Journal of Computational Physics 289 (2015) 149–168

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A numerical solver for high dimensional transient Fokker–Planck equation in modeling polymeric fluids

Yifei Sun [*], Mrinal Kumar [1]

Department of Mechanical and Aerospace Engineering, University of Florida, Gainesville, FL 32611-6250, USA

ARTICLE INFO

Article history:
Received 8 September 2014
Received in revised form 8 January 2015
Accepted 18 February 2015
Available online 25 February 2015

Keywords:
Fokker–Planck equation
Polymeric fluids
Tensor decomposition
Chebyshev spectral method
Curse of dimensionality
Numerical partial differential equation

ABSTRACT

In this paper, a tensor decomposition approach combined with Chebyshev spectral differentiation is presented to solve the high dimensional transient Fokker–Planck equations (FPE) arising in the simulation of polymeric fluids via multi-bead-spring (MBS) model. Generalizing the authors' previous work on the stationary FPE, the transient solution is obtained in a single CANDECOMP/PARAFAC decomposition (CPD) form for all times via the alternating least squares algorithm. This is accomplished by treating the temporal dimension in the same manner as all other spatial dimensions, thereby decoupling it from them. As a result, the transient solution is obtained without resorting to expensive time stepping schemes. A new, relaxed approach for imposing the vanishing boundary conditions is proposed, improving the quality of the approximation. The asymptotic behavior of the temporal basis functions is studied. The proposed solver scales very well with the dimensionality of the MBS model. Numerical results for systems up to 14 dimensional state space are successfully obtained on a regular personal computer and compared with the corresponding matrix Riccati differential equation (for linear models) or Monte Carlo simulations (for nonlinear models).

1. Introduction

In the study of polymeric fluids, the multi-bead-spring (MBS) model is commonly employed by rheologists to represent coarse-grained molecular configurations. This model is a chain of N beads connected by $N-1$ springs, where the beads represent interaction points with the solvent and springs capture the local stiffness property depending on local stretching [1]: see Fig. 1. For the special case of $N=2$, it is often referred to as the dumbbell model. The configuration of an MBS chain can be specified by the connector vectors $\mathbf{q}_i = \mathbf{r}_{i+1} - \mathbf{r}_i$, $i = 1, \ldots, N-1$, where $\mathbf{r}_i$ is the position vector of the beads with local dimensionality d equal to 1, 2 or 3. A typical flow characterizing the polymeric fluid is the shear flow with the following linear velocity field:

$$\mathbf{v}(\mathbf{x}) = K\mathbf{x}, \quad K = \beta \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{1}$$

where β is the shear rate.

[*] Corresponding author. Graduate Research Assistant.
E-mail addresses: yfsun@ufl.edu (Y. Sun), mrinalkumar@ufl.edu (M. Kumar).

[1] Assistant Professor.

http://dx.doi.org/10.1016/j.jcp.2015.02.026
0021-9991/© 2015 Elsevier Inc. All rights reserved.

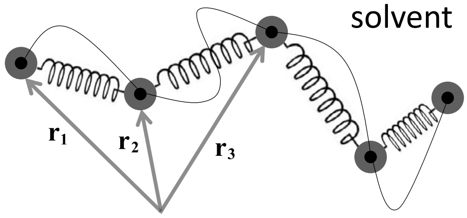


Fig. 1. Illustration of the multi-bead-spring (MBS) model (with 5 beads).

The chain is subject to viscous drag and a Brownian force exerted by the solvent and it also interacts with itself via a potential [2]. Additionally, the initial positions of the beads are not known exactly and this uncertainty is described by the initial pdf ψ_0. Consequently, the configuration of the chain at any given time is probabilistic and can be captured by a time varying pdf ψ satisfying the Fokker–Planck equation (FPE) [3] with global dimensionality $(N-1)d$, which can easily become very large as the number of beads in the chain increases.

The numerical solution of the FPE, in particular the transient FPE continues to remain a formidable challenge. It is well known that “traditional” discretization based numerical methods such as the finite element method (FEM) [4] and the finite difference method (FDM) suffer from the curse of dimensionality [5], i.e. for the same level of accuracy, the degrees of freedom of the approximation (i.e. number of unknowns to solve for) grows exponentially as the dimensionality of the underlying state space increases. Therefore, solving the transient FPE by FEM or FDM for high dimensional systems is computationally quite challenging even when parallel supercomputers are used [6]. Extensive research has been done to develop efficient numerical solvers for high dimensional FPE, such as in Refs. [1,7–9] for FPEs encountered in polymeric fluids applications and [10–14] for nonlinear vibrations.

In related literature, the tensor product representation has been recognized as a key for addressing the curse of dimensionality [15]. Numerical methods based on tensor decompositions have been established for solving high dimensional PDEs [16–18]. The essence of these tensor methods lies in the separation of dimensions, following which expensive high dimensional operations are decoupled into a series of simple one-dimensional operations, at the expense of exacerbating the problem's nonlinearity. Beylkin and Mohlenkamp [16] proposed a method in which the solution to a time independent PDE was sought in the CANDECOMP/PARAFAC decomposition (CPD) [19,20] form by the alternating least squares (ALS) algorithm and FDM was locally used for discretization. Their scheme was extended in [13] for solving the stationary FPE for nonlinear systems in up to 10 dimensions. Tensor methods for time dependent PDEs almost exclusively perform a spatial discretization as the first step while holding the temporal domain as is, then either use dynamical low rank methods [21,22] or time stepping schemes [8,9] to obtain the time evolution. If time stepping is used, the rank of the tensor approximation will keep growing as time progresses. As a result, appropriate truncation must be performed repeatedly to inhibit the growth of complexity, which is a time consuming process. In [8] the tensor train format [23,24] was used for parabolic PDEs in modeling polymeric liquids where the FPE for linear systems in up to 12D was successfully solved. The transient solution was approximated by a spatial discretization, followed by marching in the temporal domain step by step or encapsulating all time steps as a large block system. In Refs. [1,7,25,26], the proper generalized decomposition (PGD) method was developed by Ammar et al. motivated by applications in the kinetic theory of complex fluids, in which the solution of a PDE was constructed in CPD form through a sequence of enrichment steps using the finite element scheme. For time dependent PDEs, they decoupled the temporal domain from the spatial domain and essentially considered it as an additional dimension. In this manner, the strength of the tensor product structure was fully exploited and transient solution was obtained more efficiently. It is important to note that in the PGD enrichment step, the new approximation is constructed by adding a single component rank one tensor at a time. Thus, two successive approximations differ only by a single rank one tensor, due to which the PGD method can be described as a successive rank one approximation. Despite its efficiency within a single enrichment step (since only a single rank one tensor is sought in each iteration regardless of the total approximation rank), the overall strategy of PGD is a matter of debate. An example provided by Kolda [27] where the best rank one approximation of a cubic tensor is not a subset of its best rank two approximation, suggests that components of the best rank k approximation must be solved for simultaneously such as using the ALS algorithm [28–30].

The current paper constitutes a progression of the authors' previous work [13] on the solution of high dimensional stationary FPE. Here, the transient FPE is solved in the CPD structure via the ALS algorithm with the Chebyshev spectral method used for discretization of the combined spatial temporal domain. This is similar to the strategy in the PGD method involving a separation of spatial and temporal domains [7]. This approach benefits from the superior performance of the Chebyshev spectral approach over other differentiation methods for smooth functions on domains of regular geometry [31,32]. Moreover, the transient solution at all times is obtained simultaneously avoiding expensive time-stepping schemes, thereby exploiting the full benefits of the tensor product structure [33,34]. A new method for enforcing the boundary conditions is proposed in which they are expressed as a linear constraint and imposed using a penalty parameter. In addition, the asymptotic behavior of the temporal basis functions is studied and shown to attain steady state for systems admitting a stationary distribution. The proposed method is implemented on a regular personal computer, tested on numerical examples from the field of nonlinear vibrations, and then applied to the simulation of polymeric fluids with multi-bead-spring (MBS) models. Our results indicate that the method works well on dynamical systems from both these fields and the total

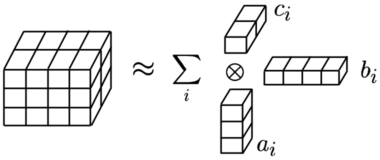


Fig. 2. Illustration of the CP decomposition for a third order tensor.

DOF required for accurate solutions scales favorably. Interested readers can find the usage of the proposed solver for orbital mechanics [35] and nonlinear Bayesian filtering [36].

The remainder of this paper is organized as follows: Section 2 introduces the concepts of CP decomposition. The Chebyshev spectral method and its benefits are discussed in Section 3. Section 4 describes the FPE and some useful solution and validation tools. Section 5 illustrates in detail the proposed solver and numerical examples are provided in Section 6, including systems up to 14 dimensional state space. Finally, a summary and future research directions are provided in Section 7.

2. CANDECOMP/PARAFAC decomposition

In current context, a tensor can be understood as a multi-dimensional array. The order of a tensor is the number of indices needed to specify a single element of that array; whereby first and second order tensors are commonly known as vectors and matrices respectively. A tensor of order P and size $n_1 \times n_2 \times \cdots \times n_P$ is denoted by $\mathcal{F} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_P}$. In numerous applications, a multivariate function can be represented by a tensor whose entries are the function evaluation on the tensor grids. If the resulting tensor is kept in its original form, its storage alone can be difficult in high dimensional cases owing to the exponential growth of its total number of elements ($\prod_{d=1}^{P} n_d$) with respect to P. One remedy is to decompose the tensor into a summation of several rank one tensors. A tensor $\mathcal{F} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_P}$ is of rank one if

$$\mathcal{F} = f_1 \otimes f_2 \otimes \cdots \otimes f_P, \tag{2}$$

where f_d are $n_d \times 1$ vectors for $d = 1, 2, \ldots, P$ and "$\otimes$" is the standard tensor product such that $\mathcal{F}(i_1, i_2, \ldots, i_P) = \prod_{d=1}^{P} f_d(i_d)$ for all $1 \le i_d \le n_d$. The following important tensor decomposition of $\mathcal{F}$ was introduced by Carroll and Chang [19] and Harshman [20], named CANDECOMP/PARAFAC decomposition (CPD):

$$\mathcal{F} \approx \sum_{r=1}^{R} \bigotimes_{d=1}^{P} f_d^r. \tag{3}$$

Note that the above decomposition represents an approximation and is illustrated in Fig. 2. The minimum R for which the "$\approx$" can be replaced with "$=$" is called the rank of tensor $\mathcal{F}$. For simplicity, in Eq. (3) we call R the approximation rank (or just rank for short) of the CP decomposition. The inner product of two tensors $\hat{\mathcal{F}}$ and $\tilde{\mathcal{F}}$ in the CPD form can be computed as

$$\langle \hat{\mathcal{F}}, \tilde{\mathcal{F}} \rangle = \sum_{\hat{r}=1}^{\hat{R}} \sum_{\tilde{r}=1}^{\tilde{R}} \prod_{d=1}^{P} \langle \hat{f}_d^{\hat{r}}, \tilde{f}_d^{\tilde{r}} \rangle. \tag{4}$$

It is customary to define factor matrices of the above decomposition as $F_d = [\, f_d^1 \quad f_d^2 \quad \cdots \quad f_d^R \,]$, $d = 1, 2, \ldots, P$. Once the CPD is found, a tensor of order P can be reconstructed using its P factor matrices. For prescribed accuracy, the total number of elements in the above approximation ($R \sum_{d=1}^{P} n_d$) grows linearly with P, provided that R does not exhibit explosive growth. Fortunately it is observed that in many applications, a tensor of high order can be very well approximated by CP decompositions of relatively low rank, although no theoretical results are available [37]. This makes the CPD structure a good candidate for numerically dealing with high dimensional PDEs despite the fact that finding the *best* low rank approximation for a given tensor is an ill-posed problem [38]. Determining the rank of a tensor is NP-hard [39], hence the selection of R is a tradeoff between accuracy and computational efficiency.

The most commonly used method for computing a CP decomposition of a tensor is the alternating least squares (ALS) algorithm. Given a tensor $\mathcal{G} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_P}$, the problem of finding its CPD of prescribed approximation rank R can be formulated as the following optimization problem

$$\min_{\{f_d^r\}} \mathcal{R}, \quad \mathcal{R} = \left\| \mathcal{G} - \sum_{r=1}^{R} \bigotimes_{d=1}^{P} f_d^r \right\|_F^2, \tag{5}$$

where "F" stands for the Frobenius norm. The unknowns are the vectors f_d^r for $d = 1, 2, \ldots, P$ and $r = 1, 2, \ldots, R$, or equivalently the factor matrices F_d. Although Eq. (5) appears to be a standard nonlinear least squares problem whose objective function is a $2P$th degree polynomial with $R \sum_{d=1}^{P} n_d$ variables, there exist no well established algorithms for its solution [37]. Among existing techniques, alternating least squares (ALS) is arguably the most popular, underlying which is the basic idea of holding all factor matrices except one constant, thus translating the original nonlinear least squares problem into a series of sequential linear least squares problems. The scheme of ALS for the problem in Eq. (5) is as follows:

- Initialize (for example, randomly) $F_d \in \mathbb{R}^{n_d \times R}$, $d = 1, 2, \ldots, P$.
- Cycle for $d = 1, 2, \ldots, P$ until $\sqrt{\frac{\mathcal{R}}{\prod_{d=1}^{P} n_d}} < \epsilon$ or the maximum number of iterations is reached:

$$F_d \leftarrow \mathcal{G}_{(d)}[(F_P \odot \cdots \odot F_{d+1} \odot F_{d-1} \odot \cdots \odot F_1)^T]^\dagger \tag{6}$$

  end of cycle.
- Return F_d for $d = 1, 2, \ldots, P$,

where $(\cdot)_{(d)}$ is the mode-d matricization of a tensor, "$\odot$" is the Khatri–Rao product and $(\cdot)^\dagger$ denotes the Moore–Penrose pseudoinverse of a matrix. It should be noted that although a convergence proof of ALS (even to a local minimum) is not yet available, it can be shown that the objective function of Eq. (5) decreases monotonically [16]. The rate of convergence is often a concern especially as dimensionality increases, leading to problems such as "bottlenecks", "swamps" and "CP-degeneracy" [37]. For a comprehensive introduction to tensor decompositions, the reader is referred to the review papers [28,40]. In application for solving PDEs, the solution is sought in the form of a tensor, and the essential idea of ALS can still be applied for obtaining the CP decomposition as discussed in detail in Section 5.

