

# An Efficient Algorithm of Yau-Yau Method for Solving Nonlinear Filtering Problems

MEI-HENG YUEH , WEN-WEI LIN AND SHING-TUNG YAU

It is well known that the nonlinear filter has important applications in military, engineering and commercial industries. In this paper, we propose efficient and accurate numerical algorithms for the realization of the Yau-Yau method for solving nonlinear filtering problems by using finite difference schemes. The Yau-Yau method reduces the nonlinear filtering problem to the initial-value problem of Kolmogorov equations. We first solve this problem by the implicit Euler method, which is stable in most cases, but costly. Then, we propose a quasi-implicit Euler method which is feasible for acceleration by fast Fourier transformations. Furthermore, we propose a superposition technique which enables us to deal with the nonlinear filtering problem in an off-time process and thus, save a large amount of computational cost. Next, we prove that the numerical solutions of Kolmogorov equations by our schemes are always nonnegative in each iteration. Consequently, our iterative process preserves the probability density functions. In addition, we prove convergence of our schemes under some mild conditions. Numerical results show that the proposed algorithms are efficient and promising.

AMS 2000 SUBJECT CLASSIFICATIONS: Primary 93E11, 60G35, 62M20; secondary 65M06, 65M12.

KEYWORDS AND PHRASES: nonlinear filtering, Kolmogorov equations.

## 1. Introduction

The nonlinear filtering problem has a variety of applications in military, engineering and commercial industries [15, 13, 14]. It involves the estimation of a stochastic process of the state  $\mathbf{x} = \mathbf{x}(t)$  that cannot be directly measured during the process. Information containing  $\mathbf{x}$  is then obtained from observation process  $\mathbf{y} = \mathbf{y}(t)$ . The goal of the nonlinear filtering is to determine the conditional probability density function  $\rho(t, \mathbf{x})$  of the state  $\mathbf{x}(t)$  via a given observation history of  $\mathbf{y}(s)$ ,  $s \in [0, t]$ . In 1961, Kalman and Bucy [6] first proposed the finite-dimensional filter, known as Kalman-Bucy filter, for solving the linear filtering problem with Gaussian initial distribution, which is useful in natural science and engineering [1]. Although the Kalman-Bucy

filter is not feasible for the nonlinear filtering problem, it gives great inspiration to the posterity in solving the nonlinear filtering problem. The core issue of the nonlinear filtering problem is to solve the Duncan-Mortensen-Zakai [3, 12, 24] (DMZ) equation in real time. Yau and Yau [21, 22, 23] have proved that the real-time solution of the DMZ equation can be reduced to an off-time solution of the Kolmogorov equation. Luo/Yau [10, 11] have extended the algorithm in [23] to the most general setting of nonlinear filterings. Based on the work of Yau and Yau [22, 23], Liu/Dong/Ding [9] proposed a numerical method to solve the nonlinear filtering problem by using explicit finite difference schemes. In order to improve the reliability of the algorithm proposed in [9], in this paper, we propose an efficient and reliable quasi-implicit numerical scheme for solving the Kolmogorov equations and estimate approximate states of a given signal-observation model.

The nonlinear filtering problem considered here is to determine approximate states for a given observation history of the following signal-observation model [5, 2]

$$(1) \quad \begin{cases} d\mathbf{x}(t) = \mathbf{f}(\mathbf{x}(t)) dt + d\mathbf{v}(t) & \mathbf{x}(0) = \mathbf{x}_0, \\ d\mathbf{y}(t) = \mathbf{h}(\mathbf{x}(t)) dt + d\mathbf{w}(t) & \mathbf{y}(0) = 0, \end{cases}$$

where  $\mathbf{x}(t) = (x_1(t), \dots, x_D(t))^T \in \mathbb{R}^D$  and  $\mathbf{y}(t) = (y_1(t), \dots, y_M(t))^T \in \mathbb{R}^M$  are the state and the measurement/observation vectors at time  $t$ , respectively,  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_D(\mathbf{x}))^T$  and  $\mathbf{h}(\mathbf{x}) = (h_1(\mathbf{x}), \dots, h_M(\mathbf{x}))^T$  are given vector-valued functions,  $\mathbf{v} \in \mathbb{R}^D$  and  $\mathbf{w} \in \mathbb{R}^M$  are mutually independent standard Brownian processes. From the main results of Yau and Yau [20, 22, 23], the state vector  $\mathbf{x}(t)$  can be estimated from the observation vectors  $\{\mathbf{y}(s) \mid s \in [0, t]\}$  by solving the Kolmogorov equations. Specifically, suppose that a set of observations  $\{\mathbf{y}(\tau_0), \dots, \mathbf{y}(\tau_{N_\tau})\}$  is measured. For each time period  $[\tau_{k-1}, \tau_k]$ ,  $k = 1, \dots, N_\tau$ , we solve the Kolmogorov equations of the form

$$(2) \quad \begin{cases} \frac{\partial \tilde{u}_k}{\partial t}(t, \mathbf{s}) = \frac{1}{2} \Delta \tilde{u}_k(t, \mathbf{s}) + \sum_{d=1}^D p_d(\mathbf{s}) \frac{\partial \tilde{u}_k}{\partial s_d}(t, \mathbf{s}) + q(\mathbf{s}) \tilde{u}_k(t, \mathbf{s}), & t \in [\tau_{k-1}, \tau_k], \\ \tilde{u}_k(\tau_{k-1}, \mathbf{s}) = \exp \left\{ \sum_{j=1}^M [y_j(\tau_{k-1}) - y_j(\tau_{k-2})] h_j(\mathbf{s}) \right\} \tilde{u}_{k-1}(\tau_{k-1}, \mathbf{s}), \\ \tilde{u}_1(0, \mathbf{s}) = \sigma_0(\mathbf{s}) \exp \left\{ \sum_{j=1}^M y_j(\tau_0) h_j(\mathbf{s}) \right\}, \end{cases}$$

for  $k = 2, \dots, N_\tau$  and  $\tau_0 = 0$ , where  $\Delta = \sum_{i=1}^D \frac{\partial^2}{\partial s_i^2}$ ,

$$(3) \quad p_d(\mathbf{s}) = -f_d(\mathbf{s}), \quad d = 1, \dots, D,$$

and

$$(4) \quad q(\mathbf{s}) = - \left[ \sum_{d=1}^D \frac{\partial f_d}{\partial s_d}(\mathbf{s}) + \frac{1}{2} \sum_{j=1}^M h_j^2(\mathbf{s}) \right].$$

For  $t \in [\tau_{k-1}, \tau_k]$ ,  $k = 2, \dots, N_\tau$ , we compute the expectation of  $\tilde{u}_k(t, \mathbf{s})$  with respect to  $s_d$  over  $\mathbb{R}^D$  by

$$(5) \quad \hat{x}_d(t) = \int_{\mathbb{R}^D} s_d \tilde{u}_k(t, \mathbf{s}) \, d\mathbf{s},$$

for  $d = 1, \dots, D$ . Then the state vector  $\mathbf{x}(t)$  in (1) can be estimated by  $\hat{\mathbf{x}}(t) = (\hat{x}_1(t), \dots, \hat{x}_D(t))^\top$ .

## 2. Numerical Algorithms

In order to solve the nonlinear filtering problem (1), we first generate states and observations  $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=0}^{N_\tau}$  by using Euler forward difference method with Gaussian noise. Based on the Yau-Yau's method, we first propose an implicit Euler method (IEM) for solving the Kolmogorov equations (2) which is stable and reliable, but costly. Furthermore, we develop the quasi-implicit Euler method (QIEM) for solving the Kolmogorov equations (2) which is also stable and reliable, but much more efficient because the fast Fourier transformation (FFT) can be applied in the QIEM.

