



Implicit Integration of Nonlinear Evolution Equations on Tensor Manifolds

Abram Rodgers¹ · Daniele Venturi¹

Received: 18 March 2023 / Revised: 16 August 2023 / Accepted: 5 September 2023 /
Published online: 23 September 2023
© The Author(s) 2023

Abstract

Explicit step-truncation tensor methods have recently proven successful in integrating initial value problems for high-dimensional partial differential equations. However, the combination of non-linearity and stiffness may introduce time-step restrictions which could make explicit integration computationally infeasible. To overcome this problem, we develop a new class of implicit rank-adaptive algorithms for temporal integration of nonlinear evolution equations on tensor manifolds. These algorithms are based on performing one time step with a conventional time-stepping scheme, followed by an implicit fixed point iteration step involving a rank-adaptive truncation operation onto a tensor manifold. Implicit step truncation methods are straightforward to implement as they rely only on arithmetic operations between tensors, which can be performed by efficient and scalable parallel algorithms. Numerical applications demonstrating the effectiveness of implicit step-truncation tensor integrators are presented and discussed for the Allen–Cahn equation, the Fokker–Planck equation, and the nonlinear Schrödinger equation.

Keywords High-dimensional PDEs · Low-rank tensor manifolds · Dynamical tensor approximation · Tensor train · Hierarchical Tucker format

1 Introduction

High-dimensional nonlinear evolution equations of the form

$$\frac{\partial f(\mathbf{x}, t)}{\partial t} = \mathcal{N}(f(\mathbf{x}, t), \mathbf{x}), \quad f(\mathbf{x}, 0) = f_0(\mathbf{x}), \quad (1)$$

This research was supported by the U.S. Army Research Office grant W911NF1810309, and by the U.S. Air Force Office of Scientific Research grant FA9550-20-1-0174.

✉ Daniele Venturi
venturi@ucsc.edu
Abram Rodgers
akrodger@ucsc.edu

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

arise in many areas of mathematical physics, e.g., in statistical mechanics [7, 39], quantum field theory [50], and in the approximation of functional differential equations (infinite-dimensional PDEs) [47, 48] such as the Hopf equation of turbulence [28], or functional equations modeling deep learning [21]. In Eq. (1), $f : \Omega \times [0, T] \rightarrow \mathbb{R}$ is a d -dimensional time-dependent scalar field defined on the domain $\Omega \subseteq \mathbb{R}^d$ ($d \geq 2$), T is the period of integration, and \mathcal{N} is a nonlinear operator which may depend on the variables $\mathbf{x} = (x_1, \dots, x_d) \in \Omega$, and may incorporate boundary conditions. For simplicity, we assume that the domain Ω is a Cartesian product of d one-dimensional domains Ω_i

$$\Omega = \Omega_1 \times \dots \times \Omega_d, \tag{2}$$

and that f is an element of a Hilbert space $H(\Omega; [0, T])$. In these hypotheses, we can leverage the isomorphism $H(\Omega; [0, T]) \simeq H([0, T]) \otimes H(\Omega_1) \otimes \dots \otimes H(\Omega_d)$ and represent the solution of (1) as

$$f(\mathbf{x}, t) \approx \sum_{i_1=1}^{n_1} \dots \sum_{i_d=1}^{n_d} f_{i_1 \dots i_d}(t) \phi_{i_1}(x_1) \dots \phi_{i_d}(x_d), \tag{3}$$

where $\phi_{i_j}(x_j)$ are one-dimensional orthonormal basis functions of $H(\Omega_i)$. Substituting (3) into (1) and projecting onto an appropriate finite-dimensional subspace of $H(\Omega)$ yields the semi-discrete form

$$\frac{d\mathbf{f}}{dt} = \mathbf{G}(\mathbf{f}), \quad \mathbf{f}(0) = \mathbf{f}_0 \tag{4}$$

where $\mathbf{f} : [0, T] \rightarrow \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ is a multivariate array with coefficients $f_{i_1 \dots i_d}(t)$, and \mathbf{G} is the finite-dimensional representation of the nonlinear operator \mathcal{N} . The number of degrees of freedom associated with the solution to the Cauchy problem (4) is $N_{\text{dof}} = n_1 n_2 \dots n_d$ at each time $t \geq 0$, which can be extremely large even for moderately small dimension d . For instance, the solution of the Boltzmann-BGK equation on a six-dimensional ($d = 6$) flat torus [9, 18, 34] with $n_i = 128$ basis functions in each position and momentum variable yields $N_{\text{dof}} = 128^6 = 4398046511104$ degrees of freedom at each time t . This requires approximately 35.18 Terabytes per temporal snapshot if we store the solution tensor \mathbf{f} in a double precision IEEE 754 floating point format. Several general-purpose algorithms have been developed to mitigate such an exponential growth of degrees of freedom, the computational cost, and the memory requirements. These algorithms include, e.g., sparse collocation methods [6, 10, 23, 36], high-dimensional model representation (HDMR) [5, 11, 33], and techniques based on deep neural networks [12, 37, 38, 49].

In a parallel research effort that has its roots in quantum field theory and quantum entanglement, researchers have recently developed a new generation of algorithms based on tensor networks and low-rank tensor techniques to compute the solution of high-dimensional PDEs [4, 8, 13, 30, 31]. Tensor networks are essentially factorizations of entangled objects such as multivariate functions or operators, into networks of simpler objects which are amenable to efficient representation and computation. The process of building a tensor network relies on a hierarchical decomposition that can be visualized in terms of trees, and has its roots in the spectral theory for linear operators. Such rigorous mathematical foundations can be leveraged to construct high-order methods to compute the numerical solution of high-dimensional Cauchy problems of the form (4) at a cost that scales linearly with respect to the dimension d , and polynomially with respect to the tensor rank.

In particular, a new class of algorithms to integrate (4) on a low-rank tensor manifold was recently proposed in [15, 16, 31, 40, 41, 47]. These algorithms are known as explicit

step-truncation methods and they are based on integrating the solution $f(t)$ off the tensor manifold for a short time using any conventional explicit time-stepping scheme, and then mapping it back onto the manifold using a tensor truncation operation (see Fig. 2). To briefly describe these methods, let us discretize the ODE (4) in time with a one-step method on an evenly-spaced temporal grid as

$$f_{k+1} = \Psi_{\Delta t}(\mathbf{G}, f_k), \quad f_0 = f(0), \quad (5)$$

where f_k denotes an approximation of $f(k\Delta t)$ for $k = 0, 1, \dots$, and $\Psi_{\Delta t}$ is an increment function. To obtain a step-truncation integrator, we simply apply a truncation operator $\mathfrak{T}_r(\cdot)$, i.e., a nonlinear projection onto a tensor manifold \mathcal{H}_r with multilinear rank r [46] to the scheme (5). This yields

$$f_{k+1} = \mathfrak{T}_r(\Psi_{\Delta t}(\mathbf{G}, f_k)). \quad (6)$$

The need for tensor rank-reduction when iterating (5) can be easily understood by noting that tensor operations such as the application of an operator to a tensor and the addition between two tensors naturally increase tensor rank [32]. Hence, iterating (6) with no rank reduction can yield a fast increase in tensor rank, which, in turn, can tax computational resources heavily.

Explicit step-truncation algorithms of the form (6) were studied extensively in [31, 40]. In particular, error estimates and convergence results were obtained for both fixed-rank and rank-adaptive integrators, i.e., integrators in which the tensor rank r is selected at each time step based on accuracy and stability constraints. Step-truncation methods are very simple to implement as they rely only on arithmetic operations between tensors, which can be performed by scalable parallel algorithms [3, 14, 25, 43].

While explicit step-truncation methods have proven successful in integrating a wide variety of high-dimensional initial value problems, their effectiveness for stiff problems is limited. Indeed, the combination of non-linearity and stiffness may introduce time-step restrictions which could make explicit step-truncation integration computationally infeasible. As an example, in Fig. 1 we show that the explicit step-truncation midpoint method applied to the Allen–Cahn equation

$$\frac{\partial f}{\partial t} = \varepsilon \Delta f + f - f^3, \quad (7)$$

undergoes a numerical instability for $\Delta t = 10^{-3}$.

The main objective of this paper is to develop a new class of rank-adaptive *implicit* step-truncation algorithms to integrate high-dimensional initial value problems of the form (4) on low-rank tensor manifolds. The main idea of these new integrators is illustrated in Fig. 2. Roughly speaking, implicit step-truncation method take $f_k \in \mathcal{H}_r$ (\mathcal{H}_r is a HT or TT tensor manifold with multilinear rank r) and $\Psi_{\Delta t}(\mathbf{G}, f_k)$ as input and generate a sequence of inexact Newton iterates $f^{[j]}$ converging to a point tensor manifold \mathcal{H}_s . Once $f^{[j]}$ is sufficiently close to \mathcal{H}_s , we project it onto the manifold via a standard truncation operation. This operation is also known as “compression step” in the HT/TT-GMRES algorithm described in [20]. Of course the computational cost of implicit step-truncation methods is higher than that of explicit step truncation methods for one single step. However, implicit methods allow to integrate stably with larger time-steps while retaining accuracy. Previous research on implicit tensor integration leveraged the Alternating Least Squares (ALS) algorithm [8, 13, 19], which essentially attempts to solve an optimization problem on a low-rank tensor manifold to compute the solution of (4) at each time step. As is well-known, ALS is equivalent to the

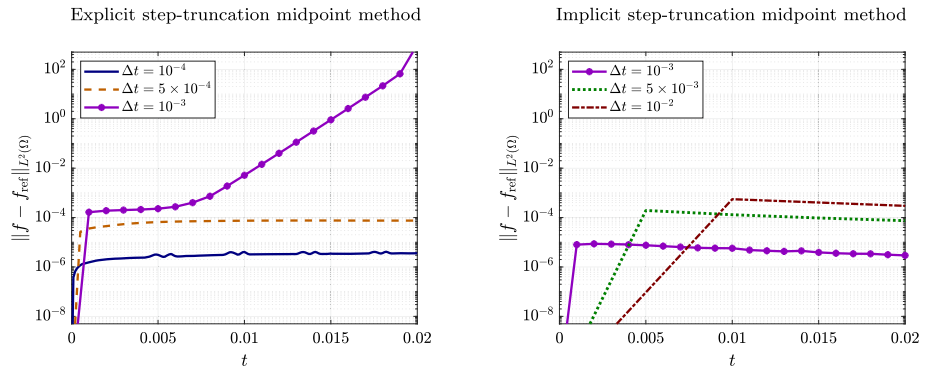


Fig. 1 Explicit and implicit step-truncation midpoint methods applied the Allen–Cahn equation (7) with $\varepsilon = 0.1$. It is seen that the explicit step-truncation midpoint method undergoes a numerical instability for $\Delta t = 10^{-3}$ while the implicit step-truncation midpoint method retains accuracy and stability for $\Delta t = 10^{-3}$, and even larger time steps. Stability implicit step-truncation midpoint is studied in Sect. 5

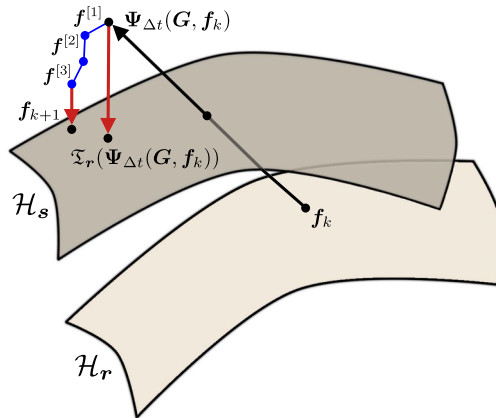


Fig. 2 Sketch of implicit and explicit step-truncation integration methods. Given a tensor f_k with multilinear rank r on the tensor manifold \mathcal{H}_r , we first perform an explicit time-step, e.g., with the conventional time-stepping scheme (5). The explicit step-truncation integrator then projects $\Psi_{\Delta t}(G, f_k)$ onto a new tensor manifold \mathcal{H}_s (solid red line). The multilinear rank s is chosen adaptively based on desired accuracy and stability constraints [40]. On the other hand, the implicit step-truncation method takes $\Psi_{\Delta t}(G, f_k)$ as input and generates a sequence of fixed-point iterates $f^{[j]}$ shown as dots connected with blue lines. The last iterate is then projected onto a low rank tensor manifold, illustrated here also as a red line landing on \mathcal{H}_s . This operation is equivalent to the compression step in the HT/TT-GMRES algorithm described in [20]

(linear) block Gauss-Seidel iteration applied to the Hessian matrix of the residual, and can have convergence issues [45].

