Journal of Scientific Computing (2021) 88:36
https://doi.org/10.1007/s10915-021-01539-3

Rank-Adaptive Tensor Methods for High-Dimensional Nonlinear PDEs

Alec Dektor[1] · Abram Rodgers[1] · Daniele Venturi[1]

Received: 11 December 2020 / Revised: 23 April 2021 / Accepted: 26 May 2021 / Published online: 24 June 2021

Abstract
We present a new rank-adaptive tensor method to compute the numerical solution of high-dimensional nonlinear PDEs. The method combines functional tensor train (FTT) series expansions, operator splitting time integration, and a new rank-adaptive algorithm based on a thresholding criterion that limits the component of the PDE velocity vector normal to the FTT tensor manifold. This yields a scheme that can add or remove tensor modes adaptively from the PDE solution as time integration proceeds. The new method is designed to improve computational efficiency, accuracy and robustness in numerical integration of high-dimensional problems. In particular, it overcomes well-known computational challenges associated with dynamic tensor integration, including low-rank modeling errors and the need to invert covariance matrices of tensor cores at each time step. Numerical applications are presented and discussed for linear and nonlinear advection problems in two dimensions, and for a four-dimensional Fokker–Planck equation.

Keywords Tensor train decomposition · Dynamical low-rank approximation · Tensor manifolds

1 Introduction

High-dimensional partial differential equations (PDEs) arise in many areas of engineering, physical sciences and mathematics. Classical examples are equations involving probability density functions (PDFs) such as the Fokker–Planck equation [49], the Liouville equation [13,14,58], and the Boltzmann equation [8,11,18]. More recently, high-dimensional PDEs have also become central to many new areas of application such as optimal mass transport [27,59], random dynamical systems [57,58], mean field games [19,52], and functional-differential equations [55,56]. Computing the numerical solution to high-dimensional PDEs is an extremely challenging problem which has attracted substantial research efforts in recent years. Techniques such as sparse collocation methods [4,9,12,25,41], high-dimensional model representations [3,10,37], deep neural networks [46,47,60], and numerical tensor

✉ Daniele Venturi
venturi@ucsc.edu

[1] Department of Applied Mathematics, University of California Santa Cruz, Santa Cruz, CA 95064, USA

methods [2,7,28,31,33,51] were proposed to mitigate the exponential growth of the degrees of freedom, the computational cost and the memory requirements.

In this paper, we build upon our recent work on dynamical tensor approximation [16,17], and develop new rank-adaptive temporal integrators to compute the numerical solution of high-dimensional initial/boundary value problems of the form

$$\begin{cases} \dfrac{\partial u(\boldsymbol{x},t)}{\partial t} = G(u(\boldsymbol{x},t)), \\ u(\boldsymbol{x},0) = u_0(\boldsymbol{x}), \end{cases} \tag{1}$$

where $\boldsymbol{x} \in \Omega \subseteq \mathbb{R}^d$ (Ω compact, $d \geq 1$), and G is a nonlinear operator which may take into account boundary conditions. A well-known challenge of dynamic tensor approximations to (1) is that the curvature of the tensor manifold in which we compute the PDE solution is inversely proportional to the energy of the tensor modes. This means that the smaller the energy of the tensor modes the higher the curvature. Hence, to integrate a solution characterized by tensor modes with a wide range of energies one has to consider time stepping schemes that can effectively handle geometric features associated with the curvature of the manifold. In projection-based approaches [16,17,34,43] the computational challenge posed by the curvature of the tensor manifold translates into the need to invert the positive semi-definite covariance matrices of the tensor cores at each time step. A time-integration scheme constructed in this way may become numerically unstable in the presence of tensor modes with small energy, or even singular when modes with zero energy are present (e.g., at a time instant in which we increase the tensor rank by adding a mode with zero energy). To mitigate this problem, Babaee et al. [1] introduced a matrix pseudo-inverse approximation method that can handle potential singularities in the covariance matrices of the tensor cores, in particular when adding modes with zero energy to the tensor series expansion of the PDE solution.

A mathematically rigorous framework to integrate dynamical tensors over manifolds with arbitrary curvature was developed by Lubich et al. in [32,38,39]. The key idea is to integrate the evolution equation generating the tensor dynamics using operator splitting schemes, e.g., the Lie–Trotter or the Strang time integrators (see [32,39] for details). This results in a scheme that does not suffer from the curvature of the tensor manifold, and even provides an exact representation in the presence of tensor modes with zero energy. The numerical method presented in this work combines all these features, i.e., functional tensor train (FTT) series expansions, operator splitting time integration, and a new rank-adaptive algorithm to add and remove tensor modes from the PDE solution based on a thresholding criterion that limits the component of the velocity vector normal to the FTT tensor manifold.

This paper is organized as follows. In Sect. 2 we briefly review finite-rank functional tensor train (FTT) expansions of high-dimensional functions. In Sect. 3 we discuss dynamic tensor approximation of nonlinear PDEs of the form (1) and develop robust temporal integration schemes based on operator splitting methods. We also discuss step-truncation algorithms [50,51] and prove that dynamic tensor approximation and step-truncation are at least order one consistent to one another. In Sect. 4 we develop new rank-adaptive time integrators on rank-structured FTT tensor manifolds and prove that the resulting scheme is consistent. In Sect. 5 we present and discuss various numerical applications of the proposed rank-adaptive tensor method, and demonstrate its accuracy and computational efficiency. The main findings are summarized in Sect. 6.

2 The Manifold of Fixed-Rank FTT Tensors

Let us consider the weighted Hilbert space[1]

$$H = L^2_\mu(\Omega), \tag{2}$$

where $\Omega \subseteq \mathbb{R}^d$ is a separable domain such as a d-dimensional flat torus $\mathbb{T}^d$ or a Cartesian product of d real intervals $\Omega_i = [a_i, b_i]$

$$\Omega = \bigtimes_{i=1}^{d} \Omega_i, \tag{3}$$

and μ is a finite product measure on Ω

$$\mu(\boldsymbol{x}) = \prod_{i=1}^{d} \mu_i(x_i). \tag{4}$$

Let τ be the counting measure on $\mathbb{N}$. Each element $u \in L^2_\mu(\Omega)$ admits a functional tensor train (FTT) expansion of the form

$$u(\boldsymbol{x}) = \sum_{\alpha_0=1}^{\infty} \sum_{\alpha_1=1}^{\infty} \cdots \sum_{\alpha_d=1}^{\infty} \sqrt{\lambda(\alpha_{d-1})}\psi_1(1; x_1; \alpha_1)\psi_2(\alpha_1; x_2; \alpha_2) \cdots \psi_d(\alpha_{d-1}; x_d; 1), \tag{5}$$

where $\{\psi_i(\alpha_{i-1}; x_i; \alpha_i)\}_{\alpha_i}$ is an orthonormal basis for the space $L^2_{\tau \times \mu_i}(\mathbb{N} \times \Omega_i)$. It can be shown that $\psi_i(\alpha_{i-1}; x_i; \alpha_i)$ are eigenfunctions for a self-adjoint compact operator and $\lambda(1) \geq \lambda(2) \geq \cdots \geq 0$ is a sequence of real numbers converging to zero (see [6,17] for more details). By truncating (5) so that only the largest singular values are retained, we obtain the approximation of $u(\boldsymbol{x})$

$$u_{\boldsymbol{r}}(\boldsymbol{x}) = \sum_{\alpha_0=1}^{r_0} \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_d=1}^{r_d} \sqrt{\lambda(\alpha_{d-1})}\psi_1(\alpha_0; x_1; \alpha_1)\psi_2(\alpha_1; x_2; \alpha_2) \cdots \psi_d(\alpha_{d-1}; x_d; \alpha_d), \tag{6}$$

where $\boldsymbol{r} = (r_0, r_1, \ldots, r_{d-1}, r_d)$ is the FTT rank. It is convenient to write (6) in a more compact form as

$$u_{\boldsymbol{r}}(\boldsymbol{x}) = \boldsymbol{\Psi}_1(x_1)\boldsymbol{\Psi}_2(x_2) \cdots \sqrt{\boldsymbol{\Lambda}}\boldsymbol{\Psi}_d(x_d), \tag{7}$$

where $\boldsymbol{\Psi}_i(x_i)$ is a $r_{i-1} \times r_i$ matrix with entries $[\boldsymbol{\Psi}_i(x_i)]_{jk} = \psi_i(j; x_i; k)$ and $\boldsymbol{\Lambda}$ is a diagonal matrix with entries $\lambda(\alpha_{d-1})$ $(\alpha_{d-1} = 1, \ldots, r_{d-1})$. The matrix-valued functions $\boldsymbol{\Psi}_i(x_i)$ will be referred to as FTT *cores*, and we denote by $M_{r_{i-1} \times r_i}(L^2_{\mu_i}(\Omega_i))$ the set of all $r_{i-1} \times r_i$ matrices with entries in $L^2_{\mu_{i-1}}(\Omega_i)$. To simplify notation even more, we will often suppress explicit tensor core dependence on the spatial variable x_i, allowing us to simply write $\boldsymbol{\Psi}_i = \boldsymbol{\Psi}_i(x_i)$ and $\psi_i(\alpha_{i-1}, \alpha_i) = \psi_i(\alpha_{i-1}; x_i; \alpha_i)$ as the spatial dependence is indicated by the tensor core subscript.

[1] The approximation theory presented in Sect. 2 may be applied to more general Hilbert spaces, e.g., to the Sobolev space $W^{2,p}$ [16] and to tree-based tensor manifolds in Banach spaces [20].

2.1 Orthogonalization and Truncation of FTT Tensors

For any tensor core $\boldsymbol{\Psi}_i \in M_{r_{i-1}\times r_i}(L^2_{\mu_i}(\Omega_i))$ we define the matrix

$$\langle \boldsymbol{\Psi}_i^{\mathrm{T}} \boldsymbol{\Psi}_i \rangle_i \in M_{r_i \times r_i}(\mathbb{R}) \tag{8}$$

with entries[2]

$$\left\langle \boldsymbol{\Psi}_i^{\mathrm{T}} \boldsymbol{\Psi}_i \right\rangle_i (j,k) = \sum_{p=1}^{r_{i-1}} \int_{\Omega_i} \psi_i(p;x_i;j)\psi_i(p;x_i;k)d\mu_i(x_i). \tag{9}$$

The FTT representation (7) is given in terms of FTT cores $\boldsymbol{\Psi}_i$ satisfying[3]

$$\begin{aligned} \left\langle \boldsymbol{\Psi}_i^{\mathrm{T}} \boldsymbol{\Psi}_i \right\rangle_i &= \boldsymbol{I}_{r_i\times r_i}, \qquad i=1,\ldots,d-1, \\ \left\langle \boldsymbol{\Psi}_d \boldsymbol{\Psi}_d^{\mathrm{T}} \right\rangle_d &= \boldsymbol{I}_{r_{d-1}\times r_{d-1}}. \end{aligned} \tag{10}$$

Other orthogonal representations can be computed, e.g., based on recursive QR decompositions. To describe different orthogonalizations of FTT tensors, let $\boldsymbol{\Psi}_i \in M_{r_{i-1}\times r_i}(L^2_{\mu_i}(\Omega_i))$ and consider each column of $\boldsymbol{\Psi}_i$ as a vector in $\bigtimes_{n=1}^{r_{i-1}} L^2_{\mu_i}(\Omega_i)$. Performing an orthogonalization process (e.g. Gram–Schmidt) on the columns of the FTT core $\boldsymbol{\Psi}_i$ relative to the inner product (8) yields a QR-type decomposition of the form

$$\boldsymbol{\Psi}_i = \boldsymbol{Q}_i \boldsymbol{R}_i, \tag{11}$$

Where $\boldsymbol{Q}_i$ is an $r_{i-1} \times r_i$ matrix with elements in $L^2_{\mu_i}(\Omega_i)$ satisfying $\left\langle \boldsymbol{Q}_i^{\mathrm{T}} \boldsymbol{Q}_i \right\rangle_i = \boldsymbol{I}_{r_i\times r_i}$, and $\boldsymbol{R}_i$ is an upper triangular $r_i \times r_i$ matrix with real entries. Next consider an arbitrary FTT tensor $u_{\boldsymbol{r}} = \boldsymbol{\Psi}_1\boldsymbol{\Psi}_2\cdots\boldsymbol{\Psi}_d$, where the matrix $\langle \boldsymbol{\Psi}_i^{\mathrm{T}} \boldsymbol{\Psi}_i \rangle_i$ may be singular. For notational convenience, we define the partial products

$$\boldsymbol{\Psi}_{\leq i} = \boldsymbol{\Psi}_1 \cdots \boldsymbol{\Psi}_i \qquad \text{and} \qquad \boldsymbol{\Psi}_{>i} = \boldsymbol{\Psi}_{i+1}\cdots\boldsymbol{\Psi}_d. \tag{12}$$

One way to orthogonalize $u_{\boldsymbol{r}}$ is by performing QR decompositions recursively from left to right as we will now describe. Begin by decomposing $\boldsymbol{\Psi}_1$ as

$$\begin{aligned} \boldsymbol{\Psi}_1 = \boldsymbol{Q}_1\boldsymbol{R}_1, \qquad &\boldsymbol{Q}_1 \in M_{r_0\times r_1}(L^2_{\mu_1}(\Omega_1)), \quad \left\langle \boldsymbol{Q}_1^{\mathrm{T}} \boldsymbol{Q}_1 \right\rangle_1 = \boldsymbol{I}_{r_1\times r_1}, \\ &\boldsymbol{R}_1 \in M_{r_1\times r_1}(\mathbb{R}) \text{ is upper triangular.} \end{aligned} \tag{13}$$

Now we may write $u_{\boldsymbol{r}} = \boldsymbol{Q}_1\boldsymbol{R}_1\boldsymbol{\Psi}_2\cdots\boldsymbol{\Psi}_d$. Next, perform another QR decomposition

$$\begin{aligned} \boldsymbol{R}_1\boldsymbol{\Psi}_2 = \boldsymbol{Q}_2\boldsymbol{R}_2, \qquad &\boldsymbol{Q}_2 \in M_{r_1\times r_2}(L^2_{\mu_2}(\Omega_2)), \quad \left\langle \boldsymbol{Q}_2^{\mathrm{T}} \boldsymbol{Q}_2 \right\rangle_2 = \boldsymbol{I}_{r_2\times r_2}, \\ &\boldsymbol{R}_2 \in M_{r_2\times r_2}(\mathbb{R}) \text{ is upper triangular.} \end{aligned} \tag{14}$$

Proceeding recursively in this way we obtain a representation for $u_{\boldsymbol{r}}$ of the form

$$u_{\boldsymbol{r}} = \boldsymbol{Q}_1\cdots\boldsymbol{Q}_{d-1}\boldsymbol{Q}_d\boldsymbol{R}_d, \tag{15}$$

[2] The averaging operation in (9) can be viewed as a an inner product on the space $\bigtimes_{n=1}^{r_{i-1}} L^2_{\mu_i}(\Omega_i)$.

