, H)^{-1}(I + \kappa \Delta t\, H + \epsilon\, H) z_{n+1/2} \\ &= z_{n+1/2} + \kappa \Delta t\, (I + \epsilon\, H) H z_{n+1/2}. \end{aligned} \tag{5.3.9}$$

5.3.2 Stability estimate

Now, we have a stability estimate for $\{z_n\}$ generated by (5.3.7)–(5.3.8):

Theorem

$$|z_n| \leq \left(1 + \frac{\kappa}{\epsilon}\Delta t\right)^n |z_0|. \tag{5.3.10}$$

Proof: From (5.3.9) we have

$$|z_{n+1}| \leq |z_{n+1/2}| + \kappa\Delta t \max_{\lambda\in\sigma(H)} \frac{\lambda}{1+\epsilon\,\lambda}\,|z_{n+1/2}|.$$

Thus,

$$|z_{n+1}| \leq \left(1 + \frac{\kappa}{\epsilon}\Delta t\right) |z_{n+1/2}|$$

Therefore, we obtain estimate

$$|z_n| \leq \left(1 + \frac{\kappa}{\epsilon}\Delta t\right)^n |z_0| \square$$

Remark:

From (5.3.10) we have

$$\left(1 + \frac{\kappa}{\epsilon}\Delta t\right)^n \to e^{\frac{\kappa}{\epsilon}t} \text{ as } \Delta t \to 0. \tag{5.3.11}$$

Thus, κT must be sufficiently small.

5.3.3 Error analysis

Now, we discuss the error analysis for u_ϵ (5.3.5) to u for the backward equation for (5.3.2):

$$\frac{\partial u}{\partial t} + A_1 u = \kappa\, A_0 u, \quad u(T) = u_1.$$

We assume $A_1 = 0$ for the sake of simplicity. Since A_0 is negative self-adjoint we have the spectral resolution [35]

$$-A_0\phi = \int_{R^+} dE(\lambda)\phi \tag{5.3.12}$$

where $E(\lambda)$ is projection operators on a Hilbert space X. For example for the case of discrete spectra

$$-A_0\phi = \sum_{k=1}^{\infty} (\phi, \psi_k)\,\psi_k$$

where $(-\lambda_k, \psi_k)$ are eigen-pairs

$$A_0\psi_k = -\lambda_k\,\psi_k, \quad \lambda_k \geq 0$$

and $\{\psi_k\}$ are orthonormal. We assume that $u_1 \in \text{ span}\{\psi_k\}_{k=1}^m$. Then by the Fourier method for u to equation (5.3.2) we have

$$u(t) = \sum_{k=1}^{m} (u_1, \psi_k)\, e^{\kappa\lambda_k t}\, \psi_k \tag{5.3.13}$$

For (5.3.6)

$$u_\epsilon(t) = \sum_{k=1}^{m} \alpha_k(t)\, \psi_k \tag{5.3.14}$$

where $\alpha_k(t)$ satisfies

$$\dot{\alpha}_k = \kappa\, \lambda_k \alpha_k(t) - \epsilon\lambda_k \dot{\alpha}_k.$$

Thus, we have

$$\alpha_k(t) = e^{\frac{\kappa\lambda_k}{1+\epsilon\lambda_k} t}\, (u_1, \psi_k)$$

It follows from (5.3.13) and (5.3.14) that

$$|u(t) - u_\epsilon(t)|_X^2 = \sum_{k=1}^{m} \left| e^{\frac{\kappa\lambda_k}{1+\epsilon\lambda_k} t} - e^{\kappa\lambda_k t} \right| |(u_1(t), \psi_k)|^2 \tag{5.3.15}$$

Therefore, from (5.3.15) we have the worth case estimate

$$|u(t) - u_\epsilon(t)|_X^2 = \max_{\lambda} |F_\lambda(t)|\, |u(t)|^2 \sim \epsilon. \tag{5.3.16}$$

where

$$F_\lambda(t) \;=\; e^{\frac{\kappa\lambda t}{1+\epsilon\lambda}} - e^{\kappa\lambda t} \tag{5.3.17}$$

$$\;=\; e^{\kappa\lambda t}(e^{-\kappa\lambda \frac{\epsilon\kappa\lambda t}{1+\epsilon\kappa\lambda}} - 1). \tag{5.3.18}$$

This estimate also holds for the spectral resolution of $-A_0$.From (5.3.15) and (5.3.16) the quasi-reversible method works if κT is sufficiently small and thus ϵ can be smaller than κT to obtain the accuracy. For example, in the case (5.3.1) we have

$$A_0 = \Delta u \text{ and } A_1 u = \nabla \cdot (a(x)u)$$

with $u = 0$ at boundary $\partial\Omega$. In this case $\lambda_{k,\ell} = (k^2 + \ell^2)\pi^2$ and

$$\psi_{k,\ell}(x_1, x_2) \simeq \sin(k\pi x_1)\sin(\ell\pi x_2)$$

for $\Omega = (0,1) \times (0,1)$.

For an alternative quasi-reversal method discussed in [23], (5.3.2) has the form

$$\frac{d}{dt}u + A_1 u(t) = \kappa\, A_0 u(t) - \epsilon\, A_0^2 u(t), \quad u(T) = u_1. \tag{5.3.19}$$

The stability and error estimate analysis can be performed for (5.3.19) using exactly the same analysis as above.

5.3.4 Numerical test for quasi-reversible method using the conduction equation

We tested the quasi-reversal algorithm (5.3.7)–(5.3.8) for the conduction equation (5.3.1) over $\Omega = (0,1) \times (0,1)$:

$$\frac{\partial u}{\partial t} + \nabla \cdot (\vec{a}u) = \kappa\, \Delta u, \quad u = 0 \text{ at boundary } \partial\Omega. \tag{5.3.20}$$

We use the two-step integration (5.2.10) for the first step (5.3.7). For the second step (5.3.8) we use the central difference approximation for Laplacian Δ by

$$\Delta_h\, u = \frac{u_{i-1,j} - 2u_{ij} + u_{i-1,j}}{\Delta x^2} + \frac{u_{i,j+1} - 2u_{ij} + u_{i,j-1}}{\Delta x^2}.$$

Thus, we have matlab code for the implicit and explicit time splitting (5.3.9). Here is the matlab that shows the backward operation for the quasi reversal.

```
uu=[];for k=kmax:-1:1;
u1(i)=yy(:,k);
v1=v1+lmd*(b1*(a.*u1)+b2*(b.*u1));
v1=(q0+ep*h0)\(q0*v1+dt*ka*h0*v1+ep*h0*v1);
u1(j)=u1(j)+lmd*(c1*(aa.*v1)+c2*(bb.*v1));
u1=(q+ep*h)\(q*u1+dt*ka*h*u1+ep*h*u1);
u1(kk)=0;
uu=[uu u1]; end
```

Here, $v1$ is the intermediate step $z_{n+\frac{1}{2}}$ and $u1$ represents z_{n+1}. The third and fifth lines perform (5.3.7) and the fourth and sixth lines perform (5.3.9).