3. Chebyshev spectral method

Suppose the solution of a linear time dependent PDE lies in a hypercubic spatio-temporal domain. Once the domain is discretized by a tensor grid and the solution is sought in the CPD form, the high dimensional operations of the PDE's operator can now be decoupled into simple one dimensional operations. For these one dimensional operations, the spatial differentiation involved is most commonly performed by FDM in the literature. Also note that it is always desirable to have a sparser grid while maintaining high accuracy. The current application (transient FPE) involves a parabolic PDE that admits smooth solutions over a hypercubic domain, for which a much better choice for approximating derivatives is the Chebyshev spectral method [41] described below.

The Cauchy interpolation error theorem [42] and Chebyshev minimal amplitude theorem [32] state that for a smooth real valued function, the optimal locations of $(N+1)$ grid nodes in terms of polynomial interpolation error are the extrema of the Chebyshev polynomial, i.e., $x_j = \cos\left(\frac{j\pi}{N}\right)$ $j = 0, 1, \ldots, N$. These points can be viewed as the projection of equally distributed nodes on the upper half of a unit circle onto the horizontal line passing through the center of this circle. Thus the Chebyshev extrema points are not evenly distributed, but denser near the two ends than in the middle region. Derivatives can then be evaluated by differentiating the interpolating polynomial at the desired point. For instance, the first order differentiation matrix D_N is of size $(N+1) \times (N+1)$ given by [31]:

$$\begin{aligned}
&(D_N)_{00} = \frac{2N^2+1}{6}, \quad (D_N)_{NN} = -\frac{2N^2+1}{6},\\
&(D_N)_{jj} = \frac{-x_j}{2(1-x_j^2)}, \quad j = 1, \ldots, N-1,\\
&(D_N)_{ij} = \frac{c_i(-1)^{i+j}}{c_j(x_i - x_j)}, \quad i \neq j,\ i, j = 1, \ldots, N-1,
\end{aligned}$$

where,

$$c_i = \begin{cases} 2 & i = 0 \text{ or } N,\\ 1 & \text{otherwise} \end{cases}$$

For solving the FPE we also need second order derivatives, which are relatively easy to obtain. For instance, the operator $\frac{\partial^2}{\partial x^2}$ can simply be approximated as the square of D_N. The Chebyshev potential [31] can be used to show that in the current application, the above spectral derivative converges geometrically, which significantly outperforms both FDM and FEM. In other words, a differentiation matrix of significantly smaller size is sufficient for approximating the same one dimensional operator with equal accuracy. This is crucial for dealing with the curse of dimensionality while solving the FPE. For more theoretical details on the Chebyshev spectral method and its application in tensor form, we recommend the reader to see the books [31,32] and papers [13].

Note that as opposed to the attractive band-limited differentiation matrix of FDM, the Chebyshev spectral differentiation matrix is dense. However, since only one-dimensional differentiation is required for the tensor based method here, the dense structure does not have an adverse impact. Also note that if used in the temporal domain, its dense matrix structure would lead to expensive time stepping schemes, whereby traditional numerical schemes only use the Chebyshev spectral differentiation in the spatial domain. However, in the current tensor approach with complete separation of temporal and spatial domains, there is no need to perform time stepping for solving the transient problem. Thus the superior accuracy of Chebyshev spectral differentiation can be extended to temporal differentiation without any penalty, as shown in Section 5.

4. Fokker–Planck equation and some validation tools

In this section, we first review the Fokker–Planck equation. For linear systems with Gaussian initial uncertainty, we describe its solution via the corresponding matrix Riccati differential equation. As a universal validation tool, especially for nonlinear dynamical systems, we describe the Monte-Carlo simulation (MCS) approach via Milstein's stochastic integration. These validation techniques will be used in Section 6 for evaluating the performance of the proposed tensor approach.

4.1. Fokker–Planck equation

Consider a P-dimensional nonlinear dynamical system with initial condition uncertainty and white noise excitation, modeled by the following stochastic differential equation:

$$d\mathbf{x} = \mathbf{f}(t,\mathbf{x})dt + \mathbf{g}(t,\mathbf{x})d\mathbf{B}(t), \quad \mathbf{x} \in \mathbb{R}^P, \tag{7}$$

where $\mathbf{B}(t)$ denotes an M dimensional Brownian motion process with zero mean and covariance $\mathbf{Q}t$. The $\mathbf{f}(t,\mathbf{x}) : [0,\infty) \times \mathbb{R}^P \to \mathbb{R}^P$ and $\mathbf{g}(t,\mathbf{x}) : [0,\infty) \times \mathbb{R}^P \to \mathbb{R}^{P\times M}$ are deterministic functions. The uncertainty in the state $\mathbf{x}(t)$ can be quantified by its time varying probability density function (pdf) $\mathcal{W}(t,\mathbf{x})$. The uncertainty in initial conditions is given by the pdf $\mathcal{W}(t_0,\mathbf{x}) = \mathcal{W}_0(\mathbf{x})$. The FPE corresponding to Eq. (7) is a linear second order parabolic PDE that captures the time evolution of the state pdf:

$$\frac{\partial}{\partial t}\mathcal{W}(t,\mathbf{x}) = \mathcal{L}_{\mathcal{FP}}[\mathcal{W}(t,\mathbf{x})], \tag{8}$$

where $\mathcal{L}_{\mathcal{FP}}$ is the Fokker–Planck operator given by

$$\mathcal{L}_{\mathcal{FP}} = \left[-\sum_{i=1}^{\mathrm{P}} \frac{\partial}{\partial x_i}\mathrm{D}_i^{(1)}(\cdot) + \sum_{i=1}^{\mathrm{P}}\sum_{j=1}^{\mathrm{P}} \frac{\partial^2}{\partial x_i \partial x_j}\mathrm{D}_{ij}^{(2)}(\cdot) \right],$$

$$\mathrm{D}^{(1)}(t,\mathbf{x}) = \mathbf{f}(t,\mathbf{x}), \qquad \mathrm{D}^{(2)}(t,\mathbf{x}) = \frac{1}{2}\mathbf{g}(t,\mathbf{x})\,\mathrm{Q}\,\mathbf{g}^{\mathrm{T}}(t,\mathbf{x}). \tag{9}$$

In Eq. (9), $\mathrm{D}^{(1)}$ and $\mathrm{D}^{(2)}$ are called drift vector and diffusion matrix respectively. The vector $\mathrm{D}^{(1)}$ captures drifting apart of the mean of the propagated $\mathcal{W}(t,\mathbf{x})$ from the propagated mean of $\mathcal{W}_0(\mathbf{x})$. Generally, this drift increases with degree of nonlinearity of underlying dynamics (i.e. $\mathbf{f}(t,\mathbf{x})$). The matrix $\mathrm{D}^{(2)}$ captures the spreading out of the substantial portion of the $\mathcal{W}(t,\mathbf{x})$ over the state-space. In case that $\mathbf{g}(t,\mathbf{x}) = 0$, i.e., the underlying governing dynamics is deterministic, the source of uncertainty lies purely in initial condition and the FPE reduces to the Liouville equation [43]. This paper is concerned with the transient solution of Eq. (9). As a valid pdf, the transient solution of FPE must satisfy the following constraints at all times:

$$\lim_{\mathbf{x}\to\infty} \mathcal{W}(t,\mathbf{x}) = 0, \tag{10}$$

$$\int_{\Omega} \mathcal{W}(t,\mathbf{x})d\mathbf{x} = 1. \tag{11}$$

Eq. (10) and Eq. (11) are called the vanishing boundary condition and normality condition respectively. Any discretization based method requires a compact domain for implementation, due to which the vanishing boundary condition is imposed on a conservatively chosen "large-enough" domain.

Besides the constraints described in Eqs. (10) and (11), $\mathcal{W}(t,\mathbf{x})$ should also be non-negative on the entire spatio-temporal domain. In this paper, although non-negativity is not actively enforced, the numerical solutions obtained are "accurate enough" to preserve this property (negative pdf values with magnitude less than 10^{-7} occasionally appear in certain small regions in some examples).

4.2. Linear systems: solution by matrix Riccati differential equation

For a linear system with constant coefficients, Eq. (7) reduces to

$$d\mathbf{x} = \mathbf{A}\mathbf{x}dt + \mathbf{g}d\mathbf{B}(t), \tag{12}$$

where $\mathbf{A}$ and $\mathbf{g}$ are constant matrices of size $P \times P$ and $P \times M$ respectively. It is known that the solution to FPE for the linear system in Eq. (12) with Gaussian initial pdf $\mathcal{W}_0(\mathbf{x})$ remains Gaussian for all times. In other words, $\mathcal{W}(t,\mathbf{x})$ is Gaussian $\mathcal{N}(\boldsymbol{\mu}(t), \boldsymbol{\Sigma}(t))$ with time-varying mean $\boldsymbol{\mu}(t)$ and covariance $\boldsymbol{\Sigma}(t)$ that satisfy the following ordinary differential equations (ODEs)

$$\dot{\boldsymbol{\mu}} = \mathbf{A}\boldsymbol{\mu}, \tag{13}$$

$$\dot{\boldsymbol{\Sigma}} = \mathbf{A}\boldsymbol{\Sigma} + \boldsymbol{\Sigma}\mathbf{A}^T + \mathbf{g}\mathbf{Q}\mathbf{g}^T, \tag{14}$$

where Eq. (14) is a special form of the matrix Riccati differential equation. If only the stationary solution is desired (when it exists), Eq. (13) yields $\boldsymbol{\mu}_\infty = \mathbf{0}$, while the stationary covariance matrix can be obtained from the continuous time algebraic Riccati equation, whose solution has been extensively studied. In high dimensional cases, 2D marginal solutions are often desired for visualization. The ensuing computationally unattractive high dimensional numerical integration can be avoided as described below: suppose the x_1–x_2 marginal is desired and let the solution at a certain time be

$$\mathcal{W}(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^P \det(\boldsymbol{\Sigma})}} \exp\left(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right). \tag{15}$$

The information matrix $\boldsymbol{\Sigma}^{-1}$ can be split as follows to isolate the target states (x_1 and x_2):

$$\boldsymbol{\Sigma}^{-1} = \begin{pmatrix} \boldsymbol{\Sigma}_1 & \boldsymbol{\Sigma}_2 \\ \boldsymbol{\Sigma}_2^T & \boldsymbol{\Sigma}_3 \end{pmatrix}, \tag{16}$$

where $\boldsymbol{\Sigma}_1$ is a matrix of size 2×2 and the remaining components can be identified accordingly. It can be shown after some algebra that the desired marginal pdf is given by

$$\mathcal{W}(\mathbf{x}_r) = \frac{1}{2\pi\sqrt{\det(\boldsymbol{\Sigma})\det(\boldsymbol{\Sigma}_3)}} \exp\left(-\frac{1}{2}(\mathbf{x}_r-\boldsymbol{\mu}_r)^T \boldsymbol{\Sigma}_4(\mathbf{x}_r-\boldsymbol{\mu}_r)\right), \tag{17}$$

where $\mathbf{x}_r = [x_1, x_2]^T$, $\boldsymbol{\mu}_r = [\mu_1, \mu_2]^T$, $\boldsymbol{\Sigma}_4 = \boldsymbol{\Sigma}_1 - \boldsymbol{\Sigma}_2\boldsymbol{\Sigma}_3^{-1}\boldsymbol{\Sigma}_2^T$. In fact, above formulas can be easily generalized for any marginal pdf of dimensionality P' with $1 \le P' \le P$. A rearrangement in the ordering of the states may be needed to ensure that the dimensions of interest for the marginal appear first in the state vector, followed by the other states that are to be "integrated out".

