

Solving high-dimensional nonlinear filtering problems using a tensor train decomposition method

Dedicated to Professor Thomas Kailath on the occasion of his 85th birthday

Sijing Li^a, Zhongjian Wang^a, Stephen S.T. Yau^{b,*}, Zhiwen Zhang^{a,*}

^a*Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong SAR, China.*

^b*Department of Mathematics, Tsinghua University, Beijing 100084, China.*

Abstract

In this paper, we propose an efficient numerical method to solve high-dimensional nonlinear filtering (NLF) problems. Specifically, we use the tensor train decomposition method to solve the forward Kolmogorov equation (FKE) arising from the NLF problem. Our method consists of offline and online stages. In the offline stage, we use the finite difference method to discretize the partial differential operators involved in the FKE and extract low-dimensional structures in the solution space using the tensor train decomposition method. In addition, we approximate the evolution of the FKE operator using the tensor train decomposition method. In the online stage using the pre-computed low-rank approximation tensors, we can quickly solve the FKE given new observation data. Therefore, we can solve the NLF problem in a real-time manner. Under some mild assumptions, we provide convergence analysis for the proposed method. Finally, we present numerical results to show the efficiency and accuracy of the proposed method in solving high-dimensional NLF problems.

AMS subject classification: 15A69, 35R60, 65M12, 60G35, 65M99.

Keywords: nonlinear filtering (NLF) problems; forward Kolmogorov equations (FKEs); Duncan-Mortensen-Zakai (DMZ) equation; tensor train decomposition method; convergence analysis; real-time algorithm.

1. Introduction

Nonlinear filtering (NLF) problem is originated from the problem of tracking and signal processing. The fundamental problem in the NLF is to give the instantaneous and accurate estimation of the states based on the noisy observations [14]. In this paper, we consider the signal based nonlinear filtering problems as follows,

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

*Corresponding author

Email addresses: `lsj17@hku.hk` (Sijing Li), `ariswang@connect.hku.hk` (Zhongjian Wang), `yau@tsinghua.edu.cn` (Stephen S.T. Yau), `zhangzw@hku.hk` (Zhiwen Zhang)

where $\mathbf{x}_t \in R^d$ is the states of the system at time t and $\mathbf{y}_t \in R^m$ is the observations at time t , and \mathbf{v}_t and \mathbf{w}_t are vector Brownian motion processes. Some growth conditions on \mathbf{f} , \mathbf{g} and \mathbf{h} are required to guarantee the existence and well-posedness of the NLF problems, which will be discussed later.

Particle filter method is the most popular method to solve (1); see e.g. [1, 11, 2] and references therein. However, the main drawback of the particle filter method is that it is hard to be implemented in a real-time manner due to its nature of the Monte Carlo simulation. In practice, the real-time manner means the running time of the numerical integrator in solving the state equation for \mathbf{x}_t is much less than the time between any two observations of \mathbf{y}_t .

Alternatively, one can solve the Duncan-Mortensen-Zakai (DMZ) equation, also known as Zakai equation, to study the NLF problems [9, 21, 30]. The DMZ equation computed the unnormalized conditional density function of the states \mathbf{x}_t , which provides a powerful tool to study the NLF problem since one can estimate the statistical quantities of the state \mathbf{x}_t based on the DMZ solution. In general, one cannot solve the DMZ equation analytically. Many efforts have been made to develop efficient numerical methods; see [22, 13, 12, 4, 18] and the references therein.

The DMZ equation allows one to study the statistical quantities of the states \mathbf{x}_t . In practice, however, one can only get one realization of the states \mathbf{x}_t (instead of thousands of repeated experiments), which motivates researchers to develop robust methods in solving the DMZ equation. Namely, the robust method should not be sensitive to the given observation paths. A novel algorithm was proposed to solve the path-wise robust DMZ equation [28]. In this approach, for each realization of the observation process denoted by \mathbf{y}_t , one can make an invertible exponential transformation and transform the DMZ equation into a deterministic partial differential equation (PDE) with stochastic coefficient. Several efficient numerical methods were developed along this direction; see [19, 20, 27], which can be efficient when the dimension of the NLF problems is small. However, it becomes expensive as the dimension of the NLF problems increases. Therefore, it is still very challenging to solve high-dimensional NLF problems in a real-time manner.

In this paper, we propose to use the tensor train decomposition method to solve the high-dimensional FKEs. Our method consists of offline and online stages. In the offline stage, we discrete the PDE operators involved in the FKEs and approximate them using the tensor train method. Moreover, we approximate the evolution of the FKE operator using the tensor train method. In the online stage, we can quickly solve the FKE given new observation data using the pre-computed low-rank approximation tensors. By exploring the low-dimensional structures in the solution space of the FKE, we can solve the NLF problem in a real-time manner. Under some mild assumptions, we provide convergence analysis for the proposed method. Finally, we present two numerical experiments to show the efficiency and accuracy of the proposed method. We find that the tensor train method is scalable in solving the FKE. Thus, we can solve the NLF problem in a real-time manner.

The rest of the paper is organized as follows. In Section 2, we give a brief introduction of the NLF problem and DMZ equation. In Section 3, we introduce the basic idea of the tensor train decomposition method. In Section 4, we propose our fast method to compute the high-dimensional FKEs. Some convergence analysis of the proposed method will also be

discussed in Section 5. In Section 6, we present numerical results to demonstrate the accuracy and efficiency of our method. Concluding remarks are made in Section 7.

2. Some basic results of the NLF problems

In this section, we shall introduce some basic results of the NLF problems. To start with, we consider the signal based model as follows,

$$\begin{cases} d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)dt + \mathbf{g}(\mathbf{x}_t, t)d\mathbf{v}_t, \\ d\mathbf{y}_t = \mathbf{h}(\mathbf{x}_t, t)dt + d\mathbf{w}_t, \end{cases} \quad (2)$$

where $\mathbf{x}_t \in R^d$ is a vector of the states of the system at time t , the initial state \mathbf{x}_0 satisfying an initial distribution, $\mathbf{y}_t \in R^m$ is a vector of the observations at time t with \mathbf{y}_0 , and \mathbf{v}_t and \mathbf{w}_t are vector Brownian motion processes with covariance matrices $E[d\mathbf{v}_t d\mathbf{v}_t^T] = \mathbf{Q}(t)dt \in R^{d \times d}$ and $E[d\mathbf{w}_t d\mathbf{w}_t^T] = \mathbf{S}(t)dt \in R^{m \times m}$, respectively. Moreover, \mathbf{x}_0 , $d\mathbf{w}_t$ and $d\mathbf{v}_t$ are assumed to be independent. Some growth conditions on \mathbf{f} and \mathbf{h} are required to guarantee the existence and uniqueness of the pathwise-robust DMZ equation [28]. In this paper, \mathbf{f} , \mathbf{h} , and \mathbf{g} are C^2 in spatial variable and C^1 in the temporal variable

The most popular method so far to solve (2) is the particle filter, see [1, 11, 2] and references therein. However, the main drawback of the particle filter method is that it is hard to be implemented in a real-time manner due to its nature of the Monte Carlo simulation.

The DMZ equation or Zakai equation [9, 21, 30] asserts that the unnormalized conditional density function of the states \mathbf{x}_t , denoted by $\sigma(\mathbf{x}, t)$, satisfies the following stochastic partial differential equation (SPDE):

$$\begin{cases} d\sigma(\mathbf{x}, t) = \mathcal{L}\sigma(\mathbf{x}, t)dt + \sigma(\mathbf{x}, t)\mathbf{h}^T(\mathbf{x}, t)\mathbf{S}^{-1}d\mathbf{y}_t, \\ \sigma(\mathbf{x}, 0) = \sigma_0(\mathbf{x}), \end{cases} \quad (3)$$

where $\sigma_0(\mathbf{x})$ is the density of the initial states \mathbf{x}_0 , and

$$\mathcal{L}(\cdot) := \frac{1}{2} \sum_{i,j=1}^d \frac{\partial^2}{\partial x_i \partial x_j} ((gQg^T)_{ij} \cdot) - \sum_{i=1}^d \frac{\partial (f_i \cdot)}{\partial x_i} \quad (4)$$

The DMZ equation laid down a solid foundation to study the NLF problem. However, one cannot solve the DMZ equation analytically in general. Many efforts have been made to develop efficient numerical methods. One of the commonly used method is the splitting-up method originated from the Trotter product formula, which was first introduced in [4] and has been extensively studied later, see [22, 13, 12]. In [18], the so-called S^3 algorithm was developed based on the Wiener chaos expansion. By separating the computations involving the observations from those dealing only with the system parameters, this approach gives rise to a new numerical scheme for NLF problems. However, the limitation of their method is that the drifting term f and the observation term h in (2) should be bounded.

To overcome this restriction, Yau and Yau [28] developed a novel algorithm to solve the path-wise robust DMZ equation. Specifically, for each realization of the observation process denoted by y_t , they make an invertible exponential transformation

$$\sigma(\mathbf{x}, t) = \exp(\mathbf{h}^T(\mathbf{x}, t)\mathbf{S}^{-1}(t)\mathbf{y}_t)u(\mathbf{x}, t), \quad (5)$$

and transform the DMZ equation (3) into a deterministic partial differential equation (PDE) with stochastic coefficient

$$\begin{cases} \frac{\partial}{\partial t}u(\mathbf{x}, t) + \frac{\partial}{\partial t}(\mathbf{h}^T \mathbf{S}^{-1})\mathbf{y}_t u(\mathbf{x}, t) = \\ \exp(-\mathbf{h}^T(\mathbf{x}, t)\mathbf{S}^{-1}(t)\mathbf{y}_t)(\mathcal{L} - \frac{1}{2}\mathbf{h}^T \mathbf{S}^{-1}\mathbf{h})(\exp(-\mathbf{h}^T(\mathbf{x}, t)\mathbf{S}^{-1}(t)\mathbf{y}_t)u(\mathbf{x}, t)), \\ u(\mathbf{x}, 0) = \sigma_0(\mathbf{x}). \end{cases} \quad (6)$$

Equation (6) is called the pathwise robust DMZ equation [18, 28]. Compared with the DMZ equation (3), the pathwise robust DMZ equation (6) is easier to solve, since the stochastic term has been transformed into the coefficients.

The existence and uniqueness of (6) has been investigated by many researchers. The well-posedness is guaranteed when the drift term $\mathbf{f} \in C^1$ and the observation term $\mathbf{h} \in C^2$ are bounded in [26]. Later on, similar results can be obtained under weaker conditions. For instance, the well-posedness results on the pathwise-robust DMZ equation with a class of unbounded coefficients were obtained in [3, 10], but the results were for one-dimensional case. In [28], the third author of this paper and his collaborator established the well-posedness result under the condition that \mathbf{f} and \mathbf{g} have at most linear growth. In [19], a well-posedness result was obtained for time-dependent pathwise-robust DMZ equation under some mild growth conditions on \mathbf{f} and \mathbf{h} .

Let us assume that the observation time sequences $0 = t_0 < t_1 < \dots < t_{N_t} = T$ are given. In each time interval $t_{j-1} \leq t < t_j$, one freezes the stochastic coefficient \mathbf{y}_t to be $\mathbf{y}_{t_{j-1}}$ in Eq.(6) and makes the exponential transformation

$$\tilde{u}_j(\mathbf{x}, t) = \exp(\mathbf{h}^T(\mathbf{x}, t)\mathbf{S}^{-1}(t)\mathbf{y}_{t_{j-1}})u(\mathbf{x}, t). \quad (7)$$

It is easy to deduce that \tilde{u}_j satisfies the FKE

$$\frac{\partial}{\partial t}\tilde{u}_j(\mathbf{x}, t) = (\mathcal{L} - \frac{1}{2}\mathbf{h}^T \mathbf{S}^{-1}\mathbf{h})\tilde{u}_j(\mathbf{x}, t), \quad (8)$$

where the operator \mathcal{L} is defined in (4). In [20], Luo and Yau investigated the Hermite spectral method to numerically solve the 1D FKE (8) and analyzed the convergence rate of the proposed method. In their algorithm, the main idea is to shift part of the heavy computations off-line, so that only computations involved observations are performed on-line and synchronized with off-line data. The numerical method based on the Hermite polynomial approximation is efficient though, it is extremely hard to extend to solve high-dimensional FKEs in the real-time manner, since the number of the Hermite polynomial basis functions grows fast for high-dimensional problems. Namely, it suffers from the curse of dimensionality.

In a very recent result [27], we proposed to use the proper orthogonal decomposition (POD) method to numerically solve the 2D FKE. By extracting the low-dimensional structures in the solution space of the FKE and building POD basis, our method provides considerable savings over the Hermite polynomial approximation method that was used in [20]. The POD method helps us alleviate the curse of dimensionality to a certain extent though, it is still very challenging to solve high-dimensional NFL problems. The reason is that in the POD

method one needs solution snapshots to construct POD basis. However, to compute solution snapshots for high-dimensional NLF problems is extremely expensive. We shall address this challenge by using the Tensor Train decomposition method in this paper.