Now, we describe the problem setup. We let the initial condition $u_0(x)$ be the same one as we had for the time reversal of the advection equation which is depicted in Figure 5.9,

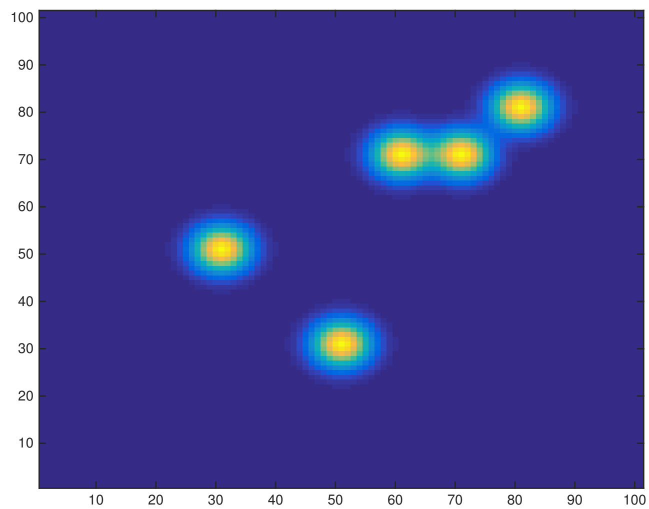
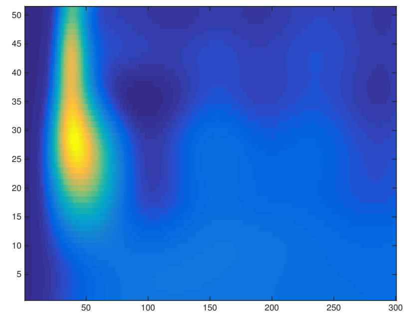


Figure 5.9: u_0.

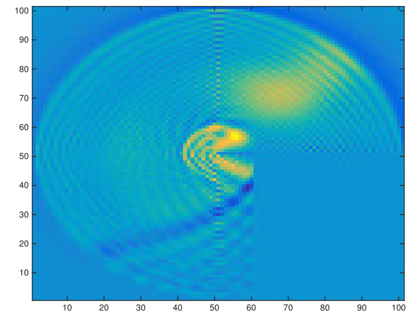
and we the vector field $\vec{a}$, the same as in Section 5.2.3

$$\vec{a} = \vec{y} - .5 \text{ and } \vec{b} = .5 - x.$$

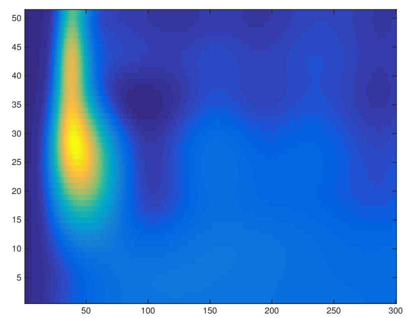
We are going to use $\kappa = .01$ and $\epsilon = .01$ in (5.3.5). As we describe in the time reversal Section 5.2.3, we use the line observation Γ as depicted in Figure 5.3. Recall that Γ is chosen so that the initial condition $u_0(x)$ is transformed by the advection to Γ. Here, we have a diffusion (a heat conduction term) and (5.3.20) is not time reversible.



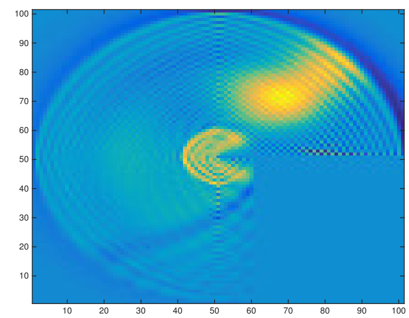
(a) y, $\kappa = .01$, $\epsilon = .01$



(b) $\hat{u}_0$, time Revresal, $\kappa = .01$, $\epsilon = 0$, $\kappa = 0$



(c) y, $\kappa = 0.01$



(d) $\hat{u}_0$, quasi reversal, $\epsilon = .01$ $\kappa = 0.01$

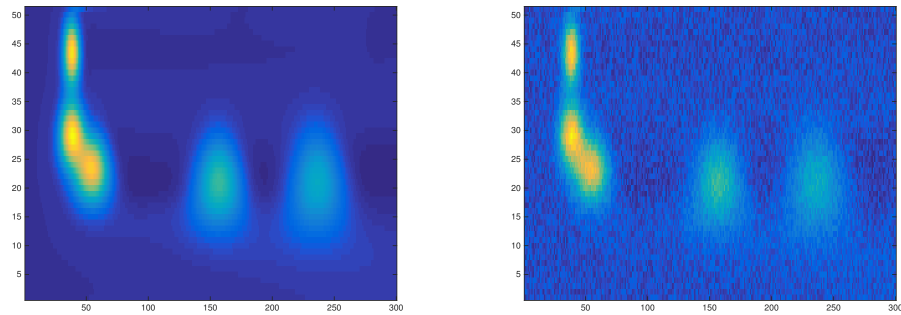
Figure 5.10: time reversal vs quasi reversal method with $\kappa = 0.01$ and $\epsilon = 0.01$.

Figure 5.10 (b) is the result by the time reversal method (5.1.1)-(5.2.3) (i.e. we use (5.2.15) to estimate the initial condition by zT). and Figure 5.10 (d) is the result by the quasi-reversal method (5.3.7)–(5.3.8). This means that using the Quasi-reversible method, in which the diffusion is not disregarded we obtain a better estimate $\hat{u}_0(x)$. Methods is tested for a large diffusion ($\kappa = .01$).

Now we examine the quasi reversal method for the case of a smaller diffusion ($\kappa = .001$). We change ϵ to examine effectiveness of quasi-reversal method when we have no noise in the observations as depicted in Figure 5.11.(a). We also test added noise observations case by

$$y = y + .1rand(size(y))$$

which is depicted in Figure 5.11.(b).

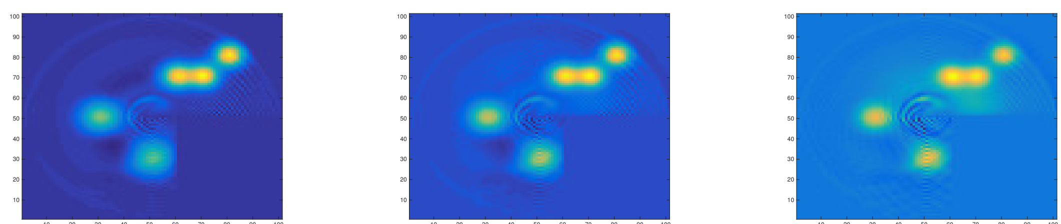


(a) y with no noise

(b) y with noise

Figure 5.11: observation obtain for quasi reversal method with $\kappa = .001$, $\epsilon = .005$

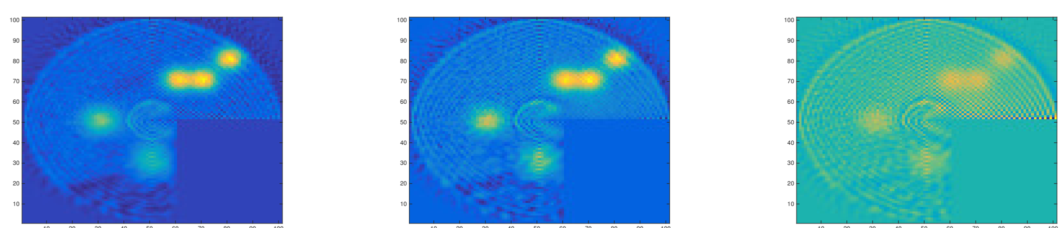
In Figures 5.12 (a)–(c) the results for non-noise case are shown with three different $\epsilon = 0.005,\ 0.002$, and 0.001. The corresponding results for noise case are shown in Figures 5.12 (d)–(f).