[3] Equation (10) follows immediately from the orthonormality of $\{\psi_i(\alpha_{i-1};x_i;\alpha_i)\}_{\alpha_i}$ relative to the inner product in $L^2_{\tau\times\mu_i}(\mathbb{N}\times\Omega_i)$.

where each $\boldsymbol{Q}_i \in M_{r_{i-1}\times r_i}(L^2_{\mu_i}(\Omega_i))$ satisfies $\left\langle \boldsymbol{Q}_i^{\mathrm{T}} \boldsymbol{Q}_i \right\rangle_i = \boldsymbol{I}_{r_i \times r_i}$. We refer to such a representation as a left orthogonalization of $u_{\boldsymbol{r}}$. We may stop orthogonolizing at any step in the recursive process to obtain the partial left orthogonalization

$$u_{\boldsymbol{r}} = \boldsymbol{Q}_{\leq i} \boldsymbol{R}_i \boldsymbol{\Psi}_{>i}. \tag{16}$$

Similar to orthogonalizing from the left, we may also orthogonalize $u_{\boldsymbol{r}}$ from the right. To do so, begin by performing a QR decomposition

$$\begin{aligned} \boldsymbol{\Psi}_d^{\mathrm{T}} = \boldsymbol{K}_d \boldsymbol{W}_d, \qquad &\boldsymbol{K}_d \in M_{r_d \times r_{d-1}}(L^2_{\mu_d}(\Omega_d)), \quad \left\langle \boldsymbol{K}_d^{\mathrm{T}} \boldsymbol{K}_d \right\rangle_d = \boldsymbol{I}_{r_{d-1}\times r_{d-1}}, \\ &\boldsymbol{W}_d \in M_{r_{d-1}\times r_{d-1}}(\mathbb{R}) \text{ is upper triangular.} \end{aligned} \tag{17}$$

A substitution of (17) into (7) yields the expansion $u_{\boldsymbol{r}} = \boldsymbol{\Psi}_1 \cdots \boldsymbol{\Psi}_{d-1} \boldsymbol{W}_d^{\mathrm{T}} \boldsymbol{K}_d^{\mathrm{T}}$. Next perform a QR decomposition

$$\begin{aligned} \boldsymbol{W}_d \boldsymbol{\Psi}_{d-1}^{\mathrm{T}} =& \boldsymbol{K}_{d-1} \boldsymbol{W}_{d-1}, \qquad \boldsymbol{K}_{d-1} \in M_{r_{d-1}\times r_{d-2}}(L^2_{\mu_{d-1}}(\Omega_{d-1})), \\ &\left\langle \boldsymbol{K}_{d-1}^{\mathrm{T}} \boldsymbol{K}_{d-1} \right\rangle_{d-1} = \boldsymbol{I}_{r_{d-2}\times r_{d-2}}, \\ &\boldsymbol{W}_{d-1} \in M_{r_{d-2}\times r_{d-2}}(\mathbb{R}) \text{ is upper triangular.} \end{aligned} \tag{18}$$

Proceeding recusively in this way we obtain the right orthogonalization

$$u_{\boldsymbol{r}} = \boldsymbol{W}_1^{\mathrm{T}} \boldsymbol{K}_1^{\mathrm{T}} \cdots \boldsymbol{K}_d^{\mathrm{T}}. \tag{19}$$

We may have stopped the orthogonalization process at any point to obtain the partial right orthogonalization

$$u_{\boldsymbol{r}} = \boldsymbol{\Psi}_{\leq i} \boldsymbol{W}_{i+1}^{\mathrm{T}} \boldsymbol{K}_{>i}^{\mathrm{T}}. \tag{20}$$

It is also useful to orthogonalize from the left and right to obtain expansions of the form

$$u_{\boldsymbol{r}} = \boldsymbol{Q}_{\leq i} \boldsymbol{R}_i \boldsymbol{W}_{i+1}^{\mathrm{T}} \boldsymbol{K}_{>i}^{\mathrm{T}}, \tag{21}$$

where the rank of the matrix $\boldsymbol{R}_i \boldsymbol{W}_{i+1}^{\mathrm{T}}$ is the i-th component of the true FTT rank of the tensor $u_{\boldsymbol{r}}$.

Another important operation is truncation of FTT tensors to smaller rank. Efficient algorithms to perform this operation for TT tensors can be found in [44, section 3] and in [15]. Such algorithms are easily adapted to FTT tensors by replacing QR decompositions of matrices with the QR of FTT cores given in (11) and SVD decomposition of matrices with Schmidt decompositions. In numerical implementations, this adaptation amounts to introducing appropriate quadrature weight matrices into the algorithms.

2.2 Tangent and Normal Spaces of Fixed-Rank FTT Manifolds

Let us denote by $V^{(i)}_{r_{i-1}\times r_i}$ the set of all tensor cores $\boldsymbol{\Psi}_i \in M_{r_{i-1}\times r_i}(L^2_{\mu_i}(\Omega_i))$ with the property that the autocovariance matrices $\left\langle \boldsymbol{\Psi}_i^{\mathrm{T}} \boldsymbol{\Psi}_i \right\rangle_i \in M_{r_i \times r_i}(\mathbb{R})$ and $\left\langle \boldsymbol{\Psi}_i \boldsymbol{\Psi}_i^{\mathrm{T}} \right\rangle_i \in M_{r_{i-1}\times r_{i-1}}(\mathbb{R})$ are invertible for $i = 1, \ldots, d$. The set

$$\mathcal{M}_{\boldsymbol{r}} = \{u_{\boldsymbol{r}} \in L^2_\mu(\Omega) : \quad u_{\boldsymbol{r}} = \boldsymbol{\Psi}_1 \boldsymbol{\Psi}_2 \cdots \boldsymbol{\Psi}_d, \quad \boldsymbol{\Psi}_i \in V^{(i)}_{r_{i-1}\times r_i}, \quad \forall i = 1, 2, \ldots, d\}, \tag{22}$$

consisting of fixed-rank FTT tensors, is a smooth Hilbert submanifold of $L^2_\mu(\Omega)$ (see [17]). We represent elements in the tangent space, $T_{u_{\boldsymbol{r}}}\mathcal{M}_{\boldsymbol{r}}$, of $\mathcal{M}_{\boldsymbol{r}}$ at the point $u_{\boldsymbol{r}} \in \mathcal{M}_{\boldsymbol{r}}$ as equivalence classes of velocities of continuously differentiable curves on $\mathcal{M}_{\boldsymbol{r}}$ passing through $u_{\boldsymbol{r}}$

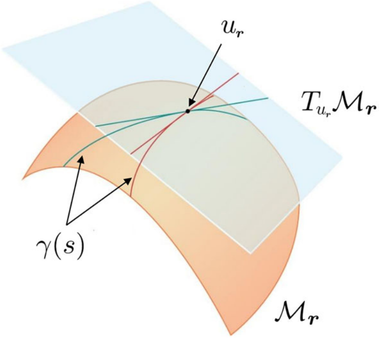

Fig. 1 Sketch of the tensor manifold $\mathcal{M}_r$ and the tangent space $T_{u_r}\mathcal{M}_r$ at $u_r \in \mathcal{M}_r$. The tangent space is defined as equivalence classes of velocities of continuously differentiable curves $\gamma(s)$ on $\mathcal{M}_r$ passing through u_r

$$T_{u_r}\mathcal{M}_r = \left\{\gamma'(s)|_{s=0} : \quad \gamma \in \mathcal{C}^1\left((-\delta,\delta),\mathcal{M}_r\right), \quad \gamma(0) = u_r\right\}. \tag{23}$$

A sketch of $\mathcal{M}_r$ and $T_{u_r}\mathcal{M}_r$ is provided in Fig. 1. Since $L^2_\mu(\Omega)$ is an inner product space, for each $u \in L^2_\mu(\Omega)$ the tangent space $T_u L^2_\mu(\Omega)$ is canonically isomorphic to $L^2_\mu(\Omega)$. Moreover, for each $u_r \in \mathcal{M}_r$ the normal space to $\mathcal{M}_r$ at the point u_r, denoted by $N_{u_r}\mathcal{M}_r$, consists of all vectors in $L^2_\mu(\Omega)$ that are orthogonal to $T_{u_r}\mathcal{M}_r$ with respect to the inner product in $L^2_\mu(\Omega)$

$$N_{u_r}\mathcal{M}_r = \{w \in L^2_\mu(\Omega) : \langle w, v\rangle_{L^2_\mu(\Omega)} = 0, \quad \forall v \in T_{u_r}\mathcal{M}_r\}. \tag{24}$$

Since the tangent space $T_{u_r}\mathcal{M}_r$ is closed, for each point $u_r \in \mathcal{M}_r$ the space $L^2_\mu(\Omega)$ admits a decomposition into tangential and normal components

$$L^2_\mu(\Omega) = T_{u_r}\mathcal{M}_r \oplus N_{u_r}\mathcal{M}_r. \tag{25}$$

3 Dynamic Tensor Approximation of Nonlinear PDEs

The idea of dynamic tensor approximation is to project the time derivative of a low-rank tensor onto the tangent space of the corresponding low-rank tensor manifold at each time. Such a projection results in evolution equations on the low-rank tensor manifold, and can be used to solve initial/boundary value problem of the form (1). This approximation technique is known in the quantum physics community as Dirac–Frenkel/Mclachlan variational principle [26,40,45]. Dynamic approximation has been recently studied by Lubich et al. [34,35,38,43] for finite-dimensional rank-structured manifolds embedded in Euclidean spaces. There have also been extensions to the Tucker format on tensor Banach spaces [21] and tree-based tensor formats on tensor Banach spaces [22].

3.1 Dynamic Tensor Approximation on Low-Rank FTT Manifolds

Let us briefly describe the method of dynamic tensor approximation for the low-rank FTT manifold (22). First we define a projection onto the tangent space of $\mathcal{M}_r$ at u_r by

$$\begin{aligned} P_{u_r} &: L^2_\mu(\Omega) \to T_{u_r}\mathcal{M}_r \\ &P_{u_r} v = \text{argmin}_{v_r \in T_{u_r}\mathcal{M}_r} \|v - v_r\|_{L^2_\mu(\Omega)}. \end{aligned} \tag{26}$$

For fixed $\boldsymbol{u_r}$, the map P_{u_r} is linear and bounded. Each $v \in L^2_\mu(\Omega)$ admits a unique representation as $v = v_t + v_n$ where $v_t \in T_{u_r}\mathcal{M}_{\boldsymbol{r}}$ and $v_n \in N_{u_r}\mathcal{M}_{\boldsymbol{r}}$ (see Eq. 25). From this representation it is clear that P_{u_r} is an orthogonal projection onto the tangent space $T_{u_r}\mathcal{M}_{\boldsymbol{r}}$. If the initial condition $u_0(\boldsymbol{x})$ is on the manifold $\mathcal{M}_{\boldsymbol{r}}$, then the solution to the initial/boundary value problem

$$\begin{cases} \dfrac{\partial u_{\boldsymbol{r}}}{\partial t} = P_{u_r} G(u_{\boldsymbol{r}}), \\ u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), \end{cases} \tag{27}$$

remains on the manifold $\mathcal{M}_{\boldsymbol{r}}$ for all $t \geq 0$. Here G is the nonlinear operator on the right hand side of equation (1). The solution to (27) is known as a dynamic approximation to the solution of (1). In the context of separable Hilbert spaces, the dynamic approximation problem (27) can be solved using dynamically orthogonal or bi-orthogonal constraints on tensor modes [16,17]. Such constraints, also referred to as gauge conditions, provide the unique solution of the minimization problem (26) with different FTT cores. However, in the presence of repeated eigenvalues the bi-orthogonal constraints result in singular equations for the tangent space projection (26). Hereafter we recall the equations which allow us to compute (26) with FTT cores subject to dynamically orthogonal (DO) constraints.

First, expand $u_r \in \mathcal{M}_{\boldsymbol{r}}$ in terms of FTT cores $u_{\boldsymbol{r}} = \boldsymbol{\Psi}_1 \boldsymbol{\Psi}_2 \cdots \boldsymbol{\Psi}_d$, where $\boldsymbol{\Psi}_i$ are orthogonalized from the left, i.e., $\left\langle \boldsymbol{\Psi}_i^{\mathrm{T}} \boldsymbol{\Psi}_i \right\rangle_i = \boldsymbol{I}_{r_i \times r_i}$, for all $i = 1, \ldots, d-1$. With this ansatz, an arbitrary element of the tangent space $T_{u_r}\mathcal{M}_{\boldsymbol{r}}$ can be expressed as

$$\dot{u}_{\boldsymbol{r}} = \dot{\boldsymbol{\Psi}}_1 \boldsymbol{\Psi}_{\geq 2} + \cdots + \boldsymbol{\Psi}_{\leq i-1} \dot{\boldsymbol{\Psi}}_i \boldsymbol{\Psi}_{\geq i+1} + \cdots + \boldsymbol{\Psi}_{\leq d-1} \dot{\boldsymbol{\Psi}}_d, \tag{28}$$

where $\dot{u}_{\boldsymbol{r}} = \partial u_{\boldsymbol{r}} / \partial t$ and $\dot{\boldsymbol{\Psi}}_i = \partial \boldsymbol{\Psi}_i / \partial t$. The DO constraints are given by

$$\left\langle \dot{\boldsymbol{\Psi}}_i^{\mathrm{T}} \boldsymbol{\Psi}_i \right\rangle_i = \boldsymbol{0}_{r_i \times r_i}, \qquad i = 1, \ldots, d-1, \tag{29}$$

which ensures that $\left\langle \boldsymbol{\Psi}_i^{\mathrm{T}}(t) \boldsymbol{\Psi}_i(t) \right\rangle_i = \boldsymbol{I}_{r_i \times r_i}$ for all $i = 1, \ldots, d-1$ and for all $t \geq 0$. We have shown in [17] that under these constraints, the convex minimization problem (26) admits a unique minimum for vectors in the tangent space (28) satisfying the PDE system

$$\begin{aligned} \dot{\boldsymbol{\Psi}}_1 &= \left[\left\langle G(u_{\boldsymbol{r}}) \boldsymbol{\Psi}_{\geq 2}^{\mathrm{T}} \right\rangle_{\geq 2} - \boldsymbol{\Psi}_1 \left\langle \boldsymbol{\Psi}_1^{\mathrm{T}} G(u_{\boldsymbol{r}}) \boldsymbol{\Psi}_{\geq 2}^{\mathrm{T}} \right\rangle_{\geq 1} \right] \left\langle \boldsymbol{\Psi}_{\geq 2} \boldsymbol{\Psi}_{\geq 2}^{\mathrm{T}} \right\rangle_{\geq 2}^{-1}, \\ \dot{\boldsymbol{\Psi}}_k &= \left[\left\langle \boldsymbol{\Psi}_{\leq k-1}^{\mathrm{T}} G(u_{\boldsymbol{r}}) \boldsymbol{\Psi}_{\geq k+1}^{\mathrm{T}} \right\rangle_{\leq k-1, \geq k+1} - \right. \\ &\qquad \left. \boldsymbol{\Psi}_k \left\langle \boldsymbol{\Psi}_{\leq k}^{\mathrm{T}} G(u_{\boldsymbol{r}}) \boldsymbol{\Psi}_{\geq k+1}^{\mathrm{T}} \right\rangle_{\geq 1} \right] \left\langle \boldsymbol{\Psi}_{\geq k+1} \boldsymbol{\Psi}_{\geq k+1}^{\mathrm{T}} \right\rangle_{\geq k+1}^{-1}, \qquad k = 2, 3, \ldots, d-1, \\ \dot{\boldsymbol{\Psi}}_d &= \left\langle \boldsymbol{\Psi}_{\leq d-1}^{\mathrm{T}} G(u_{\boldsymbol{r}}) \right\rangle_{\leq d-1}. \end{aligned} \tag{30}$$