4.3. Solution by Monte Carlo simulation

A commonly employed tool for studying uncertainty flow is the so-called Monte-Carlo simulation (MCS), primarily because of its relative simplicity and universal applicability to general nonlinear systems with arbitrary initial PDFs. The essential idea behind MCS is to first create a "sufficiently large" sample from the initial PDF $\mathcal{W}_0(\mathbf{x})$, followed by stochastic integration of the system dynamics (Eq. (7)) for each initial sample. Finally, the time varying statistics are computed using the propagated particles. Estimating the marginal PDF solution is rather easy in MCS by taking the histogram of the resulting samples on the dimensions of interest. MCS is motivated by the law of large numbers and unfortunately no guidelines exist for how large the initial sample must be in order to obtain sufficiently accurate statistics of the state, especially vis-à-vis the degree of system nonlinearity and the length of the time of propagation.

For simplicity, suppose the $\mathbf{B}(t)$ of stochastic system in Eq. (7) is an M dimensional standard Brownian motion process, i.e., $\mathbf{Q}$ is identity matrix of size $M \times M$, and we have

$$\mathbf{x}(t) = \mathbf{x_0} + \int_0^t \mathbf{f}(s, \mathbf{x}(s))ds + \int_0^t \mathbf{g}(s, \mathbf{x}(s))d\mathbf{B}(s), \quad 0 \le t \le T, \tag{18}$$

where the stochastic integral on the RHS is with respect to Brownian motion in the Itô sense. To estimate these integrals, discretize the time as $t_j = j\triangle t$ for $j = 0, \dots, L$ where $\triangle t = \frac{T}{L}$. Then the Milstein stochastic integration scheme [44] for Eq. (18) is given by

$$\begin{aligned} x_{i,j+1} &= x_{i,j} + f_i(t_j, \mathbf{x}_j)\triangle t + \sum_{k=1}^{M} g_{i,k}(t_j, \mathbf{x}_j)(B_k(t_{j+1}) - B_k(t_j)) \\ &\quad + \sum_{k_1=1}^{M}\sum_{k_2=1}^{M}\sum_{l=1}^{P} g_{l,k_1}\frac{\partial g_{i,k_2}}{\partial x_l} I_j(k_1, k_2), \quad j = 0, \dots, L-1, \quad i = 1, \dots, P \end{aligned} \tag{19}$$

where $x_{i,j}$ is the ith component of $\mathbf{x}(t_j)$, $\mathbf{B}(t_{j+1}) - \mathbf{B}(t_j) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}\triangle t)$ and $I_j(k_1, k_2)$ is the double stochastic integral given as [45]

$$\begin{aligned} I_j(k_1, k_2) &= \int_{t_j}^{t_j+\triangle t}\int_{t_j}^{s} dB_{k_1}(r)dB_{k_2}(s) \\ &\approx \sum_{h=0}^{H-1}[B_{k_1}(t_{h,j}) - B_{k_1}(t_{0,j})][B_{k_2}(t_{h+1,j}) - B_{k_2}(t_{h,j})], \end{aligned} \tag{20}$$

where $t_{h,j} = t_j + \frac{h\triangle t}{H}$ for $h = 0, \dots, H$ that equally divide $[t_j, t_{j+1}]$ into H intervals [46].

Note that having the last term on the RHS of Eq. (19) helps the method gain strong order convergence equal to 1 compared with the Euler–Maruyama method of order $\frac{1}{2}$ [47]. In addition, for additive noise, i.e., all entries of matrix $\mathbf{g}$ in Eq. (7) are either constant or functions of time only, this last term vanishes which renders the Milstein scheme the same as the Euler–Maruyama method [48].

MCS is especially useful as a validation tool when closed-form solutions are not available, which is the most common scenario in real life. It can also be used to determine the appropriate size of the solution domain for other numerical methods.

5. A Chebyshev spectral tensor decomposition solution of the transient FPE

As mentioned in Section 1, the transient solution of PDEs in the existing literature is almost exclusively obtained via time stepping schemes for the corresponding elliptic problem even when a tensor product structure is adopted. This strategy for time dependent PDEs is naturally inherited from traditional numerical methods where spatial dimensions remain coupled. Unfortunately, if time stepping is used in a tensor approach, the rank of approximation would keep growing as time progresses. Appropriate truncation must therefore be performed repeatedly to inhibit the growth of complexity.

A more effective way of solving time dependent PDEs in tensor product form is to not only separate the spatial dimensions as in the stationary problem, but also the temporal dimension from the spatial ones. Essentially, time is treated as an additional spatial dimension, whereby the full power of the tensor product structure can be unleashed and computational complexity would grow in a benign manner, similar to the stationary problem studied in our previous work [13].

5.1. Separation of spatial and temporal domains

The transient solution $\mathcal{W}(t,\mathbf{x})$ of FPE in Eq. (8) is sought in the following CPD form:

$$\mathcal{W}(t,\mathbf{x}) \approx \mathcal{U}(t,\mathbf{x}) = \sum_{l=1}^{R_U} \left[\left(\prod_{d=1}^{P} u_d^l(x_d) \right) T^l(t) \right], \tag{21}$$

where $u_d^l(x_d)$ and $T^l(t)$ for $l = 1, 2, \ldots, R_U$ are called spatial and temporal basis functions respectively. Note that each of these basis functions has only one dependent variable and can be discretized by the Chebyshev spectral method. This results in vectors u_d^l and T^l with size $n_d \times 1$ and $n_t \times 1$ respectively. For convenience, the discretized version of $\mathcal{U}$ is denoted by $\mathbb{U} = \sum_{l=1}^{R_U} \bigotimes_{d=1}^{P+1} f_d^l$, where f_d^l includes both u_d^l and T^l. Correspondingly, the Fokker–Planck (FP) operator $\mathcal{L}_{\mathcal{FP}}$ can be written in tensorized form as

$$\mathbb{A} = \sum_{i_A=1}^{R_A} \left[\left(\bigotimes_{d=1}^{P} A_d^{i_A} \right) \otimes I_t \right], \tag{22}$$

where $A_d^{i_A}$ is $n_d \times n_d$ matrix, and I_t is $n_t \times n_t$ identity matrix. The $A_d^{i_A}$ represents the action of the FP operator on each decoupled spatial domain, while the identity matrix I_t simply signifies that the FP operator does not involve any temporal derivatives and the related basis functions must not be modified. Writing the FP operator $\mathcal{L}_{\mathcal{FP}}$ in the form of Eq. (22) is straightforward if the $\mathbf{f}(t,\mathbf{x})$ and $\mathbf{g}(t,\mathbf{x})$ in Eq. (7) are given as separable functions. When this is not the case, additional function approximation step can be performed to obtain a decent separable representation of the $\mathbf{f}(t,\mathbf{x})$ and $\mathbf{g}(t,\mathbf{x})$ by algorithm discussed in Section 2. Interested readers can find more details related to individual terms in Eq. (22) from Ref. [13]. Similarly, the operator $\frac{\partial}{\partial t}$ in Eq. (8) can be approximated in CPD form as

$$\frac{\partial}{\partial t} \approx \left(\bigotimes_{d=1}^{P} I_d \right) \otimes D_t \triangleq \mathbb{A}_t, \tag{23}$$

where D_t is the $n_t \times n_t$ first order differentiation matrix corresponding to the temporal domain and I_d are $n_d \times n_d$ identity matrices corresponding to the spatial dimensions. The easiest way to construct D_t is via a finite difference scheme, either of first or second order accuracy wherein the user can employ a combination of centered and one-sided stencils for the boundary and interior nodes. Each such strategy corresponds to a different time stepping scheme in the traditional discrete differentiation sense and their stability properties are well studied [41]. However in the context of the current proposed method, based on our numerical experience with the FPE, none of these finite difference schemes for D_t were able to yield sufficiently accurate transient solutions, no matter how fine a temporal grid was used. At this point, the theoretical reason behind this behavior is not clear.

Another possible choice for D_t is the Chebyshev spectral differentiation matrix as used for spatial differentiation for the stationary FPE by the authors in Ref. [13]. In traditional numerical methods, the Chebyshev spectral differentiation is usually limited to the spatial domain, because its dense matrix structure necessitates expensive time stepping schemes. However in the current tensor method with complete separation of temporal and spatial domains, there is no need for time stepping and the superior performance of Chebyshev spectral differentiation can be extended to temporal differentiation without penalty. Moreover (and importantly), no stability issues were encountered upon its use with temporal grids of varying fineness.

5.2. Asymptotic behavior of temporal basis functions

The asymptotic behavior of the temporal basis functions is of particular interest. For systems that admit a stationary solution, the form of the approximation ($\mathcal{U}(t,\mathbf{x})$) in Eq. (21) suggests that the temporal basis functions $T^l(t)$ must eventually assume constant values. This behavior is captured by the following proposition.

Proposition 1. *Let the transient solution be written in the form of Eq. (21). Then for a system admitting a stationary solution, the temporal basis functions $T^l(t)$ approach constant values as t becomes sufficiently large.*

Proof. Let $t' \geq t_s$ where t_s is sufficiently large such that $\mathcal{U}(t',\mathbf{x})$ is viewed to be the stationary solution, i.e., $\left.\frac{\partial \mathcal{U}(t,\mathbf{x})}{\partial t}\right|_{t'} = 0$. Upon differentiating Eq. (21),

$$\sum_{l=1}^{R_U}\left[\left(\prod_{d=1}^{P} u_d^l(x_d)\right)\dot{T}^l(t')\right] = 0. \tag{24}$$

The above equation must hold for arbitrary values of $\mathbf{x} \in \Omega$ (i.e. the solution domain). Thereby, consider an arbitrary set of points $\mathbf{x}_i = (x_1^i, x_2^i, \cdots, x_P^i)$ for $i = 1, 2, \cdots, R_U$, and evaluate Eq. (24) at these points to get

$$\underbrace{\begin{pmatrix} \prod_{d=1}^P u_d^1(x_d^1) & \cdots & \prod_{d=1}^P u_d^{R_U}(x_d^1) \\ \vdots & \ddots & \vdots \\ \prod_{d=1}^P u_d^1(x_d^{R_U}) & \cdots & \prod_{d=1}^P u_d^{R_U}(x_d^{R_U}) \end{pmatrix}}_{=\mathbf{M_1}} \underbrace{\begin{pmatrix} \dot{T}^1(t') \\ \vdots \\ \dot{T}^{R_U}(t') \end{pmatrix}}_{=\mathbf{M_2}} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{25}$$

It is relatively easy to ensure that the $R_U \times R_U$ sized matrix $\mathbf{M_1}$ appearing above is full ranked because of the arbitrary nature of the selected points $\mathbf{x}_i$. Thus the vector $\mathbf{M_2}$ must be trivial, i.e., $T^l(t')$ is constant for $l = 1, 2, \cdots, R_U$. □

5.3. The ALS formulation for FPE

So far, the FPE has been reduced to the following linear algebraic form:

$$\mathbb{A}'\mathbb{U} = 0, \tag{26}$$

where $\mathbb{A}' = (\mathbb{A} + \mathbb{A}_t)$ (see Eqs. (22) and (23)) and can be rewritten as $\mathbb{A}' = \sum_{i_A=1}^{R_{A'}} \bigotimes_{d=1}^{P+1} G_d^{i_A}$ for convenience. Note that Eq. (26) is linear in $\mathbb{U}$ as a whole but nonlinear with respect to the individual f_d^l's. Eq. (26) can be solved by transformation into the following optimization problem:

$$\min_{\{f_d^l\}} \parallel \mathbb{A}'\mathbb{U} \parallel_F^2 . \tag{27}$$

The necessary condition for minimization is

$$\frac{\partial \langle \mathbb{A}'\mathbb{U}, \mathbb{A}'\mathbb{U}\rangle}{\partial f_d^l} = 0, \tag{28}$$

for $d = 1, 2, \ldots, P+1$ and $l = 1, 2, \ldots, R_U$. Collecting terms in Eq. (28) for all l's and fixed dimension d, we have

$$\underbrace{\begin{pmatrix} \mathbf{M}_{1,1} & \cdots & \mathbf{M}_{1,R_U} \\ \vdots & \ddots & \vdots \\ \mathbf{M}_{R_U,1} & \cdots & \mathbf{M}_{R_U,R_U} \end{pmatrix}}_{=\mathbf{M}} \begin{pmatrix} f_d^1 \\ \vdots \\ f_d^{R_U} \end{pmatrix} = 0, \tag{29}$$

where $\mathbf{M}_{i,j}$ is a submatrix of the block matrix $\mathbf{M}$, given by

$$\mathbf{M}_{i,j} = \sum_{i_A=1}^{R_{A'}} \sum_{j_A=1}^{R_{A'}} (G_d^{j_A})^T G_d^{i_A} \prod_{k\neq d} \langle G_k^{i_A} f_k^j, G_k^{j_A} f_k^i \rangle \tag{30}$$

The general ALS formulation for solving an equation in the form of Eq. (26) was first provided in Ref. [16] (in an element-wise form different from above), and a detailed derivation is provided in [13]. In the ALS framework, Eq. (29) is reduced to a linear system and solved sequentially for each dimension in an iterative manner. Therefore once the user prescribes R_U, the number of unknowns in a single iteration is independent of the dimensionality P. Moreover, a significant amount of computation is saved by noting that only a small portion of the terms involved need to be recalculated for different d

and R_U. The overall dominant complexity of this ALS scheme is $\mathcal{O}((P+1)R_U^3 n^3)$ (see Ref. [16] for more details) where n is the number of nodes in each spatial and temporal dimension. In implementation, we begin with $R_U = 1$ and random initial values for f_d^1. For each subsequent enrichment step ($R_U = R_U + 1$), ALS initializes with a random rank one tensor added to the final iteration of the previous step. This procedure is repeated until stopping criteria are met. Because of the dependence of the current enrichment step on the previous one (in terms of its starting point) as well as the dependence between consecutive ALS iterations, implementation on a parallel computing architecture is possible only within a single ALS iteration. In the next section, we discuss the enforcement of constraints described in Section 4 which are essential for the ALS scheme to return a nontrivial answer.