3. The Tensor Train decomposition method

We shall introduce the tensor train (TT) decomposition method for approximating solutions of high-dimensional NLF problems. Let us assume the dimension of the NLF problem is d . For any fixed time t , if we discretize the solution $u(\mathbf{x}, t)$, $\mathbf{x} \in R^d$ of the FKE (8) using conventional numerical methods, such as finite difference method, we obtain a d -dimensional $n_1 \times n_2 \times \cdots \times n_d$ tensor $\mathbf{U}(i_1, i_2, \cdots, i_d)$, which is a multidimensional array. The number of unknowns in this representation grows fast as d increases and is subject to the curse of dimensionality. To attack this challenge, one should extract potential low-dimensional structures in the tensor and approximate the tensor in a certain data-sparse way.

The TT decomposition method is an efficient method for tensor approximation [24]. A brief introduction of the TT-format is given below. If a d -dimensional $n_1 \times n_2 \times \cdots \times n_d$ tensor $\mathbf{U}(i_1, i_2, \cdots, i_d)$ can be written as the element-wise form

$$\mathbf{U}(i_1, i_2, \cdots, i_d) = \mathbf{G}_1(i_1)\mathbf{G}_2(i_2) \cdots \mathbf{G}_d(i_d), \quad 1 \leq i_k \leq n_k, \quad (9)$$

where $\mathbf{G}_k(i_k)$ is a $r_{k-1} \times r_k$ matrix for each fixed i_k , $1 \leq k \leq d$ and $r_0 = r_d = 1$. We call the tensor \mathbf{U} is in the TT-format, if it is represented in the form (9). Furthermore, each element \mathbf{G}_k can be regarded as an 3-dimensional tensor of the size $r_{k-1} \times n_k \times r_k$. In the representation (9), $\mathbf{G}_1, \mathbf{G}_2, \cdots, \mathbf{G}_d$ are called the cores of the TT-format tensor \mathbf{U} , numbers r_k are called TT-ranks, and numbers n_1, n_2, \cdots, n_d are called mode sizes. With these definitions, the representation (9) can be rewritten as

$$\mathbf{U}(i_1, i_2, \cdots, i_d) = \sum_{\alpha_1, \cdots, \alpha_{d-1}} \mathbf{G}_1(\alpha_0, i_1, \alpha_1)\mathbf{G}_2(\alpha_1, i_2, \alpha_2) \cdots \mathbf{G}_d(\alpha_{d-1}, i_d, \alpha_d) \quad (10)$$

where $\alpha_0 = \alpha_d = 1, 1 \leq \alpha_k \leq r_k$ for $1 \leq k \leq d-1$. In practice, one only needs to store all the cores \mathbf{G}_k in the TT-format, in order to save a tensor. Thus, if all the TT-ranks r_k are bounded by a constant r and the mode sizes n_k are bounded by N , the storage of the d -dimensional tensor \mathbf{U} is $\mathcal{O}(dNr^2)$ in the TT-format. Recall that the storage of the tensor \mathbf{U} is about $\mathcal{O}(N^d)$, if no approximation is used.

To further reduce the storage of the TT-format, a quantized tensor train (QTT) format was introduced in [15, 8, 23]. The QTT format is derived by introducing virtual dimensions along each real dimension of a tensor. Specifically, suppose each one-dimensional size of the tensor \mathbf{U} is a power of 2, i.e. $n_1 = n_2 = \cdots = n_d = 2^L$. The d -dimensional tensor \mathbf{U} can be reshaped to a D -dimensional tensor with $D = dL$, while each mode size is equal to 2. The QTT-format is the TT-format of the reshaped tensor, which has a larger number of dimension but much smaller mode sizes (here is 2) than the TT-format. The concepts of cores, QTT-ranks and mode sizes (all are equal to 2) of QTT-format are defined similarly as the TT-format. The storage of the QTT-format is further reduced to $\mathcal{O}(d \log_2(N)r^2)$.

One can also simply formulate the TT-format of d -dimensional matrices [15][23]. The TT-format of a d -dimensional $(m_1 \times \cdots \times m_d) \times (n_1 \times \cdots \times n_d)$ matrix \mathbf{B} can be written as

$$\mathbf{B}(i_1, \dots, i_d, j_1, \dots, j_d) = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_{d-1}=1}^{r_{d-1}} \mathbf{G}_1(1, i_1, j_1, \alpha_1) \mathbf{G}_2(\alpha_1, i_2, j_2, \alpha_2) \cdots \mathbf{G}_d(\alpha_{d-1}, i_d, j_d, 1), \quad (11)$$

where the index 1 in \mathbf{G}_1 and \mathbf{G}_d is due to the TT-ranks $\alpha_0 = \alpha_d = 1$. The definitions of cores, ranks and mode sizes are similar as tensors. The QTT representation of matrices will be used in this paper.

Simple calculations show that in the TT-format the computational complexity of addition (together with TT-rounding after addition) is $\mathcal{O}(dNr^3)$, matrix-by-vector multiplication is $\mathcal{O}(dN^2r^4)$, and Hadamard product is $\mathcal{O}(dNr^4)$. Therefore, the TT/QTT-format allows lower complexity of algebraic operation than dense matrix or tensor form. In practice, especially in solving time-dependent problems, one needs to apply the TT-rounding procedure in the computation. The computational complexity of the TT-rounding is $\mathcal{O}(dNr^3)$. The purpose of TT-rounding is to decrease TT/QTT-rank of a matrix or tensor already in the TT/QTT-format while preserving a given accuracy ϵ . The QTT-format can further reduce the factor N of complexity to $\log_2(N)$.

4. The fast algorithm and its implementation

In this section, we present the fast algorithm to solve the NLF problem. Let us assume that the observation time sequences $0 = t_0 < t_1 < \cdots < t_{N_t} = T$ are given. But the observation data $\{\mathbf{y}_{t_j}\}$ at each observation time t_j , $j = 0, \dots, N_t$ are unknown until the on-line experiment runs. For simplicity, we assume $t_j - t_{j-1} = \Delta T$. We shall study how to solve the FKE (8) within each time interval $[t_{j-1}, t_j]$ and compute the exponential transformation (7) using the QTT-format. If we use an explicit scheme to discretize the time derivative in the FKE (8), we get an semi-discrete scheme as follows,

$$\tilde{u}_j^n(\mathbf{x}) = \left\{ \tau(\mathcal{L} - \frac{1}{2}\mathbf{h}^T \mathbf{S}^{-1} \mathbf{h}) + I \right\} \tilde{u}_j^{n-1}(\mathbf{x}), \quad n = 1, \dots, \frac{\Delta T}{\tau}, \quad (12)$$

where τ is the time step in discretizing the FKE (8) and ΔT is the time interval between two sequential observations.

Our algorithm consists of an offline procedure and an online procedure. In the offline procedure, we compute the matrices of spatial discretization of operators in (12) and convert them into the QTT-format in advance, which will reduce the computational time in the online procedure. In the online procedure, we solving the FKE and update the solution with new observation data. We shall show that the online procedure can be finished in a real time manner since all the computations are done in the QTT-format.

4.1. Spatial discretization and low-rank approximation

We shall discuss how to discretize the FKE (8) and represent the spatial discretization in the QTT-format. To simplify the notations and illustrate the main idea of our algorithm,

we choose a squared domain and uniform grid. Specifically, we take a large $a > 0$ and let $\Omega = [-a, a]^d$ denote the physical domain of the FKE. A uniform grid is set on each dimension with $N = 2^L$ points and mesh size $h = \frac{2a}{(N-1)}$. We use $x_k(i_k)$, $i_k = 1, \dots, N$ to record the coordinates of grid points on k -th dimension.

First, the Laplace operator in (8) is discretized by using a finite difference (FD) scheme. The resulting matrix is of the form

$$\Delta_d = \Delta_1 \otimes \cdots \otimes \mathbf{I} + \cdots + \mathbf{I} \otimes \cdots \otimes \Delta_1, \quad (13)$$

where

$$\Delta_1 = \frac{1}{h^2} \text{tridiag}(1, -2, 1),$$

and \mathbf{I} is an $N \times N$ identity matrix. In particular, when $d = 3$ the Laplace operator has the form

$$\Delta_3 = \Delta_1 \otimes \mathbf{I} \otimes \mathbf{I} + \mathbf{I} \otimes \Delta_1 \otimes \mathbf{I} + \mathbf{I} \otimes \mathbf{I} \otimes \Delta_1. \quad (14)$$

The QTT-format of matrix Δ_d has a low-rank representation that is bounded by 4 (see Corollary 5.3 of [15]).

Second, the convection operator $\sum_{i=1}^d \frac{\partial(f_i \cdot)}{\partial x_i}$ in (8) (see also Eq.(4)) is discretized by using a central difference scheme. Thus, the corresponding d -dimensional matrix has the form

$$\mathbf{C}_d = (\mathbf{C} \otimes \mathbf{I} \otimes \cdots \otimes \mathbf{I})\mathbf{F}_1 + (\mathbf{I} \otimes \mathbf{C} \otimes \cdots \otimes \mathbf{I})\mathbf{F}_2 + \cdots + (\mathbf{I} \otimes \mathbf{I} \otimes \cdots \otimes \mathbf{C})\mathbf{F}_d, \quad (15)$$

where \mathbf{F}_k 's are diagonal matrices associated with the diagonalization of tensor discretization of the drift functions f_k , i.e.,

$$\mathbf{F}_k(i_1, i_2, \dots, i_d, i_1, i_2, \dots, i_d) = f_k(x_1(i_1), x_2(i_2), \dots, x_d(i_d)),$$

for all $1 \leq k \leq d$, and \mathbf{C} is an one-dimensional central difference operator,

$$\mathbf{C} = \frac{1}{h} \text{tridiag}\left(-\frac{1}{2}, 0, \frac{1}{2}\right).$$

Under certain conditions for the drift terms f_k , the QTT-format of matrix \mathbf{C}_d has a bounded low-rank representation. We summary the result into the following lemma and the proof can be found in [8].

Proposition 4.1. *Suppose that the QTT-ranks of the functions f_k on a tensor grid are bounded by r . Then, the QTT-rank of the matrix \mathbf{C}_d in (15) is bounded by $5dr$.*

Although an exact TT-decomposition of any tensor is feasible [24], it rarely has a low-rank structure. Therefore, one should apply TT-rounding procedures in order to decrease the TT/QTT-ranks while preserving a given accuracy ϵ . Let us consider the drift terms f_k as an example. In order to construct the QTT-format of functions f_k with low ranks, one can use the TT-SVD algorithm [24]. QTT-ranks of these QTT-format tensors are guaranteed to be smaller than a prescribed approximation error ϵ in the sense of Frobenius norm [24].

Proposition 4.2 (Theorem 2.2 of [25]). *For any tensor \mathbf{A} with size $n_1 \times n_k \times \cdots \times n_d$, there exists a tensor \mathbf{B} in the TT-format with TT-ranks r_k such that*

$$\|\mathbf{A} - \mathbf{B}\|_F \leq \sqrt{\sum_{k=1}^{d-1} \epsilon_k^2},$$

where ϵ_k is the distance from \mathbf{A}_k to its best rank- r_k approximation in the Frobenius norm,

$$\epsilon_k = \min_{\text{rank } \mathbf{C} \leq r_k} \|\mathbf{A}_k - \mathbf{C}\|_F,$$

where \mathbf{A}_k is the k -th unfolding matrix of tensor \mathbf{A}

$$\mathbf{A}_k = \text{reshape} \left(\mathbf{A}, \prod_{s=1}^k n_s, \prod_{s=k+1}^d n_s \right).$$

Remark 4.1. The Lemma 4.2 allows us to control the accuracy and TT/QTT-ranks, when we compute the approximation of any tensor in the TT/QTT-format.

Finally, the approximation of the function $\mathbf{h}^T \mathbf{S}^{-1} \mathbf{h}$ in the FKE (8) in the QTT-format can be obtained using the same approach as f_k in the convection operator. Specifically, we discretize the function $\mathbf{h}^T \mathbf{S}^{-1} \mathbf{h}$ on the spatial tensor grid and diagonalize it to a matrix denoted by \mathbf{Q}_d , i.e.,

$$\mathbf{Q}_d(i_1, i_2, \dots, i_d, i_1, i_2, \dots, i_d) = (\mathbf{h}^T \mathbf{S}^{-1} \mathbf{h})(x_1(i_1), x_2(i_2), \dots, x_d(i_d)).$$

Then, we approximate it by a low-rank QTT-format using the TT-SVD algorithm.

4.2. The offline procedure

In the offline procedure, we first assemble the operators involved in the FKE, including the Laplace operator (13), the convection operator (15) and the multiplication operator \mathbf{Q}_d associated with the function $\mathbf{h}^T \mathbf{S}^{-1} \mathbf{h}$, into a tensor \mathbf{A} , i.e.

$$\mathbf{A} = \frac{1}{2} \Delta_d - \mathbf{C}_d - \frac{1}{2} \mathbf{Q}_d \quad (16)$$

In this paper, we assume the drift and observation functions are time-independent. Thus, the semi-discrete scheme (12) becomes

$$\widehat{U}_{\mathbf{l},j}^n = (\widehat{\tau \mathbf{A} + \mathbf{I}}) \widehat{U}_{\mathbf{l},j}^{n-1}, \quad n = 1, \dots, \frac{\Delta T}{\tau}, \quad (17)$$

where $\widehat{U}_{\mathbf{l},j}^n$ is the QTT-format solution of $\tilde{u}(\mathbf{x}_1, t_{j-1} + n\tau)$, $\mathbf{l} \in \{1, 2, \dots, N\}^d$, and $(\widehat{\tau \mathbf{A} + \mathbf{I}})$ is the QTT-format of the tensor $(\tau \mathbf{A} + \mathbf{I})$.

