PARAMETER ESTIMATION FOR PARTIAL DIFFERENTIAL EQUATIONS USING STOCHASTIC METHODS

by

Roxana Elena Tanase

B.S. in Mathematics and Computer Science, University of Bucharest, 2005

M.S. in Mathematics, University of Bucharest, 2007

M.S. in Mathematics, University of Cincinnati, 2009

Submitted to the Graduate Faculty of
the Kenneth P. Dietrich School of Arts and Sciences in partial fulfillment
of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2016

UNIVERSITY OF PITTSBURGH

THE KENNETH P. DIETRICH SCHOOL OF ARTS AND SCIENCES

This dissertation was presented

by

Roxana Elena Tanase

It was defended on

April 23, 2016

and approved by

Professor Ivan Yotov, Department of Mathematics

Professor Catalin Trenchea, Department of Mathematics

Professor David Swigon, Department of Mathematics

Professor Gilles Clermont, Critical Care Medicine

Dissertation Advisors: Professor Ivan Yotov, Department of Mathematics,

Professor Catalin Trenchea, Department of Mathematics

PARAMETER ESTIMATION FOR PARTIAL DIFFERENTIAL EQUATIONS USING STOCHASTIC METHODS

Roxana Elena Tanase, PhD

University of Pittsburgh, 2016

The aim of this thesis is to compare the efficiency of different algorithms on estimating parameters that arise in partial differential equations: Kalman Filters (Ensemble Kalman Filter, Stochastic Collocation Kalman Filter, Karhunen-Loève Ensemble Kalman Filter, Karhunen-Loève Stochastic Collocation Kalman Filter), Markov-Chain Monte Carlo sampling schemes and Adjoint variable-based method.

We also present the theoretical results for stochastic optimal control for problems constrained by partial differential equations with random input data in a mixed finite element form. We verify experimentally with numerical simulations using Adjoint variable-based method with various identification objectives that either minimize the expectation of a tracking cost functional or minimize the difference of desired statistical quantities in the appropriate L^p norm.

Keywords: parameter estimation, Kalman Filter, Stochastic Collocation, Markov Chain Monte Carlo, Adjoint variable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 INTRODUCTION

Driven by the needs from applications both in industry and other sciences, the field of inverse problems [1, 37] has undergone a tremendous growth within the last two decades, where recent emphasis has been laid more than before on nonlinear problems. Advances in this field and the development of sophisticated numerical techniques for treating the direct problems allow to address and solve industrial inverse problems on a level of high complexity [20, 21, 22].

Parameter estimation is an important field in the area of modeling physical or biological processes. The set of parameters that maximize the model's agreement with experimental data, i.e. the ideal parameter set, can be used to yield important insight into a given system. It can help scientists more clearly describe the behavior of the system, predict behavioral changes in the system during pathological situations, and assess the efficacy of various corrective options [44]. In addition, once those ideal parameters have been found, other mathematical techniques can be used to obtain further insight into the system's behavior. Local sensitivity analysis [10] at the optimal parameter set can be used to assess the local importance of the parameters. Also, the ideal parameter set can be used as a starting point for obtaining (via, e.g., Markov-Chain Monte Carlo methods [37]) distributions of the parameters that produce computational estimates that agree reasonably well with experiment. These distributions can be used to assess the global importance of each parameter.

As mathematical/computational models become more complex in order to better describe physical systems, parameter estimation can grow in difficulty and cost due to the increase in number of parameters and consequently computational runtime. The problem of calibrating a model in a reasonable amount of time depends more and more on efficient methods of parameter estimation.

The research for this thesis is mainly focused on estimating parameters that arise in partial differential equations using different techniques: **Kalman Filters** [46] (Ensemble Kalman Filter [16], Stochastic Collocation Kalman Filter, Karhunen-Loève Ensemble Kalman Filter, Karhunen-Loève Stochastic Collocation Kalman Filter [43]), **Markov-Chain Monte Carlo** method [19] and **Adjoint variable-based** method [8].

The **Kalman Filter** is a two step process that evolves the state and uncertainty/variance associated with a system optimally by using experimental data corresponding to that system. The first step (predict or forecast step)

$$\mathbf{x}_n^f = \mathbf{f}(\mathbf{x}_{n-1}^a) + \mathbf{w}_n$$

uses a computational model and the uncertainty associated with that model to evolve both the system's mean and variance to the next time step at which experimental data is available. The second step (the analysis or assimilation step)

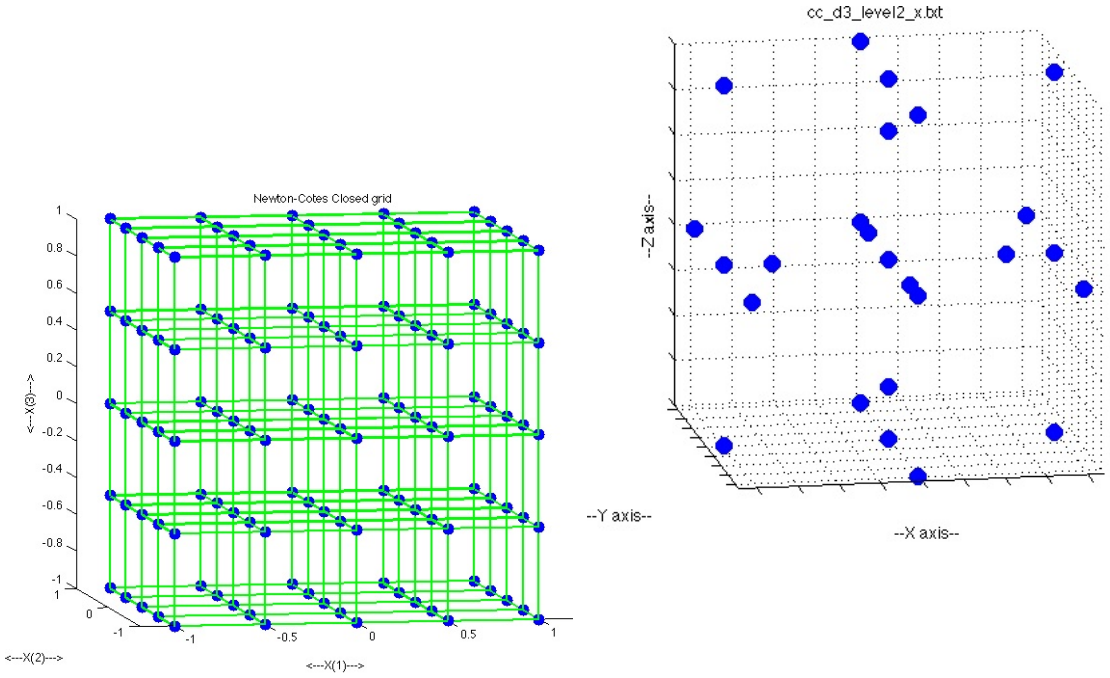
$$\mathbf{x}_n^a = \mathbf{f}(\mathbf{x}_n^f) + K_n(\mathbf{y}_n - \mathbf{h}(\mathbf{x}_n^f))$$

uses experimental data and the uncertainty associated with the experiments (measurement error) to adjust the variable means and variances to more closely agree with the experimental data.

The Ensemble Kalman Filter(EnKF) tracks the underlying distributions of the state variables and measurements by representing the underlying distributions using an ensemble of size q randomly chosen samples for state and measurement vectors and advancing those distributions over time by advancing each member of the ensemble independently.

For the **Stochastic Collocation Kalman Filter(SCKF)**, probabilistic discretization is done by collocating the solution on a particular set of points strategically chosen from the underlying stochastic space and then connect the realizations with suitable interpolatory basis (Lagrangian). We use sparse grids constructed by the Smolyak algorithm based on one-dimensional polynomial interpolation at the extrema of the Hermite polynomials (i.e. Gaussian abscissas)[32]. In most cases, Smolyak can match the precision of the product rule while avoiding the crushing explosion in function evaluations (see Figure 1.1). Discretizations on sparse grids involve only $O(N(\log N)^{d-1})$ degrees of freedom, where d is the problem

dimension and N denotes the number of degrees of freedom in one coordinate direction. The accuracy obtained this way is comparable to the one using a full tensor product basis involving $O(N^d)$ degrees of freedom, if the underlying problem has smooth (analytic) dependence on the random variables. This way, the curse of dimensionality, i.e. the exponential dependence of conventional approaches on the dimension d, can be overcome to a certain extent.



(a) (b)

Figure 1.1: (a)5x5x5 tensor product rule, (b)sparse grid with 25 collocation points constructed by Smolyak algorithm (see http://people.sc.fsu.edu/~jburkardt/presentations/siam_uq_2012_part3.pdf)

Markov-Chain Monte Carlo (MCMC) is a family of algorithms for modeling uncertainty. Since calculation of posterior model probabilities is rarely achievable in closed form for realistic models, approximation methods may be used. MCMC technique produces a Markov chain $(\theta_n)_{n\geq 0}$ which has equilibrium distribution that matches the one of the posterior probability distribution. The state of the chain after a number of steps is then used as a sample of the desired posterior distribution. The quality of the sample improves as a function of the number of steps. Popular examples of MCMC methods include Gibbs sam-

pling, Metropolis-Hastings algorithm, slice sampling, Hamiltonian Monte Carlo, and many others.

Adjoint variable-based method solves a large class of optimization, inverse, and control problems. It is one of the gradient-based techniques in which gradient vector of the cost functional with respect to the unknown parameters is calculated indirectly by solving an adjoint equation. Although an additional cost arises from solving the adjoint equation, the gradients of the cost functional can be altogether achieved with respect to each parameter. Thus, the total cost to obtain these gradients is independent of the number of parameters and amounts to the cost of solving two partial differential equations (PDEs) roughly. From a control theory point of view, the algorithm is based on the Pontryagin maximum principle, since it tries to iteratively solve the necessary conditions for optimality. From an optimization point of view, the algorithm consists of a gradient descent, in which the gradient of the cost functional is efficiently computed via the adjoint variable-based method.

2.0 KALMAN FILTER AND MARKOV CHAIN MONTE CARLO METHODS

2.1 DERIVING THE KALMAN FILTER

The Kalman filter [5] is a two step process that evolves the state and uncertainty/variance associated with a system optimally by using experimental data corresponding to that system. The first step (predict or forecast step) uses a computational model and the uncertainty associated with that model to evolve both the system's mean and variance to the next time step at which experimental data is available. The second step (the analysis or assimilation step) uses experimental data and the uncertainty associated with the experiments (measurement error) to adjust the variable means and variances to more closely agree with the experimental data.

The true system. To help the explanation process, we consider a particular physical system, an epithelial layer with a small hole in it that closes as time progresses. By taking multiple epithelial layers (on a petri dish), making random holes of different shapes but approximately the same size (one hole per dish), and watching the system close as time progresses, we can make estimates of the average behavior of such systems and the variance on those behaviors. Those behaviors could be qualitative or quantitative in nature. Taking an infinite number of such experiments there would eventually be a "true" average state and a corresponding "true" covariance function. There are two ways to investigate such a system. One can use experiment or a mathematical model. The Kalman Filter uses both of these to attempt to create the best estimate of the "true" system.

Computational Model. Stepping back to generality, assume that we have a mathematical model of a particular system. In order to use Kalman filter techniques, we discretize

the system temporally and, if the system has a PDE or a system of PDEs associated with it, spatially. The discretized system can then be solved using a suitable computational technique. The system and accompanying solution technique is referred to as the computational model of the system. This computational model can be used to evolve the discretized system to a later time. The new discretized system can then be used to reconstruct information as needed. We refer to such information as computational information.

Experimental Data. For the experimental data, discretized data usually comes in the form of measurements at various points in the domain or integrals of state variables over a set region of space. In our case, a two-dimensional grid of computational cells is overlaid on the pictures of the epithelial cells and corresponding hole. If a particular computational cell in the grid lies within the hole, the value associated with it is zero. If a particular computational cell in the grid lies outside of the hole, the value associated with it is one. If part of the computational cell lies inside the hole while the other part lies outside the hole, the value assigned to the cell is equal to one minus the fraction of the cell that lies within the wound. Hence, in general, the experimental data collected in this manner represents the average amount of epithelial cells in a given region/computational cell.

Computational Error. Given perfect data at a particular time, this computational model can be used to evolve the system to a later time. There are inherent errors that arise from using the computational model. These errors include normal spatial and temporal discretization errors as well as model errors that arise because the mathematical model does not perfectly describe the underlying physical or biological system. In biological systems, mathematical model errors are probably responsible for most of the computational error. In the Kalman filter, these computational errors are assumed to be stochastic and are denoted by $\mathbf{w}_n$, where the subscript n corresponds to the discrete time $t = t_n$. The covariance matrix associated with this noise is denoted by $Q_n =< \mathbf{w}_n, \mathbf{w}_n >$.

Cumulative Error. During the course of the model evolution, errors from previous time steps accumulate in the current estimate of the solution. The Kalman filter solution process helps mitigate this error accumulation process and, depending on the amount of error in the data, the cumulative error may even decrease as time goes on.

Experimental Error. Another source of error during the Kalman filter process is the

error coming from experiments corresponding to the system in consideration. These errors come from measurement error and the error that comes from the natural variations that arise between experiments. In biological systems, these latter errors are probably responsible for most of the experimental error. In the Kalman filter, these experimental errors are assumed to be stochastic and are denoted by $\mathbf{v}_n$. The covariance matrix associated with this noise is denoted by $R_n =< \mathbf{v}_n, \mathbf{v}_n >$.

System before Kalman Filtering a given step. Given a particular time step, the following equations describe the relationships between the computational and true system and between the experimental measurements and the true system:

$$\mathbf{x}_n = \tilde{\mathbf{x}}_n + \mathbf{u}_n, \mathbf{y}_n = \tilde{\mathbf{y}}_n + \mathbf{v}_n.$$

The vector $\mathbf{x}$ corresponds to the computational system and $\mathbf{y}$ corresponds to the experimental measurements. $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ are vectors that correspond to the true systems.

The vectors $\mathbf{x}$ and $\tilde{\mathbf{x}}$ include the values of the state variables at the appropriate locations in the computational grid and any unknown parameters (which may be defined as globally constant or may vary with respect to space). The values of the state variables may correspond to values of the state variables at a specific point (appropriate for finite differences) or the average value of the state variable over an entire computational cell (appropriate for finite volume).

The vectors $\mathbf{y}$ and $\tilde{\mathbf{y}}$ include the values of the state variables at the appropriate locations in the experimental setup. Again, these values may correspond to point specific measurements or be averaged over a specified region or be some other descriptor of the system.

Initial states. The initial noises or errors $\mathbf{u}_0$ and $\mathbf{v}_0$ are sometimes assumed to be zero corresponding to initial conditions and experimental data that agrees perfectly with the true system. This is a suitable assumption if the initial state of the true system is known very well.

For now, however, we assume that $\mathbf{x}_0 = \mathbf{y}_0$ and that $\mathbf{u}_0$ and $\mathbf{v}_0$ are nonzero. This is not an assumption critical for any derivation, but it is nonetheless used. The errors $\mathbf{u}_0$ and $\mathbf{v}_0$ are assumed to be stochastic and to have nonzero covariance matrices given by: $P_{xx,0} =< \mathbf{u}_0, \mathbf{u}_0 >$ and $R_0 =< \mathbf{v}_0, \mathbf{v}_0 >$.

Prediction and analysis. There are two steps in the Kalman Filter, prediction and analysis.

Prediction. In the prediction or forecast step, one can write the step as:

$$\mathbf{x}_n^f = \mathbf{f}(\mathbf{x}_{n-1}^a). \tag{2.1.1}$$

Here $\mathbf{x}_n^f$ and $\mathbf{x}_{n-1}^a \in \mathbf{R}^m$ are vectors of the state variables and parameters for the given system (state vectors) from the n^{th} forecast and $n-1^{st}$ assimilation time steps, respectively, and $\mathbf{f}$ is a forecasting model that is used to evolve the state vector in time. This is the computational prediction for the values in the state vector/values of the state variables and parameters at the next time step given the best estimate using both computational model and experimental data from the previous time step, $\mathbf{x}_{n-1}^a$. The vector $\mathbf{x}_{n-1}^a$ is the adjusted state vector after data has been taken into account. This adjustment will be considered later.

In reality, if we had the true state, $\tilde{\mathbf{x}}_{n-1}$, at a previous time step, then the true state at the next time step would be given by:

$$\tilde{\mathbf{x}}_n = \mathbf{f}(\tilde{\mathbf{x}}_{n-1}) + \mathbf{w}_n. \tag{2.1.2}$$

If we had the true state, the model would predict an accurate true state at the next time step plus or minus the computational error term. As a reminder, we are assuming this error is stochastic in nature and is white gaussian noise. One can now use (2.1.1) and (2.1.2) to estimate the covariance matrix at the next time step:

$$\begin{aligned}
P_{n,xx}^f &= < \mathbf{x}_n^f - \tilde{\mathbf{x}}_n, \mathbf{x}_n^f - \tilde{\mathbf{x}}_n > \\
&= < \mathbf{f}(\mathbf{x}_{n-1}^a) - \mathbf{f}(\tilde{\mathbf{x}}_{n-1}) - \mathbf{w}_n, \mathbf{f}(\mathbf{x}_{n-1}^a) - \mathbf{f}(\tilde{\mathbf{x}}_{n-1}) - \mathbf{w}_n > \\
&= < \mathbf{f}(\mathbf{x}_{n-1}^a) - \mathbf{f}(\tilde{\mathbf{x}}_{n-1}), \mathbf{f}(\mathbf{x}_{n-1}^a) - \mathbf{f}(\tilde{\mathbf{x}}_{n-1}) > - \\
&\quad < \mathbf{w}_n, \mathbf{f}(\mathbf{x}_{n-1}^a) - \mathbf{f}(\tilde{\mathbf{x}}_{n-1}) > - < \mathbf{f}(\mathbf{x}_{n-1}^a) - \mathbf{f}(\tilde{\mathbf{x}}_{n-1}), \mathbf{w}_n > + \\
&\quad < \mathbf{w}_n, \mathbf{w}_n > \\
&= < \mathbf{f}(\mathbf{x}_{n-1}^a) - \mathbf{f}(\tilde{\mathbf{x}}_{n-1}), \mathbf{f}(\mathbf{x}_{n-1}^a) - \mathbf{f}(\tilde{\mathbf{x}}_{n-1}) > + < \mathbf{w}_n, \mathbf{w}_n > \\
&= < \mathbf{f}(\mathbf{x}_{n-1}^a) - \mathbf{f}(\tilde{\mathbf{x}}_{n-1}), \mathbf{f}(\mathbf{x}_{n-1}^a) - \mathbf{f}(\tilde{\mathbf{x}}_{n-1}) > + Q_n.
\end{aligned}$$

Here we used the fact that $< \mathbf{w}_n, \mathbf{f}(\mathbf{x}^a_{n-1}) - \mathbf{f}(\tilde{\mathbf{x}}^a_{n-1}) >$ is zero because $\mathbf{w}_n$ is symmetrically distributed about zero (assumed here) and the difference $\mathbf{f}(\mathbf{x}^a_{n-1}) - \mathbf{f}(\tilde{\mathbf{x}}^a_{n-1})$ and noise $\mathbf{w}_n$ are independently distributed.

Temporarily we define: $P_{n,ff} =< \mathbf{f}(\mathbf{x}_n) - \mathbf{f}(\tilde{\mathbf{x}}_n), \mathbf{f}(\mathbf{x}_n) - \mathbf{f}(\tilde{\mathbf{x}}_n) >$. It turns out this is the crucial quantity in the prediction step. If $\mathbf{f}(\mathbf{x}) = A\mathbf{x}$, then the linear Kalman filter can be used and the corresponding predicted covariance matrix becomes:

$$\begin{aligned} P^f_{n,xx} &= < \mathbf{f}(\mathbf{x}^a_{n-1}) - \mathbf{f}(\tilde{\mathbf{x}}_{n-1}), \mathbf{f}(\mathbf{x}^a_{n-1}) - \mathbf{f}(\tilde{\mathbf{x}}_{n-1}) > + Q_n \\ &= < A\mathbf{x}^a_{n-1} - A\tilde{\mathbf{x}}_{n-1}, A\mathbf{x}^a_{n-1} - A\tilde{\mathbf{x}}_{n-1} > + Q_n \\ &= A < \mathbf{x}^a_{n-1} - \tilde{\mathbf{x}}_{n-1}, \mathbf{x}^a_{n-1} - \tilde{\mathbf{x}}_{n-1} > A^T + Q_n \\ &= A P^a_{n-1,xx} A^T + Q_n. \end{aligned}$$

Here we have defined the covariance matrix for the adjusted state vector at time t_n as $P^a_{n,xx} =< \mathbf{x}^a_n - \tilde{\mathbf{x}}_n, \mathbf{x}^a_n - \tilde{\mathbf{x}}_n >$. In addition, it can be seen that Q_n represents the computational error while $AP^a_{n-1,xx}A^T$ represents the cumulative error (which includes both computational and experimental errors from times past).

If $\mathbf{f}(\mathbf{x})$ is a nonlinear function, then one can perform a Taylor expansion in order to estimate the value of $P_{n,ff}$. If one neglects higher order nonlinear terms (which are "neglectable" if, for instance, $\|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|^2 << \|\mathbf{x}_n - \tilde{\mathbf{x}}_n\|$), then one comes up with the following formula for the forecast covariance matrix:

$$P^f_{n,xx} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_n} P^a_{n-1,xx} \left(\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_n} \right)^T + Q_n,$$

where it can be seen that the Jacobian (tangent linear operator) takes the place of the A in the linear Kalman Filter. This choice for $P^f_{n,xx}$ is made in the extended Kalman Filter.

Adjustment. After the prediction step there is a forecast mean, $\mathbf{x}^f_n$, and a forecast covariance estimate, $P^f_{n,xx}$. It is necessary to compare computational and experimental data. In order to do so, a measurement function is used and is defined by:

$$\tilde{\mathbf{y}} = \mathbf{h}(\tilde{\mathbf{x}}).$$

Using this function, we then define the adjusted mean, $\mathbf{x}_n^a$, as a linear combination of the predicted mean and the difference between the measured experimental data, $\mathbf{y}_n$ and what the experimental data would have been if the predicted state vector really did correspond to the state of the given system, $\mathbf{h}(\mathbf{x}_n^f)$:

$$\mathbf{x}_n^a = \mathbf{x}_n^f + K_n(\mathbf{y}_n - \mathbf{h}(\mathbf{x}_n^f)). \tag{2.1.3}$$

K_n is the Kalman gain and is a blending factor of sorts which blends the model information with the measurement information.

Kalman Gain We have not yet derived the specific form of this factor. The Kalman gain is chosen to minimize the amount of uncertainty in the new estimate of the state vector for the system, $\mathbf{x}_n^a$, and depends on the covariances of the forecast state vectors and measurement values. In order to derive this factor, we desire that the new adjusted covariance,

$$P_{n,xx}^a = < \mathbf{x}_n^a - \tilde{\mathbf{x}}_n, \mathbf{x}_n^a - \tilde{\mathbf{x}}_n >,$$

be optimal in some sense. To do so, first consider the covariance matrix associated with these adjusted state vectors:

$$\begin{aligned}
P_{n,xx}^a &= < \mathbf{x}_n^f + K_n(\mathbf{y}_n - \mathbf{h}(\mathbf{x}_n^f)) - \tilde{\mathbf{x}}_n, \mathbf{x}_n^f + K_n(\mathbf{y}_n - \mathbf{h}(\mathbf{x}_n^f)) - \tilde{\mathbf{x}}_n > \\
&= < \mathbf{x}_n^f - \tilde{\mathbf{x}}_n, \mathbf{x}_n^f - \tilde{\mathbf{x}}_n > + < K_n(\mathbf{y}_n - \mathbf{h}(\mathbf{x}_n^f)), \mathbf{x}_n^f - \tilde{\mathbf{x}}_n > + \\
&\quad < \mathbf{x}_n^f - \tilde{\mathbf{x}}_n, K_n(\mathbf{y}_n - \mathbf{h}(\mathbf{x}_n^f)) > + \\
&\quad < K_n(\mathbf{y}_n - \mathbf{h}(\mathbf{x}_n^f)), K_n(\mathbf{y}_n - \mathbf{h}(\mathbf{x}_n^f)) > \\
&= P_{n,xx}^f - K_n < \mathbf{h}(\mathbf{x}_n^f) - \mathbf{y}_n, \mathbf{x}_n^f - \tilde{\mathbf{x}}_n > - \\
&\quad < \mathbf{x}_n^f - \tilde{\mathbf{x}}_n, \mathbf{h}(\mathbf{x}_n^f) - \mathbf{y}_n > K_n^T + \\
&\quad K_n < \mathbf{h}(\mathbf{x}_n^f) - \mathbf{y}_n, \mathbf{h}(\mathbf{x}_n^f) - \mathbf{y}_n > K_n^T.
\end{aligned}$$

We define the additional covariance matrices appearing in the above equation as follows:

$$P_{n,yx}^f = (P_{n,xy}^f)^T = < \mathbf{h}(\mathbf{x}_n^f) - \mathbf{y}_n, \mathbf{x}_n^f - \tilde{\mathbf{x}}_n > \tag{2.1.4}$$

$$P_{n,yy}^f = < \mathbf{h}(\mathbf{x}_n^f) - \mathbf{y}_n, \mathbf{h}(\mathbf{x}_n^f) - \mathbf{y}_n > \tag{2.1.5}$$

and rewrite the equation for $P_{n,xx}^a$ as:

$$P^a_{n,xx} = P^f_{n,xx} - K_n P^f_{n,yx} - P^f_{n,xy} K_n^T + K_n P^f_{n,yy} K_n^T. \quad (2.1.6)$$

This is an expression for the analyzed covariance matrix that is quadratic in K_n. The diagonal terms correspond to the uncertainty/variance in each element of the analyzed state vector $\mathbf{x}^a_n$. We wish to minimize the sum of these uncertainties. To do so we set:

$$\frac{d(trace(P^a_{n,xx}))}{dK_n} = 0.$$

Using a couple of matrix derivative rules for traces:

$$\frac{d(tr(AB))}{dA} = B^T; \qquad \frac{d(tr(ABA^T))}{dA} = 2AB \text{ (B symmetric)} \quad (2.1.7)$$

and a simple trace rule, $tr(A^T) = tr(A)$, we can do the following:

$$\begin{aligned} tr(P^a_{n,xx}) &= tr(P^f_{n,xx}) - tr(K_n P^f_{n,yx}) - tr(P^f_{n,xy} K_n^T) + \\ &\quad tr(K_n P^f_{n,yy} K_n^T) \\ &= tr(P^f_{n,xx}) - 2tr(K_n P^f_{n,yx}) + tr(K_n P^f_{n,yy} K_n^T). \end{aligned}$$

Now we can take the derivative of both sides with respect to K_n:

$$\begin{aligned} \frac{d(tr(P^a_{n,xx}))}{dK_n} &= -2(P^f_{n,yx})^T + 2K_n P^f_{n,yy} = 0 \Rightarrow \\ P^f_{n,xy} &= K_n (P^f_{n,yy})^{-1}. \end{aligned} \quad (2.1.8)$$

Here we have used the fact that $(P^f_{n,yx})^T = P^f_{n,xy}$. Hence:

$$\begin{aligned} K_n P^f_{n,yy} &= P^f_{n,xy} \\ K_n &= P^f_{n,xy} (P^f_{n,yy})^{-1} \end{aligned} \quad (2.1.9)$$

It is important to note that the formulae for $P^f_{n,yy}$ and $P^f_{n,yx}$ have alternate representations in terms of just state vector quantities. We assume below that the data noise is distributed independently from the forecast state vectors about the true value. For the forecast data covariance matrix, the formula is:

$$
\begin{aligned}
P^f_{n,yy} &= < \mathbf{h}(\mathbf{x}^f_n) - \mathbf{y}_n, \mathbf{h}(\mathbf{x}^f_n) - \mathbf{y}_n > \\
&= < \mathbf{h}(\mathbf{x}^f_n) - (\tilde{\mathbf{y}}_n + \mathbf{v}_n), \mathbf{h}(\mathbf{x}^f_n) - (\tilde{\mathbf{y}}_n + \mathbf{v}_n) > \\
&= < \mathbf{h}(\mathbf{x}^f_n) - \tilde{\mathbf{y}}_n, \mathbf{h}(\mathbf{x}^f_n) - \tilde{\mathbf{y}}_n > - < \mathbf{v}_n, \mathbf{h}(\mathbf{x}^f_n) - (\tilde{\mathbf{y}}_n + \mathbf{v}_n) > \\
&\quad - < \mathbf{h}(\mathbf{x}^f_n) - \tilde{\mathbf{y}}_n, \mathbf{v}_n > + < \mathbf{v}_n, \mathbf{v}_n > \\
&= < \mathbf{h}(\mathbf{x}^f_n) - \mathbf{h}(\tilde{\mathbf{x}}_n), \mathbf{h}(\mathbf{x}^f_n) - \mathbf{h}(\tilde{\mathbf{x}}_n) > + R_n. \qquad (2.1.10)
\end{aligned}
$$

For the covariance between the model and data, we have the formula:

$$
\begin{aligned}
P^f_{n,xy} &= < \mathbf{x}^f_n - \tilde{\mathbf{x}}_n, \mathbf{h}(\mathbf{x}_n) - \mathbf{y}_n > \\
&= < \mathbf{x}^f_n - \tilde{\mathbf{x}}_n, \mathbf{h}(\mathbf{x}_n) - (\tilde{\mathbf{y}}_n + \mathbf{v}_n) > \\
&= < \mathbf{x}^f_n - \tilde{\mathbf{x}}_n, \mathbf{h}(\mathbf{x}_n) - \tilde{\mathbf{y}}_n > + < \mathbf{x}^f_n - \tilde{\mathbf{x}}_n, \mathbf{v}_n > \\
&= < \mathbf{x}^f_n - \tilde{\mathbf{x}}_n, \mathbf{h}(\mathbf{x}_n) - \mathbf{h}(\tilde{\mathbf{x}}_n) > . \qquad (2.1.11)
\end{aligned}
$$

When the measurement function is a simple function (e.g. a measurement matrix where the state vector is linearly related to the data vector), the expressions for $P^f_{n,yy}$ and $P^f_{n,xy}$ found in equations (2.1.10) and (2.1.11) simplify and the two covariances can be related to the state vector covariance, $P^f_{n,xx}$.

We can now substitute the Kalman gain back into the formula for the state vector covariance matrix:

$$
\begin{aligned}
P^a_{n,xx} &= P^f_{n,xx} - P^f_{n,xy}(P^f_{n,yy})^{-1}P^f_{n,yx} - P^f_{n,xy}(P^f_{n,xy}(P^f_{n,yy})^{-1})^T + \\
&\quad (P^f_{n,xy}(P^f_{n,yy})^{-1})P^f_{n,yy}(P^f_{n,xy}(P^f_{n,yy})^{-1})^T \\
&= P^f_{n,xx} - P^f_{n,xy}(P^f_{n,yy})^{-1}P^f_{n,yx} - P^f_{n,xy}(P^f_{n,xy}(P^f_{n,yy})^{-1})^T + \\
&\quad P^f_{n,xy}(P^f_{n,xy}(P^f_{n,yy})^{-1})^T \\
&= P^f_{n,xx} - P^f_{n,xy}(P^f_{n,yy})^{-1}P^f_{n,yx} \\
&= P^f_{n,xx} - K_n P^f_{n,yx}. \qquad (2.1.12)
\end{aligned}
$$

This covariance matrix gives information on the uncertainties associated with the state vector *after* the adjustment step has taken place.

There are two primary ways in which the means and variances of the variables are tracked in the Kalman filter. Traditionally, in both the linear and extended Kalman filter, the mean and covariance matrix of the state variables each have their own evolution equation and are explicitly tracked as time evolves. With the advent of the Ensemble Kalman Filter it has become common to instead track the means and variances by evolving each member of a sampling, or ensemble, of stochastic variable space. In this latter case, if more specific information such as the mean or covariance of the variables in the actual system is desired, they can be estimated by calculating the mean and covariance matrix of the ensemble. Often the two approaches are mixed (see, e.g. [18, 39, 28, 38]).

Finally we mention that while the Kalman filter has traditionally been used to correct just the state variables in a given model, it has become common practice to use the Kalman filter in a parameter estimation role (see [29, 30]). By appending guesses for the unknown parameter to the state vector, evolving those parameters with the identity function during the predict step, and then allowing the analysis step to adjust the parameter values so that the state variables more closely agree with experiment, the parameter values will tend to evolve towards the ideal values for the system, that is, towards a parameter set that reproduces the experimental data fairly well. In addition, it is often, though not always, the case that the first guess for the parameters need not be close to the ideal parameter set in order for the guesses to converge to that set.

Summary of KF Algorithm The general Kalman filter algorithm consists of the main two steps, the forward prediction step using the model and the analysis or adjustment step that assimilates the data information into the model. It also consists of initialization where we must make a guess for the initial best state vector estimate ($\mathbf{x}_0^a$), the initial uncertainty associated with that estimate ($P_{0,xx}^a$), and the uncertainties associated with the model and measurements ($Q_n/\mathbf{w}_n$ and $R_n/\mathbf{v}_n$). The model and measurement uncertainties can depend on time, provided we prescribe how to find those uncertainties at a given time step. In our calculations, however, we usually assume that these matrices are constant over all time.

Note this algorithm takes the point of view that we are tracking the mean and covariance of the best estimate for the state vector. There are different ways of finding the covariances involved in the method. Some of them are detailed above. We now give the explicit, complete,

Initialize

$\mathbf{x}_0^a$	Initial best state vector estimate
$P_{0,xx}^a$	Initial uncertainty on best state vector (can be zero matrix)
$Q_n/\mathbf{w}_n$	Uncertainty associated from applying model
$R_n/\mathbf{v}_n$	Uncertainty associated with making experimental measurements

Predict Step

$\mathbf{x}_n^f = \mathbf{f}(\mathbf{x}_{n-1}^a)$	Predict new state
$P_{n,xx}^f =< \mathbf{f}(\mathbf{x}_{n-1}^a) - \tilde{\mathbf{x}}_n, \mathbf{f}(\mathbf{x}_{n-1}^a) - \tilde{\mathbf{x}}_n >$	Find new covariance
$P_{n,xy}^f =< \mathbf{h}(\mathbf{x}_n^f) - \mathbf{y}_n, \mathbf{x}_n^f - \tilde{\mathbf{x}}_n >$	Find new covariance
$P_{n,yy}^f =< \mathbf{h}(\mathbf{x}_n^f) - \mathbf{y}_n, \mathbf{h}(\mathbf{x}_n^f) - \mathbf{y}_n >$	Find new covariance

Adjustment Step

$K_n = P_{n,xy}^f (P_{n,yy}^f)^{-1}$	Find Kalman gain
$\mathbf{x}_n^a = \mathbf{x}_n^f + K_n(\mathbf{y}_n - \mathbf{h}(\mathbf{x}_n^f))$	Find analyzed state
$P_{n,xx}^a = P_{n,xx}^f - K_n P_{n,yx}^f$	Find new covariance

Repeat Process starting from Predict Step

algorithms for the linear, extended, ensemble, stochastic collocation, and Karhunen-Loève (ensemble and stochastic collocation versions) below.

2.1.1 Linear Kalman Filter

If we assume a linear model, $\mathbf{x}_n^f = A_n \mathbf{x}_{n-1}^a$, and a linear measurement function, $\mathbf{h}(\mathbf{x}_n^f) = H_n \mathbf{x}_n^f$, then the algorithm becomes:

Initialize

$\mathbf{x}_0^a$	Initial best state vector
$P_{0,xx}^a$	Initial best state vector uncertainty
Q_n	Uncertainty associated with using the model
R_n	Uncertainty associated with making experimental measurements

Predict Step

$\mathbf{x}_n^f = A_n \mathbf{x}_{n-1}^a$	Predict new state
$P_{n,xx}^f = A_n P_{n-1,xx}^a A_n^T + Q_n$	Find new covariance
$P_{n,xy}^f = P_{n,xx}^f H_n^T$	Find new covariance
$P_{n,yy}^f = H_n P_{n,xx} H_n^T + R_n$	Find new covariance

Adjustment Step

$K_n = P_{n,xy}^f (P_{n,yy}^f)^{-1}$	Find Kalman gain
$\mathbf{x}_n^a = \mathbf{x}_n^f + K_n(\mathbf{y}_n - H_n \mathbf{x}_n^f)$	Find analyzed state
$P_{n,xx}^a = P_{n,xx}^f - K_n P_{n,xx}^f K_n^T$	Find new covariance

Repeat Process from Predict Step

Extended Kalman Filter In the extended Kalman filter, we linearize the functions $\mathbf{f}$ and $\mathbf{h}$ from equations (2.1.1) and (2.1.3) respectively, by using their Jacobians. The resulting algorithm is below. Note that for linear functions this algorithm will give the exact same result as for the linear Kalman filter. The difference comes in the fact that the linear Kalman filter simply isn't defined for nonlinear functions while the extended Kalman filter, through using the Jacobian in the algorithmic process, is:

Initialize

$\mathbf{x}_0^a$	Initial best state vector
$P_{0,xx}^a$	Initial best state vector uncertainty
Q_n	Uncertainty associated with using the model
R_n	Uncertainty associated with making experimental measurements

Predict Step

$\mathbf{x}_n^f = \mathbf{f}(\mathbf{x}_{n-1}^a)$	Predict new state
$A_n = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}\Big\vert_{\mathbf{x}=\mathbf{x}_{n-1}^a}$	Define Jacobian for $\mathbf{f}$
$H_n = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\Big\vert_{\mathbf{x}=\mathbf{x}_{n-1}^a}$	Define Jacobian for $\mathbf{h}$
$P_{n,xx}^f = A_n P_{n-1,xx}^a A_n^T + Q_n$	Find new covariance
$P_{n,xy}^f = P_{n,xx}^f H_n^T$	Find new covariance
$P_{n,yy}^f = H_n P_{n,xx} H_n^T + R_n$	Find new covariance

Adjustment Step

$K_n = P_{n,xy}^f (P_{n,yy}^f)^{-1}$	Find Kalman gain
$\mathbf{x}_n^a = \mathbf{x}_n^f + K_n(\mathbf{y}_n - \mathbf{h}(\mathbf{x}_n^f))$	Find analyzed state
$P_{n,xx}^a = P_{n,xx}^f - K_n (P_{n,xy}^f)^T$	Find new covariance

Repeat Process from Predict Step

2.1.2 Ensemble Kalman filter

The Ensemble Kalman Filter (see [16, 12]) tracks the underlying distributions of the state and measurement vectors not by tracking the mean and covariances associated with the state vectors as time evolves, but by representing the underlying distribution using an ensemble of state and measurement vectors and advancing that distribution over time. Given an initial distribution described by a probability distribution function (e.g. a gaussian), a sampling of that distribution can be generated (using randn in Matlab for gaussian distributions or MCMC for more complicated distributions). That sampling or ensemble (of size q) represents the original distribution. In addition, the noise for the model and measurements are also represented by ensembles of size q. To initialize the procedure, we must initialize (or decide upon) the following vector, covariance, ensemble of state vectors, and distributions. Subscript en corresponds to ensembles of vectors.

The algorithm, along with a short description of each step, is listed in Table 2.1.1. In this and the other three algorithms (Stochastic Collocation Kalman filter, Karhunen-Loève Stochastic Collocation Kalman filter, Karhunen-Loève Ensemble Kalman filter), the dimension of the state/parameter vector $\mathbf{x}$ is $m = n + 2$, where n is the number of grid cells. In particular, we have one free state variable per grid cell and two parameters D and k_p. In this and the next algorithm, since the spatial noise is uncorrelated, the dimension of the stochastic space is also m. Ensembles of state vectors, measurement noise, and model noise are of size q and are random, rather than structured, samplings of the stochastic space, see Figure 2.1. When q is large enough, the ensembles should have a mean and variance that is approximately equal to the mean and variance of the underlying distributions. The index k corresponds to the k^{th} ensemble member. The ensembles of noise vectors $\{\mathbf{w}_{n,k}\}_{k=1}^{q}$ and $\{\mathbf{v}_{n,k}\}_{k=1}^{q}$ are drawn from the normal distributions with covariance matrices Q_n and R_n, respectively.

If the ensembles of vectors include enough members, q, the ensembles should have a mean and variance that is approximately equal to the mean and variance of the original underlying distributions. The state vector ensemble is evolved through time using the model and each member of the ensemble is updated using the measured data as follows. The index k corre-

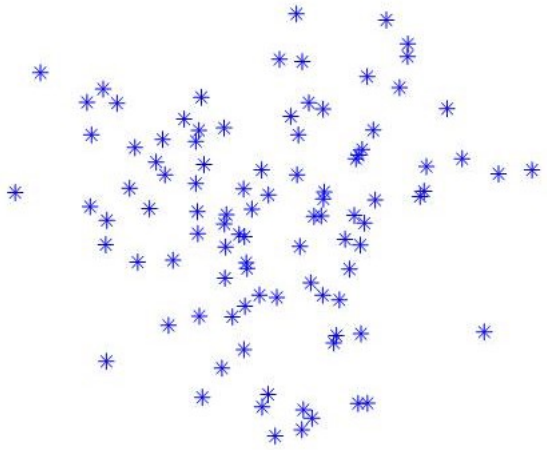


Figure 2.1: En KF random sampling of 2-d stochastic space

Initialize

$\mathbf{x}_0^a$	Initial best state vector
$P_{0,xx}^a$	Initial best state vector uncertainty
$\mathbf{x}_{0,en}^a$	Use $\mathbf{x}_0^a$ and $P_{0,xx}^a$ to obtain a sampling or ensemble of q vectors that correspond to/represent the underlying distribution
$\mathbf{w}$	Distribution of model noise corresponding to the uncertainty of using the model. At every time step a new ensemble of noise vectors (of size q) is drawn from this distribution.
$\mathbf{v}$	Distribution of measurement noise corresponding to the uncertainty associated with making experimental measurements. At every time step a new ensemble of noise vectors (of size q) is drawn from this distribution.

sponds to the k^{th} ensemble member, the index i or j correspond to the i^{th} or j^{th} component in the given ensemble member vector.

It is important to note that it is usually the case that the dependence of results on the initialization values chosen decreases as the number of data assimilation steps increase.

Table 2.1.1: EnKF Algorithm

Initialize	
$\mathbf{x}_0^a$	Initial best state vector
$P_{0,xx}^a$	Initial best state vector uncertainty
$\{\mathbf{x}_{0,k}^a\}_{k=1}^q$	Use $\mathbf{x}_0^a$ and $P_{0,xx}^a$ to obtain a sampling or ensemble of q vectors
Prediction Step	
$\mathbf{x}_{n,k}^f = \mathbf{f}(\mathbf{x}_{n-1,k}^a) + \mathbf{w}_{n,k}$	Predict new state for each ensemble member
$\overline{\mathbf{x}}_n^f = \frac{1}{q}\sum_{k=1}^q \mathbf{x}_{n,k}^f$	Mean new state, according to model
$\overline{\mathbf{y}}_n^f = \frac{1}{q}\sum_{k=1}^q \mathbf{h}(\mathbf{x}_{n,k}^f)$	Mean new measurement, *according to model*
$E_{x,k}^f = \mathbf{x}_{n,k}^f - \overline{\mathbf{x}}_n^f$	Deviation of k^{th} forecast ensemble member from mean
$E_{y,k}^f = \mathbf{h}(\mathbf{x}_{n,k}^f) - \overline{\mathbf{y}}_n^f$	Deviation of k^{th} forecast measurement of ensemble member from mean measurement
$P_{n,xx}^f = \frac{1}{q-1}\sum_{k=1}^q E_{x,k}^f (E_{x,k}^f)^T$	New xx-covariance
$P_{n,xy}^f = \frac{1}{q-1}\sum_{k=1}^q E_{x,k}^f (E_{y,k}^f)^T$	New xy-covariance
$P_{n,yy}^f = \frac{1}{q-1}\sum_{k=1}^q E_{y,k}^f (E_{y,k}^f)^T$	New yy-covariance
Adjustment Step	
$K_n = P_{n,xy}^f (P_{n,yy}^f)^{-1}$	Find Kalman gain
$\mathbf{x}_{n,k}^a = \mathbf{x}_{n,k}^f + K_n(\mathbf{y}_n + \mathbf{v}_{n,k} - \mathbf{h}(\mathbf{x}_{n,k}^f))$	Find analyzed state for each ensemble member
$\overline{\mathbf{x}}_n^a = \frac{1}{q}\sum_{k=1}^q \mathbf{x}_{n,k}^a$	Mean best estimate state, after measurement adjustment
$E_{x,k}^a = \mathbf{x}_{n,k}^a - \overline{\mathbf{x}}_n^a$	Deviation of k^{th} analyzed ensemble member from mean
$P_{n,xx}^a = \frac{1}{q-1}\sum_{k=1}^q E_{x,k}^a (E_{x,k}^a)^T$	Find new covariance

end

Hence, with enough data assimilation steps, initialization values become unimportant, just necessary to begin the data assimilation process.

2.1.3 Stochastic Collocation Kalman filter

In the ensemble Kalman filter, the mean and variance of the ensemble converge to the true mean of the ensemble (according to Monte Carlo sampling) as $1/\sqrt{q}$. Because of this, a large number of ensemble members is often required if the ensemble Kalman filter is going to effectively track the true underlying distribution as it evolves in time. When the model function $\mathbf{f}$ is costly to evaluate, the large number of required ensemble members lead to a very slow ensemble Kalman filter.