(a) $\hat{u}_0$, $\kappa = .001$, $\epsilon = 0.005$ no noise

(b) $\hat{u}_0$, $\kappa = .001$, $\epsilon = 0.002$ no noise

(c) $\hat{u}_0$, $\kappa = .001$, $\epsilon = 0.001$ no noise



(d) $\hat{u}_0$, $\kappa = .001$, $\epsilon = 0.005$ with noise

(e) $\hat{u}_0$, $\kappa = .001$, $\epsilon = 0.002$ with noise

(f) $\hat{u}_0$, $\kappa = .001$, $\epsilon = 0.001$ with noise

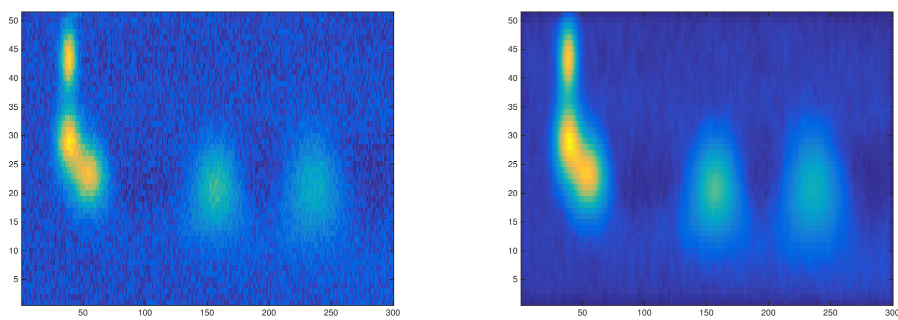
Figure 5.12: $\hat{u}_0$ using quasi reversible algorithm with No noise vs with Noise ϵ updated

Note that $\epsilon = .001$ provides the best estimate for non-noise came, which agree with for error estimate (5.3.14). But with noisy observation $\epsilon = .002$ is best among three case we tested, which also agrees with our error estimate (5.3.14).

As shown in our analysis. data must be smooth, so we have to find a way to remove noise from data and we use the regularization method:

$$y \leftarrow (I + \beta H)^{-1} y$$

where H is the time and space negative Laplacian. We use the regularization constant $\beta = 10^{-3}$. The regularized data is depicted in Figure 5.13.(b),

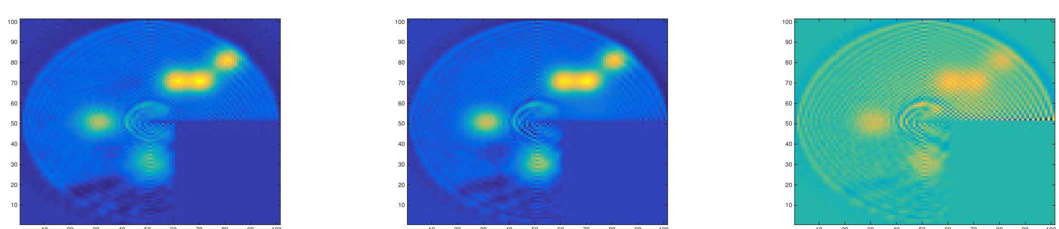


(a) y with noise

(b) y regularized

Figure 5.13: y with noise vs regularized noised with $\kappa = .001$ and $\epsilon = .005$

In Figures 5.14.(a)–(c) we show the quasi-reversal results for y regularized.



(a) $\hat{u}$, $\kappa = .001$, $\epsilon = 0.005$ with$y$ regularized

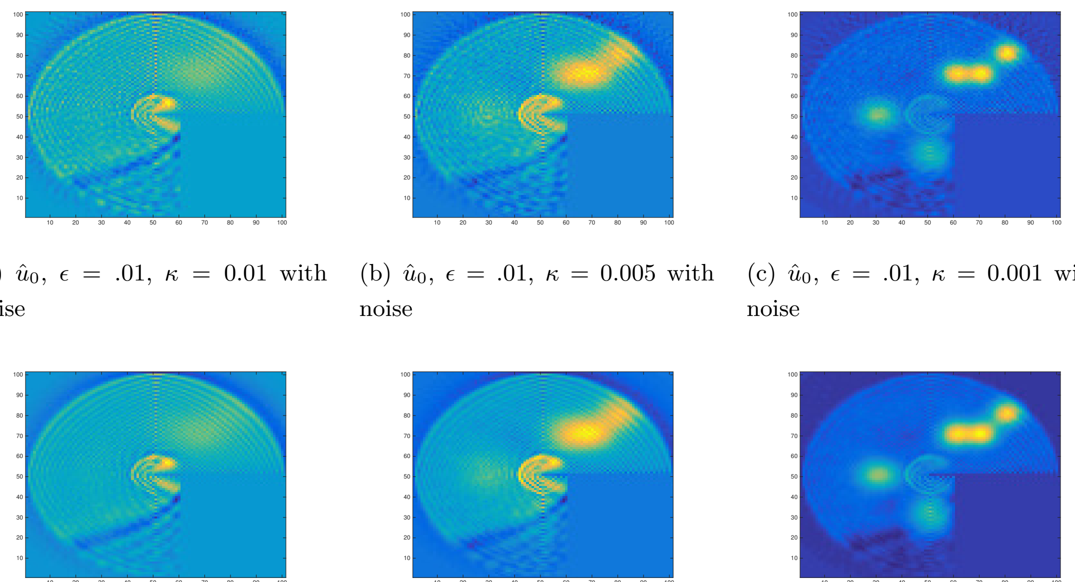
(b) $\hat{u}$, $\kappa = .001$, $\epsilon = 0.002$ with y regularized

(c) $\hat{u}$, $\kappa = .001$, $\epsilon = 0.001$ with y regularized

Figure 5.14: $\hat{u}$ using quasi reversal results with regularized observations

Comparing the result Figures 5.12 (d)-(f) without shooting of y for all three selected ϵ the one with smoothing data does much better job. Again it agree with our error analysis (5.3.14).