Recall that the discretizations of the Laplace operator, the convection operator and the multiplication operator associated with the function $\mathbf{h}^T \mathbf{S}^{-1} \mathbf{h}$ all have low-rank approximations. Moreover, addition of matrices or tensors in the QTT-format only causes addition of QTT-ranks. Therefore, the tensor \mathbf{A} has a low rank QTT-format approximation with a given maximal QTT-rank r or with a certain given precision ϵ in the sense of Frobenius norm.

Notice that in the NLF problem, there will be no observation available during the time period with length ΔT . Thus, we directly compute the tensor $(\tau\mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}$ and approximate it in the QTT-format. Then, we rewrite the scheme (17) as

$$\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}} = \widehat{(\tau\mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}} \widehat{U}_{1,j}^0. \quad (18)$$

where $\widehat{(\tau\mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}}$ is the QTT-format of the tensor $(\tau\mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}$. Exact addition of $\tau\mathbf{A}$ and \mathbf{I} in the QTT format only increases the rank by one. However, exact multiplication of matrices in the QTT-format will lead to a significant growth of QTT-ranks. In our algorithm, we apply TT-rounding to control the growth of the QTT-rank caused by matrix-matrix multiplication, which can be easily achieved and maintain accuracy [24, 8].

4.3. Online procedure

In this section, we shall demonstrate that using the tensor train decomposition method and the precomputed time integration results we can achieve fast computing in the online stage.

At first, we set an initial probability density function according to initial state \mathbf{x}_0 , and solve the FKE (8) with such initial condition. At each observing time t_j , when a new observation \mathbf{y}_{t_j} arrives, we do the exponential transformation (7) to get the initial condition of the FKE (8). We then solve the FKE (8) by our algorithm (18). All of these operations are done in QTT-format, thus we need to do TT-rounding operation after both exponential transformation and solving the FKE (8).

Proposition 4.3. *Suppose the QTT-ranks of all functions required in online procedure on a tensor grid, including $u(\mathbf{x}, t_j)$ and $\exp[\mathbf{h}^T(\mathbf{x}, t_j)\mathbf{S}^{-1}(t_j)(\mathbf{y}_{t_j} - \mathbf{y}_{t_{j-1}})]$, are bounded by r . The accuracy ϵ of TT-rounding is properly specified to ensure QTT-ranks of $u(\mathbf{x}, t)$ are also bounded by r after any TT-rounding procedure. Then, the complexity of the online procedure within each time interval $[t_{j-1}, t_j]$ is $\mathcal{O}(N^d r^2 + d \log_2(N) r^6)$, where N is the grid number on each dimension.*

Proof. The complexity of constructing the QTT-format of $\exp[\mathbf{h}^T(\mathbf{x}, t_j)\mathbf{S}^{-1}(t_j)(\mathbf{y}_{t_j} - \mathbf{y}_{t_{j-1}})]$ from a full multidimensional array is $\mathcal{O}(N^d r^2)$ by Theorem 2.1 in [23]. The exponential transformation is essentially a Hadamard product in the QTT-format whose complexity is $\mathcal{O}(d \log_2(N) r^4)$ [24]. Solving the FKE (8) is practically a matrix-vector multiplication (18) in the QTT-format whose complexity is $\mathcal{O}(d \log_2(N) r^4)$ [24]. Requirement of TT-rounding through standard TT-SVD algorithm is $\mathcal{O}(d \log_2(N) r^6)$ [24]. \square

Notice that the total degree of freedom is N^d in the spatial discretization. Prop.4.3 shows that the QTT method is very efficient in the online procedure in solve the NLF problem. More details will be represented in Section 6. We observe that the maximal QTT-rank r has very slow growth with respect to N (see Table 1–Table 4), which allows us to solve high-dimensional NLF problems in a real time manner.

4.4. The complete algorithm of the NLF problem

In this subsection, we give the complete algorithm of the NLF problem. The off- and on-line computing stages in our algorithm are summarized in the Algorithm 1 and Algorithm 2, respectively. The performance of our method will be demonstrated in Section 6.

Algorithm 1 Offline computing

- 1: Compute matrices of spatial discretization of operators mentioned in the Section 4.1, including the Laplace operator, i.e., Eq.(13), the convection operator, i.e., Eq.(15), and the multiplication operator \mathbf{Q}_d associated with the function $\mathbf{h}^T \mathbf{S}^{-1} \mathbf{h}$.
 - 2: Convert these matrices into the QTT-format.
 - 3: Compute the addition of operator matrices in the QTT-format by taking into account the time stepping τ , i.e. compute $\tau \mathbf{A} + \mathbf{I}$ in Eq.(17).
 - 4: Compute the power of the tensor $\tau \mathbf{A} + \mathbf{I}$ in the QTT-format, i.e. compute $(\widehat{\tau \mathbf{A} + \mathbf{I}})^{\frac{\Delta T}{\tau}}$ in Eq.(18).
-

Algorithm 2 Online computing

- 1: Set up the initial data $u(\mathbf{x}, 0) = \sigma_0(\mathbf{x})$ of the FKE according to the distribution of the initial state \mathbf{x}_0 , convert $u(\mathbf{x}, 0)$ into a QTT-format, and apply the propagator operator (18) to get the predicted solution at time t_1 , denoted by $\widehat{U}_{1,1}^{\frac{\Delta T}{\tau}}$.
- 2: **for** $j = 1 \rightarrow N_t - 1$ **do**
- 3: Convert the term $\exp[\mathbf{h}^T(\mathbf{x}, t_j) \mathbf{S}^{-1}(t_j)(\mathbf{y}_{t_j} - \mathbf{y}_{t_{j-1}})]$ into the QTT-format.
- 4: Assimilate the new observation data \mathbf{y}_{t_j} into the predicted solution $\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}}$ using a QTT-format Hadamard product:

$$\widehat{U}_{1,j+1}^0 = \exp[\mathbf{h}^T(\mathbf{x}, t_j) \widehat{\mathbf{S}^{-1}(t_j)}(\mathbf{y}_{t_j} - \mathbf{y}_{t_{j-1}})] \widehat{U}_{1,j}^{\frac{\Delta T}{\tau}}.$$

- 5: Compute the predicted solution at time t_{j+1} using a matrix-vector multiplication in the QTT-format:

$$\widehat{U}_{1,j+1}^{\frac{\Delta T}{\tau}} = (\widehat{\tau \mathbf{A} + \mathbf{I}})^{\frac{\Delta T}{\tau}} \widehat{U}_{1,j+1}^0.$$

- 6: Calculate related statistics of prediction by using $\widehat{U}_{1,j+1}^{\frac{\Delta T}{\tau}}$ as the unnormalized density function at time t_{j+1} .
 - 7: **end for**
-

5. Convergence analysis

In this section, we shall study the convergence of the numerical solution obtained by our method to the exact solution. For simplicity of notations in the analysis, we assume $\mathbf{S} = I_d$. Note that the proof is straightforward if \mathbf{S} is a general covariance matrix.

5.1. Some assumptions and propositions

Before proceeding to the main analysis, let us first introduce some assumptions as follows.

[Asm.1] The following term is bounded in $\mathcal{R}^d \times [0, T]$, i.e.,

$$-\frac{1}{2} \mathbf{h}^T \mathbf{h} - \frac{1}{2} \Delta K - \mathbf{f} \cdot \nabla K + \frac{1}{2} |\nabla K|^2 + |\mathbf{f} - \nabla K| \leq c_1, \quad \forall (\mathbf{x}, t) \in \mathcal{R}^d \times [0, T], \quad (19)$$

where $K = \mathbf{h}^T \mathbf{y}_t$, c_1 is a constant possibly depending on T .

[Asm.2] The drift function \mathbf{f} is bounded in a bounded domain Ω , i.e. $\sup |f_i(\mathbf{x})| \leq C_f < \infty, \forall \mathbf{x} \in \Omega, \quad i = 1, 2, \dots, d$ and Lipschitz continuous, i.e. $|f_i(\mathbf{x}_1) - f_i(\mathbf{x}_2)| \leq L_f |\mathbf{x}_1 - \mathbf{x}_2|, \forall \mathbf{x}_1, \mathbf{x}_2 \in \Omega, \quad i = 1, 2, \dots, d$, where L_f is the Lipschitz constant.

[Asm.3] The observation function \mathbf{h} is bounded in a bounded domain Ω , i.e. $\sup |h_i(\mathbf{x})| \leq C_h < \infty, \forall \mathbf{x} \in \Omega, \quad i = 1, 2, \dots, m$.

[Asm.4] The observation series $K = \mathbf{h}^T \mathbf{y}_t$ is bounded in a bounded domain Ω on the observation time sequence $0 = t_0 < t_1 < \dots < t_{N_t} = T$, i.e.

$$|2K| \leq c_2, \quad \forall (\mathbf{x}, t) \in \Omega \times \{t_0, t_1, \dots, t_{N_t}\}. \quad (20)$$

After introducing necessary assumptions, we are in the position to proceed the convergence analysis. When the condition (19) in Asm.1 is satisfied, one can choose a bounded domain Ω large enough to capture almost all the density of the DMZ equation (3), since (3) is essentially a parabolic-type PDE. Thus, we can restrict the DMZ equation (3) on the bounded domain Ω .

Let $u(\mathbf{x}, t)$ be the solution of the DMZ equation (3) restricted on $\Omega \times [0, T]$ satisfying

$$\begin{cases} \frac{\partial u}{\partial t}(\mathbf{x}, t) = \frac{1}{2} \Delta u(\mathbf{x}, t) + \mathbf{F}(\mathbf{x}, t) \cdot \nabla u(\mathbf{x}, t) + J(\mathbf{x}, t)u(\mathbf{x}, t), \\ u(\mathbf{x}, 0) = \sigma_0(\mathbf{x}), \\ u(\mathbf{x}, t)|_{\partial\Omega} = 0, \end{cases} \quad (21)$$

where $\mathbf{F} = -\mathbf{f} + \nabla K$, $K = \mathbf{h}^T \mathbf{y}_t$, and $J = -\operatorname{div} \mathbf{f} - \frac{1}{2} \mathbf{h}^T \mathbf{h} + \frac{1}{2} \Delta K - \mathbf{f} \cdot \nabla K + \frac{1}{2} |\nabla K|^2$.

Let $\mathcal{P}_{N_t} = \{0 = t_0 < t_1 < \dots < t_{N_t} = T\}$ be a partition of $[0, T]$, where $t_j = \frac{jT}{N_t}$, $j = 0, \dots, N_t$. Let $u_j(\mathbf{x}, t)$ be the solution of the following equation defined on $\Omega \times [t_{j-1}, t_j]$,

$$\begin{cases} \frac{\partial u_j}{\partial t}(\mathbf{x}, t) = \frac{1}{2} \Delta u_j(\mathbf{x}, t) + \mathbf{F}(\mathbf{x}, t_{j-1}) \cdot \nabla u_j(\mathbf{x}, t) + J(\mathbf{x}, t_{j-1})u_j(\mathbf{x}, t), \\ u_j(\mathbf{x}, t_{j-1}) = u_{j-1}(\mathbf{x}, t_{j-1}), \\ u_j(\mathbf{x}, t)|_{\partial\Omega} = 0, \end{cases} \quad (22)$$

where we use the convection $u_0(\mathbf{x}, t) = \sigma(x)$. Then, the restriction of the solution $u(\mathbf{x}, t)$ of (21) on each domain $\Omega \times [t_{j-1}, t_j]$ can be approximated by the solution $u_j(\mathbf{x}, t)$ of (22). Specifically, we have the following error estimate.

Proposition 5.1 (Theorem C of [28]). *Let Ω be a bounded domain in \mathcal{R}^d . Let $\mathbf{F} : \Omega \times [0, T] \rightarrow \mathcal{R}^d$ be a family of vector fields that are \mathcal{C}^∞ in \mathbf{x} and Holder continuous in t with exponent α and $J : \Omega \times [0, T] \rightarrow \mathcal{R}$ be a \mathcal{C}^∞ function in \mathbf{x} and Holder continuous in t with exponent α such that following properties are satisfied*

$$|\operatorname{div} \mathbf{F}(\mathbf{x}, t)| + 2|J(\mathbf{x}, t)| + |\mathbf{F}(\mathbf{x}, t)| \leq c_3 \quad \text{for } (\mathbf{x}, t) \in \Omega \times [0, T], \quad (23)$$

$$\begin{aligned} & |\mathbf{F}(\mathbf{x}, t) - \mathbf{F}(\mathbf{x}, \bar{t})| + |\operatorname{div} \mathbf{F}(\mathbf{x}, t) - \operatorname{div} \mathbf{F}(\mathbf{x}, \bar{t})| + |J(\mathbf{x}, t) - J(\mathbf{x}, \bar{t})| \leq c_4 |t - \bar{t}|^\alpha, \\ & \text{for } (\mathbf{x}, t), (\mathbf{x}, \bar{t}) \in \Omega \times [0, T]. \end{aligned} \quad (24)$$

Then, we have the following estimate holds

$$\int_{\Omega} |u - u_{N_t}|(\mathbf{x}, t) d\mathbf{x} \leq \frac{2c_5}{\alpha + 1} \frac{T^{\alpha+1} e^{c_3 T}}{N_t^\alpha}, \quad (25)$$

where $c_5 = c_4 e^{c_3 T} + c_4 \sqrt{\text{Vol}(\Omega)} e^{c_3 T} \sqrt{2c_3^2 T \int_{\Omega} u^2(\mathbf{x}, 0) + \int_{\Omega} |\nabla u(\mathbf{x}, 0)|^2}$. Specifically, $u(\mathbf{x}, t) = \lim_{N_t \rightarrow \infty} u_{N_t}(\mathbf{x}, t)$ in the L^1 sense on Ω . The convergence rate α depends on the regularity of \mathbf{F} and J in the time variable t .