The paper is organized as follows. In Sect. 2, we briefly review rank-adaptive explicit step-truncation methods, and present a new convergence proof for these methods which applies also to implicit step-truncation methods. In Sect. 3, we discuss the proposed new algorithms for implicit step-truncation integration. In Sect. 4 we study convergence of particular implicit step-truncation methods, namely the step-truncation implicit Euler and midpoint methods. In Sect. 5 we prove that the stability region of an implicit step-truncation method is identical to that of the implicit method without tensor truncation. Finally, in Sect. 6 we present numerical applications of implicit step-truncation algorithms to stiff PDEs. In particular, we study a

two-dimensional Allen–Cahn equation, a four-dimensional Fokker–Planck equation, and a six-dimensional nonlinear Schrödinger equation. We also include a brief appendix in which we discuss numerical algorithms to solve linear and nonlinear algebraic equations on tensor manifolds via the inexact Newton’s method with HT/TT-GMRES iterations.

2 Explicit Step-Truncation Methods

In this section we briefly review explicit step-truncation methods to integrate the tensor-valued ODE (4) on tensor manifolds with variable rank. For a complete account of this theory see [40]. We begin by first discretizing the ODE in time with any standard explicit one-step method on an evenly-spaced temporal grid

$$f_{k+1} = \Psi_{\Delta t}(\mathbf{G}, f_k). \tag{8}$$

Here, f_k denotes an approximation of the exact solution $f(k\Delta t)$ for $k = 1, 2, \dots, N$, and $\Psi_{\Delta t}$ is an increment function. For example, $\Psi_{\Delta t}$ can be the increment function corresponding to the Euler forward method

$$\Psi_{\Delta t}(\mathbf{G}, f_k) = f_k + \Delta t \mathbf{G}(f_k). \tag{9}$$

In the interest of saving computational resources when iterating (8) we look for an approximation of f_k on a low-rank tensor manifold \mathcal{H}_r [46] with multilinear rank r . \mathcal{H}_r is taken to be the manifold of Hierarchical Tucker (HT) tensors. The easiest way for approximating (8) on \mathcal{H}_r is to apply a nonlinear projection operator [24] (truncation operator)

$$\mathfrak{T}_r : \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d} \rightarrow \overline{\mathcal{H}}_r, \tag{10}$$

where $\overline{\mathcal{H}}_r$ denotes the closure of \mathcal{H}_r . This yields the explicit step-truncation scheme

$$f_{k+1} = \mathfrak{T}_r(\Psi_{\Delta t}(\mathbf{G}, f_k)). \tag{11}$$

The rank r can vary with the time step based on appropriate error estimates as time integration proceeds [40]. We can also project $\mathbf{G}(f)$ onto $\overline{\mathcal{H}}_r$ before applying $\Psi_{\Delta t}$. With reference to (9) this yields

$$f_{k+1} = \mathfrak{T}_{r_2}(f_k + \Delta t \mathfrak{T}_{r_1}(\mathbf{G}(f_k))). \tag{12}$$

Here r_1 and r_2 are truncation ranks determined by the inequalities¹

$$\|\mathbf{G}(f_k) - \mathfrak{T}_{r_1}(\mathbf{G}(f_k))\| \leq e_1, \quad \|\tilde{f}_{k+1} - \mathfrak{T}_{r_2}(\tilde{f}_{k+1})\| \leq e_2, \tag{13}$$

where e_1 and e_2 are chosen error thresholds. As before, r_1 and r_2 can change with every time step. In particular, if we choose $e_1 = K_1 \Delta t$ and $e_2 = K_2 \Delta t^2$ (with K_1 and K_2 given constants) then the step-tuncation method (12) is convergent (see [40] for details). More generally, let

$$f_{k+1} = \Phi_{\Delta t}(\mathbf{G}, f_k, e) \tag{14}$$

be an explicit step-truncation method in which we project all $\mathbf{G}(f_k)$ appearing in the increment function $\Psi_{\Delta t}(\mathbf{G}, f_k)$ onto tensor manifolds $\overline{\mathcal{H}}_{r_i}$ by setting suitable error thresholds $e = (e_1, e_2, \dots)$. For instance, if $\Psi_{\Delta t}$ is defined by the explicit midpoint method, i.e.,

$$\Psi_{\Delta t}(\mathbf{G}, f_k) = f_k + \mathbf{G}\left(f_k + \frac{\Delta t}{2} \mathbf{G}(f_k)\right) \tag{15}$$

¹ Throughout the paper, $\|\cdot\|$ denotes the standard tensor 2-norm [24, 32], or a weighted version of it.

then

$$\Phi_{\Delta t}(\mathbf{G}, \mathbf{f}_k, \mathbf{e}) = \mathfrak{T}_{\mathbf{r}_3} \left(\mathbf{f}_k + \mathfrak{T}_{\mathbf{r}_2} \left[\mathbf{G} \left(\mathbf{f}_k + \frac{\Delta t}{2} \mathfrak{T}_{\mathbf{r}_1} [\mathbf{G}(\mathbf{f}_k)] \right) \right] \right), \tag{16}$$

where $\mathbf{e} = (e_1, e_2, e_3)$ is a vector collecting the truncation error thresholds yielding the multilinear ranks $\mathbf{r}_1, \mathbf{r}_2$ and \mathbf{r}_3 . By construction, step-truncation methods of the form (14) satisfy

$$\|\Psi_{\Delta t}(\mathbf{G}, \mathbf{f}) - \Phi_{\Delta t}(\mathbf{G}, \mathbf{f}, \mathbf{e})\| \leq R(\mathbf{e}), \tag{17}$$

where $R(\mathbf{e})$ is the error due to tensor truncation. We close this section with a reformulation of the convergence theorem for explicit step-truncation methods in [40], which applies also to implicit methods.

Theorem 1 (Convergence of step-truncation methods) *Let $\varphi_{\Delta t}(\mathbf{G}, \mathbf{f})$ be the one-step exact flow map defined by (4), and $\Phi_{\Delta t}(\mathbf{G}, \mathbf{f}, \mathbf{e})$ be the increment function of a step-truncation method with local error of order p , i.e.,*

$$\|\varphi_{\Delta t}(\mathbf{G}, \mathbf{f}) - \Phi_{\Delta t}(\mathbf{G}, \mathbf{f}, \mathbf{e})\| \leq K \Delta t^{p+1} \text{ as } \Delta t \rightarrow 0. \tag{18}$$

If there exist truncation errors $\mathbf{e} = \mathbf{e}(\Delta t)$ (function of Δt) and constants $C, E > 0$ (dependent on \mathbf{G}) so that the stability condition

$$\|\Phi_{\Delta t}(\mathbf{G}, \hat{\mathbf{f}}, \mathbf{e}) - \Phi_{\Delta t}(\mathbf{G}, \tilde{\mathbf{f}}, \mathbf{e})\| \leq (1 + C \Delta t) \|\hat{\mathbf{f}} - \tilde{\mathbf{f}}\| + E \Delta t^{m+1} \tag{19}$$

holds as $\Delta t \rightarrow 0$, then the step-truncation method is convergent with order $z = \min(m, p)$.

Proof Under the assumption that Δt is small enough for our stability and consistency to hold, we proceed by induction on the number of steps. The one-step case coincides with the consistency condition. Next, we assume that

$$\|\varphi_{\Delta t(N-1)}(\mathbf{G}, \mathbf{f}_0) - \mathbf{f}_{N-1}\| \leq Q_{N-1} \Delta t^z. \tag{20}$$

Let $T = N \Delta t$ be the final integration time. By applying triangle inequality and the semigroup property of the flow map,

$$\begin{aligned} \|\varphi_T(\mathbf{G}, \mathbf{f}_0) - \Phi_{\Delta t}(\mathbf{G}, \mathbf{f}_{N-1}, \mathbf{e})\| &\leq \|\varphi_{\Delta t}(\mathbf{G}, \varphi_{\Delta t(N-1)}(\mathbf{G}, \mathbf{f}_0)) - \Phi_{\Delta t}(\mathbf{G}, \varphi_{\Delta t(N-1)}(\mathbf{G}, \mathbf{f}_0), \mathbf{e})\| \\ &\quad + \|\Phi_{\Delta t}(\mathbf{G}, \varphi_{\Delta t(N-1)}(\mathbf{G}, \mathbf{f}_0), \mathbf{e}) - \Phi_{\Delta t}(\mathbf{G}, \mathbf{f}_{N-1}, \mathbf{e})\|. \end{aligned} \tag{21}$$

Using the consistency condition (18) we can bound the first term at the right hand side of (21) by $K_{N-1} \Delta t^{p+1}$, where K_{N-1} represents a local error coefficient. On the other hand, using the stability condition (19) we can bound the second term at the right hand side of (21) as

$$\begin{aligned} &\|\Phi_{\Delta t}(\mathbf{G}, \varphi_{\Delta t(N-1)}(\mathbf{G}, \mathbf{f}_0), \mathbf{e}) - \Phi_{\Delta t}(\mathbf{G}, \mathbf{f}_{N-1}, \mathbf{e})\| \\ &\leq (1 + C \Delta t) \|\varphi_{\Delta t(N-1)}(\mathbf{G}, \mathbf{f}_0) - \mathbf{f}_{N-1}\| + E \Delta t^{m+1}. \end{aligned} \tag{22}$$

A substitution of (22) and (20) into (21) yields

$$\|\varphi_T(\mathbf{G}, \mathbf{f}_0) - \Phi_{\Delta t}(\mathbf{G}, \mathbf{f}_{N-1}, \mathbf{e})\| \leq K_{N-1} \Delta t^{p+1} + (1 + C \Delta t) Q_{N-1} \Delta t^z + E \Delta t^{m+1}.$$

Recalling that $z = \min(m, p)$ completes the proof. □

We remark that the above proof may be modified to include explicit step-truncation linear multi-step methods, i.e., step-truncation Adams methods. To this end, it is sufficient to replace f_k with the vector $(f_{k-s}, f_{k-s+1}, \dots, f_k)$ and the stability condition (19) with

$$\begin{aligned} & \left\| \Phi_{\Delta t}(\mathbf{G}, \hat{f}_1, \hat{f}_2, \dots, \hat{f}_s, \mathbf{e}) - \Phi_{\Delta t}(\mathbf{G}, \tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_s, \mathbf{e}) \right\| \\ & \leq \left\| \hat{f}_s - \tilde{f}_s \right\| + \sum_{j=1}^s C_j \Delta t \left\| \hat{f}_j - \tilde{f}_j \right\| + E \Delta t^{m+1}. \end{aligned} \tag{23}$$

We then have an analogous theorem proven using an inductive argument based on the initial conditions $f(0), f(\Delta t), \dots, f((s-1)\Delta t)$.

3 Implicit Step-Truncation Methods

To introduce implicit step-truncation tensor methods, let us begin with the standard Euler backward scheme

$$f_{k+1} = f_k + \Delta t \mathbf{G}(f_{k+1}), \tag{24}$$

and the associated root-finding problem

$$\mathbf{H}_k(f_{k+1}) = f_{k+1} - f_k - \Delta t \mathbf{G}(f_{k+1}) = \mathbf{0}. \tag{25}$$

Equation (25) allows us to compute f_{k+1} as a zero of the function \mathbf{H}_k . This can be done, e.g., using the Netwon’s method with initial guess $f_{k+1}^{[0]} = f_k$. As is well-known, if the Jacobian of \mathbf{H}_k is invertible within a neighborhood of f_k , then the implicit function theorem guarantees the existence of a locally differentiable (in some neighborhood of f_k) nonlinear map $\Theta_{\Delta t}$ depending on \mathbf{G} such that

$$f_{k+1} = \Theta_{\Delta t}(\mathbf{G}, f_k). \tag{26}$$

In the setting of Newton’s method described above, the map $\Theta_{\Delta t}$ is computed iteratively. An implicit step-truncation scheme can be then formulated by applying the tensor truncation operator \mathfrak{T}_r to the right hand side of (26), i.e.,

$$f_{k+1} = \mathfrak{T}_r(\Theta_{\Delta t}(\mathbf{G}, f_k)). \tag{27}$$

The tensor truncation rank r can be selected based on the inequality

$$\|\mathfrak{T}_r(\Theta_{\Delta t}(\mathbf{G}, f_k)) - \Theta_{\Delta t}(\mathbf{G}, f_k)\| \leq A \Delta t^2. \tag{28}$$

It was shown in [40] that this yields an order one (in time) integration scheme. Of course, if the Jacobian of \mathbf{H}_k in Eq. (25) can be computed and stored in computer memory, then we can approximate $f_{k+1} = \Theta_{\Delta t}(\mathbf{G}, f_k)$ for any given f_k and \mathbf{G} using Newton’s iterations. However, the exact Newton’s method is not available to us in the high-dimensional tensor setting.