Here, $u_{\boldsymbol{r}} = \boldsymbol{\Psi}_1 \boldsymbol{\Psi}_2 \cdots \boldsymbol{\Psi}_d \in \mathcal{M}_r$ and we have introduced the notation

$$\begin{aligned} \langle \boldsymbol{\Psi} \rangle_{\leq k} &= \int_{\Omega_1 \times \cdots \times \Omega_k} \boldsymbol{\Psi}(\boldsymbol{x}) d\mu_1(x_1) \cdots \mu_k(x_k), \\ \langle \boldsymbol{\Psi} \rangle_{\geq k} &= \int_{\Omega_k \times \cdots \times \Omega_d} \boldsymbol{\Psi}(\boldsymbol{x}) d\mu_k(x_k) \cdots \mu_d(x_d), \\ \langle \boldsymbol{\Psi} \rangle_{\leq k-1, \geq k+1} &= \int_{\Omega_1 \times \cdots \times \Omega_{k-1} \times \Omega_{k+1} \times \cdots \times \Omega_d} \\ &\qquad \boldsymbol{\Psi}(\boldsymbol{x}) d\mu_1(x_1) \cdots \mu_{k-1}(x_{k-1}) \mu_{k+1}(x_{k+1}) \cdots \mu_d(x_d), \end{aligned} \tag{31}$$

for any matrix $\boldsymbol{\Psi}(\boldsymbol{x}) \in M_{r\times s}\left(L^2_\mu(\Omega)\right)$. The DO-FTT system (30) involves several inverse covariance matrices $\left\langle \boldsymbol{\Psi}_{\geq k}\boldsymbol{\Psi}^{\mathrm{T}}_{\geq k}\right\rangle^{-1}_{\geq k}$, which can become poorly conditioned in the presence of tensor modes with small energy (i.e. autocovariance matrices with small singular values). This phenomenon has been shown to be a result of the fact that the curvature of the tensor manifold at a tensor is inversely proportional to the smallest singular value present in the tensor [34, section 4]. A slight improvement to the numerical stability of (30) can be obtained by right orthogonalizing the partial products

$$\boldsymbol{\Psi}_{\geq k} = \boldsymbol{R}_k^{\mathrm{T}}\boldsymbol{Q}^{\mathrm{T}}_{\geq k}, \qquad k = 2, \ldots, d. \tag{32}$$

Using the orthogonality of $\boldsymbol{Q}_k$ it can easily be verified that $\boldsymbol{R}_k = \left\langle \boldsymbol{\Psi}_{\geq k}\boldsymbol{\Psi}^{\mathrm{T}}_{\geq k}\right\rangle^{1/2}_{\geq k}$. With these right orthogonalized cores, the DO-FTT system (30) can be written as

$$\begin{aligned}
\dot{\boldsymbol{\Psi}}_1 &= \left[\left\langle G(u_{\boldsymbol{r}})\,\boldsymbol{Q}_{\geq 2}\right\rangle_{\geq 2} - \boldsymbol{\Psi}_1\left\langle \boldsymbol{\Psi}_1^{\mathrm{T}} G(u_{\boldsymbol{r}})\,\boldsymbol{Q}_{\geq 2}\right\rangle_{\geq 1}\right]\left\langle \boldsymbol{\Psi}_{\geq 2}\boldsymbol{\Psi}^{\mathrm{T}}_{\geq 2}\right\rangle^{-1/2}_{\geq 2},\\
\dot{\boldsymbol{\Psi}}_k &= \Big[\left\langle \boldsymbol{\Psi}^{\mathrm{T}}_{\leq k-1} G(u_{\boldsymbol{r}})\,\boldsymbol{Q}_{\geq k+1}\right\rangle_{\leq k-1,\geq k+1} - \\
&\qquad \boldsymbol{\Psi}_k\left\langle \boldsymbol{\Psi}^{\mathrm{T}}_{\leq k} G(u_{\boldsymbol{r}})\,\boldsymbol{Q}_{\geq k+1}\right\rangle_{\geq 1}\Big]\left\langle \boldsymbol{\Psi}_{\geq k+1}\boldsymbol{\Psi}^{\mathrm{T}}_{\geq k+1}\right\rangle^{-1/2}_{\geq k+1}, \qquad k = 2, 3, \ldots, d-1,\\
\dot{\boldsymbol{\Psi}}_d &= \left\langle \boldsymbol{\Psi}^{\mathrm{T}}_{\leq d-1} G(u_{\boldsymbol{r}})\right\rangle_{\leq d-1},
\end{aligned} \tag{33}$$

where $\left\langle \boldsymbol{\Psi}_{\geq k}\boldsymbol{\Psi}^{\mathrm{T}}_{\geq k}\right\rangle^{-1/2}_{k,\ldots,d}$ denotes the inverse of the matrix square root. Since the condition number of $\left\langle \boldsymbol{\Psi}_{\geq k}\boldsymbol{\Psi}^{\mathrm{T}}_{\geq k}\right\rangle_{\geq k}$ is larger than the condition number of $\left\langle \boldsymbol{\Psi}_{\geq k}\boldsymbol{\Psi}^{\mathrm{T}}_{\geq k}\right\rangle^{1/2}_{\geq k}$, the inverse covariances at the right hand side of (33) can be computed more accurately than the ones in (30) in the presence of small singular values.

3.1.1 Temporal Integration Using Operator Splitting Methods

As we mentioned previously, one of the challenges of dynamic approximation of PDEs on low-rank tensor manifolds relates to the curvature of the manifold, which is proportional to the inverse of the smallest singular value of $\left\langle \boldsymbol{\Psi}_{\geq k}\boldsymbol{\Psi}^{\mathrm{T}}_{\geq k}\right\rangle_{\geq k}$ [34, section 4]. Such curvature appears naturally at the right hand side of the DO-FTT system (30) in the form of inverse covariances $\left\langle \boldsymbol{\Psi}_{\geq k}\boldsymbol{\Psi}^{\mathrm{T}}_{\geq k}\right\rangle^{-1}_{\geq k}$. Clearly, if the tensor solution is comprised of cores with small singular values, then the covariance matrices $\left\langle \boldsymbol{\Psi}_{\geq k}\boldsymbol{\Psi}^{\mathrm{T}}_{\geq k}\right\rangle_{\geq k}$ are ill-conditioned and therefore not easily invertible. Moreover, it is desirable to add and remove tensor modes adaptively during temporal integration, and adding a mode with zero energy immediately yields singular covariance matrices (see [16]). The problem of inverting the covariance matrices $\left\langle \boldsymbol{\Psi}_{\geq k}\boldsymbol{\Psi}^{\mathrm{T}}_{\geq k}\right\rangle_{\geq k}$ when integrating (30) or (33) can be avoided by using projector-splitting methods. These methods were originally proposed for integration on tensor manifolds by Lubich et al. in [32,38,39]. The key idea is to apply an exponential operator splitting scheme, e.g., the Lie–Trotter scheme, directly to the projection operator onto the tangent space defining the dynamic approximation (see Eq. 26). To describe the method, we begin by introducing a general framework for operator splitting of dynamics on the FTT tangent space. We first rewrite the right hand side of (27) as

$$
\begin{aligned}
P_{u_{\boldsymbol{r}}} G(u_{\boldsymbol{r}}) &= \dot{\boldsymbol{\Psi}}_1 \boldsymbol{\Psi}_{\geq 2} + \boldsymbol{\Psi}_1 \dot{\boldsymbol{\Psi}}_2 \boldsymbol{\Psi}_{\geq 3} + \cdots + \boldsymbol{\Psi}_{\leq d-1} \dot{\boldsymbol{\Psi}}_d \\
&= \dot{\boldsymbol{\Psi}}_1 \left\langle \boldsymbol{\Psi}_{\geq 2} \boldsymbol{\Psi}_{\geq 2}^{\mathrm{T}} \right\rangle_{\geq 2}^{1/2} \boldsymbol{Q}_{\geq 2}^{\mathrm{T}} + \boldsymbol{\Psi}_1 \dot{\boldsymbol{\Psi}}_2 \left\langle \boldsymbol{\Psi}_{\geq 3} \boldsymbol{\Psi}_{\geq 3}^{\mathrm{T}} \right\rangle_{\geq 3}^{1/2} \boldsymbol{Q}_{\geq 3}^{\mathrm{T}} + \cdots + \boldsymbol{\Psi}_{\leq d-1} \dot{\boldsymbol{\Psi}}_d,
\end{aligned}
\tag{34}
$$

where in the second line we used the right orthogonalizations in Eq. (32). A substitution of the expressions for $\dot{\boldsymbol{\Psi}}_k$ we obtained in (33) into (34) yields

$$
P_{u_{\boldsymbol{r}}} G(u_{\boldsymbol{r}}) = P_d^{+} G(u_{\boldsymbol{r}}) + \sum_{i=1}^{d-1} P_i^{+} G(u_{\boldsymbol{r}}) - P_i^{-} G(u_{\boldsymbol{r}}),
\tag{35}
$$

where we defined the following projection operators from $L_\mu^2(\Omega)$ onto $T_{u_{\boldsymbol{r}}} \mathcal{M}_{\boldsymbol{r}}$

$$
\begin{aligned}
P_k^{+} z(\boldsymbol{x}) &= \boldsymbol{\Psi}_{\leq k-1} \left\langle \boldsymbol{\Psi}_{\leq k-1}^{\mathrm{T}} z(\boldsymbol{x}) \boldsymbol{Q}_{\geq k+1} \right\rangle_{\leq k-1, \geq k+1} \boldsymbol{Q}_{\geq k+1}^{\mathrm{T}}, \qquad k = 1, \ldots, d, \\
P_k^{-} z(\boldsymbol{x}) &= \boldsymbol{\Psi}_{\leq k} \left\langle \boldsymbol{\Psi}_{\leq k}^{\mathrm{T}} z(\boldsymbol{x}) \boldsymbol{Q}_{\geq k+1} \right\rangle_{\geq 1} \boldsymbol{Q}_{\geq k+1}^{\mathrm{T}}, \qquad k = 1, \ldots, d-1,
\end{aligned}
\tag{36}
$$

for any $z(\boldsymbol{x}) \in L_\mu^2(\Omega)$. Also we set $\boldsymbol{\Psi}_0 = 1$. The key point in (35) is that inverse covariance matrices no longer appear. To establish a general operator splitting framework, let us assume that there exists an evolution operator $\mathcal{E}_{P_{u_{\boldsymbol{r}}} G}$ for the solution of the initial/boundary value problem (27), where $P_{u_{\boldsymbol{r}}} G$ is given in (35). Such an evolution operator $\mathcal{E}_{P_{u_{\boldsymbol{r}}} G} : L_\mu^2(\Omega) \times [0, T] \to L_\mu^2(\Omega)$ satisfies a semi-group property and it maps the initial condition $u_0(\boldsymbol{x})$ into the solution to (27) at a later time

$$
u(\boldsymbol{x}, t) = \mathcal{E}_{P_{u_{\boldsymbol{r}}} G}(u_0(\boldsymbol{x}), t).
\tag{37}
$$

We write such an evolution operator formally as an exponential operator with generator $D_{P_{u_{\boldsymbol{r}}} G}$ (see e.g. [36])

$$
u(\boldsymbol{x}, t) = e^{t D_{P_{u_{\boldsymbol{r}}} G}} u_0(\boldsymbol{x}), \qquad 0 \leq t \leq T,
\tag{38}
$$

where $D_{P_{u_{\boldsymbol{r}}} G}$ is the Lie derivative associated with $P_{u_{\boldsymbol{r}}} G$. We now discretize the temporal domain of interest $[0, T]$ into $N + 1$ evenly-spaced time instants,

$$
t_i = i \Delta t, \qquad \Delta t = \frac{T}{N}, \qquad i = 0, 1, \ldots, N.
\tag{39}
$$

An approximation to the exact solution of (27) is then obtained by the recurrence relation

$$
u_{\boldsymbol{r}}(\boldsymbol{x}, t_{n+1}) \approx \mathcal{S}(\Delta t, u_{\boldsymbol{r}}(\boldsymbol{x}, t_n)),
\tag{40}
$$

where $\mathcal{S}$ is an exponential splitting operator that approximates the exact evolution operator

$$
\mathcal{S}(t, \cdot) = \prod_{i=1}^{s} \left[e^{\gamma_{i,d} t P_d^{+} G} \prod_{j=1}^{d-1} \left(e^{\gamma_{i,j} t P_j^{+} G} e^{\gamma_{i,j} t P_j^{-} G} \right) \right].
\tag{41}
$$

Setting $s = 1$ and $\gamma_{1,j} = 1$ for all $j = 1, \ldots, d$ in (41) yields the well-known Lie–Trotter splitting, which is first-order in time. The discrete time version of this scheme can be written as

$$\begin{cases} u_1^+(t_{i+1}) = u_1^+(t_i) + \Delta t P_1^+ G(u_{\boldsymbol{r}}), & u_1^+(t_i) = u_{\boldsymbol{r}}(t_i), \\ u_1^-(t_{i+1}) = u_1^-(t_i) - \Delta t P_1^- G(u_{\boldsymbol{r}}), & u_1^-(t_i) = u_1^+(t_{i+1}), \\ \quad\vdots & \\ u_j^+(t_{i+1}) = u_j^+(t_i) + \Delta t P_j^+ G(u_{\boldsymbol{r}}), & u_j^+(t_i) = u_{j-1}^-(t_{i+1}), \\ u_j^-(t_{i+1}) = u_j^-(t_i) - \Delta t P_j^- G(u_{\boldsymbol{r}}), & u_j^-(t_i) = u_j^+(t_{i+1}), \\ \quad\vdots & \\ u_d^+(t_{i+1}) = u_d^+(t_i) + \Delta t P_d^+ G(u_{\boldsymbol{r}}), & u_d(t_i) = u_{d-1}^-(t_{i+1}), \\ u_{\boldsymbol{r}}(t_{i+1}) = u_d^+(t_{i+1}). & \end{cases} \tag{42}$$

This allows us to compute $u_{\boldsymbol{r}}(t_{i+1})$ given $u_{\boldsymbol{r}}(t_i)$. Although each equation in (42) involves a FTT tensor, it was shown in [38, Theorem 4.1] that each equation only updates one tensor core. Clearly this is computationally more efficient than updating a full tensor. Moreover, in (42) there is no need to invert covariance matrices, which is a distinct advantage over iterating a discrete form of (30) or (33).

Regarding computational cost, suppose we discretize the d-dimensional domain Ω using a tensor product grid with n points per dimension. It was pointed out in [38] that the computational complexity of the sweeping algorithm to update the tensor cores for the Lie–Trotter scheme (42) applied to a linear PDE (i.e. Eq. 27 with linear G) is linear in the dimension d but has high polynomial complexity in the tensor rank. On the other hand, discretizing such linear PDE on the same tensor product grid and performing one time step with a first-order time stepping scheme (e.g. Euler forward) has computational complexity which scales exponentially with the dimension d. Specifically, assuming that the operator G in (1) is linear with rank r_G (see [5]), the computational cost of one time step of Euler forward is $dn^{d+1}r_G + n^d r_G$ floating point operations, hence exponential in d.