Now test the quasi-reversal method (5.3.7)–(5.3.8) with $\epsilon = .01$ fixed and varying $\kappa = .01,\ .005,\ .001$. The first case with noise y and y regularized. Figure 5.15 shows the estimated initial conditioned $\hat{u}_0$.



(a) $\hat{u}_0$, $\epsilon = .01$, $\kappa = 0.01$ with noise

(b) $\hat{u}_0$, $\epsilon = .01$, $\kappa = 0.005$ with noise

(c) $\hat{u}_0$, $\epsilon = .01$, $\kappa = 0.001$ with noise

(d) $\hat{u}_0$, $\epsilon = .01$, $\kappa = 0.01$ with cleaned noise

(e) $\hat{u}_0$, $\epsilon = .01$, $\kappa = 0.005$ with cleaned noise

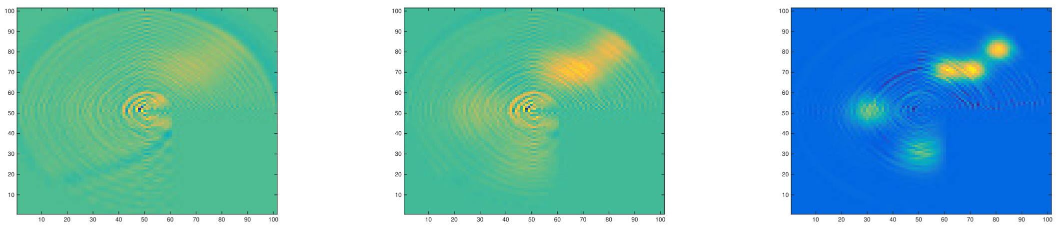
(f) $\hat{u}_0$, $\epsilon = .01$, $\kappa = 0.001$ with cleaned noise

Figure 5.15: $\hat{u}_0$ with quasi reversal and $\epsilon = .01$ with noise added and regularized

The next test is for the case of randomly perturbed vector field by

$$a(x, y) = \bar{a} + .02(rand(size(a)) - .5) \text{ and } b(x, y) = \bar{a} + .02(rand(size(a)) - .5).$$

and the corresponding estimates are shown in Figure 5.16. With random perturbation in the vector rotational field $\vec{a}$ and $\kappa = 10^{-3}$, we get an estimate that is close to the initial condition as depicted in Figure 5.16.(c).



(a) $\hat{u}_0$, $\epsilon = .01$, $\kappa = 0.01$ (b) $\hat{u}_0$, $\epsilon = .01$, $\kappa = 0.005$ (c) $\hat{u}_0$, $\epsilon = .01$, $\kappa = 0.001$

Figure 5.16: $\hat{u}_0$ with quasi reversal algorithm results and $\epsilon = .01$ for perturbed vector field $\vec{a}$

Thus we conclude that the numerical results agree with our error analysis.

Chapter 6

Parameter Dependent Case

In this Chapter, we consider the parameter dependent discrete dynamics:

$$x_k = f(x_{k-1}, a_{k-1}) + w_k$$

$$a_k = a_{k-1} + \tilde{w}_k.$$

where the parameter in R^{N_p} is augmented to the system as random a coefficient by the second equation. In this case

$$F(x,a) = \begin{pmatrix} f(x,a) \\ a \end{pmatrix}.$$

We also consider the mean return model:

$$a_k - \bar{a} = \gamma(a_{k-1} - \bar{a}) + \tilde{w}_k \tag{6.0.1}$$

where $0 < \gamma < 1$ and $\bar{a}$ is a known mean of a_k. It is a time discretization of a SDE

$$da_t = \alpha(\bar{a} - a_t) + \tilde{\sigma} dB_t.$$

In fact we have

$$a_{t+\Delta t} = e^{-\alpha\Delta t} a_t + (1 - e^{-\alpha\,\Delta t})\,\bar{a} + \int_t^{t+\Delta t} e^{t+\Delta t - s} \sigma\, dB_s$$

If we let $\gamma = e^{-\alpha\Delta t}$, then we obtain (6.0.1) For such a case

$$F(x,a) = \begin{pmatrix} f(x,a) \\ \gamma(\bar{a} - a) \end{pmatrix}.$$

For (2.5.8) and (2.5.9) CDF compete the central difference components of $F(x,a)$ for the joint state $\tilde{x}=(x,a)$. The extended Kalman filter uses the Jacobian of the form

$$F' = \begin{pmatrix} f_x & f_a \\ 0 & I \end{pmatrix}, \quad F' = \begin{pmatrix} f_x & f_a \\ 0 & -\gamma\, I \end{pmatrix}$$

For the parameter dependent case it is important to update the state and the parameters jointly, then we use the assumed covariance update for the predictor step following (2.3.4) as follows:

$$\begin{aligned}
F'\Sigma F'^T &= \begin{bmatrix} f_x & f_a \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma_{xx} & \Sigma_{xa} \\ \Sigma_{ax} & \Sigma_{aa} \end{bmatrix} \begin{bmatrix} f_x^T & 0 \\ f_a^T & I \end{bmatrix} \\
&= \begin{bmatrix} f_x\Sigma_{xx} + f_a\Sigma_{ax} & f_x\Sigma_{xa} + f_a\Sigma_{aa} \\ \Sigma_{ax} & \Sigma_{aa} \end{bmatrix} \begin{bmatrix} f_x^T & 0 \\ f_a^T & I \end{bmatrix} \\
&= \begin{bmatrix} f_x\Sigma_{xx}f_x^T + f_a\Sigma_{ax}f_x^T + f_x\Sigma_{xa}f_a^T + f_a\Sigma_{aa}f_a^T & f_x\Sigma_{xa} + f_a\Sigma_{aa} \\ \Sigma_{ax}f_x^T + \Sigma_{aa}f_a^T & \Sigma_{aa} \end{bmatrix}
\end{aligned}$$

where Σ_{xx} is the covariance of the state, $\Sigma_{xa} = \Sigma_{ax}^T$ represents the covariance of the state knowing the parameter a and finally Σ_{aa} is the covariance of the parameter. For CDF case we use the covariance update for x (with $\Sigma_{11} = S^T S$ or reduced $S_m^T S_m$ as factor). We may used the assumed covariance update of the form:

$$a_i = \frac{f(\bar{x} + S_{:,i}h, a) - f(\bar{x} - S_{:,i}h, a)}{2h}$$

$$H_{ii} = \frac{f(\bar{x} + S_{:,i}h, a) - 2f(\bar{x}, a) + f(\bar{x} - S_{:,i}h, a)}{h^2}$$

Then,

$$(\Sigma_{11})_{k|k-1} = \sum_{i=1}^{m} a_i a_i^T + \frac{1}{2}\sum_{i=1}^{m} H_i H_i^T.$$