Hence, we look for an approximation of $\Theta_{\Delta t}(\mathbf{G}, f_k)$ computed using the inexact Newton method [17]. The inexact Newton method is an algorithm for optimization and root finding which allows us to apply approximate inverses of the Jacobian of \mathbf{H}_k in (25) in exchange for weakening the convergence rate of the Cauchy sequence being produced. In practice, convergence is either superlinear or linear rather than quadratic. To describe the inexact

Newton’s method, consider the conventional Newton’s iteration for solving the system of nonlinear equations $\mathbf{H}_k(\mathbf{f}_{k+1}) = \mathbf{0}$, i.e.,

$$\mathbf{f}_{k+1}^{[j+1]} = \mathbf{f}_{k+1}^{[j]} - \left[\mathbf{J}_{\mathbf{H}_k} \left(\mathbf{f}_{k+1}^{[j]} \right) \right]^{-1} \mathbf{H}_k \left(\mathbf{f}_{k+1}^{[j]} \right), \quad j = 1, 2, \dots$$

where $\mathbf{J}_{\mathbf{H}_k} \left(\mathbf{f}_{k+1}^{[j]} \right)$ is the Jacobian of \mathbf{H}_k evaluated at $\mathbf{f}_{k+1}^{[j]}$. The inexact version of this iteration replaces the inverse Jacobian term with an approximate inverse Jacobian obtained by solving a system of linear equations up to a specified relative error. It turns out that inaccuracies generated by low rank tensor approximations of the inverse Jacobian are compatible with the assumptions on inexactness outlined in [17]. Hence, to compute the inexact matrix inverse, we evaluate $\mathbf{J}_{\mathbf{H}_k}$ analytically and apply the relaxed TT-GMRES algorithm described in [20] to the linear system

$$\mathbf{J}_{\mathbf{H}_k} \left(\mathbf{f}_{k+1}^{[j]} \right) \mathbf{s}_{k+1}^{[j]} = -\mathbf{H}_k \left(\mathbf{f}_{k+1}^{[j]} \right). \tag{29}$$

and then update $\mathbf{f}_{k+1}^{[j]}$ using $\mathbf{f}_{k+1}^{[j+1]} = \mathfrak{T}_r \left(\mathbf{f}_{k+1}^{[j]} + \mathbf{s}_{k+1}^{[j]} \right)$. TT-GMRES is an iterative method for the solution of linear equations which makes use of an inexact matrix–vector product defined by low-rank truncation. Though the algorithm was developed for TT tensors, it may be also applied to HT tensors without significant changes.

In Appendix A we describe the proposed inexact Newton’s method with HT/TT-GMRES iterations to solve an arbitrary algebraic equation of the form $\mathbf{H}_k(\mathbf{f}_{k+1}) = \mathbf{0}$ (e.g., equation (25)) on a tensor manifold with a given rank. The algorithm can be used to approximate the mapping $\Theta_{\Delta t}(\mathbf{G}, \mathbf{f}_k)$ corresponding to any implicit integrator, and it returns a tensor which can be then truncated further as in (27). This operation is equivalent to the so-called “compression step” in [20] and it is described in detail in the next section.

3.1 The Compression Step

While increasing the tensor rank may be necessary for convergence of the HT/TT-GMRES iterations, it is possible that we raise the rank by more than is required for the desired order of accuracy in a single time step. Therefore, it is convenient to apply an additional tensor truncation after computing, say, j steps of the HT inexact Newton’s method which returns $\mathbf{f}_{k+1}^{[j]}$. This is the same as the “compression step” at the end of HT/TT-GMRES algorithm as presented in [20]. This gives us our final estimate of $\mathbf{f}((k + 1)\Delta t)$ as

$$\mathbf{f}_{k+1} = \mathfrak{T}_r \left(\mathbf{f}_{k+1}^{[j]} \right), \quad \left\| \mathfrak{T}_r \left(\mathbf{f}_{k+1}^{[j]} \right) - \mathbf{f}_{k+1}^{[j]} \right\| \leq e_r. \tag{30}$$

Regarding the selection of the truncation error e_r we proceed as follows. Suppose that $\Psi_{\Delta t}(\mathbf{G}, \mathbf{f}_k)$ is an explicit integration scheme of the same order (or higher) than the implicit scheme being considered. Then we can estimate the local error as

$$\begin{aligned} \left\| \mathbf{f}_{k+1}^{[j]} - \mathbf{f}((k + 1)\Delta t) \right\| &\leq \left\| \mathbf{f}_{k+1}^{[j]} - \Psi_{\Delta t}(\mathbf{G}, \mathbf{f}_k) \right\| + \left\| \Psi_{\Delta t}(\mathbf{G}, \mathbf{f}_k) - \mathbf{f}((k + 1)\Delta t) \right\| \\ &= \left\| \mathbf{f}_1^{[j]} - \Psi_{\Delta t}(\mathbf{G}, \mathbf{f}_k) \right\| + O(\Delta t^{p+1}), \end{aligned} \tag{31}$$

Thus, we may roughly estimate the local truncation error and set this as the chosen error for approximation to HT or TT rank r using

$$e_r = \left\| \mathbf{f}_{k+1}^{[j]} - \Psi_{\Delta t}(\mathbf{G}, \mathbf{f}_k) \right\|. \tag{32}$$

We may drop more singular values than needed, especially if the choice of Δt is outside the region of stability of the explicit scheme $\Psi_{\Delta t}(\mathbf{G}, \mathbf{f}_k)$. However, this estimate guarantees that we do not change the overall convergence rate. Moreover, it cannot impact stability of the implicit step-truncation integrator since the compression step has operator norm equal to one, regardless of the rank chosen. This statement is supported by the analysis presented in [41] for step-truncation linear multi-step methods. In all our numerical experiments we use the explicit step-truncation midpoint method to estimate local error, i.e., $\Psi_{\Delta t}(\mathbf{G}, \mathbf{f}_k)$ in (32) is set as in (15).

4 Convergence Analysis of Implicit Step-Truncation Methods

In this section we show that applying the inexact Newton iterations with HT/TT-GMRES to the implicit Euler and the implicit midpoint methods result in implicit step-truncation schemes that fit into the framework of Theorem 1, and therefore are convergent.

4.1 Implicit Step-Truncation Euler Method

Consider the implicit Euler scheme (25), and suppose that the solution of the nonlinear equation $\mathbf{H}_k(\mathbf{f}_{k+1}) = \mathbf{0}$ is computed using the inexact Newton method with HT/TT-GMRES iterations as discussed in Appendix A. Let $\mathbf{f}_{k+1}^{[j]}$ ($j = 1, 2, \dots$) be the sequence of tensors generated by the algorithm and approximating \mathbf{f}_{k+1} (exact solution of $\mathbf{H}_k(\mathbf{f}_{k+1}) = \mathbf{0}$). We set a stopping criterion for terminating Newton’s iterations based on the residual, i.e.,

$$\|\mathbf{H}_k(\mathbf{f}_{k+1}^{[j]})\| \leq \varepsilon_{\text{tol}}. \tag{33}$$

This allows us to adjust the rank of $\mathbf{f}_{k+1}^{[j]}$ from one time step to the next, depending on the desired accuracy ε_{tol} . Our goal is to analyze convergence of such rank-adaptive implicit method when a finite number of contraction mapping steps is taken. To this end, we will fit the method into the framework of Theorem 1. We begin by noticing that

$$\begin{aligned} \|\mathbf{f}_{k+1} - \mathbf{f}_{k+1}^{[j]}\| &= \|\mathbf{H}_k^{-1}(\mathbf{H}_k(\mathbf{f}_{k+1})) - \mathbf{H}_k^{-1}(\mathbf{H}_k(\mathbf{f}_{k+1}^{[j]}))\| \\ &\leq L_{\mathbf{H}_k^{-1}} \|\mathbf{H}_k(\mathbf{f}_{k+1}^{[j]})\| \leq L_{\mathbf{H}_k^{-1}} \varepsilon_{\text{tol}}, \end{aligned} \tag{34}$$

where $L_{\mathbf{H}_k^{-1}}$ is the local Lipschitz constant of the smooth inverse map \mathbf{H}_k^{-1} , the existence of which is granted by the inverse function theorem. This allows us to write the local truncation error as

$$\|\mathbf{f}(\Delta t) - \mathbf{f}_1^{[j]}\| \leq \|\mathbf{f}(\Delta t) - \mathbf{f}_1\| + \|\mathbf{f}_1 - \mathbf{f}_1^{[j]}\| \leq K_1 \Delta t^2 + L_{\mathbf{H}^{-1}} \varepsilon_{\text{tol}}, \tag{35}$$

where K_1 is a local error coefficient. In order to maintain order one convergence, we require that $\varepsilon_{\text{tol}} \leq K \Delta t^2$ for some constant $K \geq 0$.

Next, we discuss the stability condition (19) in the context of the implicit step-truncation Euler scheme, assuming that \mathbf{f}_{k+1} can be found exactly by the inexact Newton’s method, eventually after an infinite number of iterations. Denote by $\hat{\mathbf{f}}_0, \tilde{\mathbf{f}}_0$ two different initial conditions. Performing one step of the standard implicit Euler’s method yields the following bound

$$\|\hat{f}_1 - \tilde{f}_1\| = \|(\hat{f}_0 + \Delta t G(\hat{f}_1)) - (\tilde{f}_0 + \Delta t G(\tilde{f}_1))\| \leq \|\hat{f}_0 - \tilde{f}_0\| + \Delta t L_G \|\hat{f}_1 - \tilde{f}_1\|.$$

where L_G is the Lipschitz constant of G . By collecting like terms, we obtain

$$\|\hat{f}_1 - \tilde{f}_1\| \leq \frac{1}{1 - \Delta t L_G} \|\hat{f}_0 - \tilde{f}_0\| \leq (1 + 2\Delta t L_G) \|\hat{f}_0 - \tilde{f}_0\|. \tag{36}$$

The last inequality comes by noting that $1/(1 - \Delta t L_G) \leq 1 + 2\Delta t L_G$ when Δt is sufficiently small, i.e., $\Delta t \leq 1/(2L_G)$. This is zero-stability condition, i.e., a stability condition that holds for small Δt , which will be used for convergence analysis. Regarding the behavior of the scheme for finite Δt we will show in Sect. 5 that the implicit step-truncation Euler scheme is unconditionally stable. Next, we derive a stability condition of the form (36) when the root of $H_k(f_{k+1}) = \mathbf{0}$ is computed with the inexact Newton’s method with HT/TT-GMRES iterations. In this case we have

$$\begin{aligned} \|\hat{f}_1^{[j]} - \tilde{f}_1^{[m]}\| &\leq \|\hat{f}_1^{[j]} - \hat{f}_1^{[\infty]}\| + \|\hat{f}_1^{[\infty]} - \tilde{f}_1^{[\infty]}\| + \|\tilde{f}_1^{[m]} - \tilde{f}_1^{[\infty]}\| \\ &= \|\hat{f}_1^{[j]} - \hat{f}_1\| + \|\hat{f}_1 - \tilde{f}_1\| + \|\tilde{f}_1^{[m]} - \tilde{f}_1\| \\ &\leq (1 + 2L_G \Delta t) \|\hat{f}_0 - \tilde{f}_0\| + 2L_{H^{-1}} \varepsilon_{\text{tol}}. \end{aligned} \tag{37}$$

Thus, the stability condition is satisfied by the same condition on ε_{tol} that satisfies the first-order consistency condition (35), i.e., $\varepsilon_{\text{tol}} \leq K \Delta t^2$. At this point we apply Theorem 1 with consistency and stability conditions (18)–(19) replaced by (35) and (37), respectively, and conclude that the implicit step-truncation Euler scheme is convergent with order one if $\varepsilon_{\text{tol}} \leq K \Delta t^2$.