### 2.1. The State and Observation Generator

Given a terminal time  $\Gamma$ , we generate a set of states and observations  $\{\mathbf{x}_k, \mathbf{y}_k\}_{k=0}^{N_\tau}$  by Euler forward difference method. We partition the time interval  $[0, \Gamma]$  uniformly as

$$(6) \quad \mathcal{P}_{[0, \Gamma]} = \{0 = \tau_0 < \tau_1 < \dots < \tau_{N_\tau} = \Gamma\},$$

where  $\tau_k - \tau_{k-1} = \Delta\tau$ ,  $k = 1, 2, \dots, N_\tau$ . With the Euler forward discretization, the signal observation model in (1) can be simulated by

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

where  $\mathbf{x}_0$  is the 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 Brownian motion with  $\mathbf{v}, \mathbf{w} \sim \mathcal{N}(0, 1)$ . The algorithm in detail is stated as follows.

---

**Algorithm 1** State and Observation Generator

---

**Input:** a terminal time  $\Gamma$ , time step  $\Delta\tau$ , the initial state  $\mathbf{x}_0$ , the vector-valued functions  $\mathbf{f}, \mathbf{h}$

**Output:** the state  $\mathbf{x}(t)$  and observation  $\mathbf{y}(t)$  at  $t = \tau_0, \dots, \tau_{N_\tau}$

- 1:  $N_\tau = \frac{\Gamma}{\Delta\tau} + 1$ .
  - 2: Set the initial state  $\mathbf{x}(\tau_0) = \mathbf{x}_0$ .
  - 3: Generate  $\mathbf{v}, \mathbf{w} \sim \mathcal{N}(0, 1)$ ,  $\mathbf{v} \perp \mathbf{w}$ .
  - 4: **for**  $k = 1$  to  $N_\tau$  **do**
  - 5:    $\mathbf{x}(\tau_k) = \mathbf{x}(\tau_{k-1}) + \mathbf{f}(\mathbf{x}(\tau_{k-1}))\Delta\tau + \mathbf{v}\sqrt{\Delta\tau}$ .
  - 6: **end for**
  - 7:  $\mathbf{y}(\tau_0) = 0$ .
  - 8: **for**  $k = 1$  to  $N_\tau$  **do**
  - 9:    $\mathbf{y}(\tau_k) = \mathbf{y}(\tau_{k-1}) + \mathbf{h}(\mathbf{x}(\tau_{k-1}))\Delta\tau + \mathbf{w}\sqrt{\Delta\tau}$ .
  - 10: **end for**
- 

## 2.2. Implicit Euler Method (IEM) for Kolmogorov Equations

We now propose the IEM for solving the Kolmogorov equations (2). From the equation (5), the state vector  $\mathbf{x}(t)$  can be estimated by the solution of the equation (2). For each time interval, we partition  $[\tau_{k-1}, \tau_k]$ ,  $k = 1, \dots, N_\tau$ , uniformly by

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

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

$$\mathcal{P}_{[0, \Gamma]}^* = \bigcup_{k=1}^{N_\tau} \mathcal{P}_{[\tau_{k-1}, \tau_k]}$$

$$= \left\{ 0 = \tau_0 = t_0^{(1)} < \cdots < t_{N_t}^{(1)} = \tau_1 = t_0^{(2)} < \cdots \right. \\ \left. < t_{N_t}^{(2)} = \tau_2 = t_0^{(3)} < \cdots < t_{N_t}^{(N_\tau)} = \tau_{N_\tau} = \Gamma \right\}$$

forms a refinement of the partition  $\mathcal{P}_{[0,\Gamma]}$  in (6). On the other hand, the space interval  $[-R, R]$  can be uniformly discretized by

$$\mathcal{P}_{[-R,R]} = \{-R = s_0 < s_2 < \cdots < s_{N_s} = R\},$$

where  $s_j - s_{j-1} = \Delta s$ ,  $j = 1, 2, \dots, N_s$  and  $R$  is a suitably large number so that the Gaussian distribution can be ignored outside  $[-R, R]$ . For the discretization of an  $D$ -cell  $[-R, R]^D \subset \mathbb{R}^D$ , we consider an ordered set of the power set of  $\mathcal{P}_{[-R,R]}$ ,

$$\mathcal{P}_{[-R,R]}^D = \{\mathbf{s}_j\}_{j=1}^{(N_s)^D},$$

where  $\mathbf{s}_j = (s_j^{(1)}, s_j^{(2)}, \dots, s_j^{(D)})^\top$ ,  $s_j^{(d)} \in \mathcal{P}_{[-R,R]}$ ,  $j = 1, \dots, (N_s)^D$ ,  $d = 1, \dots, D$ . In the discrete model of the equation (2), we set up the Dirichlet boundary condition of the domain  $[-R, R]^D$  to be zero. For the  $d$ -th dimension of the space,  $d = 1, \dots, D$ , the second order partial differential operator can be approximated by using the Euler central difference scheme

$$(7) \quad \frac{\partial^2 \tilde{u}}{\partial s^2}(t_n, s_j) \approx \left[ \alpha \left( \frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}}{(\Delta s)^2} \right) + \beta \left( \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{(\Delta s)^2} \right) \right],$$

where  $U_j^n \equiv \tilde{u}(t_n, s_j)$  and  $\alpha + \beta = 1$ ,  $\alpha, \beta \geq 0$ . Similarly, the partial differential operator can be approximated by

$$(8) \quad \frac{\partial \tilde{u}}{\partial s}(t_n, s_j) \approx \left[ \alpha \left( \frac{U_{j+1}^{n+1} - U_{j-1}^{n+1}}{2\Delta s} \right) + \beta \left( \frac{U_{j+1}^n - U_{j-1}^n}{2\Delta s} \right) \right].$$

In other words, the discretized Laplacian operator in (2) can be represented by the matrix

$$(9) \quad \mathbb{T}_d \equiv \left[ \left( \bigotimes_{k=1}^{D-d} I_{N_s} \right) \otimes \mathbb{T}_d \otimes \left( \bigotimes_{k=D-d+2}^D I_{N_s} \right) \right],$$

where  $\otimes$  denotes the Kronecker product (or tensor product),  $I_{N_s}$  is the identity matrix of size  $N_s$  and the matrix

$$\mathbb{T}_d = \frac{1}{(\Delta s)^2} \begin{bmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{bmatrix}.$$

Similarly, the discretized partial differential operator can be represented by the matrix

$$(10) \quad \mathbb{K}_d \equiv \left[ \left( \bigotimes_{k=1}^{D-d} I_{N_s} \right) \otimes \mathbb{K}_d \otimes \left( \bigotimes_{k=D-d+2}^D I_{N_s} \right) \right],$$

where the matrix

$$\mathbb{K}_d = \frac{1}{2\Delta s} \begin{bmatrix} 0 & 1 & & & \\ -1 & 0 & 1 & & \\ & -1 & 0 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & -1 & 0 \end{bmatrix}.$$

For each time period  $t_n^{(k)} \in [\tau_{k-1}, \tau_k]$ , the partial differential of time  $\frac{\partial \tilde{u}_k}{\partial t}(t_n, \mathbf{s})$  in (2) can be discretized by

$$(11) \quad \frac{\partial \tilde{u}_k}{\partial t}(t_n, \mathbf{s}) \approx \frac{U^{(k),n+1} - U^{(k),n}}{\Delta t},$$

where  $U^{(k),n} \equiv (\tilde{u}_k(t_n, s_1), \tilde{u}_k(t_n, s_2), \dots, \tilde{u}_k(t_n, s_{(N_s)^D}))^\top$ . Hence the numerical scheme can be written in the form

$$(12) \quad \frac{U^{(k),n} - U^{(k),n-1}}{\Delta t} = \alpha \mathbb{A} U^{(k),n} + \beta \mathbb{A} U^{(k),n-1},$$

where  $\alpha + \beta = 1$ ,  $\alpha, \beta \geq 0$  and the matrix

$$(13) \quad \mathbb{A} = \frac{1}{2} \sum_{d=1}^D \mathbb{T}_d + \sum_{d=1}^D \mathbb{P}_d \mathbb{K}_d + \mathbb{Q},$$