3.2 Step-Truncation Temporal Integration Methods

Another methodology to integrate nonlinear PDEs on fixed-rank tensor manifolds $\mathcal{M}_{\boldsymbol{r}}$ is step-truncation [33,50,51]. The idea is to integrate the solution off of $\mathcal{M}_{\boldsymbol{r}}$ for short time, e.g., by performing one time step of the full equation with a conventional time-stepping scheme, followed by a truncation operation back onto $\mathcal{M}_{\boldsymbol{r}}$. To describe this method further let us define the truncation operator

$$\begin{aligned} \mathfrak{T}_{\boldsymbol{r}} : L^2_\mu(\Omega) &\to \mathcal{M}_{\boldsymbol{r}} \\ \mathfrak{T}_{\boldsymbol{r}}(u) &= \operatorname{argmin}_{u_{\boldsymbol{r}} \in \mathcal{M}_{\boldsymbol{r}}} \|u - u_{\boldsymbol{r}}\|_{L^2_\mu(\Omega)}, \end{aligned} \tag{43}$$

which provides the best approximation of u on $\mathcal{M}_{\boldsymbol{r}}$. Such a map is known as a metric projection or closest point function and in general it may be multivalued, i.e., the set of $u_{\boldsymbol{r}} \in \mathcal{M}_{\boldsymbol{r}}$ which minimize $\|u - u_{\boldsymbol{r}}\|_{L^2_\mu(\Omega)}$ is not a singleton set. However, since $\mathcal{M}_{\boldsymbol{r}}$ is a smooth submanifold of $L^2_\mu(\Omega)$, we have by [53, Proposition 5.1] that for each $u_0 \in \mathcal{M}_{\boldsymbol{r}}$ there exists an open neighborhood U of u_0 such that $\mathfrak{T}_{\boldsymbol{r}}$ is well-defined and smooth on U. Let

$$u(\boldsymbol{x}, t_{k+1}) = u(\boldsymbol{x}, t_k) + \Delta t \Phi\left(G, u(\boldsymbol{x}, t_k), \Delta t\right) \tag{44}$$

be a convergent one-step time integration scheme[4] approximating the solution to the initial value problem (1). Assume that the solution $u(\boldsymbol{x}, t_0)$ at time t_0 is on $\mathcal{M}_{\boldsymbol{r}}$.[5] In order to guarantee the solution $u(\boldsymbol{x}, t_k)$ at time step t_k is an element of the manifold $\mathcal{M}_{\boldsymbol{r}}$ for each $k = 1, 2, \ldots$, we apply the truncation operator to the right hand side (44). This yields the following step-truncation method

$$u_{\boldsymbol{r}}(\boldsymbol{x}, t_{k+1}) = \mathfrak{T}_{\boldsymbol{r}}\left(u_{\boldsymbol{r}}(\boldsymbol{x}, t_k) + \Delta t \Phi\left(G, u_{\boldsymbol{r}}(\boldsymbol{x}, t_k), \Delta t\right)\right). \tag{45}$$

3.3 Consistency of Dynamic Approximation and Step-Truncation Methods

Next we ask what happens in the step-truncation algorithm in the limit of time step Δt approaching zero. The result of such a limiting procedure results in a scheme which keeps the solution $u(\boldsymbol{x}, t)$ on the manifold $\mathcal{M}_{\boldsymbol{r}}$ for all time $t \geq t_0$ in an optimal way. We now show that this limiting procedure in fact results in precisely the dynamic approximation method described in Sect. 3.1. In other words, by sending Δt to zero in (45) we obtain a solution of (27). For similar discussions connecting these two approximation methods in closely related contexts see [23,24,33]. To prove consistency between step-truncation and dynamic approximation methods we need to compute $\mathfrak{T}_{\boldsymbol{r}}(u(\boldsymbol{x}, t))$ for t infinitesimally close to t_0. Such a quantity depends on the derivative

$$\left.\frac{\partial \mathfrak{T}_{\boldsymbol{r}}(u(\boldsymbol{x}, t))}{\partial t}\right|_{t=t_0} = \lim_{\Delta t \to 0} \frac{\mathfrak{T}_{\boldsymbol{r}}(u(\boldsymbol{x}, t)) - \mathfrak{T}_{\boldsymbol{r}}(u(\boldsymbol{x}, t_0))}{\Delta t}. \tag{46}$$

The following proposition provides a representation of the derivative $\partial \mathfrak{T}_{\boldsymbol{r}}(u(\boldsymbol{x}, t))/\partial t$ in terms of $G(u(\boldsymbol{x}, t))$ and the Fréchet derivative [56] of the operator $\mathfrak{T}_{\boldsymbol{r}}(u)$.

Proposition 3.1 *If the solution $u_0 = u(\boldsymbol{x}, t_0)$ to (1) at time t_0 is on the manifold $\mathcal{M}_{\boldsymbol{r}}$, then*

$$\left.\frac{\partial \mathfrak{T}_{\boldsymbol{r}}(u(\boldsymbol{x}, t))}{\partial t}\right|_{t=t_0} = (\mathfrak{T}_{\boldsymbol{r}})'_{u_0} G(u(\boldsymbol{x}, t)), \tag{47}$$

where $(\mathfrak{T}_{\boldsymbol{r}})'_{u_0}$ is the Fréchet derivative of the nonlinear operator $\mathfrak{T}_{\boldsymbol{r}}$ at the point u_0.

Proof Express the solution of (1) at time $t \geq t_0$ as

$$u(\boldsymbol{x}, t) = u_0(\boldsymbol{x}) + h(\boldsymbol{x}, t), \tag{48}$$

where

$$h(\boldsymbol{x}, t) = \int_{t_0}^{t} G(u(\boldsymbol{x}, \tau)) d\tau. \tag{49}$$

Expanding $\mathfrak{T}_{\boldsymbol{r}}(u(\boldsymbol{x}, t))$ in a Taylor series around $u_0(\boldsymbol{x})$ we obtain [42, Theorem 6.1]

$$\mathfrak{T}_{\boldsymbol{r}}(u(\boldsymbol{x}, t)) = u_0(\boldsymbol{x}) + (\mathfrak{T}_{\boldsymbol{r}})'_{u_0} h(\boldsymbol{x}, t) + \frac{1}{2}(\mathfrak{T}_{\boldsymbol{r}})''_{u_0} h(\boldsymbol{x}, t)^2 + \cdots. \tag{50}$$

Differentiating (50) with respect to t and evaluating at $t = t_0$ we obtain

$$\left.\frac{\partial \mathfrak{T}_{\boldsymbol{r}}(u(\boldsymbol{x}, t))}{\partial t}\right|_{t=t_0} = (\mathfrak{T}_{\boldsymbol{r}})'_{u_0} G(u(\boldsymbol{x}, t_0)), \tag{51}$$

[4] Time stepping schemes of the form (44) include Runge–Kutta methods and linear multi-step methods [50].

[5] If $u(\boldsymbol{x}, t_0)$ is not on $\mathcal{M}_{\boldsymbol{r}}$ then it may be mapped onto $\mathcal{M}_{\boldsymbol{r}}$ by evaluating $\mathfrak{T}_{\boldsymbol{r}}(u(\boldsymbol{x}, t_0))$.

where we assumed that $\partial/\partial t$ commutes with $(\mathfrak{T}_{\boldsymbol{r}})'_{u_0}$ and used the fact that $\partial h(\boldsymbol{x}, t)/\partial t = G(u(\boldsymbol{x}, t))$ for the first order term. All of the higher order terms are seen to be zero by commuting $\partial/\partial t$ with $(\mathfrak{T}_{\boldsymbol{r}})^{(n)}_{u_0}$ and using chain rule. □

Since $\mathfrak{T}_{\boldsymbol{r}}(u(\boldsymbol{x}, t))$ is an element of $\mathcal{M}_{\boldsymbol{r}}$ for all $t \geq t_0$, it follows that (47) is an element of $T_{u_0}\mathcal{M}_{\boldsymbol{r}}$. Arguing on the optimality of the tangent space element $(\mathfrak{T}_{\boldsymbol{r}})'_{u_0} G(u(\boldsymbol{x}, t_0))$ it is seen that (51) is the same problem as dynamic approximation (27), i.e., $(\mathfrak{T}_{\boldsymbol{r}})'_{u_0} = P_{u_0}$. Now consider the scheme (45) and use a Taylor expansion of $\mathfrak{T}_{\boldsymbol{r}}$ around $u_{\boldsymbol{r}}(\boldsymbol{x}, t_k)$ on the right hand side

$$u_{\boldsymbol{r}}(\boldsymbol{x}, t_{k+1}) = u_{\boldsymbol{r}}(\boldsymbol{x}, t_k) + \Delta t (\mathfrak{T}_{\boldsymbol{r}})' \Phi\left(G, u_{\boldsymbol{r}}(\boldsymbol{x}, t_k), \Delta t\right) + O(\Delta t^2). \tag{52}$$

Discarding higher order terms in Δt yields

$$u_{\boldsymbol{r}}(\boldsymbol{x}, t_{k+1}) \simeq u_{\boldsymbol{r}}(\boldsymbol{x}, t_k) + \Delta t P_{u_{\boldsymbol{r}}} \Phi\left(G, u_{\boldsymbol{r}}(\boldsymbol{x}, t_k), \Delta t\right). \tag{53}$$

Moreover if the increment function Φ defines the Euler forward scheme

$$\Phi(G, u_{\boldsymbol{r}}(\boldsymbol{x}, t_k), \Delta t) = G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_k)), \tag{54}$$

then the scheme (53) is equivalent to the scheme in (27). Thus, we just proved the following lemma.

Lemma 3.1 *Step-truncation and dynamic approximation methods are consistent at least to first-order in Δt.*

This Lemma applies to any first-order time integrator for dynamic approximation and step-truncation, including the Lie–Trotter splitting integrator we discussed in Sect. 3.1.1.

4 Rank-Adaptive Integration

The solution to the initial/boundary value problem (1) is often not accurately represented on a tensor manifold with fixed rank, even for short integration times. In this section we discuss effective methods to adaptively add and remove tensor modes from the solution based on appropriate criteria.

In the context of step-truncation algorithms, if the solution rank naturally decreases in time then the operator $\mathfrak{T}_{\boldsymbol{r}}$ in (45) is no longer well-defined. In this situation, replacing the operator $\mathfrak{T}_{\boldsymbol{r}}$ with $\mathfrak{T}_{\boldsymbol{s}}$ for an appropriate[6] $\boldsymbol{s} \leq \boldsymbol{r}$ allows for integration to continue. On the other hand, if the solution rank increases in during integration then the operator $\mathfrak{T}_{\boldsymbol{r}}$ will still be well-defined for small enough Δt but the approximation on $\mathcal{M}_{\boldsymbol{r}}$ will not retain accuracy. To address this problem of constant rank integration we shall introduce a criterion for rank increase of the FTT solution. Both decreasing and increasing rank are based on FTT orthogonalization and truncation (see Sect. 2.1). For the remainder of this section let $u(\boldsymbol{x}, t)$ be the solution to (1) and $u_{\boldsymbol{r}}(\boldsymbol{x}, t) \in \mathcal{M}_{\boldsymbol{r}}$ an approximation of $u(\boldsymbol{x}, t)$ obtained by either the solution of the dynamical approximation problem (27) or step-truncation methods (see Sect. 3.2).

4.1 Decreasing Tensor Rank

For decreasing tensor rank at time t, we are interested in determining if $u_{\boldsymbol{r}}(\boldsymbol{x}, t) \in \mathcal{M}_{\boldsymbol{r}}$ is close to an element $u_{\boldsymbol{s}}(\boldsymbol{x}, t) \in \mathcal{M}_{\boldsymbol{s}}$ for $\boldsymbol{s} \leq \boldsymbol{r}$. This can be achieved by simply performing a

[6] Here $\leq$ denotes component-wise inequality of rank vectors, i.e., $\boldsymbol{s} < \boldsymbol{r}$ if and only if $s_i \leq r_i$ for all $i = 0, 1, \ldots, d$.

FTT truncation on $u_{\boldsymbol{r}}(\boldsymbol{x}, t)$ with small threshold ϵ_{dec}. Since the splitting integrator described in Sect. 3.1.1 is robust to over approximation by tensor rank, it may not be strictly necessary to decrease rank during integration. However, it is desirable to have solutions of the lowest rank possible (while retaining accuracy) when solving high dimensional problems. For these reasons it is advisable not to perform a FTT truncation at each time step (as this would be unnecessary and inefficient when using an operator splitting integrator) but only every once and a while. One may choose a criterion for when to check for rank decrease based on the problem, step size, current rank, and dimension. If one is using a step-truncation method with a tolerance based FTT truncation algorithm such as the one described in Sect. 2.1 then rank decrease is already built into each time step.