Remark 1 When G is linear, i.e., when the tensor ODE (4) is linear, then we may apply HT/TT-GMRES algorithm in Appendix A without invoking Newton’s method. In this case, the local error coefficient $L_{H_k^{-1}}$ can be exchanged for the coefficient of ε at the right side of inequality (79).

4.2 Implicit Step-Truncation Midpoint Method

The implicit midpoint rule [26],

$$f_{k+1} = f_k + \Delta t G \left(\frac{1}{2} (f_k + f_{k+1}) \right), \tag{38}$$

is a symmetric and symplectic method of order 2. By introducing

$$H_k(f_{k+1}) = f_{k+1} - f_k - \Delta t G \left(\frac{1}{2} (f_k + f_{k+1}) \right) = \mathbf{0} \tag{39}$$

we again see the implicit method as a root finding problem at each time step. To prove convergence of the implicit step-truncation midpoint method we follow the same steps described in the previous section. To this end, consider the sequence of tensors $f_{k+1}^{[j]}$ generated by the inexact Newton method with HT/TT-GMRES iterations (see Appendix A) applied to (39). The sequence of tensors $f_{k+1}^{[j]}$ approximates f_{k+1} satisfying (39). As before, we terminate the inexact Newton’s iterations as soon as condition (33) is satisfied. By repeating the same steps that led us to inequality (34), we have

$$\|f(\Delta t) - f_1^{[j]}\| \leq K_1 \Delta t^3 + L_{H_k^{-1}} \varepsilon_{\text{tol}}. \tag{40}$$

Hence, setting the stopping tolerance as $\varepsilon_{\text{tol}} \leq K \Delta t^3$ we get second-order consistency. We use this to determine the stability condition (19). As before, we derive the condition for when the zero of (39) is exact. Denote by \hat{f}_0, \tilde{f}_0 two different initial conditions. Performing one step of the standard implicit midpoint method yields the following bound

$$\|\hat{f}_1 - \tilde{f}_1\| \leq \left(1 + \Delta t \frac{LG}{2}\right) \|\hat{f}_0 - \tilde{f}_0\| + \Delta t \frac{LG}{2} \|\hat{f}_1 - \tilde{f}_1\|.$$

By collecting like terms we see that when $\Delta t \leq 2/(3LG)$,

$$\begin{aligned} \|\hat{f}_1 - \tilde{f}_1\| &\leq \frac{2 + \Delta t LG}{2 - \Delta t LG} \|\hat{f}_0 - \tilde{f}_0\| \\ &\leq \left(1 + \Delta t \frac{3LG}{2}\right) \|\hat{f}_0 - \tilde{f}_0\|. \end{aligned} \tag{41}$$

The zero-stability condition for the implicit step-truncation midpoint method can now be found by repeating the arguments of inequality (37). This gives

$$\|\hat{f}_1^{[j]} - \tilde{f}_1^{[m]}\| \leq \left(1 + \Delta t \frac{3LG}{2}\right) \|\hat{f}_0 - \tilde{f}_0\| + 2L_{H_k^{-1}} \varepsilon_{\text{tol}}. \tag{42}$$

Thus, the stability condition is satisfied by the same condition on ε_{tol} that satisfies the second-order consistency condition (40), i.e., $\varepsilon_{\text{tol}} \leq K \Delta t^3$. At this point we apply Theorem 1 with consistency and stability conditions (18)–(19) replaced by (40) and (42), respectively, and conclude that the implicit step-truncation midpoint scheme is convergent with order two if $\varepsilon_{\text{tol}} \leq K \Delta t^3$.

5 Stability Analysis

We now address absolute stability of the proposed implicit step-truncation methods. The study on absolute stability allows us to determine whether the timing stepping scheme is robust to perturbations due finite Δt and finite tensor truncation tolerance. As is well-known, absolute stability analysis of classical time integration schemes allows us to claim stability over a large number iterations for finite Δt and for specific linear prototype problems. For step-truncation integration on tensor manifolds, we have additional perturbations due to the tensor truncation induced by nonzero HT/TT-GMRES stopping tolerance. To study absolute stability of implicit step-truncation schemes we consider the following linear initial value problem

$$\frac{df}{dt} = Lf, \quad f(0) = f_0, \tag{43}$$

where L is a linear operator with eigenvalues in in the left half complex plane. After applying any standard implicit time stepping scheme, we end up with a system of linear equations of the form

$$Af_{k+1} = Wf_k. \tag{44}$$

Specifically, for the implicit Euler we have $A = I - \Delta tL$, $W = I$ while for the implicit midpoint method we have $A = I - 0.5\Delta tL$, $W = I + 0.5\Delta tL$. As is well known, both implicit Euler and Implicit midpoint are unconditionally stable, in the sense that for any

$\Delta t > 0$, $\|f_k\| \rightarrow 0$ as $k \rightarrow \infty$. One way of proving this is by noting that whenever the eigenvalues of L have negative real part, we get

$$\|A^{-1}W\| < 1, \tag{45}$$

and therefore the mapping $A^{-1}W$ is contractive. This implies the sequence f_k defined by (44) converges to zero. The following theorem characterizes what happens when we exchange exact matrix inverse A^{-1} with an inexact inverse computed by tensor HT/TT-GMRES iterations.

Theorem 2 (Absolute stability of implicit step-truncation methods) *Consider an implicit time stepping scheme of the form (44), and suppose that $\|A^{-1}W\| < 1$. Denote by \hat{f}_k the solution of $A\hat{f}_k = W\hat{f}_{k-1}$ ($k = 1, 2, \dots$) obtained with the HT/TT-GMRES tensor solver described in Appendix A, with m Krylov iterations and stopping tolerance η . If*

$$\|A\hat{f}_k - W\hat{f}_{k-1}\| \leq m\|A\| \|A^{-1}\| \eta, \tag{46}$$

then the distance between \hat{f}_k and the exact solution $f_k = A^{-1}Wf_{k-1}$ can be bounded as

$$\|\hat{f}_k - f_k\| \leq \frac{m\|A\| \|A^{-1}\|^2}{1 - \|A^{-1}W\|} \eta. \tag{47}$$

Note that (47) implies that $\|\hat{f}_k\| = O(\eta)$ as $k \rightarrow \infty$. In the context of HT/TT-GMRES iterations, the number η can be controlled by setting the stopping tolerance in Lemma 1 (Appendix A) as

$$\varepsilon_k = \frac{\eta}{\|W\hat{f}_k\|} \tag{48}$$

at each time step k .

Proof The proof follows from a straightforward inductive argument. For $k = 1$ we have

$$\begin{aligned} \hat{f}_1 - f_1 &= A^{-1}A\hat{f}_1 - A^{-1}Wf_0 \\ &= A^{-1} (A\hat{f}_1 - Wf_0). \end{aligned} \tag{49}$$

By using (46), we can bound $\|\hat{f}_1 - f_1\|$ as

$$\|\hat{f}_1 - f_1\| \leq m\|A\| \|A^{-1}\|^2 \eta. \tag{50}$$

For $k = 2$ we have

$$\begin{aligned} \|\hat{f}_2 - f_2\| &\leq \|\hat{f}_2 - A^{-1}W\hat{f}_1\| + \|A^{-1}W\hat{f}_1 - f_2\| \\ &= \|\hat{f}_2 - A^{-1}W\hat{f}_1\| + \|A^{-1}W\hat{f}_1 - A^{-1}Wf_1\| \\ &\leq m\|A\| \|A^{-1}\|^2 \eta + \|A^{-1}W\| \|\hat{f}_1 - f_1\| \\ &\leq (1 + \|A^{-1}W\|) m\|A\| \|A^{-1}\|^2 \eta, \end{aligned}$$

where the last inequality follows from (50). More generally,

$$\left\| \hat{f}_{k-1} - f_{k-1} \right\| \leq m \|A\| \|A^{-1}\|^2 \eta \sum_{j=0}^{k-2} \|A^{-1}W\|^j. \tag{51}$$

Repeating the string of inequalities above and replacing the right sum with the inductive hypothesis, we obtain

$$\begin{aligned} \left\| \hat{f}_k - f_k \right\| &\leq m \|A\| \|A^{-1}\|^2 \eta \sum_{j=0}^{k-1} \|A^{-1}W\|^j \\ &\leq \frac{m \|A\| \|A^{-1}\|^2}{1 - \|A^{-1}W\|} \eta, \end{aligned} \tag{52}$$

which completes the proof. □

Recall that the stability region of an explicit step-truncation method is the same as the corresponding method without truncation [40, 41]. Similarly, Theorem 2 shows that the stability region of an implicit step-truncation method is identical to the corresponding method without truncation, though by relaxing accuracy we see that instead our iterates decay to within the solver tolerance of zero rather than converging to zero in an infinite time horizon. In other words, both implicit step-truncation Euler and implicit step-truncation midpoint methods are *unconditionally stable*. However, if the tolerance of HT/TT-GMRES is set too large, then we could see poor stability behavior akin to an explicit method.

6 Numerical Results

In this section we study the performance of the proposed implicit step-truncation methods in three numerical applications involving time-dependent PDEs. Specifically, we study the Allen–Cahn equation [35] in two spatial dimensions, the Fokker–Planck equation [39] in four dimensions, and the nonlinear Schrödinger equation [44] in six dimensions. Our code was built on the backbone of the HTucker Matlab package [32]. All tensor operations for the step-truncation schemes described in Sects. 2 and 3 use function calls to the HTucker Matlab library.

6.1 Allen–Cahn Equation

The Allen–Cahn equation is a reaction-diffusion equation that describes the process of phase separation in multi-component alloy systems [1, 2]. In its simplest form, the equation has a cubic polynomial non-linearity (reaction term) and a diffusion term [29], i.e.,

$$\frac{\partial f}{\partial t} = \varepsilon \Delta f + f - f^3. \tag{53}$$

In our application, we set $\varepsilon = 0.1$, and solve (53) on the two-dimensional flat torus $\Omega = [0, 2\pi]^2$. We employ a second order splitting [22] method to solve the Laplacian Δf as a fixed rank temporal integration and and cubic $f - f^3$ term using our rank adaptive integration. The initial condition is set as

$$f_0(x, y) = u(x, y) - u(x, 2y) + u(3x + \pi, 3y + \pi) - 2u(4x, 4y) + 2u(5x, 5y), \tag{54}$$

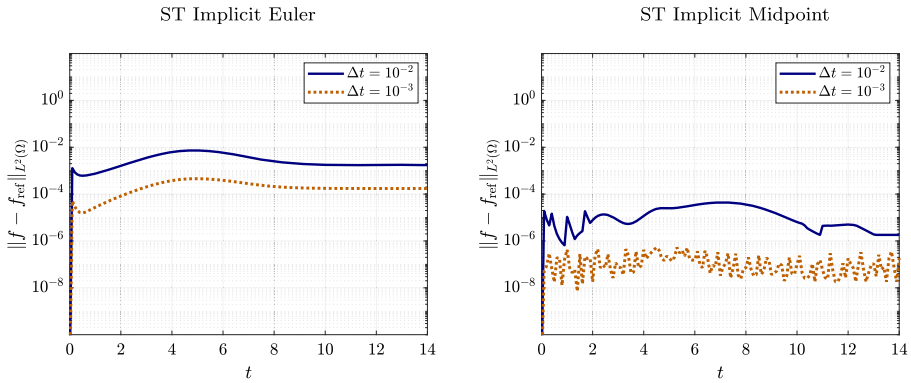


Fig. 3 Error versus time for step-truncation numerical solutions of Allen–Cahn equation (58) in dimension $d = 2$ with initial condition (54)