5.4. *Enforcement of constraints*

Besides the governing FPE, several other ingredients are needed for solution:

- Initial conditions: In this work, the initial conditions are enforced using a penalty parameter as described below in Section 5.4.1.
- Vanishing boundary conditions: A new method is presented in which the vanishing boundary conditions are posed as a linear constraint and enforced using a penalty parameter. It will be shown that this method improves the quality of results as compared with Ref. [13].
- Normality condition: Interestingly, its enforcement is not really needed. In Ref. [13], it was enforced primarily to transform the discretized problem into a non-homogeneous equation, thereby avoiding the need to search the null space of the resulting tensor equation. In the current paper, the penalty terms corresponding to the initial and boundary conditions described above already achieve this objective. At the same time, it was found through numerical experiments that imposing the normality constraint significantly improves the overall accuracy of the solution, especially when the approximation rank is relatively small. This is useful because its inclusion entails essentially a negligible additional computational cost.

Putting the above elements together, the transient FPE can be posed as the following minimization problem:

$$\min_{\{u_k^r, T^r\}} \langle \mathcal{R}, \mathcal{R} \rangle = \| \mathbb{A}'\mathbb{U} \|_F^2 + \frac{\alpha_1}{2} \| \mathbb{P}\mathbb{U} - \mathbb{U}_0 \|_F^2 + \frac{\alpha_2}{2} \| \mathbb{M}\mathbb{U} \|_F^2 + \frac{\alpha_3}{2} \| \mathbb{B}\mathbb{U} - \mathbb{Q} \|_F^2, \tag{31}$$

where the terms in the RHS reflect the equation error in the FPE and errors from the enforcement of initial conditions, boundary conditions and normality constraint respectively, described in greater detail below. The root mean value of the residual $\mathcal{R}$ is $\sqrt{\frac{\mathcal{R}}{n_t \prod_{d=1}^{P} n_d}}$, which indicates the overall accuracy of the approximated solution and is abbreviated as RMR. An ALS procedure similar to the one described in Section 5.3 above can be used to solve the minimization problem of Eq. (31), where essentially the procedure is applied to each term in the RHS. Interested readers can find formula for the non-homogeneous version of Eq. (26) in Refs. [13,16].

5.4.1. *Initial conditions*

The initial condition of the approximation $\mathcal{U}(t, \mathbf{x})$ can be given in CPD form as follows

$$\mathbb{U}_0 = \sum_{l=1}^{R_{U_0}} \left(\bigotimes_{d=1}^{P} u_{0,d}^l \right) \otimes [1], \tag{32}$$

where the scalar [1] is used purely to maintain the correct order of tensor $\mathbb{U}_0$ since it has no contribution from the temporal bases. If $\mathbb{U}_0$ is not available in CPD form, an approximation can be obtained using the standard ALS algorithm described above, for example see Ref. [28] or [13]. The effect of tensor $\mathbb{P}$ in the "$\mathbb{P}\mathbb{U}$" term in Eq. (31) is to restrict the evaluation of the temporal basis functions of $\mathbb{U}$ to the first node of temporal domain ($t = 0$). It is given by

$$\mathbb{P} = \left(\bigotimes_{d=1}^{P} I_{\mathbf{x}_d^v} \right) \otimes e_1, \tag{33}$$

where e_1 is $n_t \times 1$ row vector $[1, 0, \cdots, 0]$ and $I_{\mathbf{x}_d^v}$ is an identity matrix whose column size is the same as that of vector $u_{0,d}^l$.

5.4.2. *Boundary conditions*

In Ref. [13], the vanishing boundary conditions (Eq. (10)) were enforced by simply deleting the boundary rows and columns of all relevant matrices, which we refer to as the "deletion method". The underlying logic is that all spatial basis functions should vanish at the boundary of the spatial domain. This is an effective approach and can be easily extended to the transient problem. However, it imposes a much stronger condition than what is actually stipulated by Eq. (10), which only requires the *summation* of the terms $\prod_{d=1}^{P} u_d^l(x_d) T^l(t)$ in Eq. (21) to be 0 at the boundary. Motivated by this fact, we propose a new relaxed method for boundary condition enforcement, which is equivalent to Eq. (10) and does not require individual basis functions ($u_d^l(x_d)$) to vanish at the boundaries. Moreover, this new method is significantly more general in that it can easily impose other boundary conditions, such as arbitrary Dirichlet or Neumann boundary conditions. Numerical examples in Section 6 will show that ensuing solutions are significantly more accurate. The main idea is to consider the boundary condition as a linear constraint $\mathbb{M}\mathbb{U} = 0$ where $\mathbb{M}$ is given by

$$\mathbb{M} = \sum_{i_M=1}^{P} \left[\left(\bigotimes_{d=1}^{i_M-1} I'_{\mathbf{x}_d^v} \right) \otimes I''_{\mathbf{x}_{i_M}^v} \otimes \left(\bigotimes_{d=i_M+1}^{P} I_{\mathbf{x}_d^v} \right) \otimes I_t \right], \tag{34}$$

where $I_{\mathbf{x}_d^v}$ is the same as in Eq. (33), $I'_{\mathbf{x}_d^v}$ is diag([0, 1, $\cdots$, 1, 0]) corresponding to vector $\mathbf{x}_d^v$, and $I''_{\mathbf{x}_{i_M}^v}$ is diag([1, 0, $\cdots$, 0, 1]) corresponding to vector $\mathbf{x}_{i_M}^v$. The essential effect of $\mathbb{M}$ in $\mathbb{M}\mathbb{U}$ is to evaluate the spatial boundary values of $\mathbb{U}$. Since the rank of $\mathbb{M}$ is merely the system's dimensionality P and it only contains diagonal matrices, the computational burden of imposing $\mathbb{M}\mathbb{U} = 0$ is rather small.

5.4.3. Normality constraint

In Eq. (31), the normality condition is captured by the term $\mathbb{B}\mathbb{U} - \mathbb{Q}$ where

$$\mathbb{B} = \left(\bigotimes_{d=1}^{P} b_d \right) \otimes I_t \quad \text{and} \quad \mathbb{Q} = \left(\bigotimes_{d=1}^{P} [1] \right) \otimes e, \tag{35}$$

where, b_d contains the Clenshaw–Curtis quadrature and I_t are the same as in Eq. (26), and e is the $n_t \times 1$ column vector $[1, \cdots, 1]^T$. Note that the scalar terms "[1]" in above equation serve to maintain the correct order of the tensor $\mathbb{Q}$ as in Eq. (32). For the rank one tensors $\mathbb{B}$ and $\mathbb{Q}$ with component matrices shown above, the computational burden of imposing this normality constraint is insignificant.

6. Numerical examples

In this section, the proposed numerical solver will first be tested on transient FPEs encountered in the study of nonlinear vibrations with examples up to 14 dimensional state space, then it will be applied to the simulation of polymeric fluids, namely a 3D dumbbell example and a 9D multi-spring example. These numerical experiments illustrate the effectiveness of the numerical solver for relatively high dimensional dynamical systems. Note that for all examples in this section, the penalty parameters $\alpha_1, \alpha_2, \alpha_3$ in Eq. (31) are chosen to be 100.

6.1. Nonlinear vibrations systems

6.1.1. Example 1: 2-state system

Consider the following two-state nonlinear oscillator [49]:

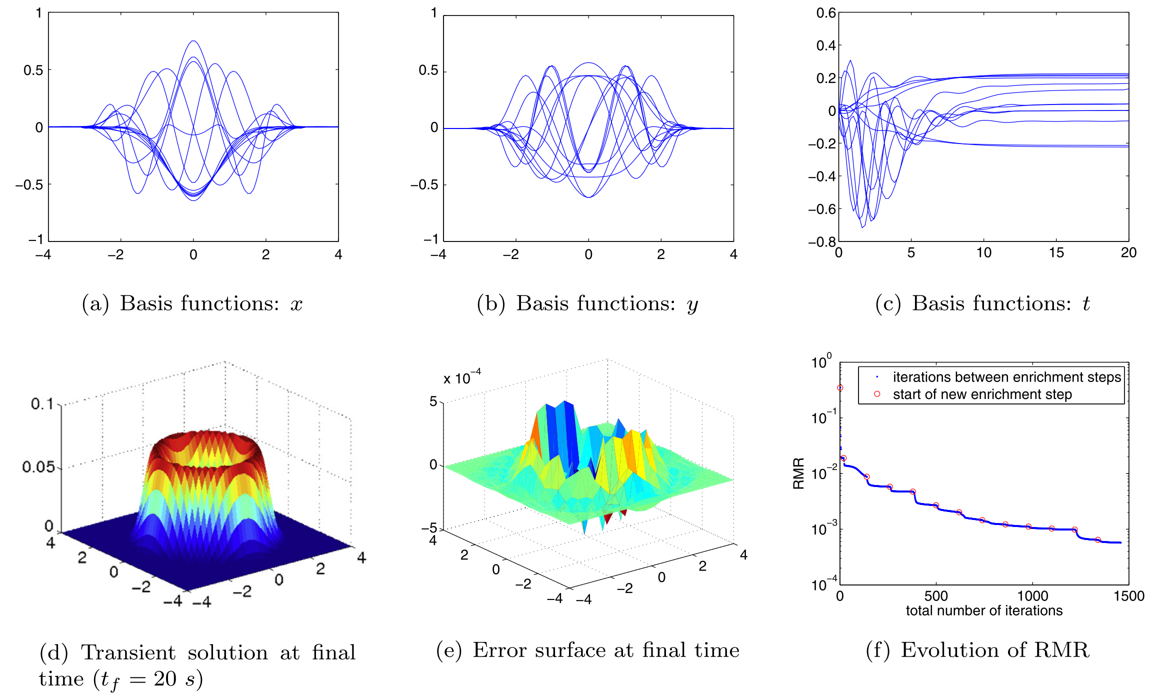
$$\ddot{x} + b\dot{x} + x + a(x^2 + \dot{x}^2)\dot{x} = g\xi(t), \tag{36}$$

where, $a = 0.125, b = -0.5, g = 1$ and $\xi(t)$ is a white noise term with intensity $Q = 0.4$. We assume that the initial distribution of the state $(x, \dot{x})$ is a two dimensional Gaussian pdf with zero mean and covariance of $0.5I$ where I is a 2×2 identity matrix. Unlike the stationary solution, the analytical form of the transient solution for this system is not available. We seek the transient solution for the time duration $t \in [0, 20]$ s using 29 nodes per spatial domain and 32 temporal node. The final time considered (20 s) is sufficiently large for the system to settle to a "near stationary state" (see Fig. 3(c)). Following 13 enrichment steps (consuming 126 seconds, the total DOF is $(29 \times 2 + 32) \times 13 = 1170$), the obtained RMR is 5.734×10^{-4} and the mean squared error (MSE) of the final time transient response (at $t_f = 20$ s) with respect to the true stationary solution is 7.294×10^{-9}. The spatial and temporal basis functions, transient solution as well as its error surface at the final time (estimated by comparing with the known analytical stationary solution), and evolution of RMR are given in Fig. 3. Note that in agreement with Proposition 1, the temporal basis functions settle to constant values after 16 seconds. Also, the RMR decays monotonically as additional enrichment steps are performed.

6.1.2. Example 2: 4-state system

Next consider the 4-state nonlinear isolating suspension model given in [50],

$$\dot{x_1} = x_3, \qquad \dot{x_2} = x_4,$$
$$\dot{x_3} = -ax_3 - \frac{1}{M_0}\frac{\partial V}{\partial x_1} + \xi_1,$$
$$\dot{x_4} = -bx_4 - \frac{1}{I_0}\frac{\partial V}{\partial x_2} + \xi_2.$$



(a) Basis functions: x

(b) Basis functions: y

(c) Basis functions: t

(d) Transient solution at final time ($t_f = 20\ s$)

(e) Error surface at final time

(f) Evolution of RMR

Fig. 3. Results for the 2-state system.

The coupled potential function V is chosen as $V(x_1, x_2) = k_1 x_1^2 + k_2 x_2^2 + \epsilon(\lambda_1 x_1^4 + \lambda_2 x_2^4 + \mu x_1^2 x_2^2)$. System parameters are $a = 0.5, b = 1, k_1 = 0.5, k_2 = -0.5, \epsilon = 0.5, \lambda_1 = 0.25, \lambda_2 = 0.125, \mu = 0.375$. The noise process is two-dimensional with intensities $2D_1(\sim \xi_1) = 4$ and $2D_2(\sim \xi_2) = 8$. Parameters have been selected such that $\frac{D_1 M_0}{a} = \frac{D_2 I_0}{b} = 2T$, which guarantees the existence of a stationary solution.