$\mathbb{P}_d \equiv \text{diag}\{p_d(\mathbf{s}_i)\}_{i=1}^{(N_s)^D}$ ,  $\mathbb{Q} \equiv \text{diag}\{q(\mathbf{s}_i)\}_{i=1}^{(N_s)^D}$  are diagonal matrices. For each  $t_j^{(k)} \in [\tau_{k-1}, \tau_k]$ ,  $j = 1, \dots, N_t$ , we solve the linear system

$$(14) \quad [I - \alpha(\Delta t)\mathbb{A}]U^{(k),n} = [I + \beta(\Delta t)\mathbb{A}]U^{(k),n-1},$$

for  $k = 1, \dots, N_\tau$  with the initial vector  $U^{(k),0} \equiv (U_1^{(k),0}, U_2^{(k),0}, \dots, U_{(N_s)^D}^{(k),0})^\top$ , in which

$$(15) \quad U_j^{(k),0} = \exp \left\{ \sum_{d=1}^M [y(\tau_{k+1}) - y(\tau_k)] h_d(s_j) \right\} U^{(k-1),N_t},$$

$$j = 1, \dots, (N_s)^D.$$

Each vector  $U^{(k),n}$  in (14) should be normalized such that  $\sum_{j=1}^{(N_s)^D} U_j^{(k),n} = 1$ . Then the vector  $U^{(k),n}$  represents the probability distribution of the state at time  $t_n^{(k)}$ . Finally, we compute the expectation

$$(16) \quad \hat{\mathbf{x}} \left( t_n^{(k)} \right) = \sum_{j=1}^{(N_s)^D} s_j U_j^{(k),n}$$

as our estimation for the real state  $\mathbf{x} \left( t_n^{(k)} \right)$ . In particular, we choose the parameter  $\alpha = 1$  and  $\beta = 0$ , since the implicit scheme is stable in most of case while the explicit scheme ( $\alpha = 0, \beta = 1$ ) is usually unstable. The algorithm in detail for solving the nonlinear filtering problem is stated in Algorithm 2.

### 2.3. Quasi-Implicit Euler Method (QIEM) for Kolmogorov Equations with Fast Fourier Transformations (FFTs)

The FFTs for the discretized Laplacian matrix is well-known. In the equation (13), we separate the matrix  $\mathbb{A}$  into 2 parts,

$$\mathbb{A} = \frac{1}{2} \Delta_D + \mathbb{A}_f,$$

where  $\Delta_D = \sum_{d=1}^D \mathbb{T}_d$  and  $\mathbb{A}_f = \sum_{d=1}^D \mathbb{P}_d \mathbb{K}_d + \mathbb{Q}$ . Then we slightly modify the equation (12) into the form of the quasi-implicit scheme

$$(17) \quad \frac{U^{(k),n} - U^{(k),n-1}}{\Delta t} = \frac{1}{2} \Delta_D U^{(k),n} + \mathbb{A}_f U^{(k),n-1}.$$

**Algorithm 2** IEM for Nonlinear Filter

---

**Input:** A terminal time  $\Gamma$ , the space interval  $[-R, R]^D$ , the time step size  $\Delta t$ , the space step size  $\Delta s$ , the vector-valued functions  $\mathbf{f} = (f_1, \dots, f_D)^\top$ ,  $\mathbf{h} = (h_1, \dots, h_M)^\top$ , the observations  $\{\mathbf{y}_k\}_{k=1}^{N_\tau}$  and the initial state  $\sigma_0$

**Output:** the approximation of the state  $\hat{\mathbf{x}}(t)$

- 1: Set the values  $N_\tau = \frac{\Gamma}{\Delta\tau} + 1$ ,  $N_t = \frac{\Delta\tau}{\Delta t} + 1$  and  $N_s = \frac{2R}{\Delta s} + 1$ .
- 2: **for**  $j = 1$  to  $(N_s)^D$  **do**
- 3:    $U_j \leftarrow \sigma_0(s_j) \exp \left\{ \sum_{d=1}^M y_d(\tau_0) h_d(s_j) \right\}$ .
- 4: **end for**
- 5: **for**  $k = 1$  to  $N_\tau$  **do**
- 6:   **for**  $j = 1$  to  $(N_s)^D$  **do**
- 7:      $U_j \leftarrow \exp \left\{ \sum_{d=1}^M [y_d(\tau_{k+1}) - y_d(\tau_k)] h_d(s_j) \right\} U_j$  as in (15)
- 8:   **end for**
- 9:   **for**  $n = 1$  to  $N_t$  **do**
- 10:      $U_{tmp} = [I + \beta(\Delta t)\mathbb{A}] U$ .
- 11:     Solve the linear system  $[I - \alpha(\Delta t)\mathbb{A}] U = U_{tmp}$  as in (14)
- 12:     Normalize the solution  $U \leftarrow \frac{U}{\sum_{j=1}^{(N_s)^D} U_j}$ .
- 13:     Set the approximation of state  $\hat{\mathbf{x}}(t_n^{(k)}) = \sum_{j=1}^{(N_s)^D} s_j U_j$  as in (16)
- 14:   **end for**
- 15: **end for**

---

Similarly, for each  $t_j^{(k)} \in [\tau_{k-1}, \tau_k]$ ,  $j = 1, \dots, N_t$ , we solve the linear system

$$(18) \quad \left( I - \frac{\Delta t}{2} \Delta_D \right) U^{(k),n} = [I + \Delta t \mathbb{A}_f] U^{(k),n-1}.$$

Most important of all, the linear system (18) can be efficiently solved by FFTs. For the case of  $D = 1$  (one-dimensional case), the Laplacian matrix in (18) satisfies  $\Delta_1 \equiv \mathbb{T}_1$ . By Fourier sine transformation, we have the spectral decomposition

$$(19) \quad \mathbb{T}_1 = \frac{1}{(\Delta s)^2} W S W^*,$$

where  $W \equiv [W_{ij}]$  with

$$W_{ij} = \sqrt{\frac{2}{N_s + 1}} \sin \left( \frac{ij\pi}{N_s + 1} \right),$$

and

$$S = \text{diag} \left\{ -4 \sin^2 \left( \frac{i\pi}{2(N_s + 1)} \right) \right\}_{i=1}^{N_s}.$$

Then the linear system of  $\Delta_1$  can be solved by calling the MATLAB function "FFT". The algorithm in detail for solving the  $D$ -dimensional nonlinear filtering problem by applying the FFTs is stated in Algorithm 3.

---

**Algorithm 3** QIEM for Nonlinear Filter with FFTs

---

**Input:** A terminal time  $\Gamma$ , the space interval  $[-R, R]^D$ , the time step size  $\Delta t$ , the space step size  $\Delta s$ , the vector-valued functions  $\mathbf{f} = (f_1, \dots, f_D)^\top$ ,  $\mathbf{h} = (h_1, \dots, h_M)^\top$ , the observations  $\{\mathbf{y}_k\}_{k=1}^{N_\tau}$  and the initial state  $\sigma_0$

**Output:** the approximation of the state  $\hat{\mathbf{x}}(t)$

- 1: Set the values  $N_\tau = \frac{\Gamma}{\Delta\tau} + 1$ ,  $N_t = \frac{\Delta\tau}{\Delta t} + 1$  and  $N_s = \frac{2R}{\Delta s} + 1$ .
- 2: **for**  $j = 1$  to  $(N_s)^D$  **do**
- 3:      $U_j \leftarrow \sigma_0(s_j) \exp \left\{ \sum_{d=1}^M y_d(\tau_0) h_d(s_j) \right\}$ .
- 4: **end for**
- 5: **for**  $k = 1$  to  $N_\tau$  **do**
- 6:     **for**  $j = 1$  to  $(N_s)^D$  **do**
- 7:          $U_j \leftarrow \exp \left\{ \sum_{d=1}^M [y_d(\tau_{k+1}) - y_d(\tau_k)] h_d(s_j) \right\} U_j$  as in (15)
- 8:     **end for**
- 9:     **for**  $n = 1$  to  $N_t$  **do**
- 10:          $U_{tmp} \leftarrow \left[ I + \Delta t \left( \sum_{d=1}^D \mathbb{P}_d \mathbb{K}_d + Q \right) \right] U$ .
- 11:         Call FFTs:  $U \leftarrow \left( \otimes_{d=1}^D W^* \right) U_{tmp}$ .
- 12:         **for**  $j = 1$  to  $N_s$  **do**
- 13:              $U_j \leftarrow U_j / \left[ 1 + \frac{2\Delta t}{(\Delta s)^2} \sin^2 \left( \frac{j\pi}{2(N_s+1)} \right) \right]$ .
- 14:         **end for**
- 15:         Call IFFTs:  $U \leftarrow \left( \otimes_{d=1}^D W \right) U$
- 16:         Normalize the solution  $U \leftarrow \frac{U}{\sum_{j=1}^{(N_s)^D} U_j}$ .
- 17:         Set the approximation of state  $\hat{\mathbf{x}} \left( t_n^{(k)} \right) = \sum_{j=1}^{(N_s)^D} s_j U_j$  as in (16)
- 18:     **end for**
- 19: **end for**