To alleviate this problem, it has become practice (unscented Kalman filter, sigma point Kalman filter, Gaussian filters, stochastic collocation Kalman filter) to build ensembles that consist of points strategically chosen from the underlying stochastic space [18, 39, 28, 43]. This is in contrast to the ensemble Kalman filter where the stochastic space is randomly sampled. When points are strategically chosen, numerical integration techniques on the corresponding structured grid can be used to obtain good estimates of what the mean and covariance of the underlying matrix are.

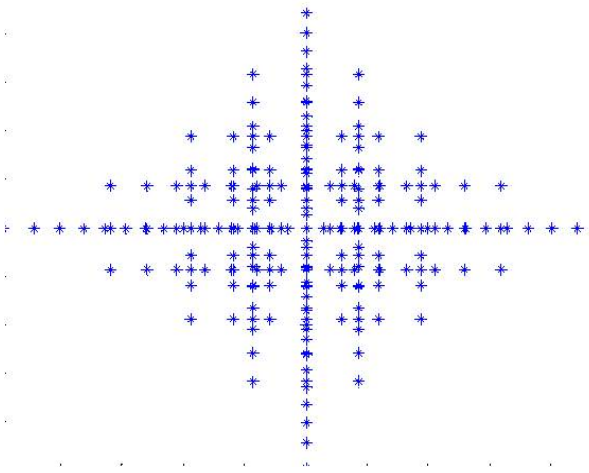


Figure 2.2: SC KF structured sampling of 2-d stochastic space

The stochastic collocation method builds an interpolant in the stochastic space using solution values at q_{SC} collocation points. Therefore its computational complexity is q_{SC} times

that of a deterministic problem. Thus, we need to choose a nodal set Θ with fewest possible number of points under a prescribed accuracy requirement. There are several choices of such collocation points, using either tensor products of one-dimensional nodal sets, or sparse grids constructed by the Smolyak algorithm [32, 42]. The Smolyak approximation is a linear combination of product formulae, and the linear combination is chosen in such a way that an interpolation property for one-dimensional spaces is preserved for multidimensional spaces. Only products with a relatively small number of points are used and the resulting nodal set has significantly fewer number of nodes compared to the tensor product rule. In this paper, we use Smolyak formulae that are based on one-dimensional polynomial interpolation at the extrema of the Hermite polynomials, which are the orthogonal polynomials with a weight given by the probability density function of the normal distribution (i.e. Gaussian abscissas). Other choices, such as the extrema of the Chebyshev polynomials (i.e. Clenshaw-Curtis abscissas), can be considered as well. The SCKF algorithm is given in Table 2.1.2. Here q_{SC} is the size of the ensemble, $\{\mathbf{r}_{SC,k}\}_{k=1}^{q_{SC}} \in \mathbf{R}^m$, are the collocation points, and $\{c_{SC,k}\}_{k=1}^{q_{SC}}$ are the collocation weights. In our implementation we use Smolyak level one sparse grid [32, 42], which has two collocation points in each dimension plus the origin, resulting in $q_{sc} = 2m+1$.

Remark 2.1.1. *It is important to note that we have modified the standard SC implementation. In particular, since the set of collocation points is fixed, sampling the noise at these points at each data assimilation step would result in adding noise to the model and measurements in the same stochastic direction. To avoid this, at each data assimilation step the Kalman gain is used to adjust the mean and a new ensemble is generated using the new mean, the vector of collocation points and the new covariance matrix.*

2.1.4 Karhunen-Loève Stochastic Collocation Kalman filter, Karhunen-Loève Ensemble Kalman filter

The most costly portion of the Kalman filter is the functional evaluation of $\mathbf{f}$, which corresponds to advancing the computational model in time. As such, the fewer ensemble members a method needs to obtain satisfactory results, the faster the method is. For low-dimensional systems, the stochastic collocation Kalman filter needs few ensemble members in its or-

Table 2.1.2: SCKF Algorithm

Initialize

$\overline{\mathbf{x}}_0^a$	Initial best state vector (corresponds to a mean)
$P_{0,xx}^a$	Initial best state vector uncertainty/covariance
$\{\mathbf{r}_{SC,k}\}_{k=1}^{q_{SC}}$	Collocation points
$\{c_{SC,k}\}_{k=1}^{q_{SC}}$	Weights for stochastic collocation

For $n = 1 \dots N$

Prediction Step

$\mathbf{x}_{n-1,k}^a = \overline{\mathbf{x}}_{n-1}^a + \sqrt{P_{n-1,xx}^a}\mathbf{r}_{SC,k}$	Use variance associated with each component to readjust the ensemble
$\mathbf{x}_{n,k}^f = \mathbf{f}(\mathbf{x}_{n-1,k}^a)$	Predict new state
$\overline{\mathbf{x}}_n^f = \sum_{k=1}^{q_{SC}} c_{SC,k}\mathbf{x}_{n,k}^f$	Mean new state, according to model
$\overline{\mathbf{y}}_n^f = \sum_{k=1}^{q_{SC}} c_{SC,k}\mathbf{h}(\mathbf{x}_{n,k}^f)$	Mean new measurement, *according to model*
$E_{x,k}^f = \mathbf{x}_{n,k}^f - \overline{\mathbf{x}}_n^f$	Deviation of k^{th} ensemble member from mean
$E_{y,k}^f = \mathbf{h}(\mathbf{x}_{n,k}^f) - \overline{\mathbf{y}}_n^f$	Deviation of measurement of k^{th} ensemble member from mean measurement
$P_{n,xx}^f = \sum_{k=1}^{q_{SC}} c_{SC,k}E_{x,k}^f(E_{x,k}^f)^T + Q_n$	Find new covariance
$P_{n,xy}^f = \sum_{k=1}^{q_{SC}} c_{SC,k}E_{x,k}^f(E_{y,k}^f)^T$	Find new covariance
$P_{n,yy}^f = \sum_{k=1}^{q_{SC}} c_{SC,k}E_{y,k}^f(E_{y,k}^f)^T + R_n$	Find new covariance

Adjustment Step

$K_n = P_{n,xy}^f(P_{n,yy}^f)^{-1}$	Find Kalman gain
$\overline{\mathbf{x}}_n^a = \overline{\mathbf{x}}_n^f + K_n(\mathbf{y}_n - \overline{\mathbf{y}}_n^f)$	Find adjusted mean state
$P_{n,xx}^a = P_{n,xx}^f - K_n P_{n,yy}^f K_n^T$	Find new covariance

end

ganized ensemble, while the Ensemble Kalman filter needs many in its randomly chosen ensemble. As the dimension is increased, however, the stochastic collocation Kalman filter suffers from the curse of dimensionality. For instance, for one spatially dependent variable on a coarse $10 \times 10 \times 10$ computational grid, the dimension of the stochastic space is one thousand, therefore over one thousand ensemble members are required to run the stochastic collocation Kalman filter. A $20 \times 20 \times 20$ grid would require over eight thousand ensemble members. The ensemble Kalman filter often requires only around one thousand ensemble members for similarly sized grids.

To address this problem, one can explore a parametrized noise representation, such as the Karhunen-Loève (KL) expansion. The uncertainties associated with each component of the state vector are often correlated with each other. This is especially true when the components correspond to spatially dependent variables on computational grids. We use the KL expansion to represent these spatially correlated uncertainties. It is very similar to a Fourier expansion of a function with eigenfunctions that look sinusoidal. On a discrete grid of size $p \times p \times p$, one needs p^3 KL eigenfunctions to completely represent a given discrete correlation function on the grid. Like a Fourier series, however, it can be shown that in continuous space the eigenvalues decay fast and the KL expansion of a given function converges to that function as more terms are included in the expansion [14]. As such, including the first few terms of the KL expansion in discrete space, instead of p^3 terms, will allow to sufficiently represent the distribution of the possible state of the system. Doing so reduces the effective stochastic space and allows one to use a much smaller ensemble to represent the underlying distributions. This corresponds to fewer necessary evaluations of the model function $\mathbf{f}$ and faster Kalman filtering.

Given a correlation function in two dimensions $C_v(\vec{x}_\alpha, \vec{x}_\beta)$ for a stochastic variable v, the corresponding Karhunen-Loève expansion for that variable is given by [14]:

$$v(\vec{x}, \omega) = E[v](\vec{x}) + \sum_{i=1}^{\infty} \xi_i(\omega) \sqrt{\lambda_i} f_i(\vec{x}),$$

where the corresponding eigenfunctions $f_i(\vec{x})$ satisfy the following integral equation

$$\iint_D C(\vec{x}_\alpha, \vec{x}_\beta) f_i(\vec{x}_\alpha) d\vec{x}_\alpha = \lambda_i f_i(\vec{x}_\beta).$$

Due to the symmetry and positive definiteness of the covariance function, the corresponding eigenfunctions are mutually orthogonal. In addition, since in our case the noise being represented is normally distributed at each point, $\xi_i(\omega)$ are uncorrelated normal distributions with mean zero and standard deviation one.

As we discretize the model, it is useful to discuss the corresponding discrete version of the KL expansion, which is just an eigenfunction expansion. Recall that we consider a cell-centered finite difference method on a two-dimensional rectangular grid with n grid cells. Let $\mathbf{C} \in \mathbf{R}^{n \times n}$ be the covariance matrix where $\mathbf{C}_{ij}$ is the covariance between the noise at the two cell centers (x_i, y_i) and (x_j, y_j). The corresponding expansion is

$$\vec{v}(\omega) = E[\vec{v}] + \sum_{i=1}^{n} \xi_i(\omega) \sqrt{\lambda_i} \vec{e}_i,$$

where $\mathbf{C}$ the eigenvectors $\vec{e}_i \in \mathbf{R}^n$ satisfy

$$\mathbf{C}\vec{e}_i = \lambda_i \vec{e}_i.$$

The eigenvectors are mutually orthogonal because of the symmetry and positive definiteness of the covariance matrix. In our computations we use the covariance matrix [15]

$$\mathbf{C}_{ij} = \sigma^2 e^{-|x_i - x_j|/L_x - |y_i - y_j|/L_y},$$

where σ is the variance and L_x, L_y are the correlation lengths. The above function is widely used for modeling diffusion processes in porous media [45, 15]. We assume that it is also suitable for cell migration processes. The better our estimate of the covariance function/-matrix of the process, the better the corresponding estimate of the KL expansion for the process will be and the better the KL version of the KF will perform [46, 45]. We also note that the variance σ above and the corresponding $\sqrt{\lambda_i}$'s can be rescaled from one time step to another.

Our primary assumption that we employ when applying the Karhunen-Loève expansion in the Kalman filter is that the covariance matrix of the process in question looks *approximately* like the above covariance function. Approximately is an important word for as we do not really know what the corresponding covariance matrix looks like for the cell migration process.

Table 2.1.3: KLSCKF Algorithm

Initialize

$\overline{\mathbf{x}}_0^a$	Initial best state vector
$P_{0,xx}^a$	Initial best state vector uncertainty
$\mathbf{E}$	Matrix of orthonormalized eigenvectors
$\{\mathbf{r}_{KL,k}\}_{k=1}^{q_{KL}}$	Collocation points
$\{c_{KL,k}\}_{k=1}^{q_{KL}}$	Weights for stochastic collocation

For $n = 1 \dots N$

Prediction Step

$P_E = \mathbf{E}^T P_{n-1,xx}^a \mathbf{E}$	Project covariance onto the KL eigenspace
$\mathbf{x}_{n-1,k}^a = \overline{\mathbf{x}}_{n-1}^a + \mathbf{E}\sqrt{P_E}\mathbf{r}_{KL,k}$	Use projected covariance to readjust the ensemble
$\mathbf{x}_{n,k}^f = \mathbf{f}(\mathbf{x}_{n-1,k}^a)$	Predict new state
$\overline{\mathbf{x}}_n^f = \sum_{k=1}^{q_{KL}} c_{KL,k}\mathbf{x}_{n,k}^f$	Mean new state, according to model
$\overline{\mathbf{y}}_n^f = \sum_{k=1}^{q_{KL}} c_{KL,k}\mathbf{h}(\mathbf{x}_{n,k}^f)$	Mean new measurement, *according to model*
$E_{x,k}^f = \mathbf{x}_{n,k}^f - \overline{\mathbf{x}}_n^f$	Deviation of k^{th} ensemble member from mean
$E_{y,k}^f = \mathbf{h}(\mathbf{x}_{n,k}^f) - \overline{\mathbf{y}}_n^f$	Deviation of measurement of k^{th} ensemble member from mean measurement
$P_{n,xx}^f = \sum_{k=1}^{q_{KL}} c_{KL,k}E_{x,k}^f(E_{x,k}^f)^T + Q_n$	Find new covariance
$P_{n,xy}^f = \sum_{k=1}^{q_{KL}} c_{KL,k}E_{x,k}^f(E_{y,k}^f)^T$	Find new covariance
$P_{n,yy}^f = \sum_{k=1}^{q_{KL}} c_{KL,k}E_{y,k}^f(E_{y,k}^f)^T + R_n$	Find new covariance

Adjustment Step

$K_n = P_{n,xy}^f(P_{n,yy}^f)^{-1}$	Find Kalman gain
$\overline{\mathbf{x}}_n^a = \overline{\mathbf{x}}_n^f + K_n(\mathbf{y}_n - \overline{\mathbf{y}}_n^f)$	Find adjusted mean state
$P_{n,xx}^a = P_{n,xx}^f - K_n P_{n,yy}^f K_n^T$	Find new covariance

end

The better our estimate of the covariance function/matrix of the process, the better the corresponding estimate of the Karhunen-Loève expansion for the process will be and the better the Karhunen-Loève version of the KF will perform (see e.g., [46, 45]). Unfortunately, finding out what the covariance function/matrix really looks would probably require the use of extensive MCMC simulations coupled with an appropriate definition of the probability space for the initial wound shapes and sizes. Both running the MCMC appropriately and defining an appropriate probability space would take much time and effort. In addition, during that process we would also obtain parameter estimates along the way to obtaining estimates of the covariance function/matrix rendering employment of the KF for parameter estimation afterwards unnecessary.

Hence we proceed instead without a highly accurate covariance function/matrix by making some of the following assumptions (some of which have already been discussed). We assume the noise associated with the process being modeled is distributed normally at each point in the domain (however, each the noise may have a different standard deviation at each point). We also assume that the covariance may be represented approximately by the above covariance function/matrix. This means, in particular, that the noise can be written using an expansion:

$$n(\vec{x},\omega) \approx \sum_{i=1}^{\infty} n_i(\omega) f_i(\vec{x})$$
$$\vec{n}(\omega) \approx \sum_{i=1}^{\infty} n_i(\omega) \vec{e}_i,$$

where the normal distributions $n_i(\omega)$ have mean zero, are independent of each other, and have standard deviations approximately equal to $\sqrt{\lambda_i}$. Define these standard deviations to be s_i for the time being. To allow for some flexibility, we do not always set s_i to be equal to $\sqrt{\lambda_i}$ but rather allow them to vary. The corresponding expansion for the noise becomes

$$n(\vec{x},\omega) \approx \sum_{i=1}^{\infty} s_i \xi_i(\omega) f_i(\vec{x})$$

$$\vec{n}(\omega) \approx \sum_{i=1}^{M} s_i \xi_i(\omega) \vec{e}_i.$$

If the process indeed has a covariance equal to the covariance function/matrix above, the s_i will not stray too far from $\sqrt{\lambda_i}$. In addition, another critical assumption, we assume that the s_i decay to zero relatively rapidly as i grows so that there are only a few dominant eigenfunctions/vectors.

Recovering the "best" s_i for a given covariance function/matrix given above (with covariance $C(\vec{x}_\alpha, \vec{x}_\beta)/\mathbf{C}$) from another random covariance matrix (with covariance $\tilde{C}(\vec{x}_\alpha, \vec{x}_\beta)/\tilde{\mathbf{C}}$) becomes an important question later on. Hence consider a random normal distribution with an associated covariance matrix that may or may not look like the covariance function/-matrix given above. The corresponding noise for that matrix (denoted with tildes) is given by:

$$\vec{\tilde{n}}(\omega) \approx \sum_{i=1}^{M} \sqrt{\tilde{\lambda}_i} \tilde{\xi}_i(\omega) \vec{\tilde{e}}_i.$$

We wish to know how much of this noise will lie along the i^{th} eigenvector direction for the first covariance function/matrix. This is easily found (note the eigenvectors have been normalized to have unitary length):

$$\begin{aligned}
\tilde{p}_j(\omega) = \vec{\tilde{n}}_{||\vec{e}_j}(\omega) &\approx \vec{e}_j^T \sum_{i=1}^{M} \sqrt{\tilde{\lambda}_i} \tilde{\xi}_i(\omega) \vec{\tilde{e}}_i \\
&\approx \vec{e}_j^T \sum_{i=1}^{M} \tilde{z}_i(\omega) \vec{\tilde{e}}_i \\
&\approx \vec{e}_j^T \tilde{\mathbf{E}} \vec{\tilde{z}}(\omega) \\
\vec{\tilde{p}}(\omega) &\approx \mathbf{E}^T \tilde{\mathbf{E}} \vec{\tilde{z}}(\omega).
\end{aligned}$$

The corresponding noise can then be rewritten in the new basis as:

$$\vec{\tilde{n}}(\omega) = \sum_{j=1}^{M} \tilde{p}_j(\omega) \vec{e}_j$$

or if we truncate using only the dominant eigenfunctions/vectors we get:

$$\begin{aligned}\vec{\tilde{n}}(\omega) &= \sum_{j=1}^{N} \tilde{p}_j(\omega)\vec{e}_j \\ &= \mathbf{E}\vec{\tilde{p}}(\omega) \\ &= \mathbf{E}\mathbf{E}^T\tilde{\mathbf{E}}\vec{\tilde{z}}(\omega),\end{aligned}$$

where usually $N << M$ (though $N = M$ still works) and the dimensions of the vector $\vec{\tilde{p}}$ and matrix of eigenvectors $\mathbf{E}$ have been adjusted appropriately.

In order to use this formula, however, we must eigendecompose the random matrix each time we want to project the noise corresponding to that matrix onto the known probability space given by the covariance function/matrix formulae given above.

Finally, Table 2.1.4 presents the KL version of the EnKF algorithm. The only difference is that the ensemble is projected onto the KL eigenspace at each assimilation step.

Table 2.1.4: KLEnKF Algorithm

$\mathbf{x}_0^a$	Initial best state vector
$P_{0,xx}^a$	Initial best state vector uncertainty
$\{\mathbf{x}_{0,k}^a\}_{k=1}^q$	Use $\mathbf{x}_0^a$ and $P_{0,xx}^a$ to obtain a sampling or ensemble of q vectors
Prediction Step	
$\mathbf{x}_{n-1,k}^a = \overline{\mathbf{x}}_{n-1}^a + \mathbf{E}\mathbf{E}^T(\mathbf{x}_{n-1,k}^a - \overline{\mathbf{x}}_{n-1}^a)$	Project the ensemble's distance from the mean onto the eigenspace and use this to construct an appropriately structured ensemble
$\mathbf{x}_{n,k}^f = \mathbf{f}(\mathbf{x}_{n-1,k}^a) + \mathbf{w}_{n,k}$	Predict new state for each ensemble member
$\overline{\mathbf{x}}_n^f = \frac{1}{q}\sum_{k=1}^q \mathbf{x}_{n,k}^f$	Mean new state, according to model
$\overline{\mathbf{y}}_n^f = \frac{1}{q}\sum_{k=1}^q \mathbf{h}(\mathbf{x}_{n,k}^f)$	Mean new measurement, *according to model*
$E_{x,k}^f = \mathbf{x}_{n,k}^f - \overline{\mathbf{x}}_n^f$	Deviation of k^{th} forecast ensemble member from mean
$E_{y,k}^f = \mathbf{h}(\mathbf{x}_{n,k}^f) - \overline{\mathbf{y}}_n^f$	Deviation of k^{th} forecast measurement of ensemble member from mean measurement
$P_{n,xx}^f = \frac{1}{q-1}\sum_{k=1}^q E_{x,k}^f (E_{x,k}^f)^T$	New xx-covariance
$P_{n,xy}^f = \frac{1}{q-1}\sum_{k=1}^q E_{x,k}^f (E_{y,k}^f)^T$	New xy-covariance
$P_{n,yy}^f = \frac{1}{q-1}\sum_{k=1}^q E_{y,k}^f (E_{y,k}^f)^T$	New yy-covariance
Adjustment Step	
$K_n = P_{n,xy}^f (P_{n,yy}^f)^{-1}$	Find Kalman gain
$\mathbf{x}_{n,k}^a = \mathbf{x}_{n,k}^f + K_n(\mathbf{y}_n + \mathbf{v}_{n,k} - \mathbf{h}(\mathbf{x}_{n,k}^f))$	Find analyzed state for each ensemble member
$\overline{\mathbf{x}}_n^a = \frac{1}{q}\sum_{k=1}^q \mathbf{x}_{n,k}^a$	Mean best estimate state, after meas. adjustment
$E_{x,k}^a = \mathbf{x}_{n,k}^a - \overline{\mathbf{x}}_n^a$	Deviation of k^{th} analyzed ens. member from mean
$P_{n,xx}^a = \frac{1}{q-1}\sum_{k=1}^q E_{x,k}^a (E_{x,k}^a)^T$	Find new covariance

end

2.2 MARKOV CHAIN MONTE CARLO METHODS

Uncertainty is often an appropriate model for systems of high complexity, which arise in a broad spectrum of scientific fields and research areas. The mathematical treatment of the models and algorithms in chapters 4 and 5 is Bayesian, which means that all the results consider the unknown parameters as random variables and search the probabilistic distribution of the unknowns. Probability distributions are used for modeling the uncertainties in the models. By introducing the concept of the stochastic prior state space to the Bayesian formulation, the deterministic forward problem is reformulated as a stochastic one ([19, 24]). In Bayesian inference framework, we consider the forward problem $F(m) \approx d$, where m is a vector of unknown model parameters and d is a vector of measurements, both m and d are assumed to be random variables. The Bayesian approach combines a prior distribution model with the likelihood to formulate the posterior probability density function (PPDF): $p(m|d) \approx p(d|m)p(m)$, where $p(m|d)$ is the PPDF, $p(d|m)$ is the likelihood function and $p(m)$ is the prior probability density function. The PPDF is considered as a solution to the inverse problem, and various statistics can be estimated from the samples of this distribution.

2.2.1 The Metropolis-Hastings Algorithm

Suppose we wish to draw samples from the posterior distribution $\pi(\theta)$, where θ is considered our parameter that needs to be estimated. The Metropolis-Hastings Markov Chain (MHMC) sampling algorithm (see, e.g. [49, 50]) is a Monte Carlo method for finding samples $(\theta_1, \ldots, \theta_n)$ for any distribution $\pi(\theta)$. The algorithm begins with some arbitrary initial value θ_0, then for each iteration n, where $n = 1, 2, 3, \ldots$, generate a candidate value θ_* from proposal distribution $Q(x, \theta_{n-1})$, which gives us a candidate for the next value θ_n, given the previous sample value θ_{n-1}. When Q is symmetric, i.e. $Q(x, y) = Q(y, x)$, we have a special case of MHMC, called Metropolis algorithm. Next, we must calculate the acceptance ratio

$$\alpha = \frac{\pi(\theta_*) \cdot Q(\theta_*, \theta_{n-1})}{\pi(\theta_{n-1}) \cdot Q(\theta_{n-1}, \theta_*)},$$

which will be used to decide whether to accept or reject the candidate. If $\alpha \geq 1$, then automatically accept the candidate by setting $\theta_n = \theta_*$. Otherwise, accept the candidate with

probability α; if the candidate is rejected, set $\theta_n = \theta_{n-1}$, instead. The resulting sequence $(\theta_n)_{n\geq 0}$ is a Markov Chain that has a stationary probability distribution that converges to $\pi(\theta)$ when considering the limit upon large n. We present here the algorithm, which was used in chapter 5:

Metropolis-Hastings Algorithm:

INITIALIZATION: θ_0.

for $n = 1, 2, 3, \ldots$ **do**

Sample a candidate value θ_* from proposal distribution Q.

Compute $\alpha = \frac{\pi(\theta_*)\cdot Q(\theta_*, \theta_{n-1})}{\pi(\theta_{n-1})\cdot Q(\theta_{n-1}, \theta_*)}$.

if $\alpha \geq 1$ **then**

Accept $\theta_n = \theta_*$.

else

Accept with probability α.

end if

end for

2.2.2 Parallel Tempering

We incorporated parallel tempering, or replica exchange, into our numerical simulations for improving exploration of the parameter space. In parallel tempering, we simulate M replicas of a system in parallel, for a set of temperatures $T_1 < T_2 < \cdots < T_{M-1} < T_M$. It is not clear how to choose the optimal number M of replicas. Chains at neighboring temperature levels partially exchange their configurations through swapping. At higher temperature levels, corresponding to lower β values, swaps are easily accepted because the distribution is flat and thus hotter chains travel around the sample space a lot. At lower temperature levels, swaps are rarely accepted because the distribution is very spiky and hence any colder chains tend to get stuck in the local energy minima (see [11]). At convergence, the simulation from the first chain represent draws from the target distribution. By carefully choosing the β_i values, where $\beta_i = \frac{1}{k_B T_i}, i = 1, \ldots, M$ and k_B is the Boltzmann constant, significant

improvement in the mixing properties of several Monte Carlo simulations can be achieved that exceeds the extra computational cost of running parallel simulations [31].

3.0 PARAMETER ESTIMATION FOR A NONLINEAR DIFFUSION MODEL OF EPITHELIAL CELL MIGRATION USING STOCHASTIC COLLOCATION AND THE KARHUNEN-LOÈVE EXPANSION

Traditional approaches to obtaining ideal parameter sets include least-squares or maximum likelihood approaches in which a cost functional (usually a sum of weighted squared differences between model and experimental values) is minimized. Direct optimization methods such as the Nelder-Mead simplex method or the conjugate gradient method are often used to find the corresponding minimum of the cost functional, which serves as the ideal parameter set [35]. Probability based methods, such as the MCMC method, have also been used to explore parameter space and search for optimal parameters. Direct optimization techniques may get stuck in local minima and may require large amounts of time to find minima in high dimensional space. While improved implementations of the MCMC algorithms exist, see [11, 19, 41, 40, 2], MCMC generally suffers from the need of a large number of simulations before optimal parameter sets are obtained.

Usually, in the above methods, a simulation is run from the start of an experiment to the end of an experiment before the cost function is evaluated and the best guess for the optimal parameter set is adjusted. In contrast, sequential data assimilation techniques adjust parameter sets at every time at which experimental data is available. Adjusting parameters more often usually allows for quicker convergence to desired parameter estimates. In this chapter, which follows the paper [7], we study several Kalman filter (KF) techniques [12] that update the parameter estimate multiple times per simulation run and compare their performance for parameter estimation for a partial differential equation (PDE) model of cell migration in an *in vitro* experiment [1].

Kalman filter methods usually take a running temporal model and periodically update

it by using experimental measurements. While traditionally these methods have been used to update the values of the dependent variables for a given model of a physical system, they can also be used to update estimates of the parameter values of the model if an initial guess for those parameter values is given [18].

The original linear Kalman filter can only be used for models with linear dynamics. The extended Kalman filter uses the Jacobian to linearize and deal with nonlinear dynamics. Both the linear and extended Kalman filters track the underlying distributions of the dependent variables and unknown parameters through time by evolving and tracking the mean and variance of the variables in the model. While the extended Kalman filter can be used on moderately nonlinear problems, it can suffer when presented with highly nonlinear problems. This is because the Jacobians provide only local information.

Other Kalman filters alleviate the problem of potentially misleading local information by using a global sampling of points, rather than a local mean and Jacobian-derived variance, to represent the underlying distribution. The ensemble Kalman filter [16, 12] tracks the evolution of the variable distributions by using a Monte Carlo sampling of the variable space that is evolved through time. Recently other Kalman filters have been introduced that use structured samplings of stochastic space that are based upon quadrature rules [18, 39, 43]. Our stochastic collocation Kalman filter (SCKF) is of this type as it uses sparse grid collocation or quadrature in order to estimate the mean and variance resulting when the model is propagated in time. For moderately sized problems, filters based on structured sampling can be more efficient than the ensemble Kalman filter, since they require fewer realizations to obtain comparable accuracy.

For spatial models where PDEs are discretized on a grid, the model errors, i.e., the errors introduced to the variables when using the model to evolve those variables in time, can be either spatially uncorrelated or correlated. We consider cell-centered discretization with a free variable at each grid cell. Therefore, with uncorrelated model errors, there is one stochastic dimension per each element of the spatial grid, resulting in an increased dimension of the stochastic space when the spatial grid is refined. However, for PDE models, the model errors at one grid location tend to be associated with model errors at nearby grid locations. To incorporate this correlation, we have assumed that the errors at the grid cells

can be represented by a Karhunen-Loève (KL) expansion, which is based on an eigenfunction expansion of the covariance. The KL expansion can be truncated due the fast decay of the eigenvalues (see [14]). This results in a reduction in the dimension of the effective stochastic space which corresponds to fewer computations needed for a desired parameter estimation. In particular, there is one stochastic dimension for each KL term and the dimension of the stochastic space is independent of the spatial grid.

We investigate the efficiency and accuracy of using sequential data assimilation via KF methods, structured sampling via SC methods, and the KL expansion for parameter estimation in a model of intestinal epithelial cell migration. We do this by comparing parameter estimates obtained from five different parameter estimation techniques (see Table 3.0.1): direct optimization of a cost functional (DO), ensemble Kalman filter (EnKF), stochastic collocation Kalman filter (SCKF), ensemble Kalman filter with KL expansion (KLEnKF), and stochastic collocation Kalman filter with KL expansion (KLSCKF). In the SC algorithms, a new random ensemble is generated after each data assimilation step to avoid adding noise in the same stochastic direction. We present computational results for two cases with synthetic data with and without noise, as well as experimental data from the laboratory of David Hackam [1]. We observe that all algorithms are able to match the target solution or experimental data and to estimate the diffusion coefficient and the growth rate in section 3.2. However, the algorithms that employ SC acceleration and the KL expansion are computationally more efficient, as they require fewer ensemble members for comparable accuracy.

Technique	sequential data assimilation	structured sampling	Karhunen-Loève expansion
DO			
EnKF	X		
SCKF	X	X	
KLEnKF	X		X
KLSCKF	X	X	X

Table 3.0.1: Parameter estimation techniques used

3.1 METHODS

3.1.1 Experiments

The experimental data was obtained in the Hackam Lab at the University of Pittsburgh and the experimental procedures have been presented in [1].

3.1.2 Model

The mathematical model consists of a two-dimensional domain representing a layer of epithelial cells that evolves in time according to the partial differential equation

$$\frac{\partial e_c}{\partial t} = D\nabla \cdot \left(\left(\frac{e_c^2}{e_c^2 + (e_{c,max} - e_c)^2} \right) \nabla e_c \right) + k_p e_c (e_{c,max} - e_c). \tag{3.1.1}$$

This nonlinear diffusion equation for the epithelial cell concentration e_c has been used to model wound closure in necrotizing enterocolitis [3]. Here D is the diffusion coefficient, k_p is the growth rate, and $e_{c,max} = 1$ is the maximum concentration.

The S-shaped nonlinear diffusion term is chosen from the Buckley-Leverett model of two-phase flow in porous media [9]. It results in increase in epithelial cell migration when the epithelial layer integrity increases, with no migration at $e_c = 0$ and maximal migration at $e_c = e_{c,max}$.

A standard cell-centered finite difference method was implemented in MATLAB to discretize this equation on a 10×10 grid (100 free state variables) including appropriate upwinding of the nonlinear diffusion term [17] and using Forward Euler in time with step sizes that do not violate the CFL condition, see [3] for details. The simulation domain is the rectangle $[-0.05, 0.05] \times [-0.035, 0.035]$ discretized on a 10×10 spatial mesh. The initial condition for all tests is obtained from the initial image from the experimental data. It corresponds to an initial wound with irregular shape that closes during the simulation. We take $e_c = 0$ inside the wound and $e_c = 1$ outside.

3.1.3 Measurements

To compare the efficiency and accuracy of the four methods for this particular model, we consider three separate sets of measurements for the given system. The first set of measurements is manufactured by running the model for 3.75 h with $k_p = 1/\text{h}$ and $D = 3 \times 10^{-6}$ cm^2/h. The second set of measurements is obtained by adding white noise with variance 3×10^{-3} to the values in the first set of measurements. The third set of measurements is taken directly from the *in vitro* experiment mentioned in Section 3.1. We use a time series of images in order to determine the edge of the wound, see Figure 3.5, and the measured value of e_c in a grid cell is equal to the fraction of the cell that resides outside the wound edge yielding 0 for cells entirely inside the wound edge and 1 for those entirely outside the edge. We refer to the three sets of measurements as noiseless simulated measurements, noisy simulated measurements, and real measurements. Both the simulated and real measurements are assimilated every fifteen minutes.

3.1.4 Comparisons

For both the noiseless and noisy simulated measurements, we calculate the parameters errors, i.e., the difference between the actual value of the parameter and the estimated value. For the real measurements case, we compare the parameter values obtained via the KF methods with parameter values obtained using a direct optimization simplex method [35]. The latter

is based on minimizing the residual error

$$R(t) = \sqrt{\int_{\Omega} (\vec{m}^{model}(t) - \vec{m}^{experiment}(t))^2 dxdy},$$

where $\vec{m}^{model}$ is the vector of estimated measurements of state variables and parameters according to the model and $\vec{m}^{experiment}$ is the vector of actual measurements. In particular, the method finds the parameter values that minimize the local residual at time t_{n+1}, i.e., given $\vec{m}^{model}(t_n) = \vec{m}^{experiment}(t_n)$, find $D^{model}(t_{n+1})$ and $k_p^{model}(t_{n+1})$ such that $R(t_{n+1})$ is minimized. The direct optimization solution is used as a reference solution, i.e., the closer a KF result is to the direct optimization result, the more accurately the KF method estimates the parameters.

3.2 NUMERICAL RESULTS

The numerical experiments are performed using MATLAB. The equation (3.1.1) is discretized on a rectangular mesh using cell-centered finite differences in space, an upwind scheme [17] for the effective diffusion coefficient, and Forward Euler for the time integration with step sizes that do not violate the CFL condition, see [3] for details.

The simulation domain is the rectangle $[-0.05, 0.05] \times [-0.035, 0.035]$ discretized on a 10×10 spatial mesh. For the simulated measurements, the initial condition is $e_c = 0$ within a disk centered at the origin with an area 20% of the total area, and $e_c = 1$ outside of the disk. Dirichlet condition $e_c = 1$ is specified on the boundary. This corresponds to an initial wound at the center that closes during the simulation. The parameter values used for producing the simulated measurements are $D = 3 \times 10^{-6}$ for the diffusion coefficient and $k_p = 1$ for the growth rate. The parameter estimation methods for all three measurement types are implemented with initial guesses $D = 1 \times 10^{-6}$ and $k_p = .5$.

For each of the three types of aforementioned measurements, we use each of the four parameter estimation techniques presented in Section 3.1: the EnKF, SCKF, KLSCKF and KLEnKF methods. Since the function evaluation to advance the model in the prediction step is the dominant computational cost, and each ensemble member requires one function

	Number of spatial parameters	Dimension of stochastic space	Ensemble size
EnKF	$n = 10 \times 10 = 100$	$m = n + 2 = 102$	$q = 10n = 1000$
SCKF	$n = 10 \times 10 = 100$	$m = n + 2 = 102$	$q_{SC} = 2m + 1 = 205$
KLSCKF	$n_{KL} = 7 \times 7 = 49$	$m_{KL} = n_{KL} + 2 = 51$	$q_{KL} = 2m_{KL} + 1 = 103$
KLEnKF	$n_{KL} = 7 \times 7 = 49$	$m_{KL} = n_{KL} + 2 = 51$	$q = 10n_{KL} = 490$

Table 3.2.1: Number of parameters, stochastic space dimension, and ensemble size for the four methods.

evaluation at each data assimilation step, for comparison purpose we define the computational cost to be the size of the ensemble. Recall that for the EnKF the dimension of the stochastic space is $m = n + 2$, where n is the number of grid cells. Here $n = 10 \times 10 = 100$ and $m = 102$. We choose for the size of the ensemble $q = 1000$, which corresponds to 10 ensemble members per grid cell. In the SCKF, the dimension of the stochastic space is also $m = n + 2 = 102$, but the size of the ensemble for level one Smolyak sparse grid is $q_{SC} = 2m + 1 = 205$. In the KL-based methods the dimension of the stochastic space is independent of the number of cell in the physical grid, but depends on the number of terms in the KL expansion. In our simulations we choose $n_{KL} = 7 \times 7 = 49$ KL terms, using 7 eigenfunctions in each x and y directions. Since the KL eigenvalues decay exponentially fast, the truncated series provides a highly accurate approximation of the full one, see Section 2.1.4. The dimension of the stochastic space is $m_{KL} = n_{KL} + 2 = 51$ and the size of the SC ensemble is $q_{KL} = 2m_{KL} + 1 = 103$. Finally, in the KLEnKF, the dimension of the stochastic space is as in the KLSCKF, $m_{KL} = 51$, but the size of the ensemble needs to be chosen to provide an accurate Monte Carlo sampling. For a fair comparison to the EnKF where $q = 10n$, we choose here $q = 10n_{KL} = 10 \times 49 = 490$. These dimensions are summarized in Table 3.2.1. Note that the computational cost of EnKF is approximately twice the cost of the KLEnKF, five times the cost of the SCKF, and ten times the cost of KLSCKF.

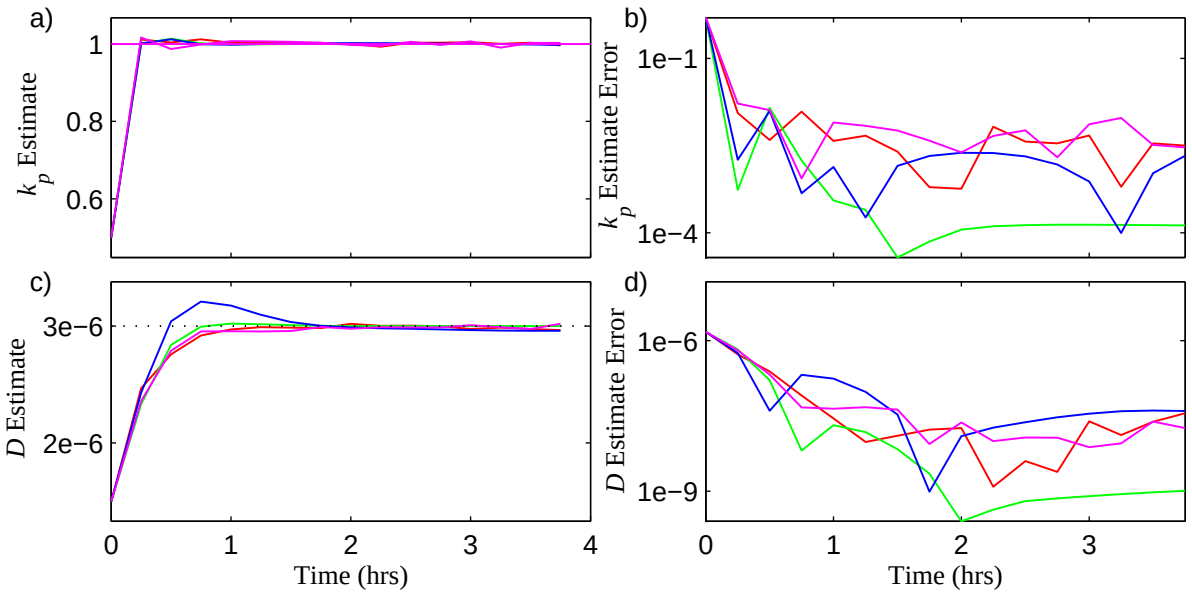


Figure 3.1: Parameter estimates and errors using noiseless simulated data for the EnKF(red), SCKF(green), KLSCKF(blue) and KLEnKF(magenta).

3.2.1 Noiseless Simulated Measurements

Figures 3.1a and 3.1c show the Kalman filter parameter estimates as a function of time for noiseless simulated measurements. It can be seen that as time goes on, all methods converge to the actual parameter values (horizontal lines) used to produce the noiseless simulated measurements. Figures 3.1b and 3.1d show the error associated with the parameter estimates. The most accurate parameter estimate is produced by the SCKF (green), but overall the accuracy in all four methods is comparable. This is also evident from the time-averaged estimates and relative errors for the parameters k_p and D given in Table 3.2.2. Note that the averaging is done over the time interval [2,3] in order to minimize the effect of the incorrect initial guess and the possible singular behavior at the end of the simulation when the wound is closing.

Table 3.2.2: Time-averaged estimates on interval [2,3] for k_p and D using noiseless simulated data.

	k_p			D		
	Mean	Rel.Error	Std.Dev.	Mean	Rel.Error	Std.Dev.
EnKF	0.9990	0.10 %	0.0028	2.9999e-06	0.003 %	1.9551e-08
SCKF	1.0001	0.01 %	9.2122e-06	2.9995e-06	0.010 %	3.8195e-10
KLSCKF	1.0018	0.18 %	6.1535e-04	2.9759e-06	0.803 %	8.0139e-09
KLEnKF	1.0006	0.06 %	0.0052	3.0128e-06	0.426 %	3.1334e-08

3.2.2 Noisy Simulated Measurements

Figures 3.2a and 3.2c show the Kalman filter parameter estimates as a function of time for the noisy simulated measurements. All methods converge to the actual parameter values used. The convergence, however, is not nearly as tight as in the cases with noiseless simulated measurements, Figures 3.2b and 3.2d show the error associated with the parameter estimates. The errors for all Kalman filter techniques are approximately the same. In this case, the accuracy of the parameter estimation techniques is limited by the noise in the measurements. As it can be seen in Table 3.2.3, the relative errors in the time-averaged mean estimates are slightly larger than in the noiseless measurement case.

3.2.3 Real Measurements

Figures 3.3a and 3.3b show the Kalman filter parameter estimates as a function of time for real data measurements. For comparison we have used the parameters obtained by direct optimization (dashed line) as a best estimate. The time-averaged estimates over time interval [2,3] for all five techniques are given in Table 3.2.4.

We observe that the KL techniques are able to obtain parameter estimates that are in agreement with the direct parameter estimation technique for both the proliferation rate and the effective diffusion rate. As opposed to the two previous simulated measurement cases, the parameter estimates obtained using the parameter estimation techniques with the real

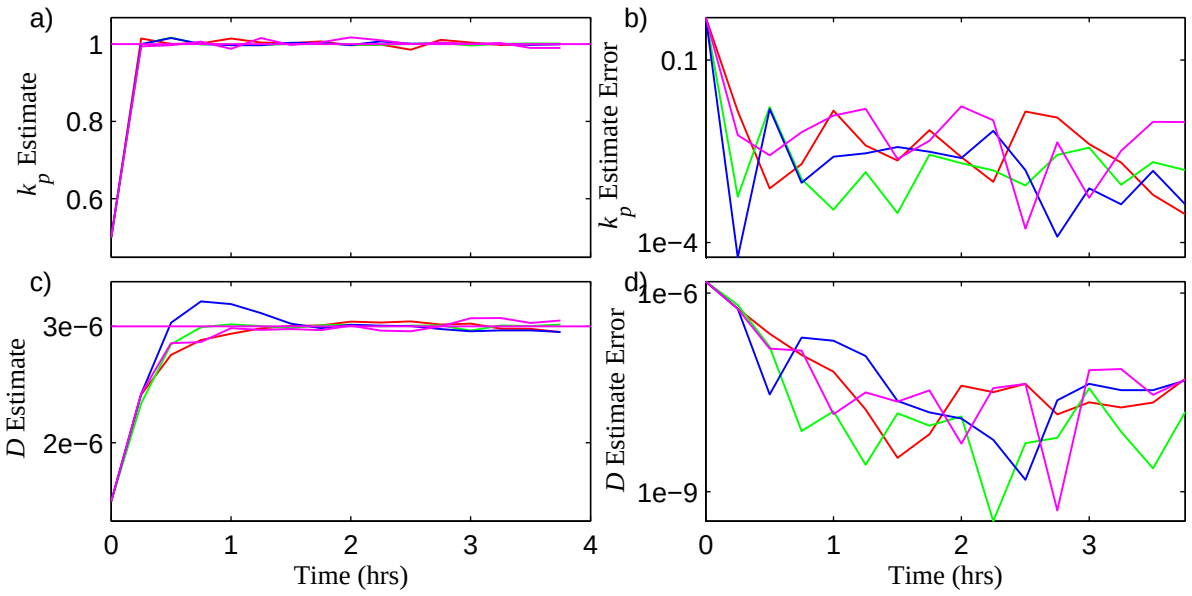


Figure 3.2: Parameter estimates and errors using noisy simulated data for the EnKF(red), SCKF(green), KLSCKF(blue) and KLEnKF(magenta).