It is shown in [28] (see Proposition 2.1) that the pathwise robust DMZ equations can be computed by solving the FKE (8).

Proposition 5.2 (Proposition 2.1 of [28]). $\tilde{u}_j(\mathbf{x}, t)$ satisfies the forward Kolmogorov equation

$$\frac{\partial \tilde{u}_j}{\partial t}(\mathbf{x}, t) = \frac{1}{2} \Delta \tilde{u}_j(\mathbf{x}, t) - \mathbf{f}(\mathbf{x}) \cdot \nabla \tilde{u}_j(\mathbf{x}, t) - \left(\text{div } \mathbf{f}(\mathbf{x}) + \frac{1}{2} \mathbf{h}(\mathbf{x})^T \mathbf{h}(\mathbf{x}) \right) \tilde{u}_j(\mathbf{x}, t) \quad (26)$$

for $t_{j-1} \leq t \leq t_j$ if and only if

$$u_j(\mathbf{x}, t) = \exp(-\mathbf{h}(\mathbf{x})^T \mathbf{y}_{t_{j-1}}) \tilde{u}_j(\mathbf{x}, t) \quad (27)$$

satisfies the robust DMZ equation with observation being frozen at $\mathbf{y}_{t_{j-1}}$:

$$\frac{\partial u_j}{\partial t}(\mathbf{x}, t) = \frac{1}{2} \Delta u_j(\mathbf{x}, t) + \mathbf{F}(\mathbf{x}, t_{j-1}) \cdot \nabla u_j(\mathbf{x}, t) + J(\mathbf{x}, t_{j-1}) u_j(\mathbf{x}, t), \quad (28)$$

where $\mathbf{F} = -\mathbf{f} + \nabla K$, $K = \mathbf{h}^T \mathbf{y}_t$, and $J = -\text{div } \mathbf{f} - \frac{1}{2} \mathbf{h}^T \mathbf{h} + \frac{1}{2} \Delta K - \mathbf{f} \cdot \nabla K + \frac{1}{2} |\nabla K|^2$.

In fact, the FKE (26) is obtained from the general form of the FKE (8) by letting $\mathbf{S} = I_d$.

5.2. Convergence analysis for the FD scheme

From time $t = t_{j-1}$ to time $t = t_j$, one can solve the FKE (26) by using the finite difference method. Specifically for time discretization, we partition the time interval $[t_{j-1}, t_j]$ into an equispaced grid, i.e. $t_{j-1} + n\tau$, $n = 0, 1, \dots, \frac{\Delta T}{\tau}$, where τ is the time stepping. We will analyze the error of FD scheme here. The spatial discretization has been discussed in Section 4.1. We use the same notations here, i.e., h is the spatial mesh size. Notice that \mathbf{h} is the observation function. Let $U_{\mathbf{l},j}^n$ denote the approximations of the solution at these grid points and time steps, i.e.,

$$U_{\mathbf{l},j}^n \approx \tilde{u}_j(\mathbf{x}_{\mathbf{l}}, t_{j-1} + n\tau) = \tilde{u}_j(x_1(l_1), x_2(l_2), \dots, x_d(l_d), t_{j-1} + n\tau). \quad (29)$$

Then, the FD scheme for the FKE (26) reads

$$\begin{aligned} \frac{U_{\mathbf{l},j}^{n+1} - U_{\mathbf{l},j}^n}{\tau} &= \frac{1}{2} \frac{\sum_{i=1}^d (U_{\mathbf{l}+\mathbf{e}_i,j}^n + U_{\mathbf{l}-\mathbf{e}_i,j}^n) - (2d)U_{\mathbf{l},j}^n}{h^2} - \frac{\sum_{i=1}^d (f_i)_{\mathbf{l}} (U_{\mathbf{l}+\mathbf{e}_i,j}^n - U_{\mathbf{l}-\mathbf{e}_i,j}^n)}{2h} \\ &\quad - \frac{\sum_{i=1}^d ((f_i)_{\mathbf{l}+\mathbf{e}_i} - (f_i)_{\mathbf{l}-\mathbf{e}_i}) U_{\mathbf{l},j}^n}{2h} - \frac{1}{2} (\mathbf{h}^T \mathbf{h})_{\mathbf{l}} U_{\mathbf{l},j}^n. \end{aligned} \quad (30)$$

for all $\mathbf{l} \in \{1, 2, \dots, N\}^d$ which is a d -dimensional index vector and \mathbf{e}_i is a unit vector with a 1 in the i th coordinate and 0's elsewhere. The convergence result for the FD scheme (30) can be summarized into the following two lemmas.

Lemma 5.3. *The truncation error of the FD scheme (30) for the FKE (26) is $\mathcal{O}(\tau + h^2)$. Hence the FD scheme (30) is consistent.*

Proof. For $\mathbf{x}_1, \mathbf{l} \in \{1, 2, \dots, N\}^d$ and $t_j^n = t_{j-1} + n\tau$, $n = 0, 1, \dots, \frac{\Delta T}{\tau} - 1$, the second-order Taylor expansion of $\tilde{u}_j(\mathbf{x}_1, t_j^n + \tau)$ at the point (\mathbf{x}_1, t_j^n) gives

$$\frac{\tilde{u}_j(\mathbf{x}_1, t_j^n + \tau) - \tilde{u}_j(\mathbf{x}_1, t_j^n)}{\tau} = \frac{\partial \tilde{u}_j}{\partial t}(\mathbf{x}_1, t_j^n) + \frac{\tau}{2} \frac{\partial^2 \tilde{u}_j}{\partial t^2}(\mathbf{x}_1, t_j^n) + o(\tau). \quad (31)$$

The fourth-order Taylor expansion of $\tilde{u}_j(\mathbf{x}_1 + h\mathbf{e}_i, t_j^n)$ and $\tilde{u}_j(\mathbf{x}_1 - h\mathbf{e}_i, t_j^n)$ at the point (\mathbf{x}_1, t_j^n) , gives

$$\frac{\tilde{u}_j(\mathbf{x}_1 + h\mathbf{e}_i, t_j^n) - 2\tilde{u}_j(\mathbf{x}_1, t_j^n) + \tilde{u}_j(\mathbf{x}_1 - h\mathbf{e}_i, t_j^n)}{2h^2} = \frac{1}{2} \frac{\partial^2 \tilde{u}_j}{\partial x_i^2}(\mathbf{x}_1, t_j^n) + \frac{h^2}{24} \frac{\partial^4 \tilde{u}_j}{\partial x_i^4}(\mathbf{x}_1, t_j^n) + o(h^2) \quad (32)$$

The third-order Taylor expansion of $(f_i \tilde{u}_j)(\mathbf{x}_1 + h\mathbf{e}_i, t_j^n)$ and $(f_i \tilde{u}_j)(\mathbf{x}_1 - h\mathbf{e}_i, t_j^n)$ at the point (\mathbf{x}_1, t_j^n) gives

$$\frac{(f_i \tilde{u}_j)(\mathbf{x}_1 + h\mathbf{e}_i, t_j^n) - (f_i \tilde{u}_j)(\mathbf{x}_1 - h\mathbf{e}_i, t_j^n)}{2h} = \frac{\partial (f_i \tilde{u}_j)}{\partial x_i}(\mathbf{x}_1, t_j^n) + \frac{h^2}{6} \frac{\partial^3 (f_i \tilde{u}_j)}{\partial x_i^3}(\mathbf{x}_1, t_j^n) + o(h^2), \quad (33)$$

for $i = 1, 2, \dots, d$. Combining (31), (32) and (33), we get the truncation error of the FD scheme (30) as

$$T_{1,j}^n = \frac{\tau}{2} \frac{\partial^2 \tilde{u}_j}{\partial t^2}(\mathbf{x}_1, t_j^n) + o(\tau) - \sum_{i=1}^d \frac{h^2}{24} \frac{\partial^4 \tilde{u}_j}{\partial x_i^4}(\mathbf{x}_1, t_j^n) - \sum_{i=1}^d \frac{h^2}{6} \frac{\partial^3 (f_i \tilde{u}_j)}{\partial x_i^3}(\mathbf{x}_1, t_j^n) + o(h^2) = \mathcal{O}(\tau + h^2). \quad (34)$$

The consistency of the FD scheme is proved. \square

Lemma 5.4. *Let $U_{l,j}^n$, $n = 0, 1, \dots, \frac{\Delta T}{\tau}$ denote the numerical solution obtained by the FD scheme (30) and $\tilde{u}_j(\mathbf{x}_l, t_{j-1} + n\tau)$ denote the corresponding exact solution to the FKE (26) on $[t_{j-1}, t_j]$, respectively, where $t_j = t_{j-1} + \Delta T$. Suppose the assumptions Asm.1 to Asm.4 and the stability condition $h < \frac{1}{C_f}$, $\tau < (\frac{d}{h^2} + dL_f + \frac{d}{2}C_h^2)^{-1}$ are satisfied. Then, we have the following error estimate for the FD scheme (30),*

$$\|U_{l,j}^{\frac{\Delta T}{\tau}} - \tilde{u}_j(\mathbf{x}_l, t_j)\|_\infty \leq e^{\Delta T d L_f} \|U_{l,j}^0 - \tilde{u}_j(\mathbf{x}_l, t_{j-1})\|_\infty + \frac{C}{dL_f} (\tau + h^2), \quad \forall \mathbf{l} \in \{1, 2, \dots, N\}^d. \quad (35)$$

where C is a generic constant that does not depend on τ and h . Finally, we have the estimate

$$\|U_{l,N_t}^{\frac{\Delta T}{\tau}} - \tilde{u}_{N_t}(\mathbf{x}_l, T)\|_\infty \leq \frac{C e^{c_2 N_t + T d L_f}}{dL_f (e^{c_2 + \Delta T d L_f} - 1)} (\tau + h^2), \quad \forall \mathbf{l} \in \{1, 2, \dots, N\}^d. \quad (36)$$

where c_2 is a constant defined in (20) of Asm.4.

Proof. Let $e_{\mathbf{l},j}^n = U_{\mathbf{l},j}^n - \tilde{u}_j(\mathbf{x}_{\mathbf{l}}, t_{j-1} + n\tau)$ denote the error between FD solution and exact solution of the FKE (26) on $[t_{j-1}, t_j]$. By the definition of the truncation error, we have

$$\begin{aligned} \frac{e_{\mathbf{l},j}^{n+1} - e_{\mathbf{l},j}^n}{\tau} &= \frac{1}{2} \frac{\sum_{i=1}^d (e_{\mathbf{l}+\mathbf{e}_i,j}^n + e_{\mathbf{l}-\mathbf{e}_i,j}^n) - (2d)e_{\mathbf{l},j}^n}{h^2} - \frac{\sum_{i=1}^d (f_i)_{\mathbf{l}} (e_{\mathbf{l}+\mathbf{e}_i,j}^n - e_{\mathbf{l}-\mathbf{e}_i,j}^n)}{2h} \\ &\quad - \frac{\sum_{i=1}^d ((f_i)_{\mathbf{l}+\mathbf{e}_i} - (f_i)_{\mathbf{l}-\mathbf{e}_i}) e_{\mathbf{l},j}^n}{2h} - \frac{1}{2} (\mathbf{h}^T \mathbf{h})_{\mathbf{l}} e_{\mathbf{l},j}^n + T_{\mathbf{l},j}^n, \end{aligned}$$

i.e.

$$\begin{aligned} e_{\mathbf{l},j}^{n+1} &= \left(1 - \frac{d\tau}{h^2} - \tau \sum_{i=1}^d \frac{((f_i)_{\mathbf{l}+\mathbf{e}_i} - (f_i)_{\mathbf{l}-\mathbf{e}_i})}{2h}\right) e_{\mathbf{l},j}^n \\ &\quad + \sum_{i=1}^d \left(\frac{\tau}{2h^2} - \frac{\tau}{2h} (f_i)_{\mathbf{l}}\right) e_{\mathbf{l}+\mathbf{e}_i,j}^n + \sum_{i=1}^d \left(\frac{\tau}{2h^2} + \frac{\tau}{2h} (f_i)_{\mathbf{l}}\right) e_{\mathbf{l}-\mathbf{e}_i,j}^n + \tau T_{\mathbf{l},j}^n, \end{aligned} \quad (37)$$

where $T_{\mathbf{l},j}^n$ is the truncation error; see (34). Let the maximum error at a time step $t_{j-1} + n\tau$ denoted by

$$E_j^n := \max\{|e_{\mathbf{l},j}^n|, \mathbf{l} \in \{1, 2, \dots, N\}^d\}. \quad (38)$$

We introduce the stability conditions

$$\frac{\tau}{2h^2} > \left|\frac{\tau}{2h} (f_i)_{\mathbf{l}}\right|, \quad 1 > \frac{d\tau}{h^2} + \tau \sum_{i=1}^d \frac{((f_i)_{\mathbf{l}+\mathbf{e}_i} - (f_i)_{\mathbf{l}-\mathbf{e}_i})}{2h} + \frac{\tau}{2} (\mathbf{h}^T \mathbf{h})_{\mathbf{l}}. \quad (39)$$

