An efficient numerical method for solving high-dimensional nonlinear filtering problems

Mei-Heng Yueh, Wen-Wei Lin and Shing-Tung Yau

In this paper, a brief introduction of the nonlinear filtering problems and a review of the quasi-implicit Euler method are presented. The major contribution of this paper is that we propose a nonnegativity-preserving algorithm of Yau-Yau method for solving high-dimensional nonlinear filtering problems by applying quasi-implicit Euler method with discrete sine transform. Furthermore, our algorithms are directly applicable on the compact difference schemes, so that the number of spatial points can be substantially reduced and retain the same accuracy. Numerical results indicate that the proposed algorithm is capable of solving up to six-dimensional nonlinear filtering problems efficiently and accurately.

AMS 2000 subject classifications: Primary 93E11, 60G35, 62M20; Secondary 65M06, 65M12.

Keywords and Phrases: nonlinear filtering, Kolmogorov equations, discrete sine transform.

1. Introduction

The nonlinear filtering problem (NFP) is originated from the problem of tracking and signal processing [10]. The theory of filtering has been widely studied for more than 60 years. One of the most influential work on the filtering theory is the Kalman-Bucy filter [9]. It provided an iterative way to solve the linear filtering problem for the discrete data with Gaussian initial distribution, which is useful in science and engineering industries [1, 15–17]. In order to break out the restriction to the linear dynamical system with Gaussian initial distribution, the problem of nonlinear filtering is gradually getting much more attention. A significant progress on the nonlinear

filtering theory is the Yau-Yau filter [21–23], which reduces the stochastic differential equations of NFP into the Kolmogorov partial differential equation. Few years later, Luo/Yau have further extended the Yau-Yau filter to the general setting of NFP [12]. Also, they proposed a spectral method for solving one-dimensional NFP [13] in the same year. Based on the theory of Yau-Yau filter, Liu/Dong/Ding proposed an explicit finite difference method for solving NFP with one and two dimensions [11]. In order to improve the efficiency and the reliability of the algorithm proposed in [11], Yueh/Lin/Yau proposed a numerical scheme based on the quasi-implicit Euler method (QIEM) [24] for solving the Kolmogorov equations. Under some mild conditions, QIEM has been proved to be convergent and to preserve the conditional probability density functions [24]. However, the efficiency of QIEM can be further improved when we consider the higher-dimensional NFP. In this paper, we develop an efficient algorithm of Yau-Yau method for solving high-dimensional NFP by applying QIEM [24] with discrete sine transform (DST) [7, Chapter 4.8] and present the numerical experiment on up to six-dimensional NFPs. Furthermore, DST is directly applicable on the higher-order difference schemes, so that the number of spatial points can be substantially reduced. This indicates that we can further save more computational costs and retain the same accuracy.

In the following, we consider the classical discrete-time NFP for an $\mathbb{R}^D$-valued signal-observation model [2, 8]. The $\mathbb{R}^D$-valued signal/state process $\mathbf{x} = \mathbf{x}(t)$ satisfies the equation

$$(1) \qquad\qquad \mathrm{d}\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t))\,\mathrm{d}t + \mathrm{d}\mathbf{v}(t),$$

where $\mathbf{f}(\mathbf{x}(t))$ is an $\mathbb{R}^D$-valued drift term, $\mathbf{v}$ is an $\mathbb{R}^D$-valued standard Wiener/Brownian process. In addition, the $\mathbb{R}^M$-valued observation/measurement process $\mathbf{y} = \mathbf{y}(t)$ satisfies the equation

$$(2) \qquad\qquad \mathrm{d}\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t))\,\mathrm{d}t + \mathrm{d}\mathbf{w}(t),$$

where $\mathbf{h}(\mathbf{x}(t))$ is an $\mathbb{R}^M$-valued observation function, $\mathbf{w}$ is an $\mathbb{R}^M$-valued standard Wiener/Brownian process, which is mutually independent of $\mathbf{v}$. To solve the NFP, specifically, we suppose that the value of measurement $\mathbf{y}$ at time $t = \tau_0, \ldots, \tau_{N_\tau}$ can be observed. Then, by applying the theory of Yau-Yau method [20, 22, 23], the NFP can be reduced into a Kolmogorov partial differential equation in $\mathbb{R}^D$ of the form

$$(3) \qquad \frac{\partial u}{\partial t}(t, \mathbf{s}) = \frac{1}{2}\triangle u(t, \mathbf{s}) + \mathbf{p}(\mathbf{s}) \cdot \nabla u(t, \mathbf{s}) + q(\mathbf{s})u(t, \mathbf{s}),$$

where $\mathbf{p}(\mathbf{s}) = -\mathbf{f}(\mathbf{s})$, $q(\mathbf{s}) = -\left(\nabla \cdot \mathbf{f}(\mathbf{s}) + \frac{1}{2} \|\mathbf{h}(\mathbf{s})\|_2^2\right)$, u vanishes at infinity, and the initial $u(0, \mathbf{s}) = \sigma_0(\mathbf{s})$ is a given probability density function over $\mathbb{R}^D$. Once a new measurement $\mathbf{y}(\tau_k)$ is observed, we correct our solution $u(\tau_k, \mathbf{s})$ at time $t = \tau_k$ by

$$\exp\{[\mathbf{y}(\tau_k) - \mathbf{y}(\tau_{k-1})] \cdot \mathbf{h}(\mathbf{s})\} \, u(\tau_k, \mathbf{s}).$$

In the whole process, we use the normalized vector $\widetilde{u}(t, \mathbf{s})$ of the solution $u(t, \mathbf{s})$ representing the probability density function for the state $\mathbf{x}$ at time t. In other words, suppose the random variable X_t with respect to the state at time t is defined by $X_t(\mathbf{s}) = \mathbf{s}$. As a consequence of the Yau-Yau filter theory, we have the probability $P[X_t = \mathbf{s}] = \widetilde{u}(t, \mathbf{s})$. Hence we compute the expectation $E(X_t)$ as an estimate of the state $\mathbf{x}(t)$.

2. Numerical algorithms for solving Kolmogorov equations

In this section, we briefly review the QIEM for solving the Kolmogorov equation (3). Then, the state $\mathbf{x}(t)$ in (1) can be estimated by the solution of (3) at each time t. Let a terminal time T be given. We discretize the time interval $[0, T]$ by taking a uniform partition

$$(4) \qquad \mathcal{P}_{[0,T]} = \{0 = \tau_0 < \tau_1 < \cdots < \tau_{N_\tau} = T\},$$

where $\tau_k - \tau_{k-1} = \Delta\tau$, and assume that the information of $\mathbf{y}$ at each time step τ_k can be observed, for $k = 1, \ldots, N_\tau$. Furthermore, for each time interval $[\tau_{k-1}, \tau_k]$, $k = 1, \ldots, N_\tau$, we partition it uniformly as

$$\mathcal{P}_{[\tau_{k-1}, \tau_k]} = \left\{\tau_{k-1} = t_0^{(k)} < t_1^{(k)} < \cdots < t_{N_t}^{(k)} = \tau_k\right\},$$

where $t_n^{(k)} - t_{n-1}^{(k)} = \Delta t$, $n = 1, \ldots, N_t$. Then the partition

$$\begin{aligned} \mathcal{P}_{[0,T]}^* &= \bigcup_{k=1}^{N_\tau} \mathcal{P}_{[\tau_{k-1}, \tau_k]} \\ &= \Big\{0 = \tau_0 = t_0^{(1)} < \cdots < t_{N_t}^{(1)} = \tau_1 = t_0^{(2)} < \cdots \\ &\qquad < t_{N_t}^{(2)} = \tau_2 = t_0^{(3)} < \cdots < t_{N_t}^{(N_\tau)} = \tau_{N_\tau} = T\Big\} \end{aligned}$$

of $[0, T]$ forms a refinement of the partition $\mathcal{P}_{[0,T]}$ in (4). Our goal is to compute the estimate of the state at each time step $t \in \mathcal{P}_{[0,T]}^*$.