For Σ_{12} we compute

$$b_i = \frac{f(\bar{x}, \bar{a} + h\, e_i) - f(\bar{x}, \bar{a} - h\, e_i)}{2h}, \quad 1 \le i \le N_p$$

and

$$(\Sigma_{12})_{k|k-1} = \sum_i^m a_i U_m \, (\Sigma_{12})_{k-1|k-1}$$

$$\Sigma_{22} = \Sigma_{22}$$

For the specific case where F is linear in x:

$$F(x,a) = A(a)x \tag{6.0.2}$$

and $a \to A(a)$ is linear, i.e., $A(a)$ are coefficients matrix for the linear dynamics. Then, we have

$$F' = \begin{pmatrix} A(a) & \dot{A}x \\ 0 & I \end{pmatrix}, \quad F' = \begin{pmatrix} A(a) & \dot{A}x \\ 0 & -\gamma\, I \end{pmatrix}$$

where $\dot{A}$ is the derivative of the coefficient matrix with respect a.
For example if we consider the second order mass spring system of the form,

$$\ddot{y} + \gamma\dot{y} + \omega^2 y = \text{noise}.$$

For the first order system for $x(t) = (y(t), y'(t))$ we have

$$\frac{d}{dt}x(t) = \begin{pmatrix} 0 & 1 \\ & \\ -\omega^2 & -\gamma \end{pmatrix} x(t)$$

Since linear in x the Gaussian filter has $H_{ii} = 0$.

But, if we do the joint covariance for $F(x,a)$ we have the Hessian for cross term $\frac{\partial^2 F}{\partial x \partial a} = (\dot{A})$. Also,

$$a = AS^T, \quad b = U\dot{A}x, \quad H = U\dot{A}$$

Now, we describe how to a_i and H_{ii} in formulae (4.1.2) with the coordinate transform

$$x = S^T s + \bar{x}$$

and

$$\tilde{F}(s) = F(S^T s + \bar{x}).$$

In fact we have

$$\frac{\partial \tilde{F}}{\partial s} = S^T \frac{\partial F}{\partial x}$$

i.e.,

$$a^{(k)} = S^T \frac{\partial F^{(k)}}{\partial x}.$$

Next, since

$$\frac{\partial^2 \tilde{F}}{\partial s_i \partial s_j} = S^T \frac{\partial^2 F}{\partial x_i \partial x_j} S,$$

we have

$$H^{(k)} = diag\left(S^T \frac{\partial^2 F^{(k)}}{\partial x_i \partial x_j} S\right).$$

6.1 Lorenz equations

In this section we introduce an SDE model for premature dependent dynamics and test the Gaussian filter in Section 2.4. Consider the Lorenz system as

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = x(\rho - z) - y \\ \frac{dz}{dt} = xy - \beta z. \end{cases} \tag{6.1.1}$$

It is introduced by by Edward Lorenz. It is an example of 3 dimensional dynamics having chaotic solutions for certain parameter values and initial conditions. That is, in 1963, Edward Lorenz developed a simplified mathematical model for atmospheric convection [26]. Here, (x, y, z) represent the the system state, (two for the stream function and the one for temperature). σ, β and ρ the system parameters. These parameters are scaled parameters from the atmospheric convection model, for example $\sigma > 0$ is the *Prandtl number*.

We test the Gaussian filter for the case $\sigma = 10\, \rho = 28$ (fixed) and β is a random parameter around $\bar{\beta} = 8/3$. Thus, we estimate state (x, y, z) and a parameter β in model (6.1.1) based on the observations

$$y = x(3) + .1sqrt(dt)randn,$$

which only observes the third component of the state.

We first describe the one-step map f for (6.1.1). The system can be rewritten in the form of

$$\frac{d}{dt}x = A(a)x$$

we get

$$\frac{d}{dt}\begin{pmatrix} x \\ \\ y \end{pmatrix} = \begin{pmatrix} \sigma & -\sigma \\ \\ \rho - z & -1 \end{pmatrix} \cdot \begin{pmatrix} x \\ \\ y \end{pmatrix} \tag{6.1.2}$$

where

$$A(a) = \begin{pmatrix} \sigma & -\sigma \\ \\ \rho - z & -1 \end{pmatrix}$$

The solution of (6.1.2) is given by

$$\begin{pmatrix} x \\ \\ y \end{pmatrix} = e^{A(a)\Delta t}\begin{pmatrix} x \\ \\ y \end{pmatrix}.$$

Now we can find a solution for

$$\begin{aligned} \frac{dz}{dt} &= xy - \beta z \\ &= \beta\left(\frac{xy}{\beta} - z\right). \end{aligned}$$

since x and y are given, then

$$z_{n+1} = \frac{xy}{\beta}\left(1 - e^{-\beta\Delta t}\right) + e^{-\beta\Delta t} z_n. \tag{6.1.3}$$

Thus, we define the one-step map for (6.1.1) as

$$f(x_n, y_n, z_n) = \begin{pmatrix} e^{A(a)\Delta t}\begin{pmatrix} x_n \\ \\ y_n \end{pmatrix} \\ \\ \frac{x_{n+1}y_{n+1}}{\beta}\left(1 - e^{-\beta\Delta t}\right) + e^{-\beta\Delta t} z_n. \end{pmatrix} \tag{6.1.4}$$

The first two coordinates of (6.1.4) are the frozen coefficients $A(a)$ with $z = z_n$ and the last coordinate is updated by (6.1.3) frozen $x = x_{n+1}$ and $y + y_{n+1}$. Here is the matlab that summarizes the Gaussian Filter for the joint state-parameter estimation for (6.1.1):

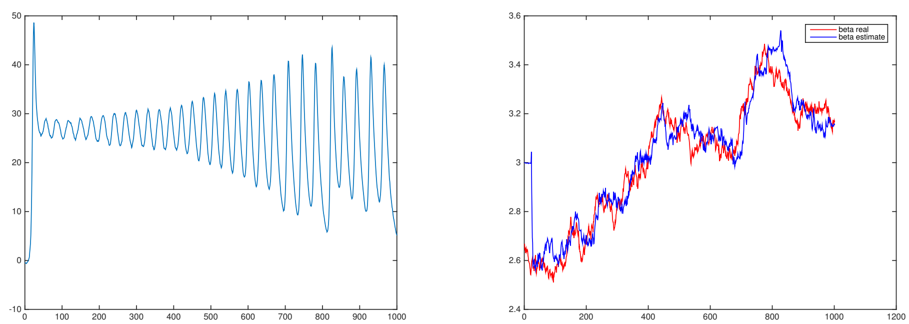
```
MM=M;for kk=1:nmax;
S=chol(P); x=S'*xx+M*ones(1,9);
for k=1:9; A=[-sgm sgm; rho-x(3,k) -1]; f(1:2,k)=expm(A*dt)*x(1:2,k);
bt=x(4,k); s=exp(-bt*dt);
```

```
f(3,k)=(1-s)*f(1,k)*f(2,k)/bt+s*x(3,k); f(4,k)=x(4,k); end
for k=1:4;
g1(:,k)=f(:,2*k+1)-f(:,2*k); g2(:,k)=f(:,2*k+1)-2*f(:,1)+f(:,2*k); end
g1=g1/2/h; g2=g2/h/h;
M=f(:,1); for k=1:4;M=M+.5*g2(:,k); end
P=[0 0 0;0 0 0;0 0 1]; P=c*[P zeros(3,1); zeros(1,3) .01];
for k=1:4; P=P+g1(:,k)*g1(:,k)'+.5*g2(:,k)*g2(:,k)'; end
Pxz=P*H'; Pzz=R+H*Pxz; L=Pxz/Pzz; z=H*M;
M=M+L*(y(:,kk)-z); P=P-L*Pxz'; MM=[MM M]; end
```