According to Asm.3 and Asm.4, the stability conditions (39) require $h < \frac{1}{C_f}$ and $\tau < (\frac{d}{h^2} + dL_f + \frac{d}{2}C_h^2)^{-1}$, where C_h, C_f, L_f are defined in Asm.3 and Asm.4, and C is a generic constant in the truncation error. Under the stability conditions (39), (37) implies that

$$\begin{aligned} E_j^{n+1} &\leq \left(1 - \frac{d\tau}{h^2} - \tau \sum_{i=1}^d \frac{((f_i)_{\mathbf{l}+\mathbf{e}_i} - (f_i)_{\mathbf{l}-\mathbf{e}_i})}{2h}\right) E_j^n \\ &\quad + \sum_{i=1}^d \left(\frac{\tau}{2h^2} - \frac{\tau}{2h} (f_i)_{\mathbf{l}}\right) E_j^n + \sum_{i=1}^d \left(\frac{\tau}{2h^2} + \frac{\tau}{2h} (f_i)_{\mathbf{l}}\right) E_j^n + \tau C(\tau + h^2) \\ &\leq \left(1 - \tau \sum_{i=1}^d \frac{((f_i)_{\mathbf{l}+\mathbf{e}_i} - (f_i)_{\mathbf{l}-\mathbf{e}_i})}{2h}\right) E_j^n + \tau C(\tau + h^2), \\ &\leq (1 + \tau dL_f) E_j^n + \tau C(\tau + h^2). \end{aligned} \quad (40)$$

Hence, by using (40) recursively, we have the error estimate for the FKE between two observations from t_{j-1} to t_j ,

$$E_j^{\frac{\Delta T}{\tau}} \leq e^{\Delta T dL_f} E_j^0 + \frac{C}{dL_f} (\tau + h^2), \quad (41)$$

which gives the estimate (35), i.e. the error estimate in $[t_{j-1}, t_j]$. Notice that the initial data for the FD scheme (30) is obtained by an exponential transform, which assimilates the new observation data into the predicted solution. Thus, the term E_j^0 satisfies

$$E_j^0 \leq \exp(\mathbf{h}^T(\mathbf{y}_{t_{j-1}} - \mathbf{y}_{t_{j-2}})) E_{j-1}^{\frac{\Delta T}{\tau}} \leq e^{c_2} E_{j-1}^{\frac{\Delta T}{\tau}}. \quad (42)$$

Combining (41) and (42), we obtain

$$E_j^{\frac{\Delta T}{\tau}} \leq e^{c_2 + \Delta T dL_f} E_{j-1}^{\frac{\Delta T}{\tau}} + \frac{C}{dL_f} (\tau + h^2). \quad (43)$$

Recursively using the above estimate and the condition $E_1^0 = 0$, we have

$$\begin{aligned} E_{N_t}^{\frac{\Delta T}{\tau}} &\leq e^{(c_2 + \Delta T dL_f)(N_t - 1)} \left(E_1^{\frac{\Delta T}{\tau}} + \frac{C(\tau + h^2)}{dL_f(e^{c_2 + \Delta T dL_f} - 1)} \right), \\ &\leq e^{(c_2 + \Delta T dL_f)(N_t - 1)} \left(e^{\Delta T dL_f} E_1^0 + \frac{C(\tau + h^2)}{dL_f} + \frac{C(\tau + h^2)}{dL_f(e^{c_2 + \Delta T dL_f} - 1)} \right), \\ &\leq \frac{C e^{c_2 N_t + T dL_f}}{dL_f(e^{c_2 + \Delta T dL_f} - 1)} (\tau + h^2), \end{aligned} \quad (44)$$

which gives the estimate (36). \square

5.3. Convergence analysis for the QTT method

Finally, we analyze the error between the solutions obtained by using the QTT method and the FD method. Let ϵ_1 denote a given precision in the construction of QTT-format and TT-rounding, and ϵ_2 denote the error (in the sense of Frobenius norm) of operator $(\tau \mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}$ between the FD matrix and QTT-format approximation matrix, respectively.

We first analyze the convergence of the QTT method to FD scheme. Note that the QTT method gives approximate solutions only at time t_j , $j = 0, 1, \dots, N_t$. Hence in the following analysis, we only need $U_{l,j}^0$ and $U_{l,j}^{\frac{\Delta T}{\tau}}$ to denote the FD solutions at time t_{j-1} , $j = 1, \dots, N_t$ after the exponential transformation, and at time t_j , $j = 1, \dots, N_t$ before the exponential transformation, respectively.

Lemma 5.5. *Let $\widehat{U}_{l,j}^0$ and $\widehat{U}_{l,j}^{\frac{\Delta T}{\tau}}$ denote the QTT solutions at time t_{j-1} , $j = 1, \dots, N_t$ after the exponential transformation, and at time t_j , $j = 1, \dots, N_t$ before the exponential transformation, respectively. We have the error estimate*

$$\|\widehat{U}_{l,j}^{\frac{\Delta T}{\tau}} - U_{l,j}^{\frac{\Delta T}{\tau}}\|_2 \leq (1 + \epsilon_2) c_6 \|\widehat{U}_{l,j}^0 - U_{l,j}^0\|_2 + \epsilon_2 c_6 \|U_{l,j}^0\|_2, \quad (45)$$

where $c_6 = \|(\tau \mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}\|_2$. Finally, we have the error estimate at the final time

$$\|\widehat{U}_{l,N_t}^{\frac{\Delta T}{\tau}} - U_{l,N_t}^{\frac{\Delta T}{\tau}}\|_2 \leq c_7^{N_t} \left(\epsilon_1 \|U_{l,0}^{\frac{\Delta T}{\tau}}\|_2 \right) + \sum_{j=1}^{N_t} c_7^{N_t-j} \left((1 + \epsilon_2) \epsilon_1 c_6 e^{c_2} \|U_{l,j-1}^{\frac{\Delta T}{\tau}}\|_2 + \epsilon_2 c_6 \|U_{l,j}^0\|_2 \right), \quad (46)$$

where $c_7 = (1 + \epsilon_2)(1 + \epsilon_1) c_6 e^{c_2}$.

Proof. Notice that the online procedure, i.e., the Algorithm 2, is divided into two main parts of computations. The first part is assimilating the observation data into the predicted solution with a TT-rounding procedure afterward. By using the triangular inequality and stability of the FD scheme, we easily obtain

$$\begin{aligned}
\|\widehat{U}_{1,j+1}^0 - U_{1,j+1}^0\|_2 &\leq \|\exp[\widehat{\mathbf{h}}_j^T(\mathbf{y}_{t_j} - \mathbf{y}_{t_{j-1}})]\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}} - \exp[\mathbf{h}_j^T(\mathbf{y}_{t_j} - \mathbf{y}_{t_{j-1}})]U_{1,j}^{\frac{\Delta T}{\tau}}\|_2, \\
&\leq \|\exp[\widehat{\mathbf{h}}_j^T(\mathbf{y}_{t_j} - \mathbf{y}_{t_{j-1}})]\|_2 \left(\|\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}} - U_{1,j}^{\frac{\Delta T}{\tau}}\|_2 \right) + \\
&\quad \left(\|\exp[\widehat{\mathbf{h}}_j^T(\mathbf{y}_{t_j} - \mathbf{y}_{t_{j-1}})] - \exp[\mathbf{h}_j^T(\mathbf{y}_{t_j} - \mathbf{y}_{t_{j-1}})]\|_2 \right) \|\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}}\|_2, \\
&\leq e^{c_2} \|\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}} - U_{1,j}^{\frac{\Delta T}{\tau}}\|_2 + \epsilon_1 e^{c_2} \left(\|\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}} - U_{1,j}^{\frac{\Delta T}{\tau}}\|_2 + \|U_{1,j}^{\frac{\Delta T}{\tau}}\|_2 \right), \\
&\leq (1 + \epsilon_1) e^{c_2} \|\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}} - U_{1,j}^{\frac{\Delta T}{\tau}}\|_2 + \epsilon_1 e^{c_2} \|U_{1,j}^{\frac{\Delta T}{\tau}}\|_2, \tag{47}
\end{aligned}$$

where $\exp[\widehat{\mathbf{h}}_j^T(\mathbf{y}_{t_j} - \mathbf{y}_{t_{j-1}})]$ denotes the QTT-format of the vector $\exp[\mathbf{h}_j^T(\mathbf{y}_{t_j} - \mathbf{y}_{t_{j-1}})]$.

The second part of computation is a matrix-vector multiplication in the QTT-format with a TT-rounding procedure. By using the triangular inequality and the fact $\|\mathbf{M}\|_2 \leq \|\mathbf{M}\|_F$ for any matrix \mathbf{M} , we easily obtain

$$\begin{aligned}
\|\widehat{U}_{1,j+1}^{\frac{\Delta T}{\tau}} - U_{1,j+1}^{\frac{\Delta T}{\tau}}\|_2 &\leq \|(\tau\widehat{\mathbf{A}} + \mathbf{I})^{\frac{\Delta T}{\tau}}\widehat{U}_{1,j+1}^0 - (\tau\mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}U_{1,j+1}^0\|_2, \\
&\leq \|(\tau\mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}\|_2 \left(\|\widehat{U}_{1,j+1}^0 - U_{1,j+1}^0\|_2 \right) + \left(\|(\tau\mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}} - (\tau\widehat{\mathbf{A}} + \mathbf{I})^{\frac{\Delta T}{\tau}}\|_2 \right) \|\widehat{U}_{1,j+1}^0\|_2, \\
&\leq c_6 \|\widehat{U}_{1,j+1}^0 - U_{1,j+1}^0\|_2 + \epsilon_2 c_6 \left(\|\widehat{U}_{1,j+1}^0 - U_{1,j+1}^0\|_2 + \|U_{1,j+1}^0\|_2 \right), \\
&\leq (1 + \epsilon_2) c_6 \|\widehat{U}_{1,j+1}^0 - U_{1,j+1}^0\|_2 + \epsilon_2 c_6 \|U_{1,j+1}^0\|_2, \tag{48}
\end{aligned}$$

where $(\tau\widehat{\mathbf{A}} + \mathbf{I})^{\frac{\Delta T}{\tau}}$ is the QTT-format of the operator matrix $(\tau\mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}$, and we have denoted $c_6 = \|(\tau\mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}\|_2$. The estimate (45) is proved. Combining the above two estimates (47) and (48), we get

$$\begin{aligned}
\|\widehat{U}_{1,j+1}^{\frac{\Delta T}{\tau}} - U_{1,j+1}^{\frac{\Delta T}{\tau}}\|_2 &\leq (1 + \epsilon_2)(1 + \epsilon_1) c_6 e^{c_2} \|\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}} - U_{1,j}^{\frac{\Delta T}{\tau}}\|_2 \\
&\quad + (1 + \epsilon_2) \epsilon_1 c_6 e^{c_2} \|U_{1,j}^{\frac{\Delta T}{\tau}}\|_2 + \epsilon_2 c_6 \|U_{1,j+1}^0\|_2. \tag{49}
\end{aligned}$$

We denote $c_7 = (1 + \epsilon_2)(1 + \epsilon_1) c_6 e^{c_2}$ for notational simplicity. Recursively using the above estimate (49), we obtain

$$\begin{aligned}
\|\widehat{U}_{1,N_t}^{\frac{\Delta T}{\tau}} - U_{1,N_t}^{\frac{\Delta T}{\tau}}\|_2 &\leq c_7^{N_t} \left(\|\widehat{U}_{1,0}^{\frac{\Delta T}{\tau}} - U_{1,0}^{\frac{\Delta T}{\tau}}\|_2 \right) + \sum_{j=1}^{N_t} c_7^{N_t-j} \left((1 + \epsilon_2) \epsilon_1 c_6 e^{c_2} \|U_{1,j-1}^{\frac{\Delta T}{\tau}}\|_2 + \epsilon_2 c_6 \|U_{1,j}^0\|_2 \right), \\
&\leq c_7^{N_t} \left(\epsilon_1 \|U_{1,0}^{\frac{\Delta T}{\tau}}\|_2 \right) + \sum_{j=1}^{N_t} c_7^{N_t-j} \left((1 + \epsilon_2) \epsilon_1 c_6 e^{c_2} \|U_{1,j-1}^{\frac{\Delta T}{\tau}}\|_2 + \epsilon_2 c_6 \|U_{1,j}^0\|_2 \right), \tag{50}
\end{aligned}$$

which completes the proof. \square

Remark 5.1. The estimate (46) reveals the dependence of the error of the QTT solution on different parameters. Since $\|U_{\cdot,j}^{\frac{\Delta T}{\tau}}\|_2$, $j = 0, \dots, N_t$ are bounded, given parameters c_6 , c_7 , and N_t , one can choose ϵ_1 and ϵ_2 accordingly so that the error $\|\widehat{U}_{1,N_t}^{\frac{\Delta T}{\tau}} - U_{1,N_t}^{\frac{\Delta T}{\tau}}\|_2$ is small.

As an immediate result of Lemma 5.4 and Lemma 5.5, we can estimate the error between the solution of the QTT method and the exact solution for the FKE (26) on each time interval $[t_{j-1}, t_j]$. We provide such an estimate in the following lemma, of which the proof is a simple application of the triangular inequality.