4.2 Increasing Tensor Rank

As a general heuristic one would like to increase rank at the time when the error between the low-rank approximation $u_{\boldsymbol{r}}(\boldsymbol{x}, t)$ and the PDE solution $u(\boldsymbol{x}, t)$ will become large after the subsequent time step. Such critical time instant for rank increase can be determined by examining the normal component of the dynamics

$$N_{u_{\boldsymbol{r}}} G(u_{\boldsymbol{r}}) = G(u_{\boldsymbol{r}}) - P_{u_{\boldsymbol{r}}} G(u_{\boldsymbol{r}}). \tag{55}$$

To describe this situation further, suppose we are integrating one time step forward from t_i to t_{i+1}. The error at t_{i+1} is given by

$$\begin{aligned} E(t_i, t_{i+1}) &= u_{\boldsymbol{r}}(\boldsymbol{x}, t_{i+1}) - u(\boldsymbol{x}, t_{i+1}) \\ &= u(\boldsymbol{x}, t_i) + \int_{t_i}^{t_{i+1}} G(u(\boldsymbol{x}, \tau)) d\tau \\ &\quad - \left(\mathfrak{T}_{\boldsymbol{r}}(u(\boldsymbol{x}, t_i)) + \int_{t_i}^{t_{i+1}} P_{u_{\boldsymbol{r}}(\boldsymbol{x},\tau)} G(u_{\boldsymbol{r}}(\boldsymbol{x}, \tau)) d\tau \right). \end{aligned} \tag{56}$$

If $u(\boldsymbol{x}, t_i) \in \mathcal{M}_{\boldsymbol{r}}$ then

$$E(t_i, t_{i+1}) = \int_{t_i}^{t_{i+1}} \left[G(u(\boldsymbol{x}, \tau)) - P_{u_{\boldsymbol{r}}(\boldsymbol{x},\tau)} G(u_{\boldsymbol{r}}(\boldsymbol{x}, \tau)) \right] d\tau. \tag{57}$$

For small Δt the above integral can be approximated by the left endpoint

$$\begin{aligned} E(t_i, t_{i+1}) &= \Delta t \left(G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_i)) - P_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)} G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_i)) \right) + O(\Delta t^2) \\ &= \Delta t N_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)} G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_i)) + O(\Delta t^2), \end{aligned} \tag{58}$$

where $N_{u_{\boldsymbol{r}}(\boldsymbol{x},\tau)}$ denotes the orthogonal projection onto the normal space of $\mathcal{M}_{\boldsymbol{r}}$ at the point $u_{\boldsymbol{r}}(\boldsymbol{x}, t)$. Hence, up to first-order in Δt we have that

$$\| E(t_i, t_{i+1}) \| \simeq \Delta t \| N_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)} G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_i)) \|. \tag{59}$$

From this approximation we see that a reasonable criterion for increasing rank at time t_i is when the norm of the normal component of $G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_i))$ is larger than some threshold ϵ_{inc} (see Fig. 2)

$$\| N_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)} G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_i)) \| > \epsilon_{\text{inc}}. \tag{60}$$

To efficiently compute the normal component $N_{u_{\boldsymbol{r}}\boldsymbol{x},t_i)} G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_i))$ at each time instant t_i we use the formula

$$N_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)} G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_i)) = G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_i)) - P_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)} G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_i)), \tag{61}$$

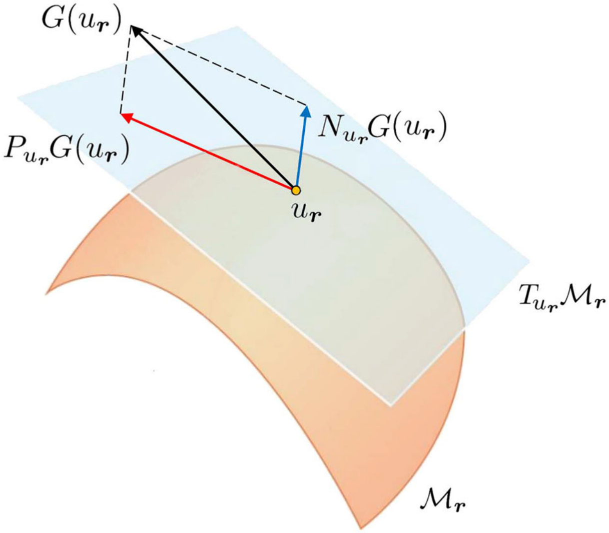

Fig. 2 Tangent and normal components of $G(u_r) = \partial u_r/\partial t$ at u_r. The tensor rank of the solution is increased at time t_i if the norm of the normal component $N_{u_r}G(u_r)$ is larger than a specified threshold ϵ_{inc}

where $N_r G(u_r)$ and $P_{u_r}G(u_r)$ represent the normal and tangential components of $G(u_r)$. The tangential component can be approximated at a low computational cost via backward differentiation formulas (BDF) as

$$\widetilde{P}^{(2)}_{u_r}G(u_r) = \frac{u_r(\boldsymbol{x}, t_i) - u_r(\boldsymbol{x}, t_{i-1})}{\Delta t} + O(\Delta t^2) \qquad \text{(two-point formula)}, \tag{62}$$

$$\widetilde{P}^{(3)}_{u_r}G(u_r) = \frac{3u_r(\boldsymbol{x}, t_i) - 4u_r(\boldsymbol{x}, t_{i-1}) + u_r(\boldsymbol{x}, t_{i-2})}{2\Delta t} + O(\Delta t^3) \qquad \text{(three-point formula)}, \tag{63}$$

$$\widetilde{P}^{(p)}_{u_r}G(u_r) = BD_p(\Delta t, u_r(\boldsymbol{x}, t_i), u_r(\boldsymbol{x}, t_{i-1}), \ldots, u_r(\boldsymbol{x}, t_{i-p})) + O(\Delta t^{p+1}) \qquad (p\text{-point formula}). \tag{64}$$

With a p-point backward difference approximation of the tangent space projection available at t_i we easily obtain an approximation of the normal component of $G(u_r)$ at t_i

$$N_{u_r(\boldsymbol{x},t_i)}G(u_r(\boldsymbol{x}, t_i)) = G(u_r(\boldsymbol{x}, t_i)) - \widetilde{P}^{(p)}_{u_r(\boldsymbol{x},t_i)}G(u_r(\boldsymbol{x}, t_i)) + O(\Delta t^{p+1}), \tag{65}$$

which allows us to implement the criterion (60) for rank increase at time t_i. Clearly, the p-point formula (64), and the corresponding approximation of the normal component (65), are effectively of order p in Δt if and only if the time snapshots $u_r(\boldsymbol{x}, t_i)$ are computed via a temporal integrator of order p. We emphasize that this method of using a finite difference stencil based on the temporal grid for approximating the tangential component of the dynamics (and thus the normal component) creates a lower bound for the choice of normal vector threshold ϵ_{inc}. In particular, we must have that $K_1(\Delta t)^p \geq \epsilon_{\text{inc}}$ for some constant K_1 otherwise the error incurred from our approximation of the normal component may trigger unnecessary mode addition. This approximation of the normal component is cheap but only informs on whether or not it is appropriate to add modes at time instant t_i.

The subsequent question is which entries of the rank vector $\boldsymbol{r}$ need to be increased. In order to make such a determination we expand the approximate solution at time t as

$$\begin{aligned} u_{\boldsymbol{r}}(\boldsymbol{x},t) = \boldsymbol{\Psi}_1(t)\cdots\boldsymbol{\Psi}_d(t)+\boldsymbol{\Gamma}_1(t)\cdots\boldsymbol{\Gamma}_d(t), \qquad \boldsymbol{\Psi}_i\in M_{r_{i-1}\times r_i}(L^2_{\mu_i}(\Omega_i)),\\ \boldsymbol{\Gamma}_i \in M_{f_{i-1}\times f_i}(L^2_{\mu_i}(\Omega_i)), \end{aligned} \tag{66}$$

where $\boldsymbol{\Gamma}_1(t)\cdots\boldsymbol{\Gamma}_d(t) = 0$ for all $t\in[0,T]$. Differentiating (66) with respect to time yields

$$\frac{\partial u_{\boldsymbol{r}}(\boldsymbol{x},t)}{\partial t} = \frac{\partial}{\partial t}\left[\boldsymbol{\Psi}_1(t)\cdots\boldsymbol{\Psi}_d(t)\right] + \frac{\partial}{\partial t}\left[\boldsymbol{\Gamma}_1(t)\cdots\boldsymbol{\Gamma}_d(t)\right]. \tag{67}$$

Subtracting off the tangential component (28) we have the normal component at time t

$$N_{u_{\boldsymbol{r}}(\boldsymbol{x},t)}\frac{\partial u_{\boldsymbol{r}}(\boldsymbol{x},t)}{\partial t} = \frac{\partial}{\partial t}\left[\boldsymbol{\Gamma}_1(t)\cdots\boldsymbol{\Gamma}_d(t)\right]. \tag{68}$$

Next, orthogonalize the partial product $\boldsymbol{\Gamma}_{\leq i-1}(t)$ from the left and the partial product $\boldsymbol{\Gamma}_{\geq i}(t)$ from the right to obtain

$$N_{u_{\boldsymbol{r}}(\boldsymbol{x},t)}\frac{\partial u_{\boldsymbol{r}}(\boldsymbol{x},t)}{\partial t} = \frac{\partial}{\partial t}\left[\boldsymbol{\Gamma}_1(t)\cdots\boldsymbol{\Gamma}_{i-1}(t)\boldsymbol{C}_i(t)\boldsymbol{\Gamma}_i^{\mathrm{T}}(t)\cdots\boldsymbol{\Gamma}_d^{\mathrm{T}}(t)\right], \tag{69}$$

where $\boldsymbol{C}_i = \boldsymbol{0}_{r_{i-1}\times r_i}$ and $\left\langle\boldsymbol{\Gamma}_i^{\mathrm{T}}\boldsymbol{\Gamma}_i\right\rangle_i = \boldsymbol{I}$ for all $i=1,2,\ldots,d$. Expand (69) using a product rule and evaluate at $t=t_i$

$$\left[N_{u_{\boldsymbol{r}}(\boldsymbol{x},t)}\frac{\partial u_{\boldsymbol{r}}(\boldsymbol{x},t)}{\partial t}\right]_{t=t_i} = \boldsymbol{\Gamma}_1(t_i)\cdots\boldsymbol{\Gamma}_{j-1}(t_i)\left.\frac{\partial \boldsymbol{C}_j(t)}{\partial t}\right|_{t=t_i}\boldsymbol{\Gamma}_j(t_i)\cdots\boldsymbol{\Gamma}_d(t_i). \tag{70}$$

From the previous equation we see that the FTT autocorrelation matrices of the normal component at time instant t_i are the time derivatives of the zero energy modes in the current solution. Thus, if the normal component has FTT rank $\boldsymbol{n}$ then the solution $u_{\boldsymbol{r}}(\boldsymbol{x},t)$ at time t_i should be represented by an FTT tensor of rank $\boldsymbol{r}+\boldsymbol{n}$. Certainly, the solution will be over represented at t_i with rank $\boldsymbol{r}+\boldsymbol{n}$. However, after one step of the splitting integrator the additional ranks will ensure that the low-rank solution $u_{\boldsymbol{r}+\boldsymbol{n}}(\boldsymbol{x},t)\in\mathcal{M}_{\boldsymbol{r}+\boldsymbol{n}}$ retains its accuracy.

The main steps of the algorithm we propose to adaptively increase the tensor rank are summarized in Algorithm 1. The operation "$*$" appearing within the conditional statement if/end denotes scalar times FTT tensor, and is meant to indicate that the multiplication is done by scaling the first core of the tensor with the scalar 0 and leaving the remainder of the cores unchanged [44]. As we will demonstrate in Sect. 5, Algorithm 1 is robust and it yields accurate results that do no require ad-hoc approximations such the matrix pseudo-inverse approximation introduced in [1].

4.3 Order of the Rank-Adaptive Tensor Scheme

Let us choose the threshold ϵ_{inc} in (60) to satisfy

$$\epsilon_{\mathrm{inc}} \leq K_2\Delta t, \tag{71}$$

and assume that the condition

$$\left|N_{u_{\boldsymbol{r}}(\boldsymbol{x},t)}G(u_{\boldsymbol{r}}(\boldsymbol{x},t))\right| \leq \epsilon_{\mathrm{inc}} \tag{72}$$

Algorithm 1: One step integration with adaptive rank increase

```
Input:
    $u_{\boldsymbol{r}}(\boldsymbol{x},t_i), u_{\boldsymbol{r}}(\boldsymbol{x},t_{i-1}), \dots, u_{\boldsymbol{r}}(\boldsymbol{x},t_{i-p})$ → time snapshots of the PDE solution with rank $\boldsymbol{r}$,
    $G(u_{\boldsymbol{r}}(\boldsymbol{x},t_i))$ → velocity vector defined by the right hand side of the PDE (1) at time $t_i$,
    $\Delta t$ → time step,
    $\epsilon_{\text{inc}}$ → threshold for the norm of normal component $N_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)}G(u_{\boldsymbol{r}}(\boldsymbol{x},t_i))$.
Output: $u_{\boldsymbol{r}+\boldsymbol{n}}(\boldsymbol{x},t_{i+1})$ → PDE solution with rank $\boldsymbol{r}+\boldsymbol{n}$ at time $t_{i+1}$
Initialization:
  • Approximate the constant rank velocity vector via the BDF formula:
    $\widetilde{P}^{(p)}_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)}G(u_{\boldsymbol{r}}(\boldsymbol{x},t_i)) = BD_p(u_{\boldsymbol{r}}(\boldsymbol{x},t_i), u_{\boldsymbol{r}}(\boldsymbol{x},t_{i-1}), \dots, u_{\boldsymbol{r}}(\boldsymbol{x},t_{i-p}))$
  • Compute the normal component:
    $N_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)}G(u_{\boldsymbol{r}}(\boldsymbol{x},t_i)) = G(u_{\boldsymbol{r}}(\boldsymbol{x},t_i)) - \widetilde{P}^{(p)}_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)}G(u_{\boldsymbol{r}}(\boldsymbol{x},t_i))$
Runtime:
  • if $\|N_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)}G(u_{\boldsymbol{r}}(\boldsymbol{x},t_i))\| > \epsilon_{\text{inc}}$ then
      Compute the FTT decomposition of normal component:
      $N_{\text{TT}}(\boldsymbol{x},t_i) = \text{FTT}(N_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)}G(u_{\boldsymbol{r}}(\boldsymbol{x},t_i)))$
      Initialize to zero additional tensor modes in $u_{\boldsymbol{r}}(\boldsymbol{x},t_i)$, as many as the rank of $N_{\text{TT}}$ (say $\boldsymbol{n}$):
      $u_{\boldsymbol{r}+\boldsymbol{n}}(\boldsymbol{x},t_i) = u_{\boldsymbol{r}}(\boldsymbol{x},t_i) + 0 * N_{\text{TT}}(\boldsymbol{x},t_i)$
    end
  • Use one step of Lie–Trotter splitting integrator to map $u_{\boldsymbol{r}+\boldsymbol{n}}(\boldsymbol{x},t_i)$ into $u_{\boldsymbol{r}+\boldsymbol{n}}(\boldsymbol{x},t_{i+1})$
```

is satisfied for all $t \in [0, T]$. Then we have the following bound for the local truncation error

$$\begin{aligned} \|E(t_i, t_{i+1})\| &= \left\| \int_{t_i}^{t_{i+1}} N_{u_{\boldsymbol{r}}(\boldsymbol{x},\tau)}G(u_{\boldsymbol{r}}(\boldsymbol{x},\tau))d\tau \right\| \\ &\le \int_{t_i}^{t_{i+1}} \|N_{u_{\boldsymbol{r}}(\boldsymbol{x},\tau)}G(u_{\boldsymbol{r}}(\boldsymbol{x},\tau))\| d\tau \\ &\le \int_{t_i}^{t_{i+1}} K_2 \Delta t d\tau \\ &= K_2 \Delta t^2. \end{aligned} \tag{73}$$

In particular, we have that the continuous-time rank-adaptive scheme is order one consistent in Δt if the normal vector threshold is set as in (71).

When implementing the adaptive scheme we usually discretize the time domain $[0, T]$ into a mesh of time instants as in (39). Therefore, we do not necessarily have control over the normal vector for all $t \in [0, T]$ but rather only at a finite number of time instants. However, an analogous argument as we have made for order one consistency in the continuous time rank-adaptive scheme holds for the discrete time rank-adaptive scheme by considering the first-order approximation of the local truncation error given in (58). In particular by using the equality in (58) and discrete time thresholding of the normal component

$$\|N_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)}G(u_{\boldsymbol{r}}(\boldsymbol{x},t_i))\| \le \epsilon_{\text{inc}}, \qquad \forall i = 0, 1, \dots, N, \tag{74}$$

we have that

$$\begin{aligned}\|E(t_i, t_{i+1})\| &= \left\|\Delta t N_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)} G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_i)) + O(\Delta t^2)\right\| \\ &\leq \left\|\Delta t N_{u_{\boldsymbol{r}}(\boldsymbol{x},t_i)} G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_i))\right\| + \|O(\Delta t^2)\| \\ &= K_2 \Delta t^2 + O(\Delta t^2) \\ &= O(\Delta t^2).\end{aligned} \tag{75}$$

This proves that the discrete time rank-adaptive scheme with normal threshold given by (74) is consistent with order one in Δt. Higher-order consistency results can be obtained with higher-order time integration methods and higher-order estimators for the normal vector $N_{u_r} G(u_{\boldsymbol{r}})$.