---

#### 2.4. Higher-Order QIEM for 2-D Kolmogorov Equations with FFTs

The Laplacian matrix in (18) is a second-order approximation of the Laplacian operator. Hereafter, we consider a fourth-order accurate scheme for Laplacian operator which reduces the size of the discretization matrix considerably, but preserves the same accuracy as the second-order approximation. Since there is no general form of the higher-order scheme for Laplacian operator, for convenience in practice, we consider the Kolmogorov equations (2) in two-dimensional case.

The 9-point scheme for the discretized Laplacian operator  $\tilde{\Delta}_2$  is defined by

$$(20) \quad \begin{aligned} \tilde{\Delta}_2 U_{i,j} = \frac{1}{6(\Delta s)^2} [ & 4U_{i-1,j} + 4U_{i+1,j} + 4U_{i,j-1} + 4U_{i,j+1} \\ & + U_{i-1,j-1} + U_{i+1,j-1} + U_{i-1,j+1} + U_{i+1,j+1} - 20U_{i,j} ] \end{aligned}$$

which is a fourth-order approximation of Laplacian operator. The matrix form of (20) is represented as

$$(21) \quad \tilde{\Delta}_2 = \frac{1}{(\Delta s)^2} \begin{bmatrix} \Sigma & \Phi & & & \\ \Phi & \Sigma & \ddots & & \\ & \ddots & \ddots & \Phi & \\ & & & \Phi & \Sigma \end{bmatrix},$$

where

$$\Sigma = \frac{1}{3} \begin{bmatrix} -10 & 2 & & & \\ 2 & -10 & \ddots & & \\ & \ddots & \ddots & 2 & \\ & & & 2 & -10 \end{bmatrix}, \quad \Phi = \frac{1}{6} \begin{bmatrix} 4 & 1 & & & \\ 1 & 4 & \ddots & & \\ & \ddots & \ddots & 1 & \\ & & & 1 & 4 \end{bmatrix}.$$

In the following, we derive the fast Fourier transformation for solving the linear system  $\tilde{\Delta}_2 U = b$ . Note that

$$\begin{aligned} \tilde{\Delta}_2 &= \left( \frac{1}{6} [(T_1 + 6I) \otimes (T_1 + 6I)] - 6I \right) \\ &= \frac{1}{6(\Delta s)^2} ((WSW^* + 6I) \otimes (WSW^* + 6I)) - 36I \\ &= \frac{1}{6(\Delta s)^2} ((W \otimes W) ((S + 6I)W^* \otimes (S + 6I)W^*) - 36I) \\ &= \frac{1}{6(\Delta s)^2} ((W \otimes W) (((S + 6I) \otimes (S + 6I)) - 36(I \otimes I)) (W^* \otimes W^*)), \end{aligned}$$

where  $T_1 = \frac{1}{(\Delta s)^2} WSW^*$  as given in (19). Based on the QIEM Algorithm 3, the algorithm in detail for solving the two-dimensional nonlinear filtering problem by applying the fourth-order QIEM with FFTs is stated in Algorithm 4.

---

**Algorithm 4** Fourth-order QIEM for 2-D Nonlinear Filter with FFTs
 

---

**Input:** A terminal time  $\Gamma$ , the space interval  $[-R, R]$ , the time step size  $\Delta t$ , the space step size  $\Delta s$ , the functions  $\mathbf{f} = (f_1, f_2)^\top$ ,  $\mathbf{h} = (h_1, h_2)^\top$ , the observations  $\{\mathbf{y}(\tau_k) = (y_1(\tau_k), y_2(\tau_k))^\top\}_{k=1}^{N_\tau}$  and the initial state  $\sigma_0$

**Output:** the approximation of the state  $\hat{\mathbf{x}}(t)$

- 1: Set the values  $N_\tau = \frac{\Gamma}{\Delta\tau} + 1$ ,  $N_t = \frac{\Delta\tau}{\Delta t} + 1$  and  $N_s = \frac{2R}{\Delta s} + 1$ .
  - 2: **for**  $j = 1$  to  $(N_s)^2$  **do**
  - 3:      $U_j = \sigma_0(s_j) \exp\{y_1(\tau_0)h_1(s_j) + y_2(\tau_0)h_2(s_j)\}$ .
  - 4: **end for**
  - 5: **for**  $k = 1$  to  $N_\tau$  **do**
  - 6:     **for**  $j = 1$  to  $N_s$  **do**
  - 7:          $U_j = \exp\{[y_1(\tau_{k+1}) - y_1(\tau_k)]h_1(s_j) + [y_2(\tau_{k+1}) - y_2(\tau_k)]h_2(s_j)\} U_j$ .
  - 8:     **end for**
  - 9:     **for**  $n = 1$  to  $N_t$  **do**
  - 10:          $U_{tmp} \leftarrow [I + \Delta t (\mathbb{P}_1 \mathbb{D}_1 + \mathbb{P}_2 \mathbb{D}_2 + \mathbb{Q})] U$ .
  - 11:         % Here the matrices  $W$  and  $S$  are defined in (19).
  - 12:         Call FFT:  $U \leftarrow (W^* \otimes W^*) U_{tmp}$ .
  - 13:         **for**  $j = 1$  to  $N_s$  **do**
  - 14:              $U_j \leftarrow \left[ (I \otimes I) - \frac{\Delta t}{12(\Delta s)^2} ((S + 6I) \otimes (S + 6I) - 36(I \otimes I)) \right]^{-1} U_j$ .
  - 15:         **end for**
  - 16:         Call IFFT:  $U \leftarrow (W \otimes W) U$ .
  - 17:         Normalize the solution  $U \leftarrow \frac{U}{\sum_{j=1}^{N_s} U_j}$ .
  - 18:         Set the approximation of state  $\hat{\mathbf{x}}\left(t_n^{(k)}\right) = \sum_{j=1}^{(N_s)^2} s_j U_j$ .
  - 19:     **end for**
  - 20: **end for**
- 

## 2.5. Superposition Technique

The computation of nonlinear filtering problems is a real time problem. Saving the computational cost becomes an essential issue. In order to solve the nonlinear filtering problem in a more efficient way, we adopt the superposition technique. First, we use the Dirac delta functions

$$\left\{ \delta_{\mathbf{c}_k}(\mathbf{s}) = e^{-\eta \|\mathbf{s} - \mathbf{c}_k\|^2} \mid \mathbf{c}_k = (c_{k_1}, \dots, c_{k_D})^\top \in [-R, R]^D \right\}_{k=1}^{N_\delta}$$

with  $\mathbf{s} = (s_1, \dots, s_D)^\top \in \mathbb{R}^D$ ,  $\eta$  being a suitable real number and  $\|\mathbf{s} - \mathbf{c}_k\|^2 = \sum_{j=1}^D (s_j - c_{k_j})^2$ , as various initials to compute the approximate states for the nonlinear filtering problem, separately. Then we store all the fundamental solutions  $\{v_k\}_{k=1}^{N_\delta}$  corresponding to the Dirac delta functions  $\{\delta_{\mathbf{c}_k}\}_{k=1}^{N_\delta}$ .