Lemma 5.6. *Let $\widehat{U}_{l,j}^{\frac{\Delta T}{\tau}}$ denote the solution of the QTT method and $\tilde{u}_j(\mathbf{x}_l, t_j)$ denote the exact solution for FKE (26) on each time interval $[t_{j-1}, t_j]$, respectively. We have the error estimate*

$$\|\widehat{U}_{l,j}^{\frac{\Delta T}{\tau}} - \tilde{u}_j(\mathbf{x}_l, t_j)\|_\infty \leq (\epsilon_1 + \epsilon_1\epsilon_2 + \epsilon_2)c_6\|U_{l,j}^0\|_2 + \frac{C(\tau + h^2)}{dL_f}. \quad (51)$$

Proof. From the error estimate (35) in Lemma 5.4 and the estimate (45) in Lemma 5.5, and the infinity norm of any vector is bounded by its 2-norm, we have

$$\begin{aligned} \|\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}} - \tilde{u}_j(\mathbf{x}_1, t_j)\|_\infty &\leq \|\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}} - U_{1,j}^{\frac{\Delta T}{\tau}}\|_\infty + \|U_{1,j}^{\frac{\Delta T}{\tau}} - \tilde{u}_j(\mathbf{x}_1, t_j)\|_\infty \\ &\leq (1 + \epsilon_2)c_6\|\widehat{U}_{1,j}^0 - U_{1,j}^0\|_2 + \epsilon_2c_6\|U_{1,j}^0\|_2 + e^{\Delta T dL_f}\|U_{1,j}^0 - \tilde{u}_j(\mathbf{x}_1, t_{j-1})\|_\infty + \frac{C}{dL_f}(\tau + h^2). \end{aligned} \quad (52)$$

Since at each initial time t_{j_1} the FD method uses the exact initial data $\tilde{u}_j(\mathbf{x}_1, t_{j-1})$ to compute, so we have $\|U_{1,j}^0 - \tilde{u}_j(\mathbf{x}_1, t_{j-1})\|_\infty = 0$. In addition, we have $\|\widehat{U}_{1,j}^0 - U_{1,j}^0\|_2 \leq \epsilon_1\|U_{1,j}^0\|_2$. Thus, the error estimate in (51) is proved. \square

Now we are in a position to present the main result. Recall that $u(\mathbf{x}, t)$ denote the solution of pathwise robust DMZ equation (21) on $\Omega \times [0, T]$, $u_{N_t}(\mathbf{x}, t)$ denote the solution of frozen time equation (22) on $\Omega \times [t_{N_t-1}, t_{N_t}]$. Let \mathcal{I} denote an interpolation operator, which can be a polynomial interpolation or spline interpolation, and let $\widehat{u}(\mathbf{x}, t_j) = \mathcal{I}\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}}$ denote the obtained function based on the QTT solution.

Theorem 5.7. *The QTT solution converges to the solution of the pathwise robust DMZ equation on a bounded domain in the sense of L^1 norm. Specifically, we have the error estimate as follows,*

$$\begin{aligned} &\|u(\mathbf{x}, T) - \exp(-\mathbf{h}(\mathbf{x})^T \mathbf{y}_{t_{N_t-1}})\widehat{u}(\mathbf{x}, t_{N_t})\|_{L^1} \leq \frac{2c_5}{\alpha + 1} \frac{T^{\alpha+1}e^{c_3T}}{N_t^\alpha} + \\ &C e^{c_2} \left(\frac{C e^{c_2 N_t + T dL_f}}{dL_f (e^{c_2 + \Delta T dL_f} - 1)} (\tau + h^2) + h^2 \|\Delta \tilde{u}_{N_t}(\mathbf{x}, T)\|_{L^2} \right) + \\ &C e^{c_2} c_7^{N_t} \left(\epsilon_1 \|U_{l,0}^{\frac{\Delta T}{\tau}}\|_2 \right) + C e^{c_2} \sum_{j=1}^{N_t} c_7^{N_t-j} \left((1 + \epsilon_2)\epsilon_1 c_6 e^{c_2} \|U_{l,j-1}^{\frac{\Delta T}{\tau}}\|_2 + \epsilon_2 c_6 \|U_{l,j}^0\|_2 \right). \end{aligned} \quad (53)$$

Proof. By the triangle inequality, we split the error into three main parts

$$\begin{aligned}
& \left\| u(\mathbf{x}, T) - \exp(-\mathbf{h}(\mathbf{x})^T \mathbf{y}_{t_{N_t-1}}) \widehat{u}(\mathbf{x}, t_{N_t}) \right\|_{L^1} \\
& \leq \left\| u(\mathbf{x}, T) - u_{N_t}(\mathbf{x}, T) \right\|_{L^1} + \left\| \exp(-\mathbf{h}(\mathbf{x})^T \mathbf{y}_{t_{N_t-1}}) \widetilde{u}_{N_t}(\mathbf{x}, T) - \exp(-\mathbf{h}(\mathbf{x})^T \mathbf{y}_{t_{N_t-1}}) \mathcal{I}U_{1, N_t}^{\frac{\Delta T}{\tau}} \right\|_{L^1} \\
& \quad + \left\| \exp(-\mathbf{h}(\mathbf{x})^T \mathbf{y}_{t_{N_t-1}}) \mathcal{I}U_{1, N_t}^{\frac{\Delta T}{\tau}} - \exp(-\mathbf{h}(\mathbf{x})^T \mathbf{y}_{t_{N_t-1}}) \widehat{\mathcal{I}U}_{1, N_t}^{\frac{\Delta T}{\tau}} \right\|_{L^1}, \\
& := E_1 + E_2 + E_3,
\end{aligned} \tag{54}$$

where we have used the condition $u_{N_t}(\mathbf{x}, T) = \exp(-\mathbf{h}(\mathbf{x})^T \mathbf{y}_{t_{N_t-1}}) \widetilde{u}_{N_t}(\mathbf{x}, T)$ (see Eq.(27)). In what follows, we shall estimate these three error terms separately.

From Prop.5.1, we know the error term E_1 satisfies

$$E_1 = \left\| u(\mathbf{x}, T) - u_{N_t}(\mathbf{x}, T) \right\|_{L^1} \leq \frac{2c_5}{\alpha + 1} \frac{T^{\alpha+1} e^{c_3 T}}{N_t^\alpha}. \tag{55}$$

Then, we have the estimate for the error term E_2 as follows,

$$\begin{aligned}
E_2 & = \left\| \exp(-\mathbf{h}(\mathbf{x})^T \mathbf{y}_{t_{N_t-1}}) \widetilde{u}_{N_t}(\mathbf{x}, T) - \exp(-\mathbf{h}(\mathbf{x})^T \mathbf{y}_{t_{N_t-1}}) \mathcal{I}U_{1, N_t}^{\frac{\Delta T}{\tau}} \right\|_{L^1} \\
& \leq e^{c_2} \left\| \widetilde{u}_{N_t}(\mathbf{x}, T) - \mathcal{I}U_{1, N_t}^{\frac{\Delta T}{\tau}} \right\|_{L^1} \\
& \leq e^{c_2} \left(\left\| \mathcal{I}U_{1, N_t}^{\frac{\Delta T}{\tau}} - \mathcal{I}\widetilde{u}_{N_t}(\mathbf{x}_1, T) \right\|_{L^1} + \left\| \mathcal{I}\widetilde{u}_{N_t}(\mathbf{x}_1, T) - \widetilde{u}_{N_t}(\mathbf{x}, T) \right\|_{L^1} \right) \\
& \leq C e^{c_2} \left(\left\| U_{1, N_t}^{\frac{\Delta T}{\tau}} - \widetilde{u}_{N_t}(\mathbf{x}_1, T) \right\|_\infty + h^2 \left\| \Delta \widetilde{u}_{N_t}(\mathbf{x}, T) \right\|_{L^2} \right) \\
& \leq C e^{c_2} \left(\frac{C e^{c_2 N_t + T d L_f}}{d L_f (e^{c_2 + \Delta T d L_f} - 1)} (\tau + h^2) + h^2 \left\| \Delta \widetilde{u}_{N_t}(\mathbf{x}, T) \right\|_{L^2} \right)
\end{aligned} \tag{56}$$

where C is a generic constant depending on Ω . Here we have used the fact that the norm of the operator \mathcal{I} is bounded since the function $\widetilde{u}_{N_t}(\mathbf{x}, T)$ is smooth. In addition, the estimate (36) in Lemma 5.4 is used.

Finally, we estimate the error term E_3 and get

$$\begin{aligned}
& \left\| \exp(-\mathbf{h}(\mathbf{x})^T \mathbf{y}_{t_{N_t-1}}) \widehat{\mathcal{I}U}_{1, N_t}^{\frac{\Delta T}{\tau}} - \exp(-\mathbf{h}(\mathbf{x})^T \mathbf{y}_{t_{N_t-1}}) \mathcal{I}U_{1, N_t}^{\frac{\Delta T}{\tau}} \right\|_{L^1} \\
& \leq e^{c_2} \left\| \widehat{\mathcal{I}U}_{1, N_t}^{\frac{\Delta T}{\tau}} - \mathcal{I}U_{1, N_t}^{\frac{\Delta T}{\tau}} \right\|_{L^1} \leq C e^{c_2} \left\| \widehat{U}_{1, N_t}^{\frac{\Delta T}{\tau}} - U_{1, N_t}^{\frac{\Delta T}{\tau}} \right\|_2 \\
& \leq C e^{c_2} c_7^{N_t} \left(\epsilon_1 \left\| U_{1, 0}^{\frac{\Delta T}{\tau}} \right\|_2 \right) + C e^{c_2} \sum_{j=1}^{N_t} c_7^{N_t-j} \left((1 + \epsilon_2) \epsilon_1 c_6 e^{c_2} \left\| U_{1, j-1}^{\frac{\Delta T}{\tau}} \right\|_2 + \epsilon_2 c_6 \left\| U_{1, j}^0 \right\|_2 \right)
\end{aligned} \tag{57}$$

where C is a constant depending on Ω and the estimate (46) in Lemma 5.5 is used. Combining above formulas (55)(56)(57), we prove the statement in the theorem 5.7. \square

Remark 5.2. The accuracy of the QTT method in computing the the solution of pathwise robust DMZ equation controlled by three components of approximations, i.e. E_1 , E_2 , and E_3 . In practice, we can choose N_t , τ , h , ϵ_1 and ϵ_2 accordingly so that the error $\left\| u(\mathbf{x}, T) - \exp(-\mathbf{h}(\mathbf{x})^T \mathbf{y}_{t_{N_t-1}}) \widehat{u}(\mathbf{x}, t_{N_t}) \right\|_{L^1}$ is small.

6. Numerical results

In this section, we are interested in investigating the approximation properties of our method and computational savings over the existing methods, such as the FD method and particle filter (PF) method. We shall carry out numerical experiments on two 3D NLF problems. The definitions of these two NLF problems are given as follows,

Example 1: An almost linear problem This problem is modeled by a SDE in the Ito form as follows,

$$\begin{cases} dx_1 = -0.3x_1 + dv_1, \\ dx_2 = -0.3x_2 + dv_2, \\ dx_3 = -0.3x_3 + dv_3, \\ dy_1 = (x_2 + \sin(x_1))dt + dw_1, \\ dy_2 = (x_3 + \sin(x_2))dt + dw_2, \\ dy_3 = (x_1 + \sin(x_3))dt + dw_3. \end{cases} \quad (58)$$

where $E[d\mathbf{v}_t d\mathbf{v}_t^T] = 1.5I_3 dt$ with $\mathbf{v} = [v_1, v_2, v_3]^T$, $E[d\mathbf{w}_t d\mathbf{w}_t^T] = I_3 dt$ with $\mathbf{w} = [w_1, w_2, w_3]^T$, and I_3 is the identity matrix of size 3×3 . The noise are independent Brownian motions. The initial state is $\mathbf{x}(0) = [x_1(0), x_2(0), x_3(0)]^T = [0, 0, 0]^T$ with $\mathbf{x}(t) = [x_1(t), x_2(t), x_3(t)]^T$.

Example 2: A cubic sensor problem This problem is modeled by a SDE in the Ito form as follows,

$$\begin{cases} dx_1 = (-0.6x_1 - 0.1x_2)dt + dv_1, \\ dx_2 = (-0.5x_2 + 0.1x_3)dt + dv_2, \\ dx_3 = (-0.6x_3 + 0.1x_1)dt + dv_3, \\ dy_1 = x_2^3 dt + dw_1, \\ dy_2 = x_3^3 dt + dw_2, \\ dy_3 = x_1^3 dt + dw_3, \end{cases} \quad (59)$$

where $E[d\mathbf{v}_t d\mathbf{v}_t^T] = 1.5I_3 dt$ with $\mathbf{v} = [v_1, v_2, v_3]^T$, $E[d\mathbf{w}_t d\mathbf{w}_t^T] = I_3 dt$ with $\mathbf{w} = [w_1, w_2, w_3]^T$, I_3 is the identity matrix of size 3×3 . The initial state is $\mathbf{x}(0) = [x_1(0), x_2(0), x_3(0)]^T = [0, 0, 0]^T$ with $\mathbf{x}(t) = [x_1(t), x_2(t), x_3(t)]^T$.

The total experimental time is $T = 20s$ for both examples. The cubic sensor problem has higher nonlinearity than the almost linear one. Thus, it is more difficult.