Table 3.2.3: Time-averaged estimates on interval [2,3] for k_p and D using noisy simulated data.

	k_p			D		
	Mean	Rel.Error	Std.Dev.	Mean	Rel.Error	Std.Dev.
EnKF	0.9963	0.37 %	0.0046	2.9930e-06	0.23 %	1.8997e-08
SCKF	1.0002	0.02 %	0.0017	3.0098e-06	0.32 %	1.1484e-08
KLSCKF	1.0015	0.15 %	0.0016	2.9817e-06	0.61 %	2.3084e-08
KLEnKF	0.9997	0.03 %	0.0063	3.0149e-06	0.49 %	3.7723e-08

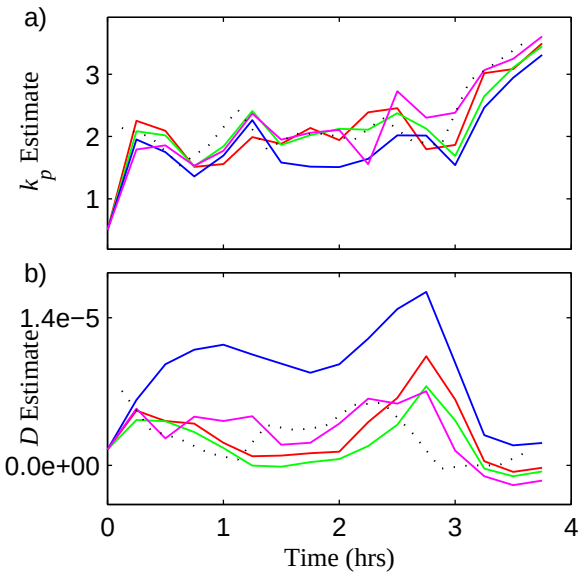


Figure 3.3: Parameter estimation using real data for the EnKF(red), SCKF(green), KLSCKF(blue), KLEnKF(magenta), and Direct Optimization (dashed line).

Table 3.2.4: Time-averaged estimates on interval [2,3] for k_p and D using real data.

	k_p		D	
	Mean	Std.Dev.	Mean	Std.Dev.
En KF	2.0275	0.2511	5.2521e-06	2.0617e-06
SC KF	2.0848	0.2230	3.6097e-06	2.3661e-06
KL SC KF	1.7446	0.2261	1.2520e-05	2.7342e-06
KL En KF	2.0176	0.2258	5.9341e-06	2.9548e-06
Direct Optimization	2.0077	0.1715	3.6211e-06	2.5226e-06

measurements do not converge to one value. Rather, they appear to depend on time. There is clearly a significant variation toward the end of the simulation. This could be explained by the fact that the wound is almost closed at the end, see Figures 3.4 and 3.5 and the data becomes less reliable. It is also possible that the current model may not completely capture all dynamics of the system. One may consider the potential need to eliminate, adjust, or add terms to the model equation.

3.2.4 Matching the experimental results

Here we demonstrate that, using the estimated parameter values, the model produces simulation results that match very well the *in vitro* experiment. We use the parameters estimated by the SCKF since the temporal variance of the SCKF estimates is closest to the temporal variance of the direct optimization technique.

We note that the parameter estimate at $t = 0$ corresponds to the initial guess for the parameters and does not incorporate any data information into that parameter estimate. Additionally, as seen in Figure 3.3, as the wound closes, the parameter estimates begin to change more rapidly with respect to time. For these reasons, to obtain a single parameter estimate for the entire time course of the simulation, we take the time-averaged values of the parameter estimates appearing in Figure 3.3 for $2 \leq t \leq 3$. This averaging on the SCKF gives parameter estimates of $D = 3.30 \times 10^{-6} \pm 1.51$ cm^2/h and $k_p = 1.99 \pm 0.25$/h.

The model is then run, without filtering, with $D = 3.30 \times 10^{-6}$ cm^2/h and $k_p = 1.99$/h. The resulting simulation produces Figure 3.4. The figure shows a time sequence of the values of $e_c(x, y, t)$. To obtain an estimate of where the wound edge is, we take the $e_c = 50\%$ contour and project it into the $x - y$ plane.

To compare this wound edge estimate with the actual wound edge seen in experiment, we take the contours obtained and overlay them on the images from the in vitro wound healing experiment in Figure 3.5. It can be seen that using the parameter estimates obtained from the SCKF parameter estimation technique produces results that are in a very good agreement with the experiment.

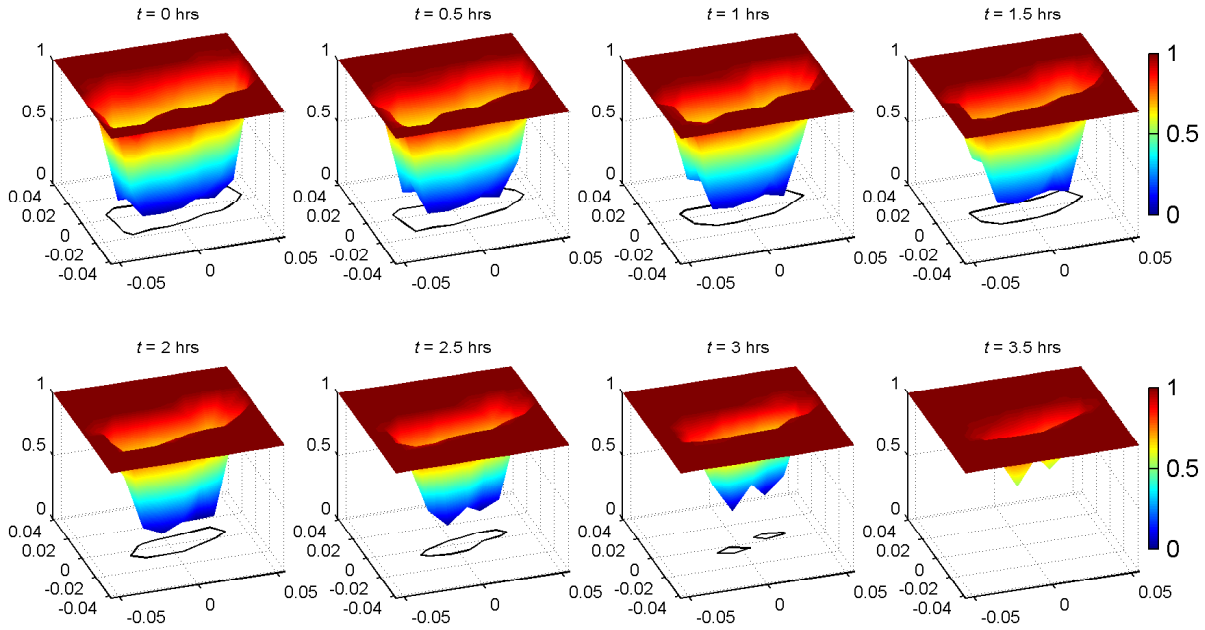


Figure 3.4: Time sequence of surfaces obtained by running the model without any filtering using the parameter estimates of D = 3.30e-6 cm^2/h and k_p=1.99/h.

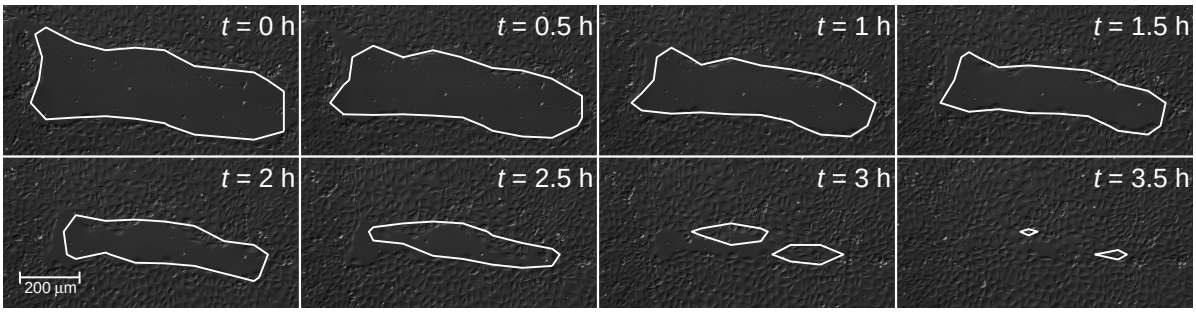


Figure 3.5: Overlay of the 50% contours for e_c obtained by the model and the experimental images.

3.3 DISCUSSION AND CONCLUSIONS

In chapter 3 we developed and analyzed four Kalman filter algorithms for data assimilation and parameter estimation for time dependent nonlinear diffusion equations and compared their performance for a model of epithelial cell migration. The methods are based on either random Monte Carlo sampling (ensemble methods) or structured stochastic collocation

sampling. In addition, either uncorrelated random noise or correlated noise parametrized by the Karhunen-Loève expansion is considered. This results in the methods EnKF, SCKF, KLSCKF, and KLEnKF. The SC methods with sparse grid collocation points provide improved approximation in stochastic space compared to Monte Carlo sampling, and thus result in comparable accuracy with a smaller ensemble size. Furthermore, KL parametrization of the noise results in a stochastic space of smaller dimension (one stochastic dimension per KL term) compared to uncorrelated noise (one stochastic dimension per element of the spatial grid). Consequently, the most efficient method is KLSCKF, followed by SCKF, KLEnKF, and EnKF, see Table 3.2.1.

We compared the performance of the four methods for two cases of simulated measurements, with and without noise, as well as data from *in vitro* experiment of epithelial cell migration. In all cases the four methods exhibited similar accuracy, making the more efficient methods preferable. In the simulated data cases, all methods converged to the correct parameter values for the growth rate k_p and the diffusion D, with small relative errors for the time-averaged mean value estimates. In the real measurements case, all four methods performed comparably to a much more expensive direct optimization simplex method. The methods exhibited certain time variation in the estimated parameters, especially near the end of the simulation. This could be due to singularity in the data when the wound is almost closed, but could also indicate the need to consider a more complex model. Nevertheless using the estimated parameters provided an excellent match of the computed wound shape to the experimental data, as evident from the series of images in Figure 3.5.

4.0 STOCHASTIC OPTIMAL CONTROL FOR ELLIPTIC DIFFUSION EQUATION

4.1 PROBLEM SETTING

Let D be a convex bounded polygonal domain in $\mathbb{R}^d$, $d = 1, 2, 3$, and $(\Omega, \mathcal{F}, P)$ a complete probability space, where Ω is the set of outcomes, $\mathcal{F} \subset 2^\Omega$ is the σ-algebra of events and $P : \mathcal{F} \to [0, 1]$ is a probability measure. The general framework for the stochastic inverse problem is the following: we seek random parameters, coefficients $\kappa(\omega, x)$ and/or forcing terms $f(\omega, x)$, with $x \in D$, $\omega \in \Omega$, that minimize the mismatch between stochastic measured and simulated data. We denote by $W(D)$ a Banach space of functions $v : D \to \mathbb{R}$ and define the stochastic Banach space $L^2_P(\Omega; W(D))$, consisting of Banach valued functions that have finite second moments:

$$L^2_P(\Omega; W(D)) = \left\{ v : \Omega \to W(D) \mid v \text{ is strongly measurable}, \int_\Omega \|v(\omega, \cdot)\|^2_{W(D)} dP(\omega) < +\infty \right\}.$$

4.1.1 State Equations

Soil properties are difficult to measure on the whole spatial domain, therefore the material properties used in the simulation of groundwater flows are usually flawed by uncertainties. There has been recently an increasing interest in the modeling and computational aspects of the uncertainties of the groundwater flow [63, 69, 70] and porous media, see e.g., [54, 55, 72, 65, 71, 74].

We consider the groundwater flow problem in a region $D \subset \mathbb{R}^d$, where the flux is related to the hydraulic head gradient by Darcy's law. We model the uncertainties in the soil by describing the conductivity coefficient κ as a random field denoted $\kappa(\omega, x)$. Similarly, the stochastic

forcing term $f(\omega, x)$ models the uncertainty in the sources and sinks (see, e.g. [47, 73, 53] and the references therein). Therefore the hydraulic head p and velocity u are also random fields satisfying the elliptic stochastic partial differential equation (SPDE):

$$\begin{cases} u(\omega, x) = -\kappa(\omega, x)\nabla p(\omega, x) & \text{in } \Omega \times D, \\ \nabla \cdot u = f & \text{in } \Omega \times D, \\ p = 0 & \text{on } \Omega \times \partial D. \end{cases} \tag{4.1.1}$$

In order to write an appropriate weak formulation for (4.1.1), we need to introduce the Hilbert space (see [58])

$$H(div, D) = \left\{ v \in (L^2(D))^d \mid \nabla \cdot v \in L^2(D) \right\}$$

with the corresponding norm

$$\|v\|_{H(div,D)} = (\|v\|^2_{L^2(D)} + \|\nabla \cdot v\|^2_{L^2(D)})^{1/2}.$$

Currently, numerical methods for Darcy flow consider two different approaches: the first one using the primal, single-phase formulation for pressure, which involves solving a Poisson equation for pressure, and the second one using a mixed, two-phase formulation, with velocity and pressure as the variables of interest.

We will now make the following assumptions concerning the abstract state equations given by (4.1.1):

A_1) the solution u, p to (4.1.1) has realizations in the Banach spaces $H(div, D)$ and $L^2(D)$ respectively, i.e., $u(\omega, \cdot) \in H(div, D)$, $p(\omega, \cdot) \in L^2(D)$ almost surely and $\forall \omega \in \Omega$

$$\|u(\omega, \cdot)\|_{H(div,D)} + \|p(\omega, \cdot)\|_{L^2(D)} \leq C\|f(\omega, \cdot)\|_{L^2(D)}$$

where C is a constant independent of the realization $\omega \in \Omega$.

A_2) the forcing term $f \in L^2_P(\Omega; L^2(D))$ is such that the solution u, p is unique and bounded in $L^2_P(\Omega; H(div, D))$ and $L^2_P(\Omega; L^2(D))$ respectively.

The linear elliptic SPDE (4.1.1) with $\kappa(\omega,\cdot)$ uniformly bounded and coercive, i.e., there exists $\kappa_{min} > 0$ and $\kappa_{max} < \infty$ such that

$$P\left[\omega \in \Omega : \ \kappa_{min} \leq \kappa(\omega, x) \leq \kappa_{max} \, \forall x \in \overline{D}\right] = 1, \tag{4.1.2}$$

and $f(\omega,\cdot)$ square integrable with respect to P, satisfies assumptions A_1 and A_2 (see [33, 34]). We shall assume that D is a bounded and open subset of $\mathbb{R}^d$, either with smooth boundary (of class C^2 for instance) or convex. This implies that for every $f \in L^2_P(\Omega; L^2(D))$, problem (4.1.1) has a unique solution $(u,p) \in L^2_P(\Omega; H(div,D)) \times L^2_P(\Omega; L^2(D))$.
Throughout this chapter, the expected value of a random variable $X(\omega)$ with probability density function (p.d.f.) ρ will be denoted

$$\mathbb{E}[X] = \int_\Omega X(\omega) dP(\omega) = \int_{\mathbb{R}} x\rho(x) dx.$$

The usual multiplication by test functions $v \in H(div,D)$ and $w \in L^2(D)$ and subsequent application of Green's Theorem in the system (4.1.1) yield the standard weak mixed formulation, namely: find $u(\omega,x) \in L^2_P(\Omega; H(div,D))$ and $p(\omega,x) \in L^2_P(\Omega; L^2(D))$ such that

$$\begin{cases} \mathbb{E}\left[\left(\kappa^{-1}u, v\right)_{(L^2(D))^d} - \left(p, \nabla\cdot v\right)_{L^2(D)}\right] = 0, & \forall v \in H(div,D) \\ \mathbb{E}\left[\left(\nabla\cdot u, w\right)_{L^2(D)}\right] = \mathbb{E}\left[\left(f, w\right)_{L^2(D)}\right], & \forall w \in L^2(D). \end{cases} \tag{4.1.3}$$

Throughout the rest of this chapter, for simplicity of notation, the inner product in $L^2(D)$ or $(L^2(D))^d$ will be denoted by $(\cdot,\cdot)$.

4.2 GENERALIZED STOCHASTIC INVERSE PROBLEMS

First we define the admissible set of conductivity coefficients given by

$$\mathcal{A}_{ad} = \{\kappa \in L^\infty(\Omega; L^\infty(D)) \mid \kappa(\omega, x) \text{ satisfies (4.1.2)}\}, \tag{4.2.1}$$

then given $\kappa \in \mathcal{A}_{ad}$ let the admissible set of states and controls be defined as

$$\mathcal{B}_{ad} = \left\{(u, p, f) \mid u \in L^2_P(\Omega; H(div, D)), \quad p \in L^2_P(\Omega; L^2(D)) \text{ and } f \in L^2_P(\Omega; L^2(D))\right\}. \tag{4.2.2}$$

Finally, given $f \in L^2_P(\Omega; L^2(D))$ let the admissible set of states and coefficients be described as

$$\mathcal{C}_{ad} = \left\{(u, p, \kappa) \mid u \in L^2_P(\Omega; H(div, D)), \quad p \in L^2_P(\Omega; L^2(D)) \text{ and } \kappa \in \mathcal{A}_{ad}\right\}. \tag{4.2.3}$$

We also introduce the stochastic target functions $\overline{p} \in L^2_P(\Omega; L^2(D))$ a given possible perturbed observation of the pressure, and $\overline{u} \in L^2_P(\Omega; H(div, D))$ a given possible perturbed observation of the Darcy velocity.

4.2.1 Stochastic optimal control problems

In this section we consider a general class of minimization problems for solving the stochastic inverse problem for the random forcing function $f(\omega, x)$ and the solution $(u(\omega, x), p(\omega, x))$ satisfying a.s. (4.1.1). Here we assume as given the input random process $\kappa \in \mathcal{A}_{ad}$ and the targets $\overline{p} \in L^2_P(\Omega; L^2(D))$ and $\overline{u} \in L^2_P(\Omega; H(div, D))$ and we want to recover (u^*_J, p^*_J, f^*_J) such that

$$(u^*_J, p^*_J, f^*_J) = \inf_{(u,p,f)\in\mathcal{B}_{ad}} \{J(u, p, f) : \text{ subject to (4.1.1)}\} \tag{4.2.4}$$

where $J(u, p, f)$ is a given stochastic functional constructed to track the desired random fields $(\overline{u}, \overline{p})$ or the statistical quantities of interest (QoI) of such stochastic functions. This leads to the following definition.

Definition 1 (Stochastic optimal control)**.** *A 3-tuple $(u^*_J, p^*_J, f^*_J) \in \mathcal{B}_{ad}$ satisfying (4.1.1) a.s., for which the infimum in (4.2.4) is attained is called the stochastic optimal solution and the control f^*_J is referred as stochastic optimal control.*

In what follows we will describe two functionals, denoted $J_1(u,p,f)$ and $J_2(u,p,f)$, used to solve stochastic optimal control problems. The first functional, defined in (4.2.5), is based on the standard classical approach based on stochastic least squares approximation. The second functional, defined in (4.2.8), uses statistical tracking objectives and is easily generalizable for higher order moments, similarly to (4.2.20). We will derive the corresponding adjoint equations, state the necessary conditions for existence and uniqueness of the stochastic optimal solution and prove the necessary conditions for optimality.

4.2.1.1 The optimal control problem using stochastic least squares minimization

For $\kappa \in \mathcal{A}_{ad}$ given data, we consider the following optimal control problem associated with a stochastic elliptic boundary value problem:

$$\text{(P.1)} \quad \begin{cases} \textit{Minimize the cost functional} \\ \\ J_1(u,p,f) = \mathbb{E}\left[\frac{1}{2}\|u(\omega,\cdot)-\overline{u}(\omega,\cdot)\|^2_{(L^2(D))^d} + \frac{1}{2}\|p(\omega,\cdot)-\overline{p}(\omega,\cdot)\|^2_{L^2(D)}\right] \\ \qquad\qquad +\mathbb{E}\left[\frac{\alpha}{2}\|f(\omega,\cdot)\|^2_{L^2(D)}\right], \\ \\ \textit{on all } (u,p,f) \in \mathcal{B}_{ad} \textit{ subject to the stochastic mixed state equations (4.1.1).} \end{cases} \tag{4.2.5}$$

Using standard techniques (see e.g. [66, 67, 48, 51, 68, 61, 60, 56]) one can prove that the problem (4.2.5)-(4.1.1) has a unique optimal solution that is characterized by a maximum principle type result.

We introduce the co-state elliptic equations, written in weak mixed form:

$$\begin{cases} \mathbb{E}\left[(\kappa^{-1}q, v) - (z, \nabla\cdot v)\right] = & -\mathbb{E}\left[(u-\overline{u}, v)\right], \qquad \forall v \in H(div, D), \\ \mathbb{E}\left[(\nabla\cdot q, w)\right] = & \mathbb{E}\left[(p-\overline{p}, w)\right], \qquad \forall w \in L^2(D). \end{cases} \tag{4.2.6}$$

We now state the necessary conditions for optimality in problem (P.1).

Proposition 1. *$(\widehat{u},\widehat{p},\widehat{f})$ is the unique optimal solution in problem (4.2.5)-(4.1.3) if and only if there exists a co-state $(q,z)\in L^2_P(\Omega;H(div,D))\times L^2_P(\Omega;L^2(D))$ such that $(\widehat{u},\widehat{p},\widehat{f},q,z)$ satisfies the following optimality conditions:*

$$\begin{cases} \mathbb{E}\left[(\kappa^{-1}\widehat{u},v)-(\widehat{p},\nabla\cdot v)\right] = 0, & \forall v\in H(div,D)\\ \mathbb{E}\left[(\nabla\cdot\widehat{u},w)\right] = \mathbb{E}\left[(\widehat{f},w)\right], & \forall w\in L^2(D)\\ \mathbb{E}\left[(\kappa^{-1}q,v)-(z,\nabla\cdot v)\right] = -\mathbb{E}\left[(\widehat{u}-\overline{u},v)\right], & \forall v\in H(div,D),\\ \mathbb{E}\left[(\nabla\cdot q,w)\right] = \mathbb{E}\left[(\widehat{p}-\overline{p},w)\right], & \forall w\in L^2(D)\\ \mathbb{E}\left[(z+\alpha\widehat{f},f_s-\widehat{f})\right]\geq 0, & \forall(\widehat{u},\widehat{p},f_s)\in\mathcal{B}_{ad}. \end{cases} \tag{4.2.7}$$

The proof of this result follows in similar manner with the next result, Theorem 1. We note that it is possible to solve the coupled optimality system in one-shot, see e.g. [66].

4.2.1.2 The optimal control problem utilizing statistical tracking objectives

Now we aim at matching expected values, i.e., we consider the following problem:

$$\text{(P.2)}\quad \begin{cases} \textit{Minimize the cost functional}\\ \\ J_2(u,p,f)=\dfrac{1}{2}\|\mathbb{E}u(\cdot,x)-\mathbb{E}\overline{u}(\cdot,x)\|^2_{(L^2(D))^d}+\dfrac{1}{2}\|\mathbb{E}p(\cdot,x)-\mathbb{E}\overline{p}(\cdot,x)\|^2_{L^2(D)}\\ \qquad\qquad +\frac{\alpha}{2}\int_D\mathbb{E}f^2(\cdot,x)dx,\\ \\ \textit{on all } (u,p,f)\in\mathcal{B}_{ad} \textit{ subject to the stochastic mixed state equations (4.1.1).} \end{cases} \tag{4.2.8}$$

Remark 1. *Note that we have*

$$\int_D\left[\mathbb{E}u(\cdot,x)-\mathbb{E}\overline{u}(\cdot,x)\right]^2dx\leq\mathbb{E}\Big(\|u-\overline{u}\|^2_{L^2(D)}\Big),$$
$$\int_D\left[\mathbb{E}p(\cdot,x)-\mathbb{E}\overline{p}(\cdot,x)\right]^2dx\leq\mathbb{E}\Big(\|p-\overline{p}\|^2_{L^2(D)}\Big),$$

which justifies the functional (4.2.8).

Theorem 1. *The 3-tuple $(\widetilde{u},\widetilde{p},\widetilde{f})$ is the unique optimal solution in problem (4.2.8)-(4.1.3) if and only if there exists a co-state $(q,z)\in L^2_P(\Omega;H(div,D))\times L^2_P(\Omega;L^2(D))$ such that $(\widetilde{u},\widetilde{p},\widetilde{f},q,z)$ satisfies the following optimality conditions:*

$$\begin{cases} \mathbb{E}\left[(\kappa^{-1}\widetilde{u},v)-(\widetilde{p},\nabla\cdot v)\right]= & 0, & \forall v\in H(div,D)\\ \mathbb{E}\left[(\nabla\cdot\widetilde{u},w)\right]= & \mathbb{E}\left[(\widetilde{f},w)\right], & \forall w\in L^2(D)\\ \mathbb{E}\left[(\kappa^{-1}q,v)-(z,\nabla\cdot v)\right]= & -\mathbb{E}\left[(\mathbb{E}\widetilde{u}-\mathbb{E}\overline{u},v)\right], & \forall v\in H(div,D),\\ \mathbb{E}\left[(\nabla\cdot q,w)\right]= & \mathbb{E}\left[(\mathbb{E}\widetilde{p}-\mathbb{E}\overline{p},w)\right], & \forall w\in L^2(D)\\ \mathbb{E}\left[(z+\alpha\widetilde{f},f_s-\widetilde{f})\right]\geq & 0, & \forall(\widetilde{u},\widetilde{p},f_s)\in\mathcal{B}_{ad}. \end{cases} \tag{4.2.9}$$

Proof. The sensitivity equations corresponding to the state equations (4.2.6) are

$$\begin{cases} \mathbb{E}\left[(\kappa^{-1}u_s,v)-(p_s,\nabla\cdot v)\right]= & 0, & \forall v\in H(div,D)\\ \mathbb{E}\left[(\nabla\cdot u_s,w)\right]= & \mathbb{E}\left[(f_s,w)\right], & \forall w\in L^2(D), \end{cases} \tag{4.2.10}$$

where $f_s\in L^2_P(\Omega,L^2(D))$, $p_s\in L^2_P(\Omega,L^2(D))$ and $u_s\in L^2_P(\Omega,H(div,D))$. Then the optimality condition for problem (4.2.8) writes

$$0\leq\frac{dJ_2(u|_{\widetilde{f}},p|_{\widetilde{f}},\widetilde{f})}{df}f_s\equiv\frac{dJ_2(\widetilde{u},\widetilde{p},\widetilde{f})}{d(u,p,f)}(u_s,p_s,f_s) \tag{4.2.11}$$

$$=\int_D\mathbb{E}[u_s(\cdot,x)]\mathbb{E}[\widetilde{u}(\cdot,x)-\overline{u}(\cdot,x)]dx+\int_D\mathbb{E}[p_s(\cdot,x)]\mathbb{E}[\widetilde{p}(\cdot,x)-\overline{p}(\cdot,x)]dx+\alpha\int_D\mathbb{E}[\widetilde{f}f_s]dx$$

$$=\int_D\mathbb{E}\Big[u_s(\cdot,x)\mathbb{E}[\widetilde{u}(\cdot,x)-\overline{u}(\cdot,x)]\Big]dx+\int_D\mathbb{E}\Big[p_s(\cdot,x)\mathbb{E}[\widetilde{p}(\cdot,x)-\overline{p}(\cdot,x)]\Big]dx+\alpha\int_D\mathbb{E}\Big[\widetilde{f}f_s\Big]dx$$

(since $\mathbb{E}[\widetilde{u}(\cdot,x)-\overline{u}(\cdot,x)]$ is deterministic)

$$=\int_D\mathbb{E}\Big[-u_s(\cdot,x)\kappa^{-1}q+z\nabla\cdot u_s\Big]dx+\int_D\mathbb{E}\Big[p_s(\cdot,x)\nabla\cdot q\Big]dx+\alpha\int_D\mathbb{E}\Big[\widetilde{f}f_s\Big]dx$$

(by (4.2.9) with $v=u_s$)

$$=\mathbb{E}\left[\int_D-u_s(\cdot,x)\kappa^{-1}q+z\nabla\cdot u_s dx\right]+\mathbb{E}\left[\int_D p_s(\cdot,x)\nabla\cdot q dx\right]+\alpha\mathbb{E}\left[\int_D\widetilde{f}f_s dx\right]$$

(by Fubini's theorem)

$$=\mathbb{E}\left[\int_D-u_s(\cdot,x)\kappa^{-1}q dx+\int_D\nabla\cdot u_s z dx\right]+\mathbb{E}\left[\int_D\kappa^{-1}u_s(\cdot,x)q dx\right]+\mathbb{E}\left[\int_D\alpha\widetilde{f}f_s dx\right]$$

(by (4.2.10))

$$= \mathbb{E}\left[\int_D f_s z dx\right] + \mathbb{E}\left[\int_D \alpha \widetilde{f} f_s dx\right]$$

(by (4.2.10))

$$= \mathbb{E}\left[\int_D \left(z + \alpha\widetilde{f}\right) f_s dx\right]$$

$$= \mathbb{E}\left[\left(z + \alpha\widetilde{f}, f_s\right)\right], \qquad \forall (u_s, p_s, f_s) \in \operatorname{Tan}\mathcal{B}_{ad}\left(\widetilde{u}, \widetilde{p}, \widetilde{f}\right), \quad z + \alpha\widetilde{f} \in \mathrm{N}_{\mathcal{B}_{ad}},$$

where we have used the fact that $\mathbb{E}\left[\widetilde{u}(\cdot, x) - \overline{u}(\cdot, x)\right]$ is a deterministic quantity, the adjoint equations (4.2.9), Fubini's theorem, the sensitivity equation (4.2.10) and the definition of normal cone. Here $\operatorname{Tan}\mathcal{B}_{ad}$ denotes the tangent cone, while $\mathrm{N}_{\mathcal{B}_{ad}}$ is the normal cone (see [36]). □

The necessary and sufficient conditions (4.2.9) resemble the optimality system (4.2.7), the difference is only in the adjoint equations which have a deterministic right-hand side. Nevertheless, the adjoint variables (q, z) are still stochastic quantities.

4.2.2 Stochastic parameter identification problems

We also study the identification of the coefficient κ in the stochastic boundary value problem (4.1.1). In the deterministic case, for the direct problem, where κ is given, the existence and uniqueness results are well known, see e.g. [64]. The linear deterministic inverse problem related to (4.1.1) has been studied in e.g. [48, 68], for the nonlinear deterministic see e.g. [52].

For the identification problem, we are given possible perturbed observations $\overline{u}$, $\overline{p}$ corresponding to the state variables u, respectively p, and we must determine κ in (4.1.1) such that $u(\kappa) = \overline{u}$ and $p(\kappa) = \overline{p}$ in $\Omega \times D$. Of course, such a κ may not exist.

4.2.2.1 Parameter identification using stochastic least squares minimization

The least squares approach leads us to the minimization problem:

$$\text{(P.3)} \quad \begin{cases} \textit{Minimize the cost functional} \\ \\ J_3(u,p,\kappa) = \mathbb{E}\left[\frac{1}{2}\|u(\omega,\cdot)-\overline{u}(\omega,\cdot)\|^2_{(L^2(D))^d} + \frac{1}{2}\|p(\omega,\cdot)-\overline{p}(\omega,\cdot)\|^2_{L^2(D)}\right] \\ \qquad\qquad +\mathbb{E}\left[\frac{\beta}{2}\|\kappa(\omega,\cdot)\|^2_{L^2(D)}\right], \\ \\ \textit{on all } (u,p,\kappa)\in\mathcal{C}_{ad} \textit{ subject to the stochastic mixed state equations (4.1.1).} \end{cases} \tag{4.2.12}$$

We introduce the co-state elliptic equations for this problem (P.3):

$$\begin{cases} \mathbb{E}\Big[((\kappa^*)^{-1}q,v)-(\eta,\nabla\cdot v)\Big] = \mathbb{E}\Big[-(u^*-\overline{u},v)\Big], \qquad \forall v\in H(div,D) \\ \mathbb{E}\Big[\big(\nabla\cdot q,w\big)\Big] = \mathbb{E}\Big[(p^*-\overline{p},w)\Big], \qquad \forall w\in L^2(D). \end{cases} \tag{4.2.13}$$

Theorem 2. *Let (u^*,p^*,κ^*) be an optimal solution in problem (4.2.12)-(4.1.3). Then there exists a co-state $(q,\eta)\in L^2_P(\Omega;H(div,D))\times L^2_P(\Omega;L^2(D))$ such that $(u^*,p^*,\kappa^*,q,\eta)$ satisfies the following optimality conditions:*

$$\begin{cases} \mathbb{E}\Big[((\kappa^*)^{-1}u^*,v)-(p^*,\nabla\cdot v)\Big] = 0, \qquad \forall v\in H(div,D) \\ \mathbb{E}\Big[(\nabla\cdot u^*,w)\Big] = \mathbb{E}\Big[(f,w)\Big], \qquad \forall w\in L^2(D) \\ \mathbb{E}\Big[((\kappa^*)^{-1}q,v)-(\eta,\nabla\cdot v)\Big] = \mathbb{E}\Big[-(u^*-\overline{u},v)\Big], \qquad \forall v\in H(div,D) \\ \mathbb{E}\Big[\big(\nabla\cdot q,w\big)\Big] = \mathbb{E}\Big[(p^*-\overline{p},w)\Big], \qquad \forall w\in L^2(D) \\ \kappa^*(\omega,x) = \max\{\kappa_{\min},\min\{\frac{1}{\beta}(\kappa^*)^{-2}u^*(\omega,x)q(\omega,x),\kappa_{\max}\}\}, \\ \qquad\qquad \textit{a.e. in } \Omega\times D. \end{cases} \tag{4.2.14}$$

Proof. The sensitivity equations are

$$\begin{cases} \mathbb{E}\Big[\Big((\kappa^*)^{-1}u_s - (\kappa^*)^{-2}\kappa_s u^*, v\Big) - (p_s, \nabla\cdot v)\Big] = 0, & \forall v \in H(div, D) \\ \mathbb{E}\Big[(\nabla\cdot u_s, w)\Big] = 0, & \forall w \in L^2(D), \end{cases} \tag{4.2.15}$$

where $(u_s, p_s, \kappa_s) \in \mathrm{Tan}\,\mathcal{C}_{ad}\big(u^*, p^*, \kappa^*\big)$.

Let $\mathcal{S}_{ad} = \{(u, p, \kappa) \in \mathcal{C}_{ad} : (u, p, \kappa)$ satisfy the state equations (4.1.1)$\}$ be set of admissible states and parameters to problem (4.2.12). We introduce the tangential cone to the set $\mathcal{S}_{ad}$ at $(u, p, \kappa) \in \mathcal{S}_{ad}$

$$\begin{aligned} \mathrm{Tan}\mathcal{S}_{ad}(u, p, \kappa) = \{&(u_s, p_s, \kappa_s) \text{ which satisfy the sensitivity equations (4.2.15)}, \\ &u_s \in L^2_P(\Omega; H(div, D)), p_s \in L^2_P(\Omega; L^2(D)), \kappa_s \in \mathrm{Tan}\mathcal{A}_{ad}\}. \end{aligned} \tag{4.2.16}$$

Recall that if

$$J(u^*, p^*, \kappa^*) = \inf_{(u,p,\kappa)\in\mathcal{S}_{ad}} J(u, p, \kappa)$$

and the functional $J(u, p, \kappa)$ is Gâteaux differentiable, then necessarily

$$\frac{dJ(u^*, p^*, \kappa^*)}{d(u, p, \kappa)}(u_s, p_s, \kappa_s) \geq 0 \text{ for all } (u_s, p_s, \kappa_s) \in \mathrm{Tan}\mathcal{S}_{ad}(u^*, p^*, \kappa^*), \tag{4.2.17}$$

where $\frac{dJ(u^*,p^*,\kappa^*)}{d(u,p,\kappa)} \equiv \frac{dJ(u(\kappa^*),p(\kappa^*),\kappa^*)}{d\kappa}$ stands for the Gâteaux derivative of J at $(u^*, p^*, \kappa^*) \in \mathcal{S}_{ad}$, and $(u^*, p^*, \kappa^*) \equiv (u(\kappa^*), p(\kappa^*), \kappa^*)$. Applying the optimum principle given by (4.2.17) it follows that the optimality condition for problem (4.2.12) writes

$$\begin{aligned} 0 \leq \frac{dJ_3(u(\kappa^*), p(\kappa^*), \kappa^*)}{d\kappa}\kappa_s &\equiv \frac{dJ_3(u^*, p^*, \kappa^*)}{d(u, p, \kappa)}(u_s, p_s, \kappa_s) \\ &= \mathbb{E}\Big[\int_D u_s(\cdot, x)\Big(u^*(\cdot, x) - \overline{u}(\cdot, x)\Big)dx\Big] + \mathbb{E}\Big[\int_D p_s(\cdot, x)\Big(p^*(\cdot, x) - \overline{p}(\cdot, x)\Big)dx\Big] \\ &+ \mathbb{E}\Big[\beta\int_D \kappa^*(\cdot, x)\kappa_s(\cdot, x)dx\Big] \\ &= \mathbb{E}\Big[-\int_D u_s(\cdot, x)(\kappa^*)^{-1}(\cdot, x)q(\cdot, x) + \eta(\cdot, x)\nabla\cdot u_s(\cdot, x)dx\Big] + \mathbb{E}\Big[\int_D p_s(\cdot, x)\nabla\cdot q(\cdot, x)dx\Big] \\ &+ \beta\mathbb{E}\Big[\int_D \kappa^*(\cdot, x)\kappa_s(\cdot, x)dx\Big] \qquad \text{(by (4.2.13) with } v = u_s \text{ and } w = p_s) \end{aligned}$$

$$
\begin{aligned}
&= \mathbb{E}\Big[-\int_D u_s(\cdot,x)(\kappa^*)^{-1}(\cdot,x)q(\cdot,x)dx\Big] + \mathbb{E}\Big[\int_D p_s(\cdot,x)\nabla\cdot q(\cdot,x)dx\Big] \\
&+ \beta\mathbb{E}\Big[\int_D \kappa^*(\cdot,x)\kappa_s(\cdot,x)dx\Big] \qquad \text{(by (4.2.15) with } w=\eta) \\
&= \mathbb{E}\Big[-\int_D (\kappa^*)^{-2}(\cdot,x)\kappa_s(\cdot,x)u^*(\cdot,x)q(\cdot,x)dx - \int_D p_s(\cdot,x)\nabla\cdot q(\cdot,x)dx\Big] \\
&+ \mathbb{E}\Big[\int_D p_s(\cdot,x)\nabla\cdot q(\cdot,x)dx\Big] + \beta\mathbb{E}\Big[\int_D \kappa^*(\cdot,x)\kappa_s(\cdot,x)dx\Big] \qquad \text{(by (4.2.15) with } v=q) \\
&= \mathbb{E}\Big[\int_D \Big(-(\kappa^*)^{-2}(\cdot,x)u^*(\cdot,x)q(\cdot,x) + \beta\kappa^*(\cdot,x)\Big)\kappa_s(\cdot,x)\Big], \quad \forall (u_s,p_s,\kappa_s)\in \operatorname{Tan}\mathcal{B}_{ad}\big(u^*,p^*,\kappa^*\big)
\end{aligned}
$$

where we have used the adjoint equations (4.2.13), the sensitivity equations (4.2.15). □

4.2.2.2 Parameter identification utilizing statistical tracking objectives For the identification problem matching *expected values*, given a possible perturbed observation $(\overline{u},\overline{p})$ corresponding to the state variables u, p, we seek κ in (4.1.1) such that $\mathbb{E}u(\kappa) = \mathbb{E}\overline{u}$ and $\mathbb{E}p(\kappa) = \mathbb{E}\overline{p}$ in D. Therefore we consider the problem:

$$
\text{(P.4)} \quad \begin{cases} \textit{Minimize the cost functional} \\ \\ J_4(u,p,\kappa) = \dfrac{1}{2}\displaystyle\int_D \big[\mathbb{E}u(\cdot,x) - \mathbb{E}\overline{u}(\cdot,x)\big]^2 dx + \dfrac{1}{2}\int_D \big[\mathbb{E}p(\cdot,x) - \mathbb{E}\overline{p}(\cdot,x)\big]^2 dx \\ \qquad\qquad + \frac{\beta}{2}\int_D \mathbb{E}\kappa^2(\cdot,x)dx, \\ \\ \textit{on all } (u,p,\kappa)\in\mathcal{C}_{ad} \textit{ subject to the stochastic state equations (4.1.1).} \end{cases} \tag{4.2.18}
$$

Theorem 3. *Let $(\mathring{u},\mathring{p},\mathring{\kappa})$ be an optimal solution in problem (4.1.1) and (4.2.18). Then there exists a co-state $(q,\eta)\in L^2_P(\Omega;H(div,D))\times L^2_P(\Omega;L^2(D))$ such that $(\mathring{u},\mathring{p},\mathring{\kappa},q,\eta)$ satisfies*

the following optimality conditions:

$$\begin{cases} \mathbb{E}\Big[(\mathring{\kappa}^{-1}\mathring{u}, v) - (\mathring{p}, \nabla\cdot v)\Big] = 0, \qquad \forall v \in H(div, D) \\ \mathbb{E}\Big[(\nabla\cdot\mathring{u}, w)\Big] = \mathbb{E}\Big[(f, w)\Big], \qquad \forall w \in L^2(D) \\ \mathbb{E}\Big[(\mathring{\kappa}^{-1}q, v) - (\eta, \nabla\cdot v)\Big] = \mathbb{E}\Big[-(\mathbb{E}\mathring{u} - \mathbb{E}\overline{u}, v)\Big], \qquad \forall v \in H(div, D) \\ \mathbb{E}\Big[\Big(\nabla\cdot q, w\Big)\Big] = \mathbb{E}\Big[(\mathbb{E}\mathring{p} - \mathbb{E}\overline{p}, w)\Big], \qquad \forall w \in L^2(D) \\ \mathring{\kappa}(\omega, x) = \max\{\kappa_{\min}, \min\{\frac{1}{\beta}(\mathring{\kappa})^{-2}\mathring{u}(\omega, x)q(\omega, x), \kappa_{\max}\}\}, \ a.e. \ in \ \Omega\times D. \end{cases} \tag{4.2.19}$$

Proof. See the proof of Theorem 4. □

For the problem of matching *covariance*, and/or *higher order moments*, the cost functional used in problem (4.2.18) can be generalized as follows. Assume we are interested in $\mathcal{L}$-order moments, and $f \in L^{\mathcal{L}}_P(\Omega; L^{2\mathcal{L}-2}(D))$, then

$$\text{(P.5)} \quad \begin{cases} \textit{Minimize the cost functional} \\ J_5(u,p,\kappa) = \sum_{\ell=1}^{\mathcal{L}} \frac{\alpha_{u,\ell}}{2\ell}\int_D \big[\mathbb{E}u^\ell(\cdot,x) - \mathbb{E}\overline{u}^\ell(\cdot,x)\big]^2 dx + \frac{\beta}{2}\int_D \mathbb{E}\kappa^2(\cdot,x)dx + \\ \qquad + \sum_{\ell=1}^{\mathcal{L}} \frac{\alpha_{p,\ell}}{2\ell}\int_D \big[\mathbb{E}p^\ell(\cdot,x) - \mathbb{E}\overline{p}^\ell(\cdot,x)\big]^2 dx \\ \textit{on all } (u,p,\kappa) \in \mathcal{C}_{ad} \textit{ subject to the stochastic state equations } (4.1.1). \end{cases} \tag{4.2.20}$$

We introduce the co-state elliptic equations for this problem (P.5):

$$\begin{cases} \mathbb{E}\Big[((\kappa^*)^{-1}q, v) - (\eta, \nabla\cdot v)\Big] = -\mathbb{E}\Big[\sum_{\ell=1}^{\mathcal{L}} \alpha_{u,\ell}(u^*)^{\ell-1}(\mathbb{E}(u^*)^\ell - \mathbb{E}\overline{u}^\ell, v)\Big], \forall v \in H(div, D) \\ \mathbb{E}\Big[\Big(\nabla\cdot q, w\Big)\Big] = \mathbb{E}\Big[\sum_{\ell=1}^{\mathcal{L}} \alpha_{p,\ell}(p^*)^{\ell-1}(\mathbb{E}(p^*)^\ell - \mathbb{E}\overline{p}^\ell, w)\Big], \forall w \in L^2(D). \end{cases} \tag{4.2.21}$$

Theorem 4. *Let (u^*, p^*, κ^*) be an optimal solution in problem (4.2.20)-(4.1.3). Then there exists a co-state $(q, \eta) \in L^2_P(\Omega; H(div, D)) \times L^2_P(\Omega; L^2(D))$ such that $(u^*, p^*, \kappa^*, q, \eta)$ satisfies the following optimality conditions:*

$$\begin{cases} \mathbb{E}\Big[((\kappa^*)^{-1}u^*, v) - (p^*, \nabla \cdot v)\Big] = 0, \forall v \in H(div, D) \\ \mathbb{E}\Big[(\nabla \cdot u^*, w)\Big] = \mathbb{E}\Big[(f, w)\Big], \forall w \in L^2(D) \\ \mathbb{E}\Big[((\kappa^*)^{-1}q, v) - (\eta, \nabla \cdot v)\Big] = -\mathbb{E}\Big[\sum_{\ell=1}^{\mathcal{L}} \alpha_{u,\ell}(u^*)^{\ell-1}(\mathbb{E}(u^*)^\ell - \mathbb{E}\overline{u}^\ell, v)\Big], \forall v \in H(div, D) \\ \mathbb{E}\Big[\Big(\nabla \cdot q, w\Big)\Big] = \mathbb{E}\Big[\sum_{\ell=1}^{\mathcal{L}} \alpha_{p,\ell}(p^*)^{\ell-1}(\mathbb{E}(p^*)^\ell - \mathbb{E}\overline{p}^\ell, w)\Big], \forall w \in L^2(D) \\ \kappa^*(\omega, x) = \max\{\kappa_{\min}, \min\{\frac{1}{\beta}(\kappa^*)^{-2}u^*(\omega, x)q(\omega, x), \kappa_{\max}\}\}, \\ \qquad\qquad a.e.\ in\ \Omega \times D. \end{cases} \tag{4.2.22}$$

Proof. The sensitivity equations are

$$\begin{cases} \mathbb{E}\Big[\Big((\kappa^*)^{-1}u_s - (\kappa^*)^{-2}\kappa_s u^*, v\Big) - (p_s, \nabla \cdot v)\Big] = 0, \qquad \forall v \in H(div, D) \\ \mathbb{E}\Big[(\nabla \cdot u_s, w)\Big] = 0, \qquad \forall w \in L^2(D), \end{cases} \tag{4.2.23}$$

where $(u_s, p_s, \kappa_s) \in \operatorname{Tan}\mathcal{C}_{ad}\big(u^*, p^*, \kappa^*\big)$. Applying the optimum principle given by (4.2.17) it follows that the optimality condition for problem (4.2.20) writes

$$\begin{aligned} 0 \leq \frac{dJ_5(u(\kappa^*), p(\kappa^*), \kappa^*)}{d\kappa}\kappa_s &\equiv \frac{dJ_5(u^*, p^*, \kappa^*)}{d(u, p, \kappa)}(u_s, p_s, \kappa_s) \\ &= \sum_{\ell=1}^{\mathcal{L}} \int_D \alpha_{u,\ell}\mathbb{E}\Big[u_s(\cdot, x)(u^*)^{\ell-1}(\cdot, x)\Big]\mathbb{E}\Big[(u^*)^\ell(\cdot, x) - \overline{u}^\ell(\cdot, x)\Big]dx \\ &+ \sum_{\ell=1}^{\mathcal{L}} \int_D \alpha_{p,\ell}\mathbb{E}\Big[p_s(\cdot, x)(p^*)^{\ell-1}(\cdot, x)\Big]\mathbb{E}\Big[(p^*)^\ell(\cdot, x) - \overline{p}^\ell(\cdot, x)\Big]dx + \beta\int_D \mathbb{E}\Big[\kappa^*(\cdot, x)\kappa_s(\cdot, x)\Big]dx \\ &= \sum_{\ell=1}^{\mathcal{L}} \int_D \alpha_{u,\ell}\mathbb{E}\Big[u_s(\cdot, x)(u^*)^{\ell-1}(\cdot, x)\mathbb{E}\Big((u^*)^\ell(\cdot, x) - \overline{u}^\ell(\cdot, x)\Big)\Big]dx \\ &+ \sum_{\ell=1}^{\mathcal{L}} \int_D \alpha_{p,\ell}\mathbb{E}\Big[p_s(\cdot, x)(p^*)^{\ell-1}(\cdot, x)\mathbb{E}\Big((p^*)^\ell(\cdot, x) - \overline{p}^\ell(\cdot, x)\Big)\Big]dx + \beta\int_D \mathbb{E}\Big[\kappa^*(\cdot, x)\kappa_s(\cdot, x)\Big]dx \\ &= \int_D \mathbb{E}\Big[-u_s(\cdot, x)(\kappa^*)^{-1}(\cdot, x)q(\cdot, x) + \eta(\cdot, x)\nabla \cdot u_s(\cdot, x)\Big]dx + \int_D \mathbb{E}\Big[p_s(\cdot, x)\nabla \cdot q(\cdot, x)\Big]dx \end{aligned}$$

$$
\begin{aligned}
&+\beta\int_D \mathbb{E}\Big[\kappa^*(\cdot,x)\kappa_s(\cdot,x)\Big]dx \qquad \text{(by (4.2.21) with } v=u_s \text{ and } w=p_s\text{)}\\
&=\mathbb{E}\Big[\int_D -u_s(\cdot,x)(\kappa^*)^{-1}(\cdot,x)q(\cdot,x)+\eta(\cdot,x)\nabla\cdot u_s(\cdot,x)dx\Big]+\mathbb{E}\Big[\int_D p_s(\cdot,x)\nabla\cdot q(\cdot,x)dx\Big]\\
&+\beta\mathbb{E}\Big[\int_D \kappa^*(\cdot,x)\kappa_s(\cdot,x)dx\Big]\\
&=\mathbb{E}\Big[-\int_D u_s(\cdot,x)(\kappa^*)^{-1}(\cdot,x)q(\cdot,x)dx\Big]+\mathbb{E}\Big[\int_D p_s(\cdot,x)\nabla\cdot q(\cdot,x)dx\Big]\\
&+\beta\mathbb{E}\Big[\int_D \kappa^*(\cdot,x)\kappa_s(\cdot,x)dx\Big] \qquad \text{(by (4.2.23) with } w=\eta\text{)}\\
&=\mathbb{E}\Big[-\int_D (\kappa^*)^{-2}(\cdot,x)\kappa_s(\cdot,x)u^*(\cdot,x)q(\cdot,x)dx-\int_D p_s(\cdot,x)\nabla\cdot q(\cdot,x)dx\Big]\\
&+\mathbb{E}\Big[\int_D p_s(\cdot,x)\nabla\cdot q(\cdot,x)dx\Big]+\beta\mathbb{E}\Big[\int_D \kappa^*(\cdot,x)\kappa_s(\cdot,x)dx\Big] \qquad \text{(by (4.2.23) with } v=q\text{)}\\
&=\mathbb{E}\Big[\int_D \Big(-(\kappa^*)^{-2}(\cdot,x)u^*(\cdot,x)q(\cdot,x)+\beta\kappa^*(\cdot,x)\Big)\kappa_s(\cdot,x)\Big], \quad \forall(u_s,p_s,\kappa_s)\in \operatorname{Tan}\mathcal{B}_{ad}\big(u^*,p^*,\kappa^*\big)
\end{aligned}
$$

where we have used the adjoint equations (4.2.21), the sensitivity equations (4.2.23). □

4.3 NUMERICAL EXPERIMENTS

Now I briefly present the algorithm used, together with the derivation of the gradient algorithms.