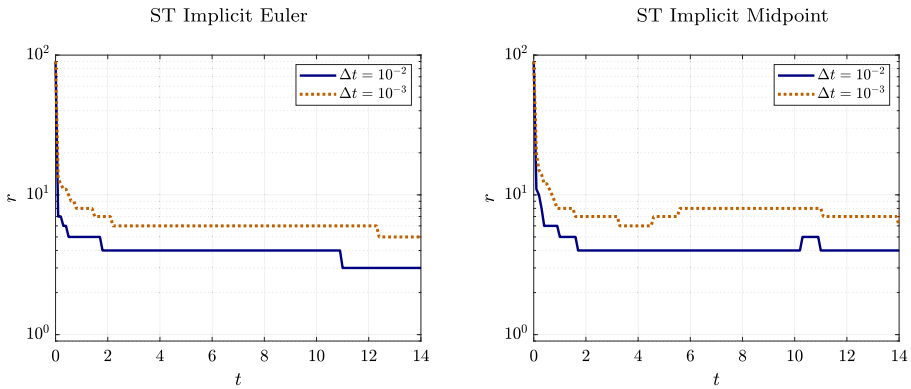


Fig. 4 Rank versus time for step-truncation numerical solutions of Allen–Cahn equation (58) in dimension $d = 2$ with initial condition (54)

where

$$u(x, y) = \frac{\left[e^{-\tan^2(x)} + e^{-\tan^2(y)} \right] \sin(x) \sin(y)}{1 + e^{|\csc(-x/2)|} + e^{|\csc(-y/2)|}}. \tag{55}$$

We discretize (53) in space using the two-dimensional Fourier pseudospectral collocation method [27] with 257×257 points in $\Omega = [0, 2\pi]^2$. This results in a matrix ODE in the form of (4). We truncate the initial condition to absolute and relative SVD tolerances of 10^{-9} , which yields an initial condition represented by a 257×257 matrix of rank 90. We also computed a benchmark solution of the matrix ODE using a variable step RK4 method with absolute tolerance set to 10^{-14} . We denote the benchmark solution as f_{ref} . In Fig. 3 we observe the transient accuracy of our order one and order two implicit methods. The stopping tolerance for inexact Newton’s iterations is set to $\varepsilon_{\text{tol}} = 2.2 \times 10^{-8}$, while and HT/TT-GMRES relative error is chosen as $\eta = 10^{-3}$. Time integration was halted at $t = 14$. After this time, the system is close to steady state and the errors stay bounded near the final values plotted in Fig. 3. Similarly, the rank also levels out around $t = 14$. In Fig. 4, we plot temporal evolution of the rank for both the implicit step-truncation Euler and midpoint methods.

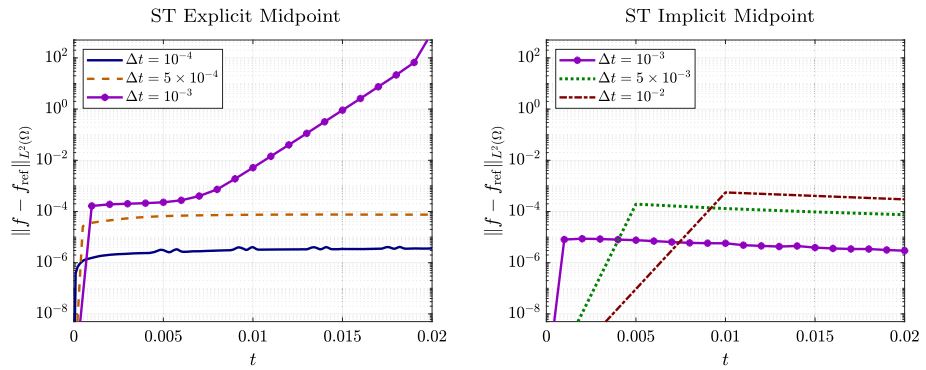


Fig. 5 Allen–Cahn equation (53). Comparison between the $L^2(\Omega)$ errors of explicit and implicit step-truncation midpoint methods for different Δt

Due to the smoothing properties of the Laplacian, the high frequencies in the initial condition quickly decay and, correspondingly, the rank drops significantly within the first few time steps. Due to the rapidly decaying rank for this problem, we have plotted it in log scale. In Fig. 5, we provide a comparison between the rank-adaptive implicit step-truncation midpoint method we propose here and the rank-adaptive explicit step-truncation midpoint method

$$f_{k+1} = \mathfrak{T}_{r_3} \left(f_k + \Delta t \mathfrak{T}_{r_2} \left(G \left(f_k + \frac{\Delta t}{2} \mathfrak{T}_{r_1} (G(f_k)) \right) \right) \right), \tag{56}$$

which was recently studied in [40]. The truncation ranks r_1 , r_2 , and r_3 time-dependent and satisfy the order conditions

$$\varepsilon_{r_3} \leq A \Delta t^3, \quad \varepsilon_{r_2} \leq B \Delta t^2, \quad \varepsilon_{r_1} \leq G \Delta t. \tag{57}$$

Such conditions guarantee that the scheme (56) is second-order convergent (see [40]). Figure 5 shows that the explicit step-truncation midpoint method undergoes a numerical instability for $\Delta t = 10^{-3}$. Indeed it is a conditionally stable method. The explicit step-truncation midpoint method also has other issues. In particular, in the rank-adaptive setting we consider here, we have that in the limit $\Delta t \rightarrow 0$ the parameters ε_{r_1} , ε_{r_2} and ε_{r_3} all go to zero (see Eq. (57)). This implies that the truncation operators may retain all singular values, henceforth maxing out the rank and thereby giving up all computational gains of low-rank tensor compression. On the other hand, if Δt is too large, then one we have stability issues as discussed above. Indeed, we see both these problems with the explicit step-truncation midpoint method, giving only a relatively narrow region of acceptable time step sizes in which the method is effective.

In Table 1 we provide a comparison between explicit and implicit step-truncation midpoint methods in terms of computational cost (CPU-time on an Intel Core i9-7980XE workstation) and accuracy at time $t = 1$. It is seen that the implicit step-truncation midpoint method is roughly 20 to 30 times faster than the explicit step-truncation midpoint method for a comparable error. Moreover, solutions with a large time step ($> 10^{-3}$) are impossible to achieve with the explicit step-truncation method due to time step restrictions associated with conditional stability.

Table 1 Allen–Cahn equation (53). Comparison between explicit and implicit step-truncation (ST) midpoint methods in terms of computational cost (CPU-time on an Intel Core I9-7980XE workstation) and accuracy at final time $t = 1$. It is seen that the implicit step-truncation midpoint method is roughly 20 to 30 times faster than the explicit step-truncation midpoint method for a comparable error

ST Explicit Midpoint		
Δt	Runtime (seconds)	$\ f - f_{\text{ref}}\ $
1.0×10^{-3}	Did not finish	Unstable
5.0×10^{-4}	2.2946×10^2	6.0713×10^{-3}
2.5×10^{-4}	4.7828×10^2	5.2628×10^{-4}
1.0×10^{-4}	1.2619×10^3	5.3648×10^{-5}
5.0×10^{-5}	2.7354×10^3	1.1723×10^{-5}
ST Implicit Midpoint		
Δt	Runtime (seconds)	$\ f - f_{\text{ref}}\ $
1.0×10^{-1}	5.3097	2.5652×10^{-2}
5.0×10^{-2}	1.0495×10^1	7.7248×10^{-3}
2.5×10^{-2}	1.8987×10^1	2.0977×10^{-3}
1.0×10^{-2}	3.8025×10^1	6.7477×10^{-5}
5.0×10^{-3}	7.6183×10^1	3.6012×10^{-6}

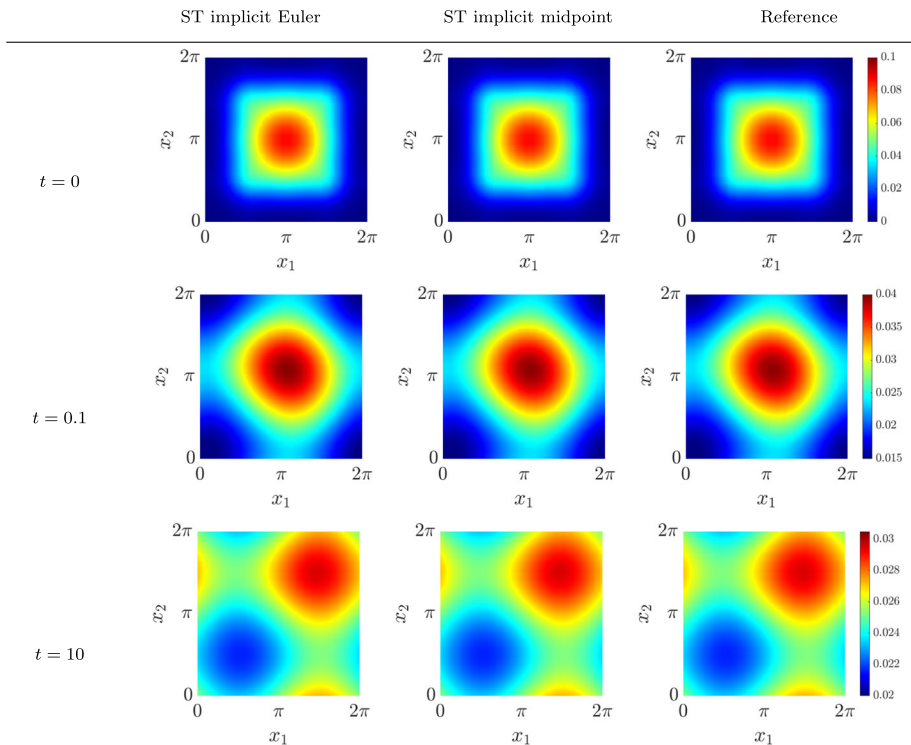


Fig. 6 Marginal probability density function (65) obtained by integrating numerically the Fokker–Planck equation (58) in dimension $d = 4$ with $\sigma = 5$ and initial condition (64) with two methods: i) rank-adaptive implicit step-truncation Euler and ii) rank-adaptive implicit step-truncation midpoint. The reference solution is a variable time step RK4 method with absolute tolerance of 10^{-14} . These solutions are computed on a grid with $20 \times 20 \times 20 \times 20$ interior points (evenly spaced). The steady state is determined for this computation by halting execution when $\|\partial f_{\text{ref}}/\partial t\|_2$ is below a numerical threshold of 10^{-8} . This happens at approximately $t \approx 10$ for the initial condition (64)

6.2 Fokker–Planck Equation

Consider the Fokker–Planck equation

$$\frac{\partial f(\mathbf{x}, t)}{\partial t} = - \sum_{i=1}^d \frac{\partial}{\partial x_i} (\mu_i(\mathbf{x})f(\mathbf{x}, t)) + \frac{\sigma^2}{2} \sum_{i=1}^d \frac{\partial^2 f(\mathbf{x}, t)}{\partial x_i^2} \tag{58}$$

on a four-dimensional ($d = 4$) flat torus $\Omega = [0, 2\pi]^4$. The components of the drift are chosen as

$$\mu_i(\mathbf{x}) = (\gamma(x_{i+1}) - \gamma(x_{i-2}))\xi(x_{i-1}) - \phi(x_i), \quad i = 1, \dots, d, \tag{59}$$

where $\gamma(x) = \sin(x)$, $\xi(x) = \exp(\sin(x)) + 1$, and $\phi(x) = \cos(x)$ are 2π -periodic functions. In (59) we set $x_{i+d} = x_i$. For this particular drift field, the right side of (58) can be split into a component tangential to the tensor manifold \mathcal{H}_r and a component that is non-tangential as

$$\begin{aligned} \frac{\partial f(\mathbf{x}, t)}{\partial t} = \sum_{i=1}^d \underbrace{\left(-\gamma(x_{i+1})\xi(x_{i-1}) \frac{\partial f(\mathbf{x}, t)}{\partial x_i} + \gamma(x_{i-2})\xi(x_{i-1}) \frac{\partial f(\mathbf{x}, t)}{\partial x_i} \right)}_{\text{Not tangential}} + \\ \underbrace{\left(\frac{\partial}{\partial x_i} \phi(x_i) f(\mathbf{x}, t) + \frac{\sigma^2}{2} \frac{\partial^2 f(\mathbf{x}, t)}{\partial x_i^2} \right)}_{\text{tangential}}. \end{aligned} \tag{60}$$