In practice, for any given initial probability density function  $u_0$ , we calculate a set of coefficients  $\{\alpha_k\}_{k=1}^{N_\delta}$  of the linear combination of Dirac delta

functions satisfying

$$u_0 \approx \sum_{k=1}^{N_\delta} \alpha_k \delta_{\mathbf{c}_k}.$$

Then the approximate probability density function of the state  $v$  can be directly obtained by computing the linear combination of the fundamental solutions

$$v \approx \sum_{k=1}^{N_\delta} \alpha_k v_k.$$

This method significantly saves a large amount of computational cost.

### 3. Nonnegativity and Convergence of IEM/QIEM

In this section, first, we will show that the linear operator defined in (14) is a nonnegative operator. Then, we will prove the convergence of IEM and QIEM.

#### 3.1. Sufficient Condition of Nonnegative Operator

In the linear system (14), the solution of each time step represents a probability distribution of the space. In order to guarantee the property that each solution is nonnegative, we find the sufficient condition such that the matrix  $(I - \Delta t \mathbb{A})^{-1}$  is a nonnegative operator. First, let us introduce the definition of an  $M$ -matrix and its equivalence condition.

**Definition.** 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). [4] *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  $Bv > 0$ .*

The following theorem shows that the vector  $U$  in each iteration of step 11 in Algorithm 2 preserves nonnegativity of the probability density function.

**Theorem 1** (Sufficient Condition of Nonnegative Operator). *Given real-valued functions  $p_d$ ,  $d = 1, \dots, D$ ,  $q$ , a time step  $\Delta t$  and a space step  $\Delta s$ . Let  $B \equiv I - \Delta t \mathbb{A}$ , where  $\mathbb{A}$  is defined as (13). If for each  $\mathbf{s} \in [-R, R]^D$ ,*

$$(22) \quad |p_d(\mathbf{s})| < \frac{1}{\Delta s}, \quad |q(\mathbf{s})| < \frac{1}{\Delta t},$$

for  $d = 1, \dots, D$ , then  $B$  is an  $M$ -matrix. That is,  $B^{-1} \geq 0$  is a nonnegative operator.

*Proof.* First we check  $B_{ij} \leq 0$  for  $i \neq j$ . From the structure of the matrices in (9) and (10), we see that for  $i \neq j$  either  $B_{ij} = 0$  or

$$\begin{aligned}
 B_{ij} &= -\Delta t \left( \frac{1}{2(\Delta s)^2} + \frac{p_d(\mathbf{s})}{2\Delta s} \right) \\
 &= -\frac{\Delta t}{2(\Delta s)^2} (1 + \Delta s p_d(\mathbf{s})) \\
 &\leq -\frac{\Delta t}{2(\Delta s)^2} (1 - \Delta s |p_d(\mathbf{s})|) \\
 (23) \quad &< 0.
 \end{aligned}$$

The inequality (23) follows from the first equation of (22). Next, we check  $B \mathbf{1} > 0$ , where  $\mathbf{1} \equiv (1, 1, \dots, 1)^\top > 0$ . Note that  $B \mathbf{1}$  is a vector whose entry is the row sum of  $B$ . Hence

$$\begin{aligned}
 B \mathbf{1} &= 1 - \Delta t \left( \frac{-k}{2(\Delta s)^2} + \frac{\sum_{d=1}^k (-1)^{m_d} p_d(\mathbf{s})}{2\Delta s} + q(\mathbf{s}) \right) \\
 &\geq 1 - \Delta t \left( \frac{-k}{2(\Delta s)^2} + \frac{k \max_d |p_d(\mathbf{s})|}{2\Delta s} + q(\mathbf{s}) \right) \\
 &= (1 - \Delta t q(\mathbf{s})) + \frac{k\Delta t}{2(\Delta s)^2} \left( 1 - \Delta s \max_d |p_d(\mathbf{s})| \right) \\
 &\geq (1 - \Delta t |q(\mathbf{s})|) + \frac{k\Delta t}{2(\Delta s)^2} \left( 1 - \Delta s \max_d |p_d(\mathbf{s})| \right) \\
 (24) \quad &> 0,
 \end{aligned}$$

for some  $k \in \{1, \dots, D\}$ ,  $m_d \in \{0, 1\}$ . The inequality (24) follows from (22). By Lemma 1,  $B$  is an  $M$ -matrix. That is,  $B^{-1} \geq 0$ .  $\square$

Consequently, by Theorem 1, the vector  $U = [I - \Delta t \mathbb{A}]^{-1} U_{tmp}$  in Step 11 of Algorithm 2 is nonnegative.

### 3.2. Convergence of IEM/QIEM

In this section, we prove the convergence of the IEM and the QIEM by checking the consistency and the stability of the schemes.

**Theorem 2** (Consistency of IEM/QIEM). *The local truncation errors of IEM (12) and QIEM (18) are  $O(\Delta t + (\Delta s)^2)$ . That is, IEM and QIEM are consistent.*

*Proof.* The first-order Taylor expansion of  $u$  at the point  $(t + \Delta t, s)$  implies

$$(25) \quad \frac{\partial u}{\partial t}(t, s) = \frac{u(t + \Delta t, s) - u(t, s)}{\Delta t} + O(\Delta t).$$

The third-order Taylor expansions of  $u$  at the points  $(t, s + \Delta s)$  and  $(t, s - \Delta s)$ , respectively, lead to

$$(26) \quad u(t, s + \Delta s) = u(t, s) + \Delta s \frac{\partial u}{\partial s}(t, s) + \frac{(\Delta s)^2}{2} \frac{\partial^2 u}{\partial s^2}(t, s) + \frac{(\Delta s)^3}{6} \frac{\partial^3 u}{\partial s^3}(t, s) + O((\Delta s)^4),$$

and

$$(27) \quad u(t, s - \Delta s) = u(t, s) - \Delta s \frac{\partial u}{\partial s}(t, s) + \frac{(\Delta s)^2}{2} \frac{\partial^2 u}{\partial s^2}(t, s) - \frac{(\Delta s)^3}{6} \frac{\partial^3 u}{\partial s^3}(t, s) + O((\Delta s)^4).$$

By adding the equations (26) and (27), we obtain

$$(28) \quad \frac{\partial^2 u}{\partial s^2}(t, s) = \frac{u(t, s + \Delta s) - 2u(t, s) + u(t, s - \Delta s)}{(\Delta s)^2} + O((\Delta s)^2).$$

Similarly, by subtracting the equation (27) from (26), we obtain

$$(29) \quad \frac{\partial u}{\partial s}(t, s) = \frac{u(t, s + \Delta s) - u(t, s - \Delta s)}{2\Delta s} + O((\Delta s)^2).$$

Hence, according to the equations (7), (8) and (11), respectively, the equations (28), (29) and (25) show the local truncation error of (12) is  $O(\Delta t + (\Delta s)^2)$ .  $\square$

**Theorem 3** (Sufficient Condition for Stability of IEM). *The IEM (12) is stable if the function  $\mathbf{f}$  in (1) satisfies*

$$(30) \quad \nabla \cdot \mathbf{f} \geq 0.$$

*Proof.* We claim that IEM (12) is stable by applying von Neumann stability analysis. Let  $U_j^n = \xi(k)^n e^{\iota kj(\Delta s)}$ , where  $\iota \equiv \sqrt{-1}$  and  $\xi(k)$  is known as the amplification factor. Substituting  $U_j^n$  into the scheme (12), we obtain

$$\begin{aligned} \frac{\xi(k) - 1}{\Delta t} &= \frac{\xi(k)}{2(\Delta s)^2} \left( e^{\iota kj(\Delta s)} - 2 + e^{-\iota kj(\Delta s)} \right) \\ &\quad + \frac{\xi(k)}{2\Delta s} \left( e^{\iota kj(\Delta s)} - e^{-\iota kj(\Delta s)} \right) p(x) + \xi(k)q(s). \end{aligned}$$