For the sake of computations, we restrict the domain $\mathbb{R}^D$ of the Kolmogorov equation (3) to a D-cell $\Omega \equiv [-R, R]^D$, where R is a suitably large number so that the Gaussian distribution can be ignored outside the ball of radius R with center zero. Then we partition the considered space region on a coordinate axis $[-R, R]$ uniformly by

$$\mathcal{P}_{[-R,R]} = \{-R = s_1 < s_2 < \cdots < s_N = R\},$$

where $s_j - s_{j-1} = \Delta s, j = 1, 2, \ldots, N$. Then the D-cell Ω can be represented by the power set of the partition $\mathcal{P}_{[-R,R]}$,

$$\mathcal{P}_{[-R,R]}^D = \left\{ \mathbf{s}_j = \left(s_j^{(1)}, s_j^{(2)}, \ldots, s_j^{(D)} \right)^\top \,\middle|\, s_j^{(d)} \in \mathcal{P}_{[-R,R]}, \ d = 1, \ldots, D \right\}_{j=1}^{N^D}.$$

2.1. Standard stencil of QIEM

In the discrete model of the Equation (3), we set up the boundary condition of the domain Ω satisfying the Dirichlet boundary condition. That is, $u(t, \mathbf{s}) = 0$ on $[0, T] \times \partial\Omega$. For the d-th dimension of the space, $d = 1, \ldots, D$, by using the QIEM, the second order differential operator $\frac{\partial^2 u}{\partial s^2}(t_n^{(k)}, s_j)$ in (3) is approximated by the second-order discretization

$$(5) \qquad \frac{\partial^2 u}{\partial s^2}(t_n^{(k)}, s_j) \approx \frac{u(t_{n+1}^{(k)}, s_{j+1}) - 2u(t_{n+1}^{(k)}, s_j) + u(t_{n+1}^{(k)}, s_{j-1})}{(\Delta s)^2},$$

which is an implicit form. The first order differential operator $\frac{\partial u}{\partial s}(t_n^{(k)}, s_j)$ in (3) is approximated explicitly by the second-order discretization

$$(6) \qquad \frac{\partial u}{\partial s}(t_n^{(k)}, s_j) \approx \frac{u(t_n^{(k)}, s_{j+1}) - u(t_n^{(k)}, s_{j-1})}{2(\Delta s)}.$$

We define the matrices

$$(7) \quad \mathrm{L}_N = \begin{bmatrix} -2 & 1 & & \\ 1 & -2 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix} \quad \text{and} \quad \mathrm{K}_N = \begin{bmatrix} 0 & 1 & & \\ -1 & 0 & \ddots & \\ & \ddots & \ddots & 1 \\ & & -1 & 0 \end{bmatrix}.$$

Then, the discretized Laplacian operator $\triangle$ in (3) can be represented by the matrix $\frac{1}{(\Delta s)^2}\mathbf{L}_N^{(D)}$, where [7, Chapter 4.8]

$$(8) \qquad \mathbf{L}_N^{(D)} = \sum_{d=1}^{D} \left[\left(\bigotimes_{k=1}^{D-d} \mathbf{I}_N \right) \otimes \mathbf{L}_N \otimes \left(\bigotimes_{k=D-d+2}^{D} \mathbf{I}_N \right) \right],$$

in which $\otimes$ denotes the Kronecker tensor product, $\mathbf{I}_N$ is the identity matrix of size N and $\mathbf{L}_N$ is given in (7). Similarly, the discretization of the term $\mathbf{p} \cdot \nabla$ in (3) can be represented by the matrix $\frac{1}{2(\Delta s)}\mathbf{K}_N^{(D)}$, where [7, Chapter 4.8]

$$(9) \qquad \mathbf{K}_N^{(D)} = \sum_{d=1}^{D} \left\{ \mathbf{P}_d \left[\left(\bigotimes_{k=1}^{D-d} \mathbf{I}_N \right) \otimes \mathbf{K}_N \otimes \left(\bigotimes_{k=D-d+2}^{D} \mathbf{I}_N \right) \right] \right\},$$

in which the matrix $\mathbf{P}_d = \mathrm{diag}\,\{p_d(\mathbf{s}_j)\}_{j=1}^{N^D}$, $d = 1, \ldots, D$ and $\mathbf{K}_N$ is given in (7). For each time step $t_n^{(k)} \in [\tau_{k-1}, \tau_k]$, the partial differential operator $\frac{\partial u}{\partial t}(t_n^{(k)}, \mathbf{s})$ of (3) with respect to time is approximated explicitly by the first-order discretization

$$(10) \qquad \frac{\partial u}{\partial t}(t_n^{(k)}, \mathbf{s}) \approx \frac{u(t_{n+1}^{(k)}, \mathbf{s}) - u(t_n^{(k)}, \mathbf{s})}{\Delta t}.$$

Then, QIEM for solving (3) can be written in the form

$$(11) \qquad \frac{u(t_{n+1}^{(k)}, \mathbf{s}) - u(t_n^{(k)}, \mathbf{s})}{\Delta t}$$
$$= \frac{1}{2(\Delta s)^2}\mathbf{L}_N^{(D)} u(t_{n+1}^{(k)}, \mathbf{s}) + \left(\frac{1}{2(\Delta s)}\mathbf{K}_N^{(D)} + \mathbf{Q}_N^{(D)} \right) u(t_n^{(k)}, \mathbf{s}),$$

where the matrix $\mathbf{Q}_N^{(D)} = \mathrm{diag}\{q(\mathbf{s}_j)\}_{j=1}^{N^D}$. Therefore, we can obtain the solution of (3) at each time step $t_{n+1}^{(k)} \in [\tau_{k-1}, \tau_k]$, $n = 0, \ldots, N_t - 1$, by solving the linear system iteratively in time

$$(12) \qquad \left[\mathbf{I}_{N^D} - \frac{\Delta t}{2(\Delta s)^2}\mathbf{L}_N^{(D)} \right] u(t_{n+1}^{(k)}, \mathbf{s})$$
$$= \left[\mathbf{I}_{N^D} + (\Delta t) \left(\frac{1}{2(\Delta s)}\mathbf{K}_N^{(D)} + \mathbf{Q}_N^{(D)} \right) \right] u(t_n^{(k)}, \mathbf{s}),$$