4.3.1 Sensitivity Analysis for the Parameter Estimation in the Elliptic Case

Consider the state equations:

$$
\begin{cases}
u=-\kappa\nabla p & \text{in } \Omega\times D,\\
\nabla\cdot u=f & \text{in } \Omega\times D,\\
p=0 & \text{on } \Omega\times\partial D.
\end{cases}
\tag{4.3.1}
$$

We introduce the adjoint equations:

$$
\begin{cases}
\nabla\cdot q=p-\overline{p} & \text{in } \Omega\times D,\\
\kappa^{-1}q+\nabla\eta=-(u-\overline{u}) & \text{in } \Omega\times D,\\
\eta=0 & \text{on } \Omega\times\partial D.
\end{cases}
\tag{4.3.2}
$$

Define the cost functional:

$$J_3(Y_i) = \frac{1}{2}\mathbb{E}\left[\|u(Y_i,\cdot) - \overline{u}(Y_i,\cdot)\|^2_{L^2(D)}\right] + \frac{1}{2}\mathbb{E}\left[\|p(Y_i,\cdot) - \overline{p}(Y_i,\cdot)\|^2_{L^2(D)}\right] + \frac{\beta}{2}\mathbb{E}\left[\|\kappa(Y_i,\cdot)\|^2_{L^2(D)}\right],$$

$\forall i = 1 \dots N$.

We assume that the map $\kappa \to u$ is differentiable. Then, the sensitivity equations are the following:

$$\begin{cases} u + \epsilon u_s = -\left((\kappa + \epsilon\kappa_s)\nabla(p + \epsilon p_s)\right) & \text{in } \Omega \times D, \\ \nabla \cdot (u + \epsilon u_s) = f & \text{in } \Omega \times D, \\ p_s = 0 & \text{on } \Omega \times \partial D. \end{cases} \tag{4.3.3}$$

Multiplying out and using the state equations (4.3.1), we get:

$$\begin{cases} \kappa^{-1}u_s - \kappa^{-2}\kappa_s u = -\nabla p_s & \text{in } \Omega \times D, \\ \nabla \cdot u_s = 0 & \text{in } \Omega \times D, \\ p_s = 0 & \text{on } \Omega \times \partial D. \end{cases} \tag{4.3.4}$$

Next, we multiply by the adjoint variables q, η and integrate over D:

$$\begin{cases} \int_D \kappa^{-1}u_s q - \int_D \kappa^{-2}\kappa_s u q = -\int_D \nabla p_s q & \text{in } \Omega \times D, \\ \int_D \nabla \cdot u_s \eta = 0 & \text{in } \Omega \times D, \\ p_s = 0 & \text{on } \Omega \times \partial D. \end{cases} \tag{4.3.5}$$

Integrate by parts:

$$\begin{cases} \int_D \kappa^{-1}u_s q - \int_D \kappa^{-2}\kappa_s u q = \int_D p_s \nabla \cdot q & \text{in } \Omega \times D, \\ \int_D u_s \nabla \eta = 0 & \text{in } \Omega \times D, \\ p_s = 0 & \text{on } \Omega \times \partial D. \end{cases} \tag{4.3.6}$$

There are no boundary terms since $\eta = 0$ and $p_s = 0$ on $\Omega \times \partial D$.

Take expectation and add the first two identities of (4.3.6):

$$\mathbb{E}\left[\int_D u_s\left(\kappa^{-1}q + \nabla\eta\right)\right] - \mathbb{E}\left[\int_D \kappa^{-2}\kappa_s u q\right] = \mathbb{E}\left[\int_D p_s \nabla \cdot q\right].$$

Using the adjoint equations (4.3.2), the previous identity becomes:

$$-\mathbb{E}\left[\int_D u_s(u-\overline{u})\right] - \mathbb{E}\left[\int_D \kappa^{-2}\kappa_s uq\right] = \mathbb{E}\left[\int_D p_s(p-\overline{p})\right].$$

We give below the pseudocode for the **Adjoint variable-based Algorithm**, as in [75]:

INITIALIZATION: $i \leftarrow 1$, $RelError \leftarrow 1000$, Choose initial conditions for Y, $\epsilon = 1$, $\epsilon \leftarrow 2\epsilon/3$

while $RelError > tol$ **do**

$\epsilon \leftarrow 3\epsilon/2$, $i \leftarrow i+1$

Solve Adjoint Equations (for the adjoint variables)

Solve Standard Gradient Update, i.e. $Y_{i+1} = Y_i - \epsilon\frac{dJ}{dY_i}$

Solve State Equations

Evaluate $J_n(i)$

while $J_n^{(i)} > J_n^{(i-1)}$ **do**

$\epsilon \leftarrow \epsilon/10$

Solve Standard Gradient Update

Solve State Equations

Evaluate $J_n(i)$

end while

$RelError \leftarrow \left|J_n^{(i)} - J_n^{(i-1)}\right| / \left|J_n^{(i)}\right|$

end while

The numerical experiments were performed using MATLAB R2012a and were solved on a square domain $[0,1] \times [0,1]$. The convergence is computed on a 40×40 spatial mesh, with Dirichlet boundary conditions. For solving the equation (4.3.1) numerically, an upwind scheme is used to find the effective diffusion coefficient and central difference for finding the hydraulic gradient. We assume the true random diffusion coefficient $\overline{\kappa}$ and the exact solution $\overline{p}$ to be given by:

$$\overline{\kappa}(\omega, x) = (1 + x^2 + y^2) + \frac{1}{N} * \sum_{n=1}^{N} \cos(n\pi x) \cdot \cos(n\pi y)\, Y_n(\omega)$$

$$\overline{p}(\omega, x) = \sum_{n=1}^{N} \sin(n\pi x) \cdot \sin(n\pi y)\, Y_n(\omega)$$

and then we calculate the source $f(\omega, x)$.

To understand the dynamics that the computational model produces, we first present some sample simulations. The tolerance was taken 10^{-4}, the step size for the adjoint algorithm is $\epsilon = 1$ and the coefficient $\beta = 10^{-6}$ in the cost functional formula.

4.3.2 Numerical Experiments for the Deterministic Elliptic Case

The exact values used for producing simulated measurements were 0.5 for the $Y_i, i = 1, \ldots, N$ in the formula for the diffusion coefficient k. Figures 4.1(a) and 4.1(b) show the plot for the cost functional J and the logarithm of J to base 10. The trajectories of the $N = 5$ Ys, the crossection of target solution versus estimated solution, the crossection of target diffusion versus estimated diffusion are presented in figures 4.1(c), (d) and (e) respectively.

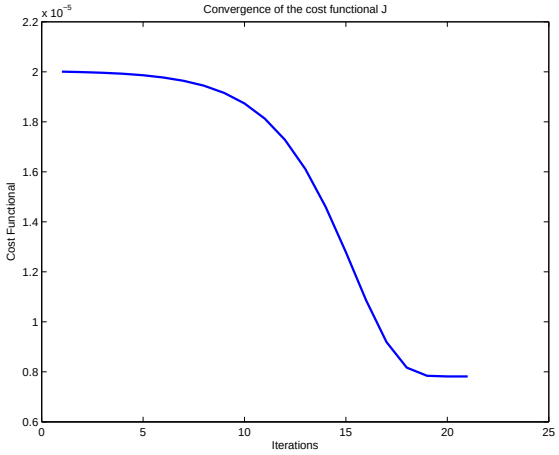
4.3.3 Numerical Experiments for the Stochastic Elliptic Case

The exact values used for producing simulated measurements were considered uniformly distributed random numbers for the $Y_i, i = 1, \ldots, n$. To understand the dynamics that the computational model produces, we present sample simulations on a 40x40 spatial mesh, where we first considered 10 realizations (see Figures 4.2 and 4.3) and then 50 realizations (see Figures 4.4 and 4.5) for our stochastic model.

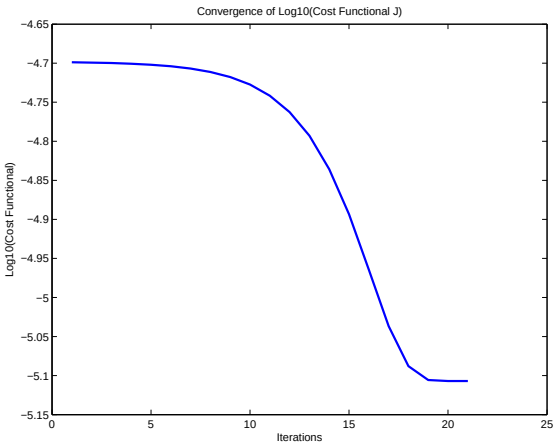
In the **first simulation** where only 10 realizations were considered, we observed that for achieving the same tolerance of 10^{-4} for the relative error, the cost functional J_3 only needs to do 27 iterations, whereas J_4 and J_5 require 76 and 61 iterations respectively. By plotting crossections, we observed that our estimated solutions corresponding to either J_3, J_4 or J_5 approximate very well the mean of the target solution, while the variance of the target solution is better approximated when using the solution corresponding to J_3 cost functional. When considering the crossections for the diffusion coefficient, the mean and variance of our estimated diffusion were not doing so well in approximating the mean and variance of the target diffusion coefficient. One explanation would be the fact that our cost functionals try to

minimize the difference between the estimated and target solutions, whereas the difference between the estimated and target diffusion coefficient is never taken into account in the formulas of the cost functionals.

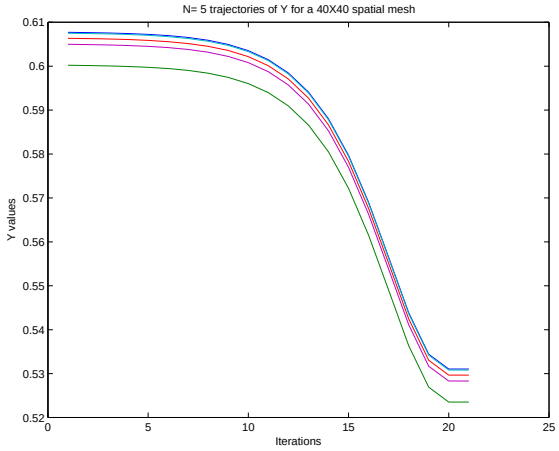
In the **second simulation** with 50 realizations being considered, the cost functional J_3 only needs 22 iterations, whereas J_4 and J_5 require 47 and 95 iterations respectively. By plotting crossections, again it was observed that our estimated solutions corresponding to either J_3, J_4 or J_5 approximate really well the mean of the target solution, whereas for the variance, it seems the solution corresponding to J_5 is closer to the variance of the target solution. By looking at the crossections for the diffusion coefficient, we can see the mean and variance of our estimated diffusion are not doing so great in approximating the mean and variance of the target diffusion coefficient.



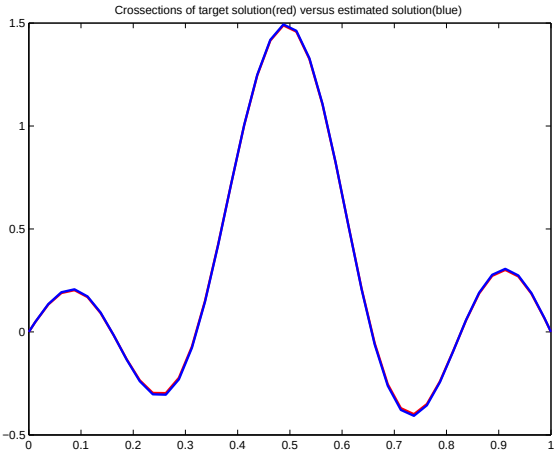
(a)



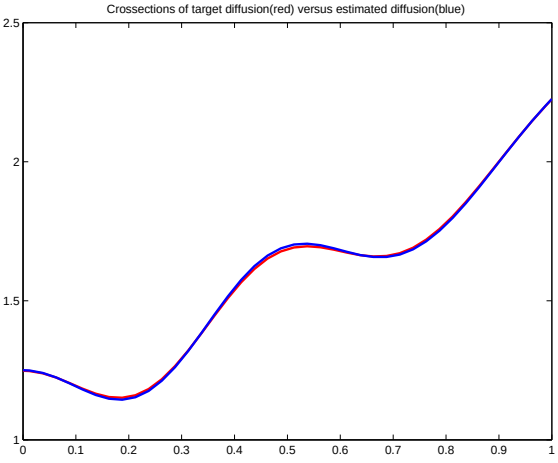
(b)



(c)

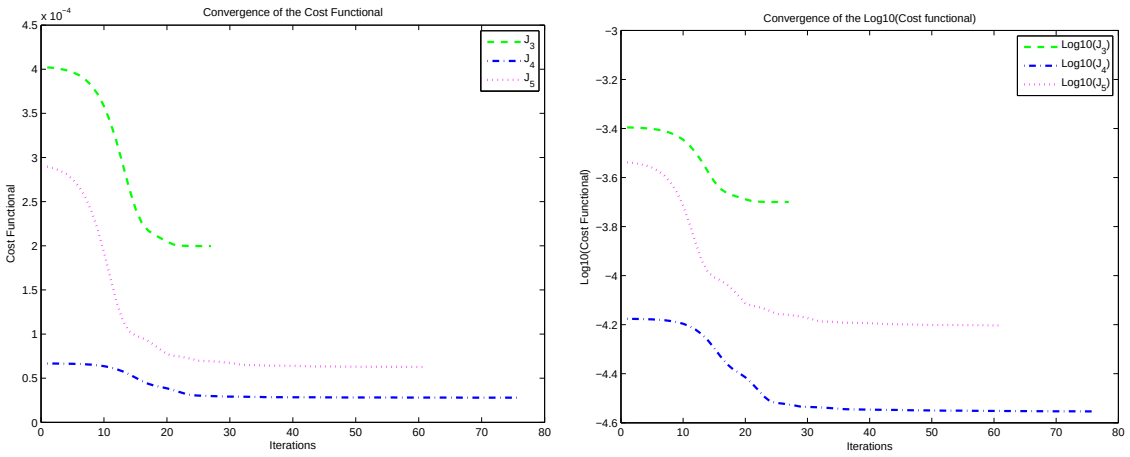


(d)



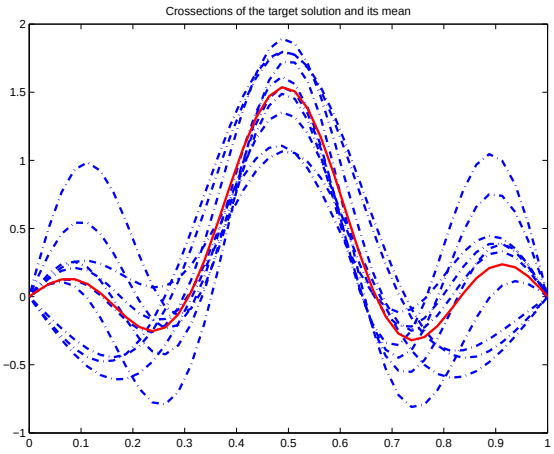
(e)

Figure 4.1: **Deterministic Case:** (a)Cost functional J, (b)Log10(J), (c)N=5 trajectories of Y's, (d)Crossection of target solution versus estimated solution, (e)Crossections of target diffusion versus estimated diffusion for a 40x40 grid; tol=10^{-4}, $\epsilon = 1$, $\beta = 10^{-6}$. The exact values of Ys are 0.5.

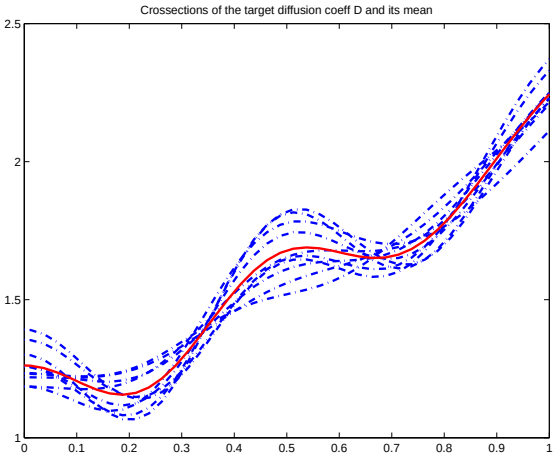


(a)

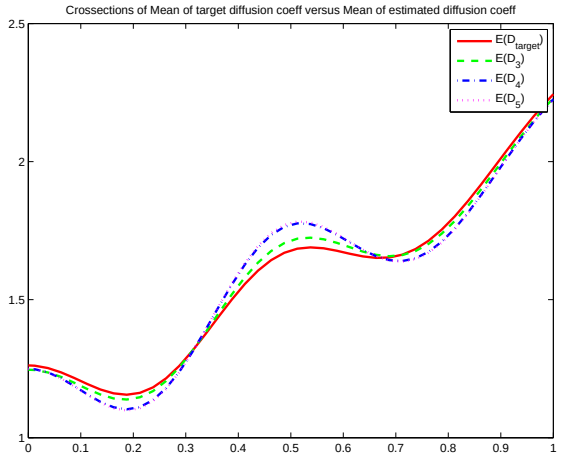
(b)



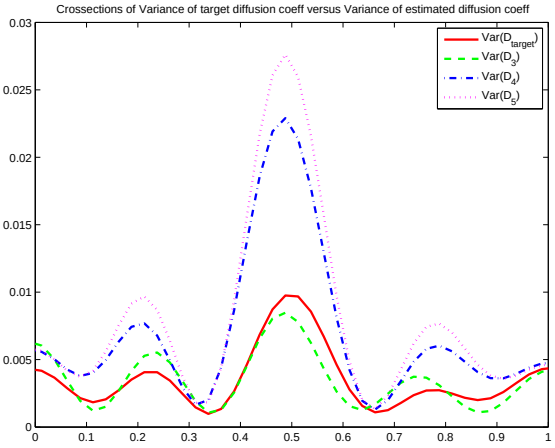
(c)



(d)

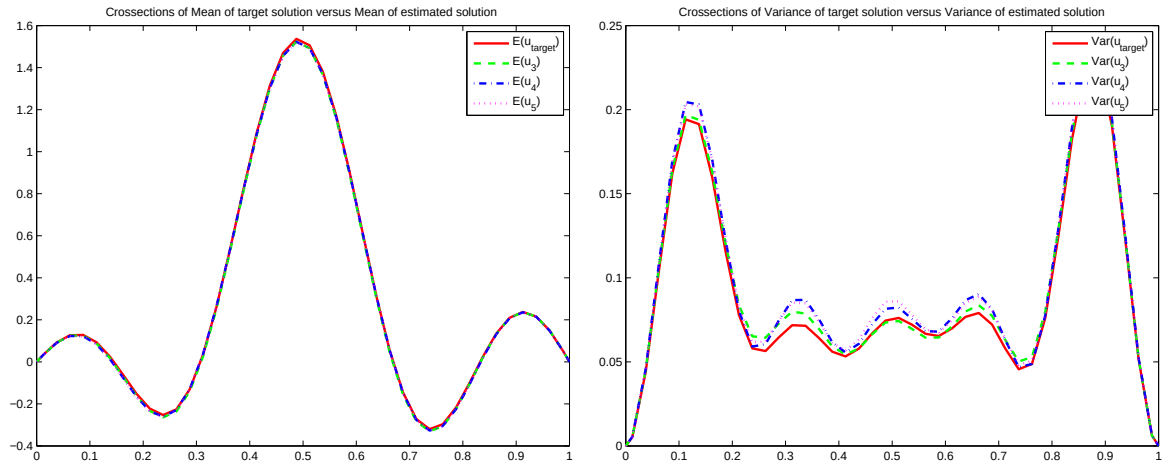


(e)



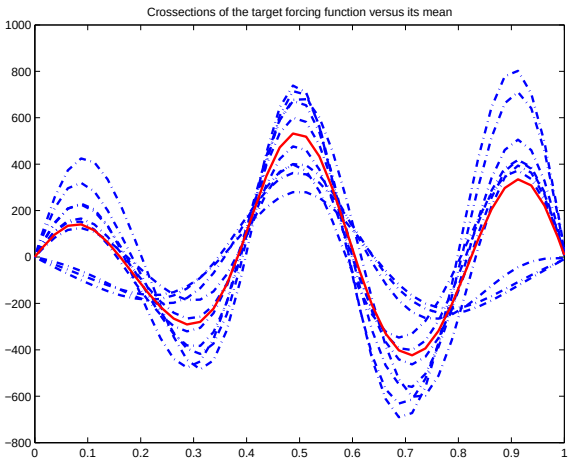
(f)

Figure 4.2: (a)J, (b)$Log_{10}(J)$ and crossections for: (c)target solution, (d)target diffusion, (e)mean of target diffusion vs. mean of estimated diffusion, (f)variance of target diffusion vs. variance of estimated diffusion. Grid considered is 40x40, tol=10^{-4}, $\epsilon = 1, \beta = 10^{-6}$, runs=10. The target values of N=5 Ys are random.

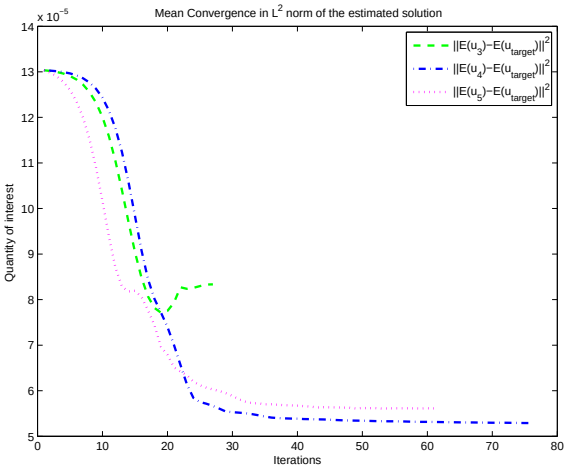


(a)

(b)

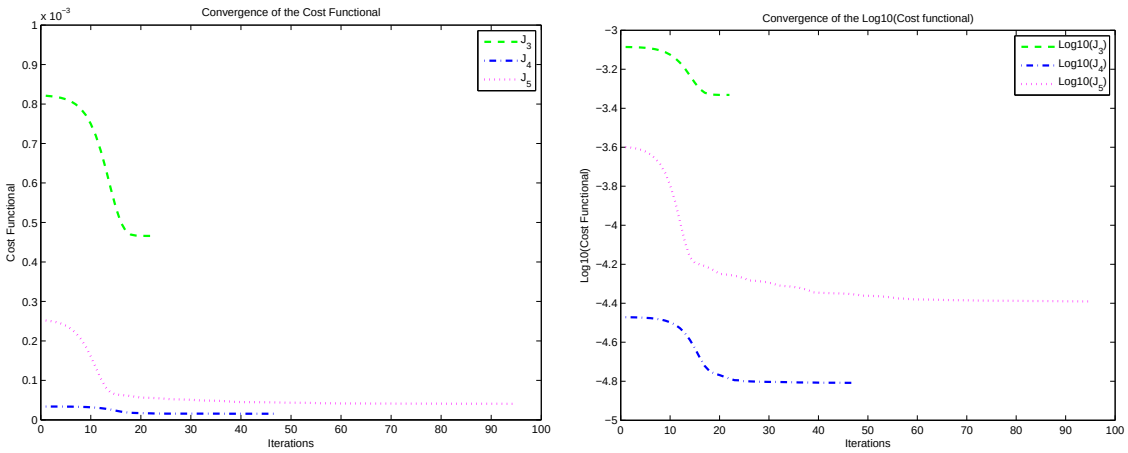


(c)



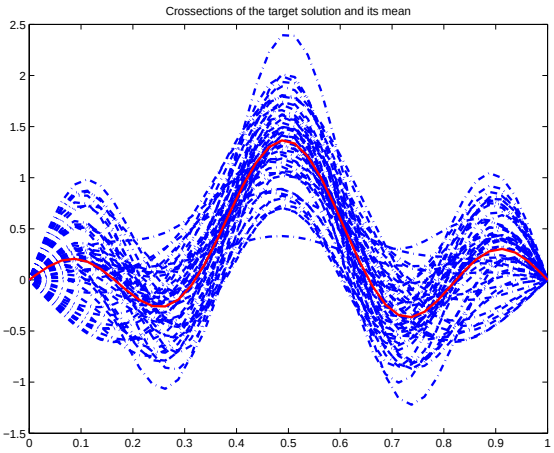
(d)

Figure 4.3: Crossections for: (a)mean of target solution vs. mean of estimated solution, (b)variance of target solution vs. variance of estimated solution, (c)forcing function f, (d)mean convergence in L^2 norm of estimated solution. Grid considered is 40x40, tol=10^{-4}, $\epsilon = 1, \beta = 10^{-6}$, runs=10. The target values of N=5 Ys are random.

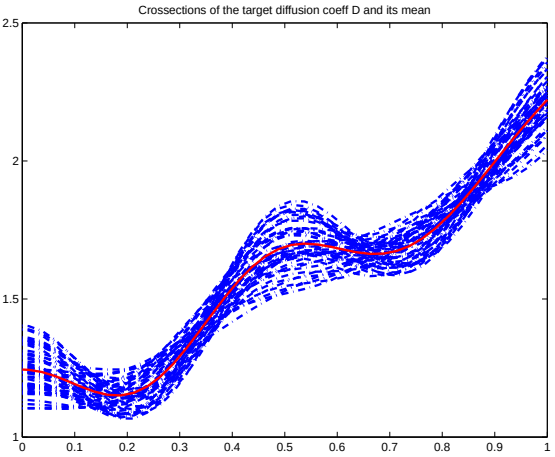


(a)

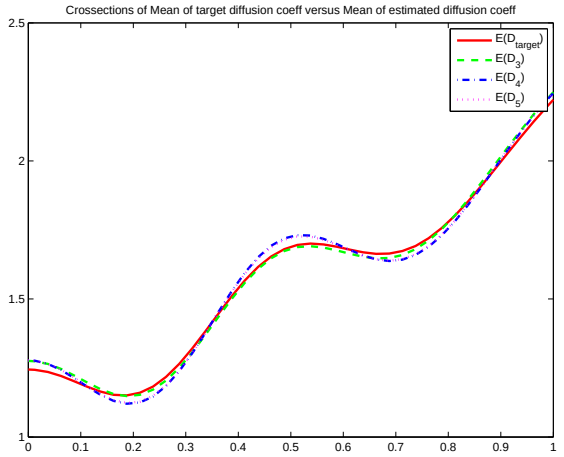
(b)



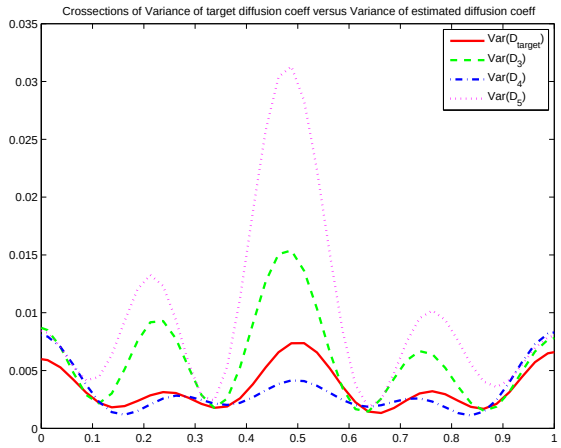
(c)



(d)

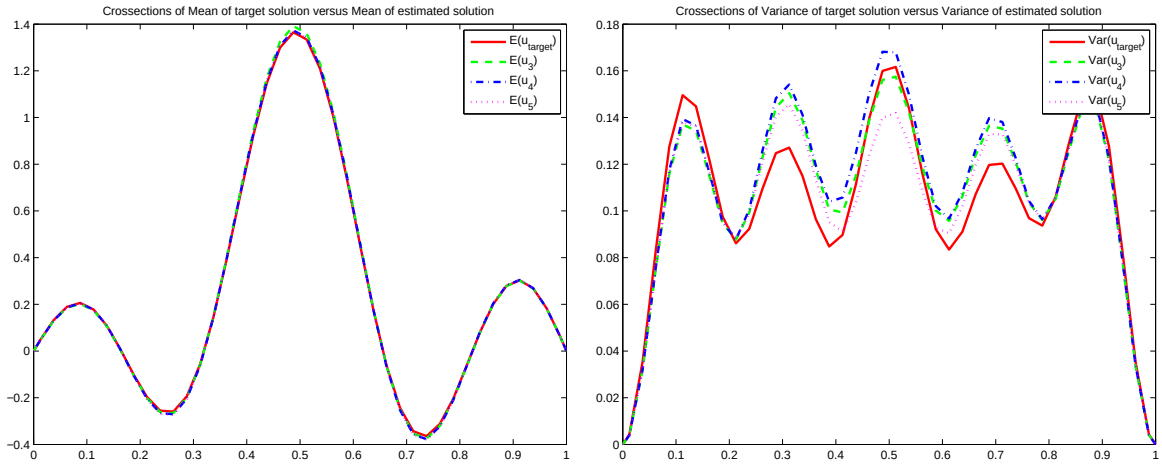


(e)



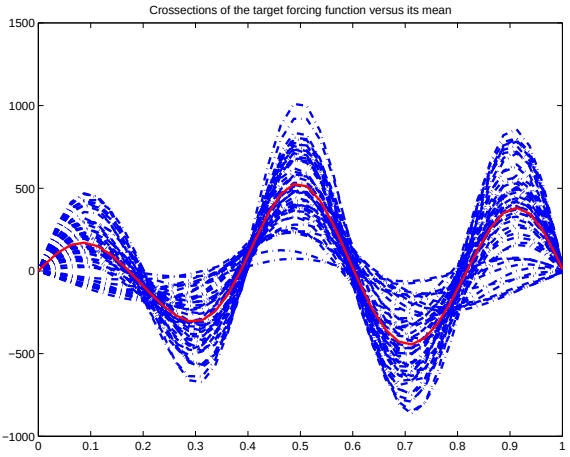
(f)

Figure 4.4: (a)J, (b)$Log_{10}(J)$ and crossections for:(c)target solution, (d)target diffusion, (e)mean of target diffusion vs. mean of estimated diffusion, (f)variance of target diffusion vs. variance of estimated diffusion. Grid considered is 40x40, tol=$10^{-4}, \epsilon = 1, \beta = 10^{-6}$, runs=50. The target values of N=5 Ys are random.

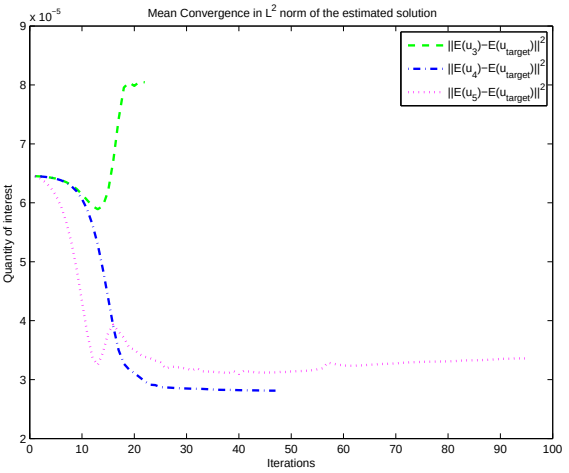


(a)

(b)



(c)



(d)

Figure 4.5: Crossections for: (a)mean of target solution vs. mean of estimated solution, (b)variance of target solution vs. variance of estimated solution, (c)forcing function f, (d)mean convergence in L^2 norm of estimated solution. Grid considered is 40x40, tol=10^{-4}, $\epsilon = 1$, $\beta = 10^{-6}$, runs=50. The target values of N=5 Ys are random.

5.0 ESTIMATING A SPATIALLY VARYING PERMEABILITY FOR THE PARABOLIC DIFFUSION EQUATION USING KALMAN FILTER, MCMC AND ADJOINT VARIABLE-BASED ALGORITHMS

Diffusion is a spontaneous movement of particles from an area of high concentration to an area of low concentration. A typical example of inverse problem is the identification of permeability of the aquifer from flow data. The permeability function for a soil may change spatially due to uncertainties in soil fabric, thus a Bayesian computational approach is developed for the estimation of permeability in flows through porous media. For complicated physical phenomena, the measurements error and model error can impact the accuracy of the estimates ([23]).

5.1 MODEL

We consider the nonlinear inverse problem of estimating the permeability in porous media flow.

$$\begin{cases} \frac{\partial p}{\partial t}(x,t) - \nabla \cdot (\kappa(x)\nabla p(x,t)) = f(x) & \text{in } D \times Time, \\ p = 0 & \text{on } \partial D \times Time \\ p(x,0) = \mathbf{1}_{\mathbf{Disk(0,0.01)}} & \text{on } D \end{cases} \tag{5.1.1}$$

We assume the source $f(x) = 0$ and the true diffusion coefficient $\overline{\kappa}$ to be given by:

$$\overline{\kappa}(x) = (1 + x^2 + y^2) + \frac{1}{1000} \sum_{n=1}^{N} \cos\left(\frac{n\pi x}{L_x}\right) \cdot \cos\left(\frac{n\pi y}{L_y}\right) Y_n$$

The exact values used for producing simulated measurements were 0.5 for the $Y_i, i = 1, \ldots, n$ in the formula for the diffusion coefficient k. We observed that one KL version of Kalman Filter, namely KLEnKF, gives the least dispersed results and can't estimate the means of Y_is as well as the other Kalman Filters. Moreover, the Kalman Filter approach shows significant improvement in efficiency(take less cpu time per modeling time step) and accuracy (more quickly converge to the correct parameter estimate) over Parallel Tempering [11], a computationally expensive MCMC method for which we considered 3 chains and 100,000 realizations.

5.2 ESTIMATES USING KALMAN FILTER ALGORITHM

The equation (5.1.1) is discretized on a rectangular domain $[-0.05, 0.05] \times [-0.035, 0.035]$, on a 10×10 spatial mesh, using cell-centered finite differences in space, an upwind scheme for the effective diffusion coeffcient, and Forward Euler for the time integration on the interval $(0, 0.2)$ with 25 time steps.

In all the types of Kalman Filter (which were presented in more detail in chapter 2), the algorithm was started with an initial guess of 1 for all the parameters Ys involved, where both model state noise and measurement noise is assumed to be 10^{-3}.

Assuming the computational cost to be the size of the ensemble, for both EnKF and SCKF the dimension of the stochastic space is $m = n+5$, where n is the number of grid cells. Here $n = 10 \times 10 = 100$ and $m = 105$. We choose for the size of the ensemble $q = 1000$, which corresponds to 10 ensemble members per grid cell. For the SCKF, the size of the ensemble is $q_{SC} = 2m + 1 = 211$. For the KL implementation,we choose $n_{KL} = 7 \times 7 = 49$ KL terms, using 7 eigenfunctions in each x and y directions. The dimension of the stochastic space is $m_{KL} = n_{KL} + 5 = 54$ and the size of the SC ensemble is $q_{KL} = 2m_{KL} + 1 = 109$. Finally, in the KLEnKF, the dimension of the stochastic space is as in the KLSCKF, $m_{KL} = 54$, but the size of the ensemble chosen is $q = 10n_{KL} = 10 \times 49 = 490$. These dimensions are summarized in Table 5.2.1. We observe that the computational cost of EnKF is approximately twice the cost of the KLEnKF, five times the cost of the SCKF, and ten times the cost of KLSCKF.

Table 5.2.1: Number of parameters, stochastic space dimension, and ensemble size for the KF methods.