That is,

$$\begin{aligned} \frac{1}{\xi(k)} &= 1 - \Delta t \left( \frac{e^{\iota kj(\Delta s)} - 2 + e^{-\iota kj(\Delta s)}}{2(\Delta s)^2} + \frac{e^{\iota kj(\Delta s)} - e^{-\iota kj(\Delta s)}}{2\Delta s} p(s) + q(s) \right) \\ &= 1 + \frac{\Delta t}{(\Delta s)^2} (1 - \cos(kj(\Delta s))) - (\Delta t)q(s) - \iota \frac{\Delta t}{\Delta s} \sin(kj(\Delta s)) p(s). \end{aligned}$$

If  $\nabla \cdot \mathbf{f} \geq 0$ , then  $q(s) \equiv - \left[ \nabla \cdot \mathbf{f} + \frac{1}{2} \sum_{j=1}^M h_j^2(\mathbf{s}) \right] \leq 0$ . It follows that

$$\begin{aligned} \frac{1}{|\xi(k)|^2} &= \left( 1 + \frac{\Delta t}{(\Delta s)^2} (1 - \cos(kj(\Delta s))) - (\Delta t)q(s) \right)^2 \\ &\quad + \left( \frac{\Delta t}{\Delta s} \sin(kj(\Delta s)) p(s) \right)^2 \\ &\geq (1 - (\Delta t)q(s))^2 \geq 1. \end{aligned}$$

That is,  $|\xi(k)|^2 \leq 1$ . This implies that IEM (12) is stable under the assumption that  $\nabla \cdot \mathbf{f} \geq 0$ .  $\square$

**Theorem 4** (Sufficient Condition for Stability of QIEM). *The QIEM (18) is stable if both the step size of time  $\Delta t$  and the step size of space  $\Delta s$  are sufficient small. More precisely,  $\Delta t$  and  $\Delta s$  satisfy*

$$(31) \quad (\Delta s)^2 (2q(s) + q(s)^2 \Delta t) + \Delta t p(s)^2 \leq 2.$$

*Proof.* As in the proof of Theorem 3, we substitute  $U_j^n = \xi(k)^n e^{\iota kj(\Delta s)}$  into the scheme (18) and obtain

$$\begin{aligned} \frac{\xi(k) - 1}{\Delta t} &= \frac{\xi(k)}{2(\Delta s)^2} \left( e^{\iota kj(\Delta s)} - 2 + e^{-\iota kj(\Delta s)} \right) \\ &\quad + \frac{1}{2\Delta s} \left( e^{\iota kj(\Delta s)} - e^{-\iota kj(\Delta s)} \right) p(s) + q(s). \end{aligned}$$

That is,

$$\xi(k) = \frac{1 + \Delta t q(s) + \frac{\Delta t}{\Delta s} i \sin(kj(\Delta s)) p(s)}{1 - \frac{\Delta t}{(\Delta s)^2} (\cos(kj(\Delta s)) - 1)}.$$

If  $\Delta s$  and  $\Delta t$  satisfy  $(\Delta s)^2 (2q(s) + q(s)^2 \Delta t) + \Delta t p(s)^2 \leq 2$ , then

$$(\Delta s)^2 (2q(s) + q(s)^2 \Delta t) + \Delta t p(s)^2 \leq 2 + \frac{\Delta t}{(\Delta s)^2}.$$

Multiplying both side by  $\frac{\Delta t}{(\Delta s)^2}$ , we have

$$2q(s)\Delta t + q(s)^2(\Delta t)^2 + \frac{(\Delta t)^2}{(\Delta s)^2} p(s)^2 \leq \frac{2\Delta t}{(\Delta s)^2} + \frac{(\Delta t)^2}{(\Delta s)^4}.$$

Adding both side by 1, we obtain

$$(1 + \Delta t q(s))^2 + \left( \frac{\Delta t}{\Delta s} p(s) \right)^2 \leq \left( 1 + \frac{\Delta t}{(\Delta s)^2} \right)^2.$$

It follows that

$$\begin{aligned} |\xi(k)|^2 &= \frac{(1 + \Delta t q(s))^2 + \left( \frac{\Delta t}{\Delta s} \sin(kj(\Delta s)) p(s) \right)^2}{\left( 1 - \frac{\Delta t}{(\Delta s)^2} (\cos(kj(\Delta s)) - 1) \right)^2} \\ &\leq \frac{(1 + \Delta t q(s))^2 + \left( \frac{\Delta t}{\Delta s} p(x) \right)^2}{\left( 1 + \frac{\Delta t}{(\Delta s)^2} \right)^2} \leq 1. \end{aligned}$$

This implies QIEM (18) is stable under the given assumption.  $\square$

**Theorem 5** (Sufficient Conditions for Convergence of IEM/QIEM). *The IEM and QIEM converge if the conditions of (30) and (31) hold, respectively.*

*Proof.* From the consistency of IEM/QIEM in Theorem 2 as well as the stabilities of IEM and QIEM in Theorem 3 and Theorem 4, respectively, the convergence of IEM/QIEM follows by the Lax-Richtmyer equivalence theorem [8] immediately.  $\square$

#### 4. Numerical Experiments

In this section, we first show some numerical results by applying IEM/QIEM to 1-D and 2-D nonlinear filtering problems. Then we compare the accuracy of IEM/QIEM with the extended Kalman filtering method [16, 7, 17].

### 4.1. Numerical Results of IEM/QIEM

First, we consider the 1-D cubic sensor problem [19, 18] which is important and widely used in the nonlinear filtering control. The associated functions  $f$  and  $h$  in (1) for the 1-D cubic sensor problem are given by

$$(32) \quad f(s) = 0, \quad h(s) = s^3.$$

The parameters  $\Delta s = 0.01$ ,  $\Delta t = 0.0001$ ,  $R = 10$  and  $N_\delta = 201$ . Figure 1(a)(b) show the numerical results by QIEM for the 1-D cubic sensor problem starting with two different random generators. The black and blue lines show the real states and the approximation for the real states, respectively. We see that two lines in Figure 1(a)(b) match very well.

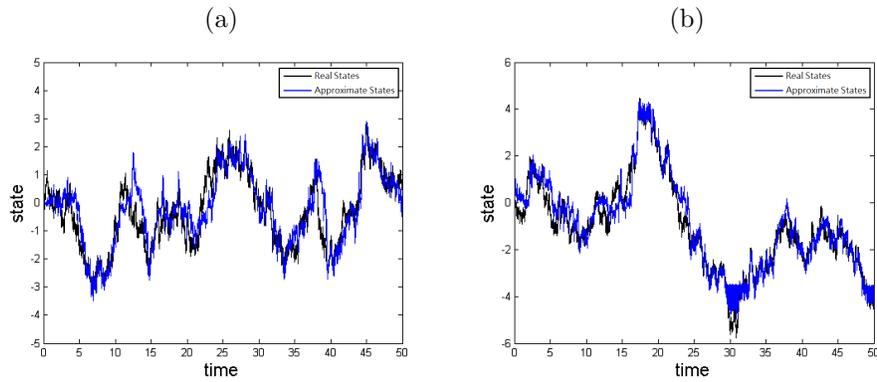


Figure 1: The numerical results by QIEM for 1-D cubic sensor problems

We now consider a 2-D nonlinear filtering problem with the associated functions  $\mathbf{f}$  and  $\mathbf{h}$  in (1) being given by

$$(33) \quad \begin{aligned} \mathbf{f}(s_1, s_2) &= (-\cos(s_1) - \sin(s_2), \cos(s_1) + \sin(s_2)), \\ \mathbf{h}(s_1, s_2) &= (s_1 - s_2, s_1 + s_2). \end{aligned}$$

Figure 3(a)(b) show the numerical results by QIEM for the 2-D nonlinear filtering problem starting with two different random generators. The black and blue lines show the real states and the approximation, respectively. We see that two lines in Figure 3(a)(b) still match well. In Figure 2, we show the density functions of the 2-D nonlinear filtering problem at  $t = 0$  and  $t = 1.2$ , respectively.