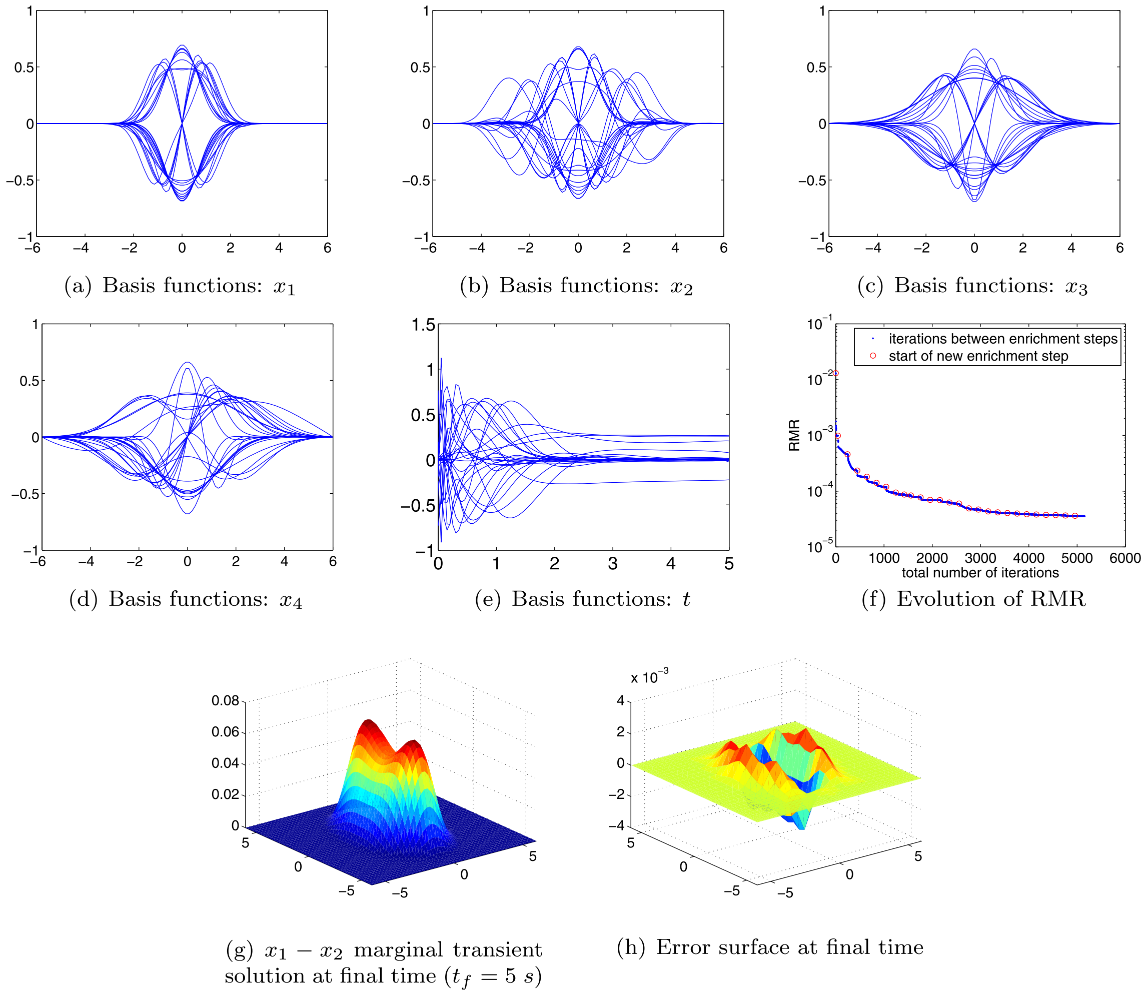
We assume that the initial pdf is Gaussian with zero mean and covariance $0.5I$ where I is a 4×4 identity matrix. Using 33 spatial nodes per dimension and 22 temporal nodes and following 27 enrichment steps (with DOF $= 4158$) that consumed about 3 hours of computation time, an RMR of 3.542×10^{-5} was obtained. The spatial and temporal basis functions, evolution of RMR, the x_1–x_2 marginal transient solution and its error surface at the final time (estimated by comparing with the known analytical stationary solution, with MSE $= 2.84 \times 10^{-7}$) are shown in Fig. 4. As expected, the transient solution requires a significantly greater number of enrichment steps compared to the stationary version (see Ref. [13]). The RMR shows a desirable gradual decay and no sluggishness in the convergence is observed, which can be attributed to the linear independence of the temporal basis functions. Further details of this behavior is provided in Section 6.1.3.

6.1.3. Example 3: 14-state system

To demonstrate the scalability of the proposed method, we extend the previous example to solve the FPE for a generalized high dimensional linear oscillator given by the following template:

$$\begin{aligned} \dot{x}_i &= y_i, \\ \dot{y}_i &= -a_i y_i - b_i \frac{\partial V}{\partial x_i} + \xi_i, \quad i = 1, 2, \ldots, P \end{aligned}$$

where ξ_i's are uncorrelated white noise processes, i.e., $D_{ij} = 0$ for $i \neq j$. We use the potential function $V(x_1, \ldots, x_P) = \frac{1}{2}\sum_{i=1}^{P} x_i^2$, along with $a_i = b_i = D_{ii} = 1$. Select $P = 7$, which corresponds to a fourteen dimensional state space. The initial uncertainty is a Gaussian zero mean pdf of covariance $0.5I$, where I is a 14×14 identity matrix. For this system, we compare the two different methods described in Section 5.4.2 for imposing vanishing boundary conditions. In both cases, approximations were obtained using 55 nodes per spatial dimension, 32 nodes along the temporal axis and an approximation rank of 30. This corresponds to a total DOF of 24,060, which is extremely small for a 14D time varying PDE. Table 1 shows the result of eight runs of both methods. Clearly, the new approach takes about the same time to run but performs better on all metrics. Additionally, it is frequently observed that the basis functions resulting from the deletion method have one oscillating component for every spatial dimension, as shown in Fig. 5(a) for a typical run. However, this phenomenon is never observed in the new method. The origin of these oscillating basis functions is not clear at this time and is under further investigation. Temporal basis functions of the deletion method are shown in Fig. 5(b). The corresponding basis functions and evolution of RMR for the new approach are shown in Fig. 6 and Fig. 7(a) respectively. Interestingly enough, it is observed that although in the new method of boundary condition enforcement we do not explicitly require the spatial basis



(a) Basis functions: x_1

(b) Basis functions: x_2

(c) Basis functions: x_3

(d) Basis functions: x_4

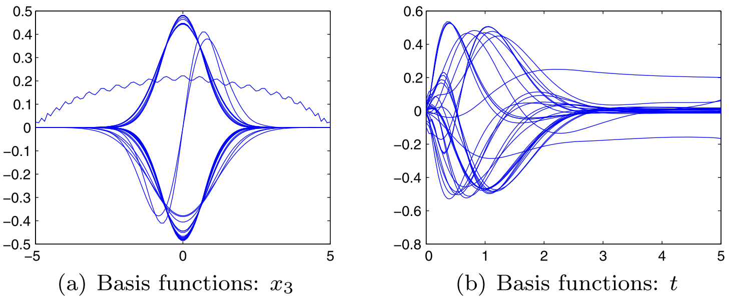
(e) Basis functions: t

(f) Evolution of RMR

(g) $x_1 - x_2$ marginal transient solution at final time ($t_f = 5\ s$)

(h) Error surface at final time

Fig. 4. Results for the 4-state system.



(a) Basis functions: x_3

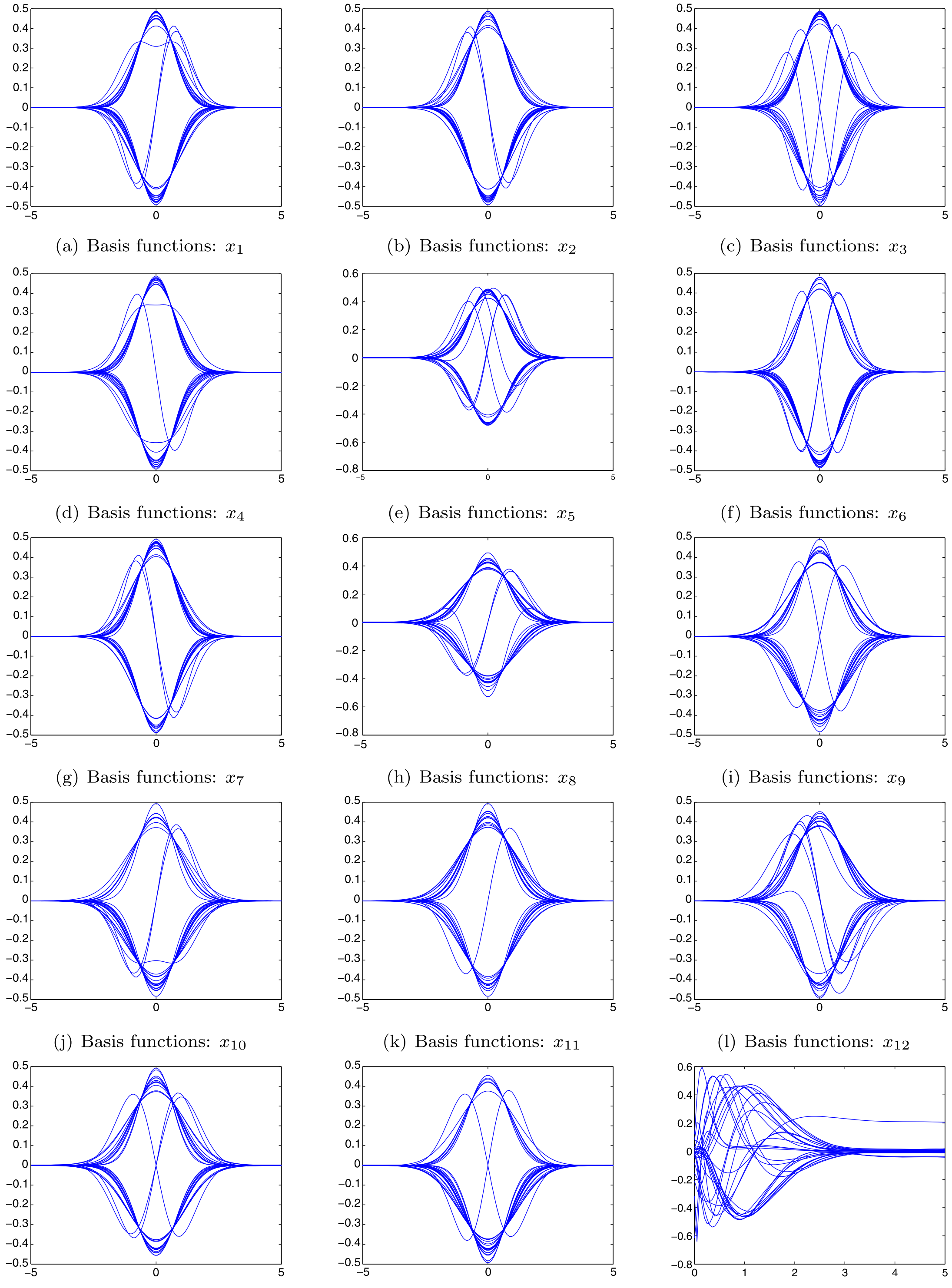
(b) Basis functions: t

Fig. 5. Typical basis functions of 14-state system with the "deletion method".

functions to vanish at both ends individually, it does in fact turn out to be the case in numerical simulations. Nevertheless, the advantages of this new method are evident in Table 1. Moreover, it provides a convenient way to impose other more general boundary conditions as discussed in Section 5.4.2.

Since the system in this example is linear, its true transient solution can be obtained using the Riccati equation (Eq. (14)). Fig. 7(b) illustrates the time history of MSE in the x_1–x_2 marginal pdf, obtained by comparing the obtained approximation against the solution to the Riccati equation.

It is interesting to note that in Ref. [13], the accuracy of the stationary solution showed significant variation across different runs of the algorithm as a result of which multiple runs were required to settle to an acceptable approximation. In addition, the rate of convergence was also slow. However when the proposed method is used for the transient problem,



(a) Basis functions: x_1

(b) Basis functions: x_2

(c) Basis functions: x_3

(d) Basis functions: x_4

(e) Basis functions: x_5

(f) Basis functions: x_6

(g) Basis functions: x_7

(h) Basis functions: x_8

(i) Basis functions: x_9

(j) Basis functions: x_{10}

(k) Basis functions: x_{11}

(l) Basis functions: x_{12}

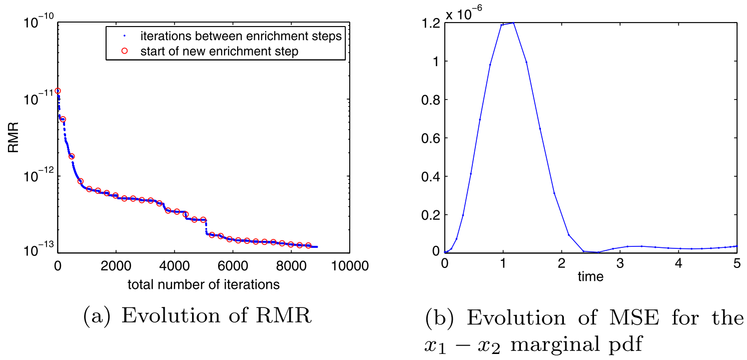
(m) Basis functions: x_{13}

(n) Basis functions: x_{14}

(o) Basis functions: t

Fig. 6. Basis functions of 14-state system with new method.

only insignificant variations are observed in the accuracy. Also, the RMR exhibits a favorable gradual decay (Fig. 7(a)), which has a much faster convergence rate. This is because the basis functions obtained here are not linearly dependent, unlike what was observed for the stationary solution in Ref. [13]. In this manner, the transient solution avoids the “bottleneck” or “swamp” behavior that traditionally causes slow convergence.