	Number of spatial parameters	Dimension of stochastic space	Ensemble size
EnKF	$n = 10 \times 10 = 100$	$m = n + 5 = 105$	$q = 10n = 1000$
SCKF	$n = 10 \times 10 = 100$	$m = n + 5 = 105$	$q_{SC} = 2m + 1 = 211$
KLSCKF	$n_{KL} = 7 \times 7 = 49$	$m_{KL} = n_{KL} + 5 = 54$	$q_{KL} = 2m_{KL} + 1 = 109$
KLEnKF	$n_{KL} = 7 \times 7 = 49$	$m_{KL} = n_{KL} + 5 = 54$	$q = 10n_{KL} = 490$

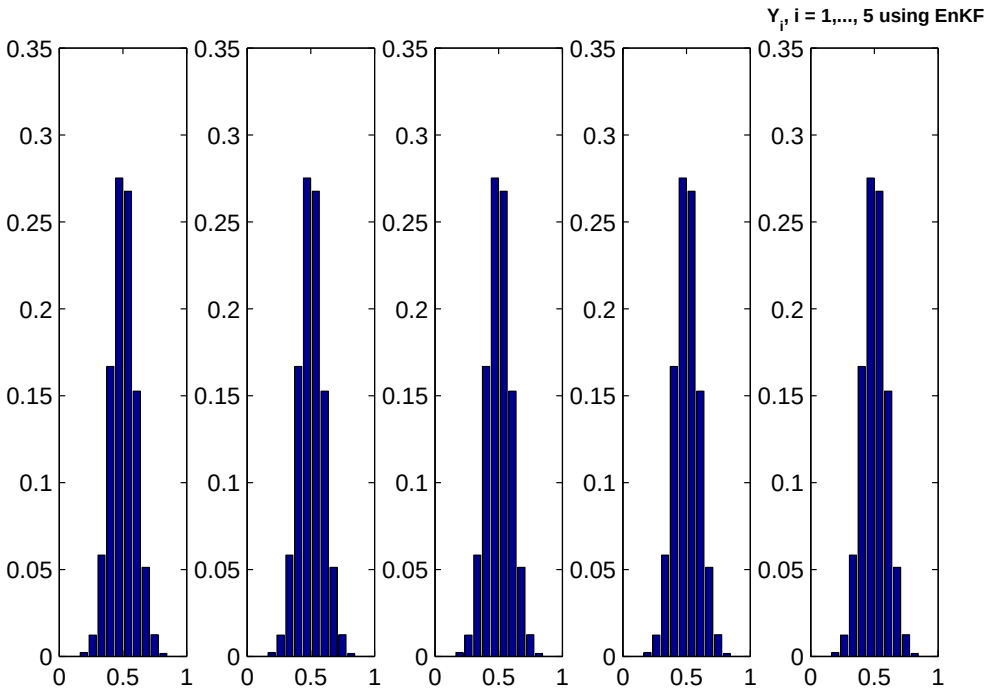


Figure 5.1: Histogram of $Y_i, i = 1, \ldots, 5$ using EnKF

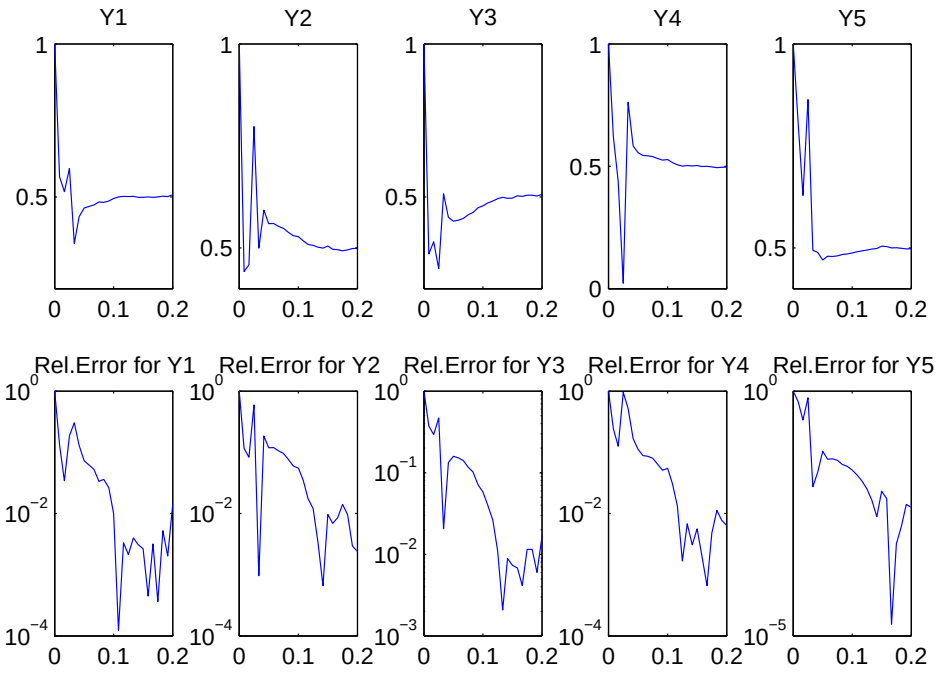


Figure 5.2: $Y_i, i = 1, \ldots, 5$ estimates and relative errors using EnKF

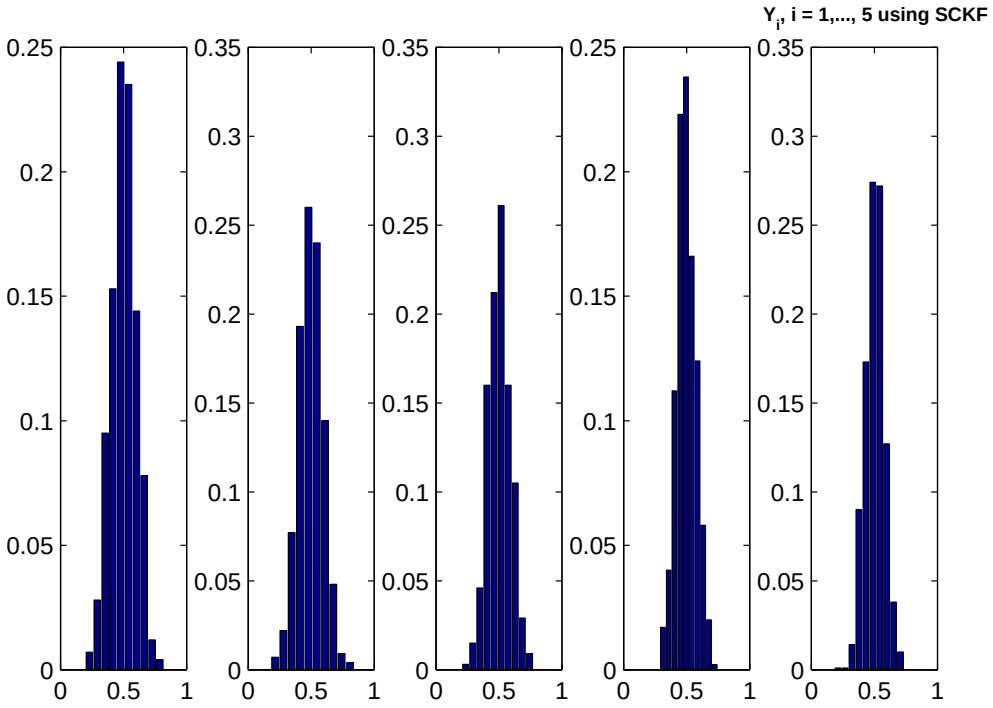


Figure 5.3: Histogram of $Y_i, i = 1, \ldots, 5$ using SCKF

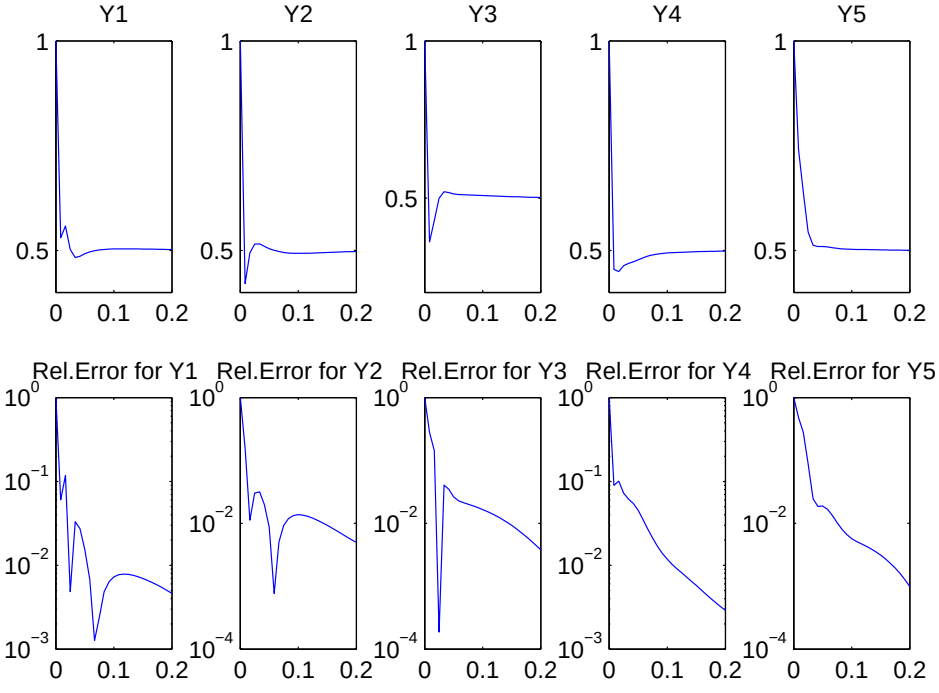


Figure 5.4: $Y_i, i = 1, \ldots, 5$ estimates and relative errors using SCKF

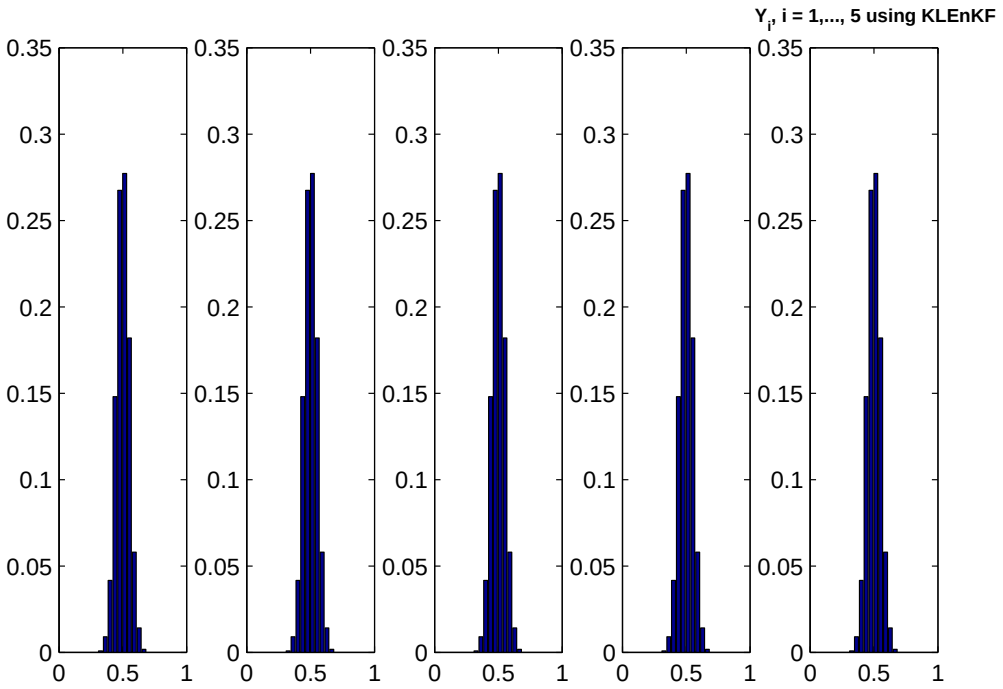


Figure 5.5: Histogram of $Y_i, i = 1, \ldots, 5$ using KLEnKF

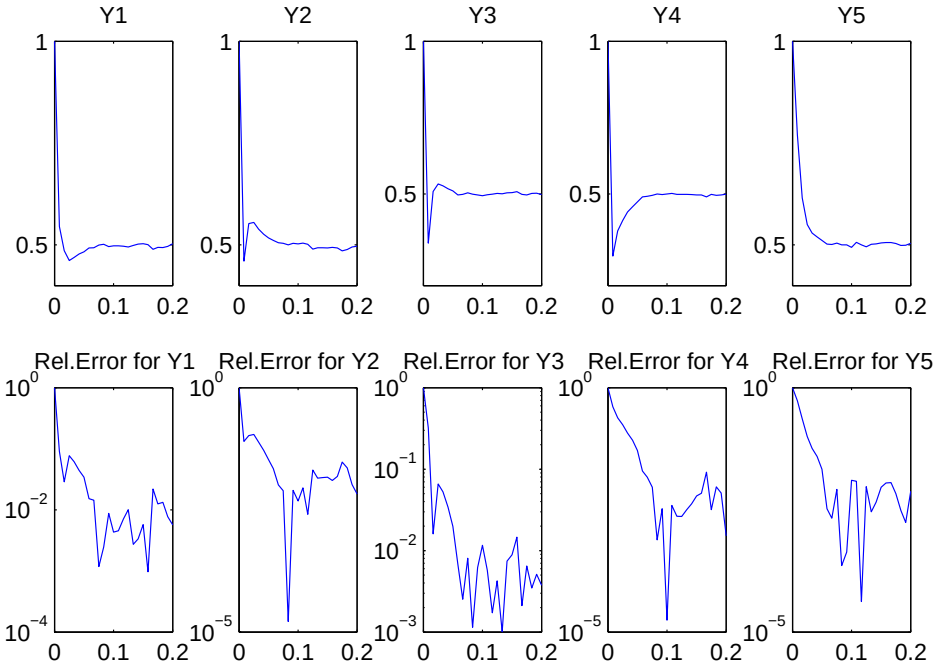


Figure 5.6: $Y_i, i = 1, \ldots, 5$ estimates and relative errors using KLEnKF

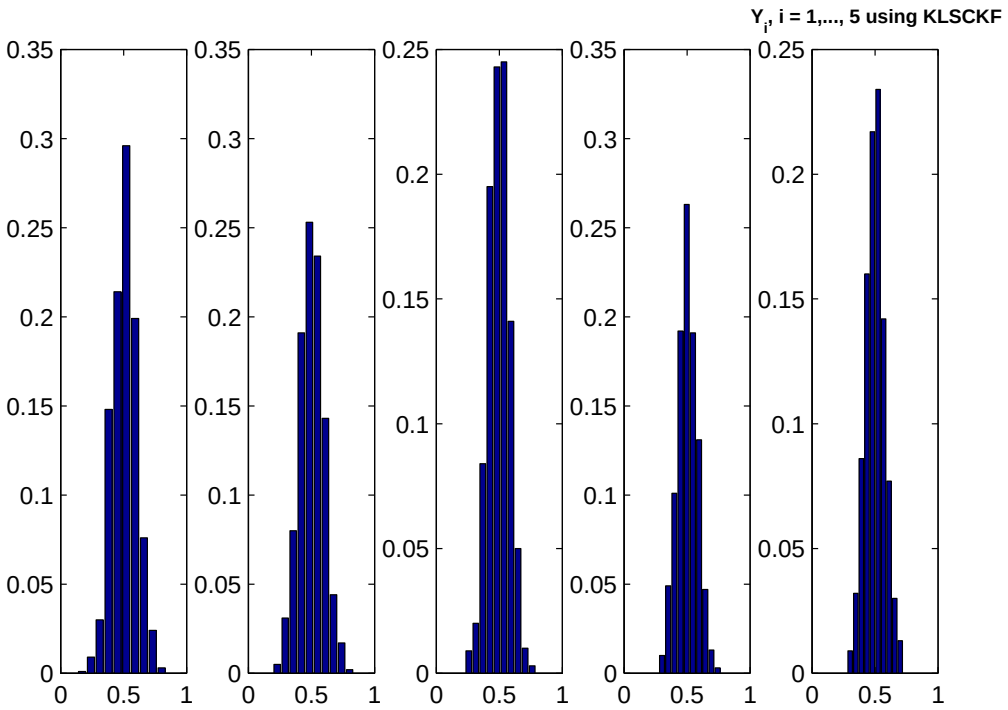


Figure 5.7: Histogram of $Y_i, i = 1, \ldots, 5$ using KLSCKF

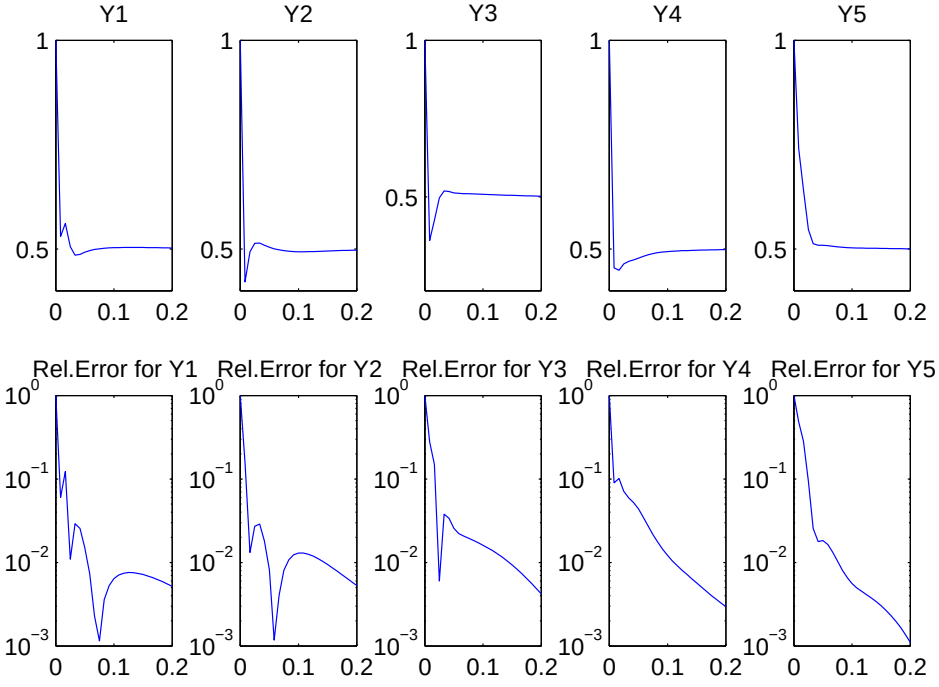


Figure 5.8: $Y_i, i = 1, \ldots, 5$ estimates and relative errors using KLSCKF

5.3 ESTIMATES USING MCMC TECHNIQUE

The Bayesian estimation of the spatially varying permeability is performed using Markov Chain Monte Carlo methods, which were presented in chapter 2.

Table 5.3.1: MCMC Rates

	Parallel Tempering Chains		
	1st chain	2nd chain	3rd chain
acceptance rate	0.3377	0.3404	0.3378
swap rate	0.2598	0.2375	

Table 5.3.2: Expected values of estimated Ys using KF and MCMC

	$E(Y_1)$	$E(Y_2)$	$E(Y_3)$	$E(Y_4)$	$E(Y_5)$
SCKF	0.5023	0.4975	0.5019	0.4986	0.5005
EnKF	0.5028	0.5022	0.4997	0.4985	0.4997
KLSCKF	0.5026	0.4974	0.5021	0.4985	0.5005
KLEnKF	0.5088	0.4991	0.5109	0.4924	0.5005
MCMC	0.4972	0.5057	0.4976	0.4984	0.5018

Table 5.3.3: Standard deviations of estimated Ys using KF and MCMC

	$\sigma(Y_1)$	$\sigma(Y_2)$	$\sigma(Y_3)$	$\sigma(Y_4)$	$\sigma(Y_5)$
SCKF	0.0945	0.0966	0.0842	0.0790	0.0754
EnKF	0.1038	0.0982	0.1008	0.0885	0.0801
KLSCKF	0.0956	0.0970	0.0843	0.0792	0.0755
KLEnKF	0.0500	0.0500	0.0500	0.0500	0.0500
MCMC	0.0400	0.0954	0.0916	0.0505	0.0161

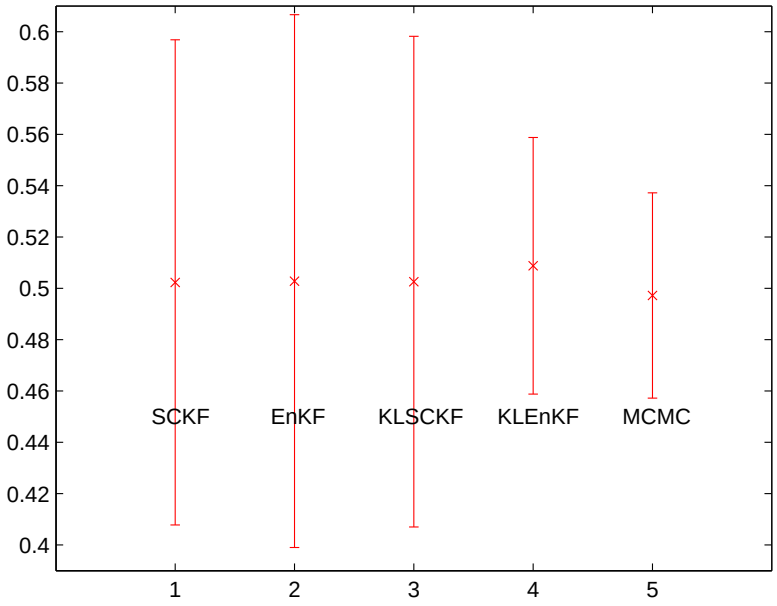


Figure 5.9: Mean $\mp$ Std. Deviation for Y_1 using KF and MCMC

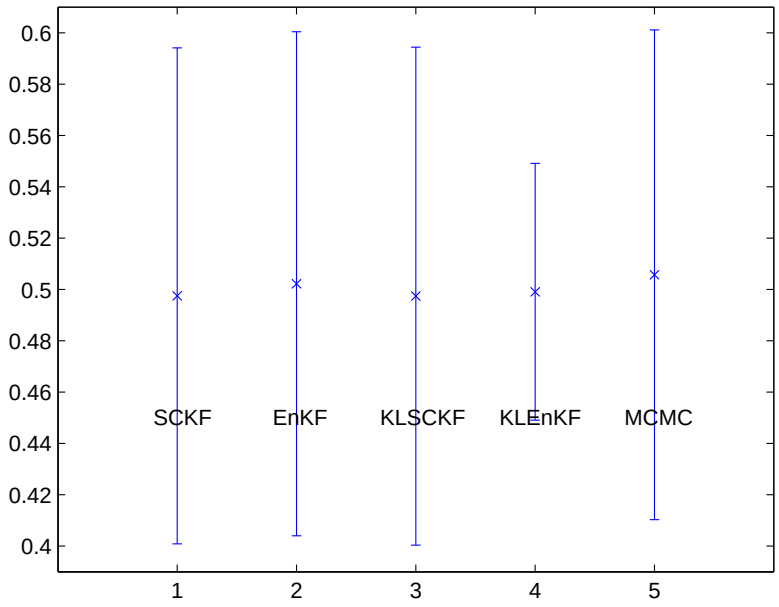


Figure 5.10: Mean $\mp$ Std. Deviation for Y_2 using KF and MCMC

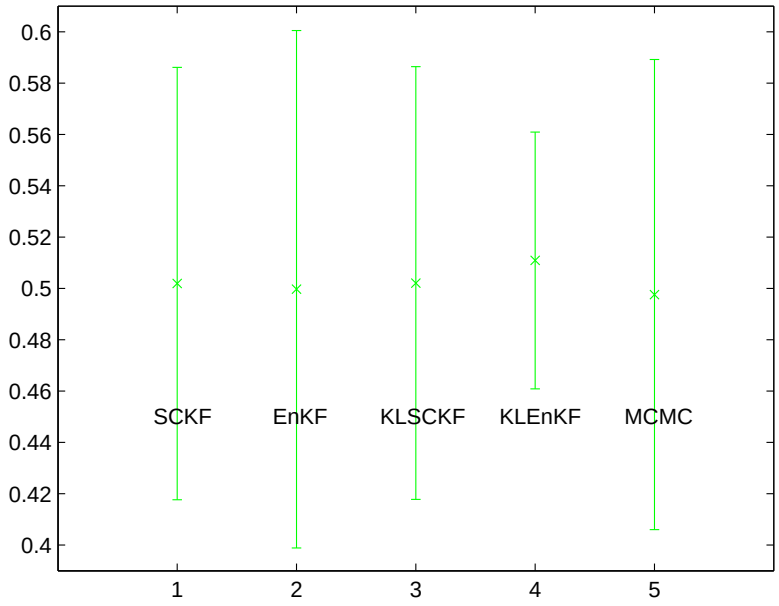


Figure 5.11: Mean $\mp$ Std. Deviation for Y_3 using KF and MCMC

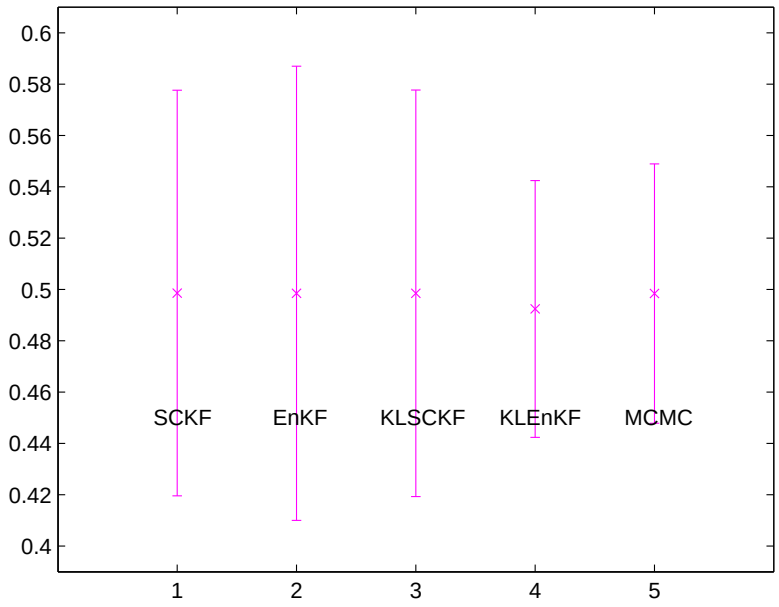


Figure 5.12: Mean $\mp$ Std. Deviation for Y_4 using KF and MCMC

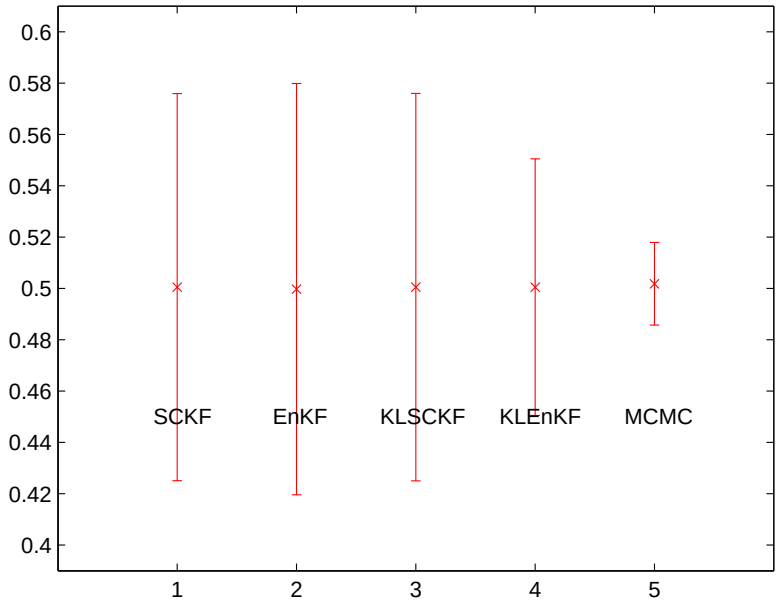


Figure 5.13: Mean $\mp$ Std. Deviation for Y_5 using KF and MCMC

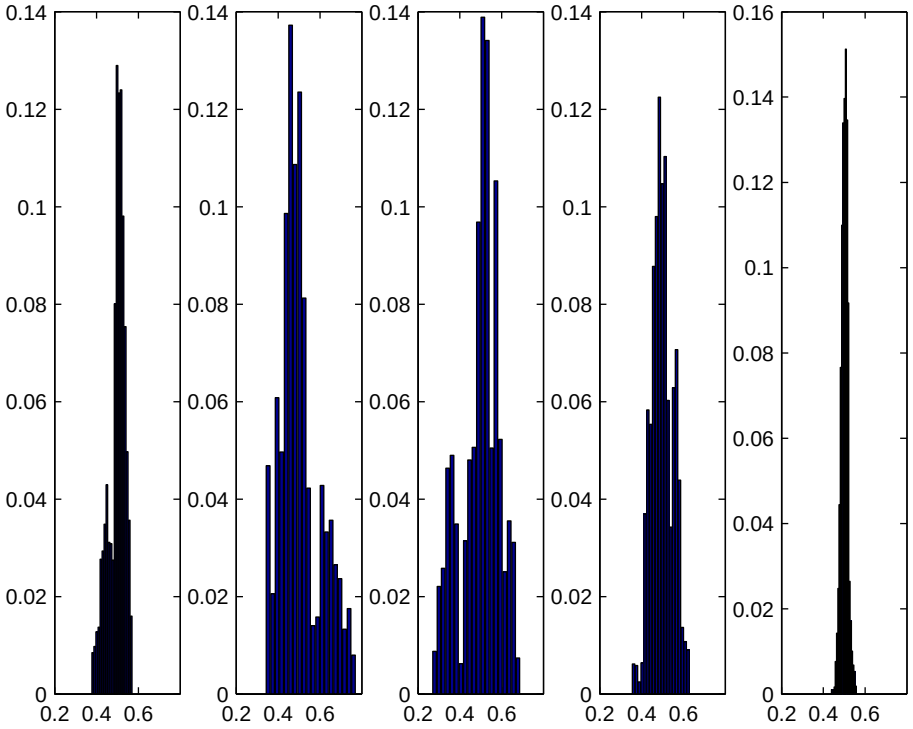


Figure 5.14: Histogram of $Y_i, i = 1 \ldots, 5$ using MCMC

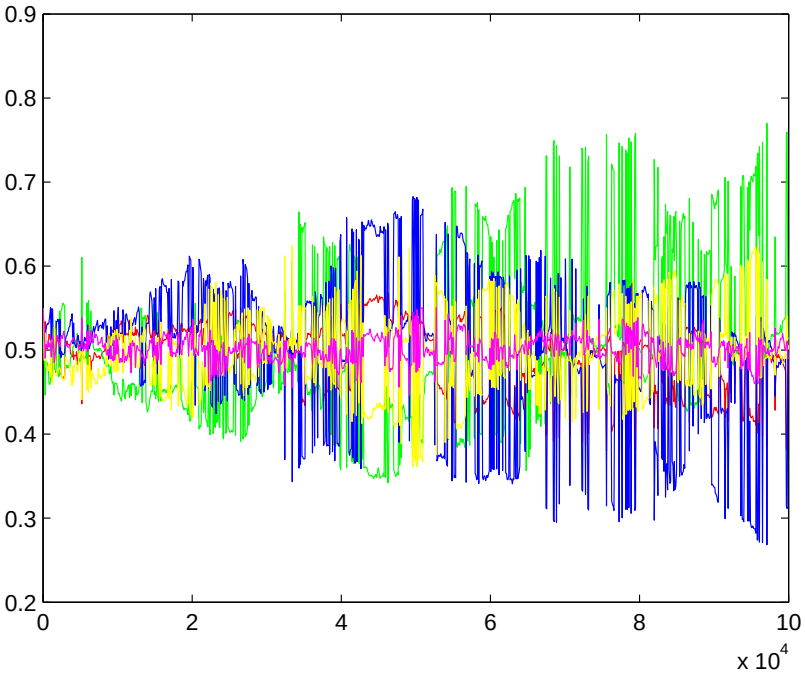


Figure 5.15: $Y_i, i = 1, \ldots, 5$ estimates using MCMC

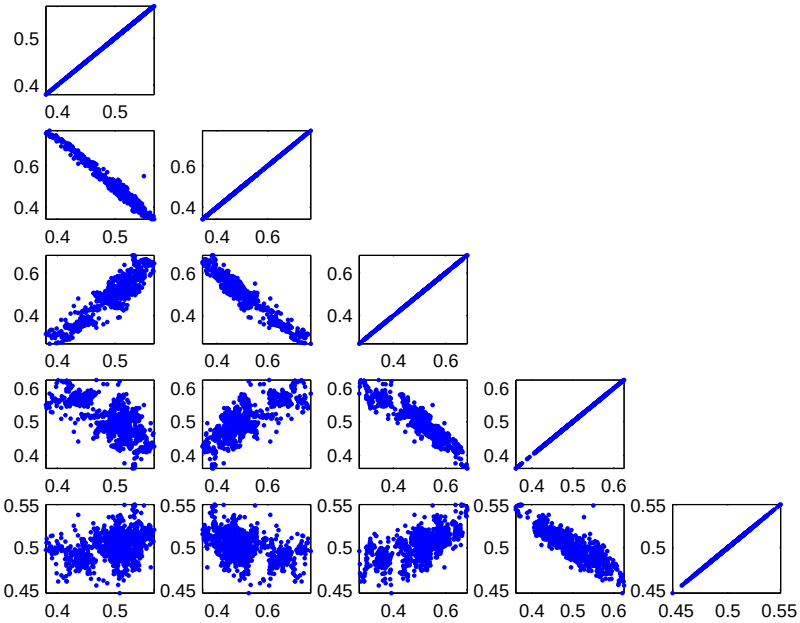


Figure 5.16: Correlation diagram using MCMC

5.4 ESTIMATES USING ADJOINT VARIABLE-BASED ALGORITHM

For the parabolic state equations, in the optimization problems, the approach we take is discretize-then-optimize [27]. We assume the existence of Lagrange multipliers [57, 59] and we present the derivation of sensitivity and adjoint equations for the continuous state equations and semi-discrete in time approximation using Backward Euler methods for both the sensitivity analysis and the Matlab implementation of algorithms. These semi-discrete optimality systems are used in the algorithms for producing the results presented in Tables 5.4.2, 5.4.3, 5.4.4, 5.4.5, 5.4.6, 5.4.7, 5.4.8, 5.4.9.

5.4.1 Sensitivity Analysis for Parabolic Case

5.4.1.1 State Equations, Adjoint Equations, Cost Functional Consider the state equations:

$$\begin{cases} \frac{p^*_{n+1}-p^*_n}{\Delta t} - \nabla\cdot\left(\kappa\nabla p^*_{n+1}\right) = f_{n+1} & \text{in } \Omega\times D, \\ p^*_0 = p_0 & \text{on } \Omega\times\partial D. \end{cases} \tag{5.4.1}$$

We introduce the adjoint equations:

$$\begin{cases} \eta_m - \eta_{m+1} - \Delta t\nabla\cdot(\kappa\nabla\eta_m) = \Delta t\,(p^*_m - \overline{p}_m) & \text{in } \Omega\times D, m = 1,2,\dots N-1 \\ \eta_N - \Delta t\nabla\cdot(\kappa\nabla\eta_N) = \Delta t\,(p^*_N - \overline{p}_N) & \text{in } \Omega\times D, \\ \eta = 0 & \text{on } \Omega\times\partial D. \end{cases} \tag{5.4.2}$$

Define the cost functional:

$$J_3(Y_i, i = 1\dots N) = \frac{1}{2}\Delta t\mathbb{E}\left[\sum_{n=1}^{N}\|p_n(Y_i,\cdot) - \overline{p}_n(Y_i,\cdot)\|^2_{L^2(D)}\right] + \frac{\beta}{2}\mathbb{E}\left[\|\kappa(Y_i,\cdot)\|^2_{L^2(D)}\right]$$

Description of the adjoint equations We start with the sensitivity equations:

$$\begin{cases} \frac{\left(p^*_{n+1}+\epsilon p_{s,n+1}\right)-\left(p^*_n+\epsilon p_{s,n}\right)}{\Delta t} - \nabla\cdot\left((\kappa+\epsilon\kappa_s)\nabla(p^*_{n+1}+\epsilon p_{s,n+1})\right) = f_{n+1} & \text{in } \Omega\times D \\ p_{s,0} = 0 & \text{on } \Omega\times\partial D. \end{cases} \tag{5.4.3}$$

Multiplying out and using the state equations (5.4.1), we get:

$$\frac{p_{s,n+1}-p_{s,n}}{\Delta t} - \nabla\cdot\left(\kappa\nabla p_{s,n+1} + \kappa_s\nabla p^*_{n+1}\right) = 0.$$

Next, we multiply by the adjoint variable η_{n+1}, integrate over D and then apply the summation operator $\Delta t \cdot \sum_{n=0}^{N-1}$:

$$\underbrace{\Delta t \sum_{n=0}^{N-1} \int_D \frac{p_{s,n+1} - p_{s,n}}{\Delta t} \eta_{n+1}}_{I} \underbrace{-\Delta t \sum_{n=0}^{N-1} \int_D \nabla \cdot \left(\kappa \nabla p_{s,n+1} + \kappa_s \nabla p^*_{n+1} \right) \eta_{n+1}}_{II} = 0. \tag{5.4.4}$$

Now, I can be rewritten as following:

$$\begin{aligned}
I &= \int_D \left(\sum_{n=0}^{N-1} p_{s,n+1}\eta_{n+1} - \sum_{n=0}^{N-1} p_{s,n}\eta_{n+1} \right) \\
&\overset{m=n+1}{=} \int_D \left(\sum_{m=1}^{N} p_{s,m}\eta_m - \sum_{n=0}^{N-1} p_{s,n}\eta_{n+1} \right) \\
&= \int_D \left(\sum_{m=1}^{N-1} p_{s,m}\eta_m + p_{s,N}\eta_N - \sum_{n=1}^{N-1} p_{s,n}\eta_{n+1} - \underbrace{p_{s,0}}_{=0} \eta_0 \right) \\
&= \int_D \left(\sum_{m=1}^{N-1} p_{s,m}(\eta_m - \eta_{m+1}) + p_{s,N}\eta_N \right).
\end{aligned}$$

Integration by parts on II yields:

$$\begin{aligned}
II &= \Delta t \sum_{n=0}^{N-1} - \int_{\partial D} \kappa \nabla p_{s,n+1} \underbrace{\eta_{n+1}}_{=0 on \partial D} \vec{n} + \int_D \kappa \nabla p_{s,n+1} \nabla \eta_{n+1} - \int_{\partial D} \kappa_s \nabla p^*_{n+1} \underbrace{\eta_{n+1}}_{=0 on \partial D} \vec{n} + \\
&+ \int_D \kappa_s \nabla p^*_{n+1} \nabla \eta_{n+1} \\
&= \Delta t \int_{\partial D} \sum_{n=0}^{N-1} p_{s,n+1} \kappa \underbrace{\nabla \eta_{n+1}}_{=0 on \partial D} \vec{n} - \Delta t \int_D \sum_{n=0}^{N-1} p_{s,n+1} \nabla \cdot (\kappa \nabla \eta_{n+1}) + \Delta t \int_D \sum_{n=0}^{N-1} \kappa_s \nabla p^*_{n+1} \nabla \eta_{n+1}
\end{aligned}$$

(by integration by parts again)

Thus (5.4.4) becomes:

$$\begin{aligned}
&0 = I + II \\
&= \int_D \sum_{m=1}^{N-1} p_{s,m}(\eta_m - \eta_{m+1}) + \int_D p_{s,N}\eta_N - \Delta t \int_D \sum_{n=0}^{N-1} p_{s,n+1} \nabla \cdot (\kappa \nabla \eta_{n+1}) + \Delta t \int_D \sum_{n=0}^{N-1} \kappa_s \nabla p^*_{n+1} \nabla \eta_{n+1}.
\end{aligned}$$

Taking expection and using the substitution $m = n + 1$, we obtain:

$$0 = \mathbb{E}\int_D \sum_{m=1}^{N-1} p_{s,m}(\eta_m - \eta_{m+1}) + \mathbb{E}\int_D p_{s,N}\eta_N - \Delta t\mathbb{E}\int_D \sum_{m=1}^{N} p_{s,m}\nabla\cdot(\kappa\nabla\eta_m) + \Delta t\mathbb{E}\int_D \sum_{m=1}^{N} \kappa_s \nabla p_m^* \nabla\eta_m. \tag{5.4.5}$$

The Derivative of the Cost Functional:

$$\begin{aligned}\frac{dJ}{dY_i}Y_{i,s} &= \lim_{\epsilon\to 0}\frac{J(Y_1,\ldots,Y_i+\epsilon Y_{i,s},\ldots) - J(Y_1,\ldots,Y_i,\ldots)}{\epsilon} \\ &= \Delta t\mathbb{E}\int_D \sum_{m=1}^{N}(p_m^* - \overline{p}_m)p_{s,m} + \beta\mathbb{E}\int_D \kappa\frac{d\kappa}{dY_i}Y_{i,s}\end{aligned} \tag{5.4.6}$$

By identifying the coefficients of $p_{s,m}$ from (5.4.5) and (5.4.6), where $m = 1\ldots N$, we infer the already stated form of the adjoint equations (5.4.2):

$$\begin{cases} \eta_m - \eta_{m+1} - \Delta t\nabla\cdot(\kappa\nabla\eta_m) = \Delta t\,(p_m^* - \overline{p}_m) & \text{in } \Omega\times D, m = 1,2,\ldots N-1 \\ \eta_N - \Delta t\nabla\cdot(\kappa\nabla\eta_N) = \Delta t\,(p_N^* - \overline{p}_N) & \text{in } \Omega\times D, \\ \eta = 0 & \text{on } \Omega\times\partial D. \end{cases}$$

Using the adjoint equations (5.4.2), the previous identity (5.4.5) becomes:

$$0 = \Delta t\mathbb{E}\int_D \sum_{m=1}^{N}(p_m^* - \overline{p}_m)p_{s,m} + \Delta t\mathbb{E}\int_D \sum_{m=1}^{N}\kappa_s\nabla p_m^*\nabla\eta_m. \tag{5.4.7}$$

Therefore (5.4.6) has the following expression:

$$\begin{aligned}\frac{dJ}{dY_i}Y_{i,s} &= \Delta t\mathbb{E}\int_D \sum_{m=1}^{N}(p_m^* - \overline{p}_m)p_{s,m} + \beta\mathbb{E}\int_D \kappa\frac{d\kappa}{dY_i}Y_{i,s} \\ &\overset{\text{(by using (5.4.7))}}{=} -\Delta t\mathbb{E}\int_D \sum_{m=1}^{N}\kappa_s\nabla p_m^*\nabla\eta_m + \beta\mathbb{E}\int_D \kappa\frac{d\kappa}{dY_i}Y_{i,s}.\end{aligned} \tag{5.4.8}$$

In order to compare the Y's estimates obtained by using adjoint variable-based technique with the results provided by Kalman filter and MCMC, sample simulations were produced on the same 10x10 spatial mesh, where we first considered 10 realizations and then 50 realizations for our stochastic model. Since Kalman filter and MCMC are stochastic methods

by nature, in order to incorporate stochasticity in our adjoint code, the target Ys were generated using a normal distribution with mean 0.5 and standard deviation denoted by the variable *noise*. For our examples, a first case with $noise = 10^{-1}$ was considered and then a smaller noise of 10^{-3}. To be more specific, we are taken into account 4 cases: the first 2 cases have 10 realizations with noises of 10^{-1} and 10^{-3} respectively, the last 2 cases have 50 realizations with noises of 10^{-1} and 10^{-3} respectively.

For all the four cases considered, by plotting crossections, we observed that both our estimated solutions and estimated diffusion coefficients corresponding to eitherJ_3, J_4 or J_5 approximate very well the mean of the target solution, respectively the mean of target diffusion coefficient, while the variance of the target solution and target diffusion coefficient is not too well approximated.

For the case where only 10 realizations were considered with a noise of 10^{-1}, we observed that for achieving the same tolerance of 10^{-7} for the relative error, the cost functional J_3 decreases 2 orders of magnitude, whereas the decrease for J_4 and J_5 was 4 orders of magnitude. By plotting the histograms, we get roughly the same mean(see Table 5.4.2) and the same standard deviation(given in Table 5.4.3) for the Ys estimates obtained when using J_4 and J_5 cost functionals. When 10 realizations were considered with a noise of 10^{-3}, the cost functional J_3 decreases 4 orders of magnitude, but 5 orders of magnitude for J_4 and J_5. Since the *noise* is so small, the mean of all estimated Ys is the same, regardless of the cost functional used(see Table 5.4.4), whereas the standard deviation(see Table 5.4.5) for the Y estimates obtained by using J_4 and J_5 cost functionals apears to be the same as in the case with 10 runs and a noise of 10^{-1}.

For the cases with 50 realizations being considered, the results are comparable to the ones obtained by having only 10 realizations.

Remark 5.4.1. *The computational cost of the adjoint variable-based algorithms is estimated as follows: the cost of one iteration of the gradient algorithm (one forward equation plus one adjoint/backward equation), times the number of iterations, times the number of realizations.*

From Remark 5.4.1 we see that the cost of the adjoint-based algorithm using noise of

10^{-1} is 30 iterations $\times$ 2 equations $\times$ 10 realizations, so 600 equations solved. For the noise of 10^{-3}, the cost is roughly 32 iterations $\times$ 2 equations $\times$ 10 realizations, so 640 equations solved. We recall the cost associated with using EnKF, SCKF, KLSCKF and KLEnKF for parameter estimation from Table 5.2.1. Hence the adjoint-based method is roughly six

Table 5.4.1: Computational cost of KF, MCMC and Adj. method using J_4 with 10 realizations, $noise = 10^{-3}$

Method	EnKF	SCKF	KLSCKF	KLEnKF	MCMC	Adjoint
eq. solved	1000	211	109	490	100,000	640

times more expensive than KLSCKF. Moreover, from figures 5.17, 5.18, 5.19, 5.20, 5.21, we observe that among all the methods considered, the adjoint-based algorithm is the most accurate, as it recovers the means with the smallest standard deviation.