For our numerical test, we let the initial estimate

$$[\hat{x};\hat{y};\hat{z}] = [-.2;-.3;-.5] + .1 * randn(3,1), \quad \hat{\beta} = 3.$$

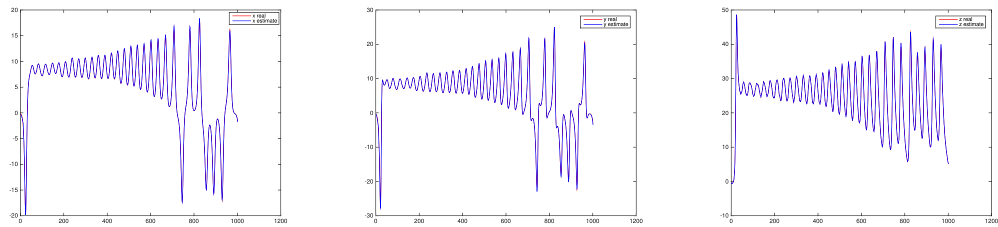
In Figure 6.1 we show the observation y on $[0, T = 10]$ with $\Delta t = .01$ and the estimated β. In Figure 6.2 we show the corresponding state estimate.



(a) Observations

(b) $\hat{\beta}$ estimate

Figure 6.1: Observation and β estimation in Lorenz equation.



(a) $\hat{x}$ estimate (b) $\hat{y}$ estimate (c) $\hat{z}$ estimate

Figure 6.2: states estimation in Lorenz equation.

We note that the Gaussian filter estimates accurately for the joint state-parameter estimation. as shown in Figures 6.1 and Figure 6.2.

6.2 Wave Equations

In the study of wave propagation it is very essential to estimate the medium (function) coefficient from surface observations and far field measurements. Consider the wave equation

$$\frac{\partial^2}{\partial t^2} u = c^2(x) \Delta u$$

where u is acoustic pressure for example and most importantly the parameter $c(x) > 0$ represents variable wave speed. We also consider

$$\frac{\partial^2}{\partial t^2} u = \Delta u - q(x) u$$

where $q(x)$ represents the potential.

6.2.1 Joint state and wave speed $c(x)$ estimations

We consider the wave equation

$$\frac{\partial^2}{\partial t^2} u = c^2(x) \Delta u \tag{6.2.1}$$

where the wave speed $c(x)$ is given by

$$a(x) = c(x)^2 = .9 + .1 \sin(2\pi x)$$

with initial condition

$$u_0(x) = \exp(-500 * (x - .2)^2) + \exp(-500 * (x - .7)^2),$$

which is the sumo of Gaussian centered at .2 and 7.

Details account for the time-space discretization of wave equations and tests for the Gaussian and the reduced Gaussian filter presented in Chapters 2 and 4. In this section we focus on the joint-estimation of the displacement u, velocity v and coefficient $c(x)$ for (6.2.1). We use the observation y_k of u at the two points $x = .3$ and $x = .8$ (no noise added) and initial estimate as $(u_0, 0)$. and $\hat{a}(x) = .9$.

Here is the matlab that shows the details of the reduced order Gaussian filter. Comparing $m = 50$ to $N = 297$.

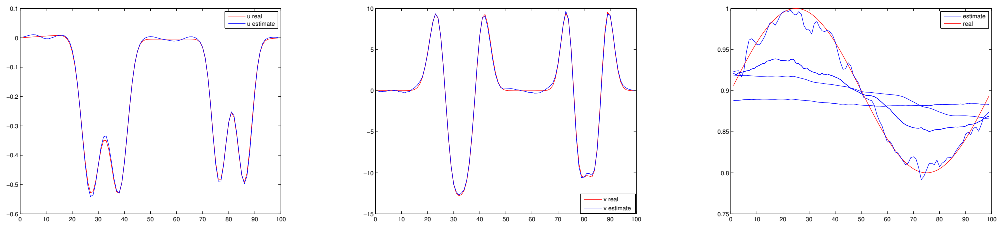
```
u=zeros(N,1); u(1:m)=u0; u(ip)=.9;q=50;
P=1.e-4*eye(N); s=rand(m); s=s+s';P(ip,ip)=eye(m)+.05*s; W=[];
for kk=1:100; P=P+1.e-5*eye(N);
[U S V]=svd(P); S=U*sqrt(S);
J=zeros(N); J(1:m,m+[1:m])=speye(m);
J(m+[1:m],1:m)=-spdiags(u(ip),0,m,m)*h0;
J(m+[1:m],2*m+1:end)=-spdiags(h0*u(1:m),0,m,m);
J(1:m,:)=J(1:m,:)+dt*J(m+[1:m],:); JJ=speye(N)+dt*J; g1=JJ*S;
tmp=-dt*u(ip).*(h0*u(1:m));u(m+[1:m])=u(m+[1:m])+tmp;
u(1:m)=u(1:m)+dt*u(m+[1:m]);
P=0*speye(N); P(ip,ip)=1.e-3*dt*speye(m); P=P+g1*g1';
Pxz=P*C'; Pzz=R+C*Pxz; L=Pxz/Pzz; z=C*u;
u=u+L*(y(:,kk)-z); u(ip)=(speye(m)+al*hh)\u(ip); P=P-L*Pxz'; W=[W u]; end
```

For this numerical test we use $\Delta x = .01$ and $\Delta t = .01$ with CFL number $\lambda = 1$ and $T = 1$. Recall we must use the regularization step for the parameter estimate $c(x)^2 = a(x)$ as

$$a^n \to (I + \beta H)^{-1} a^n.$$

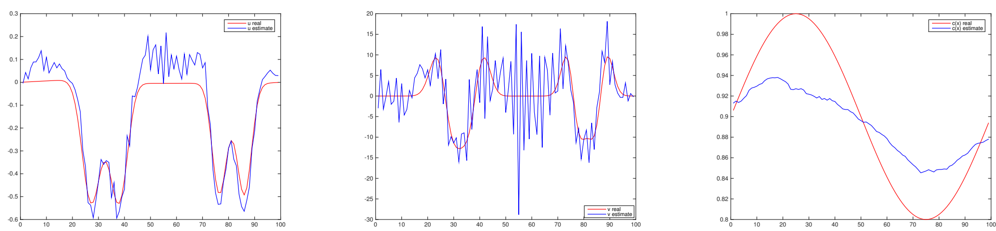
where we adjust $\beta > 0$ and $\beta = .001$ for our numerical tests.