(a) Evolution of RMR

(b) Evolution of MSE for the $x_1 - x_2$ marginal pdf

Fig. 7. Error estimate of 14-state system with new method.

Table 1
Comparison of two methods for boundary condition enforcement: Example 3.

Comparison metric	New method	Deletion method
Average execution time (s)	3.01×10^5	2.80×10^5
Average RMR	1.26×10^{-13}	1.68×10^{-13}
Max MSE of marginal transient solution of best run	1.20×10^{-6}	6.90×10^{-6}
Average of max MSE of marginal transient solution	1.64×10^{-5}	7.42×10^{-5}

6.2. Application to modeling of polymeric fluids

6.2.1. Dumbbell example

Consider the following 3D dumbbell example from Ref. [9], where the dynamics is given by

$$\dot{\mathbf{q}} = K\mathbf{q} - \frac{1}{2}\mathrm{grad}(\phi) + \xi, \tag{37}$$

where K is the same as in Eq. (1) with $\beta = 1$, $\mathbf{q} \in [-8, 8]^3$, ξ is a 3 dimensional Brownian motion process with intensity given by a 3×3 identity matrix, and ϕ is the potential given by

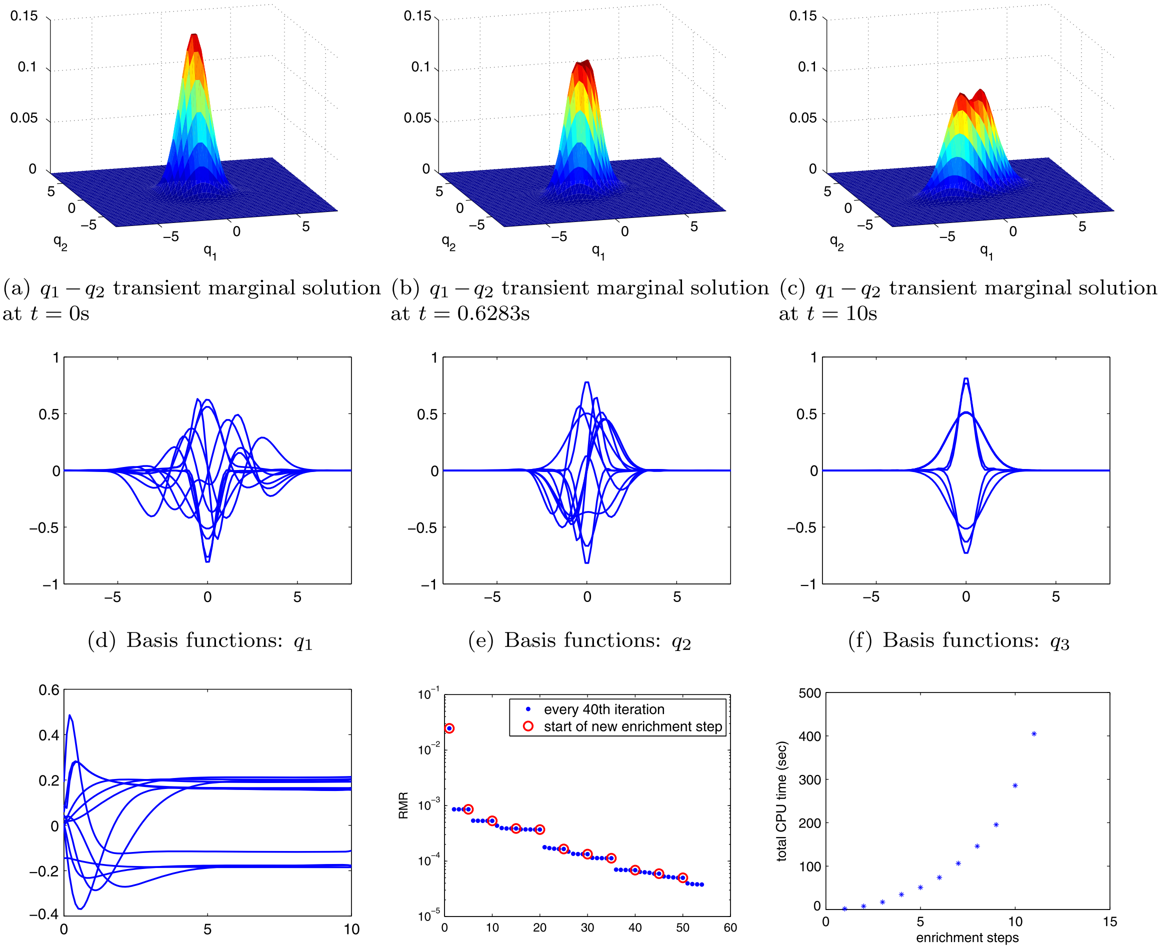
$$\phi(\mathbf{q}) = \frac{1}{2}\|\mathbf{q}\|_F^2 + \frac{\alpha}{p^3}\exp\left(\frac{-\|\mathbf{q}\|_F^2}{2p^2}\right). \tag{38}$$

The potential in Eq. (38) consists of a quadratic attractive part and a Gaussian repulsive part such that the beads of the chain neither scatter nor aggregate excessively [2]. The parameters α and p control the strength and extent of the repulsive potential, and are chosen as 0.1 and 0.5 respectively. The initial pdf is assumed to be standard 3 dimensional Gaussian and $t \in [0, 10]$.

Using 55 and 30 Chebyshev nodes for each spatial and the temporal dimension respectively and following 11 enrichment steps (consuming 405 seconds, corresponding DOF is $(55 \times 3 + 30) \times 11 = 2145$), the current method achieves an RMR of 3.774×10^{-5} as shown in Fig. 8(h). In this figure, the RMR value of every 40th iteration is shown as a data point. The q_1–q_2 marginal solution at 3 time instances and the basis functions are shown in Figs. 8(a) to 8(g). Since a closed form solution is not available for this example, Monte Carlo simulations (MCS) described in Section 4.3 were employed with 10^5 initial particles to estimate the true solution at the final time. Compared with the proposed tensor approach, an MSE of 8.69×10^{-7} was computed between the shown marginal in Figs. 8(a) to 8(c) and the MCS, which reflects a fairly accurate solution. Fig. 8(i) shows the total CPU time needed as the number of enrichment steps is increased. Each enrichment step introduces an additional $55 \times 3 + 30 = 195$ DOFs.

At this point, it is interesting to compare the above results with those obtained in Ref. [8] for the same 3D dumbbell example, wherein the quantized tensor train (QTT) format was used (which requires the grid size of each dimension to be a power of 2) along with a different transient solution scheme. A series of experiments were executed with 512 nodes for each spatial dimension and time steps ranging from 64 to 1024; resulting in solutions with error in the order of 10^{-4} to 10^{-5} consuming roughly 1200 to 5700 seconds of CPU time. However, their underlying approximation rank is unavailable due to which a precise DOF comparison is not possible.

The effect of the shear rate β is also of concern in that larger β indicates faster shear flow and thus in general the pdf of the states would have a larger support. It was reported in Ref. [8] that a larger approximation rank is required to maintain solution accuracy as β increases. To study the effect of β on the proposed tensor approach, we consider two values ($\beta = 1,\ 1.5$) while keeping the remaining configuration the same, and the results are given in Fig. 9. It is observed that now 14 instead of 11 enrichment steps are needed to obtain a similar RMR of 3.546×10^{-5} after 658 seconds of CPU time. The



(a) $q_1 - q_2$ transient marginal solution at $t = 0\text{s}$

(b) $q_1 - q_2$ transient marginal solution at $t = 0.6283\text{s}$

(c) $q_1 - q_2$ transient marginal solution at $t = 10\text{s}$

(d) Basis functions: q_1

(e) Basis functions: q_2

(f) Basis functions: q_3

(g) Basis functions: t

(h) Evolution of RMR

(i) Total CPU time versus enrichment steps

Fig. 8. Results for the 3D dumbbell example ($\beta = 1$).

corresponding MSE of the marginal pdf with respect to MCS is 7.48×10^{-7}. By comparing the marginal pdfs in Fig. 9(g) and Fig. 9(h), it is visible that the stationary pdf indeed “spreads out” as β increases from 1 to 1.5.

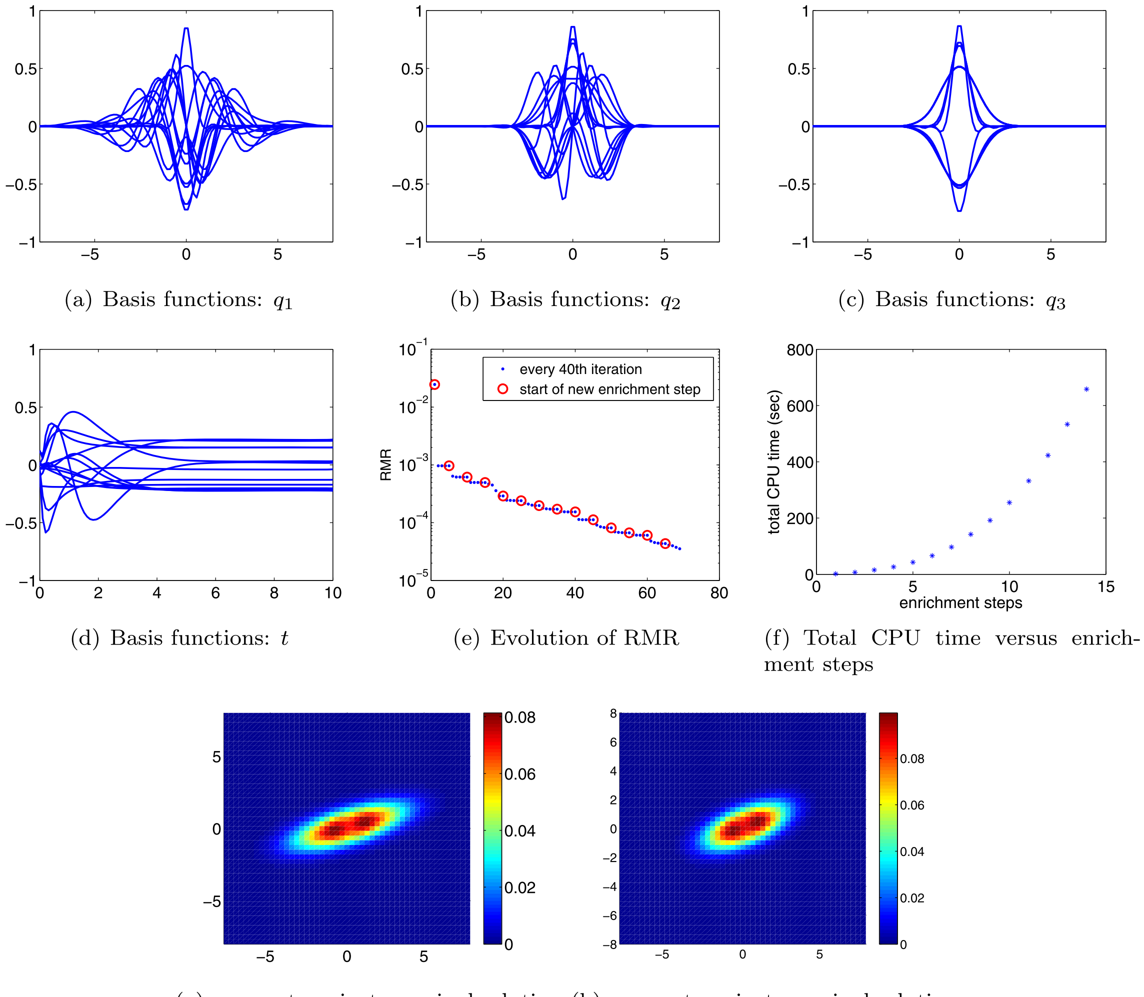
Note that all numerical examples considered so far assume Gaussian initial pdfs. This is motivated by the fact that initial uncertainty usually arises from a combination of many sources, which when combined in the spirit of the central limit theorem, results in an approximately Gaussian distribution. Additionally, in engineering applications one typically only knows the first and second order statistics of the pdf, whereby a Gaussian approximation is usually ideal [51]. Note however that the current approach is capable of handling arbitrary initial conditions provided they are sufficiently smooth. As an illustration, we consider the 3D dumbbell example (with $\beta = 1$) with the logistic distribution as its initial pdf given as follows:

$$\psi_0(q_1, q_2, q_3) = \prod_{i=1}^{3} \frac{1}{4s_i} \text{sech}^2\left(\frac{q_i - \mu_i}{2s_i}\right), \tag{39}$$

where μ_i, s_i are chosen to be 1 for $i = 1, 2, 3$. The suitable solution domain determined by MCS is $\mathbf{q} \in [-10, 10]^3, t \in [0, 15]$. Using 55 and 32 Chebyshev nodes for each spatial and the temporal dimension respectively and following 25 enrichment steps (consuming 51 minutes, corresponding DOF is $(55 \times 3 + 32) \times 25 = 4925$), it achieves an RMR of 2.286×10^{-5}, and the corresponding MSE of the marginal pdf with respect to MCS is 4.46×10^{-7} at the final time. More details of the test results are shown in Fig. 10.

