Electronic Transactions on Numerical Analysis.
Volume 62, pp. 163–187, 2024.

ISSN 1068–9613.
DOI: 10.1553/etna_vol62s163

ETNA
Kent State University and
Johann Radon Institute (RICAM)

A NOTE ON TT-GMRES FOR THE SOLUTION OF PARAMETRIC LINEAR SYSTEMS*

OLIVIER COULAUD†, LUC GIRAUD†, AND MARTINA IANNACITO‡

Abstract. We study the solution of linear systems with tensor product structure using the Generalized Minimal RESidual (GMRES) algorithm. To manage the computational complexity of high-dimensional problems, our approach relies on low-rank tensor representation, focusing specifically on the Tensor Train format. We implement and experimentally study the TT-GMRES algorithm. Our analysis bridges the heuristic methods proposed for TT-GMRES by Dolgov [Russian J. Numer. Anal. Math. Modelling, 28 (2013), pp. 149–172] and the theoretical framework of inexact GMRES by Simoncini and Szyld [SIAM J. Sci. Comput. 25 (2003), pp. 454–477]. This approach is particularly relevant in a scenario where a $(d+1)$-dimensional problem arises from concatenating a sequence of d-dimensional problems, as in the case of a parametric linear operator or parametric right-hand-side formulation. Thus, we provide backward error bounds that link the accuracy of the computed $(d+1)$-dimensional solution to the numerical quality of the extracted d-dimensional solutions. This facilitates the prescription of a convergence threshold ensuring that the d-dimensional solutions extracted from the $(d+1)$-dimensional result have the desired accuracy once the solver converges. We illustrate these results with academic examples across varying dimensions and sizes. Our experiments indicate that TT-GMRES retains the theoretical rounding-error properties observed in matrix-based GMRES.

Key words. GMRES, inexact GMRES, backward stability, Tensor Train format

AMS subject classifications. 65F10, 15A69, 65G50

1. Introduction.

In numerous scientific and engineering domains, mathematical models often involve solving d-dimensional linear systems with a tensor product structure. Such systems can be represented as

$$\mathbf{A}\mathbf{x} = \mathbf{b},$$

where $\mathbf{A}$ is a multilinear operator acting on $\mathbb{R}^{n_1 \times \cdots \times n_d}$, the tensor $\mathbf{b}$ represents the right-hand side, and the tensor $\mathbf{x}$ is the sought solution. As the number of dimensions d increases, the storage and computational costs grow exponentially — this phenomenon is commonly referred to as the "curse of dimensionality". Addressing these challenges requires algorithms that balance accuracy with tractable computational and memory demands.

Two main strategies have emerged for solving high-dimensional linear systems, one arising from optimization and one from the numerical linear algebra domain. The first approach is based on optimization methods. It includes the Alternating Linearized Scheme, the Modified Alternating Linearized Scheme [16], the Alternating Minimal Energy method [8], and the Density Matrix Renormalization Group approach [21]. These techniques break down the high-dimensional system into a sequence of lower-dimensional minimization subproblems, iteratively updating the solution. The second strategy extends iterative solvers from classical matrix computations, such as the conjugate gradient, the Generalized Minimal RESidual (GMRES), and the biconjugate gradient method [31], to high-dimensional spaces, introducing tensors and multilinear operators.

This second class of methods uses well-established techniques within a tensor context, generalizing key properties and heuristics to high-dimensional linear systems, cf. [2, 7, 19].

*Received January 31, 2024. Accepted December 31, 2024. Published online on January 27, 2025. Recommended by Lars Grasedyck.
†Concace, Inria Center at the University of Bordeaux, France.
O. Coulaud: ORCID: 0000-0003-2924-284X; L. Giraud: ORCID: 0000-0002-7062-7672.
‡Corresponding author. Dipartimento di Matematica and (AM)^2, Alma Mater Studiorum Università di Bologna, Piazza di Porta San Donato 5, I-40127 Bologna, Italy (martina.iannacito@unibo.it).
ORCID: 0000-0003-3354-2538

ETNA
Kent State University and
Johann Radon Institute (RICAM)

High-dimensional problems pose challenges due to their computational demands and the prohibitive storage requirements of dense tensors, even in moderate dimensions. To address these issues, compression techniques such as High Order Singular Value Decomposition [6], Hierarchical Tucker [11], Tensor Train (TT) [22] and Tensor Network [20] decompositions are used. Among these, the TT format has gained particular attention due to its flexibility and efficiency in handling high-dimensional tensors.

While tensor compression effectively reduces storage requirements, it introduces rounding errors that affect numerical computations, particularly for iterative solvers that heavily rely on compression. Balancing the trade-off between maintaining low ranks and achieving the desired level of accuracy is fundamental when developing an iterative solver. Assessing and controlling the propagation of rounding errors in iterative solvers has thus become a critical component of numerical analysis for high-dimensional problems.