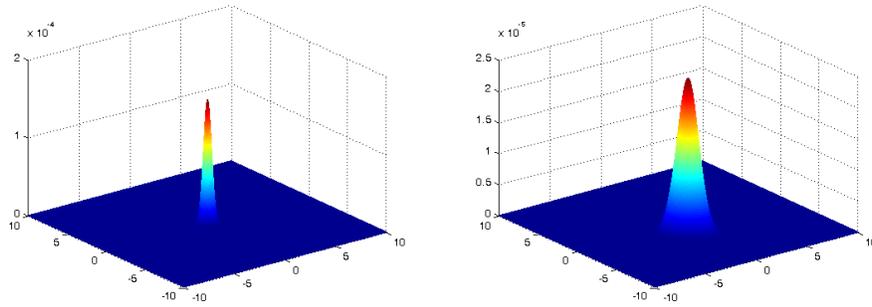
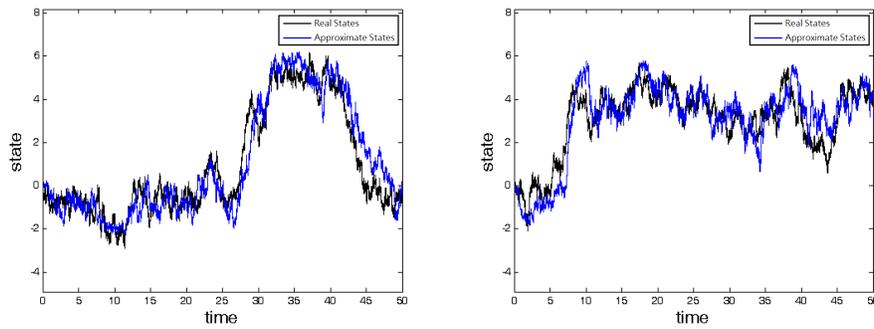


Figure 2: The density functions of the 2-D nonlinear filtering problem at  $t = 0$  (left) and  $t = 1.2$  (right), respectively.

(a) left:  $x_1$  component, right:  $x_2$  component



(b) left:  $x_1$  component, right:  $x_2$  component

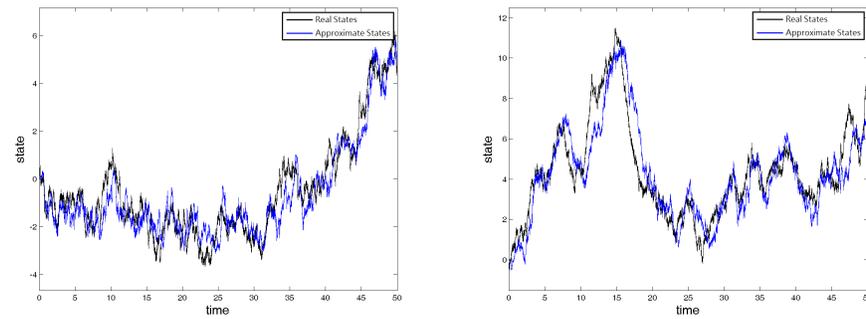


Figure 3: The numerical results of QIEM for the 2-D nonlinear filtering problems

According to our numerical experience, QIEM has the same accuracy as IEM, but QIEM has higher efficiency (less computational cost) than IEM for both 1-D and 2-D nonlinear filtering problems. Here, we ignore presenting the numerical results of IEM.

#### 4.2. Comparison between Extended Kalman Filter and QIEM

The extended Kalman filter (EKF) [16, 7, 17] is a well-known method of dealing nonlinear filtering problem by using linear approach. We consider a 1-D nonlinear filtering problem in (1) with the associated functions  $f(x) = \cos(x)$  and  $h(x) = x^n$ , for  $n = 2, \dots, 6$ .

In Figure 4, we show the comparison of numerical results between EKF and QIEM. The black, blue and red lines show the real states, the approximate states by QIEM and by EKF, respectively. We see that the approximate states by QIEM match much better than the approximate states by EKF.

In order to compare the accuracy of QIEM with EKF, we compute the root-mean-square error  $E_{RMS}$  between the approximate state  $\hat{x}(\mathbf{t}) = (\hat{x}(t_1), \dots, \hat{x}(t_{N_\tau}))^\top$  and the real state  $x(\mathbf{t}) = (x(t_1), \dots, x(t_{N_\tau}))^\top$  by

$$E_{RMS}(\hat{x}(\mathbf{t})) = \sqrt{\frac{\sum_{i=1}^{N_\tau} (\hat{x}(t_i) - x(t_i))^2}{N_\tau}}.$$

In Table 1, we show the root-mean-square error  $E_{RMS}$  of two methods. We see that the  $E_{RMS}$  of EKF increases from 0.64 to 1.33 when  $n$  of  $h(x)$  grows from 2 to 6, while  $E_{RMS}$  of QIEM is between 0.43 to 0.56.

#### 4.3. Numerical Results of Superposition Technique

In this subsection, we show some numerical results of the superposition technique (SPT) proposed in Section 2.5. We consider the 1-D cubic sensor problem and 2-D nonlinear filtering problem with  $\mathbf{f}$  and  $\mathbf{h}$  defined in (32) and (33), respectively. The initial probability density functions for 1-D and 2-D problems are chosen by  $u_0(s) = e^s$  and  $u(s_1, s_2) = e^{-5(s_1^2 + s_2^2)}$ , respectively.

Figure 5(a)(b) and 6 show numerical results of SPT for 1-D cubic sensor problem and 2D nonlinear filtering problem, respectively. The black, blue and red lines show the real states, the approximate states by QIEM and by SPT, respectively. We see that SPT matches well as long as QIEM works well. Nevertheless, SPT saves a lot of computing time.

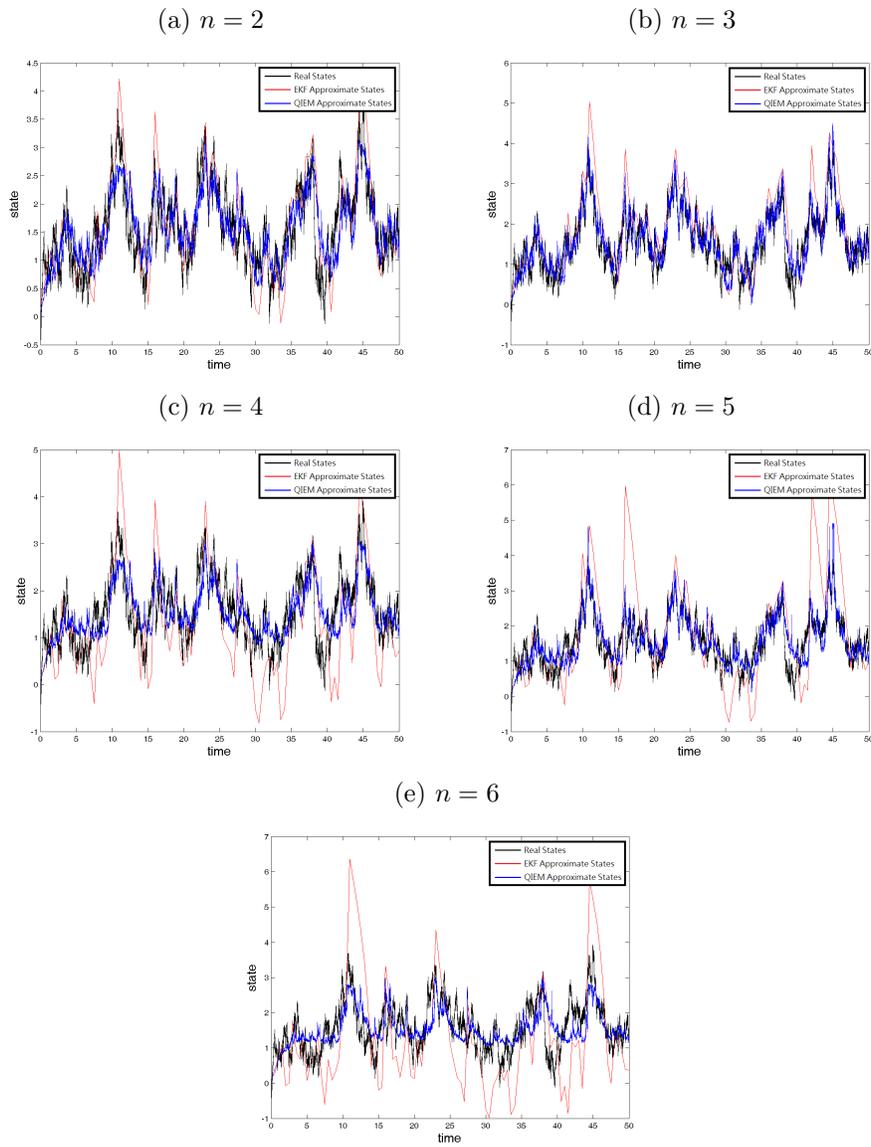


Figure 4: Comparison of real states and approximate states between EKF and QIEM with  $f(x) = \cos x$ ,  $h(x) = x^n$ ,  $n = 2, \dots, 6$ , respectively.