with the initial vector

$$(13) \qquad u(t_0^{(k)}, \mathbf{s}) = \exp\left\{ [\mathbf{y}(\tau_{k+1}) - \mathbf{y}(\tau_k)] \cdot \mathbf{h}(\mathbf{s}) \right\} u(t_{N_t}^{(k-1)}, \mathbf{s}),$$

for $k = 1, \ldots, N_\tau$. Each vector $u(t_n^{(k)}, \mathbf{s})$ in (12) should be normalized to $\widetilde{u}(t_n^{(k)}, \mathbf{s})$ such that

$$\sum_{j=1}^{N^D} \widetilde{u}(t_n^{(k)}, s_j) = 1.$$

Then the normalized vector $\widetilde{u}(t_n^{(k)}, \mathbf{s})$ represents the probability distribution of the state at time $t_n^{(k)}$. The QIEM (12) for solving the NFP is proven in [24] to be consistent and stable as long as both Δs and Δt are sufficiently small.

2.2. Compact stencils of Laplacian operator

The accuracy of the scheme (12) is $O\left((\Delta s)^2 + \Delta t\right)$. In order to obtain a higher accuracy, we can further consider the compact stencils for Laplacian operator. Suppose the spatial step size of each axis are the same as Δs. The compact difference scheme [6, 18] for the D-dimensional discretized Laplacian operator $\triangle^{(D)}$ can be written by

$$(14) \qquad \triangle^{(D)} u(\mathbf{s}) = \frac{1}{(\Delta s)^2} \sum_{\substack{j=0 \\ |\mathbf{k}| = \sqrt{j}\Delta s \\ \mathbf{k} \in \mathcal{Q}_\mathbf{s}}}^{D} \alpha_j u(\mathbf{s} + \mathbf{k}),$$

where $\mathbf{s}, \mathbf{k} \in \mathbb{R}^D$, $\mathcal{Q}_\mathbf{s}$ is the cube with the center at $\mathbf{s}$ and the side length of $2\Delta s$, α_j are the suitable constant coefficients, $j = 0, \ldots, D$. The coefficients α_j can be taken by $\alpha_j = 2^{2-D-j}$, $j = 1, \ldots, D$ with

$$\alpha_0 = -\sum_{j=1}^{D} \left(\alpha_j \# \left\{ \mathbf{k} \in \mathcal{Q}_\mathbf{s} \, \middle| \, |\mathbf{k}| = \sqrt{j}\Delta s \right\} \right).$$

(See [6, 18].) Then the matrix representation of the compact scheme (14) can be written by the matrix $\frac{1}{(\Delta s)^2} \overline{\mathbf{L}}_N^{(D)}$, where

$$(15) \qquad \overline{\mathbf{L}}_N^{(D)} = \mathrm{S}_N^{(1,\ldots,D)},$$

where the matrix $\mathrm{S}_N^{(1,\ldots,D)}$ is recursively defined by

$$(16) \qquad \mathrm{S}_N^{(i,\ldots,k)} = \mathrm{I}_N \otimes \mathrm{S}_N^{(i,\ldots,k-1)} + \mathrm{J}_N \otimes \mathrm{S}_N^{(i+1,\ldots,k)},$$
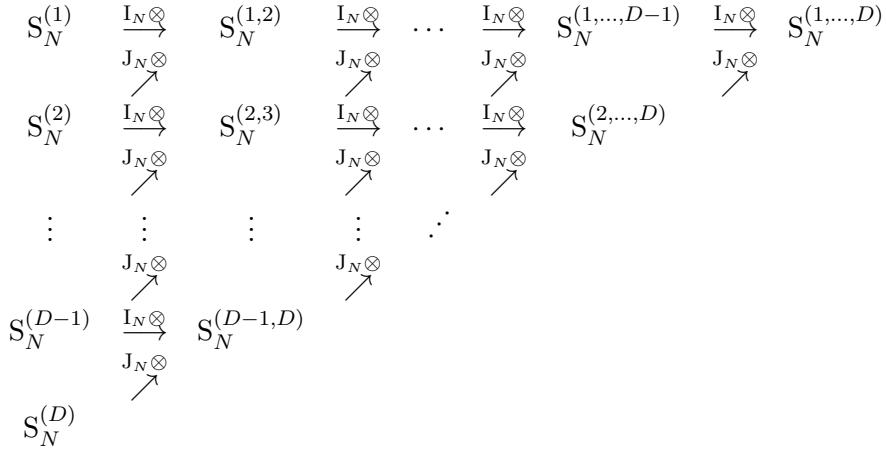
$$S_N^{(1)} \xrightarrow[J_N \otimes]{I_N \otimes} S_N^{(1,2)} \xrightarrow[J_N \otimes]{I_N \otimes} \cdots \xrightarrow[J_N \otimes]{I_N \otimes} S_N^{(1,\ldots,D-1)} \xrightarrow[J_N \otimes]{I_N \otimes} S_N^{(1,\ldots,D)}$$

$$S_N^{(2)} \xrightarrow[J_N \otimes]{I_N \otimes} S_N^{(2,3)} \xrightarrow[J_N \otimes]{I_N \otimes} \cdots \xrightarrow[J_N \otimes]{I_N \otimes} S_N^{(2,\ldots,D)}$$

$$\vdots \quad \vdots \quad \vdots \quad \vdots \quad \ddots$$

$$S_N^{(D-1)} \xrightarrow[J_N \otimes]{I_N \otimes} S_N^{(D-1,D)}$$

$$S_N^{(D)}$$

Figure 1: Diagram for construction of the D-dimensional compact Laplacian matrix.

for $k = i+1, \ldots, D$, in which the matrices J_N and $S_N^{(j)}$ are defined by

$$
J_N = \begin{bmatrix} 0 & 1 & & \\ 1 & 0 & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 0 \end{bmatrix} \quad \text{and} \quad S_N^{(j)} = \begin{bmatrix} \alpha_{j-1} & \alpha_j & & \\ \alpha_j & \alpha_{j-1} & \ddots & \\ & \ddots & \ddots & \alpha_j \\ & & \alpha_j & \alpha_{j-1} \end{bmatrix},
$$

respectively, $j = 1, \ldots, D$. A diagram regarding the construction of a compact Laplacian matrix is shown in Figure 1. Similar to (12), the linear system of the compact scheme can be formulated by

$$
(17) \qquad \left[\mathbf{I}_{N^D} - \frac{\Delta t}{2(\Delta s)^2} \overline{\mathbf{L}}_N^{(D)} \right] u(t_{n+1}^{(k)}, \mathbf{s})
$$
$$
= \left[\mathbf{I}_{N^D} + (\Delta t) \left(\frac{1}{2(\Delta s)} \mathbf{K}_N^{(D)} + \mathbf{Q}_N^{(D)} \right) \right] u(t_n^{(k)}, \mathbf{s}).
$$

In addition, according to the recursive construction of the matrix $S_N^{(1,\ldots,D)}$, we can write down an explicit form of $\overline{\mathbf{L}}_N^{(D)}$, which is shown in the following theorem.