In Figure 6.3, we show the Gaussian filter (full rank $N = 297$) results for ($\hat{u}(T)$, $\hat{v}(T)$) and $\hat{a}(T)$ estimates. We iterate our filter algorithm four times and the final estimates (blue) is close to the truth $a(x)$. Figure 6.4, shows the estimates for the first iterate (with $\beta = 10^{-3}$) and second iterate (with $\beta = 10^{-5}$). of our reduced order Gaussian filter with order $m = 20$ (compared to full rank $N = 297$).

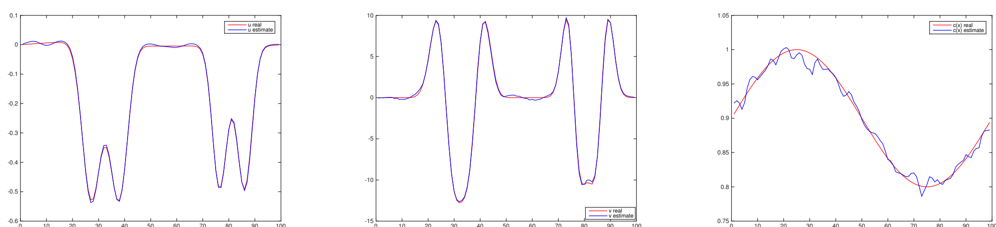


(a) $\hat{u}$, $\beta = 10^{-5}$ (b) $\hat{v}$, $\beta = 10^{-5}$ (c) $\hat{c}(x)$ $\beta = 10^{-5}$

Figure 6.3: joint estimation of u, v and $c(x)$ using Gaussian Filter, $N = 297$



(a) $\hat{u}$, $\beta = 10^{-3}$ (b) $\hat{v}$, $\beta = 10^{-3}$ (c) $\hat{c}(x)$, $\beta = 10^{-3}$



(d) $\hat{u}$, $\beta = 10^{-5}$ (e) $\hat{u}$, $\beta = 10^{-5}$ (f) $\hat{c}(x)$, $\beta = 10^{-5}$

Figure 6.4: estimation of u, v and $c(x)$ using reduced order filter with $m = 20$

Based on our tests we conclude that the reduced order Gaussian filter is very effective for the joint estimation of the state and parameter for (6.2.1), with tuned the regularization constants $\beta > 0$ and order $m = 20$.

6.2.2 Joint states and potential $q(x)$ estimations

We consider

$$\frac{\partial^2}{\partial t^2}u = \Delta u - q(x)u \tag{6.2.2}$$

where the parameter

$$a(x) = q(x) = .9 + .1\sin(2\pi x)$$

represents the potential and the initial condition is

$$u_0(x) = \exp(-500 * (x - .2)^2) + exp(-500 * (x - .7)^2).$$

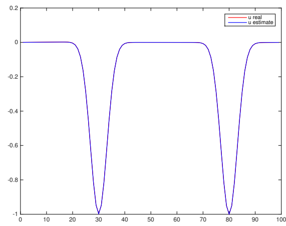
Again, detailed account for the time-space discretization of wave equations and tests for the Gaussian and the reduced Gaussian filter are presented in Chapters 2 and 4. In this section we focus on the joint-estimation of the displacement u, velocity v and coefficient $q(x)$. We use the observation y_k of u at the two points $x = .3$ and $x = .8$ (no noise added) and initial estimate as $(u_0, 0)$. and $\hat{q}(x) = .5$.

The numerical test is performed in exactly the same manner as in Section 6.2. The regularization step here for $q(x) = a(x)$ is

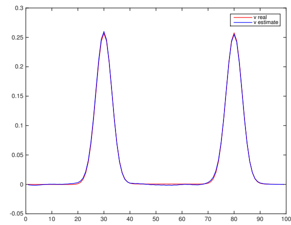
$$a^n \to (I + \beta H)^{-1} a^n.$$

Here is the matlab that shows the details of the reduced order Gaussian filter for $m = 30$.

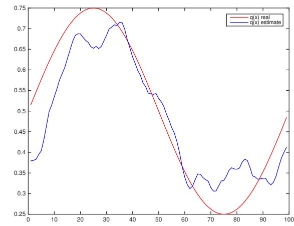
```
u=zeros(N,1); u(1:m)=u0; u(ip)=.5; q = 50;
P=1.e-4*eye(N); s=rand(m); s=s+s';P(ip,ip)=eye(m)+.05*s; W=[];
for kk=1:100; P=P+1.e-5*eye(N);
[U S V]=svd(P); S=U(:,1:q)*sqrt(S(1:q,1:q));
J=zeros(N); J(1:m,m+[1:m])=speye(m);
J(m+[1:m],1:m)=-h0-spdiags(u(ip),0,m,m);
J(m+[1:m],2*m+1:end)=-spdiags(u(1:m),0,m,m);
J(1:m,:)=J(1:m,:)+dt*J(m+[1:m],:); JJ=speye(N)+dt*J; g1=JJ*S;
uu=[0;u(1:m);0]; tmp=dt*(uu(3:n+1)-2*uu(2:n)+uu(1:m))/dx^2-dt*u(ip).*u(1:m);
u(1:m)=u(1:m)+dt*u(m+[1:m])+dt*tmp;
u(m+[1:m])=u(m+[1:m])+tmp;
P=0*speye(N); P(ip,ip)=dt*speye(m); P=P+g1*g1';
Pxz=P*C'; Pzz=R+C*Pxz; L=Pxz/Pzz; z=C*u;
u=u+L*(y(:,kk)-z); u(ip)=(speye(m)+al*hh)\u(ip); P=P-L*Pxz'; W=[W u]; end
```



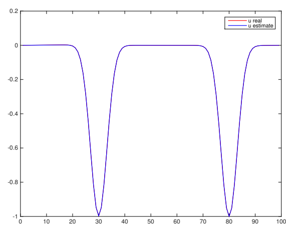
(a) $\hat{u}$, $\beta = 5.10^{-5}$



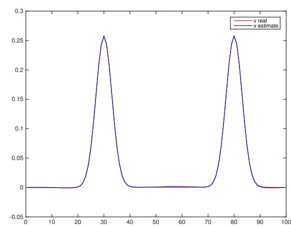
(b) $\hat{v}$, $\beta = 5.10^{-5}$



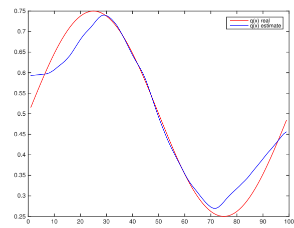
(c) $\hat{q}(x)$, $\beta = 5.10^{-5}$



(d) $\hat{u}$, $\beta = 2.10^{-5}$



(e) $\hat{v}$, $\beta = 2.10^{-5}$



(f) $\hat{q}(x)$, $\beta = 2.10^{-5}$

Figure 6.5: joint estimation of u, v and $q(x)$ using reduced order filter with $m = 30$

In this test for the wave equations with potential term (6.2.2) we also iterate the reduced order Gaussian order with tuned β shown in Figure 6.5 with order $m = 30$. The sates estimate are very accurate in this case since we have the exact initial condition for the filter and the potential term is of a much lower oder than the wave speed case (6.2.1). In summary we again demonstrates the effectiveness of the reduced order Gaussian filter for the joint state-parameter estimation in wave equation models.