Table 5.4.2: Expected values of estimated Ys using Adj. algorithm with 10 realizations and $noise = 10^{-1}$

	$E(Y_1)$	$E(Y_2)$	$E(Y_3)$	$E(Y_4)$	$E(Y_5)$
Adj J_3	0.5097	0.5134	0.5105	0.5082	0.5087
Adj J_4	0.5065	0.5102	0.5073	0.5050	0.5055
Adj J_5	0.5066	0.5103	0.5073	0.5051	0.5056
Adj Y_{target}	0.5195	0.4833	0.5226	0.5149	0.4761

Table 5.4.3: Standard deviations of estimated Ys using Adj. algorithm with 10 realizations and $noise = 10^{-1}$

	$\sigma(Y_1)$	$\sigma(Y_2)$	$\sigma(Y_3)$	$\sigma(Y_4)$	$\sigma(Y_5)$
Adj J_3	0.0397	0.0423	0.0399	0.0422	0.0453
Adj J_4	0.0088	0.0113	0.0076	0.0060	0.0084
Adj J_5	0.0088	0.0113	0.0076	0.0060	0.0084
Adj Y_{target}	0.0445	0.1169	0.1223	0.0968	0.0712

Table 5.4.4: Expected values of estimated Ys using Adj. algorithm with 10 realizations and $noise = 10^{-3}$

	$E(Y_1)$	$E(Y_2)$	$E(Y_3)$	$E(Y_4)$	$E(Y_5)$
Adj J_3	0.4989	0.5027	0.4997	0.4974	0.4980
Adj J_4	0.4989	0.5027	0.4997	0.4974	0.4980
Adj J_5	0.4989	0.5027	0.4997	0.4974	0.4980
Adj Y_{target}	0.5002	0.4998	0.5002	0.5001	0.4998

Table 5.4.5: Standard deviations of estimated Ys using Adj. algorithm with 10 realizations and $noise = 10^{-3}$

	$\sigma(Y_1)$	$\sigma(Y_2)$	$\sigma(Y_3)$	$\sigma(Y_4)$	$\sigma(Y_5)$
Adj J_3	0.0043	0.0081	0.0080	0.0072	0.0100
Adj J_4	0.0088	0.0113	0.0076	0.0060	0.0084
Adj J_5	0.0088	0.0113	0.0076	0.0060	0.0084
Adj Y_{target}	0.0004	0.0012	0.0012	0.0010	0.0007

Table 5.4.6: Expected values of estimated Ys using Adj. algorithm with 50 realizations and $noise = 10^{-1}$

	$E(Y_1)$	$E(Y_2)$	$E(Y_3)$	$E(Y_4)$	$E(Y_5)$
Adj J_3	0.5016	0.5010	0.5020	0.5008	0.5037
Adj J_4	0.4972	0.4966	0.4976	0.4964	0.4993
Adj J_5	0.4971	0.4966	0.4976	0.4963	0.4992
Adj Y_{target}	0.4902	0.5102	0.5073	0.4757	0.4980

Table 5.4.7: Standard deviations of estimated Ys using Adj. algorithm with 50 realizations and $noise = 10^{-1}$

	$\sigma(Y_1)$	$\sigma(Y_2)$	$\sigma(Y_3)$	$\sigma(Y_4)$	$\sigma(Y_5)$
Adj J_3	0.0534	0.0551	0.0560	0.0537	0.0550
Adj J_4	0.0107	0.0117	0.0122	0.0112	0.0090
Adj J_5	0.0107	0.0117	0.0122	0.0112	0.0090
Adj Y_{target}	0.0815	0.0961	0.1028	0.0828	0.0904

Table 5.4.8: Expected values of estimated Ys using Adj. algorithm with 50 realizations and $noise = 10^{-3}$

	$E(Y_1)$	$E(Y_2)$	$E(Y_3)$	$E(Y_4)$	$E(Y_5)$
Adj J_3	0.5002	0.4997	0.5007	0.4994	0.5023
Adj J_4	0.5002	0.4997	0.5007	0.4994	0.5023
Adj J_5	0.5002	0.4997	0.5007	0.4994	0.5023
Adj Y_{target}	0.4999	0.5001	0.5001	0.4998	0.5000

Table 5.4.9: Standard deviations of estimated Ys using Adj. algorithm with 50 realizations and $noise = 10^{-3}$

	$\sigma(Y_1)$	$\sigma(Y_2)$	$\sigma(Y_3)$	$\sigma(Y_4)$	$\sigma(Y_5)$
Adj J_3	0.0053	0.0074	0.0097	0.0124	0.0117
Adj J_4	0.0106	0.0116	0.0121	0.0111	0.0090
Adj J_5	0.0107	0.0117	0.0121	0.0111	0.0090
Adj Y_{target}	0.0008	0.0010	0.0010	0.0008	0.0009

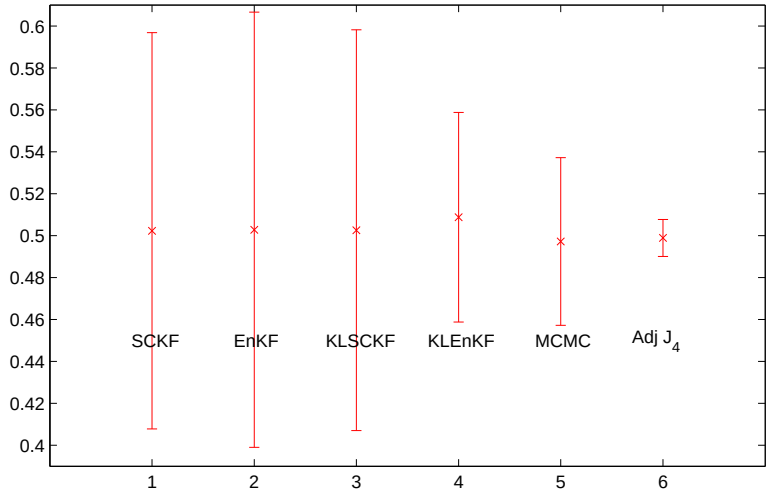


Figure 5.17: Mean $\mp$ Std. Deviation for Y_1 using KF, MCMC and Adj. method using J_4 with 10 realizations, $noise = 10^{-3}$

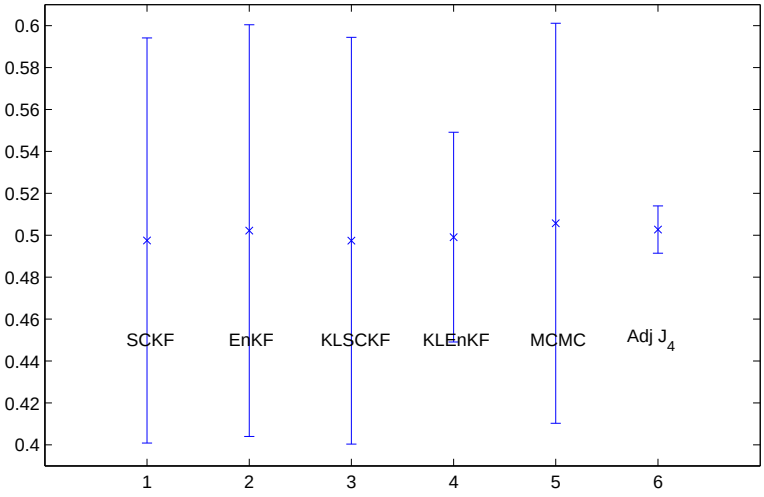


Figure 5.18: Mean $\mp$ Std. Deviation for Y_2 using KF, MCMC and Adj. method using J_4 with 10 realizations, $noise = 10^{-3}$

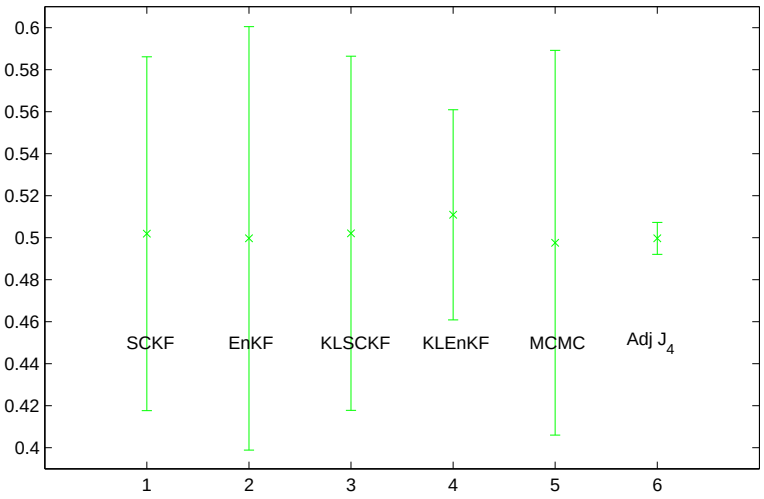


Figure 5.19: Mean $\mp$ Std. Deviation for Y_3 using KF, MCMC and Adj. method using J_4 with 10 realizations, $noise = 10^{-3}$

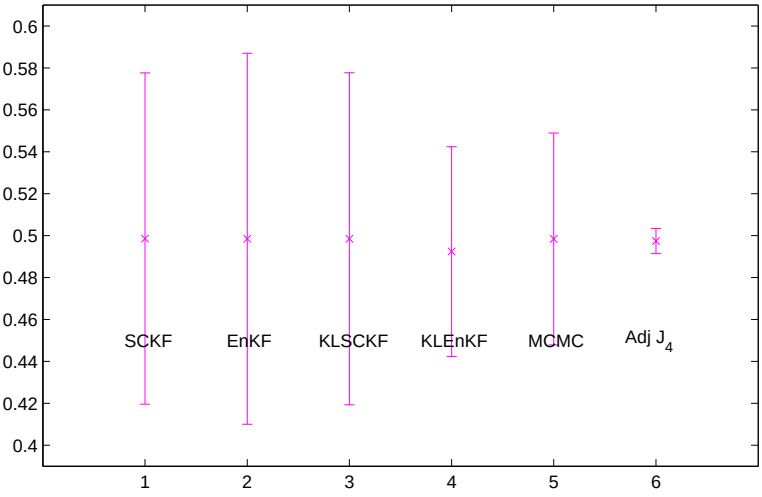


Figure 5.20: Mean $\mp$ Std. Deviation for Y_4 using KF, MCMC and Adj. method using J_4 with 10 realizations, $noise = 10^{-3}$

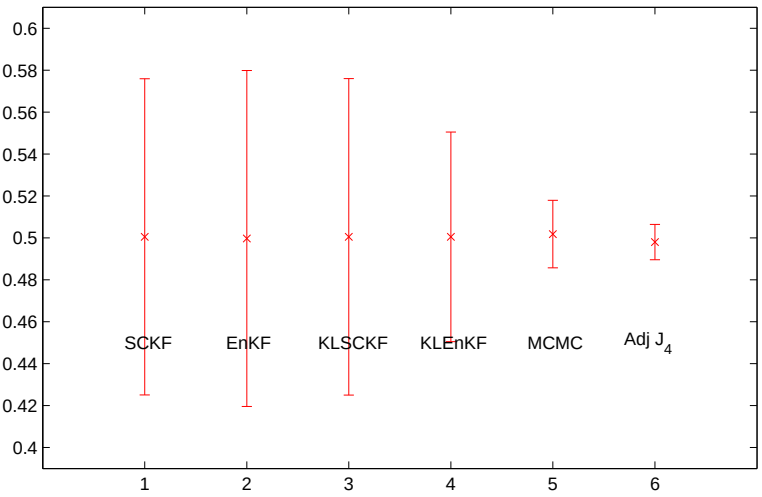
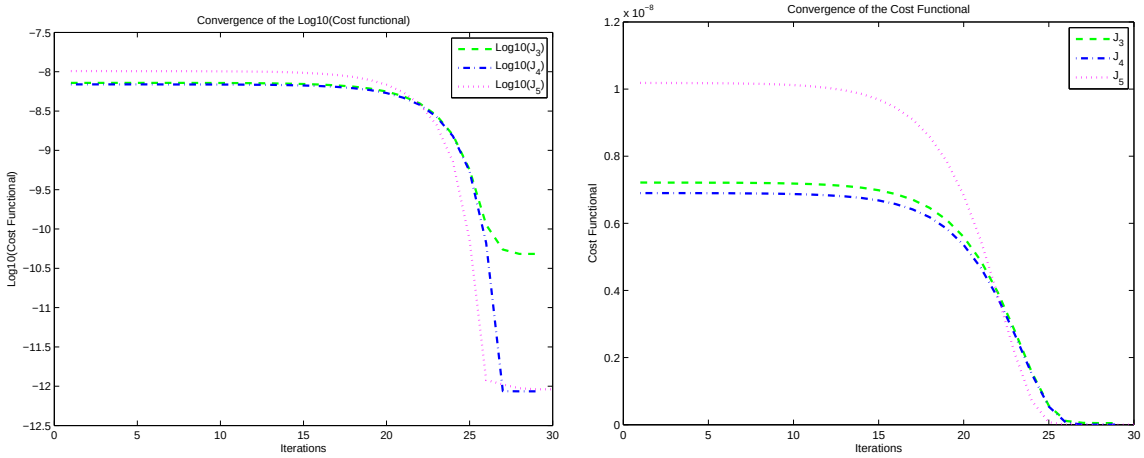
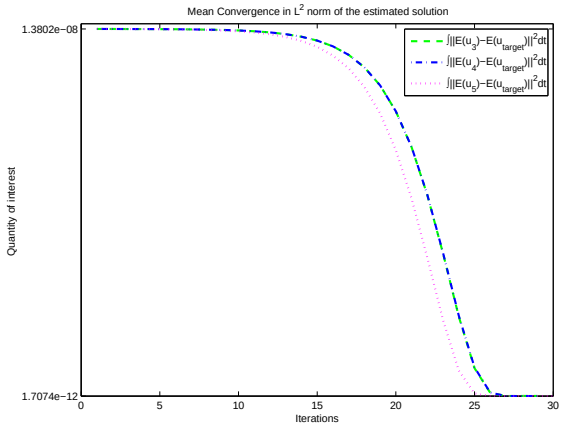


Figure 5.21: Mean $\mp$ Std. Deviation for Y_5 using KF, MCMC and Adj. method using J_4 with 10 realizations, $noise = 10^{-3}$

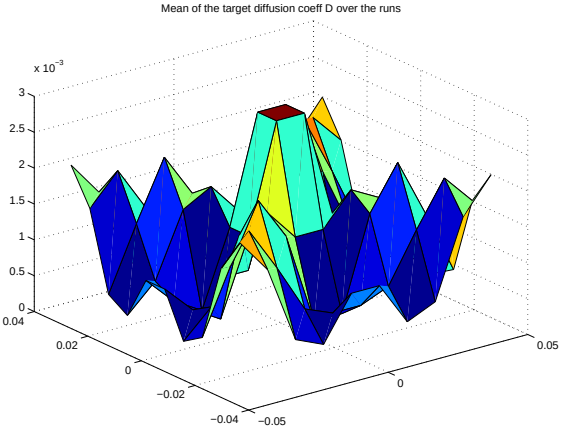


(a)

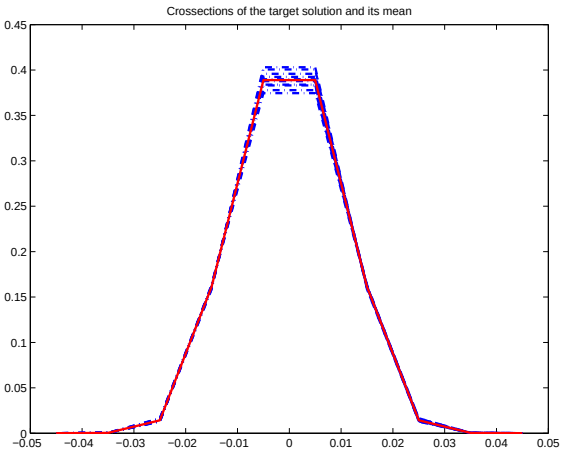
(b)



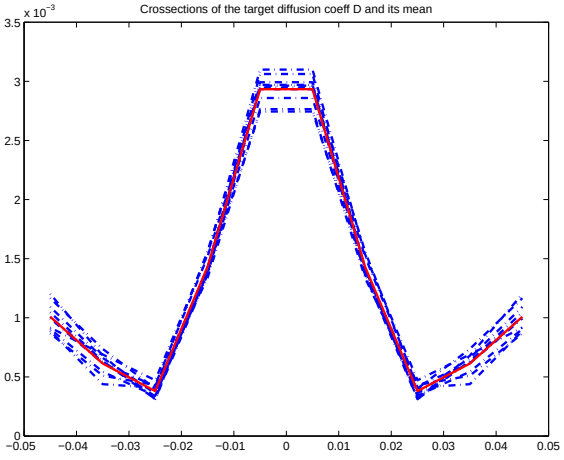
(c)



(d)



(e)



(f)

Figure 5.22: (a)$Log_{10}(J_i), i = 3, 4, 5$, (b)$J_i, i = 3, 4, 5$, (c)Mean convergence in L^2 norm of estimated solution, (d)Mean of target diff coeff. and crossections for: (e)target solution, (f)target diffusion. Grid considered is 10x10, tol=10^{-7}, $\epsilon = 50000$, $\beta = 10^{-6}$, runs=10. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-1}$.

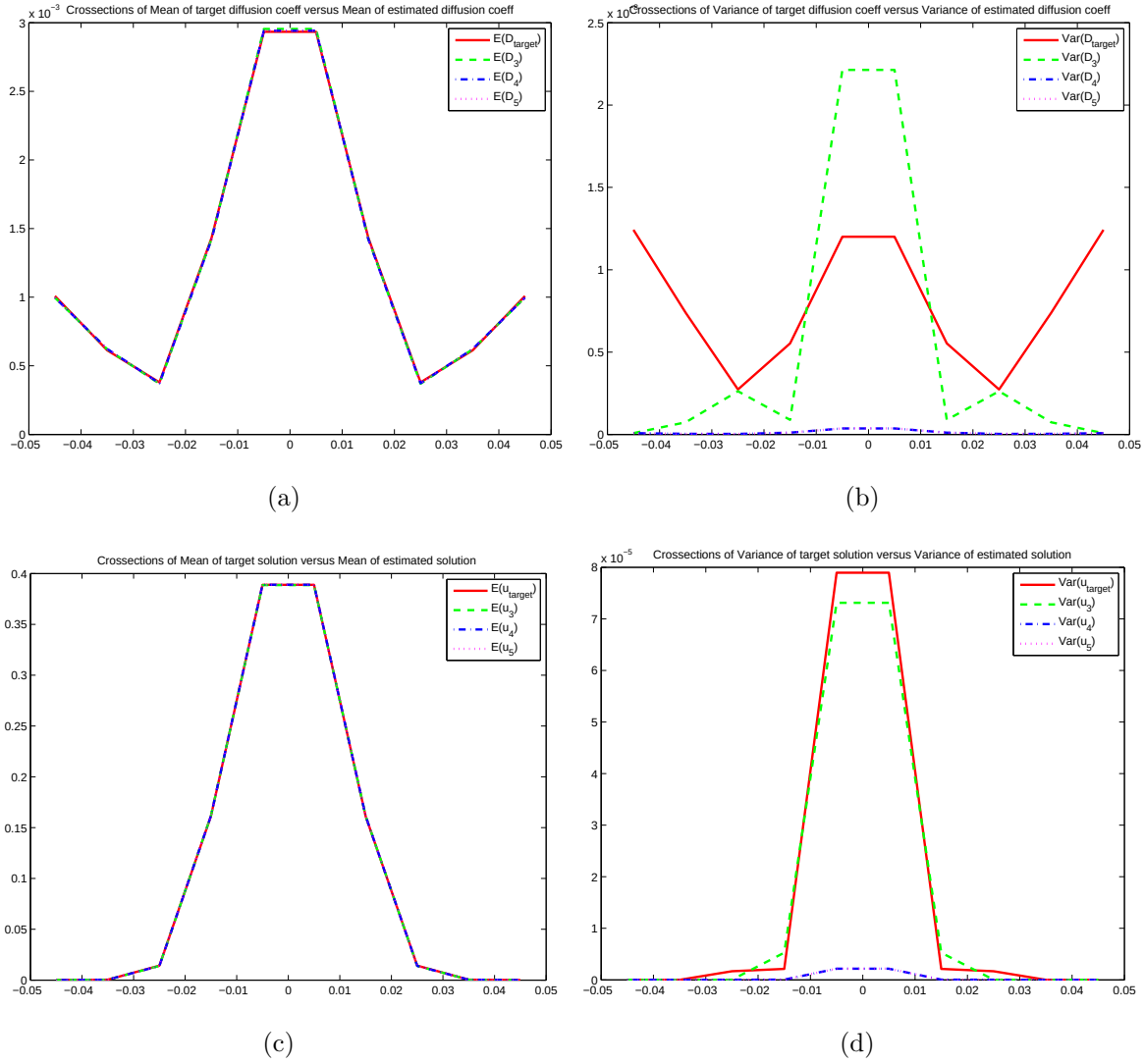
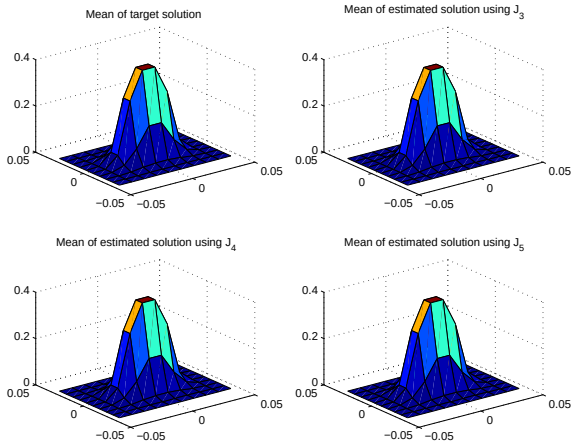
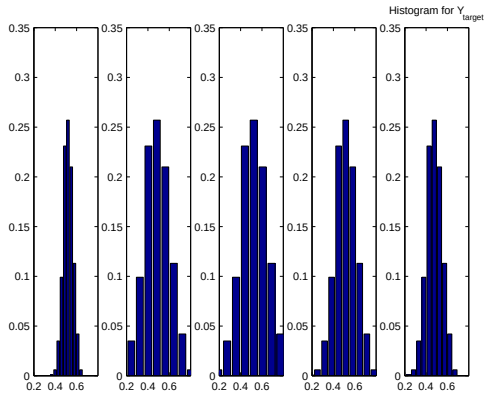


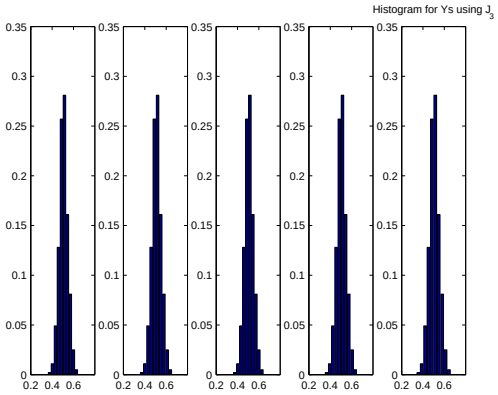
Figure 5.23: (a)Mean of target diffusion vs. mean of estimated diffusion, (b)Variance of target diffusion vs. variance of estimated diffusion, (c)Mean of target solution vs. mean of estimated solution, (d)Variance of target solution vs. variance of estimated solution. Grid considered is 10x10, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=10. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-1}$.



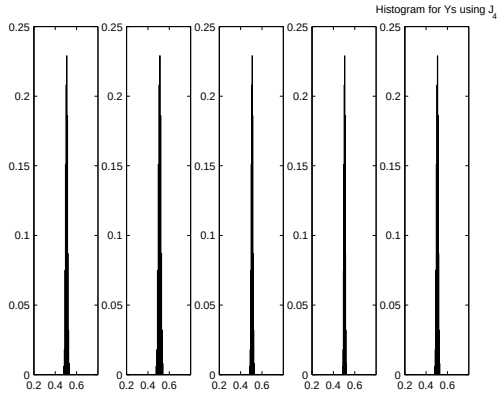
(a)



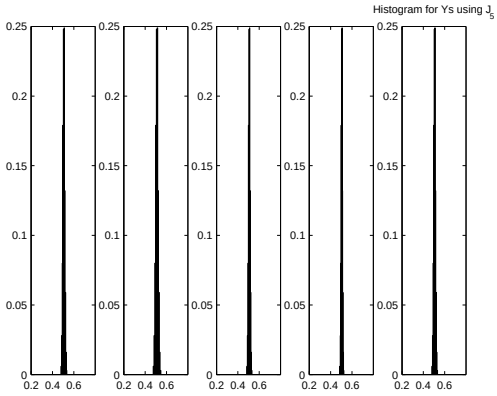
(b)



(c)

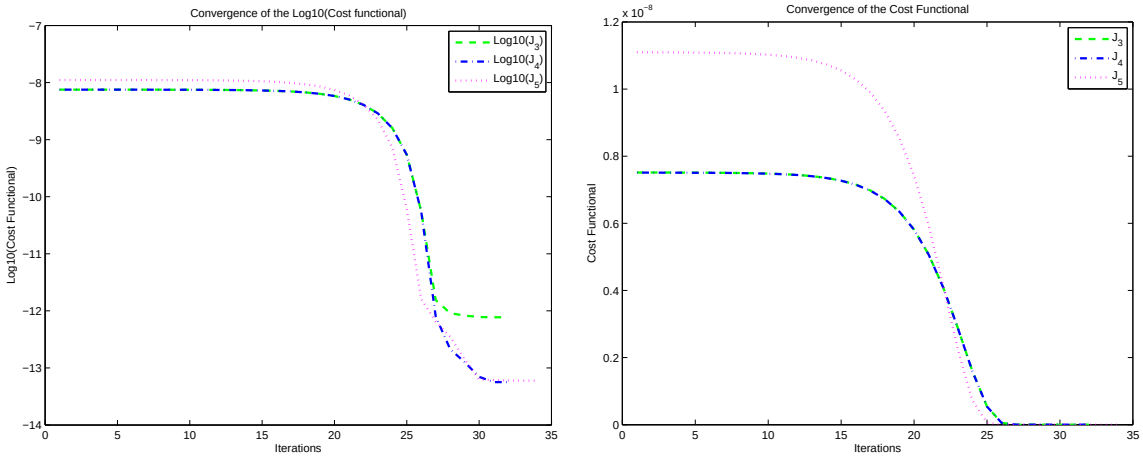


(d)



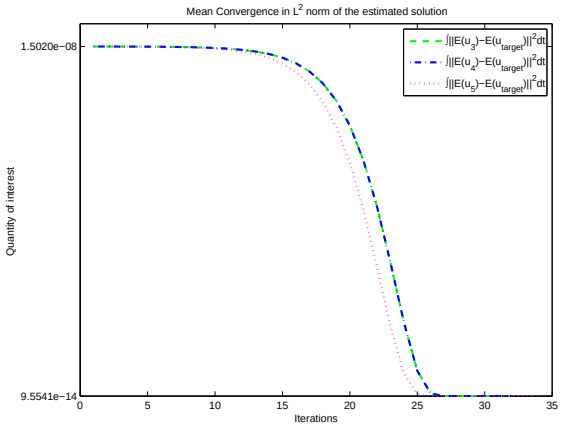
(e)

Figure 5.24: (a)mean of target solution and mean of estimated solution; histograms for: (b)target Y's, (c)estimated Y's using J_3, (d)estimated Y's using J_4, (e)estimated Y's using J_5. Grid considered is 10x10, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=10. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-1}$.

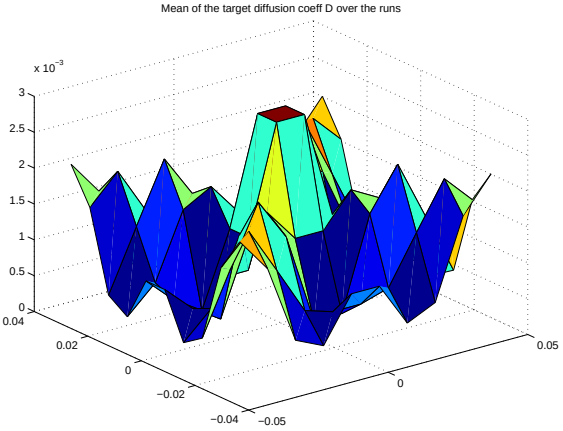


(a)

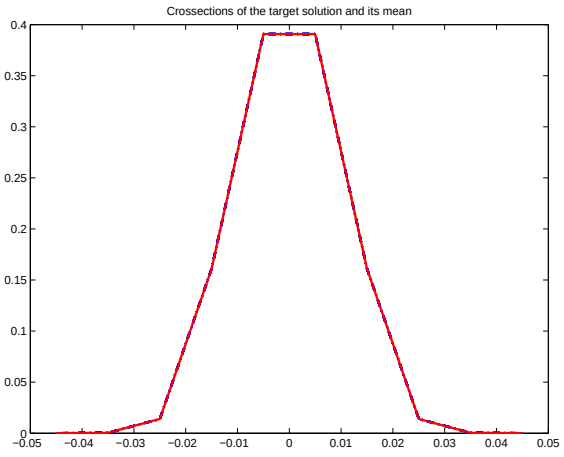
(b)



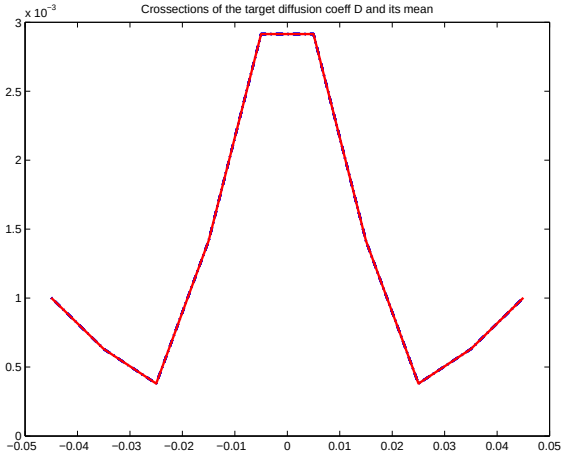
(c)



(d)

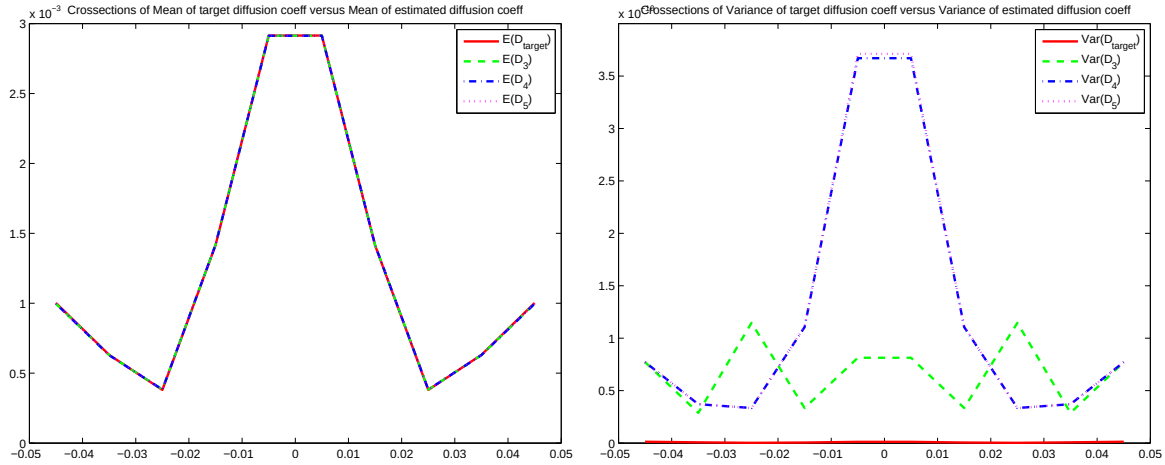


(e)

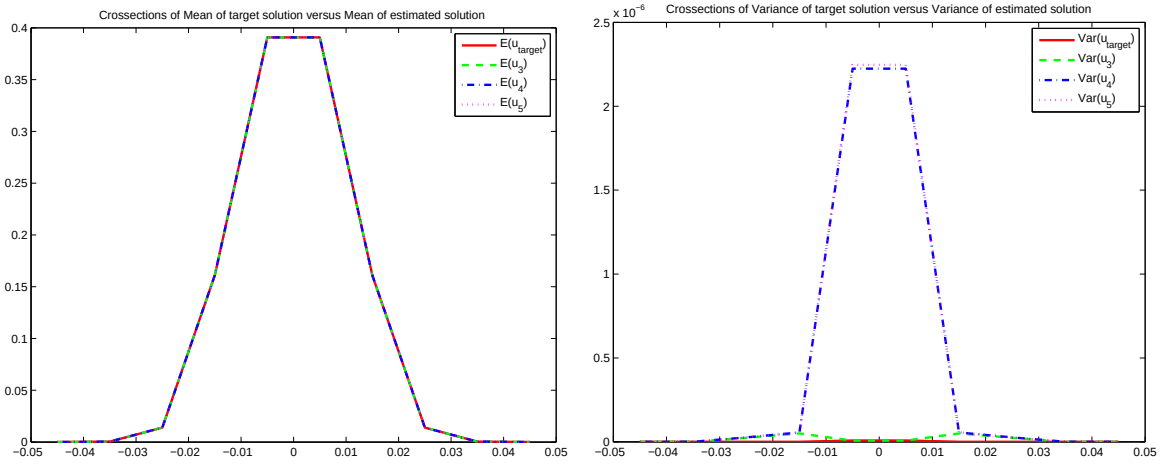


(f)

Figure 5.25: (a)$Log_{10}(J_i), i = 3, 4, 5$, (b)$J_i, i = 3, 4, 5$, (c)Mean convergence in L^2 norm of estimated solution, (d)Mean of target diff coeff. and crossections for: (e)target solution, (f)target diffusion. Grid considered is 10x10, tol=10^{-7}, $\epsilon = 50000, \beta = 10^{-6}$, runs=10. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-3}$.

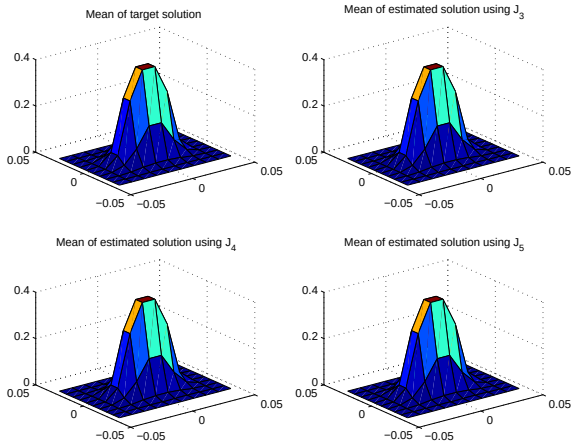


(a) (b)

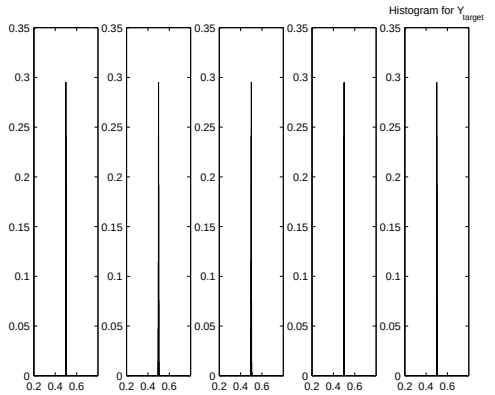


(c) (d)

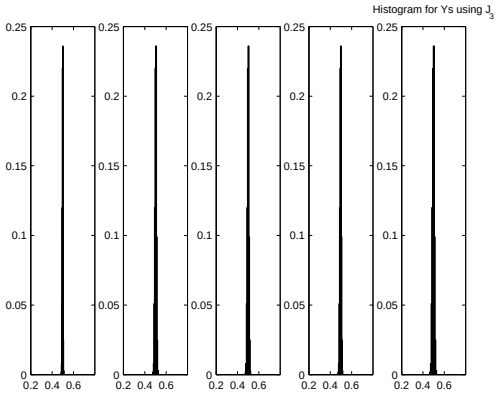
Figure 5.26: (a)Mean of target diffusion vs. mean of estimated diffusion, (b)Variance of target diffusion vs. variance of estimated diffusion, (c)Mean of target solution vs. mean of estimated solution, (d)Variance of target solution vs. variance of estimated solution. Grid considered is 10x10, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=10. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-3}$.



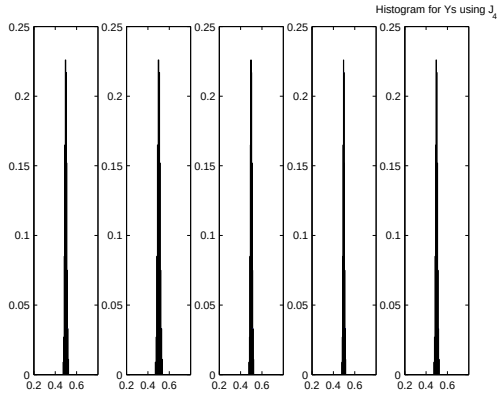
(a)



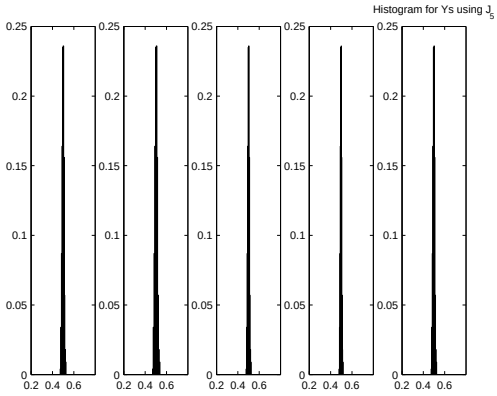
(b)



(c)

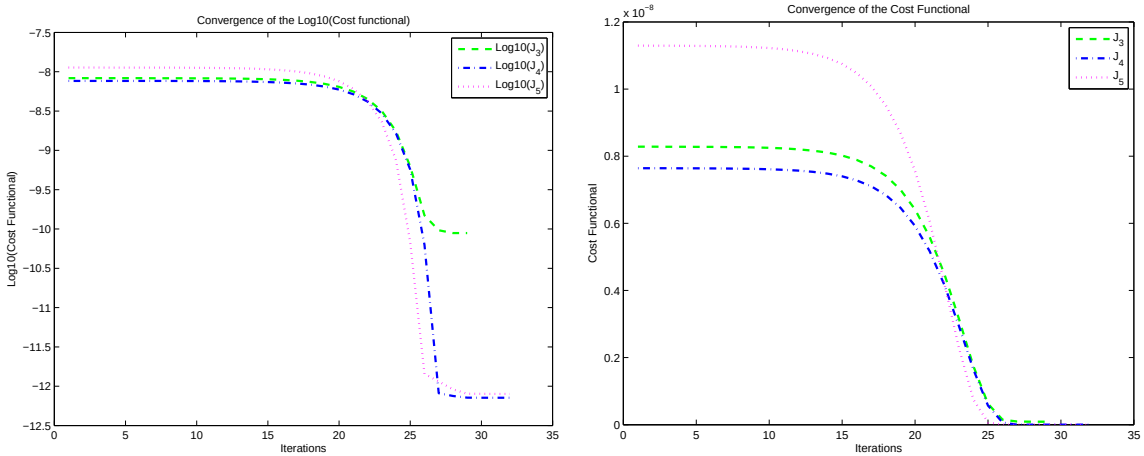


(d)



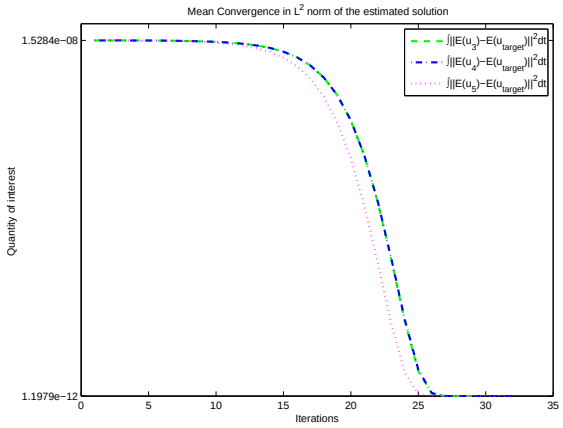
(e)

Figure 5.27: (a)mean of target solution and mean of estimated solution; histograms for: (b)target Y's, (c)estimated Y's using J_3, (d)estimated Y's using J_4, (e)estimated Y's using J_5. Grid considered is 10x10, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=10. $Y_{target} = 0.5 + noise. * randn(1,5)$, where $noise = 10^{-3}$.

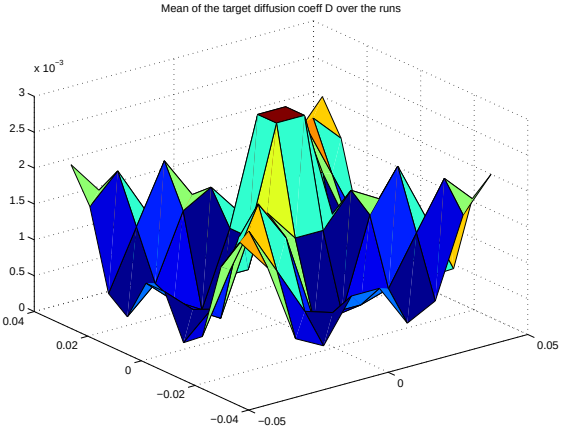


(a)

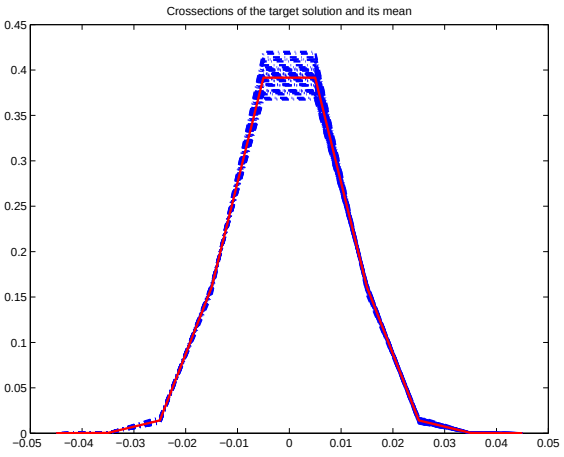
(b)



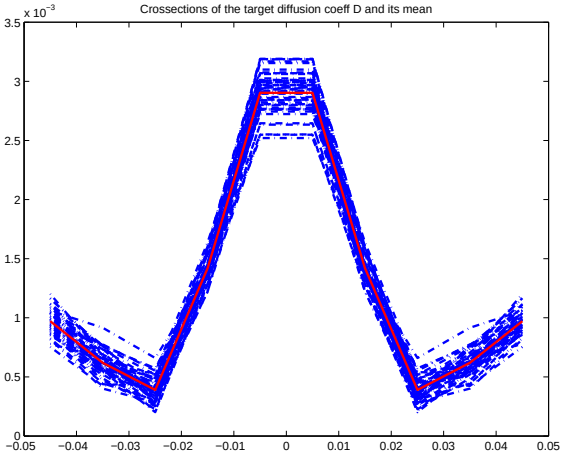
(c)



(d)



(e)



(f)

Figure 5.28: (a)$Log_{10}(J_i), i = 3, 4, 5$ (b)$J_i, i = 3, 4, 5$, (c)Mean convergence in L^2 norm of estimated solution, (d)Mean of target diff coeff. and crossections for: (e)target solution, (f)target diffusion. Grid considered is 10x10, tol=10^{-7}, $\epsilon = 50000, \beta = 10^{-6}$, runs=50. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-1}$.

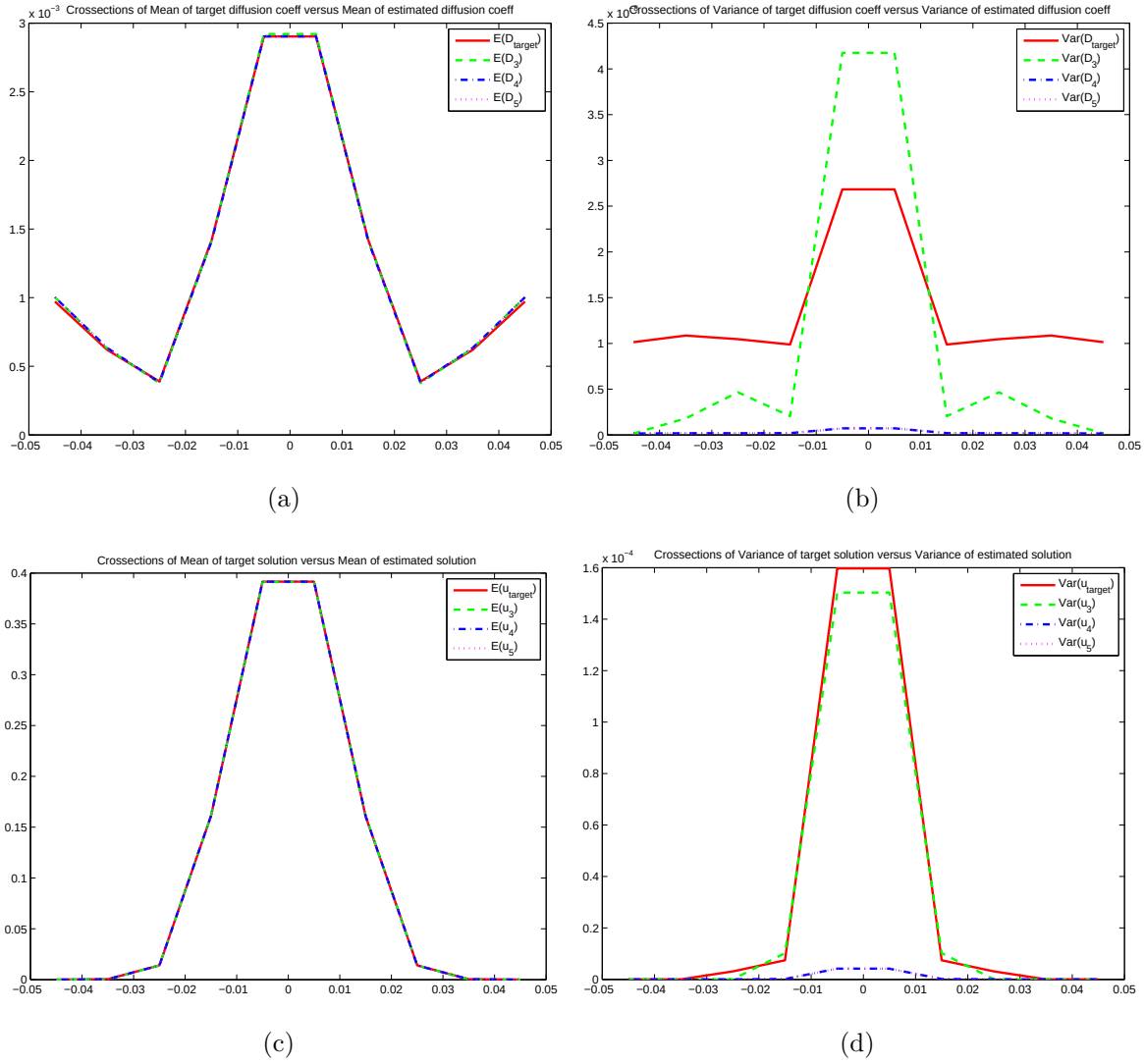
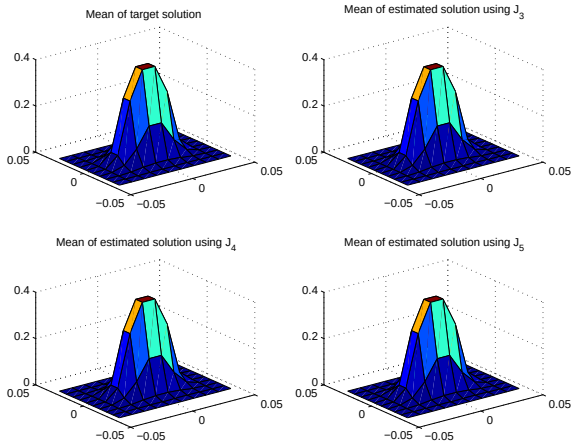
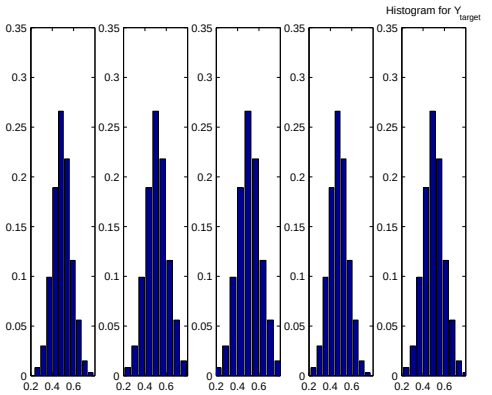


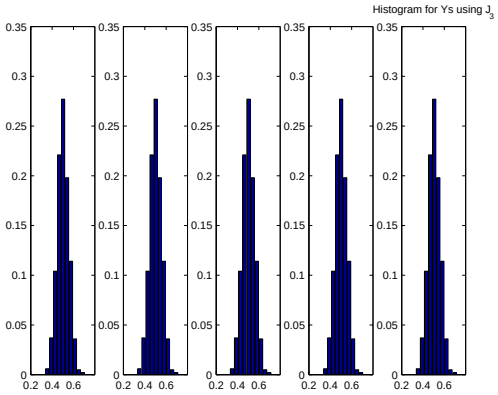
Figure 5.29: (a)Mean of target diffusion vs. mean of estimated diffusion, (b)Variance of target diffusion vs. variance of estimated diffusion, (c)Mean of target solution vs. mean of estimated solution, (d)Variance of target solution vs. variance of estimated solution. Grid considered is 10x10, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=50. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-1}$.