Theorem 1 (Explicit Form of Compact Laplacian Matrices). *The matrix* $\overline{\mathbf{L}}_N^{(D)}$ *in (15) can be written explicitly by*

$$(18) \qquad \overline{\mathbf{L}}_N^{(D)} = \sum_{d=1}^{D} \left[\sum_{\substack{\#\{i|X_i=I_N\}=D-d \\ \#\{i|X_i=J_N\}=d-1}} \left(\bigotimes_{\substack{i=1 \\ X_i \in \{I_N, J_N\}}}^{D-1} X_i \right) \otimes S_N^{(d)} \right].$$

Proof. We observe that each arrow in the diagram of Figure 1 is performed exactly once in the construction of the compact discrete Laplacian matrix $\overline{\mathbf{L}}_N^{(D)}$. In Figure 1, we take $S_N^{(d)}$ as the starting point and go to the end point $S_N^{(1,\dots,D)}$ along the arrows. Each path from $S_N^{(d)}$ to $S_N^{(1,\dots,D)}$ determines a term of $\overline{\mathbf{L}}_N^{(D)}$. Note that each path passes through the arrow "$\xrightarrow{I_N \otimes}$" and the arrow "$\overset{J_N \otimes}{\nearrow}$" for exactly $D-d$ times and $d-1$ times, respectively. Hence, all terms related to $S_N^{(d)}$ are given by the sum of the Kronecker products for all possible permutations of the matrix sequence $(\underbrace{I_N, \dots, I_N}_{D-d}, \underbrace{J_N, \dots, J_N}_{d-1})$, that is,

$$\sum_{\substack{\#\{i|X_i=I_N\}=D-d \\ \#\{i|X_i=J_N\}=d-1}} \left(\bigotimes_{\substack{i=1 \\ X_i \in \{I_N, J_N\}}}^{D-1} X_i \right) \otimes S_N^{(d)}.$$

Therefore, (18) follows. $\qquad \square$

3. Nonnegativity of numerical solutions

According to the Yau-Yau filter algorithm, the solution $u(t_n^{(k)}, \mathbf{s})$ of the linear system (12) or (17) represents the probability density function of the state $\mathbf{x}(t_n^{(k)})$ over Ω. Therefore, it is essential to guarantee the nonnegativity of each numerical solution, so that the probability density function at each time makes sense. In the following, we prove that the numerical solution of the Kolmogorov equation by using QIEM is nonnegative. That is, we prove the matrix $\left[\mathbf{I}_{N^D} - \frac{\Delta t}{2(\Delta s)^2} \overline{\mathbf{L}}_N^{(D)} \right]^{-1}$ is a nonnegative operator, so that the solution of the linear system (17) is always nonnegative.

Definition 1. A real matrix $B = [B_{ij}]$ is called an *M*-matrix if $B_{ij} \leq 0$, $i \neq j$ and B^{-1} exists with $B^{-1} \geq 0$.

Lemma 1 (Equivalence Condition of M-matrix). *[14] Let B be a real matrix with $B_{ij} \leq 0$ for $i \neq j$. Then B is an M-matrix if and only if there is a positive vector $v > 0$ such that $Bv > 0$.*

Theorem 2 (Nonnegativity for Compact Scheme). *For each pair of natural numbers N and D, the matrix $\left[\mathbf{I}_{N^D} - \frac{\Delta t}{2(\Delta s)^2} \overline{\mathbf{L}}_N^{(D)} \right]$ in (17) is an M-matrix. In other words, $\left[\mathbf{I}_{N^D} - \frac{\Delta t}{2(\Delta s)^2} \overline{\mathbf{L}}_N^{(D)} \right]^{-1}$ is a nonnegative matrix.*

Proof. Let N and D be arbitrarily given two natural numbers, and

$$\mathbf{A} = \left[\mathbf{I}_{N^D} - \frac{\Delta t}{2(\Delta s)^2} \overline{\mathbf{L}}_N^{(D)} \right].$$

First, we claim that $\mathbf{A}_{ij} \leq 0$ for $i \neq j$. That is to show that $\left[\overline{\mathbf{L}}_N^{(D)} \right]_{ij} \geq 0$ for $i \neq j$. Note that $\alpha_j \geq 0$ for $j = 1, \ldots, D$. By definition, the entries of the matrix $\mathrm{S}_N^{(d)}$ are nonnegative, for $d = 2, \ldots, D$. From (18), we have the equation

$$(19) \quad \overline{\mathbf{L}}_N^{(D)} = \mathrm{I}_{N^{D-1}} \otimes \mathrm{S}_N^{(1)} + \sum_{d=2}^{D} \left[\sum_{\substack{\#\{i|\mathrm{X}_i=\mathrm{I}_N\}=D-d \\ \#\{i|\mathrm{X}_i=\mathrm{J}_N\}=d-1}} \left(\bigotimes_{\substack{i=1 \\ \mathrm{X}_i \in \{\mathrm{I}_N, \mathrm{J}_N\}}}^{D-1} \mathrm{X}_i \right) \otimes \mathrm{S}_N^{(d)} \right],$$

where the entries of $\mathrm{S}_N^{(d)}$ are nonnegative for $d = 2, \ldots, D$. In addition, the matrix $\mathrm{S}_N^{(1)} = \alpha_0 \mathrm{I}_N + \alpha_1 \mathrm{J}_N$ where $\alpha_1 \geq 0$. From the Equation (19), the only negative coefficient α_0 will appear merely on the diagonal entries of $\overline{\mathbf{L}}_N^{(D)}$. Now, we claim that $\mathbf{A} \, \mathbf{1}_{N^D} > 0$, where $\mathbf{1}_{N^D} = (\underbrace{1, 1, \ldots, 1}_{N^D})^{\top}$. According-ing to the definition of α_0, the row sum of the compact Laplacian matrix $\overline{\mathbf{L}}_N^{(D)} \mathbf{1}_{N^D} \leq 0$. Hence $\mathbf{A} \, \mathbf{1}_{N^D} = \mathbf{1}_{N^D} - \frac{\Delta t}{2(\Delta s)^2} \overline{\mathbf{L}}_N^{(D)} \mathbf{1}_{N^D} > 0$. Therefore, by Lemma 1, $\mathbf{A}$ is an M-matrix. $\qquad \square$

Corollary 1 (Nonnegativity for Standard Scheme). *For each pair of natural numbers N and D, the matrix $\left[\mathbf{I}_{N^D} - \frac{\Delta t}{2(\Delta s)^2} \mathbf{L}_N^{(D)} \right]$ in (12) is an M-matrix. In other words, $\left[\mathbf{I}_{N^D} - \frac{\Delta t}{2(\Delta s)^2} \mathbf{L}_N^{(D)} \right]^{-1}$ is a nonnegative operator.*