6.2.2. *Multi-spring example*

Consider the MBS model with $N = 4$ beads and $\mathbf{q_i} \in [-5, 5]^3$ for $i = 1, \ldots, N - 1$ adapted from Ref. [9]. The FPE of this example can be written as [2]



(a) Basis functions: q_1

(b) Basis functions: q_2

(c) Basis functions: q_3

(d) Basis functions: t

(e) Evolution of RMR

(f) Total CPU time versus enrichment steps

(g) $q_1 - q_2$ transient marginal solution at $t = 10$s with $\beta = 1.5$

(h) $q_1 - q_2$ transient marginal solution at $t = 10$s with $\beta = 1$

Fig. 9. Results for the 3D dumbbell example ($\beta = 1.5$).

$$\frac{\partial\psi}{\partial t} = -\sum_{j=1}^{N-1}\frac{\partial}{\partial \mathbf{q}_j}\cdot\left(K\mathbf{q}_j - \frac{1}{4}\sum_{k=1}^{N-1}A_{jk}\frac{\partial\phi}{\partial\mathbf{q}_k}\right)\psi + \frac{1}{4}\sum_{j=1}^{N-1}\sum_{k=1}^{N-1}A_{jk}\frac{\partial}{\partial\mathbf{q}_j}\cdot\frac{\partial\psi}{\partial\mathbf{q}_k}, \tag{40}$$

where "$\cdot$" is the vector inner product, K is the same as in Eq. (1) with $\beta = 0.2$, $\phi = \frac{1}{2}\sum_{i=1}^{N-1}\|\mathbf{q}_i\|_F^2$ and

$$A_{ij} = \begin{cases} 2 & \text{if } |i-j| = 0, \\ -1 & \text{if } |i-j| = 1, \\ 0 & \text{otherwise.} \end{cases}$$

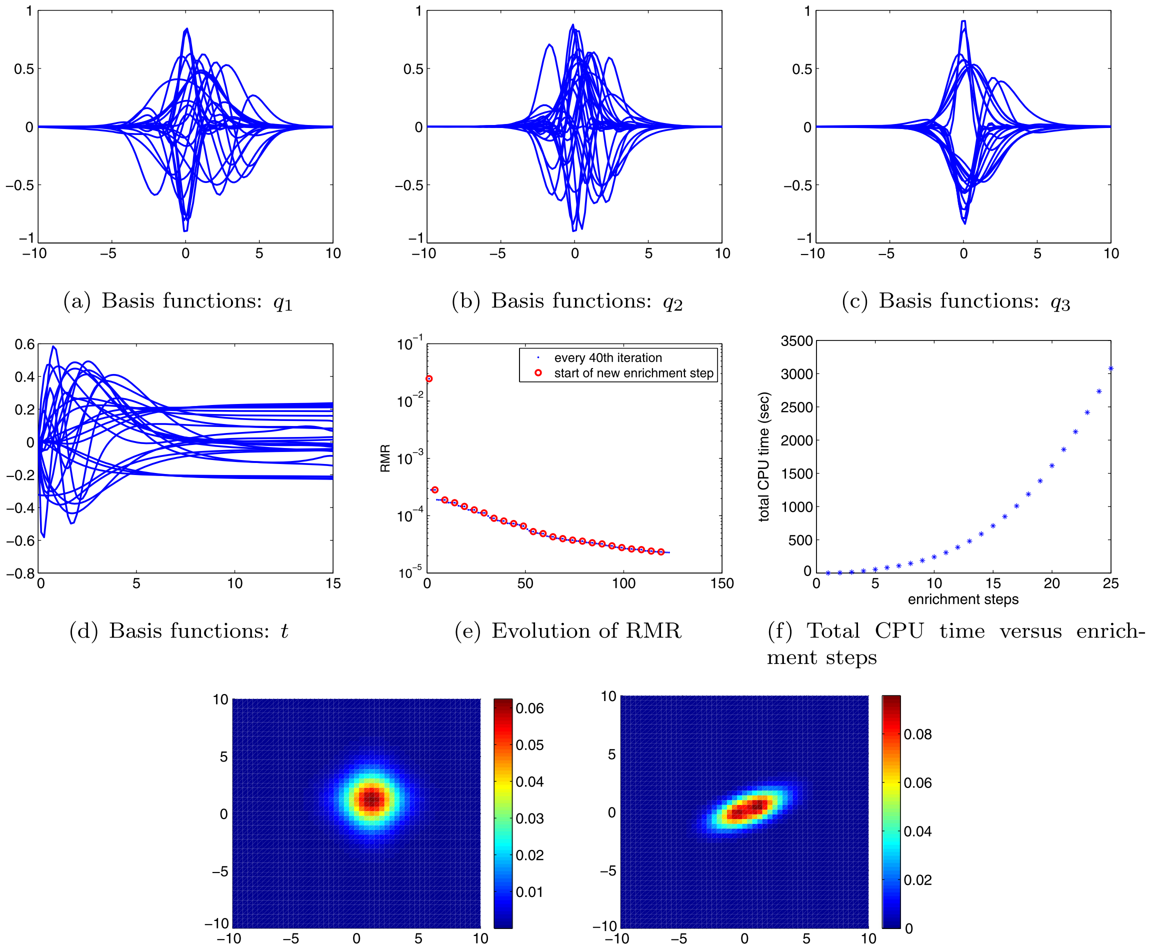
For the ϕ chosen above, the dynamics of the system is linear and can be written as

$$\begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \dot{\mathbf{q}}_3 \end{bmatrix} = \begin{pmatrix} K - \frac{A_{11}}{4}I & -\frac{A_{12}}{4}I & \mathbf{0} \\ -\frac{A_{21}}{4}I & K - \frac{A_{22}}{4}I & -\frac{A_{23}}{4}I \\ \mathbf{0} & -\frac{A_{32}}{4}I & K - \frac{A_{33}}{4}I \end{pmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix} + \xi, \tag{41}$$

where I is a 3×3 identity matrix and ξ is a 9 dimensional Brownian motion with intensity

$$Q = \frac{1}{2}\begin{pmatrix} A_{11}I & A_{12}I & \mathbf{0} \\ A_{21}I & A_{22}I & A_{23}I \\ \mathbf{0} & A_{32}I & A_{33}I \end{pmatrix}. \tag{42}$$

We assume that the initial pdf ψ_0 is a standard 9 dimensional Gaussian and $t \in [0, 20]$. Due to the linearity of the system, its true transient behavior can be obtained by the method described in Section 4.2. The current tensor approach was



(a) Basis functions: q_1

(b) Basis functions: q_2

(c) Basis functions: q_3

(d) Basis functions: t

(e) Evolution of RMR

(f) Total CPU time versus enrichment steps

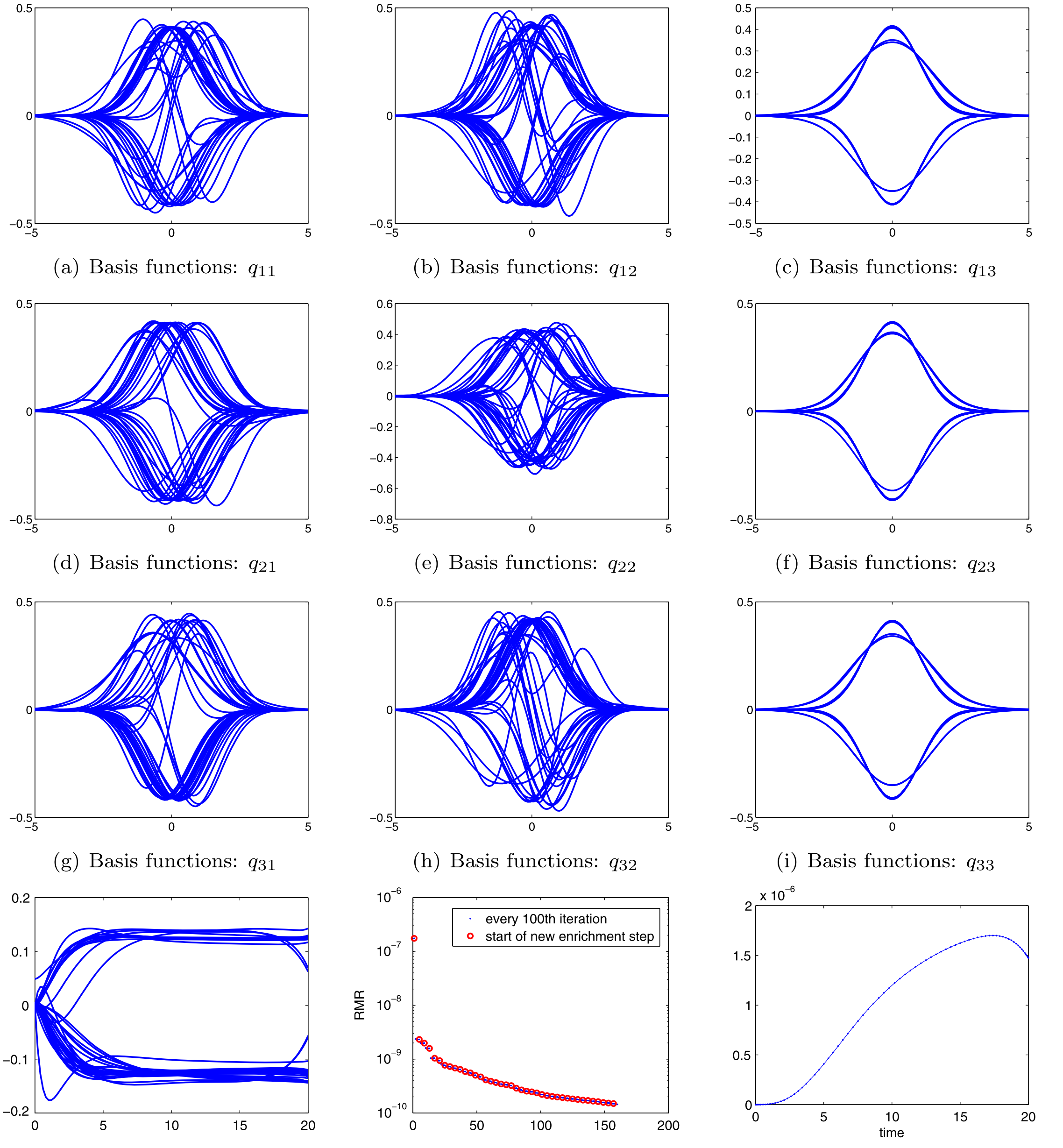
(g) q_1–q_2 transient marginal solution at initial time

(h) q_1 − q_2 transient marginal solution at final time

Fig. 10. Results for the 3D dumbbell example (non-Gaussian initial pdf).

employed using 53 spatial and 82 temporal Chebyshev nodes and after 40 enrichment steps (total DOF = 22360, consuming about 82 hours), the RMR was reduced to 1.450×10^{-10} as shown in Fig. 11(k), where RMR value of every 100th iteration is shown. The basis functions and the q_{11}–q_{12} marginal solution are shown in Fig. 11(a) to Fig. 11(j) and Fig. 12 respectively. The time history of MSE of Fig. 12 with respect to the marginal solution from the Riccati equation solver is given in Fig. 11(l). It is interesting to note from Fig. 11(j) that 4 out of the total 40 temporal basis functions do not yet achieve steady state near the final time, which is different from the other examples shown in this paper. This is likely because the number of enrichment steps used is not sufficient to completely resolve the solution features captured by these bases. In other words, while these features have been discovered, they haven't been discovered with enough resolution, leading to the anomalous behavior shown in Fig. 11(j). This was indirectly confirmed by simply dropping the poorly resolved time-bases from the approximation. The ensuing MSE profile was better than the trend shown in Fig. 11(l), underlining the fact that the approximation accuracy improved upon removal of the unresolved features. It is expected that if additional enrichment steps are executed (consuming more time, thus not actually performed here), these bases will get fully resolved and their behavior will conform to the result stated in Proposition 1. In addition, this anomalous behavior is also observed in other examples admitting a stationary solution when the approximation rank is not large enough. Note that due to the structure of the ALS scheme, all R rank-one tensors obtained thus far will change after each iteration, allowing the unresolved bases to improve as the algorithm progresses.