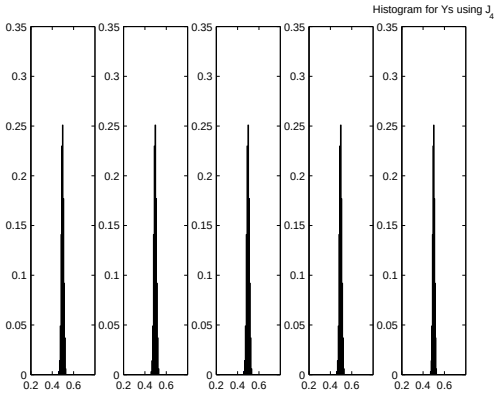
(a)



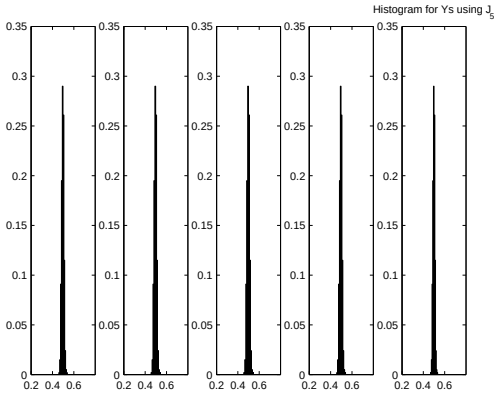
(b)



(c)

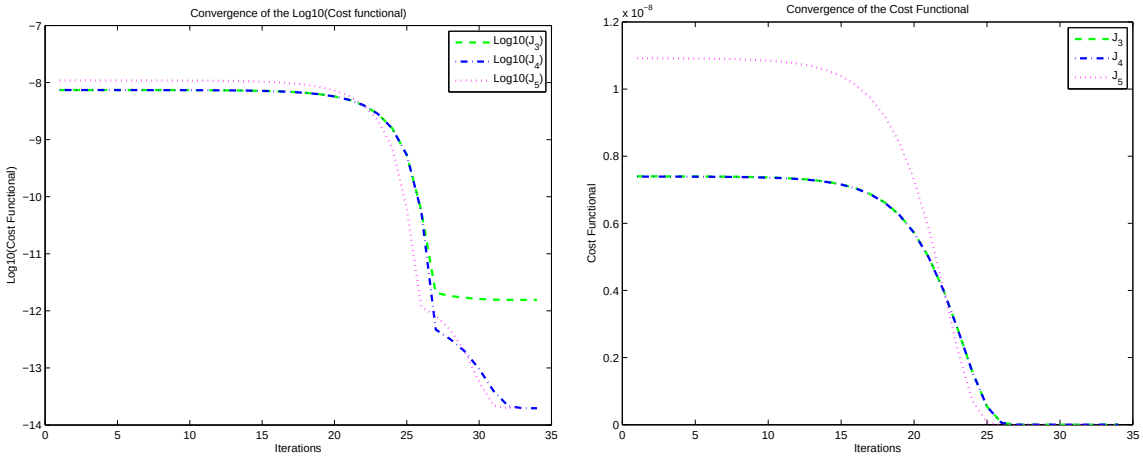


(d)



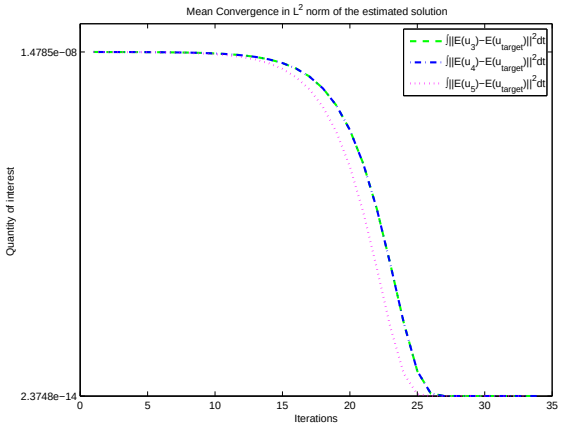
(e)

Figure 5.30: (a)mean of target solution and mean of estimated solution; histograms for: (b)target Y's, (c)estimated Y's using J_3, (d)estimated Y's using J_4, (e)estimated Y's using J_5. Grid considered is 10x10, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=50. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-1}$.

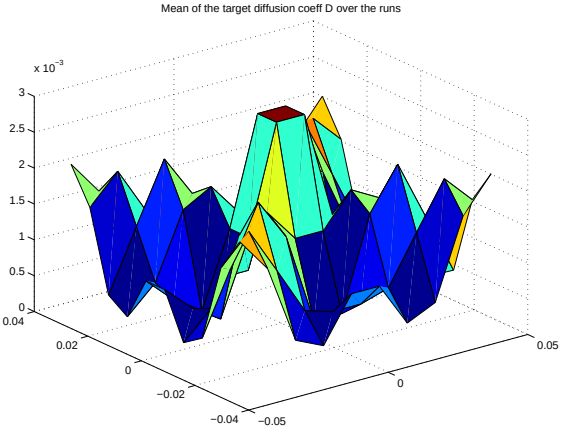


(a)

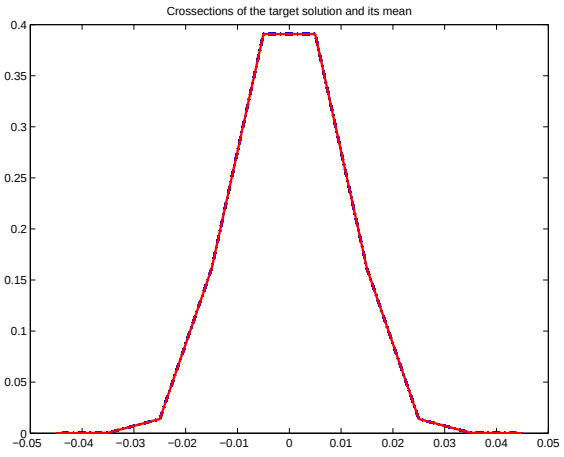
(b)



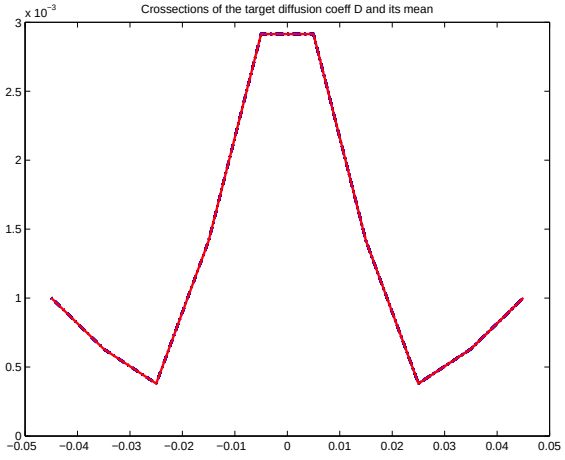
(c)



(d)



(e)



(f)

Figure 5.31: (a)$Log_{10}(J_i), i = 3, 4, 5$ (b)$J_i, i = 3, 4, 5$, (c)Mean convergence in L^2 norm of estimated solution, (d)Mean of target diff coeff. and crossections for: (e)target solution, (f)target diffusion. Grid considered is 10x10, tol=10^{-7}, $\epsilon = 50000, \beta = 10^{-6}$, runs=50. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-3}$.

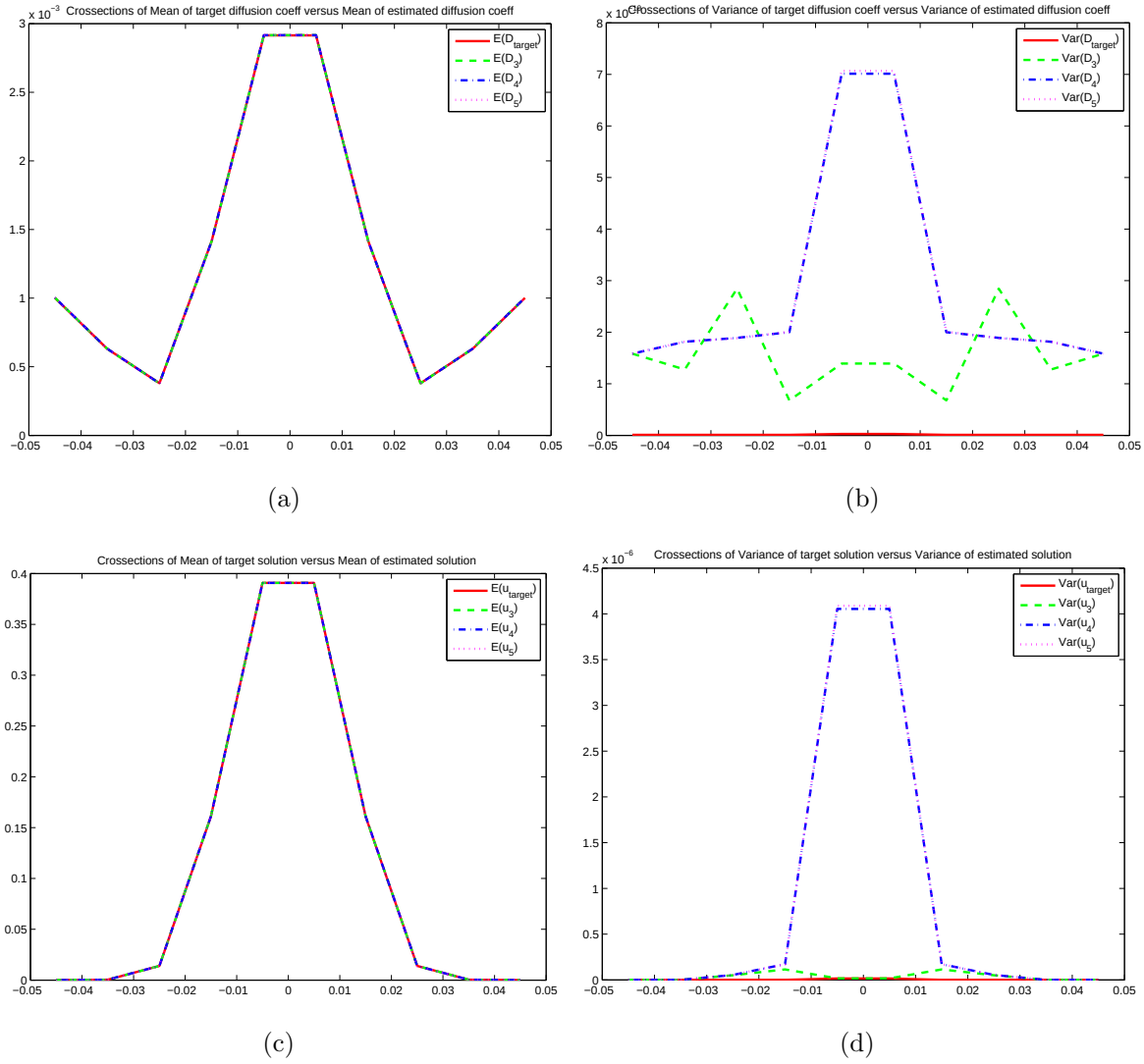
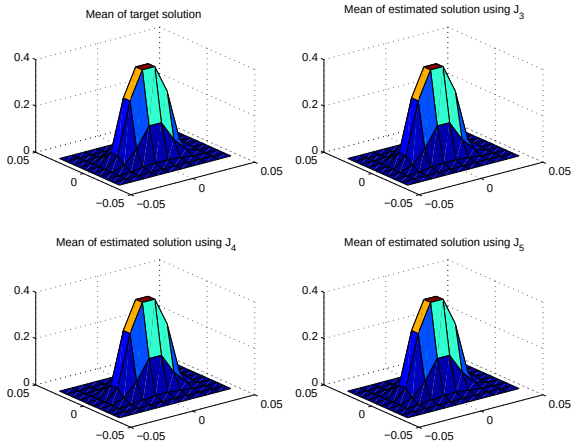
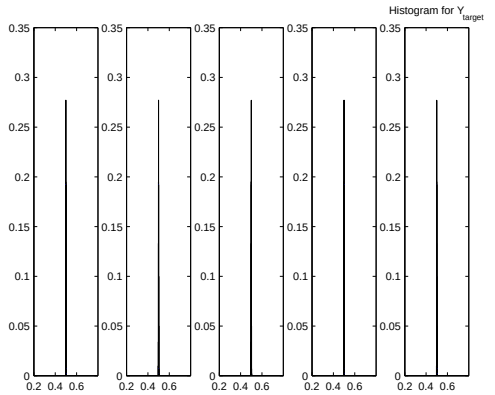


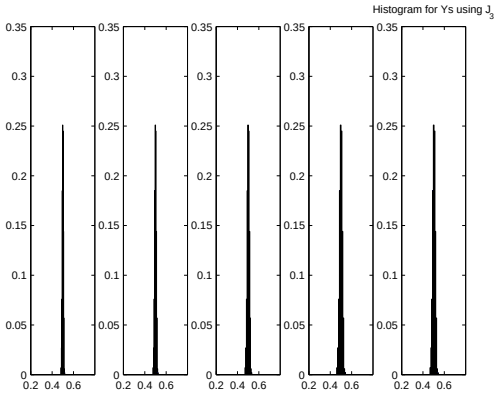
Figure 5.32: (a)Mean of target diffusion vs. mean of estimated diffusion, (b)Variance of target diffusion vs. variance of estimated diffusion, (c)Mean of target solution vs. mean of estimated solution, (d)Variance of target solution vs. variance of estimated solution. Grid considered is 10x10, tol=10^{-7}, $\epsilon = 50000$, $\beta = 10^{-6}$, runs=50. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-3}$.



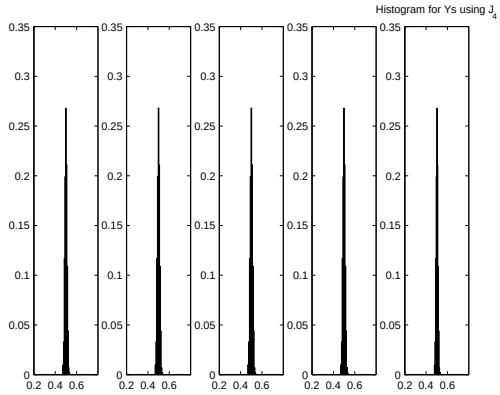
(a)



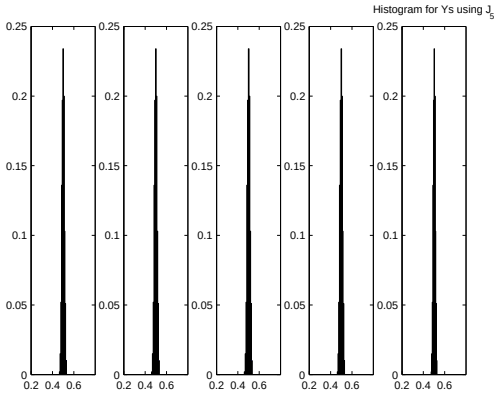
(b)



(c)



(d)



(e)

Figure 5.33: (a)mean of target solution and mean of estimated solution; histograms for: (b)target Y's, (c)estimated Y's using J_3, (d)estimated Y's using J_4, (e)estimated Y's using J_5. Grid considered is 10x10, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=50. $Y_{target} = 0.5 + noise. * randn(1,5)$, where $noise = 10^{-3}$.

5.4.2 Estimates using Adjoint variable-based algorithm on a finer mesh

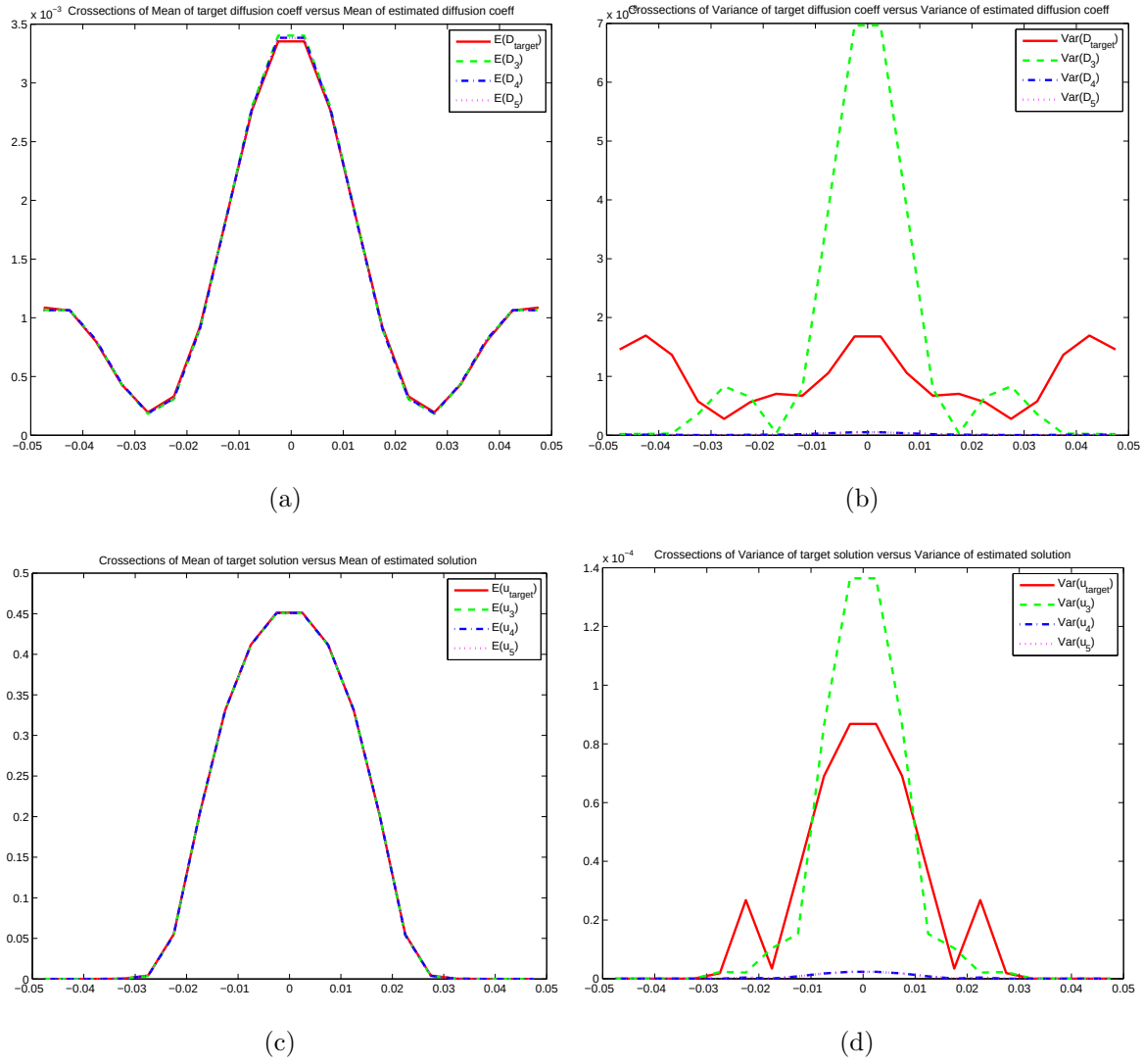
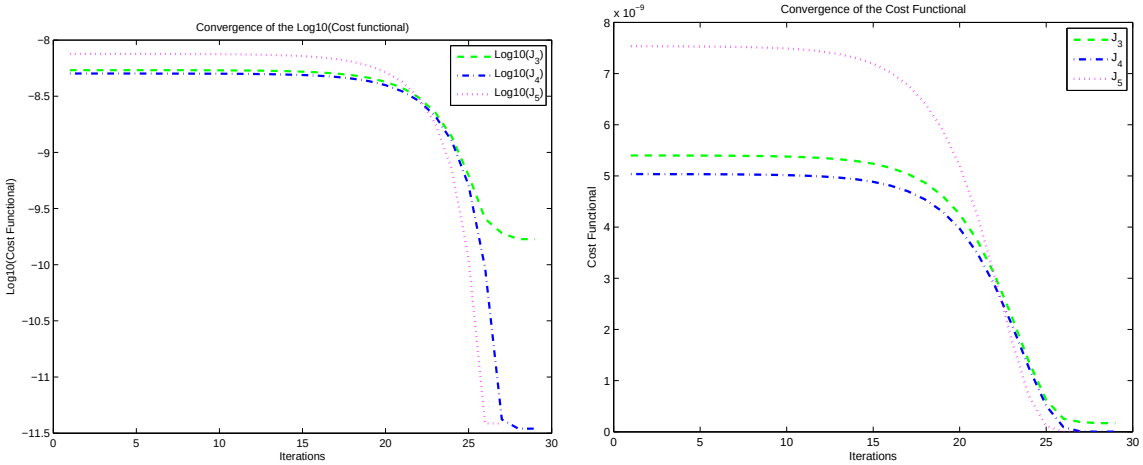
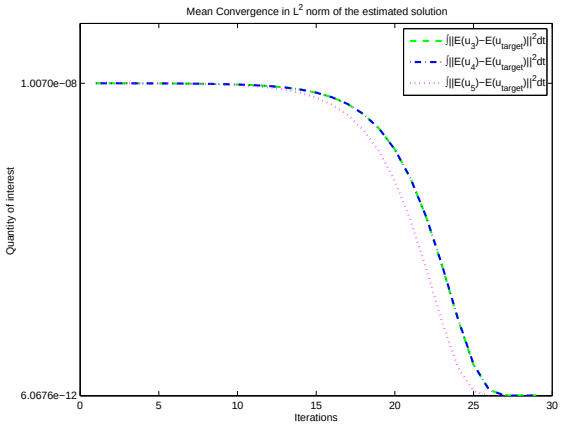


Figure 5.34: (a)Mean of target diffusion vs. mean of estimated diffusion, (b)Variance of target diffusion vs. variance of estimated diffusion, (c)Mean of target solution vs. mean of estimated solution, (d)Variance of target solution vs. variance of estimated solution. Grid considered is 20x20, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=10. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-1}$.

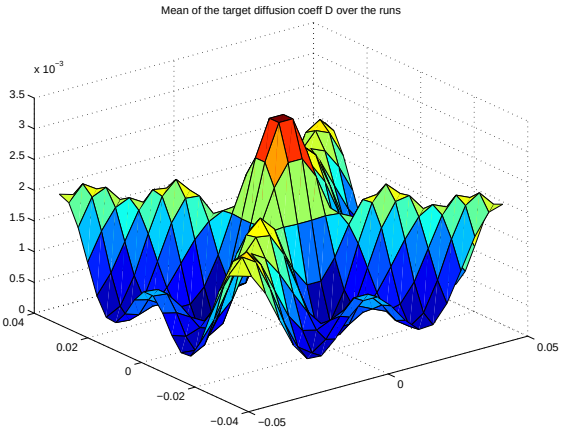


(a)

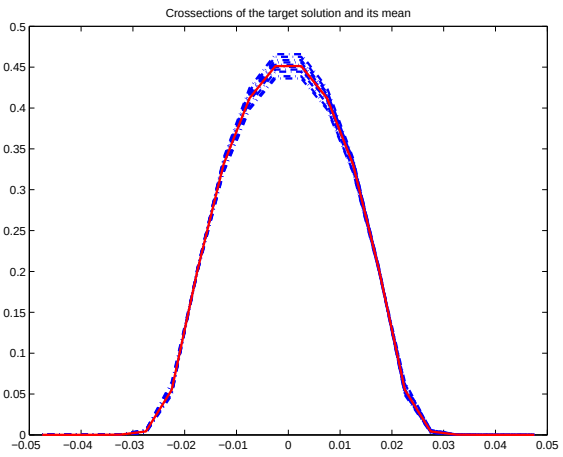
(b)



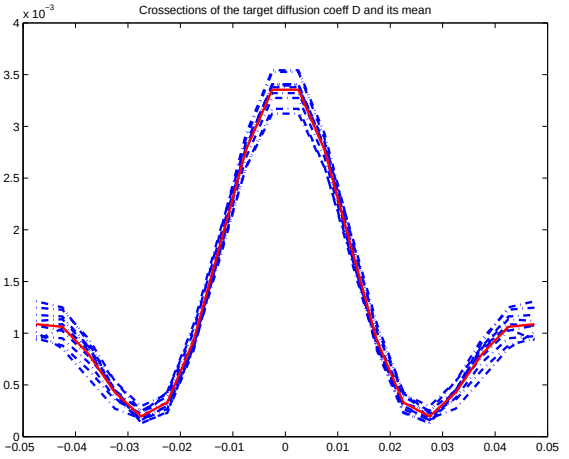
(c)



(d)

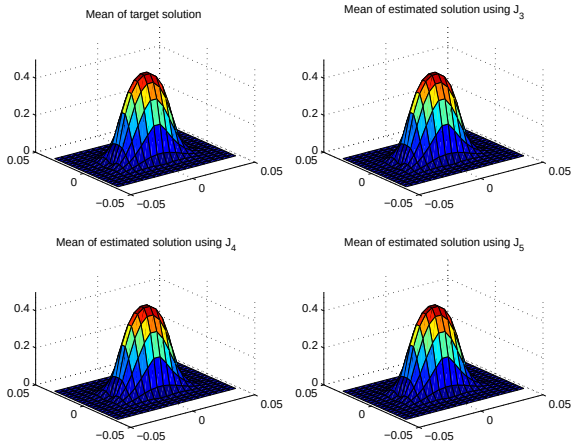


(e)

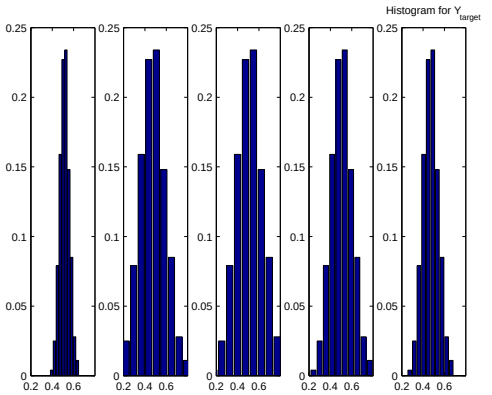


(f)

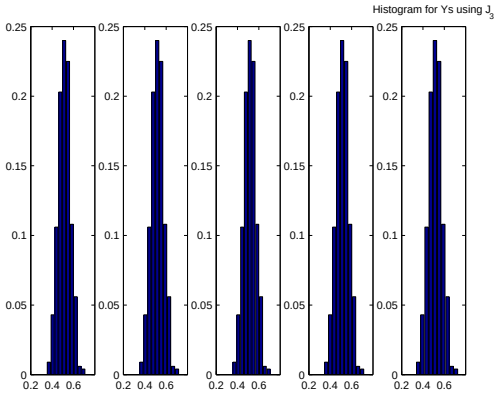
Figure 5.35: (a)$Log_{10}(J_i), i = 3, 4, 5$, (b)$J_i, i = 3, 4, 5$, (c)Mean convergence in L^2 norm of estimated solution, (d)Mean of target diff coeff. and crossections for: (e)target solution, (f)target diffusion. Grid considered is 20x20, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=10. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-1}$.



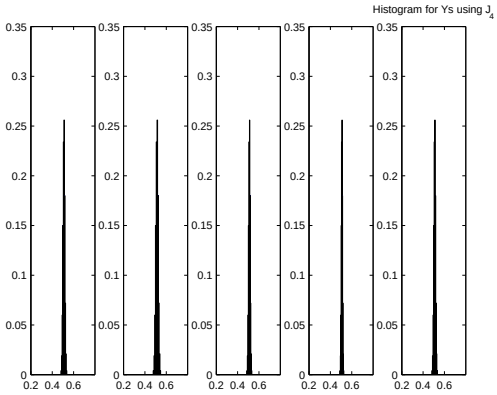
(a)



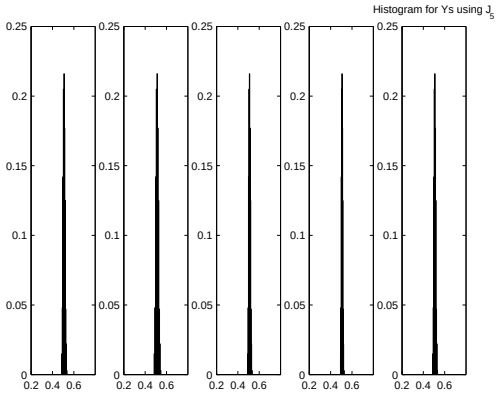
(b)



(c)



(d)



(e)

Figure 5.36: (a)mean of target solution and mean of estimated solution; histograms for: (b)target Y's, (c)estimated Y's using J_3, (d)estimated Y's using J_4, (e)estimated Y's using J_5. Grid considered is 20x20, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=10. $Y_{target} = 0.5 + noise. * randn(1,5)$, where $noise = 10^{-1}$.

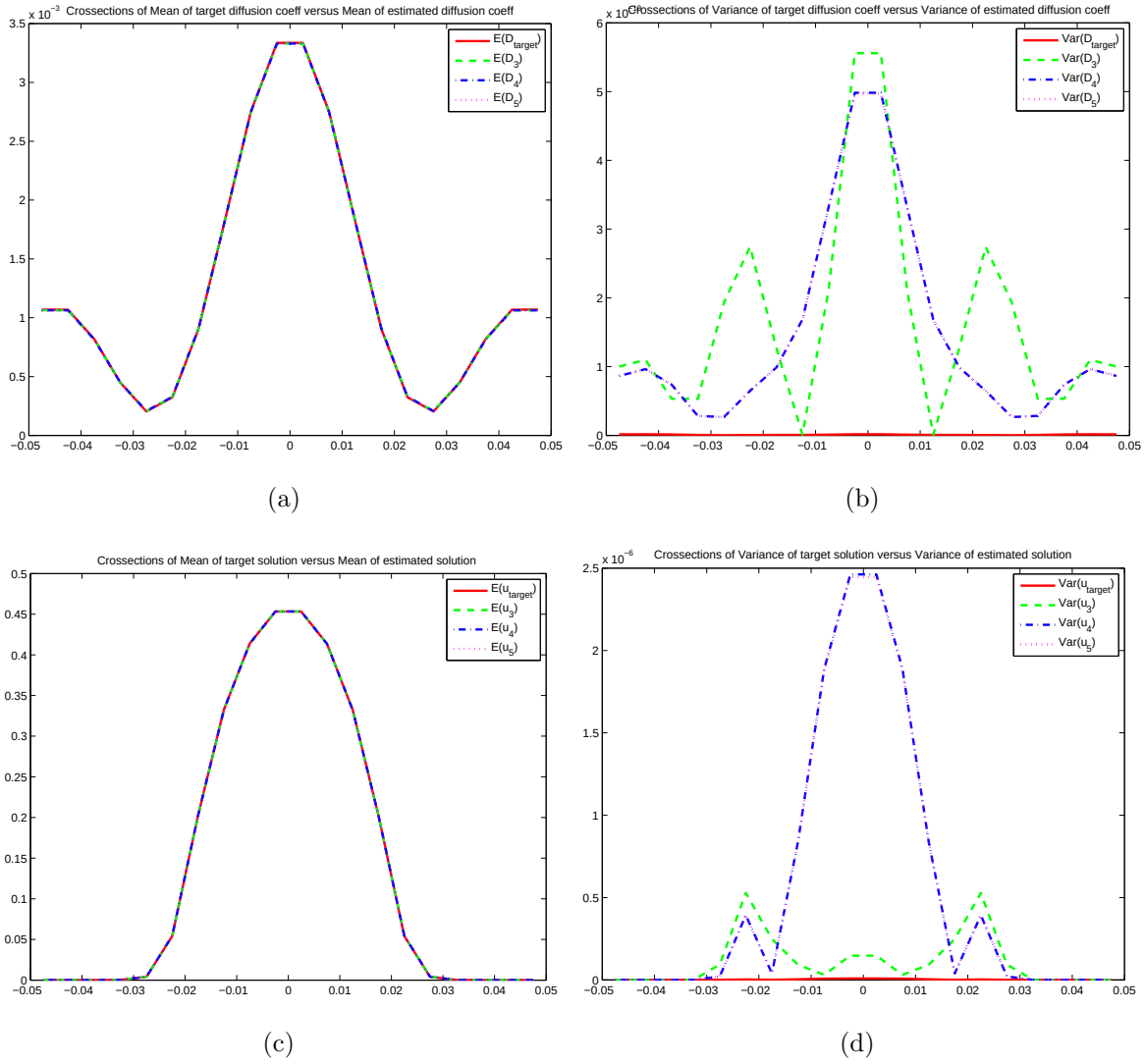
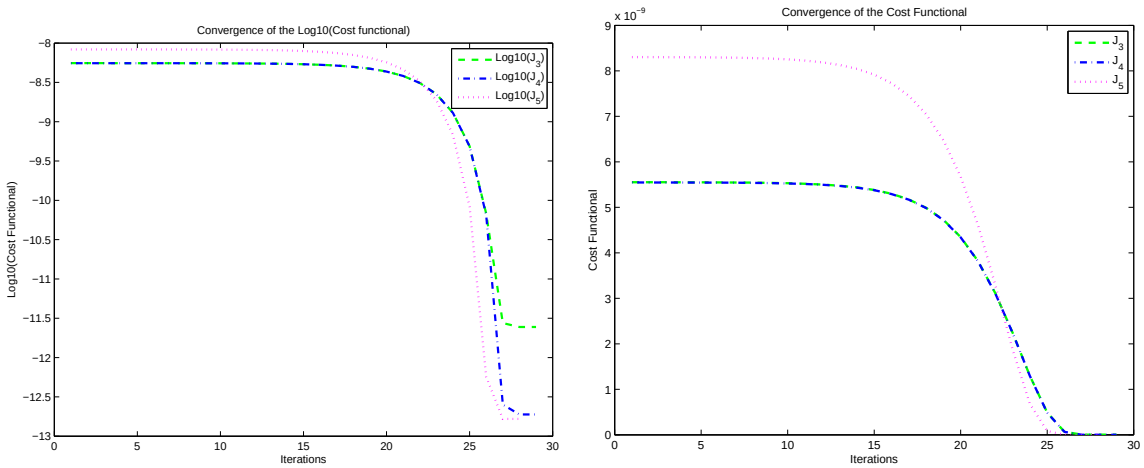
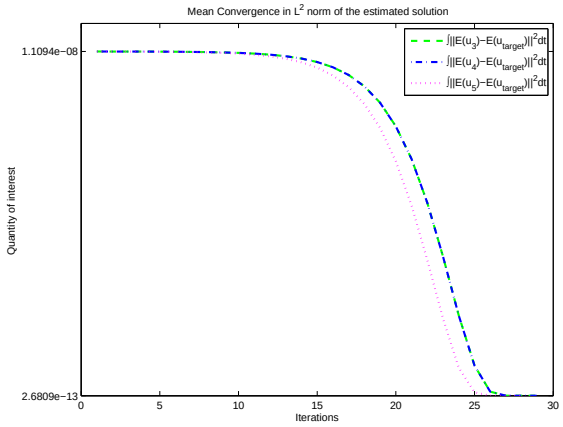


Figure 5.37: (a)Mean of target diffusion vs. mean of estimated diffusion, (b)Variance of target diffusion vs. variance of estimated diffusion, (c)Mean of target solution vs. mean of estimated solution, (d)Variance of target solution vs. variance of estimated solution. Grid considered is 20x20, tol=10^{-7}, $\epsilon = 50000$, $\beta = 10^{-6}$, runs=10. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-3}$.

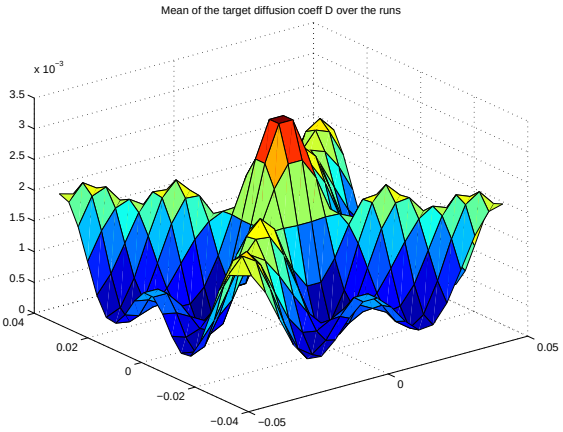


(a)

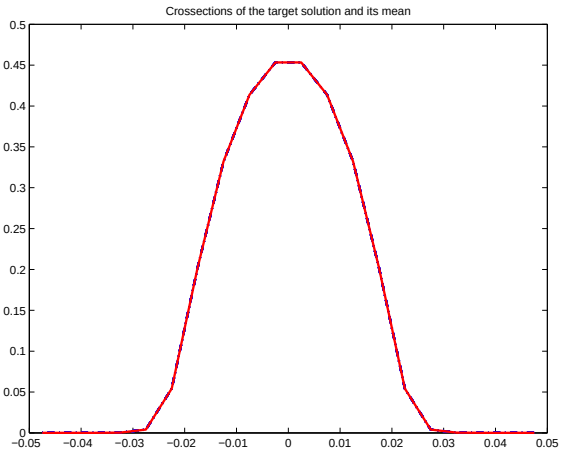
(b)



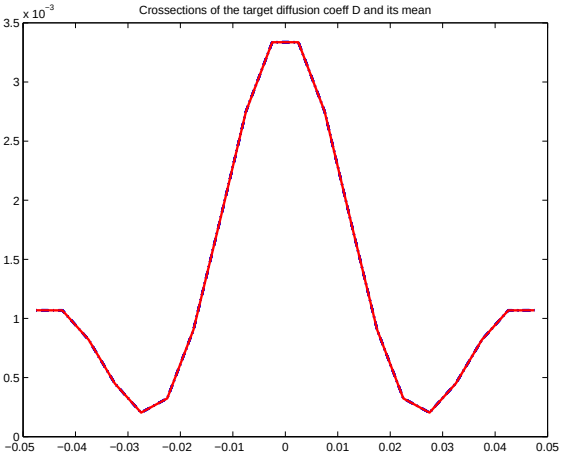
(c)



(d)

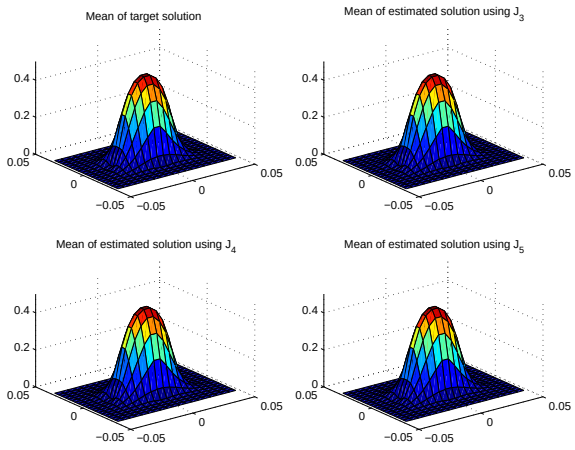


(e)

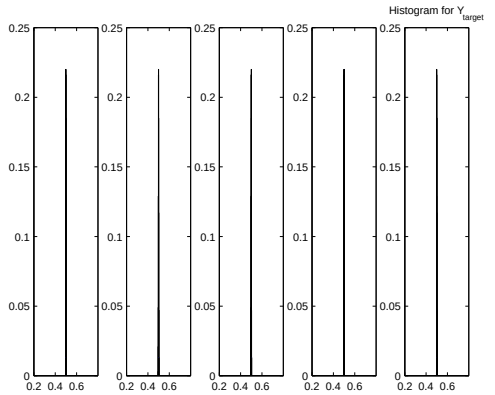


(f)

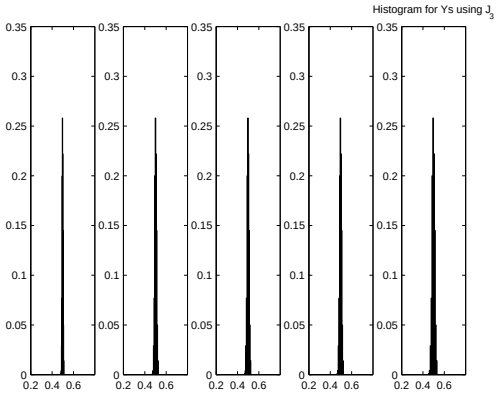
Figure 5.38: (a)$Log_{10}(J_i), i = 3, 4, 5$, (b)$J_i, i = 3, 4, 5$, (c)Mean convergence in L^2 norm of estimated solution, (d)Mean of target diff coeff. and crossections for: (e)target solution, (f)target diffusion. Grid considered is 20x20, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=10. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-3}$.



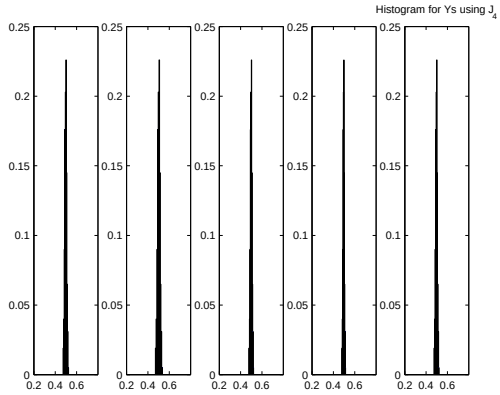
(a)



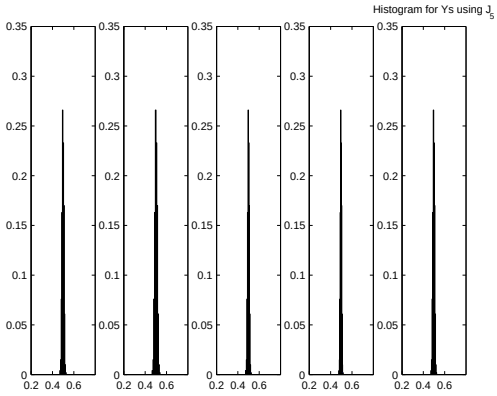
(b)



(c)



(d)



(e)

Figure 5.39: (a)mean of target solution and mean of estimated solution; histograms for: (b)target Y's, (c)estimated Y's using J_3, (d)estimated Y's using J_4, (e)estimated Y's using J_5. Grid considered is 20x20, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=10. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-3}$.

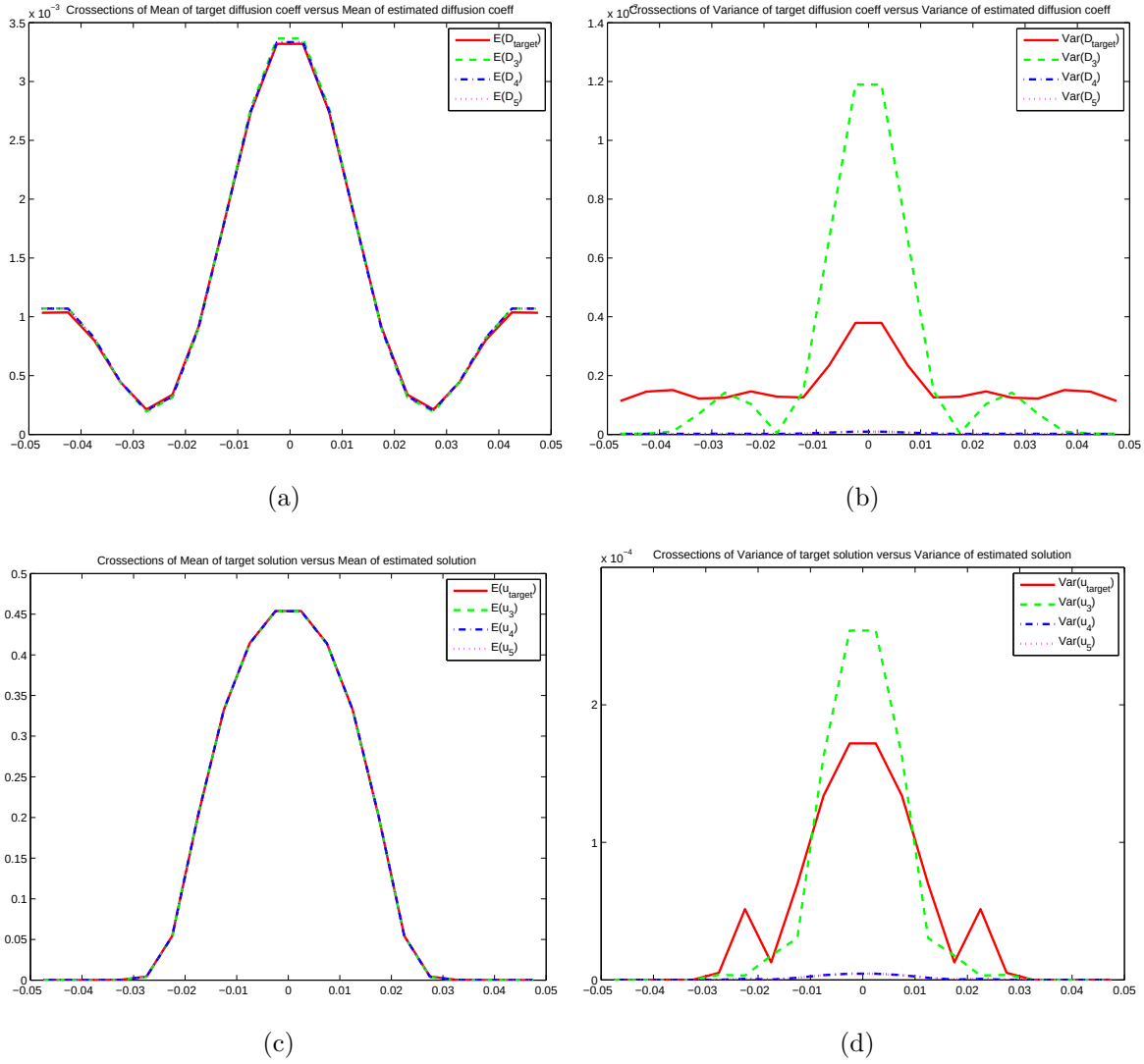
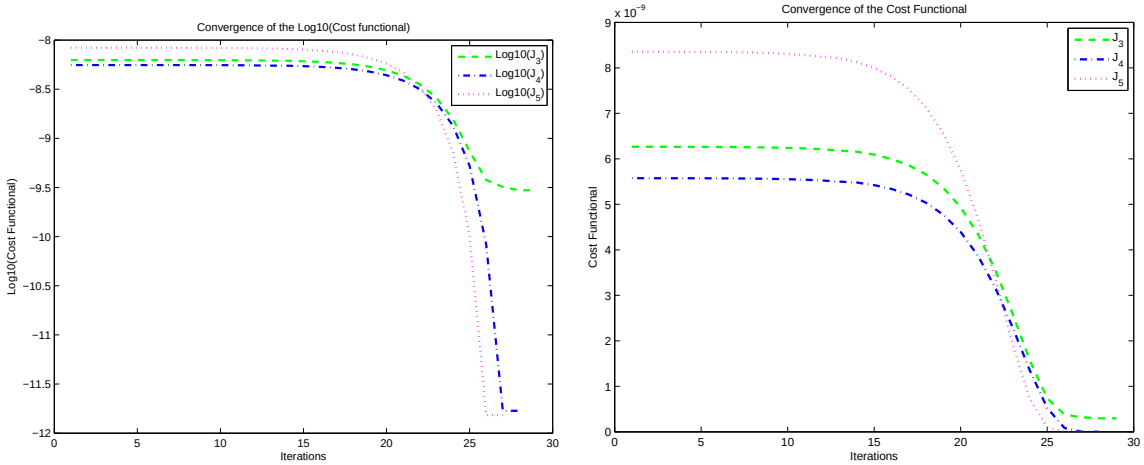
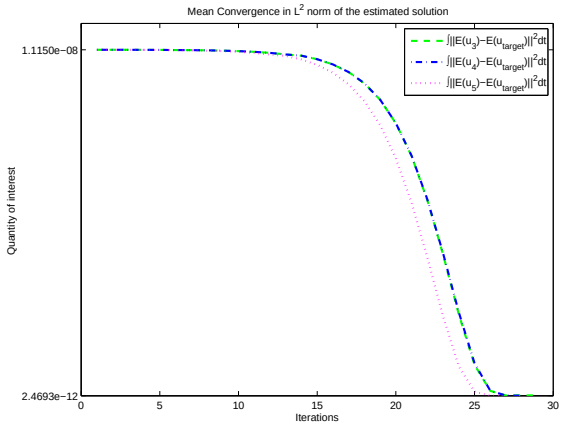


Figure 5.40: (a)Mean of target diffusion vs. mean of estimated diffusion, (b)Variance of target diffusion vs. variance of estimated diffusion, (c)Mean of target solution vs. mean of estimated solution, (d)Variance of target solution vs. variance of estimated solution. Grid considered is 20x20, tol=10^{-7}, $\epsilon = 50000$, $\beta = 10^{-6}$, runs=50. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-1}$.