Proof. The statement follows directly by picking $\alpha_0 = -2D$, $\alpha_1 = 1$ and $\alpha_j = 0$, for $j = 2, \ldots, D$, as in Theorem 2. $\qquad \square$
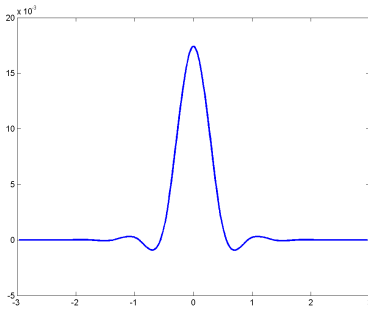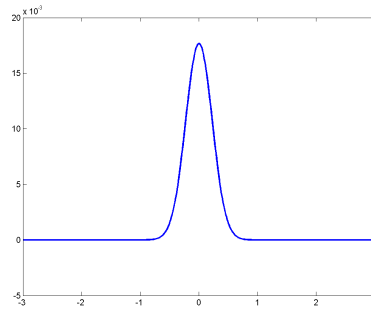
(a) Chebyshev Spectral Method   (b) Finite Difference Method

Figure 2: The numerical solution of the Kolmogorov equation

It is natural to consider applying the spectral method with Chebyshev polynomial basis [19] for solving (3) by replacing the Laplacian matrix $\mathbf{L}_N^{(D)}$ in (8) by

$$(20) \qquad \mathbb{L}_N^{(D)} = \sum_{d=1}^{D} \left[\left(\bigotimes_{k=1}^{D-d} I_N \right) \otimes (D_N)^2 \otimes \left(\bigotimes_{k=D-d+2}^{D} I_N \right) \right],$$

and the matrix $\mathbf{K}_N^{(D)}$ in (9) by

$$(21) \qquad \mathbb{K}_N^{(D)} = \sum_{d=1}^{D} \left\{ \mathbf{P}_d \left[\left(\bigotimes_{k=1}^{D-d} I_N \right) \otimes D_N \otimes \left(\bigotimes_{k=D-d+2}^{D} I_N \right) \right] \right\},$$

where the Chebyshev differentiation matrix D_N is defined in page 53 of [19]. In general, the advantage of the spectral method is to handle a small-sized discretization model. However, the probability density function of the Kolmogorov equation in each step computed by (20) and (21) may fail to be nonnegative. Thus, by using the spectral method, the estimates cannot match the real states well because the expectations cannot be precisely estimated. In Figure 2, we observe that the probability density function of the Kolmogorov equation computed by Chebyshev spectral method not all are nonnegative. But the probability density function computed by the finite difference method can be guaranteed to be nonnegative (see Theorem 2). Therefore, throughout this paper, we are motivated to consider the QIEM (12) only.

4. High-dimensional DST for accelerating QIEM

In order to further improve the efficiency of QIEM, we apply the high-dimensional discrete sine transformation (DST) for solving the Kolmogorov equations.

4.1. Discrete sine transform

It is well known that the DST can be used for solving Poisson equations with the Dirichlet boundary condition [5]. Suppose the number of discretized spatial points is N. The one-dimensional discretized Laplacian operator can be represented by the matrix L_N in (7). The DST matrix regarding L_N is defined by [7, Chapter 4.8]

$$(22) \qquad \mathrm{W}_N = \left[\sqrt{\frac{2}{N+1}} \sin\left(\frac{ij\pi}{N+1} \right) \right]_{i,j=1}^{N},$$

so that the spectral decomposition of the matrix L_N is given by [7, Chapter 4.8]

$$(23) \qquad \mathrm{L}_N = \mathrm{W}_N \Lambda_N \mathrm{W}_N^*,$$

where the diagonal matrix Λ_N is of the form [7, Chapter 4.8]

$$(24) \qquad \Lambda_N = \mathrm{diag}\left(-4\sin^2\left(\frac{i\pi}{2(N+1)} \right) \right).$$

Then, we can express the solution of the linear system $\mathrm{L}_N u = b$ as

$$u = \left(\mathrm{W}_N^* \left(\Lambda_N^{-1} \left(\mathrm{W}_N b \right) \right) \right).$$

Therefore, solving u can be reduced to matrix-vector products of W_N and W_N^*, respectively, and the inverse of the diagonal matrix Λ_N^{-1}. Furthermore, the matrix-vector products of W_N and W_N^* can be done efficiently by calling the built-in function directly.

For D-dimensional problems, we suppose the number of discretized spatial points in k-th axis is N_k, $k = 1, \ldots, D$. Then, the D-dimensional discretized Laplacian operator can be represented by the matrix [7, Chapter 4.8]

$$(25) \qquad \mathbf{L}_{N_1,\ldots,N_D}^{(D)} = \sum_{d=1}^{D} \left[\left(\bigotimes_{k=1}^{D-d} \mathrm{I}_{N_k} \right) \otimes \mathrm{L}_{N_{D-d+1}} \otimes \left(\bigotimes_{k=D-d+2}^{D} \mathrm{I}_{N_k} \right) \right].$$

For $A^{(j)} \in \mathbb{R}^{p_j \times q_j}$ and $B^{(j)} \in \mathbb{R}^{q_j \times r_j}$, $j = 1, \ldots, d$. It is well-known by the multiplication rule for Kronecker products [3] that

$$
(26) \qquad \left(\bigotimes_{j=1}^{d} A^{(j)} \right) \left(\bigotimes_{j=1}^{d} B^{(j)} \right) = \bigotimes_{j=1}^{d} A^{(j)} B^{(j)}.
$$

Then, from (23)-(25) and (26), we have the following useful theorem immediately.

Theorem 3 (D-dimensional DST for Laplacian). *[7, Chapter 4.8] [4] Let the number of partition points in k-th axis be N_k, $k = 1, \ldots, D$. The spectral decomposition of the D-dimensional Laplacian matrix $\mathbf{L}_{N_1,\ldots,N_D}^{(D)}$ in (25) can be written by*

$$
(27) \qquad \mathbf{L}_{N_1,\ldots,N_D}^{(D)} = \mathbf{W}_{N_1,\ldots,N_D}^{(D)} \mathbf{\Lambda}_{N_1,\ldots,N_D}^{(D)} \mathbf{W}_{N_1,\ldots,N_D}^{(D)*},
$$

where $\mathbf{W}_{N_1,\ldots,N_D}^{(D)} = \left(\bigotimes_{k=1}^{D} W_{N_k} \right)$, $\mathbf{W}_{N_1,\ldots,N_D}^{(D)*} = \left(\bigotimes_{k=1}^{D} W_{N_k}^* \right)$ and $\mathbf{\Lambda}_{N_1,\ldots,N_D}^{(D)}$ is the diagonal matrix of the form

$$
\mathbf{\Lambda}_{N_1,\ldots,N_D}^{(D)} = \left[\sum_{d=1}^{D} \left(\bigotimes_{k=1}^{D-d} I_{N_k} \right) \otimes \Lambda_{N_{D-d+1}} \otimes \left(\bigotimes_{k=D-d+2}^{D} I_{N_k} \right) \right].
$$

4.2. The DST on compact stencils

In order to reduce the number of spatial points for discretization and retain the same accuracy, we apply the DST on compact QIEM stencils. By doing so, we can further save the computational costs.