6.1. QTT-ranks in the spatial discretization

We consider the spatial discretization of operators on the 3D domain in the QTT-format. We have shown that the exact QTT-decomposition of the discretized Laplace operator via standard FD scheme and first partial derivative operator via central difference scheme have low QTT-ranks. Hence, we mainly compute the QTT-ranks of the discretization of velocity field $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})]^T$ and function $\mathbf{h}^T \mathbf{S}^{-1} \mathbf{h}$.

Let us define the effective QTT-rank of a QTT-format as

$$r_{eff} := \frac{\sqrt{(r_0 n_1 + r_d n_d)^2 + 4(\sum_{k=2}^{d-1} n_k) \sum_{k=1}^d r_{k-1} n_k r_k - (r_0 n_1 + r_d n_d)}}{2 \sum_{k=2}^{d-1} n_k}, \quad (60)$$

where n_k, r_k representing mode sizes and QTT-ranks. Notice that $n_k = 2, k = 1, \dots, d$, in the QTT-format case. In Table 1 and Table 2, we show the effective QTT-rank for the Example 1 and Example 2, respectively. We observe a very slow grow in the effective QTT-rank with respect to the degree of freedom in the spatial discretization.

N on each direction	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$	$\mathbf{h}^T S^{-1} \mathbf{h}$
2^4	1.32	1.32	1.32	3.03
2^5	1.34	1.34	1.34	3.53
2^6	2.20	1.35	2.04	4.27
2^7	2.23	2.34	2.29	4.80
2^8	2.40	2.35	2.30	5.15

Table 1: Effective QTT-rank of the discretized functions on the spatial grid with a given precision 1×10^{-12} in the almost linear problem.

N on each direction	$f_1(\mathbf{x})$	$f_2(\mathbf{x})$	$f_3(\mathbf{x})$	$\mathbf{h}^T S^{-1} \mathbf{h}$
2^4	1.69	1.69	2.00	4.77
2^5	1.70	1.70	2.00	5.41
2^6	1.71	1.71	2.00	5.82
2^7	2.65	2.65	2.93	6.08
2^8	2.66	2.63	2.94	6.27

Table 2: Effective QTT-rank of the discretized functions on the spatial grid with a given precision 1×10^{-12} in the cubic sensor problem.

In Table 3 and Table 4, we show the effective QTT-ranks of assembled tensors $(\tau \mathbf{A} + \mathbf{I})$ and $(\tau \mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}$ that are pre-computed in the offline procedure; see Eq.(17) and Eq.(18). In our experiments, we set $\Delta T = 0.05$ and $\frac{\Delta T}{\tau} = 100$ in Example 1 ($\frac{\Delta T}{\tau} = 200$ in Example 2). Recall the ΔT is the time between two observations and τ is the time step in discretizing the FKE (8). The time step τ in Example 1 and Example 2 is chosen in such a way that the Courant-Friedrichs-Lewy (CFL) stability condition is satisfied. Note that the requirement of a small enough τ makes the finite difference method expensive for solving high-dimensional and/or nonlinear problems. While in our QTT method, the trouble caused by small τ is avoided since we can compute $(\tau \mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}$ and approximated it using the QTT method in the offline procedure. Moreover, we can find that the accuracy of $(\tau \mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}$ is bounded by $\frac{\Delta T}{\tau} \epsilon$, if the TT-rounding precision ϵ is given.

From the results in Tables (1)-(2) and (3)-(4), we find that the QTT-ranks increase very slowly when N increases. Hence, by extracting low-dimensional structures in the solution space, the QTT method helps us alleviate the curse of dimensionality to a certain extent.

spatial N on each direction	Example 1	Example 2
2^4	15.56	15.42
2^5	16.65	16.31
2^6	19.56	17.25
2^7	22.17	22.37
2^8	22.96	22.87

Table 3: Effective QTT-rank of the assembled operator $(\tau\mathbf{A} + \mathbf{I})$ with a given precision 1×10^{-12} .

spatial N on each direction	Example 1	Example 2
2^4	8.28	9.04
2^5	9.63	12.96
2^6	12.94	17.46
2^7	17.28	21.88
2^8	23.47	28.17

Table 4: Effective QTT-rank of the assembled operator $(\tau\mathbf{A} + \mathbf{I})^{\frac{\Delta T}{\tau}}$ with a given TT-rounding precision 5×10^{-4} in Example 1 and 5×10^{-5} in Example 2.

6.2. Comparison with existing methods

To compute the reference solution, we solve Eqns.(58) and (59) using Euler-Maruyama scheme [17] with a fine time step, which generates two sequences X_{t_i} and Y_{t_i} of length $dt = 0.001$ as discrete real states at time $t_i = idt, i = 1, \dots, 20000$. We feed the observation Y_{t_j} into the online procedure at each observation time $t_j = j\Delta T$, i.e. only a subsequence Y_{t_j} of Y_{t_i} is regarded as observation sequence and utilized.

To solve the NLF problem in a real time manner, one need to solve the path-wise robust DMZ equation associated with Eqns.(58) and (59). As such, one can solve the FKEs associated with Eqns.(58) and (59) using a FD method. However, the FD method becomes expensive when the dimension of the NLF problem increases. We shall show that the QTT method provides considerable savings over the FD method.

For the FKE associated with the almost linear problem, we restrict the FKE on the domain $[-5, 5]^3$ and discretize the domain into 2^6 grids on each dimension. Thus the total degree of freedom is 2^{18} . The initial distribution is assumed to be a Gaussian function, where the corresponding unnormalized conditional density function is $\sigma_0(\mathbf{x}) = \exp(-4|\mathbf{x}|^2)$. The time step is chosen to be $\tau = \frac{\Delta T}{100}$ so that the CFL stability condition is satisfied. Based on the initial discretization of $\sigma_0(\mathbf{x})$, we implement the QTT method to solve the FKE simultaneously, where we fix the TT-rounding precision to be $\epsilon = 5 \times 10^{-4}$.

Since $\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}}$ denotes the predicted solution of the QTT method at different time, which approximates the unnormalized conditional density of the states $\mathbf{x}(t) = [x_1(t), x_2(t), x_3(t)]^T$. Thus, we can compute the state estimation results in three coordinates separately. Specifically, we first compute the Hadamard product of $\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}}$ and the QTT-format of each coordinate. Then, compute its total sum and divide it by the total sum of $\widehat{U}_{1,j}^{\frac{\Delta T}{\tau}}$. The results corresponding the

FD method can be computed similarly.

In Fig.1, we show the state estimation results of the almost linear problem in three coordinates separately. The CPU time of the FD method is 2052s, but the QTT method only requires 15.19s. A significant computational saving is achieved by our method because most of the operations in the QTT method only logarithmically depend on the total degree of freedom and polynomially depend on the QTT-ranks, which are relatively small; see Tables (1)–(4). The CPU time of the PF method is 17.36. The efficiency of the PF method is closely related to the number of particles. In this example, we use 3000 particles to avoid explosion in the tracking, which might happen frequently if only 2000 particles are used. In Fig.2, we show the profile of the density function at time $t = 10$.

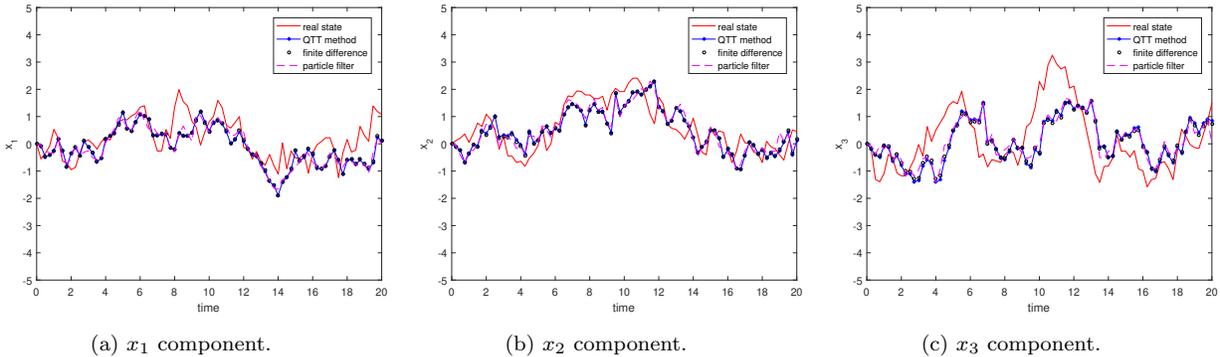


Figure 1: Comparison of a trajectory of the almost linear problem obtained by using different methods.

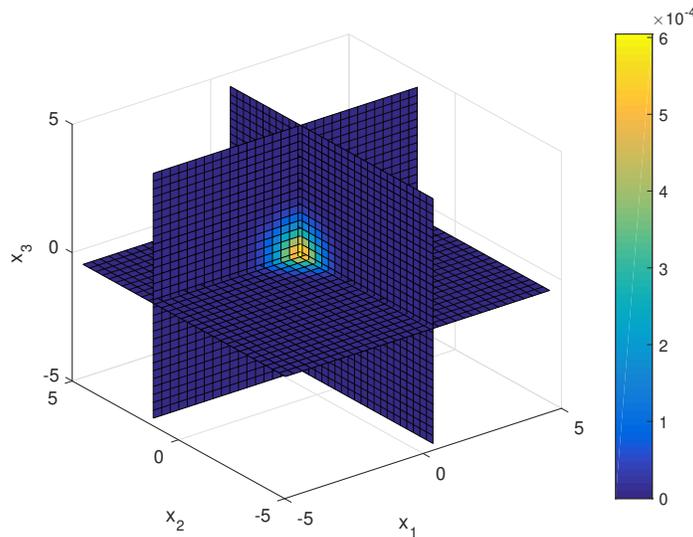


Figure 2: The estimations of density function at $t = 10s$ in the almost linear problem.

For the FKE associated with the cubic sensor problem, we restrict the FKE on the domain $[-3, 3]^3$ and discretize the domain into 2^6 grids on each dimension. Thus, the total degree of freedom is 2^{18} . The unnormalized conditional density function of the initial state is $\sigma_0(\mathbf{x}) = \exp(-10(x_1^4 + x_2^4 + x_3^4))$. The time step is chosen to be $\tau = \frac{\Delta T}{200}$, which is smaller than the

first example, in order to satisfy the CFL stability condition. We fix a higher TT-rounding precision $\epsilon = 5 \times 10^{-5}$ due to the higher nonlinearity in this example.

In Fig. 3, we show the estimation results of the cubic sensor problem in three coordinates separately. The CPU time of the FD method is 4079s, while the QTT method is only 17.11s. The time cost of offline computing of the QTT method is 17.35s. Even though the QTT-format tensor constructor has linear dependence on the degree of freedom (see Prop.4.3), it takes up a minor part of the total computational time. The CPU time of the PF is 29.45s. In this example, we use 5000 particles to avoid explosion in the tracking. In Fig.4, we show the profile of the density function at time $t = 12s$.

We repeat the experiment for $N_{path} = 100$ times and record the mean square errors (MSEs) averaged over 100 sample paths. We find that the MSE between the QTT solution and the FD solution is 0.007 in Example 1 and 0.023 in Example 2, respectively, which shows that FD discretization of the unnormalized conditional density function has a low-rank structure and such structure is approximated well in the QTT-format. Therefore, the QTT method gives a very accurate approximation to the FD solution with considerable savings.

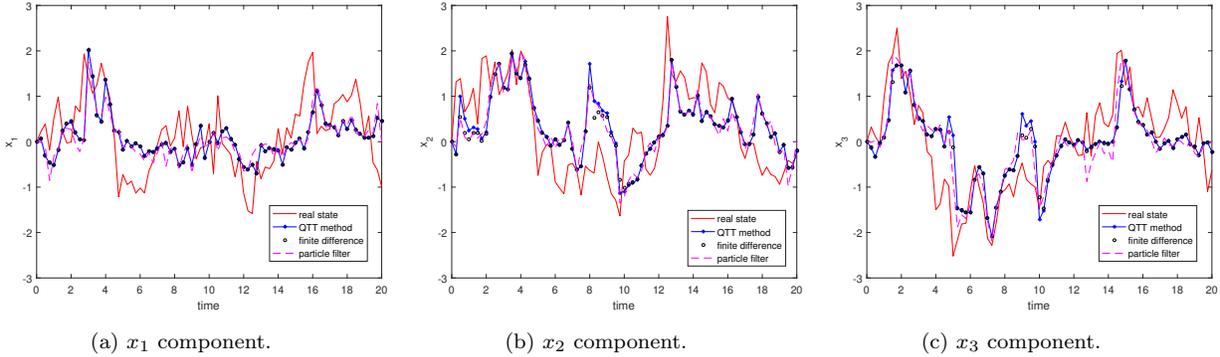


Figure 3: Comparison of a trajectory of the cubic sensor problem obtained using different methods.

Remark 6.1. The PF method is a very popular method in solving NLF problem, which is a Monte Carlo method and it requires a certain amount of sample to compute statistical quantities. Since the PF method and our method are based on a totally different methodology, we cannot reach a general conclusion about their performances for NLF problems.

In Fig.5, we show the convergence of the QTT method with respect to the TT-rounding precision at discrete time points. We use the same settings in the spatial and temporal discretization that were used in the experiments before and only change the TT-rounding precision. We compare the estimated states obtained by the QTT method and FD method, which provides the reference solution. One can see that when we decrease the TT-rounding precision, the relative error of the QTT solution decreases accordingly, which agrees with our convergence analysis. In practice, we choose the TT-rounding precision in such a way that we can balance the accuracy and computational speed.