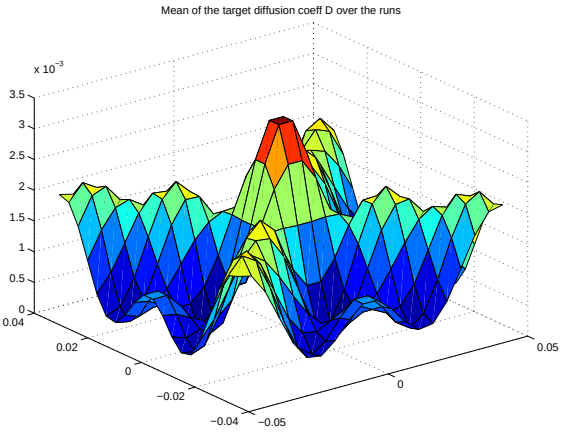


(a)

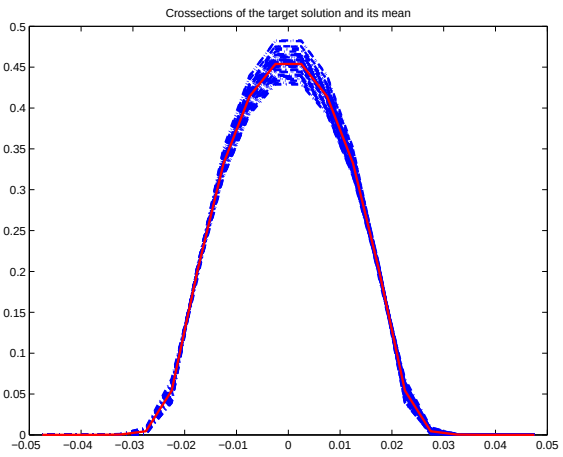
(b)



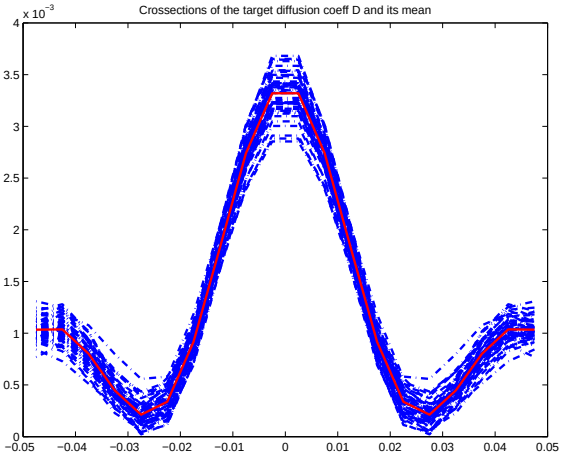
(c)



(d)

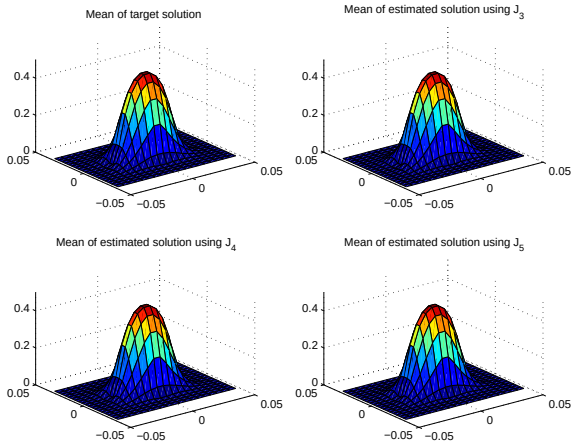


(e)

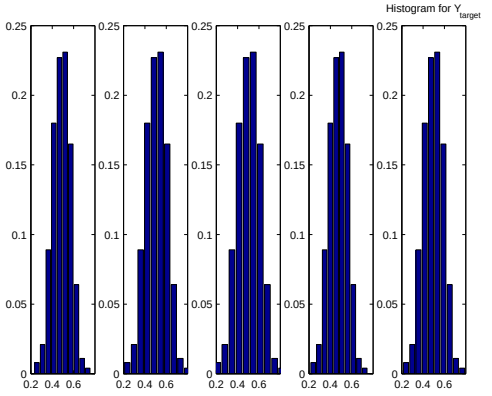


(f)

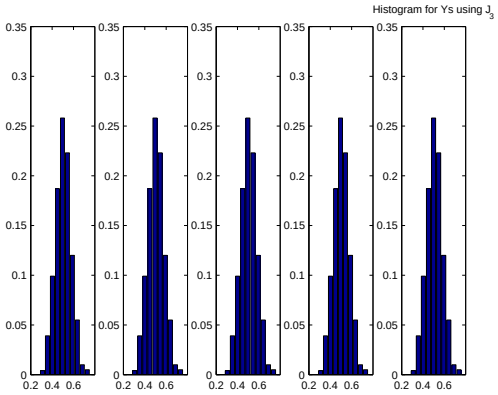
Figure 5.41: (a)$Log_{10}(J_i), i = 3, 4, 5$ (b)$J_i, i = 3, 4, 5$, (c)Mean convergence in L^2 norm of estimated solution, (d)Mean of target diff coeff. and crossections for: (e)target solution, (f)target diffusion. Grid considered is 20x20, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=50. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-1}$.



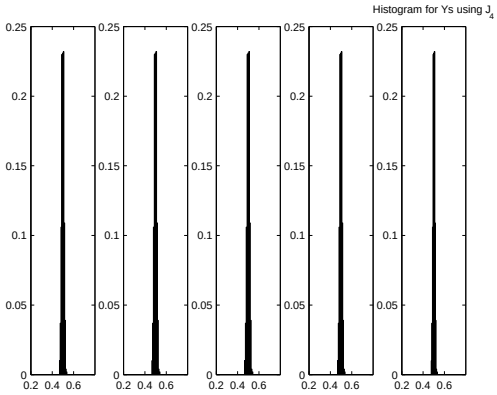
(a)



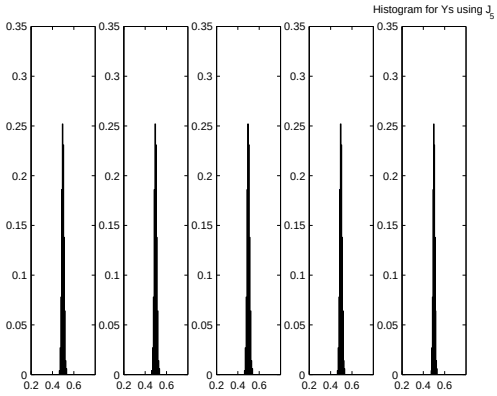
(b)



(c)

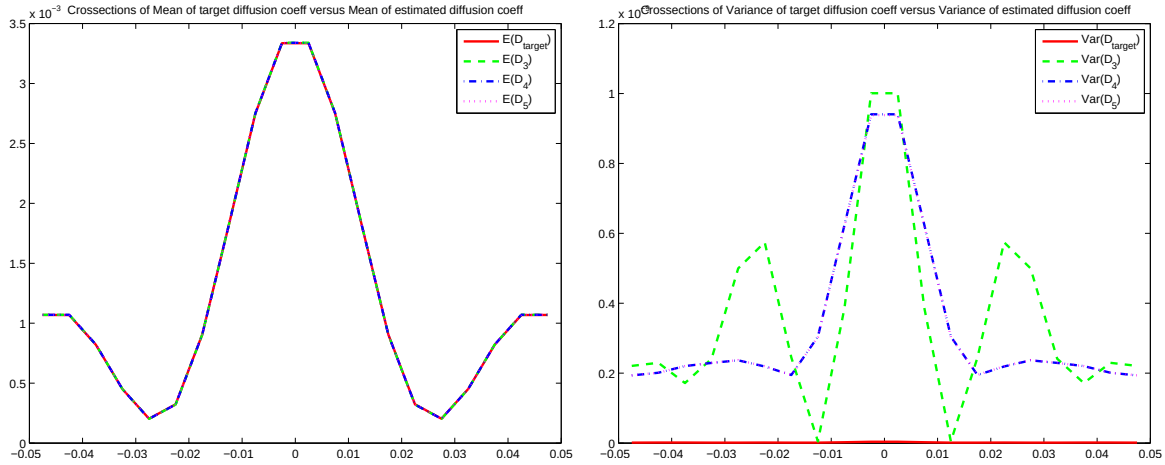


(d)

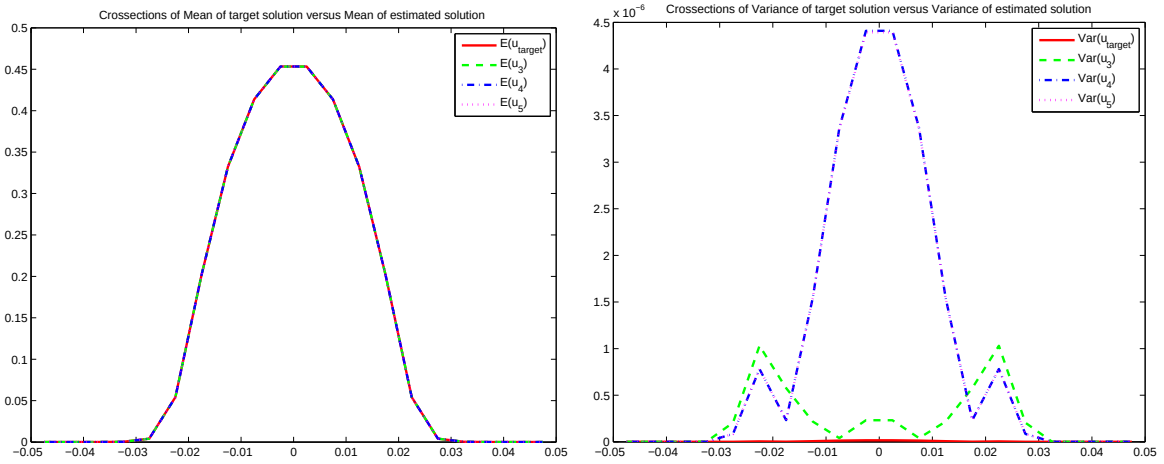


(e)

Figure 5.42: (a)mean of target solution and mean of estimated solution; histograms for: (b)target Y's, (c)estimated Y's using J_3, (d)estimated Y's using J_4, (e)estimated Y's using J_5. Grid considered is 20x20, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=50. $Y_{target} = 0.5 + noise. * randn(1,5)$, where $noise = 10^{-1}$.

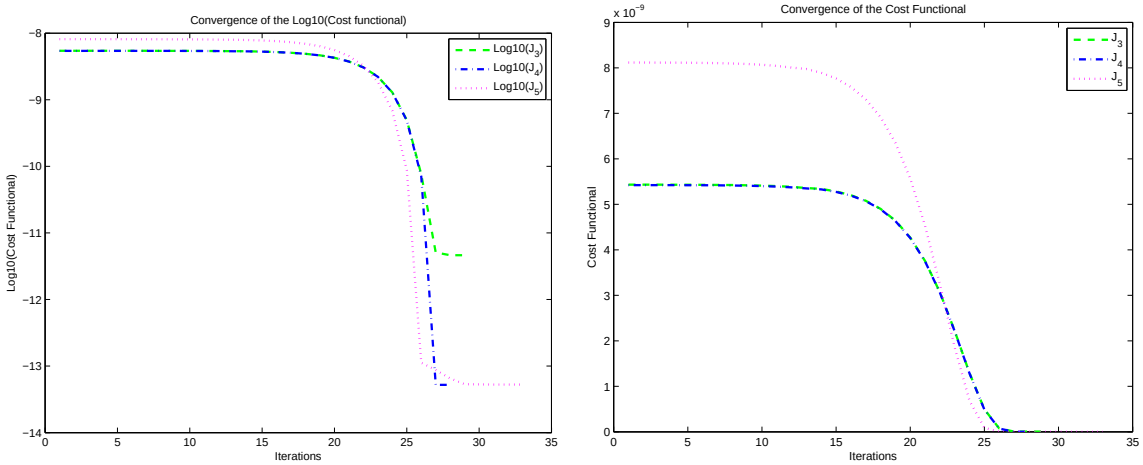


(a) (b)

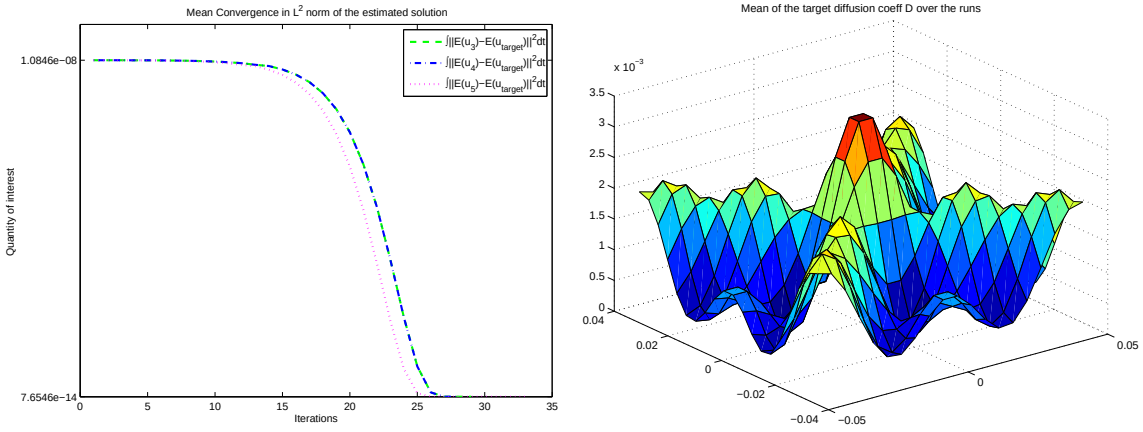


(c) (d)

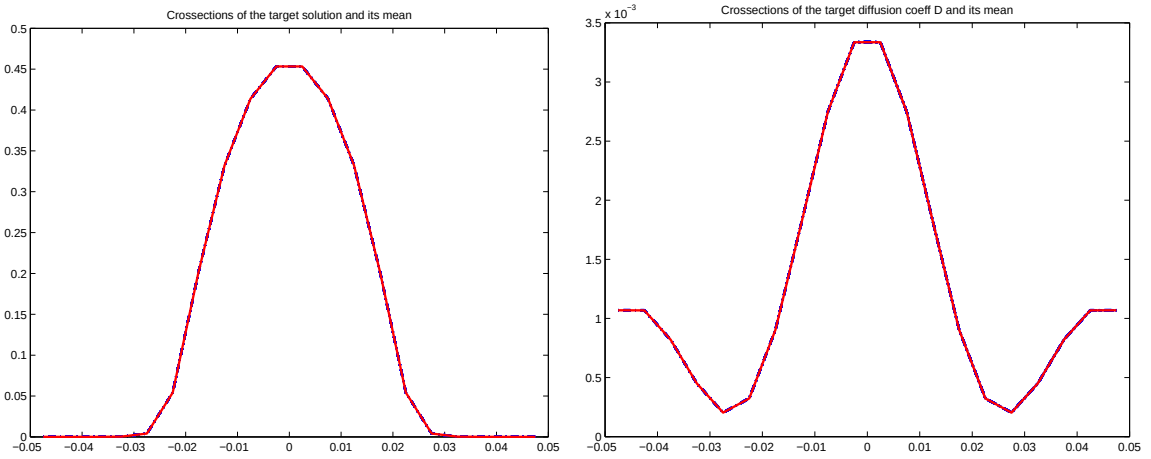
Figure 5.43: (a)Mean of target diffusion vs. mean of estimated diffusion, (b)Variance of target diffusion vs. variance of estimated diffusion, (c)Mean of target solution vs. mean of estimated solution, (d)Variance of target solution vs. variance of estimated solution. Grid considered is 20x20, tol=10^{-7}, $\epsilon = 50000$, $\beta = 10^{-6}$, runs=50. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-3}$.



(a) (b)

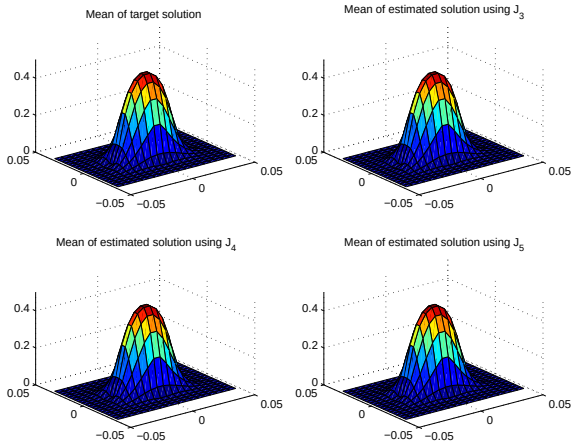


(c) (d)

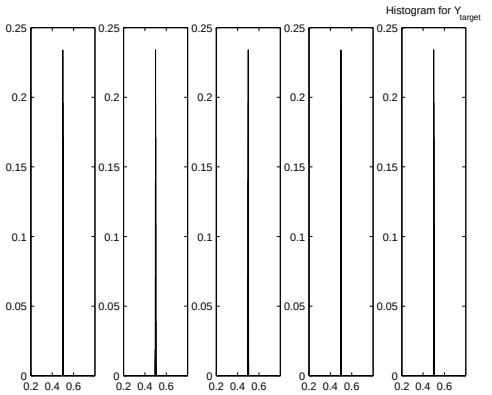


(e) (f)

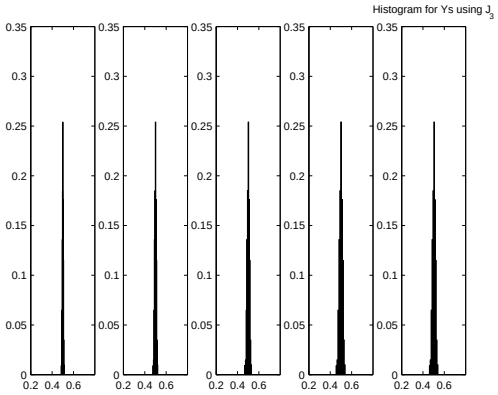
Figure 5.44: (a)$Log_{10}(J_i), i = 3, 4, 5$ (b)$J_i, i = 3, 4, 5$, (c)Mean convergence in L^2 norm of estimated solution, (d)Mean of target diff coeff. and crossections for: (e)target solution, (f)target diffusion. Grid considered is 20x20, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=50. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-3}$.



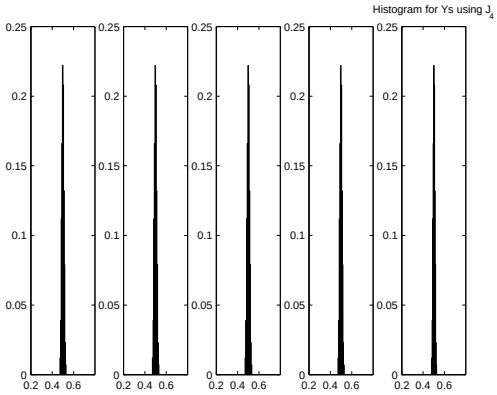
(a)



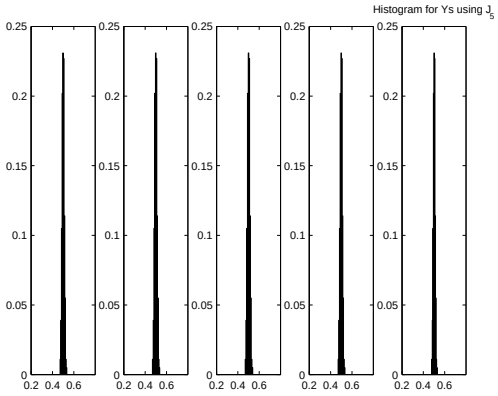
(b)



(c)



(d)



(e)

Figure 5.45: (a)mean of target solution and mean of estimated solution; histograms for: (b)target Y's, (c)estimated Y's using J_3, (d)estimated Y's using J_4, (e)estimated Y's using J_5. Grid considered is 20x20, tol=$10^{-7}, \epsilon = 50000, \beta = 10^{-6}$, runs=50. $Y_{target} = 0.5 + noise. * randn(1, 5)$, where $noise = 10^{-3}$.

6.0 CONCLUSIONS AND FUTURE WORK

6.1 THESIS SUMMARY

In chapter 3 we developed and analyzed four Kalman filter algorithms for data assimilation and parameter estimation for time dependent nonlinear diffusion equations and compared their performance for a model of epithelial cell migration. The methods are based on either random Monte Carlo sampling (ensemble methods) or structured stochastic collocation sampling. In addition, either uncorrelated random noise or correlated noise parametrized by the Karhunen-Loève expansion is considered. This results in the methods EnKF, SCKF, KLSCKF, and KLEnKF. We compared the performance of the four methods for two cases of simulated measurements, with and without noise, as well as data from *in vitro* experiment of epithelial cell migration. While it is observed that all algorithms perform reasonably well in matching the target solution and estimating the diffusion coeffcient and the growth rate, it is illustrated that the algorithms that employ SC and KL expansion are computationally more effcient, as they require fewer ensemble members for comparable accuracy. The work done for this chapter was accepted to Mathematical Biosciences [7].

In chapter 4 we formulated the stochastic optimal control theoretical results in mixed form for an elliptic diffusion equation with random input data. We proposed an Adjoint variable-based algorithm for stochastic parameter identification, that either minimize the expectation of a tracking cost functional (J_3) or minimize the difference of desired statistical quantities in the appropriate L^p norm (J_4, J_5). Some work from this chapter was incorporated in the "Identification problems for random elliptic PDEs" poster at the workshop "Computational methods for Control of Infinite-dimensional Systems", Institute for Mathematics and Applications, University of Minnesota, March 2016.

In chapter 5 we modeled a porous media flow through a parabolic diffusion equation with spatially varying permeability. We estimated the means and standard deviations of the parameters involved by using the four types of Kalman Filters aforementioned, MCMC and Adjoint variable-based algorithms.

6.2 FUTURE DIRECTIONS: PDE MODEL OF INFLAMMATION IN THE LUNG

The developed algorithms give rise to many interesting, new research problems, which will be studied as next steps of these works. We consider a PDE model of inflammation in the lung and assume that we are keeping track of eight different variables. For the first three variables, the lung is assumed to be composed of three components:

S_A air saturation

S_T tissue saturation

S_{Bl} blood saturation

In addition, five different players in the immune response are also considered:

B bacteria

M (activated) macrophages

N (activated) neutrophils

C_p pro-inflammatory cytokines

C_a anti-inflammatory cytokines

The main idea is to solve a set of reaction-diffusion equations describing the immune respone in the lungs, use the values to determine an inflammatory variable z, which will then be used to determine volumes of the alveolar air space, tissue and blood of the respective lung compartments. We have a set of equations for the other parameters above from [3], governing the immune response variables:

$$\begin{aligned} \frac{\partial B}{\partial t} - \nabla \cdot D_B(z)\nabla B &= k_{Bg}B(1 - B/B_{max}) - k_B B/(1 + B/\epsilon) \\ &\quad - R(C_a)(k_{MB}MB + k_{NB}NB) \\ \frac{\partial M}{\partial t} - \nabla \cdot (D_M(z)\nabla M &- \gamma_{MC_p}(z)M\nabla C_p - \gamma_{MB}(z)M\nabla B) \end{aligned} \tag{6.2.1}$$

$$= -k_M M + R(C_a)(k_{BM} B + k_{C_p M} C_p) M_{rest} \tag{6.2.2}$$

$$\frac{\partial C_p}{\partial t} - \nabla \cdot D_{C_p}(z) \nabla C_p = -k_{C_p} C_p + k_{M C_p} M + k_{N C_p} N -$$

$$R(c_a)(k_{C_p N} C_p N_{rest} + k_{C_p M} C_p M_{rest}) \tag{6.2.3}$$

$$\frac{\partial C_a}{\partial t} - \nabla \cdot D_{C_a}(z) \nabla C_a = -k_{C_a} C_a + k_{cnn} \frac{Q}{1+Q} \tag{6.2.4}$$

$$\frac{\partial N}{\partial t} - \nabla \cdot (D_N(z) \nabla N - \gamma_{N C_p}(z) N \nabla C_p)$$

$$= -k_N N + R(C_a) k_{C_a N} C_a N_{rest} \tag{6.2.5}$$

$$Q = \frac{k_{C_a M} M + k_{C_a N} N}{1 + k_{nc}(C_a/\bar{C}_a)^2} \qquad R(c_a) = \frac{1}{1 + k_{nc}(C_a/\bar{C}_a)^2} \tag{6.2.6}$$

Diffusion and chemotaxis

To complete the equations, we have to define diffusion and advection coefficients that depend on the air, blood, and tissue saturations. A simple definition would be:

$$D(z) = D(S_A, S_{Bl}, S_T) = D_0(D_A S_A + D_{Bl} S_{Bl} + D_T S_T) \tag{6.2.7}$$

$$\gamma(z) = \gamma(S_A, S_{Bl}, S_T) = \gamma_0(\gamma_A S_A + \gamma_{Bl} S_{Bl} + \gamma_T S_T) \tag{6.2.8}$$

We have not yet defined any of the free parameters involved except D_0. Some estimates for D_0 exist in Barber et al. [3]. These values should also appear where the coupling between the lung components volumes and the reaction-diffusion equations occurs.

Model of inflammation on the lung compartments

We can use the equations from Reynolds et al. [4] to obtain an algebraic set of equations for the saturations. It is true we plan to adjust these, but we use those equations for now:

$$S_{Bl}(z) = \frac{S_{Bl_{ref}}}{1 + m_{vtb} z} \tag{6.2.9}$$

$$S_A(z) = \frac{S_{A_{ref}}}{1 + m_{vta} z} \tag{6.2.10}$$

$$S_T(z) = S_{T_{ref}} + S_{Bl_{ref}} - \frac{S_{Bl_{ref}}}{1 + m_{vtb} z} + S_{A_{ref}} - \frac{S_{A_{ref}}}{1 + m_{vta} z} \tag{6.2.11}$$

The quantity z represents the amount of inflammation in a given area. It is given by the relatively general expression:

$$z = \frac{N + k_{C_pN} C_p}{1 + k_{C_a} C_a} \tag{6.2.12}$$

Note two desired behaviors: first, when $z = 0$ (i.e. no swelling), we have $S_T = S_{T_{ref}}$, $S_A = S_{A_{ref}}$, and $S_B = S_{B_{ref}}$. Second, when $z = \infty$, $S_T = 1$, $S_A = 0$ and $S_B = 0$.

For the initial model in this development, the following assumptions are made: first, the total volume of the lung is conserved, and the total saturations of the spatial components sum to 1, i.e $S_T + S_A + S_{Bl} = 1$.

A characteristic of inflammation is the swelling of tissue cells, which is caused by leakage of fluids due to increased permeability of the capillaries [6, 4]. The second assumption is that the volume of the tissue increases over time. The inflammation variable z drives the tissue volume, and is expressed by:

$$z = \frac{N + k_{C_pN} C_p}{1 + k_{C_a} C_a}, \tag{6.2.13}$$

where k_{C_pN} and k_{C_a} are constants that determine the amount the neutrophil and pro- and anti-inflammatory components afect inflammation.

In Reynolds et al.[4], the assumption that the volumes are directly affected by the intensity of the inflammation was also made. However, in that model, volume is lost from the blood and alveolar air space as inflammation is increased, and the tissue volume is adjusted accordingly(equations (6.2.9),(6.2.10),(6.2.11)). Our lung model aims to model tissue swelling as a result of the fluid leakage, and adjust the blood and air volumes accordingly.

Another factor we wish to take into account is compliance of media in the respiratory unit. Since air is more compliant than blood, the swelling of the tissue should affect the volume of the alveolar air with a greater impact than the volume of the blood. This is taken into account by the requirement $m_A > m_{Bl}$, where m_A and m_{Bl} are constants that determine ability of tissue swelling to increase the alveolar and blood volumes, respectively.

The inflammatory variable z, described by equation (6.2.13), drives the tissue volume, which we express by:

$$S_T = 1 - \frac{1 - S_{T_0}}{1 + m_T z},$$

We note that when $z = 0$, i.e. no inflammation, the amount of tissue is at its initial saturation($S_T = S_{T_0}$), and when $z \to \infty$, then $S_T = 1$. In this case, the tissue swells to dominate the entire respiratory unit, and both the blood and alveolar saturations tend to zero. To express this into a set of linear equations, we need to have expressions for S_A and S_{Bl} that capture the above dynamics.

In order to satisfy these constraints, we first write

$$S_A = S_{A_0} - \frac{S_{A_0}(S_T - S_{T_0})}{1 - S_{T_0}},$$

and the expression for S_{Bl} follows from volume conservation ($S_{Bl} + S_A = 1 - S_T$). For things to work out, the tissue swelling ability constant is "forced" to be

$$m_A = \frac{S_{A_0}}{1 - S_{T_0}},$$

which physically is the ratio of the initial alveolar air saturation to the initial alveolar air and blood saturations. This gives the expressions

$$S_A = S_{A_0} - m_A(1 - S_{T_0})$$
$$S_{Bl} = S_{Bl_0} - (1 - m_A)(1 - S_{T_0}),$$

which forces the compliances to be fixed constants depending on initial saturations, which are reported to be about $S_{A_0} = 0.9$ and $S_{Bl_0} = S_{T_0} = 0.05$, so the greater compliance of the alveolar air region is reflected in these numbers, but it seems kind of restricted.

BIBLIOGRAPHY

[1] Julia C. Arciero, Qi Mi, Maria F. Branca, David J. Hackam, David Swigon, *Continuum model of collective cell migration in wound healing and colony expansion*, Biophysical Journal, 100(3):535–543, 2011. 1, 33, 35, 36

[2] Yves F Atchadé, Gareth O Roberts, Jeffrey S Rosenthal, *Towards optimal scaling of Metropolis-coupled Markov Chain Monte Carlo*, Statistics and Computing, 21(4):555–568, 2011. 33

[3] Jared Barber, Mark Tronzo, Christopher Horvat, Gilles Clermont, Jeffrey Upperman, Yoram Vodovotz, Ivan Yotov, *A three-dimensional mathematical and computational model of necrotizing enterocolitis*, Journal of theoretical biology, 2012. 36, 37, 38, 118, 119

[4] Angela Reynolds, G. Bard Ermentrout, Gilles Clermont, *A mathematical model of pulmonary gas exchange under inflammatory stress*, Journal of Theoretical Biology, 264:161–173, 2010. 119, 120

[5] Robert Brown, Patrick Hwang, *Introduction to random signals and applied Kalman filtering*, Wiley, 1992. 5

[6] A.C. Guyton, J.E. Hall, *Textbook of medical physiology*, Saunders, 12th edition, 2011. 120

[7] Jared Barber, Roxana Tanase, Ivan Yotov, *Kalman filter parameter estimation for a nonlinear diffusion model of epithelial cell migration using stochastic collocation and the Karhunen-Loève expansion*, accepted to Mathematical Biosciences, 2016. 33, 117

[8] Max Gunzburger, Catalin Trenchea, Clayton Webster, *A generalized stochastic collocation approach to constrained optimization for random data identification problems*, International Journal for Uncertainty Quantification, 2012. 2

[9] G. Chavent, J. Jaffre, *Mathematical models and finite elements for reservoir simulation*, North-Holland, Amsterdam, 1986. 36

[10] N. Chitnis, J.M. Hyman, J.M. Cushing, *Determining important parameters in the spread of malaria through the sensitivity analysis of a mathematical model* Bull. Math. Biol., 70(5):1272–1296, 2008. 1

[11] David J Earl, Michael W Deem, *Parallel tempering: Theory, applications, and new perspectives*, Physical Chemistry Chemical Physics, 7(23):3910–3916, 2005. 31, 33, 71

[12] Geir Evensen, *Data Assimilation: The Ensemble Kalman Filter*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. 17, 33, 34

[13] J.C. Arciero, G.B. Ermentrout, J.S. Upperman, Y. Vodovotz, J. Rubin, *Using a mathematical model to analyze the role of probiotics and inflammation in necrotizing enterocolitis*, PLoS ONE, 5:e10066, 2010.

[14] R.G. Ghanem, P.D. Spanos, *Stochastic Finite Elements: A Spectral Approach*, Springer-Verlag, New York 1991. 23, 35

[15] B. Ganis, H. Klie, M.F. Wheeler, T. Wildey, I. Yotov, D. Zhang, *Stochastic collocation and mixed finite elements for flow in porous media*, Comput. Methods Appl. Mech. Engrg., 197, 2008. 24

[16] S. Gillijns, O. Barrero Mendoza, J. Chandrasekar, B. L. R. De Moor, D.S. Bernstein, A. Ridley, *What is the Ensemble Kalman filter and how well does it work?*, Proceedings of the 2006 American Control Conference, pages 4448–4453, 2006. 2, 17, 34

[17] C. Hall, T. Porsching, *Numerical Analysis of Partial Differential Equations*, Prentice Hall, Englewood, N.J., 1990. 37, 38

[18] Simon J Julier, Jeffrey K Uhlmann, *New extension of the kalman filter to nonlinear systems*, AeroSense'97, pages 182–193. International Society for Optics and Photonics, 1997. 13, 20, 34

[19] Xiang Ma, Nicholas Zabaras, *An efficient Bayesian inference approach to inverse problems based on an adaptive sparse grid collocation method*, Inverse Problems, 25(3):035013, 2009. 2, 30, 33

[20] Nicholas Zabaras, *Solving stochastic inverse problems: a sparse grid collocation approach*, Wiley Ser. Comput. Stat., pages 291–319, 2011. 1

[21] Jingbo Wang, Nicholas Zabaras, *Hierarchical Bayesian models for inverse problems in heat conduction*, Inverse Problems, 21(1): 183–206, 2005. 1

[22] Velamur Asokan Badri Narayanan, Nicholas Zabaras, *Stochastic inverse heat conduction using a spectral approach*, International Journal for Numerical Methods in Engineering, 60(9): 1569–1593, 2004. 1

[23] Youssef Marzouk, Habib Najm, Larry Rahn, *Stochastic spectral methods for efficient Bayesian solution of inverse problems*, Journal of Computational Physics, 224(2): 560–586, 2007. 70

[24] Guillaume Bal, Ian Langmore, Youssef Marzouk, *Bayesian inverse problems with Monte Carlo forward models*, Inverse Problems and Imaging, 7(1): 81–105, 2013. 30

[25] Peng Chen, Alfio Quarteroni, *Weighted reduced basis method for stochastic optimal control problems with elliptic PDE constraint*, SIAM/ASA Journal on Uncertainty Quantification, 2(1): 364–396, 2014.

[26] Peng Chen, Alfio Quarteroni, Gianluigi Rozza, *Stochastic optimal Robin boundary control problems of advection-dominated elliptic equations*, SIAM Journal on Numerical Analysis, 51(5): 2700–2722, 2013.

[27] Alfio Quarteroni, *Numerical models for differential problems*, Modeling, Simulation and Applications, 2, 2009. 83

[28] Rudolph Van Der Merwe, Eric A. Wan, *Sigma-point kalman filters for probabilistic inference in dynamic state-space models*, Proceedings of the Workshop on Advances in Machine Learning, 2003. 13, 20

[29] H. Moradkhani, S. Sorooshian, H. Gupta, P. Houser, *Dual state-parameter estimation of hydrological models using Ensemble Kalman filter*, Advances in water resources, 28(2), 2005. 13

[30] Emmanuel Blanchard, Adrian Sandu, Corina Sandu, *Parameter Estimation Method Using an Extended Kalman Filter*, Proceedings of the Joint North America, Asia-Pacific ISTVS Conference, 2007. 13

[31] Baris Hancioglu, *Mathematical modeling of virus dynamics in immunology*, Ph.D. dissertation, University of Pittsburgh, 2007. 32

[32] F. Nobile, R. Tempone, C. G. Webster, *A sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal., 46(5), 2008. 2, 21

[33] I. Babuška, F. Nobile, R. Tempone, *A stochastic collocation method for elliptic partial differential equations with random input data*, SIAM J. Numer. Anal., 45(3), 2007. 49

[34] F. Nobile, R. Tempone, C.G. Webster, *An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data*, SIAM J. Numer. Anal., 46(5), 2008. 49

[35] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, 2007. 33, 37

[36] Francis Clarke, *Functional analysis, calculus of variations and optimal control*, Springer, 2013. 54

[37] Albert Tarantola, *Inverse Problem Theory and Methods for Model Parameter Estimation*, SIAM, 2004. 1

[38] Rudolph van der Merwe, Eric A. Wan, Simon Julier, *Sigma-Point Kalman Filters for Nonlinear Estimation and Sensor-Fusion: Applications to Integrated Navigation*, Proceedings of the AIAA Guidance, Navigation & Control Conference, pages 16–19, 2004. 13

[39] Eric A. Wan, Rudolph van der Merwe, *The Unscented Kalman Filter*, Kalman Filtering and Neural Networks, John Wiley and Sons, Inc., New York, USA, 2002. 13, 20, 34

[40] Youssef Marzouk, Habib N. Najm, *Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems*, J. Comput. Phys., 228, 2009. 33

[41] Youssef Marzouk, Dongbin Xiu, *A stochastic collocation approach to Bayesian inference in inverse problems*, Commun. Comput. Phys., 6, 2009. 33

[42] Dongbin Xiu, Jan S. Hesthaven, *High-order collocation methods for differential equations with random inputs*, SIAM J. Sci. Comput., 27(3), 2005. 21

[43] Lingzao Zeng, Dongxiao Zhang, *A stochastic collocation based Kalman filter for data assimilation*, Computational Geosciences, 14(4):721–744, 2010. 2, 20, 34

[44] Sven Zenker, Jonathan Rubin, Gilles Clermont, *From inverse problems in mathematical physiology to quantitative differential diagnoses*, PLoS Computational Biology, 3(11):e204, 2007. 1

[45] Dongxiao Zhang, Zhiming Lu, *An efficient, high-order perturbation approach for flow in random porous media via Karhunen-Loéve and polynomial expansions*, J. of Comput. Phys., 194:773–794, 2004. 24, 26

[46] Dongxiao Zhang, Zhiming Lu, Yan Chen, *Dynamic reservoir data assimilation with an efficient, dimension-reduced Kalman filter*, SPE Journal, 12(1):108–117, 2007. 2, 24, 26

[47] R. Adler, *The Geometry of Random Fields*, John Wiley & Sons, Chichester, 1981. 48

[48] H. Banks, K. Kunisch, *Estimation Techniques for Distributed Parameter Systems*, Birkhäuser, Boston, 1989. 51, 54

[49] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, *Equations of state calculations by fast computing machines*, Journal of Chemical Physics, 21(6):10871092, 1953. 30

[50] W.K. Hastings, *Monte Carlo sampling methods using Markov Chains and their applications*, Biometrika, 57(1):97109, 1970. 30

[51] V. Barbu, *Analysis and control of nonlinear infinite-dimensional systems*, vol. 190 of Mathematics in Science and Engineering, Academic Press Inc., Boston, MA, 1993. 51

[52] V. Barbu, K. Kunisch, *Identification of nonlinear elliptic equations*, Appl. Math. Optim, 33 (1996), pp. 139–167. 54

[53] G. Christakos, *Random field models in earth sciences*, Academic Press, New York, NY, 1992. 48

[54] M. Dentz, D. Tartakovsky, E. Abarca, A. Guadagnini, X. Sanchez-Vila, J. Carrera, *Variable-density flow in porous media*, J. Fluid Mech., 561 (2006), pp. 209–235. 47

[55] R. Ghanem, S. Dham, *Stochastic finite element analysis for multiphase flow in heterogeneous porous media*, Transport in Porous Media, 32 (1998), pp. 239–262. 47

[56] M. D. Gunzburger, H.-C. Lee, J. Lee, *Error estimates of stochastic optimal Neumann boundary control problems*, SIAM J. Numer. Anal., 49 (2011), pp. 1532–1552. 51

[57] M. D. Gunzburger, *Perspectives in flow control and optimization*, vol. 5 of Advances in Design and Control, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2003. 83

[58] V. Girault, P.A. Raviart, *Finite element methods for Navier-Stokes equations*, Springer Series in Computational Mathematics, vol. 5, Springer-Verlag, Berlin, 1986. 48

[59] A.V. Fursikov, *Optimal control of distributed systems. Theory and applications*, vol. 187 of Translations of Mathematical Monographs, American Mathematical Society, Providence, RI, 2000. 83

[60] V. Isakov, *Inverse problems for partial differential equations*, vol. 127 of Applied Mathematical Sciences, Springer, New York, second ed., 2006. 51

[61] K. Ito, K. Kunisch, *Lagrange multiplier approach to variational problems and applications*, vol. 15 of Advances in Design and Control, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008. 51

[62] Pekka Neittaanmaki, Jürgen Sprekels, Dan Tiba, *Optimization of elliptic systems*, Springer Monographs in Mathematics, Springer, New York, 2006. 51, 54

[63] Andreas Keese, Hermann G. Matthies, *Parallel computation of stochastic groundwater flow*, NIC Symposium 2004, Proceedings, 20 (2003), pp. 399–408. 47

[64] O. Ladyzhenskaya, N. Ural'tseva, *Equations à Dérivées Partielles de Type Elliptique*, Dunod, Paris, 1968. 54

[65] Y. Lazarev, P. Petrov, D. Tartakovsky, *Interface dynamics in randomly heterogeneous porous media*, Advances in Water Resources, 28 (2005), pp. 393–403. 47

[66] J.-L. Lions, *Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles*, Avant propos de P. Lelong, Dunod, Paris, 1968. 51, 52

[67] J.-L. Lions, *Some methods in the mathematical analysis of systems and their control*, Kexue Chubanshe (Science Press), Beijing, 1981. 51

[68] P. Neittaanmaki, J. Sprekels, D. Tiba, *Optimization of elliptic systems*, Springer Monographs in Mathematics, Springer, New York, 2006. Theory and applications. 51, 54

[69] H. Osnes, H. Langtangen, *An efficient probabilistic finite element method for stochastic groundwater flow*, Advances in Water Resources, 22 (1998), pp. 185–195. 47

[70] D. Tartakovsky, *Probabilistic risk analysis in subsurface hydrology*, Geophys. Res. Lett., 34 (2007), p. L05404. 47

[71] D. Tartakovsky, A. Guadagnini, M. Riva, *Stochastic averaging of nonlinear flows in heterogeneous porous media*, J. Fluid Mech., 492 (2003), p. 47 –62. 47

[72] D. Tartakovsky, S. Neuman, *Extension of "transient flow in bounded randomly heterogeneous domains 1. Exact conditional moment equations and recursive approximations"*, Water Resour. Res., 35 (1999), pp. 1921–1925. 47

[73] E. Vanmarcke, *Random Fields: Analysis and Synthesis*, The MIT Press, Cambridge, MA, 3^{rd} ed., 1988. 48

[74] D. Xiu, D. Tartakovsky, *Numerical methods for differential equations in random domains*, SIAM J. Sci. Comput., 28 (2006), pp. 1167–1185. 47

[75] Marcus Garvie, Philip K. Maini, Catalin Trenchea, *An efficient and robust numerical algorithm for estimating parameters in Turing systems*, Journal of Computational Physics, 229(19), 2010. 62