In the following, we simply derive an explicit form of the spectrum of the compact Laplacian matrix $\overline{\mathbf{L}}_N^{(D)}$, so that the DST can be applied in the compact stencils directly. For convenience, we suppose the numbers of partition points in each axis are the same.

Theorem 4 (D-Dimensional DST for Compact Laplacian Stencils). *Let the number of partition points in each axis be N. The spectral decomposition of $\overline{\mathbf{L}}_N^{(D)}$ in (15) can be written by*

$$
(28) \qquad \overline{\mathbf{L}}_N^{(D)} = \mathbf{W}_N^{(D)} \overline{\mathbf{\Lambda}}_N^{(D)} \mathbf{W}_N^{(D)*},
$$

where $\mathbf{W}_N^{(D)} = \left(\bigotimes_{d=1}^{D} W_N\right)$, $\mathbf{W}_N^{(D)*} = \left(\bigotimes_{d=1}^{D} W_N^*\right)$ *and* $\overline{\mathbf{\Lambda}}_N^{(D)}$ *is the diagonal matrix of the form*

$$(29) \qquad \overline{\mathbf{\Lambda}}_N^{(D)} = \sum_{d=1}^{D} \left[\sum_{\substack{\#\{i|X_i=I_N\}=D-d \\ \#\{i|X_i=\Gamma_N\}=d-1}} \left(\bigotimes_{\substack{i=1 \\ X_i \in \{I_N, \Gamma_N\}}}^{D-1} X_i \right) \otimes \Theta_N^{(d)} \right],$$

where $\Theta_N^{(d)} = \alpha_d \Lambda_N + (2\alpha_d + \alpha_{d-1})\, I_N$, $d = 1, \ldots, D$ *and* $\Gamma_N = \Lambda_N + 2\, I_N$.

Proof. Note that

$$(30) \qquad S_N^{(d)} = \alpha_d L_N + (2\alpha_d + \alpha_{d-1})\, I_N$$
$$= W_N \left[\alpha_d \Lambda_N + (2\alpha_d + \alpha_{d-1})\, I_N\right] W_N^* = W_N \Theta_N^{(d)} W_N^*,$$

$d = 1, \ldots, D$, and

$$(31) \qquad J_N = L_N + 2\, I_N = W_N \left(\Lambda_N + 2\, I_N\right) W_N^* = W_N \Gamma_N W_N^*.$$

Plugging (30) and (31) into (18), we obtain

$$(32) \qquad \overline{\mathbf{L}}_N^{(D)} = \sum_{d=1}^{D} \sum_{\substack{\#\{i|X_i=W_N I_N W_N^*\}=D-d \\ \#\{i|X_i=W_N \Gamma_N W_N^*\}=d-1}} \left(\bigotimes_{\substack{i=1 \\ X_i \in \{W_N I_N W_N^*, W_N \Gamma_N W_N^*\}}}^{D-1} X_i \right)$$
$$\otimes \left(W_N \Theta_N^{(d)} W_N^* \right).$$

By applying (26) to (32), the theorem is proved. $\qquad\square$

Remark. (i) Theorem 4 implies that the matrices $\mathbf{L}_N^{(D)}$ and $\overline{\mathbf{L}}_N^{(D)}$ are simultaneously diagonalizable since $\mathbf{L}_N^{(D)} = \mathbf{W}_N^{(D)} \mathbf{\Lambda}_N^{(D)} \mathbf{L}_N^{(D)*}$. (ii) In (29), since $\overline{\mathbf{\Lambda}}_N^{(D)}$ is diagonal, the inverse of $\overline{\mathbf{\Lambda}}_N^{(D)}$ is easily to compute.

5. Numerical experiments

In this section, we demonstrate numerical experiments for the cubic sensor problem of dimension D, for $D = 1, \ldots, 6$, and test the efficiency of our algorithm for solving NFP. The process for generating data of states and observations for the cubic sensor problem is described in Section 5.1. The numerical results are shown in Section 5.2.

5.1. Generating data of the states and observations

In order to simulate numerical examples of (1) and (2), we use the Euler forward difference method to generate a set of states and observations $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=0}^{N_\tau}$ of the form

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{f}(\mathbf{x}_k)\Delta\tau + \mathbf{v}\sqrt{\Delta\tau}, \\[2mm] \mathbf{y}_{k+1} = \mathbf{y}_k + \mathbf{h}(\mathbf{x}_k)\Delta\tau + \mathbf{w}\sqrt{\Delta\tau}, \end{cases}$$

where $\mathbf{x}_0$ is a given initial vector and $\mathbf{y}_0$ is zero, $\Delta\tau$ is the size of the time step, $\mathbf{v}$ and $\mathbf{w}$ are mutually independent Wiener/Brownian processes with mean to be zero and variance to be one. The vector-valued functions $\mathbf{f}$ and $\mathbf{h}$ in the cubic sensor problem of (1) and (2) are chosen to be

$$\mathbf{f}(x_1, \ldots, x_D) = (\cos(x_1), \ldots, \cos(x_D)) \quad \text{and} \quad \mathbf{h}(x_1, \ldots, x_D) = \left(x_1^3, \ldots, x_D^3\right).$$

5.2. Numerical results

We use the laptop of Sony VIAO (SVS13137PW) with 12GB memory as our computing device. In order to check the accuracy of our algorithm for solving NFP, we compute both the mean error (L^1-error) E_{L^1} and the root-mean-square error (L^2-error) E_{L^2} between the estimates $\widehat{\mathbf{x}} = (\widehat{x}_1, \ldots, \widehat{x}_D)$ and the real state $\mathbf{x} = (x_1, \ldots, x_D)$ by

$$E_{L^1}(\widehat{\mathbf{x}}, \mathbf{x}) = \frac{\sum_{i=1}^{N_\tau} \sum_{d=1}^{D} |\widehat{x}_d(t_i) - x_d(t_i)|}{DN_\tau}$$

and

$$E_{L^2}(\widehat{\mathbf{x}}, \mathbf{x}) = \sqrt{\frac{\sum_{i=1}^{N_\tau} \sum_{d=1}^{D} |\widehat{x}_d(t_i) - x_d(t_i)|^2}{DN_\tau}},$$

respectively. Table 1 shows the execution time of 20,000 steps, L^1-error and L^2-error for the cubic sensor problem of dimensions $D = 1, \ldots, 6$, respectively. We take the terminal time $T = 20$, the size of time steps for generating data of signals (1) and observations (2) $\Delta\tau = 0.01$, the size of time steps for solving the Kolmogorov equation (3) $\Delta t = 0.001$, the size of spatial steps $\Delta s = 0.5$, and the initial vector $\sigma_0(\mathbf{s}) = \exp\{-10|\mathbf{s}|^2\}$. According to the experimental results, QIEM with DST is capable for solving up to 4-D NFP efficiently. For solving the 5-D and 6-D NFP, QIEM with DST is reasonable fast. The accuracy of QIEM with DST is satisfactory since