5 Numerical Examples

In this section we demonstrate the proposed rank-adaptive FTT tensor method on linear and nonlinear PDEs. In all examples the rank-adaptive scheme relies on first-order Lie–Trotter operator splitting time integration (42), and the thresholding criterion (60). For each PDE we rigorously assess the accuracy of the proposed rank-adaptive tensor method by comparing it with benchmark solutions computed with well-established numerical methods.

5.1 Two-Dimensional Variable Coefficient Advection Equation

Let us begin with the two-dimensional variable coefficient advection problem

$$\begin{cases} \dfrac{\partial u(x_1, x_2, t)}{\partial t} = (\sin(x_1) + \cos(x_2)) \dfrac{\partial u(x_1, x_2, t)}{\partial x_1} + \cos(x_2) \dfrac{\partial u(x_1, x_2, t)}{\partial x_2}, \\ u(x_1, x_2, 0) = \exp[\sin(x_1 + x_2)], \end{cases} \tag{76}$$

on the flat torus $\Omega = \mathbb{T}^2$. We have shown in previous work [16] that the tensor solution to the PDE (76) increases in rank as time increases.

As is well known, the PDE (76) can be reduced to the trivial ODE $du/dt = 0$ along the flow generated by the dynamical system (see, e.g., [48])

$$\begin{cases} \dfrac{dx_1}{dt} = \sin(x_1) + \cos(x_2), \\ \dfrac{dx_2}{dt} = \cos(x_2). \end{cases} \tag{77}$$

With the flow $\{x_1(t, x_{01}, x_{02}), x_1(t, x_{01}, x_{02})\}$ available, we can write the analytical solution to (76) as

$$u_{\text{ref}}(x_1, x_2, t) = \exp\left[\sin(x_{01}(x_1, x_2, t) + x_{02}(x_1, x_2, t))\right], \tag{78}$$

where $\{x_{01}(x_1, x_2, t), x_{02}(x_1, x_2, t)\}$ denotes the inverse flow generated by (77). We obtain a semi-analytical solution to the PDE (76) by solving the characteristic system (77) numerically for different initial conditions and then evaluating (78). A few time snapshots of the semi-analytical solution (78) are plotted in Fig. 3 (middle row).

We also solve the PDE (76) using the proposed rank-adaptive tensor method with first-order Lie–Trotter operator splitting and thresholding criterion (60). The initial condition is

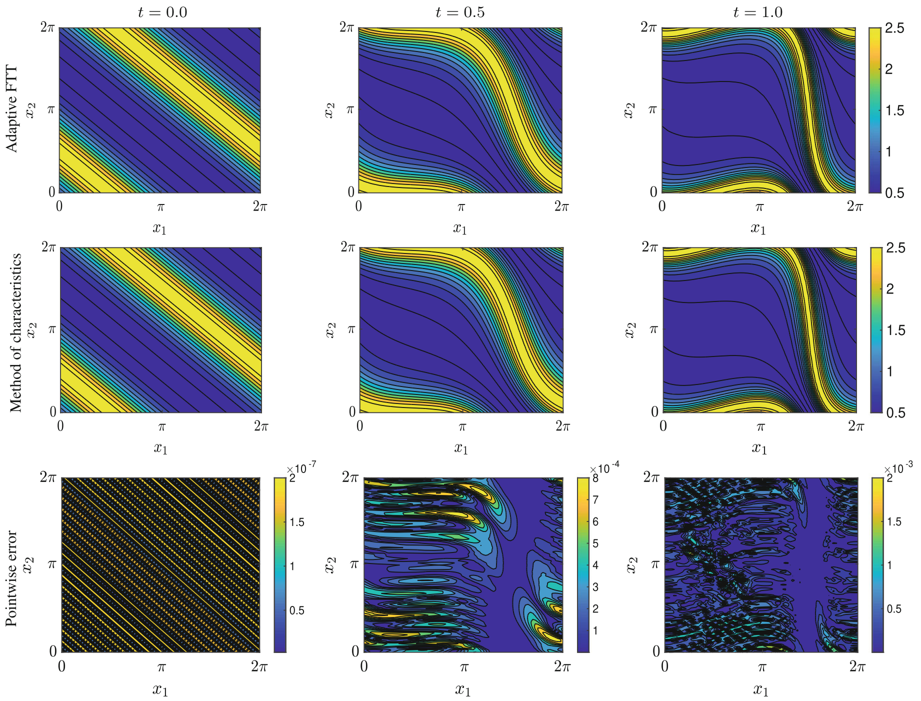

Fig. 3 Variable coefficient advection equation (76). Time snapshots of the rank-adaptive FTT solution $u_{\boldsymbol{r}}(x_1, x_2, t)$ obtained with threshold $\epsilon_{\text{inc}} = 10^{-2}$ (top), the semi-analytical solution $u_{\text{ref}}(x_1, x_2, t)$ (middle), and the pointwise error between the two solutions (bottom)

approximated by an FTT tensor $u_{\boldsymbol{r}}(x_1, x_2, 0)$ with multivariate rank $\boldsymbol{r} = \begin{bmatrix}1 & 15 & 1\end{bmatrix}$

$$u_{\boldsymbol{r}}(x_1, x_2, 0) = \boldsymbol{\Psi}_1(x_1)\sqrt{\boldsymbol{\Lambda}}\boldsymbol{\Psi}_2(x_2), \tag{79}$$

where

$$\boldsymbol{\Psi}_1(x_1) = \begin{bmatrix}\psi_1(1; x_1; 1) & \cdots & \psi_1(1; x_1; 15)\end{bmatrix}, \quad \sqrt{\boldsymbol{\Lambda}} = \begin{bmatrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_{15}\end{bmatrix},$$
$$\boldsymbol{\Psi}_2(x_2) = \begin{bmatrix}\psi_2(1; x_1; 1) \\ \vdots \\ \psi_2(15; x_2; 1)\end{bmatrix}. \tag{80}$$

Each tensor mode ψ_i is discretized on a grid of 81 evenly-spaced points in the interval $\Omega_i = [0, 2\pi]$. One-dimensional Fourier pseudo-spectral quadrature rules and differentiation matrices [29] are used to compute inner products and derivatives when needed. We run three simulations with the initial tensor decomposition (79) and time step $\Delta t = 10^{-4}$. In the first simulation we do not use any rank adaptation, in the second simulation we set the normal vector threshold to $\epsilon_{\text{inc}} = 10^{-1}$ and in the third simulation we set $\epsilon_{\text{inc}} = 10^{-2}$. At each time step the component of $G(u_{\boldsymbol{r}}(\boldsymbol{x}, t_i))$ normal to the tensor manifold is approximated with the two-point BDF formula (Sect. 4.2). In Fig. 5 we plot a few time snapshots of the singular values of the rank-adaptive FTT solution with $\epsilon_{\text{inc}} = 10^{-2}$.

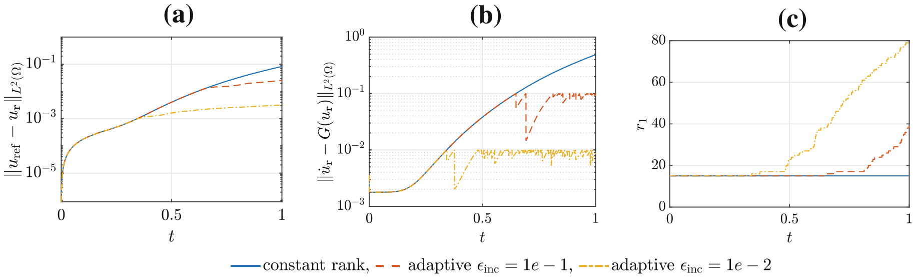

Fig. 4 **a** Global $L^2(\Omega)$ error of the FTT solution $u_{\boldsymbol{r}}$ relative to the benchmark solution u_{ref}; **b** Norm of the two-point BDF approximation to the normal component $N_{u_{\boldsymbol{r}}} G(u_{\boldsymbol{r}}(\boldsymbol{x}, t))$ (note the effect of thresholding); **c** tensor rank versus time of the constant-rank FTT solution and adaptive rank solutions with $\epsilon_{\text{inc}} = 10^{-1}$ and $\epsilon_{\text{inc}} = 10^{-2}$

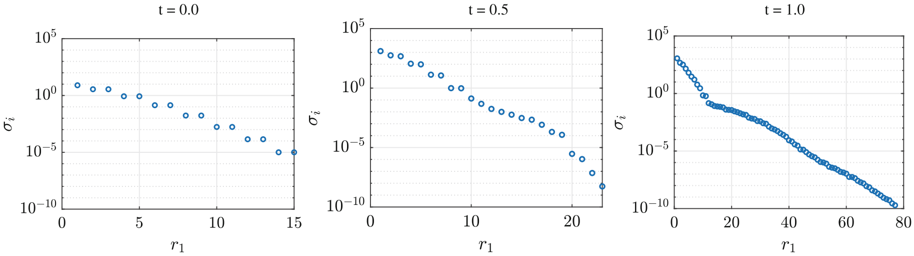

Fig. 5 Time snapshots of the singular values of the rank-adaptive FTT solution with threshold $\epsilon_{\text{inc}} = 10^{-2}$

Figure 4a–c summarize the performance and accuracy of the proposed rank-adaptive FTT solver. In particular, in Fig. 4a we plot the time-dependent $L^2(\Omega)$ error between the rank-adaptive FTT solution and the reference solution we obtained with method of characteristics. It is seen that decreasing the threshold ϵ_{inc} on the norm of the component of $G(u_{\boldsymbol{r}})$ normal to the FTT tensor manifold (Fig. 4b) yields addition of more tensor mores to the FTT solution (Fig. 4c). This, in turn, results in better accuracy as demonstrated in Fig. 4a.

5.2 Two-Dimensional Kuramoto–Sivashinsky Equation

In this section we demonstrate the rank-adaptive FTT integrator on the two-dimensional Kuramoto–Sivashinsky equation [30]

$$\begin{cases} \dfrac{\partial}{\partial t} u(x_1, x_2, t) + \dfrac{1}{2} |\nabla_\nu u(x_1, x_2, t)|^2 + \Delta_\nu u(x_1, x_2, t) + \nu_1 \Delta_\nu^2 u(x_1, x_2, t) = 0, \\ u(x_1, x_2, 0) = \sin(x_1 + x_2) + \sin(x_1) + \sin(x_2), \end{cases} \tag{81}$$

where

$$\nabla_\nu = \left(\frac{\partial}{\partial x_1}, \frac{\nu_2}{\nu_1} \frac{\partial}{\partial x_2} \right), \qquad \Delta_\nu = \frac{\partial^2}{\partial x_1^2} + \frac{\nu_2}{\nu_1} \frac{\partial^2}{\partial x_2^2}. \tag{82}$$

Here, ν_1, ν_2 are bifurcation parameters. For our demonstration we set $\nu_1 = 0.25$, $\nu_2 = 0.04$ and solve (81) on the two-dimensional flat torus $\mathbb{T}^2$. The initial condition can be written as rank $\boldsymbol{r} = \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$ FTT tensor

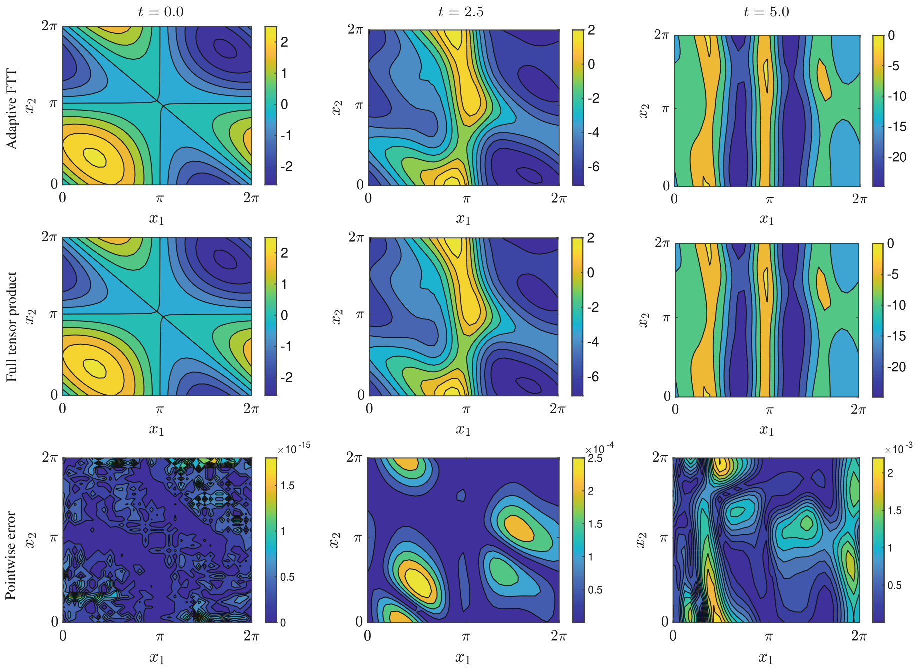

Fig. 6 Kuramoto–Sivashinsky equation (81). Time snapshots of the rank-adaptive FTT solution $u_{\boldsymbol{r}}(x_1, x_2, t)$ obtained with threshold $\epsilon_{\text{inc}} = 10^{-2}$ (top), the Fourier pseudo-spectral solution $u_{\text{ref}}(x_1, x_2, t)$ (middle), and the pointwise error between the two solutions (bottom)

$$u_0(x_1, x_2) = \psi_1(1; x_1; 1)\psi_2(1; x_2; 1)\sqrt{\lambda(1)} + \psi_1(1; x_1; 2)\psi_2(2; x_2; 1)\sqrt{\lambda(2)}, \quad (83)$$

where

$$\begin{aligned} \psi_1(1; x_1; 1) &= \frac{\sin(x_1)}{\sqrt{\pi}}, & \psi_1(1; x_1; 2) &= \frac{\cos(x_1) + 1}{\sqrt{3\pi}}, \\ \psi_2(1; x_2; 1) &= \frac{\cos(x_2) + 1}{\sqrt{3\pi}}, & \psi_2(2; x_2; 1) &= \frac{\sin(x_2)}{\sqrt{\pi}}, \end{aligned} \quad (84)$$

and

$$\sqrt{\lambda(1)} = \sqrt{\lambda(2)} = \sqrt{3}\pi. \quad (85)$$

We compute a benchmark solution by using a Fourier pseudo-spectral method [29] with 33 evenly-spaced grid points per spatial dimension (1089 total number of points). Derivatives and integrals are approximated with well-known pseudo-spectral differentiation matrices and Gauss quadrature rules. The resulting ODE system is integrated forward in time using an explicit fourth-order Runge–Kutta method with time step $\Delta t = 10^{-5}$.