Table 1: The comparison of  $E_{RMS}$  between EKF and QIEM

Function Types	Methods	
	EKF	QIEM
quadratic $h(x) = x^2$	0.6364	0.4630
cubic $h(x) = x^3$	0.6677	0.3600
quartic $h(x) = x^4$	0.9298	0.5089
quintic $h(x) = x^5$	1.2140	0.4279
sextic $h(x) = x^6$	1.3314	0.5597

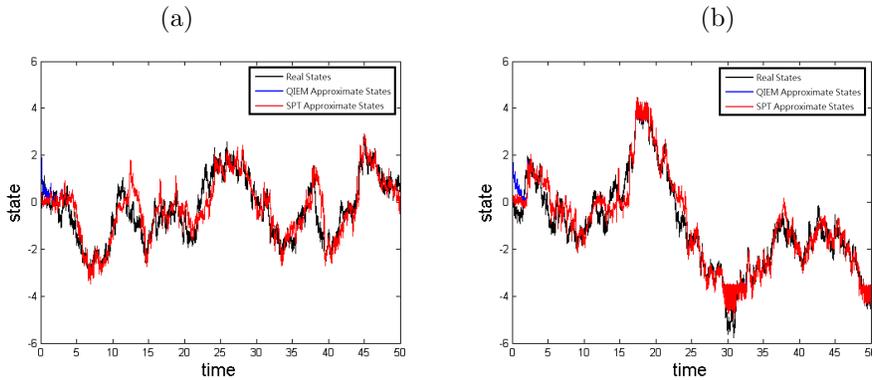


Figure 5: The numerical results for 1-D cubic sensor problem with initial  $u_0(s) = e^s$

### 5. Conclusions

In this paper, we propose efficient numerical algorithms, namely, IEM/QIEM/SPT for solving the nonlinear filtering problem by using the Yau-Yau method. QIEM is more efficient than IEM because the linear systems in QIEM is feasible by FFT-acceleration. SPT is an off-time process which saves a lot of computational cost. The solution of the initial-value problem of the Kolmogorov equations (2) forms a probability density function which must be nonnegative. We first show that the iterative matrix of the proposed schemes is an  $M$ -matrix whose inverse is a nonnegative operator and guarantees the nonnegativity of the solution in each iteration. Furthermore, we prove that the proposed numerical schemes converge to the initial-value problem (2) under some mild conditions. At the present stage, IEM/QIEM/SPT are capable for solving 1-D and 2-D nonlinear filtering problems efficiently and accurately. To make our algorithms feasible for solving the higher dimensional problems efficiently is our next goal and under investigation.

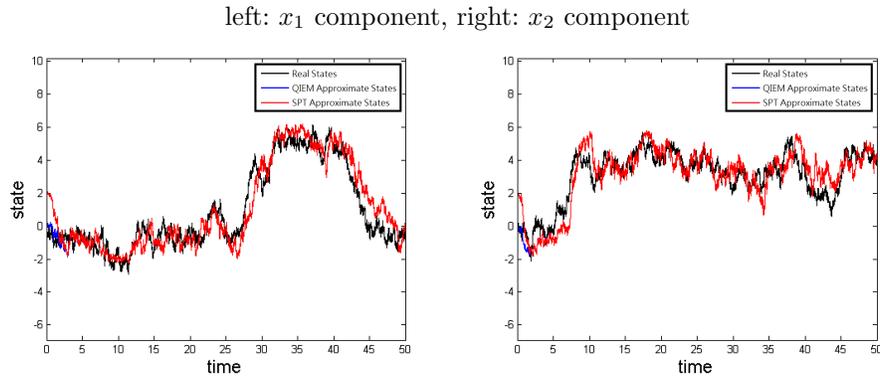


Figure 6: The numerical results for 2-D nonlinear filtering problem with initial  $u(s_1, s_2) = e^{-5(s_1^2 + s_2^2)}$

## 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. [MR2454694](#)
- [3] T. E. Duncan. Probability density for diffusion processes with applications to nonlinear filtering theory. *Ph.D. thesis Stanford University*, 1967.
- [4] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1 edition, 1994. [MR1288752](#)
- [5] A. H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, 1 edition, 1970.
- [6] R. E. Kalman and R. S. Bucy. New results in linear filtering and prediction theory. *Trans. ASME*, 83:95–108, 1961. [MR0234760](#)
- [7] K.-H. Kim, G.-I. Jee, and J.-H. Song. The stability of the adaptive two-stage extended Kalman filter. *International Conference on Control, Automation and Systems*, pages 1378–1383, 2008.
- [8] R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations*. Society for Industrial and Applied Mathematics, 1 edition, 2007. [MR2378550](#)

- [9] 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.
- [10] 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. [MR3106062](#)
- [11] 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. [MR3106057](#)
- [12] R. E. Mortensen. Optional control of continuous time stochastic systems. *Ph.D. thesis University of California Berkeley*, 1966.
- [13] G. G. Rigatos. *Modelling and Control for Intelligent Industrial Systems*. Springer-Verlag Berlin Heidelberg, 1 edition, 2011.
- [14] G. G. Rigatos. *Nonlinear Estimation and Applications to Industrial Systems Control*. Nova Science Publishers Inc, 1 edition, 2013.
- [15] 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.
- [16] E. A. Wan and R. van der Merwe. The unscented Kalman filter for nonlinear estimation. *The IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, 2000.
- [17] J. Wang, Q. Liang, K. Liang, and W. Shangguan. A new extended Kalman filter based carrier tracking loop. *3rd IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, pages 1181–1184, 2009.
- [18] C. Yan and S. S.-T. Yau. A new suboptimal filter and numerical solutions for the cubic sensor problem. *IEEE International Conference on Networking, Sensing and Control*, pages 351–356, 2006.
- [19] S. S.-T. Yau, R. Du, and L. Jia. Special solutions to some Kolmogorov equations arising from cubic sensor problems. *Communications in Information and Systems*, 7(2):195–206, 2007. [MR2344197](#)
- [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. [MR1734138](#)
- [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. [MR1809293](#)
- [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. [MR2373467](#)
- [24] M. Zakai. On the optimal filtering of diffusion processes. *Z. Wahrsch. Verw. Gebiete*, 11:230–243, 1969. [MR0242552](#)

MEI-HENG YUEH  
DEPARTMENT OF APPLIED MATHEMATICS  
NATIONAL CHIAO TUNG UNIVERSITY  
HSINCHU 300  
TAIWAN  
*E-mail address:* [mhyueh@math.nctu.edu.tw](mailto:mhyueh@math.nctu.edu.tw)

WEN-WEI LIN  
DEPARTMENT OF APPLIED MATHEMATICS  
NATIONAL CHIAO TUNG UNIVERSITY  
HSINCHU 300  
TAIWAN  
*E-mail address:* [wlin@math.nctu.edu.tw](mailto:wlin@math.nctu.edu.tw)

SHING-TUNG YAU  
DEPARTMENT OF MATHEMATICS  
HARVARD UNIVERSITY  
CAMBRIDGE, MA 02138  
USA  
*E-mail address:* [yau@math.harvard.edu](mailto:yau@math.harvard.edu)

RECEIVED JULY 21, 2014