Dimension	Execution Time	L^1-Error	L^2-Error
1	3.0929 sec.	0.2641	0.3410
2	4.7645 sec.	0.3647	0.4254
3	23.4864 sec.	0.3538	0.3930
4	80.7798 sec.	0.3643	0.4024
5	105.1858 min.	0.4364	0.4816
6	131.0143 min.	0.4011	0.4205

Table 1: The execution time and RMS error of QIEM with DST

both the L^1-error and L^2-error are less than the spatial mesh size $\Delta s = 0.5$. Figure 3–8 show the numerical results of QIEM with DST for solving D-dimensional cubic sensor problems, $D = 1, \ldots, 6$, respectively. We can see that the overall trend of the estimates in each figure match the real states well.
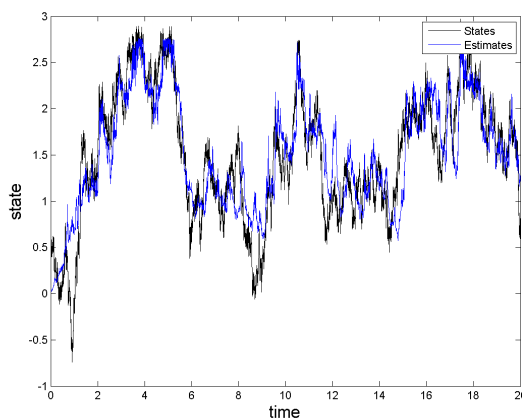

Figure 3: The numerical results of QIEM with DST for 1-D cubic sensor problem

6. Concluding remarks

In this paper, we apply the QIEM with the high-dimensional DST acceleration for solving the high-dimensional NFP. In addition, we derive an explicit form of spectrum of the compact Laplacian matrix, so that the DST is directly applicable on compact difference schemes. Furthermore, we prove that our algorithm preserves the probability density functions. Several numerical experiments are demonstrated in detail. The numerical results indicate that QIEM with the DST acceleration is capable of solving up to six-dimensional NFP efficiently and accurately.
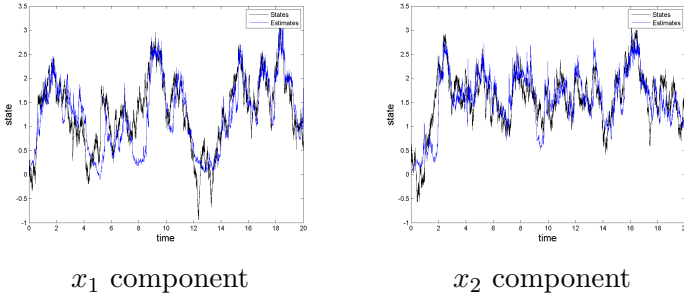
x_1 component x_2 component

Figure 4: The numerical results of QIEM with DST for 2-D cubic sensor problem

x_1 component x_2 component x_3 component
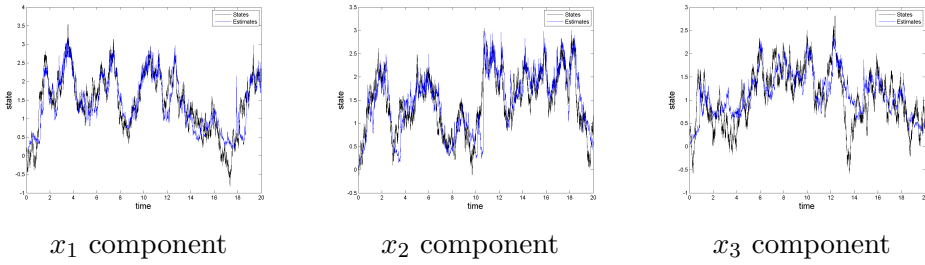
Figure 5: The numerical results of QIEM with DST for 3-D cubic sensor problem

x_1 component x_2 component

x_3 component x_4 component
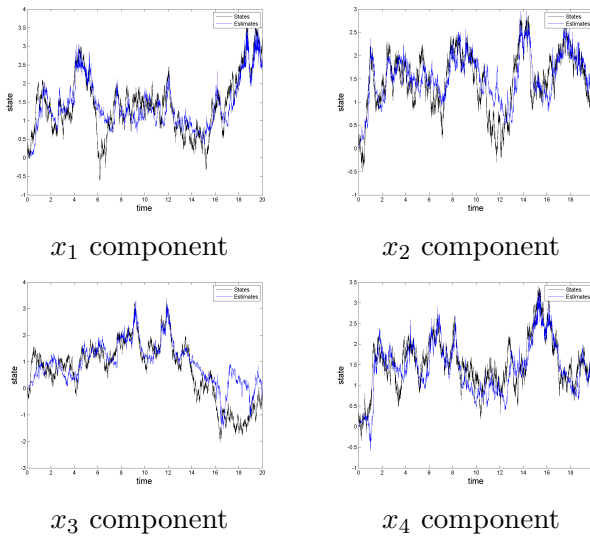
Figure 6: The numerical results of QIEM with DST for 4-D cubic sensor problem

| x_1 component | x_2 component | x_3 component |

| x_4 component | x_5 component |
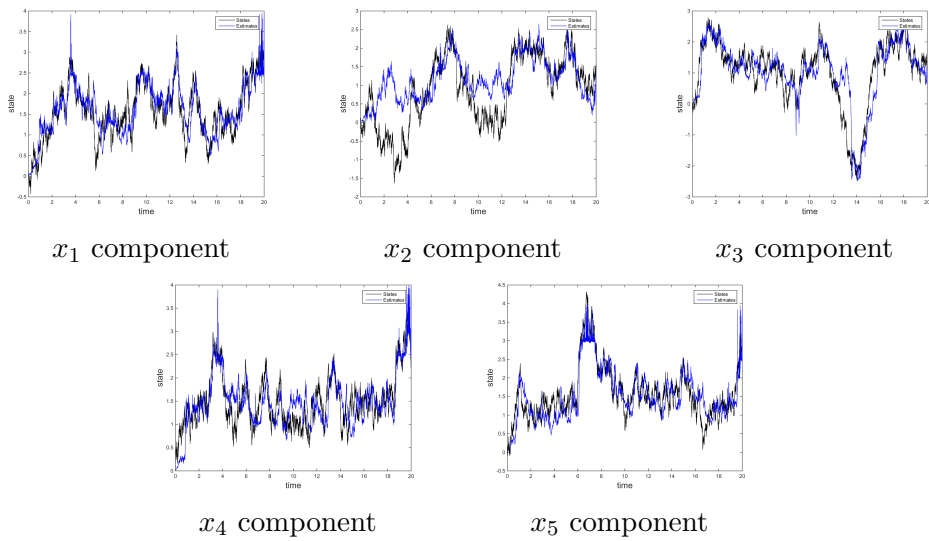
Figure 7: The numerical results of QIEM with DST for 5-D cubic sensor problem