As before, we performed multiple simulations using the proposed rank-adaptive FTT algorithm with different thresholds for the component of $G(u_{\boldsymbol{r}})$ normal to the tensor manifold. Specifically, we ran one simulation with no mode addition and three simulations with adaptive mode addition based on Algorithm 1, and thresholds set to $\epsilon_{\text{inc}} = 10$, $\epsilon_{\text{inc}} = 10^{-1}$, and $\epsilon_{\text{inc}} = 10^{-2}$. We used the two-point BDF formula (62) to approximate the component of the solution normal to the tensor manifold at each time step and the Lie–Trotter operator splitting

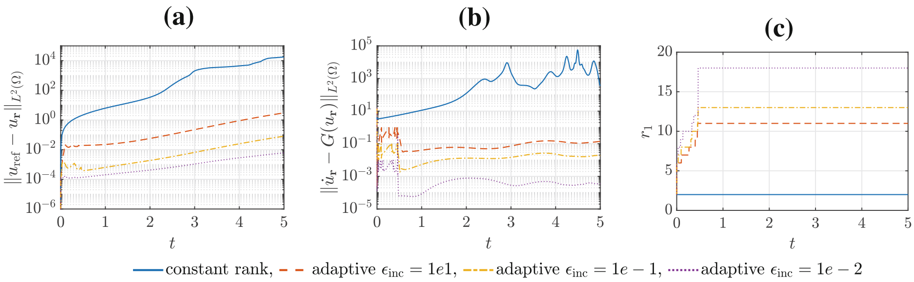

Fig. 7 **a** Global $L^2(\Omega)$ error between the FTT solution $u_{\boldsymbol{r}}$ to equation (81) and the benchmark solution $u_{\rm ref}$. **b** Norm of the approximation to $N_{u_{\boldsymbol{r}}} G(u_{\boldsymbol{r}}) = \dot{u}_{\boldsymbol{r}} - G(u_{\boldsymbol{r}})$ where the tangent space projection is computed with a two-point BDF formula at each time. **c** Rank versus time of the constant rank FTT solution and rank-adaptive FTT solutions with $\epsilon_{\rm inc} = 10, 10^{-1}, 10^{-2}$

scheme (42) with time step $\Delta t = 10^{-5}$ to integrate in time the rank-adaptive FTT solution. In Fig. 6 we compare the time snapshots of the rank-adaptive FTT solution with $\epsilon_{\rm inc} = 10^{-2}$ with the benchmark solution obtained by the Fourier pseudo-spectral method. As before, Fig. 7a–c demonstrate that the rank-adaptive FTT algorithm is effective in controlling the $L^2(\Omega)$ error of the FTT solution. Interestingly, the solution to the PDE (81) has the property that any tensor approximation with sufficient rank yields a normal component that does not grow in time. In fact, as seen in Fig. 7b the tensor rank becomes constant for each threshold $\epsilon_{\rm inc}$ after a transient of approximately 0.5 dimensionless time units.

In Fig. 7 we observe that the error associated with the constant rank 2 FTT solution increases significantly during temporal integration. This suggests that projecting the nonlinear Kuramoto–Sivashinsky equation (81) onto a rank 2 FTT manifold yields a reduced-order PDE which does not accurately capture the dynamics of the full system. A similar phenomenon occurs in other areas of reduced-order modeling, e.g., when projecting nonlinear PDEs onto proper orthogonal decomposition (POD) bases [54].

5.3 Four-Dimensional Fokker–Planck Equation

Finally, we demonstrate the proposed rank-adaptive FTT integrator on a four-dimensional Fokker–Planck equation with non-constant drift and diffusion coefficients. As is well known [49], the Fokker–Planck equation describes the evolution of the probability density function (PDF) of the state vector solving the Itô stochastic differential equation (SDE)

$$d\boldsymbol{X}_t = \boldsymbol{\mu}(\boldsymbol{X}_t, t)dt + \boldsymbol{\sigma}(\boldsymbol{X}_t, t)d\boldsymbol{W}_t. \tag{86}$$

Here, $\boldsymbol{X}_t$ is the d-dimensional state vector, $\boldsymbol{\mu}(\boldsymbol{X}_t, t)$ is the d-dimensional drift, $\boldsymbol{\sigma}(\boldsymbol{X}_t, t)$ is an $d \times m$ matrix and $\boldsymbol{W}_t$ is an m-dimensional standard Wiener process. The Fokker–Planck equation that corresponds to (86) has the form

$$\begin{cases} \dfrac{\partial p(\boldsymbol{x}, t)}{\partial t} = \mathcal{L}(\boldsymbol{x}, t)p(\boldsymbol{x}, t), \\ p(\boldsymbol{x}, 0) = p_0(\boldsymbol{x}), \end{cases} \tag{87}$$

where $p_0(\boldsymbol{x})$ is the PDF of the initial state $\boldsymbol{X}_0$, $\mathcal{L}$ is a second-order linear differential operator defined as

$$\mathcal{L}(\boldsymbol{x},t)p(\boldsymbol{x},t) = -\sum_{k=1}^{d} \frac{\partial}{\partial x_k}\left(\mu_k(x,t)p(\boldsymbol{x},t)\right) + \sum_{k,j=1}^{d} \frac{\partial^2}{\partial x_k \partial x_j}\left(D_{ij}(\boldsymbol{x},t)p(\boldsymbol{x},t)\right), \tag{88}$$

and $\boldsymbol{D}(\boldsymbol{x},t) = \boldsymbol{\sigma}(\boldsymbol{x},t)\boldsymbol{\sigma}(\boldsymbol{x},t)^{\mathrm{T}}/2$ is the diffusion tensor. For our numerical demonstration we set

$$\boldsymbol{\mu}(\boldsymbol{x}) = \alpha \begin{bmatrix} \sin(x_1) \\ \sin(x_3) \\ \sin(x_4) \\ \sin(x_1) \end{bmatrix}, \qquad \boldsymbol{\sigma}(\boldsymbol{x}) = \sqrt{2\beta} \begin{bmatrix} g(x_2) & 0 & 0 & 0 \\ 0 & g(x_3) & 0 & 0 \\ 0 & 0 & g(x_4) & 0 \\ 0 & 0 & 0 & g(x_1) \end{bmatrix}, \tag{89}$$

where $g(x) = \sqrt{1 + k\sin(x)}$. With the drift and diffusion matrices chosen in (89) the operator (88) takes the form

$$\begin{aligned} \mathcal{L} = &-\alpha\left(\cos(x_1) + \sin(x_1)\frac{\partial}{\partial x_1} + \sin(x_3)\frac{\partial}{\partial x_2} + \sin(x_4)\frac{\partial}{\partial x_3} + \sin(x_1)\frac{\partial}{\partial x_4}\right) \\ &+ \beta\left((1 + k\sin(x_2))\frac{\partial^2}{\partial x_1^2} + (1 + k\sin(x_3))\frac{\partial^2}{\partial x_2^2} + (1 + k\sin(x_4))\frac{\partial^2}{\partial x_3^2} + (1 + k\sin(x_1))\frac{\partial^2}{\partial x_4^2}\right). \end{aligned} \tag{90}$$

Clearly $\mathcal{L}$ is a linear, time-independent separable operator of rank 9, since it can be written as

$$\mathcal{L} = \sum_{i=1}^{9} L_i^{(1)} \otimes L_i^{(2)} \otimes L_i^{(3)} \otimes L_i^{(4)}, \tag{91}$$

where each $L_i^{(j)}$ operates on x_j only. Specifically, we have

$$\begin{aligned} &L_1^{(1)} = -\alpha\cos(x_1), && L_2^{(1)} = -\alpha\sin(x_1)\frac{\partial}{\partial x_1}, && L_3^{(2)} = -\alpha\frac{\partial}{\partial x_2}, && L_3^{(3)} = \sin(x_3), \\ &L_4^{(3)} = -\alpha\frac{\partial}{\partial x_3}, && L_4^{(4)} = \sin(x_4), && L_5^{(1)} = -\alpha\sin(x_1), && L_5^{(4)} = \frac{\partial}{\partial x_4}, \\ &L_6^{(1)} = \beta\frac{\partial^2}{\partial x_1^2}, && L_6^{(2)} = 1 + k\sin(x_2), && L_7^{(2)} = \beta\frac{\partial^2}{\partial x_2^2}, && L_7^{(3)} = 1 + k\sin(x_3), \\ &L_8^{(3)} = \beta\frac{\partial^2}{\partial x_3^2}, && L_8^{(2)} = 1 + k\sin(x_4), && L_9^{(4)} = \beta\frac{\partial^2}{\partial x_4^2}, && L_9^{(1)} = 1 + k\sin(x_1), \end{aligned} \tag{92}$$

and all other unspecified $L_i^{(j)}$ are identity operators. We set the parameters in (89) as $\alpha = 0.1$, $\beta = 2.0$, $k = 1.0$ and solve (87) on the four-dimensional flat torus $\mathbb{T}^4$. The initial PDF is set as

$$p_0(\boldsymbol{x}) = \frac{\sin(x_1)\sin(x_2)\sin(x_3)\sin(x_4) + 1}{16\pi^4}. \tag{93}$$

Note that (93) is a four-dimensional FTT tensor with multilinear rank $\boldsymbol{r} = \begin{bmatrix}1 & 2 & 2 & 2 & 1\end{bmatrix}$. Upon normalizing the modes appropriately we obtain the left orthogonalized initial condition required to begin integration

$$\begin{aligned} p_0(\boldsymbol{x}) = &\ \psi_1(1;x_1;1)\psi_2(1;x_2;1)\psi_3(1;x_3;1)\psi_4(1;x_4;1)\sqrt{\lambda(1)} \\ &+ \psi_1(1;x_1;2)\psi_2(2;x_2;2)\psi_3(2;x_3;2)\psi_4(2;x_4;1)\sqrt{\lambda(2)}, \end{aligned} \tag{94}$$

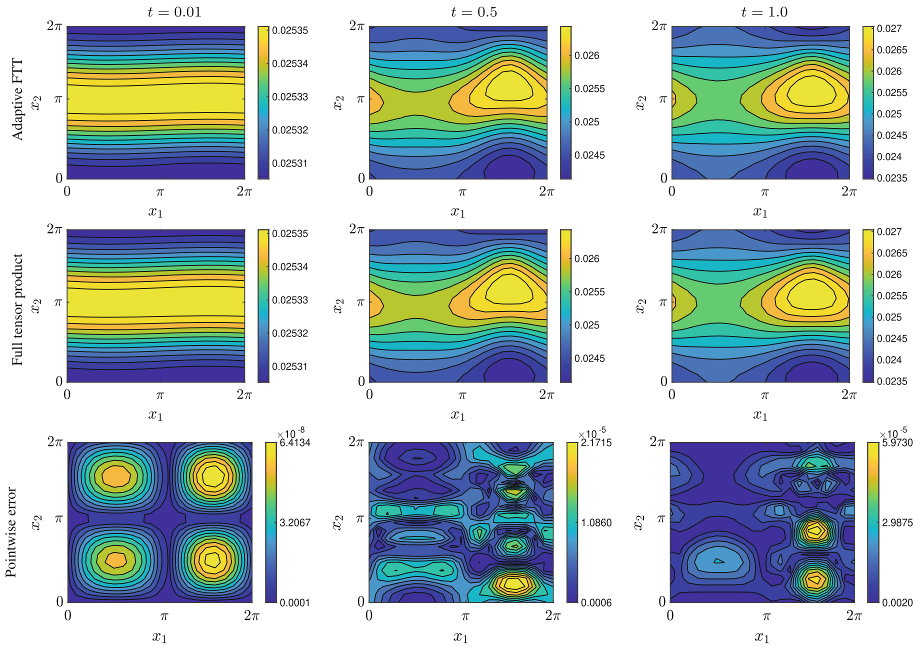

Fig. 8 Time snapshots of marginal PDF $p_{\boldsymbol{r}}(x_1, x_2, t)$ corresponding to the solution to the Fokker–Planck equation (87). We plot marginals computed with the rank-adaptive FTT integrator using $\epsilon_{\text{inc}} = 10^{-4}$ (top row) and with the full tensor product Fourier pseudo-spectral method (middle row). We also plot the pointwise error between the two numerical solutions (bottom row). The initial condition is the FTT tensor (93)

where

$$\psi_i(1; x_i; 1) = \frac{\sin(x_i)}{\sqrt{\pi}}, \qquad \sqrt{\lambda(1)} = \frac{1}{16\pi^2}. \tag{95}$$

All other tensor modes are equal to $1/\sqrt{2\pi}$, and $\sqrt{\lambda(2)} = 1/(2\pi^2)$. To obtain a benchmark solution with which to compare the rank-adaptive FTT solution, we solve the PDE (87) using a Fourier pseudo-spectral method on the flat torus $\mathbb{T}^4$ with $21^4 = 194481$ evenly-spaced points. As before, the operator $\mathcal{L}$ is represented in terms of pseudo-spectral differentiation matrices [29], and the resulting semi-discrete approximation (ODE system) is integrated with an explicit fourth-order Runge Kutta method using time step $\Delta t = 10^{-4}$. The numerical solution we obtained in this way is denoted by $p_{\text{ref}}(\boldsymbol{x}, t)$. We also solve the Fokker–Planck using the proposed rank-adaptive FTT method with first-order Lie–Trotter time integrator (Sect. 3.1.1) and normal vector thresholding (Sect. 4.2). We run three simulations all with time step $\Delta t = 10^{-4}$: one with no rank adaption, and two with rank-adaptation and normal component thresholds set to $\epsilon_{\text{inc}} = 10^{-3}$ and $\epsilon_{\text{inc}} = 10^{-4}$. In Fig. 8 we plot three time snapshots of the two-dimensional solution marginal

$$p(x_1, x_2, t) = \int_0^{2\pi} \int_0^{2\pi} p(x_1, x_2, x_3, x_4, t) dx_3 dx_4 \tag{96}$$

computed with the rank-adaptive FTT integrator ($\epsilon_{\text{inc}} = 10^{-4}$) and the full tensor product pseudo-spectral method (reference solution). In Fig. 9a we compare the $L^2(\Omega)$ errors of the rank-adaptive method relative to the reference solution. It is seen that as we decrease

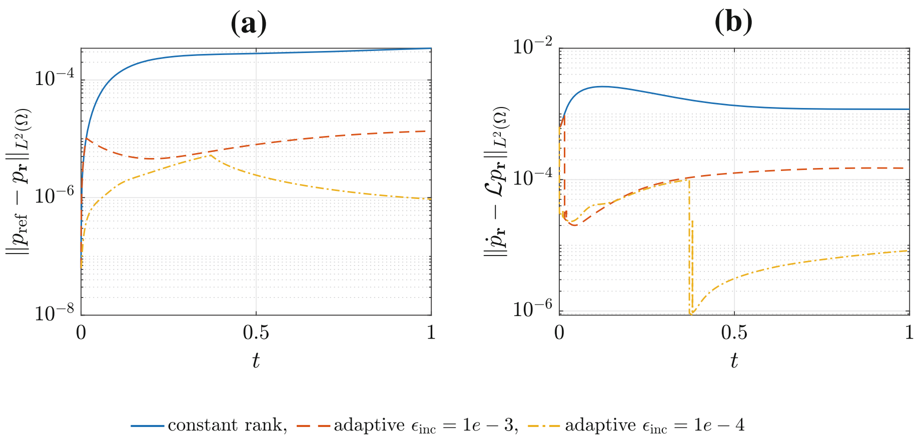

Fig. 9 **a** The $L^2(\Omega)$ error of the FTT solution $p_{\boldsymbol{r}}(\boldsymbol{x}, t)$ relative to the benchmark solution $p_{\mathrm{ref}}(\boldsymbol{x}, t)$ computed with a Fourier pseudo-spectral method on a tensor product grid. **b** Norm of the component of $\mathcal{L}p_{\boldsymbol{r}}$ normal to the tensor manifold (see Fig. 2). Such component is approximated with a two-point BDF formula at each time step