6.3. Verification of the computational complexity

In this subsection, we intend to verify the computational complexity studied in the Prop. 4.3. Notice that the main computational load in the online procedure consists of two parts. The

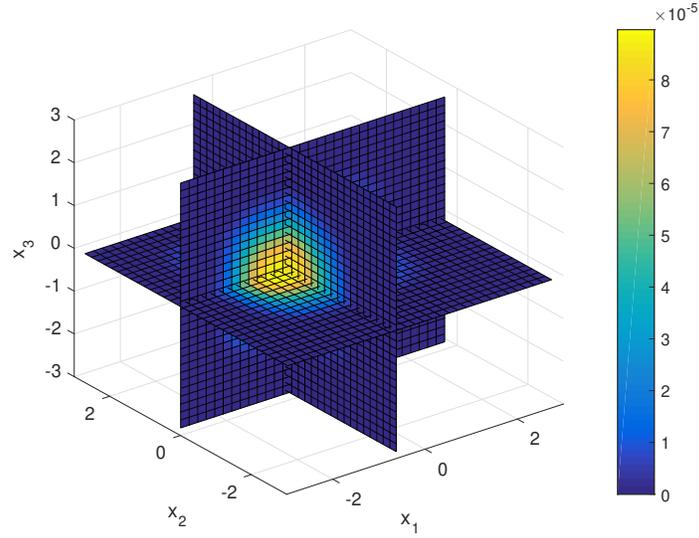


Figure 4: The estimations of density function at $t = 12s$ in cubic sensor problem

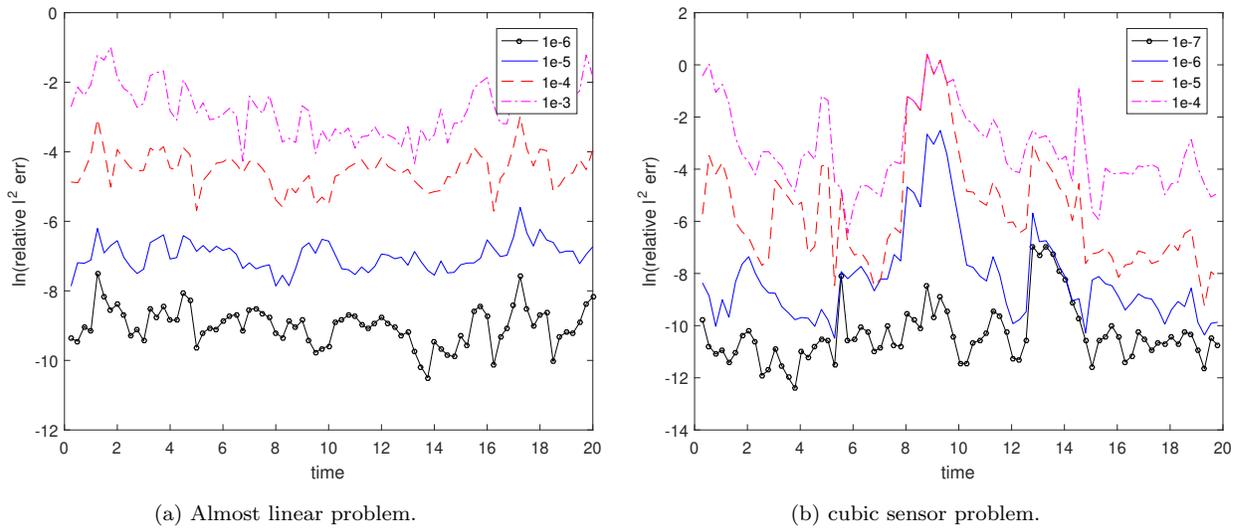


Figure 5: Error of estimated states between the QTT method and reference method under different TT-rounding precision.

first part is $\mathcal{O}(d \log_2(N)r^6)$, which comes from the QTT operations in solving the FKEs. It polynomially depends on QTT-ranks and logarithmically depends on the degree of freedom. Hence, the QTT method brings in significant savings for high-dimensional problems that have a low-dimensional approximation. The second part of the computational load is $\mathcal{O}(N^d r^2)$, which comes from assimilating the observation data into the QTT solution and depends linearly on the degree of freedom in the spatial discretization. We find that these two parts are comparable in the 3D numerical experiments that were studied in this paper.

Let t_{FKE} denote the CPU time in solving the FKE and t_{EXP} is the CPU time in computing the exponential transformation, i.e., assimilating the observation data, respectively. In Table 5, we show the computational time of the QTT method and FD method in solving the Example 1. The QTT-rank is the averaged effective QTT-rank of the FKE solution \tilde{u} at every time. We find that the growth of computational complexity of the QTT method is significantly slower than that of the FD method. When QTT operations part is dominant, i.e. $t_{FKE} > t_{EXP}$, the QTT method can achieve almost logarithmic complexity with respect to the degree of freedom. This numerical experiment shows that the QTT method can solving high-dimensional NLF problems in a real time manner, while the FD method is too expensive. Notice that in Table 5, when $N = 2^7$ the FD method would cost about 10 hours of computation, which makes no sense to do this experiment.

N	QTT method			Finite difference	
	t_{FKE}	t_{EXP}	QTT-rank	t_{FKE}	t_{EXP}
2^4	1.87	0.85	6.83	9.62	0.03
2^5	4.15	1.36	7.88	169	0.08
2^6	11.18	5.20	8.78	2802	0.65
2^7	27.49	46.28	8.89	—	—

Table 5: CPU time(sec.) of the QTT method and finite difference method. N is the grid number in each dimension.

7. Conclusions

In this paper, we develop an efficient numerical method to solve high-dimensional nonlinear filtering (NLF) problems. Specifically, we use the tensor train decomposition method to solve the forward Kolmogorov equation (FKE) arising from the NLF problem. Our method consists of offline and online stages. In the offline stage, we use the finite difference method to discrete the partial differential operators involved in the FKE and extract low-dimensional structures in the solution space of the FKE using the tensor train decomposition method. In addition, we approximate the evolution of the FKE operator using the tensor train decomposition method. With the pre-computed and saved low-rank approximation tensors, we achieve fast computing in the online stage given new observation data. Under some mild assumptions, we provide convergence analysis for the proposed method. Our analysis result reveals different sources of errors and provides some guidance on the implementation of our method so that the error is controllable. Finally, we present numerical results to verify the efficiency and accuracy of the

proposed method in solving 3D NLF problems. Numerical results show that the solutions of the FKEs indeed have certain low-dimensional structures. By using the tensor train decomposition method to extract the low-dimensional structures in the solution space of the FKE, we succeed in solving the high-dimensional NLF problems in a real-time manner.

There are two directions we want to explore in our future work. First, we are interested in developing efficient numerical methods for high-dimensional NLF problems (with $d > 3$), which will be reported in our subsequent work. In addition, we will develop numerical methods to solve NLF problems, where the drift and observation functions are time-dependent. This type of problem is more difficult since the potential low-dimensional structures in the solution space may vary with respect to time. One may need to develop some dynamically low-dimensional approximation methods to address this issue; see e.g. [5, 6].

8. Acknowledgements

The research of S. Li is partially supported by the Doris Chen Postgraduate Scholarship. The research of Z. Wang is partially supported by the Hong Kong PhD Fellowship Scheme. The research of S. S.-T. Yau was supported by the National Natural Science Foundation of China (11471184), Tsinghua University Education Foundation fund (042202008), and a start-up fund from Tsinghua University. The research of Z. Zhang is supported by the Hong Kong RGC grants (Projects 27300616, 17300817, and 17300318), National Natural Science Foundation of China (Project 11601457), Seed Funding Programme for Basic Research (HKU), and Basic Research Programme (JCYJ20180307151603959) of The Science, Technology and Innovation Commission of Shenzhen Municipality.

References

- [1] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian Bayesian tracking. *IEEE Transactions on signal processing*, 50(2):174–188, 2002.
- [2] A. Bain and D. Crisan. *Fundamentals of stochastic filtering*, volume 3. Springer, 2009.
- [3] J. Baras, G. Blankenship, and W. Hopkins. Existence, uniqueness, and asymptotic behavior of solutions to a class of Zakai equations with unbounded coefficients. *IEEE Transactions on Automatic Control*, 28(2):203–214, 1983.
- [4] A. Bensoussan, R. Glowinski, and A. Rascanu. Approximation of the Zakai equation by the splitting up method. *SIAM Journal on Control and Optimization*, 28(6):1420–1431, 1990.
- [5] M. Cheng, T. Y. Hou, and Z. Zhang. A dynamically bi-orthogonal method for stochastic partial differential equations I: derivation and algorithms. *J. Comput. Phys.*, 242:843–868, 2013.
- [6] M. Cheng, T. Y. Hou, and Z. Zhang. A dynamically bi-orthogonal method for stochastic partial differential equations II: adaptivity and generalizations. *J. Comput. Phys.*, 242:753–776, 2013.

- [7] A. Budhiraja and H. Kushner. Robustness of nonlinear filters over the infinite time interval. *SIAM journal on control and optimization*, 36(5):1618–1637, 1998.
- [8] S. V. Dolgov, B. N. Khoromskij, and I. V. Oseledets. Fast solution of parabolic problems in the tensor train/quantized tensor train format with initial application to the Fokker-Planck equation. *SIAM Journal on Scientific Computing*, 34(6):A3016–A3038, 2012.
- [9] T. Duncan. Probability densities for diffusion processes with applications to nonlinear filtering theory and detection theory. *Technical report, Stanford Univ CA Stanford Electronics Labs*, 1967.
- [10] W. Fleming and S. Mitter. Optimal control and nonlinear filtering for nondegenerate diffusion processes. *Stochastics*, 8(1):63–77, 1982.
- [11] F. Gustafsson, F. Gunnarsson, N. Bergman, U. Forssell, J. Jansson, R. Karlsson, and P. J. Nordlund. Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, 50(2):425–437, 2002.
- [12] I. Gyöngy and N. Krylov. On the splitting-up method and stochastic partial differential equations. *The Annals of Probability*, 31(2):564–591, 2003.
- [13] K. Ito. Approximation of the Zakai equation for nonlinear filtering. *SIAM Journal on Control and Optimization*, 34(2):620–634, 1996.
- [14] G. Kallianpur. *Stochastic filtering theory*, volume 13. Springer Science & Business Media, 2013.
- [15] Vlanimir A. Kazeev, and Boris N. Khoromskij. Low-rank explicit QTT representation of the laplace operator and its inverse. *SIAM Journal on Matrix Analysis and Applications*, 33(3):724–758, 2012.
- [16] Boris N. Khoromskij. $\mathcal{O}(d \log N)$ -Quantics approximation of $N - d$ tensors in high-dimensional numerical modeling. *Constructive Approximation*, 34:257–280, 2011.
- [17] P. E. Kloeden and E. Platen. *Numerical solution of Stochastic Differential Equations*. Springer-Verlag Berlin, Heidelberg, 1992.
- [18] S. Lototsky, R. Mikulevicius, and B. L. Rozovskii. Nonlinear filtering revisited: a spectral approach. *SIAM Journal on Control and Optimization*, 35(2):435–461, 1997.
- [19] X. Luo and Stephen S. T. Yau. Complete real time solution of the general nonlinear filtering problem without memory. *IEEE Transactions on Automatic Control*, 58(10):2563–2578, 2013.
- [20] X. Luo and Stephen S. T. Yau. Hermite spectral method to 1-D forward Kolmogorov equation and its application to nonlinear filtering problems. *IEEE Transactions on Automatic Control*, 58(10):2495–2507, 2013.

- [21] R. Mortensen. Optimal control of continuous-time stochastic systems. *Technical report, California Univ Berkeley Electronics Research Lab*, 1966.
- [22] N. Nagase. Remarks on nonlinear stochastic partial differential equations: an application of the splitting-up method. *SIAM Journal on Control and Optimization*, 33(6):1716–1730, 1995.
- [23] I. V. Oseledets. Approximation of $2^d \times 2^d$ matrices using tensor decomposition. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2130–2145, 2010.
- [24] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.
- [25] I. V. Oseledets and E. E. Tyrtyshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432:70–88, 2010.
- [26] E. Pardoux. Stochastic partial differential equations and filtering of diffusion processes. *Stochastics*, 3:127–167, 1980.
- [27] Z. Wang, X. Luo, Stephen S.-T. Yau, and Z. Zhang. Proper orthogonal decomposition method to nonlinear filtering problems in medium-high dimension. *to appear in IEEE Transactions on Automatic Control*, 2020.
- [28] Shing-Tung Yau and Stephen S-T Yau. Real time solution of the nonlinear filtering problem without memory II. *SIAM Journal on Control and Optimization*, 47(1):163–195, 2008.
- [29] M. Yueh, W. Lin, and S. T. Yau. An efficient numerical method for solving high-dimensional nonlinear filtering problems. *Communications in Information and Systems*, 14(4):243–262, 2014.
- [30] M. Zakai. On the optimal filtering of diffusion processes. *Probability Theory and Related Fields*, 11(3):230–243, 1969.
- [31] Q. Zhang. Nonlinear filtering and control of a switching diffusion with small observation noise. *SIAM Journal on Control and Optimization*, 36(5):1638–1668, 1998.