| x_1 component | x_2 component | x_3 component |

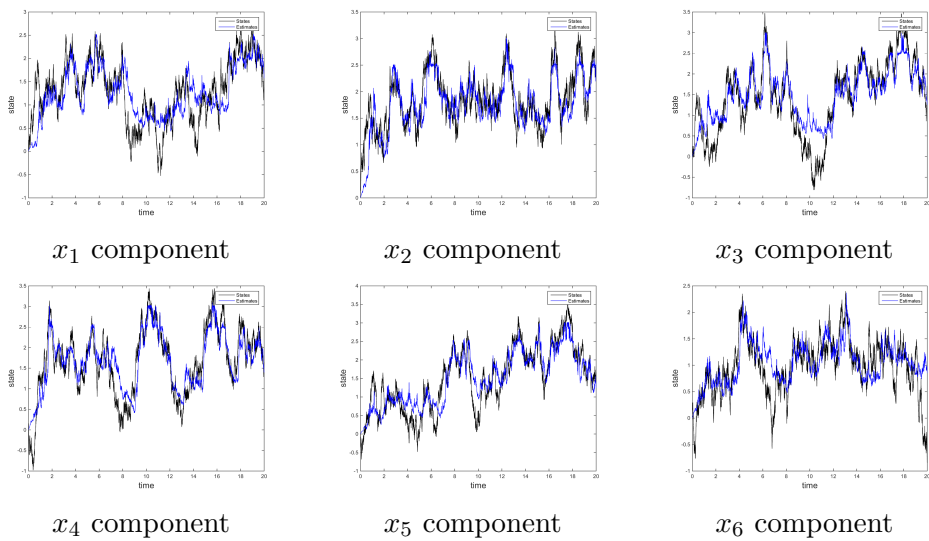
| x_4 component | x_5 component | x_6 component |

Figure 8: The numerical results of QIEM with DST for 6-D cubic sensor problem

References

[1] F. Auger, M. Hilairet, J. M. Guerrero, E. Monmasson, T. Orlowska-Kowalska and S. Katsura, *Industrial applications of the Kalman filter: A review.* IEEE Transactions on Industrial Electronics, **60**(12):5458–5471, 2013.

[2] A. Bain and D. Crisan, *Fundamentals of Stochastic Filtering.* Springer Science Business Media, 1 edition, 2009.

[3] R. Bank, R. L. Graham, W. Hackbusch, J. Stoer, R. S. Varga and H. Yserentant, *Tensor Spaces and Numerical Tensor Calculus.* Springer-Verlag Berlin Heidelberg, 1 edition, 2012.

[4] R. Chan and X. Jin, *A family of block preconditioners for block systems.* SIAM J. Sci. Stat. Comput., 13:1218–1235, 1992.

[5] F. W. Dorr, *The direct solution of the discrete poisson equation on a rectangle.* SIAM review, **12**(2):248–263, 1970.

[6] S. Dürr and G. Koutsou, *Brillouin improvement for wilson fermions.* Phys. Rev. D, **83**:114512, Jun 2011.

[7] G. H. Golub and C. F. Van Loan, *Matrix Computations.* Johns Hopkins University Press, 4 edition, 2012.

[8] A. H. Jazwinski, *Stochastic Processes and Filtering Theory.* Academic Press, 1 edition, 1970.

[9] R. E. Kalman and R. S. Bucy, *New results in linear filtering and prediction theory.* Trans. ASME, **83**:95–108, 1961.

[10] H. J. Kushner and A. S. Budhiraja, *A nonlinear filtering algorithm based on an approximation of the conditional distribution.* IEEE Transaction Automatic Control, **45**:580–585, 1999.

[11] Z. Liu, F. Dong, and L. Ding, *Numerical results of nonlinear filtering problem from Yau-Yau method.* Journal of Computers, **7**(4):971–976, 2012.

[12] X. Luo and S. S.-T. Yau, *Complete real time solution of the general nonlinear filtering problem without memory.* IEEE Trans. Automat. Control, **58**(10):2563–2578, 2013.

[13] X. Luo and S. S.-T. Yau, *Hermite spectral method to 1d forward Kolmogorov equation and its application to nonlinear filtering problems.* IEEE Trans. Automat. Control, **58**(10):2495–2507, 2013.

[14] R. J. Plemmons, *M-matrix characterizations.ixnonsingular m-matrices.* Linear Algebra and its Applications, **18**(2):175–188, 1977.

[15] G. G. Rigatos, *Modelling and Control for Intelligent Industrial Systems.* Springer-Verlag Berlin Heidelberg, 1 edition, 2011.

[16] G. G. Rigatos, *Nonlinear Estimation and Applications to Industrial Systems Control.* Nova Science Publishers Inc, 1 edition, 2013.

[17] G. G. Rigatos and P. Siano, *Sensorless control of electric motors with Kalman filters: applications to robotic and industrial systems.* International Journal of Advanced Robotic Systems, **8**(6):62–80, 2011.

[18] W. F. Spotz and G. F. Carey, *High-order compact finite difference methods.* Preliminary Proceedings International Conference on Spectral and High Order Methods, Houston, TX, pages 397–408, 1995.

[19] Lloyd N. Trefethen, *Spectral Methods in MatLab.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[20] S. T. Yau and S. S.-T. Yau, *Finite dimensional filters with nonlinear drift iii: Duncan-Mortensen-Zakai equation with arbitrary initial condition for kalman-bucy filtering system and benes filtering system.* IEEE Transactions on Aerospace and Electronic Systems, **33**:1277–1294, 1997.

[21] S.-T. Yau and S. S.-T Yau, *Existence and uniqueness and decay estimates for the time dependent parabolic equation with application to Duncan-Mortensen-Zakai equation.* Asian Journal of Mathematics, **2**:1079–1149, 1998.

[22] S.-T. Yau and S. S.-T. Yau, *Real time solution of nonlinear filtering problem without memory I.* Mathematical Research Letters, **7**:671–693, 2000.

[23] S.-T. Yau and S. S.-T. Yau, *Real time solution of nonlinear filtering problem without memory II.* SIAM. J. Control Optim., **47**(1):163–195, 2008.

[24] M.-H. Yueh, W.-W. Lin, and S.-T. Yau, *An efficient algorithm of Yau-Yau method for solving nonlinear filtering problems.* Communications in Information and Systems, **14**(2):111–134, 2014.

DEPARTMENT OF APPLIED MATHEMATICS, NATIONAL CHIAO TUNG UNIVERSITY
HSINCHU 300, TAIWAN
E-mail address: mhyueh@math.nctu.edu.tw

DEPARTMENT OF APPLIED MATHEMATICS, NATIONAL CHIAO TUNG UNIVERSITY
HSINCHU 300, TAIWAN
E-mail address: wwlin@math.nctu.edu.tw

DEPARTMENT OF MATHEMATICS, HARVARD UNIVERSITY
CAMBRIDGE, MA 02138, USA
E-mail address: yau@math.harvard.edu