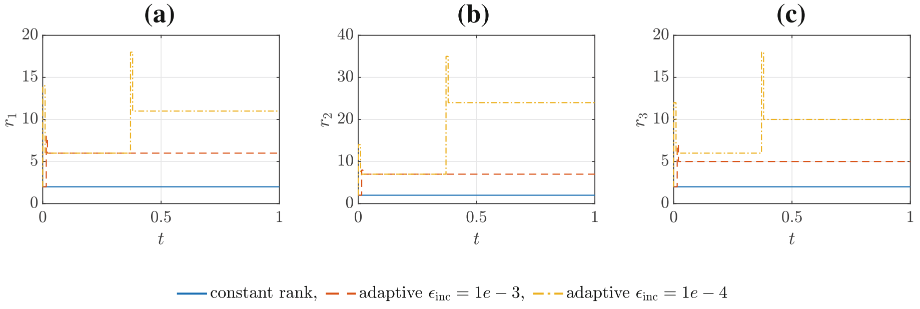

Fig. 10 Tensor rank $\boldsymbol{r} = [1\, r_1\, r_2\, r_3\, 1]$ of the adaptive FTT solution to the four dimensional Fokker–Planck equation (87)

the threshold the solution becomes more accurate. In Fig. 9b we plot the component of $\mathcal{L}p_{\boldsymbol{r}}$ normal to the tensor manifold, which is approximated using the two-point BDF formula (62). Note that in the rank-adaptive FTT solution with thresholds $\epsilon_{\mathrm{inc}} = 10^{-3}$ and $\epsilon_{\mathrm{inc}} = 10^{-4}$ the solver performs both mode addition as well as mode removal. This is documented in Fig. 10. The abrupt change in rank observed in Fig. 10a–c near time $t = 0.4$ corresponding to the rank-adaptive solution with threshold $\epsilon_{\mathrm{inc}} = 10^{-4}$ is due to the time step size Δt being equal to ϵ_{inc}. This can be justified as follows. Recall that the solution is first order accurate in Δt and therefore the approximation of the component of $\mathcal{L}p_{\boldsymbol{r}}$ normal to the tensor manifold $\mathcal{M}_{\boldsymbol{r}}$ is first-order accurate in Δt. If we set $\epsilon_{\mathrm{inc}} \leq \Delta t$, then the rank-adaptive scheme may overestimate the number of modes needed to achieve accuracy on the order of Δt. This does not affect the accuracy of the numerical solution due to the robustness of the Lie–Trotter integrator to over-approximation [38]. Moreover we notice that the rank-adaptive scheme removes the unnecessary modes ensure that the tensor rank is not unnecessarily large (see Sect. 4.1). In fact, the diffusive nature of the Fokker–Plank equation on the flat torus $\mathbb{T}^4$ yields relaxation to a statistical equilibrium state that depends on the drift and diffusion coefficients in (87). In this case such an equilibrium state is well-approximated by a low-rank FTT tensor.

6 Summary

We presented a new rank-adaptive tensor method to integrate high-dimensional nonlinear PDEs. The new method is based on functional tensor train (FTT) expansions [6,17,44], operator splitting time integration [32,39], and a new rank-adaptive algorithm to add and remove tensor modes from the PDE solution based on thresholding the component of the velocity vector normal to the FTT tensor manifold. We tested the proposed new algorithm on three different initial/boundary value problems including a 2D variable-coefficient first-order linear PDE, a 2D Kuramoto–Sivashinsky equation, and a 4D Fokker–Planck equation. In all cases the adaptive FTT solution was compared to a benchmark numerical solution constructed with well-established numerical methods. The numerical results we obtained demonstrate that the proposed rank-adaptive tensor method is effective in controlling the temporal integration error, and outperforms known integration methods for multidimensional PDEs in terms of accuracy, robustness and computational cost. We also proved that the new method is consistent with recently proposed step-truncation algorithms [33,50,51] in the limit of small time steps.

Funding This research was supported by the U.S. Air Force Office of Scientific Research (AFOSR) Grant FA9550-20-1-0174 and by the U.S. Army Research Office (ARO) Grant W911NF-18-1-0309.

Data availability statement The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Code availability The code generated during the current study is available from the corresponding author on reasonable request.

Declaration

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Babaee, H., Choi, M., Sapsis, T.P., Karniadakis, G.E.: A robust bi-orthogonal/dynamically-orthogonal method using the covariance pseudo-inverse with application to stochastic flow problems. J. Comput. Phys. **344**, 303–319 (2017)
2. Bachmayr, M., Schneider, R., Uschmajew, A.: Tensor networks and hierarchical tensors for the solution of high-dimensional partial differential equations. Found. Comput. Math. **16**(6), 1423 (2016)
3. Baldeaux, J., Gnewuch, M.: Optimal randomized multilevel algorithms for infinite-dimensional integration on function spaces with ANOVA-type decomposition. SIAM J. Numer. Anal. **52**(3), 1128–1155 (2014)
4. Barthelmann, V., Novak, E., Ritter, K.: High dimensional polynomial interpolation on sparse grids. Adv. Comput. Mech. **12**, 273–288 (2000)

5. Beylkin, G., Mohlenkamp, M.J.: Numerical operator calculus in higher dimensions. Proc. Natl. Acad. Sci. USA **99**(16), 10246–10251 (2002)
6. Bigoni, D., Engsig-Karup, A.P., Marzouk, Y.M.: Spectral tensor-train decomposition. SIAM J. Sci. Comput. **38**(4), A2405–A2439 (2016)
7. Boelens, A.M.P., Venturi, D., Tartakovsky, D.M.: Parallel tensor methods for high-dimensional linear PDEs. J. Comput. Phys. **375**, 519–539 (2018)
8. Boelens, A.M.P., Venturi, D., Tartakovsky, D.M.: Tensor methods for the Boltzmann–BGK equation. J. Comput. Phys. **421**, 109744 (2020)
9. Bungartz, H.J., Griebel, M.: Sparse grids. Acta Numer. **13**, 147–269 (2004)
10. Cao, Y., Chen, Z., Gunzbuger, M.: ANOVA expansions and efficient sampling methods for parameter dependent nonlinear PDEs. Int. J. Numer. Anal. Model. **6**, 256–273 (2009)
11. Cercignani, C.: The Boltzmann Equation and Its Applications. Springer, Berlin (1988)
12. Chkifa, A., Cohen, A., Schwab, C.: High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. Found. Comput. Math. **14**, 601–633 (2014)
13. Cho, H., Venturi, D., Karniadakis, G.E.: Statistical analysis and simulation of random shocks in Burgers equation. Proc. R. Soc. A **2171**(470), 1–21 (2014)
14. Cho, H., Venturi, D., Karniadakis, G.E.: Numerical methods for high-dimensional probability density function equations. J. Comput. Phys. **315**, 817–837 (2016)
15. Al Daas, H., Ballard, G., Benner, P.: Parallel algorithms for tensor train arithmetic. arXiv:2011.06532, pp. 1–31 (2020)
16. Dektor, A., Venturi, D.: Dynamically orthogonal tensor methods for high-dimensional nonlinear PDEs. J. Comput. Phys. **404**, 109125 (2020)
17. Dektor, A., Venturi, D.: Dynamic tensor approximation of high-dimensional nonlinear PDEs. J. Comput. Phys. **437**, 110295 (2021)
18. di Marco, G., Pareschi, L.: Numerical methods for kinetic equations. Acta Numer. **23**, 369–520 (2014)
19. Han, W .E,J., Li, Q.: A mean-field optimal control formulation of deep learning. Res. Math. Sci. **6**(10), 1–41 (2019)
20. Falcó, A., Hackbusch, W., Nouy, A.: Geometric structures in tensor representations. arXiv:1505.03027, pp. 1–50 (2015)
21. Falcó, A., Hackbusch, W., Nouy, A.: On the Dirac–Frenkel variational principle on tensor Banach spaces. Found. Comput. Math. **19**(1), 159–204 (2019)
22. Falcó, A., Hackbusch, W., Nouy, A.: Geometry of tree-based tensor formats in tensor banach spaces. ArXiv **2011**(08466), 1–14 (2020)
23. Feppon, F., Lermusiaux, P.F.J.: A geometric approach to dynamical model order reduction. SIAM J. Matrix Anal. Appl. **39**(1), 510–538 (2018)
24. Feppon, F., Lermusiaux, P.F.J.: The extrinsic geometry of dynamical systems tracking nonlinear matrix projections. SIAM J. Matrix Anal. Appl. **40**(2), 814–844 (2019)
25. Foo, J., Karniadakis, G.E.: Multi-element probabilistic collocation method in high dimensions. J. Comput. Phys. **229**, 1536–1557 (2010)
26. Frenkel, J.: Wave Mechanics: Advanced General Theory. Oxford University Press, Oxford (1934)
27. Gangbo, W., Li, W., Osher, S., Puthawala, M.: Unnormalized optimal transport. J. Comput. Phys. **399**, 108940 (2019)
28. Hackbusch, W.: Tensor Spaces and Numerical Tensor Calculus. Springer, Berlin (2012)
29. Hesthaven, J.S., Gottlieb, S., Gottlieb, D.: Spectral methods for time-dependent problems, volume 21 of Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge (2007)
30. Kalogirou, A., Keaveny, E.E., Papageorgiou, D.T.: An in-depth numerical study of the two-dimensional Kuramoto–Sivashinsky equation. Proc. A. 471(2179):20140932 (2015)
31. Khoromskij, B.N.: Tensor numerical methods for multidimensional PDEs: theoretical analysis and initial applications. In: CEMRACS 2013—Modelling and Simulation of Complex Systems: Stochastic and Deterministic Approaches, volume 48 of ESAIM Proc. Surveys, pp. 1–28. EDP Sci., Les Ulis (2015)
32. Kieri, E., Lubich, C., Walach, H.: Discretized dynamical low-rank approximation in the presence of small singular values. SIAM J. Numer. Anal. **54**(2), 1020–1038 (2016)
33. Kieri, E., Vandereycken, B.: Projection methods for dynamical low-rank approximation of high-dimensional problems. Comput. Methods Appl. Math. **19**(1), 73–92 (2019)
34. Koch, O., Lubich, C.: Dynamical low-rank approximation. SIAM J. Matrix Anal. Appl. **29**(2), 434–454 (2007)
35. Koch, O., Lubich, C.: Dynamical tensor approximation. SIAM J. Matrix Anal. Appl. **31**(5), 2360–2375 (2010)

36. Koch, O., Neuhauser, C., Thalhammer, M.: Error analysis of high-order splitting methods for nonlinear evolutionary Schrödinger equations and application to the MCTDHF equations in electron dynamics. ESAIM Math. Model. Numer. Anal. **47**(5), 1265–1286 (2013)
37. Li, G., Rabitz, H.: Regularized random-sampling high dimensional model representation (RS-HDMR). J. Math. Chem. **43**(3), 1207–1232 (2008)
38. Lubich, C., Oseledets, I.V., Vandereycken, B.: Time integration of tensor trains. SIAM J. Numer. Anal. **53**(2), 917–941 (2015)
39. Lubich, C., Vandereycken, B., Walach, H.: Time integration of rank-constrained Tucker tensors. SIAM J. Numer. Anal. **56**(3), 1273–1290 (2018)
40. McLachlan, A.D.: A variational solution of the time-dependent Schrödinger equation. Mol. Phys. **8**, 39–44 (1964)
41. Narayan, A., Jakeman, J.: Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation. SIAM J. Sci. Comput. **36**(6), A2952–A2983 (2014)
42. Nashed, M.Z.: Differentiability and related properties of nonlinear operators: Some aspects of the role of differentials in nonlinear functional analysis. In: Nonlinear Functional Anal. and Appl. (Proc. Advanced Sem., Math. Res. Center, Univ. of Wisconsin, Madison, WI, 1970), pp. 103–309. Academic Press, New York (1971)
43. Nonnenmacher, A., Lubich, C.: Dynamical low-rank approximation: applications and numerical experiments. Math. Comput. Simul. **79**(4), 1346–1357 (2008)
44. Oseledets, I.V.: Tensor-train decomposition. SIAM J. Sci. Comput. **33**(5), 2295–2317 (2011)
45. Raab, A.: On the Dirac–Frenkel/Mclachlan variational principle. Chem. Phys. Lett. **319**, 674–678 (2000)
46. Raissi, M., Karniadakis, G.E.: Hidden physics models: machine learning of nonlinear partial differential equations. J. Comput. Phys. **357**, 125–141 (2018)
47. Raissi, M., Perdikaris, P., Karniadakis, G.E.: Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. **378**, 606–707 (2019)
48. Rhee, H.-K., Aris, R., Amundson, N.R.: First-Order Partial Differential Equations, Volume 1: Theory and Applications of Single Equations. Dover, New York (2001)
49. Risken, H.: The Fokker–Planck Equation: Methods of Solution and Applications, vol. 60, 2nd edn. Springer (1989)
50. Rodgers, A., Dektor, A., Venturi, D.: Adaptive integration of nonlinear evolution equations on tensor manifolds. arXiv:2008.00155:1–22 (2020)
51. Rodgers, A., Venturi, D.: Stability analysis of hierarchical tensor methods for time-dependent PDEs. J. Comput. Phys. (2020). https://doi.org/10.1016/j.jcp.2020.109341
52. Ruthotto, L., Osher, S., Li, W., Nurbekyan, L., Fung, S.W.: A machine learning framework for solving high-dimensional mean field game and mean field control problems. PNAS **117**(17), 9183–9193 (2020)
53. Salas, D., Thibault, L.: On characterizations of submanifolds via smoothness of the distance function in Hilbert spaces. J. Optim. Theory Appl. **182**(1), 189–210 (2019)
54. Sirisup, S., Karniadakis, G.E.: A spectral viscosity method for correcting the long-term behavior of pod models. J. Comput. Phys. **194**(1), 92–116 (2004)
55. Venturi, D.: The numerical approximation of nonlinear functionals and functional differential equations. Phys. Rep. **732**, 1–102 (2018)
56. Venturi, D., Dektor, A.: Spectral methods for nonlinear functionals and functional differential equations. Res. Math. Sci. **8**(27), 1–39 (2021)
57. Venturi, D., Karniadakis, G.E.: Convolutionless Nakajima–Zwanzig equations for stochastic analysis in nonlinear dynamical systems. Proc. R. Soc. A **470**(2166), 1–20 (2014)
58. Venturi, D., Sapsis, T.P., Cho, H., Karniadakis, G.E.: A computable evolution equation for the joint response-excitation probability density function of stochastic dynamical systems. Proc. R. Soc. A **468**(2139), 759–783 (2012)
59. Villani, C.: Optimal Transport: Old and New. Springer, Berlin (2009)
60. Zhu, Y., Zabaras, N., Koutsourelakis, P.-S., Perdikaris, P.: Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. J. Comput. Phys. **394**, 56–81 (2019)