Chapter 7

Conclusion

We have developed and analyzed very effective filtering and smoothing algorithms. Especially for a large scale dynamics and PDEs governed systems. The reduced order method enhances the performance of our proposed Gaussian filter in addition to being an efficient algorithm. The Lorenz 96 model and the wave equation models we tested for the Gaussian and reduced order Gaussian filters. For dissipative systems, the alternative method we developed to the reduced order filter is an assumed covariance filter. For system that are time reversible, we developed the time reversal filter which used the forward and backward smoothing algorithm. The time reversal filter algorithm we tested for conservation law and the advection equation. The Quasi reversal method is used for system with mild diffusion like the conduction-advection equation which we tested. For the joint estimate of state and parameters, we tested the Lorenz equation and the wave equations models. In the future we will test on more real world examples like the Navier-Stokes, the 3-dimensional wave propagation.

Bibliography

[1] S. M. Alekseeva, N. I. Yurchuk, *The quasi-reversibility method for the problem of the control of an initial condition for the heat equation with an integral boundary condition,* Differential Equations 34 (1998), no. 4, 493-500.

[2] D. L. Alspace and H. W. Sorenson, "Nonlinear Bayesian estimation using Gaussian sum approximation," *IEEE Trans. Automat. Contr.,* vol.17, pp. 439-448, 1972.

[3] M. Athans, R.P. Wishner and A. Bertolini, *Suboptimal state estimation for continuous-time nonlinear systems for discrete time measurements*, 1968.

[4] Y. Bar-Shalom and X. R. Liu, *Estimation and Tracking: Principles, Techniques, and Software.* New York: Artech, 1993.

[5] A. Bensoussan, R.Golwinski and A. Rascanu, *Approximation of the Zakai equation by the splitting up method*, 1990.

[6] G. Clark and C. Oppenheimer, *Quasireversibility Methods for Non-Well-Posed Problem,* Electronic Journal of Differential Equations, Vol. 1994 (1994), no. 08, 1-9.

[7] L. C. Evans, *Partial Differential Equations,* Amer. Math. Soc., Providence, RI, 1993.

[8] C. P. Fang, "New algorithms of Gaussian assumed density filter and a convergence result," *IEEE Trans. Automat. Contr.*

[9] M. Fink, *Time reversed acoustics,* Physics Today, 50(3) (1997), pp. 34-40.

[10] M. Fujisaki, G. Kallianpur, and H. Kunita, Stochastic differential equations for the nonlinear filtering problem, Osaka J. Math., 9 (1972), pp. 19-44.

[11] K. Itô, On stochastic differential equations. *Memoirs of the American Mathematical Society* 4 (1951), 1-51.

[12] K. Ito and B. Rozovskii, Approximation of the Kushner Equation for Nonlinear filtering, *SIAM J. CONTROL OPTIM* Vol. 38, No. 3, pp. 893-915.

[13] K. Ito and K. Xiong,*Gaussian Filter for Nonlinear Filtering Problems*, 2000.

[14] K. Ito, "Approximation of the Zakai equation for nonlinear filtering," *SIAM J. Contr. Optim.,* vol. 34, pp. 620-634, 1996.

[15] A. H. Jazwinski, *Stochastic Process and Filtering Theory*, New York: Academic, 1970.

[16] S. J. Julier and J. K. Uhlmann. (1994) *A general method for approximating nonlinear transformations of probability distributions.* [Online]. Available: www http://phoebe.robots.ox.ac.uk/default.htm Proc. Amer. Contr. Conf., Seattle, WA, June 1995, pp. 1628-1632

[17] S. J. Julier, J. K. Uhlmann, and H. F. Durrant-Whyte, "A new approach for filtering nonlinear systems," in Proc. Amer. Contr. Conf

[18] T. Kailath, *A note on least squares estimation by the innovations method,* this Journal, 10 (1972), pp. 477-486.

[19] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME, J. Basic Eng.,* vol. 82D, pp. 35-45, Mar. 1960.

[20] Krishnamurti, T. N. (January 1995). "Numerical Weather Prediction". *Annual Review of Fluid Mechanics* 27 (1): 195-225.

[21] H. J. Kushner, "Approximation to optimal nonlinear filters," *IEEE Trans. Automat. Contr.,* vol. 12, pp. 546-556, 1967.

[22] H. J. Kushner, Probability Methods for Approximations in Stochastic Control and for Elliptic Equations, Academic Press, New York, 1977.

[23] R. Lattes and J. L. Lion, *Méthode de Quasi-Reversibilité et Applications,* Dunod, Paris, 1967.

[24] P.D Lax and B. Wendroff (1960). "Systems of conservation laws". *Commun. Pure Appl Math.* 13 (2): 217-237.

[25] R. Sh. Lipster and A. N. Shiryayev, The Statistics of Random Process, I, II, Springer-Verlag, Berlin, 1977.

[26] Lorenz, Edward Norton (1963). "Deterministic nonperiodic flow". *Journal of the Atmospheric Sciences* 20 (2): 130-141.

[27] Lynch, Peter (2006). "The Fundamental Equations". *The Emergence of Numerical Weather Prediction. Cambridge University Press.* pp. 45-46. ISBN 978-0-521-85729-1.

[28] N. S. Mera, "The Method of Fundamental Solutions for the Backward Heat Conduction Problem" *Inverse Problems in Engineering,* Vol. 13, No. 1, 2005, pp. 79-98.

[29] R. D. Richtmyer, *A Survey of Difference Methods for Non-Steady Fluid Dynamics,* N.C.A.R. Tech. Notes 63-2, 1963.

[30] B. L. Rozovskii, *Stochastic Evolution Systems, Linear Theory and Application to Nonlinear Filtering.* Norwell, MA: Kluwer, 1991.

[31] Jerome Sacks, William J. Welch, Toby J. Mitchell and Henry P. Wynn, *Design and Analysis of Computer Experiments, Statistical Science,* Vol. 4, No. 4 (Nov., 1989), pp. 409-423

[32] A.Y. Veretennikov, N.V. Krylov, On explicit formulas for solutions of stochastic equations. *Math. USSR Sb.* 29(2) (1976), 239-256.

[33] A.Y. Veretennikov, On the criteria for existence of a strong solutions to a stochastic equation. *Theory Probab. Appl. 27* 441-449 (1982).

[34] C. B. Vreugdenhil, Numerical Methods for Shallow Water Flow, Boston: Kluwer Academic Publishers (1994)

[35] Yosida, K. (1965). Functional Analysis. Academic Press.

[36] Zakai, Moshe (1969). "On the optimal filtering of diffusion processes". Probability Theory and Related Fields 11 (3): 230-243.