We solve (60) using an operator splitting method. To this end, we notice that there are $3d$ many terms in the summation above, and therefore we first solve the first d time dependent PDEs which are tangential to the tensor manifold \mathcal{H}_r , i.e.,

$$\frac{\partial g_i}{\partial t} = \frac{\partial}{\partial x_i} \phi(x_i) g_i + \frac{\sigma^2}{2} \frac{\partial^2 g_i}{\partial x_i^2}, \quad i = 1, \dots, d. \tag{61}$$

Then we solve the non-tangential equations in two batches,

$$\frac{\partial u_j}{\partial t} = \gamma(x_{i+1})\xi(x_{i-1}) \frac{\partial u_j}{\partial x_i}, \quad j = 1, \dots, d \tag{62}$$

$$\frac{\partial u_k}{\partial t} = \gamma(x_{i-2})\xi(x_{i-1}) \frac{\partial u_k}{\partial x_i}, \quad k = 2, \dots, 2d. \tag{63}$$

This yields the first-order (Lie-Trotter) approximation $f(\mathbf{x}, \Delta t) = u_{2d}(\mathbf{x}, \Delta t) + O(\Delta t^2)$. We also use these same list of PDEs for the second-order (Strang) splitting integrator. For each time step in both first- and second-order splitting methods, we terminate the HT/TT-GMRES iterations by setting the stopping tolerance $\epsilon_{\text{tol}} = 10^{-9}$. We set the initial probability density function (PDF) as

$$f_0(x_1, x_2, x_3, x_4) = \frac{1}{F_0} \sum_{j=1}^M \left(\prod_{i=1}^4 \frac{\sin((2j-1)x_i - \pi/2) + 1}{2^{2(j-1)}} + \prod_{i=1}^4 \frac{\exp(\cos(2jx_i + \pi))}{2^{2j-1}} \right), \tag{64}$$

where F_0 is a normalization constant. This gives an HTucker tensor with rank bounded by $2M$. We set $M = 10$ to give ranks bounded by 20. We discretize (58)–(64) in Ω with the

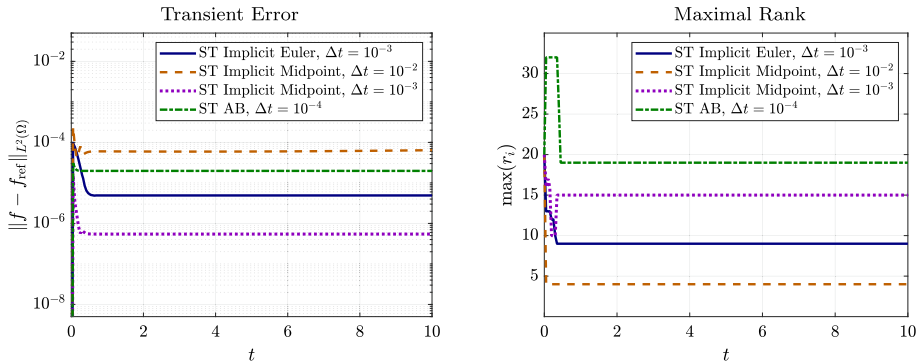


Fig. 7 $L^2(\Omega)$ error and rank versus time for numerical solutions of Fokker–Planck equation (58) in dimension $d = 4$ with initial condition (64). The rank plotted here is the largest rank for all tensors being used to represent the solution in HT format. Rank of the reference solution is in HT format

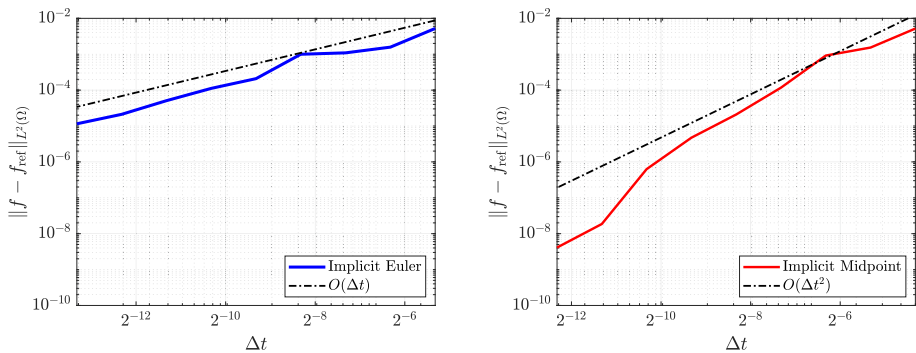


Fig. 8 $L^2(\Omega)$ errors at $t = 0.1$ for the implicit rank-adaptive step-truncation implicit Euler and midpoint methods versus Δt . The reference solution of (58) if computed using a variable time step RK4 method with absolute tolerance of 10^{-14}

Fourier pseudospectral collocation method [27] on a tensor product grid with $N = 20$ evenly-spaced points along each coordinate x_i , giving the total number of points $(N + 1)^4 = 194481$. This number corresponds to the number of entries in the tensor $f(t)$ appearing in equation (4). Also, we set $\sigma = 5$ in (58).

In Fig. 6 we compare a few time snapshots of the marginal PDF

$$f_{12}(x_1, x_2) = \int_0^{2\pi} \int_0^{2\pi} f(x_1, x_2, x_3, x_4) dx_3 dx_4, \tag{65}$$

we obtained with the rank-adaptive implicit step-truncation Euler and midpoint methods, as well as the reference marginal PDF. The solution very quickly relaxes to nearly uniform by $t = 0.1$, then slowly rises to its steady state distribution by $t = 10$.

In Fig. 7 we study accuracy and rank of the proposed implicit step-truncation methods in comparison with the rank-adaptive explicit Adams-Bashforth (AB) method of order 2 (see [40]). The implicit step-truncation methods use a time step size of $\Delta t = 10^{-3}$, while step-truncation AB2 uses a step size of $\Delta t = 10^{-4}$. The highest error and lowest rank come from the implicit step-truncation midpoint method with $\Delta t = 10^{-2}$. The highest rank and second highest error go to the step-truncation AB2 method, which runs with time step size 10^{-4} for

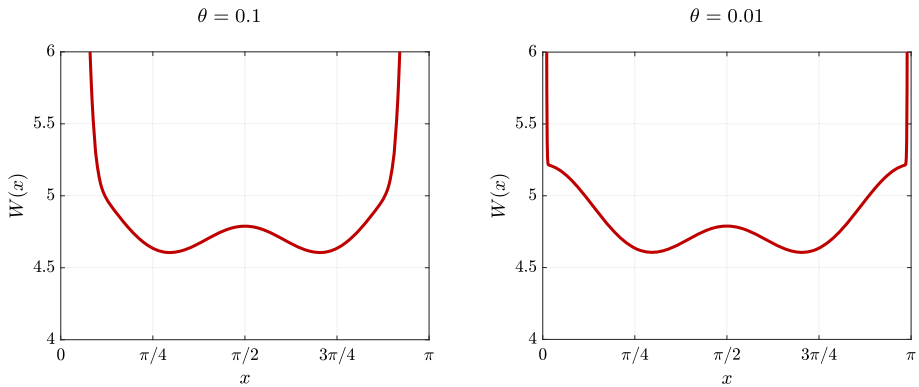


Fig. 9 Double-well potential (68)–(69) for different values of θ . It is seen that as $\theta \rightarrow 0$, the potential barrier at $x = 0$ and $x = \pi$ becomes infinitely high. This is identical to the well-known homogeneous boundary conditions for particles trapped in a box

stability. This causes a penalty in the rank, since the as the time step is made small, the rank must increase to maintain convergence order (see, e.g., (57) for similar conditions on explicit step-truncation midpoint method) The implicit step-truncation midpoint method performs the best, with error of approximately 10^{-6} and rank lower than the step-truncation AB2 method at steady state. Overall, the proposed implicit step-truncation methods perform extremely well on linear problems of this form, especially when the right hand side is explicitly written as a sum of tensor products of one dimensional operators. In Fig. (8) we show a plot of the convergence rate of implicit step-truncation Euler and midpoint methods. For this figure, we set $\sigma = 2$. The convergence rates are order one and order two respectively, verifying Theorem 1.

6.3 Nonlinear Schrödinger Equation

The nonlinear Schrödinger equation is complex-valued PDE whose main applications are wave propagation in nonlinear optical fibers and Bose-Einstein condensates [42, 44]. The equation can be written as²

$$\frac{\partial \phi(\mathbf{x}, t)}{\partial t} = \frac{i}{2} \Delta \phi(\mathbf{x}, t) - iV(\mathbf{x})\phi(\mathbf{x}, t) - i\varepsilon |\phi(\mathbf{x}, t)|^2 \phi(\mathbf{x}, t). \tag{67}$$

where $V(\mathbf{x})$ is the particle interaction potential. In our example, we consider 6 particles trapped on a line segment in the presence of a double-well potential defined as

$$V(\mathbf{x}) = 50 \sum_{k=1}^6 W(x_k), \quad W(x_k) = \left[1 + e^{\cos(x_k)^2} + \frac{3}{4} \left(1 + e^{\sin(x_k)^2} \right) \right] \eta_\theta(x_k). \tag{68}$$

² As is well-known, the nonlinear Schrödinger equation (67) is a Hamiltonian PDE which can be derived as a stationary point of the energy density (Hamilton’s functional)

$$H(\phi) = \int_{\Omega} \left(\frac{1}{4} \|\nabla \phi\|^2 + \frac{1}{2} V(\mathbf{x})|\phi|^2 + \frac{\varepsilon}{4} |\phi|^4 \right) d\mathbf{x}. \tag{66}$$

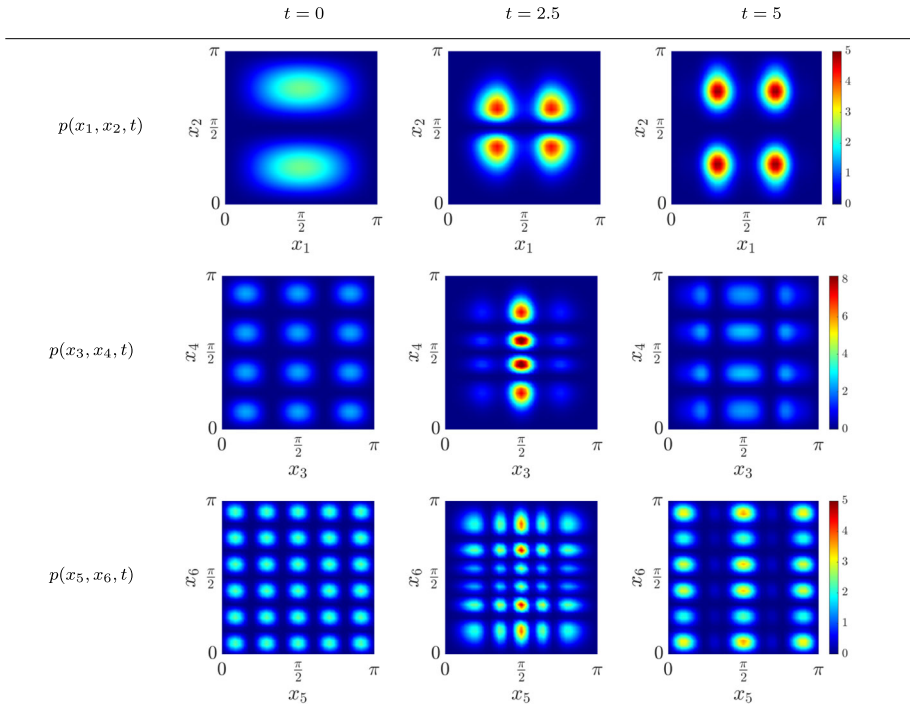


Fig. 10 Marginal probability density functions representing particle positions generated by the nonlinear Schrödinger equation (67) with $\varepsilon = 10^{-4}$, interaction potential (68) and initial condition (70)