To study how the number of beads affects the CPU time for the proposed method, the 3 beads (6D) version of Eq. (40) is solved using the same parameters and discretization grids. After 40 enrichment steps (total DOF = 16000, consuming about 23 hours), the RMR decreases to 5.726×10^{-9} as shown in Fig. 13(b), where RMR value of every 105th iteration is shown. The corresponding temporal basis functions and the time history of MSE of q_{11}–q_{12} transient marginal solution with respect to Riccati equation solver is given in Fig. 13(a) and Fig. 13(c) respectively. As can be seen, 40 enrichment steps are sufficient for the 6D case to yield an extremely accurate solution (MSE $< 10^{-9}$) and the temporal bases approach constant values as



(a) Basis functions: q_{11}

(b) Basis functions: q_{12}

(c) Basis functions: q_{13}

(d) Basis functions: q_{21}

(e) Basis functions: q_{22}

(f) Basis functions: q_{23}

(g) Basis functions: q_{31}

(h) Basis functions: q_{32}

(i) Basis functions: q_{33}

(j) Basis functions: t

(k) Evolution of RMR

(l) Evolution of MSE for $q_{11} - q_{12}$ marginal pdf

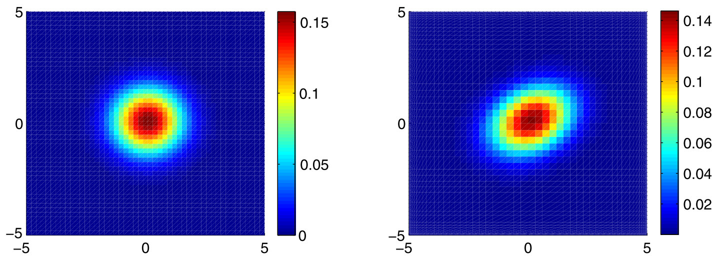
Fig. 11. Basis functions for the 9D MBS example.

indicated by Proposition 1. Fig. 14 shows how the CPU time scales with the number of beads for the 3D dumbbell example (2 beads with $\beta = 1.5$) considered in Section 6.2.1, 6D MBS (3 beads) and 9D MBS (4 beads) cases.

In Ref. [9], the 5 beads (12D) version of Eq. (40) was solved by a different tensor method using the QTT format, where an average QTT-rank of 38 (maximal QTT-rank of 84) was reported for the $\beta = 0.2$ case with “relative tensor rounding and residual accuracy” $\epsilon = 10^{-4}$. However, the number of nodes used for spatial dimensions, the time steps as well as the total computational time are not provided for that case, due to which a detailed comparison is not possible.

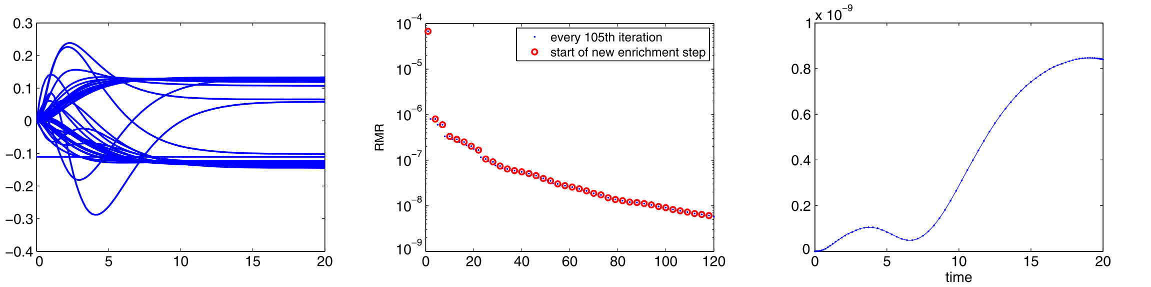
7. Conclusions

In this paper an efficient numerical solver for transient FPE was developed and applied to the simulation of polymeric fluids, which extends the authors' former work [13] on the stationary FPE. The strategy of separating the spatial and temporal domains, previously used in the proper generalized decomposition method [7] is introduced in the ALS framework



(a) $q_{11} - q_{12}$ transient marginal solution at initial time (b) $q_{11} - q_{12}$ transient marginal solution at final time

Fig. 12. Transient marginal solution for the 9D MBS example.



(a) Temporal basis functions (b) Evolution of RMR (c) Evolution of MSE for $q_{11} - q_{12}$ marginal pdf

Fig. 13. Results for the 6D MBS example.

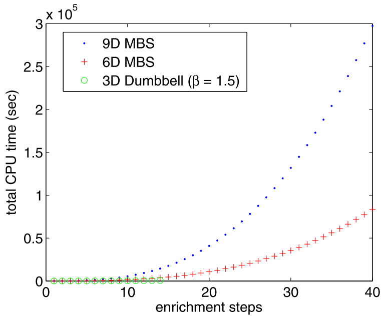


Fig. 14. CPU time versus number of beads in the MBS model.

for solving the transient FPE in a CPD format. A new method for enforcing the boundary conditions is proposed in which they are expressed as a linear constraint and imposed via a penalty parameter. The asymptotic behavior of the temporal basis functions is studied and shown to attain steady state for systems admitting a stationary distribution. The efficiency of the proposed method is captured by the favorable growth of DOF with system dimensionality, along with the fact that it retains the transient approximation in a single CPD format such that no time stepping is required. The latter property is highly desirable and will allow the method to be used successfully in nonlinear filtering problems. The overall accuracy of approximations as well as convergence characteristics are shown to be better than its precursor [13] with potential for use in a wide range of applications.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. ECCS-1254244 and the NASA Marshall Space Flight Center Grant No. NNM13AA07G.

References

[1] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids, J. Non-Newton. Fluid Mech. 139 (2006) 153–176.

[2] G. Venkiteswaran, M. Junk, A qmc approach for high dimensional Fokker–Planck equations modelling polymeric liquids, Math. Comput. Simul. 68 (2005) 43–56.

[3] A. Fuller, Analysis of nonlinear stochastic systems by means of the Fokker–Planck equation, Int. J. Control 9 (1969) 603–655.

[4] R. Langley, A finite element method for the statistics of non-linear random vibration, J. Sound Vib. 101 (1985) 41–54.

[5] R.E. Bellman, Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1957.

[6] M. Kumar, S. Chakravorty, J.L. Junkins, A semianalytic meshless approach to the transient Fokker–Planck equation, Probab. Eng. Mech. 25 (2010) 323–331.

[7] A. Ammar, B. Mokdad, F. Chinesta, R. Keunings, A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modelling of complex fluids: Part II: transient simulation using space–time separated representations, J. Non-Newton. Fluid Mech. 144 (2007) 98–121.

[8] S. Dolgov, B.N. Khoromskij, 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 J. Sci. Comput. 34 (2012) A3016–A3038.

[9] S. Dolgov, B. Khoromskij, Simultaneous state-time approximation of the chemical master equation using tensor product formats, preprint, arXiv:1311.3143, 2013 (submitted).

[10] M. Kumar, S. Chakravorty, P. Singla, J.L. Junkins, The partition of unity finite element approach with hp-refinement for the stationary Fokker–Planck equation, J. Sound Vib. 327 (2009) 144–162.

[11] M. Kumar, S. Chakravorty, J.L. Junkins, On the curse of dimensionality in the Fokker–Planck equation, in: AAS Astrodynamics Specialist Conference, Pittsburgh, PA, USA, Aug 9–13, 2009.

[12] Y. Sun, M. Kumar, A meshless p-pufem Fokker–Planck equation solver with automatic boundary condition enforcement, in: American Control Conference (ACC), 2012, IEEE, 2012, pp. 74–79.

[13] Y. Sun, M. Kumar, Numerical solution of high dimensional stationary Fokker–Planck equations via tensor decomposition and Chebyshev spectral differentiation, Comput. Math. Appl. 67 (2014) 1960–1977.

[14] Y. Sun, M. Kumar, A tensor decomposition approach to high dimensional stationary Fokker–Planck equations, in: American Control Conference (ACC), Portland, OR, Jun. 4–6, 2014, IEEE, 2014.

[15] B.N. Khoromskij, Tensors-structured numerical methods in scientific computing: survey on recent advances, Chemom. Intell. Lab. Syst. 110 (2012) 1–19.

[16] G. Beylkin, M. Mohlenkamp, Algorithms for numerical analysis in high dimensions, SIAM J. Sci. Comput. 26 (2005) 2133–2159.

[17] J. Ballani, L. Grasedyck, A projection method to solve linear systems in tensor format, Numer. Linear Algebra Appl. 20 (2013) 27–43.

[18] D. Kressner, C. Tobler, Krylov subspace methods for linear systems with tensor product structure, SIAM J. Matrix Anal. Appl. 31 (2010) 1688–1714.

[19] J.D. Carroll, J.-J. Chang, Analysis of individual differences in multidimensional scaling via an n-way generalization of “Eckart–Young” decomposition, Psychometrika 35 (1970) 283–319.

[20] R.A. Harshman, Foundations of the parafac procedure: models and conditions for an “explanatory” multimodal factor analysis, UCLA Working Papers in Phonetics, 1970.

[21] O. Koch, C. Lubich, Dynamical low-rank approximation, SIAM J. Matrix Anal. Appl. 29 (2007) 434–454.

[22] J. Haegeman, J.I. Cirac, T.J. Osborne, I. Pižorn, H. Verschelde, F. Verstraete, Time-dependent variational principle for quantum lattices, Phys. Rev. Lett. 107 (2011) 070601.

[23] I.V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput. 33 (2011) 2295–2317.

[24] I.V. Oseledets, S. Dolgov, Solution of linear systems and matrix inversion in the tt-format, SIAM J. Sci. Comput. 34 (2012) A2718–A2739.

[25] A. Ammar, F. Chinesta, E. Cueto, M. Doblaré, Proper generalized decomposition of time-multiscale models, Int. J. Numer. Methods Eng. 90 (2012) 569–596.

[26] F. Chinesta, A. Ammar, E. Cueto, Recent advances and new challenges in the use of the proper generalized decomposition for solving multidimensional models, Arch. Comput. Methods Eng. 17 (2010) 327–350.

[27] T.G. Kolda, Orthogonal tensor decompositions, SIAM J. Matrix Anal. Appl. 23 (2001) 243–255.

[28] T. Kolda, B. Bader, Tensor decompositions and applications, SIAM Rev. 51 (2009) 455–500.

[29] A. Smilde, R. Bro, P. Geladi, Multi-way Analysis: Applications in the Chemical Sciences, Wiley, 2005.

[30] R. Bro, Parafac. tutorial and applications, Chemom. Intell. Lab. Syst. 38 (1997) 149–171.

[31] L. Trefethen, Spectral methods in MATLAB, EngineeringPro collection, SIAM, 2000.

[32] J.P. Boyd, Chebyshev and Fourier Spectral Methods, Courier Dover Publications, 2001.

[33] Y. Sun, M. Kumar, Solution of high dimensional transient Fokker–Planck equations by tensor decomposition, in: American Control Conference (ACC), July 1–3, 2015, IEEE, Chicago, IL, 2015.

[34] Y. Sun, M. Kumar, A tensor decomposition method for high dimensional Fokker–Planck equation modeling polymeric liquids, in: Science and Technology Forum and Exposition, AIAA, Kissimmee, FL, Jan. 5–9, 2015.

[35] Y. Sun, M. Kumar, Uncertainty propagation via tensor decomposition in nearly circular orbital mechanics, in: International Conference on Information Fusion, IEEE, Washington, DC, 2015, submitted for publication.

[36] Y. Sun, M. Kumar, Nonlinear Bayesian filtering based on Fokker Planck equation and tensor decomposition, in: International Conference on Information Fusion, IEEE, Washington, DC, 2015, submitted for publication.

[37] P. Comon, X. Luciani, A.L.F. de Almeida, Tensor decompositions, alternating least squares and other tales, J. Chemom. 23 (2009) 393–405.

[38] V. De Silva, L.-H. Lim, Tensor rank and the ill-posedness of the best low-rank approximation problem, SIAM J. Matrix Anal. Appl. 30 (2008) 1084–1127.

[39] J. Håstad, Tensor rank is np-complete, J. Algorithms 11 (1990) 644–654.

[40] L. Grasedyck, D. Kressner, C. Tobler, A literature survey of low-rank tensor approximation techniques, preprint, arXiv:1302.7121, 2013.

[41] R. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems, SIAM, 2007.

[42] K.E. Atkinson, An Introduction to Numerical Analysis, John Wiley & Sons, 2008.

[43] M. Kumar, P. Singla, S. Chakravorty, J.L. Junkins, The partition of unity finite element approach to the stationary Fokker–Planck equation, in: AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Keystone, CO, 2006, 2006, pp. 21–24.

[44] G.N. Milstein, Numerical Integration of Stochastic Differential Equations, Springer, 1995.

[45] E. Allen, Modeling with Itô Stochastic Differential Equations, vol. 22, Springer, 2007.

[46] G.N. Milstein, M.V. Tretyakov, Stochastic Numerics for Mathematical Physics, Springer, 2010.

[47] D.J. Higham, An algorithmic introduction to numerical simulation of stochastic differential equations, SIAM Rev. 43 (2001) 525–546.

[48] S. Cyganowski, A maple package for stochastic differential equations, in: Computational Techniques and Applications: CTAC95, Melbourne, 1995, 1996, pp. 223–230.

[49] G.R.G. Muscolino, M. Vasta, Stationary and non-stationary probability density function for nonlinear oscillators, Int. J. Non-Linear Mech. 32 (1997) 1051–1064.

[50] S.T. Ariaratnam, Random vibrations of non-linear suspensions, J. Mech. Eng. Sci. 2 (1960) 195–201.

[51] P.S. Maybeck, Stochastic Models, Estimation, and Control, vol. 3, Academic Press, 1982.