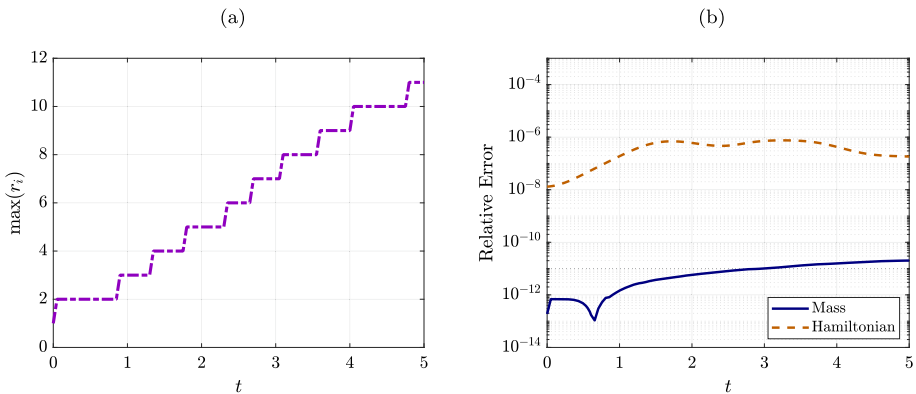


Fig. 11 **a** Maximum tensor rank versus time, and **b** relative error in the the solution mass and Hamiltonian (66) for nonlinear Schrödinger equation (67) in dimension $d = 6$, with $\varepsilon = 10^{-4}$, interaction potential (68) and initial condition (70)

Here, $W(x_k)$ is a potential with barriers at $x_k = 0$ and $x_k = \pi$ (see Fig.9). The function $\eta_\theta(x_i)$ is a mollifier which converges weakly to $1 + \delta(x_i) + \delta(x_i - \pi)$ as $\theta \rightarrow 0$. One such mollifier is

$$\eta_\theta(x) = 1 + \frac{1}{\sqrt{2\pi}\theta} \left(\exp \left[-\frac{x^2}{2\theta^2} \right] + \exp \left[-\frac{(x - \pi)^2}{2\theta^2} \right] \right). \tag{69}$$

As $\theta \rightarrow 0$, the weak limit of η_θ translates to zero Dirichlet boundary conditions on the domain $\Omega = [0, \pi]^6$. The Dirichlet conditions naturally allow us to use a discrete sine transform to compute the Laplacian’s differentiation matrices. We discretize the domain Ω on a uniform grid with 35 points per dimension. This gives us a tensor with $35^6 = 838265625$ entries, or 14.7 Gigabytes per temporal solution snapshot if we store the uncompressed tensor in a double precision IEEE 754 floating point format. We choose a product of pure states for our initial condition, i.e.,

$$\phi(\mathbf{x}, 0) = \prod_{k=1}^6 \frac{6^{1/6}4k}{2k\pi - \sin(2\pi k)} \sin(kx_k), \tag{70}$$

The normalizing constant $6^{1/6}4k/(2k\pi - \sin(2\pi k))$ guarantees that the wavefunction has an initial mass of 6 particles. We now apply an operator splitting method to solve (67). The linear components are all tangential to the tensor manifold \mathcal{H}_r , and have a physical interpretation. Specifically,

$$\frac{\partial g_k}{\partial t} = \frac{i}{2} \frac{\partial^2 g_k}{\partial x_k^2} - 50iW(x_k)g_k, \quad k = 1, \dots, 6, \tag{71}$$

is a sequence of one-dimensional linear Schrödinger equations. The non-tangential part reduces to

$$\frac{du}{dt} = i\varepsilon|u|^2u, \tag{72}$$

which may be interpreted as an ODE describing all pointwise interactions of the particles. Here we set $\varepsilon = 10^{-4}$ to model weak interactions. Clearly, the linear terms in (67) have purely imaginary eigenvalues. Therefore to integrate the semi-discrete form of (67) in time we need a numerical scheme that has the imaginary axis within its stability region. Since implicit step-truncation Euler method introduces a significant damping, thereby exacerbating inaccuracy due to discrete time stepping, we apply the implicit step-truncation midpoint method. For this problem, we set $\Delta t = 5 \times 10^{-2}$ and the tensor truncation error to be constant in time at $100\Delta t^3$ to maintain second-order consistency. Tolerance of the inexact Newton method was set to 5×10^{-5} and the HT/TT-GMRES relative error to $5 \times \eta = 10^{-4}$. In Fig. 10 we plot the time-dependent marginal probability density functions for the joint position variables (x_k, x_{k+1}) , $k = 1, 3, 5$. Such probability densities are defined as

$$p(x_1, x_2, t) = \frac{1}{6} \int_{[0,1]^4} \phi^*(\mathbf{x}, t)\phi(\mathbf{x}, t)dx_3dx_4dx_5dx_6, \tag{73}$$

and analogously for $p(x_3, x_4, t)$ and $p(x_5, x_6, t)$. It is seen that the lower energy pure states (position variables (x_1, x_2)) quickly get trapped in the two wells, oscillating at their bottoms. Interestingly, at $t = 2.5$ it appears that particle x_3 is most likely to be observed in between the two wells whenever the particle x_4 is in a well bottom.

In Fig. 11a, we plot the rank over time for this problem. The rank also has physical meaning. A higher rank HT tensor is equivalent to a wavefunction with many entangled states, regardless of which $L^2(\Omega)$ basis we choose. In the example discussed in this section, the particles interacting over time monotonically increase the rank. We emphasize that the nonlinearity in (67) poses a significant challenge to tensor methods. In fact, in a single application of the function $i\varepsilon|u|^2u$, we may end up cubing the rank. This can be mitigated somewhat by using the approximate element-wise tensor multiplication routine. Even so, if the inexact Newton Method requires many dozens of iterations to halt, the rank may grow

very rapidly in a single time step, causing a slowing due to large array storage. This problem is particularly apparent when $\varepsilon \approx 1$ or if Δt is made significantly smaller, e.g. $\Delta t = 10^{-4}$. A more effective way of evaluating nonlinear functions on tensors decompositions would certainly mitigate this issue. In Fig. 11b, we plot the relative error of the solution mass and the Hamiltonian (66) over time. The relative errors hover around 10^{-11} and 10^{-6} , respectively. It is remarkable that even though the additional tensor truncation done after the inner loop of the implicit solver in principle destroys the symplectic properties of the midpoint method, the mass and Hamiltonian are still preserved with high accuracy.

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 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.

Declarations

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

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

A Solving Algebraic Equations On Tensor Manifolds

We have seen in Sect. 3 that a large class of implicit step-truncation methods can be equivalently formulated as a root-finding problem for a nonlinear system algebraic equations of the form

$$\mathbf{H}(\mathbf{f}) = \mathbf{0} \quad (74)$$

at each time step. In this appendix we develop numerical algorithms to compute an approximate solution of (74) on a tensor manifold \mathcal{H}_r with a given rank r . In other words, we are interested in finding $\mathbf{f} \in \mathcal{H}_r$ that solves (74) with controlled accuracy. To this end, we combine the inexact inexact Newton method [17, Theorem 2.3 and Corollary 3.5] with the TT-GMRES linear solver proposed in [20].

Theorem 3 (Inexact Newton method [17]) *Let $\mathbf{H} : \mathbb{R}^N \rightarrow \mathbb{R}^N$ be continuously differentiable in a neighborhood of a zero \mathbf{f}^* , and suppose that the Jacobian of \mathbf{H} , i.e., $\mathbf{J}_{\mathbf{H}}(\mathbf{f}) = \partial \mathbf{H}(\mathbf{f}) / \partial \mathbf{f}$, is invertible at \mathbf{f}^* . Given $\mathbf{f}^{[0]} \in \mathbb{R}^N$, consider the sequence*

$$\mathbf{f}^{[j+1]} = \mathbf{f}^{[j]} + \mathbf{s}^{[j]}, \quad j = 0, 1, \dots \quad (75)$$

where each $s^{[j]}$ solves the Newton iteration up to relative error $\eta^{[j]}$, i.e., it satisfies

$$\left\| \mathbf{J}_H \left(\mathbf{f}^{[j]} \right) s^{[j]} + \mathbf{H} \left(\mathbf{f}^{[j]} \right) \right\| \leq \left\| \mathbf{H} \left(\mathbf{f}^{[j]} \right) \right\| \eta^{[j]}. \tag{76}$$

If $\eta^{[j]} < 1$ for all j , then there exists $\varepsilon > 0$ so that for any initial guess satisfying $\left\| \mathbf{f}^{[0]} - \mathbf{f}^* \right\| < \varepsilon$ the sequence $\{ \mathbf{f}^{[j]} \}$ converges linearly to \mathbf{f}^* . If $\eta^{[j]} \rightarrow 0$ as $j \rightarrow \infty$, then the convergence speed is superlinear.

The next question is how to compute an approximate solution of the linear system

$$\mathbf{J}_H \left(\mathbf{f}^{[j]} \right) s^{[j]} = -\mathbf{H} \left(\mathbf{f}^{[j]} \right) \tag{77}$$

satisfying the bound (76), without inverting the Jacobian $\mathbf{J}_H \left(\mathbf{f}^{[j]} \right)$ and assuming that $s^{[j]} \in \mathcal{H}_{\mathbf{r}_j}$, i.e., that $s^{[j]}$ is a tensor with rank \mathbf{r}_j . To this end we utilize the relaxed HT/TT-GMRES method discussed in [20] HT/TT-GMRES is an adapted tensor-structured generalized minimal residual (GMRES) method to solve linear systems in a tensor format. The solver employs an indirect accuracy check and a stagnation restart check in its halting criterion which we summarize in the following Lemma.

Lemma 1 (Accuracy of HT/TT-GMRES [20]) *Let $\mathbf{J} \mathbf{f} = \mathbf{b}$ be a linear system where \mathbf{f}, \mathbf{b} are tensors in HT or TT format, and \mathbf{J} is a bounded linear operator on \mathbf{f} . Let $\{ \mathbf{f}^{[0]}, \mathbf{f}^{[1]}, \dots \}$ be the sequence of approximate solutions generated by HT/TT-GMRES algorithm in [20], and $\varepsilon > 0$ be the stopping tolerance for the iterations. Then*

$$\left\| \mathbf{J} \mathbf{f}^{[j]} - \mathbf{b} \right\| \leq m \left\| \mathbf{J} \right\| \left\| \mathbf{J}^{-1} \right\| \left\| \mathbf{b} \right\| \varepsilon, \tag{78}$$

where m is the number of Krylov iterations performed before restart. Similarly, the distance between $\mathbf{f}^{[j]}$ and the exact solution \mathbf{f} can be bounded as

$$\left\| \mathbf{f}^{[j]} - \mathbf{f} \right\| \leq m \left\| \mathbf{J} \right\| \left\| \mathbf{J}^{-1} \right\|^2 \left\| \mathbf{b} \right\| \varepsilon. \tag{79}$$

We can now combine the HT/TT-GMRES linear solver with the inexact Newton method, to obtain an algorithm that allows us to solve nonlinear algebraic equations of the form (74) on a tensor manifold.

Theorem 4 (HT/TT Newton method) *Let $\mathbf{H} : \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d} \rightarrow \mathbb{R}^{n_1 \times n_2 \times \dots \times n_d}$ be a continuously differentiable nonlinear map which operates on HT or TT tensor formats, and let \mathbf{f}^* be a zero of \mathbf{H} . Suppose that the Jacobian of \mathbf{H} , denoted as $\mathbf{J}_H(\mathbf{f})$, is invertible at \mathbf{f}^* . Given an initial guess $\mathbf{f}^{[0]}$, consider the iteration*

$$\mathbf{f}^{[j+1]} = \mathfrak{T}_r \left(\mathbf{f}^{[j]} + s^{[j]} \right) \tag{80}$$

where $s^{[j]}$ is the HT/TT-GMRES solution of $\mathbf{J}_H \left(\mathbf{f}^{[j]} \right) s^{[j]} = -\mathbf{H} \left(\mathbf{f}^{[j]} \right)$ satisfying

$$\left\| \mathbf{J}_H \left(\mathbf{f}^{[j]} \right) s^{[j]} + \mathbf{H} \left(\mathbf{f}^{[j]} \right) \right\| \leq \frac{1}{2} \left\| \mathbf{H} \left(\mathbf{f}^{[j]} \right) \right\| \eta^{[j]}, \tag{81}$$

where $\eta^{[j]}$ is the relative error, which can be any value in the range $0 \leq \eta^{[j]} < 1$. Then the Newton iteration converges linearly so long as the rank \mathbf{r} of the truncation operator \mathfrak{T}_r is chosen to satisfy

$$\left\| \mathfrak{T}_r \left(\mathbf{f}^{[j]} + s^{[j]} \right) - \mathbf{f}^{[j]} - s^{[j]} \right\| \leq \frac{\left\| \mathbf{H} \left(\mathbf{f}^{[j]} \right) \right\|}{2 \left\| \mathbf{J}_H \left(\mathbf{f}^{[j]} \right) \right\|} \eta^{[j]}. \tag{82}$$

Proof Let $f^{[0]}$ be an initial guess. Consider the sequence

$$\tilde{f}^{[j+1]} = f^{[j]} + s^{[j]} \quad j = 0, 1, \dots \tag{83}$$

where $s^{[j]}$ is the HT/TT-GMRES solution of $J_H(f^{[j]})s^{[j]} = -H(f^{[j]})$ obtained with tolerance

$$\varepsilon_j < \frac{\eta^{[j]}}{2m \|J_H(f^{[j]})\| \|J_H^{-1}(f^{[j]})\|}, \tag{84}$$

and $0 \leq \eta^{[j]} < 1$. The next step is to truncate $\tilde{f}^{[j+1]}$ to a tensor

$$f^{[j+1]} = \mathfrak{T}_r(\tilde{f}^{[j+1]}) \tag{85}$$

with rank r chosen so that

$$\|f^{[j+1]} - \tilde{f}^{[j+1]}\| < \frac{\|H(f^{[j]})\|}{2 \|J_H(f^{[j]})\|} \eta^{[j]}. \tag{86}$$

Then the error in solving the Newton iteration this step is

$$\tilde{r}^{[j]} = J_H(f^{[j]}) [f^{[j+1]} - \tilde{f}^{[j+1]} + s^j] + H(f^{[j]}), \tag{87}$$

which we constructed to satisfy the bound

$$\|\tilde{r}^{[j]}\| \leq \|J_H(f^{[j]})s^{[j]} + H(f^{[j]})\| + \|J_H(f^{[j]}) [f^{[j+1]} - \tilde{f}^{[j+1]})\| < \|H(f^{[j]})\| \eta^{[j]}. \tag{88}$$

Hence, the inexact HT/TT-GMRES Newton method converges linearly. This completes the proof. □

An Example

Let us provide a brief numerical demonstration of the inexact Newton method with HT/TT-GMRES iteration. To this end, consider the cubic function

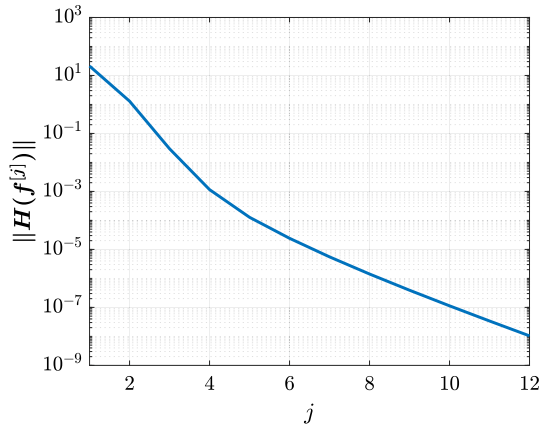
$$H(f) = 1.5f + 0.5f_0 + 0.125(f + f_0)^3 \tag{89}$$

where all products are computed using the approximate element-wise Hadamard tensor product with accuracies set to 10^{-12} , and f_0 is a given tensor which corresponds to the initial condition used in Sect. 6.1 truncated to an absolute tolerance of 10^{-4} . The Jacobian operator is easily obtained as

$$J_H(f)s = 1.5s + 0.375(f + f_0)^2s. \tag{90}$$

We set the relative error of the matrix inverse to be $\eta = 10^{-3}$. In Fig. 12 we plot the results of the proposed inexact Newton method with HT/TT-GMRES iterations. We see that that the target tolerance of 2.2×10^{-8} is hit in just 12 iterations.

Fig. 12 Error versus iteration count of inexact Newton's method in the HT format



References

- Allen, S.M., Cahn, J.W.: Ground state structures in ordered binary alloys with second neighbor interactions. *Acta Metall.* **20**(3), 423–433 (1972)
- Allen, S.M., Cahn, J.W.: A correction to the ground state of FCC binary ordered alloys with first and second neighbor pairwise interactions. *Scr. Metall.* **7**(12), 1261–1264 (1973)
- Austin, W., Ballard, G., Kolda, T.G.: Parallel tensor compression for large-scale scientific data. In *IPDPS'16: Proceedings of the 30th IEEE International Parallel and Distributed Processing Symposium*, pp. 912–922. (2016)
- 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), 1423–1472 (2016)
- 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)
- Barthelmann, V., Novak, E., Ritter, K.: High dimensional polynomial interpolation on sparse grids. *Adv. Comput. Mech.* **12**, 273–288 (2000)
- Beran, M.J.: *Statistical Continuum Theories*. Interscience Publishers, New York (1968)
- Boelens, A.M.P., Venturi, D., Tartakovsky, D.M.: Parallel tensor methods for high-dimensional linear PDEs. *J. Comput. Phys.* **375**, 519–539 (2018)
- Boelens, A.M.P., Venturi, D., Tartakovsky, D.M.: Tensor methods for the Boltzmann-BGK equation. *J. Comput. Phys.* **421**, 109744 (2020)
- Bungartz, H.J., Griebel, M.: Sparse grids. *Acta Numer.* **13**, 147–269 (2004)
- Cao, Y., Chen, Z., Gunzburger, M.: ANOVA expansions and efficient sampling methods for parameter dependent nonlinear PDEs. *Int. J. Numer. Anal. Model.* **6**, 256–273 (2009)
- Chen, Y., Zhang, L., Wang, H., Weinan, E.: Ground state energy functional with Hartree–Fock efficiency and chemical accuracy. *J. Phys. Chem. A* **124**(35), 7155–7165 (2020)
- Cho, H., Venturi, D., Karniadakis, G.E.: Numerical methods for high-dimensional probability density function equations. *J. Comput. Phys.* **305**, 817–837 (2016)
- Al Daas, H., Ballard, G., Benner, P.: Parallel algorithms for tensor train arithmetic. *SIAM J. Sci. Comput.* **44**(1), C25–C53 (2022)
- Dektor, A., Venturi, D.: Dynamically orthogonal tensor methods for high-dimensional nonlinear PDEs. *J. Comput. Phys.* **404**, 109125 (2020)
- Dektor, A., Venturi, D.: Dynamic tensor approximation of high-dimensional nonlinear PDEs. *J. Comput. Phys.* **437**, 110295 (2021)
- Dembo, R.S., Eisenstat, S.C., Steihaug, T.: Inexact newton methods. *SIAM J. Numer. Anal.* **19**(2), 400–408 (1982)
- Di Marco, G., Pareschi, L.: Numerical methods for kinetic equations. *Acta Numer.* **23**, 369–520 (2014)
- Dolgov, S., Khoromskij, B., Oseledets, I.: 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**(6), A3016–A3038 (2012)

20. Dolgov, S.V.: TT-GMRES: solution to a linear system in the structured tensor format. *Russ. J. Numer. Anal. Math. Model.* **28**(2), 149–172 (2013)
21. Weinan, E., Han, J., Li, Q.: A mean-field optimal control formulation of deep learning. *Res. Math. Sci.* **6**(10), 1–41 (2019)
22. Faragó, I., Havasiy, A.: *Operator Splittings and Their Applications*. Nova Science Publishers (2009)
23. Foo, J., Karniadakis, G.E.: Multi-element probabilistic collocation method in high dimensions. *J. Comput. Phys.* **229**, 1536–1557 (2010)
24. Grasedyck, L.: Hierarchical singular value decomposition of tensors. *SIAM J. Matrix Anal. Appl.* **31**(4), 2029–2054 (2010)
25. Grasedyck, L., Löbber, C.: Distributed hierarchical SVD in the hierarchical Tucker format. *Numer. Linear Algebra Appl.* **25**(6), e2174 (2018)
26. Hairer, E., Lubich, C., Wanner, G.: *Geometric Numerical Integration*, volume 31 of Springer Series in Computational Mathematics. 2nd edn, Springer-Verlag, Berlin, (2006). Structure-preserving algorithms for ordinary differential equations
27. Hesthaven, J.S., Gottlieb, S., Gottlieb, D.: *Spectral Methods for Time-dependent Problems*. Cambridge University Press (2007)
28. Hopf, E.: Statistical hydromechanics and functional calculus. *J. Rat. Mech. Anal.* **1**(1), 87–123 (1952)
29. Kassam, A.-K., Trefethen, L.N.: Fourth-order time stepping for stiff PDEs. *SIAM J. Sci. Comput.* **26**(4), 1214–1233 (2005)
30. 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)
31. Kieri, E., Vandereycken, B.: Projection methods for dynamical low-rank approximation of high-dimensional problems. *Comput. Methods Appl. Math.* **19**(1), 73–92 (2019)
32. Kressner, D., Tobler, C.: Algorithm 941: htucker - a Matlab toolbox for tensors in hierarchical Tucker format. *ACM Trans. Math. Softw.* **40**(3), 1–22 (2014)
33. Li, G., Rabitz, H.: Regularized random-sampling high dimensional model representation (RS-HDMR). *J. Math. Chem.* **43**(3), 1207–1232 (2008)
34. Li, L., Qiu, J., Russo, G.: A high order semi-Lagrangian finite difference method for nonlinear Vlasov and BGK models. *Comm. Appl. Math. Comput.* **5**, 170–198 (2023)
35. Montanelli, H., Nakatsukasa, Y.: Fourth-order time-stepping for stiff PDEs on the sphere. *SIAM J. Sci. Comput.* **40**(1), A421–A451 (2018)
36. 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)
37. Raissi, M., Karniadakis, G.E.: Hidden physics models: machine learning of nonlinear partial differential equations. *J. Comput. Phys.* **357**, 125–141 (2018)
38. 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)
39. Risken, H.: *The Fokker-Planck Equation: Methods of Solution and Applications*, 2nd edn. Springer-Verlag (1989)
40. Rodgers, A., Dektor, A., Venturi, D.: Adaptive integration of nonlinear evolution equations on tensor manifolds. *J. Sci. Comput.* **92**(39), 1–31 (2022)
41. Rodgers, A., Venturi, D.: Stability analysis of hierarchical tensor methods for time-dependent PDEs. *J. Comput. Phys.* **409**, 109341 (2020)
42. Rudy, S.H., Brunton, S.L., Proctor, J.L., Kutz, J.N.: Data-driven discovery of partial differential equations. *Sci. Adv.* **3**(4), e1602614 (2017)
43. Shi, T., Ruth, M., Townsend, A.: Parallel algorithms for computing the tensor-train decomposition. pp. 1–23. [arXiv:2111.10448](https://arxiv.org/abs/2111.10448) (2021)
44. Trombettoni, A., Smerzi, A.: Discrete solitons and breathers with dilute Bose–Einstein condensates. *Phys. Rev. Lett.* **86**, 2353–2356 (2001)
45. Uschmajew, A.: Local convergence of the alternating least squares algorithm for canonical tensor approximation. *SIAM J. Matrix Anal. Appl.* **33**(2), 639–652 (2012)
46. Uschmajew, A., Vandereycken, B.: The geometry of algorithms using hierarchical tensors. *Linear Algebra Appl.* **439**(1), 133–166 (2013)
47. Venturi, D.: The numerical approximation of nonlinear functionals and functional differential equations. *Phys. Rep.* **732**, 1–102 (2018)
48. Venturi, D., Dektor, A.: Spectral methods for nonlinear functionals and functional differential equations. *Res. Math. Sci.* **8**(27), 1–39 (2021)

49. 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)
50. Zinn-Justin, J.: *Quantum Field Theory and Critical Phenomena*, 4th edn. Oxford Univ. Press (2002)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.