This work focuses on the analysis of GMRES with the Modified Gram–Schmidt orthogonalization kernel (MGS-GMRES) algorithm adapted to the Tensor Train format (TT-GMRES) for high-dimensional linear systems. Our TT-GMRES algorithm incorporates tensor compression at various steps of the iterative process, raising important questions about the stability and accuracy of the computed solutions. The first theoretical demonstration that MGS-GMRES is backward stable dates back to 2006. In [24], the authors analyse MGS-GMRES in the standard IEEE arithmetic. The fundamental assumptions are that the unit round-off u bounds both the data representation and the rounding error of all the elementary floating-point operations. In [1], the authors consider a variable-accuracy framework for studying experimentally MGS-GMRES. In this context, the data storage precision is decoupled from the unit round-off that controls the rounding of floating-point operations. Additionally, it is assumed that the data storage precision is independent of the hardware and that the perturbation on the data is norm-wise bounded. Under these working hypotheses, they experimentally show that the backward stability of MGS-GMRES holds. Building on the theoretical backward stability results of [24] for MGS-GMRES in classical matrix computation, we examine our TT-GMRES within the variable accuracy framework. Specifically, this study experimentally investigates the interplay between tensor compression, inexact arithmetic and backward error analysis, linking these aspects to ensure robust performance in the tensor setting. Our TT-GMRES approach is compared with the heuristic TT-GMRES variant proposed in [7]. Additionally, we theoretically justify the heuristic proposed in [7] for TT-GMRES and link that variant with the theory of inexact GMRES properties, presented in [30]. Our experimental examples emphasize that TT-GMRES from [7] inherits the numerical features of the inexact GMRES variant. Furthermore, we investigate the relationship between TT-GMRES and the block-GMRES variant.

Additionally, we provide backward error bounds that relate the quality of the computed $(d+1)$-dimensional solutions to the accuracy of the d-dimensional solutions extracted from them. This analysis is particularly relevant for parametric problems, which involve efficiently solving a sequence of d-dimensional problems by utilizing the tensor structure in a $(d+1)$-dimensional space. The theoretical findings are supported by numerical experiments showcasing the effectiveness of TT-GMRES on academic problems of varying dimensions and sizes. These experiments highlight that TT-GMRES can achieve accurate solutions while preserving the memory-efficient benefits of tensor compression. Furthermore, they confirm that TT-GMRES inherits desirable numerical stability properties similar to its matrix-based counterpart.

The remaining sections of this paper are organized as follows. Section 2 provides the necessary background on tensors and their representation in TT format, along with the formulation of parametric problems. Section 3 introduces GMRES and its tensor variants,

ETNA
Kent State University and
Johann Radon Institute (RICAM)

detailing the algorithmic structure of TT-GMRES. Section 4 establishes theoretical connections between inexact GMRES and the heuristics used for TT-GMRES from [7]. It presents the backward error bounds for parametric systems. Section 5 includes numerical experiments illustrating the algorithm's performance and its application to high-dimensional test cases. Finally, Section 6 offers concluding remarks and directions for future research.

2. Preliminaries on tensors and parametric problems. To enhance readability, we utilize the following notation for the various mathematical objects described. Small Latin letters represent scalars and vectors (e.g., a), with the context clarifying the objects' nature. Matrices are represented by capital Latin letters (e.g., A), tensors by bold small Latin letters (e.g., $\mathbf{a}$), the multilinear operators between two tensor spaces are denoted by bold calligraphic capital letter (e.g., $\boldsymbol{\mathcal{A}}$), and the tensor representation of linear operators by bold capital Latin letters (e.g., $\mathbf{A}$). We use the "MATLAB notation", that is, we denote all the indices along a mode with a colon (":"). For example, if we are given a matrix $A \in \mathbb{R}^{m\times n}$, then $A(:,i)$ represents the ith column of A. The tensor product is denoted by $\otimes$ and the Kronecker product by $\otimes_{\mathrm{K}}$. The Euclidean inner product is denoted by $\langle\cdot,\cdot\rangle$ for both vectors and tensors. We use $\|\cdot\|$ to denote the Euclidean norm for vectors and the Frobenius norm for matrices and tensors. A linear operator $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n_1\times\cdots\times n_d} \to \mathbb{R}^{n_1\times\cdots\times n_d}$ between tensor spaces is represented by a tensor $\mathbf{A} \in \mathbb{R}^{(n_1\times n_1)\times\cdots\times(n_d\times n_d)}$ with respect to the canonical basis. The L2 norm of the linear operator $\boldsymbol{\mathcal{A}}$ is denoted by $\|\mathbf{A}\|_2$. If $d = 2$, then the L2 norm of the matrix associated with a simpler linear operator between two linear vector spaces is considered.

Section 2.1 describes the main key elements of the Tensor Train (TT) notation for tensors and linear operators between tensor product of spaces. The advantages of using this formalism to solve linear systems naturally defined in high-dimensional vector spaces are also presented. Section 2.2 examines the scenario where one of the linear operator modes is associated with a parameter. When dealing with parametric linear operators, our focus is on solving a single linear system for all discrete parameter values in TT format. Section 2.3 presents the scenario in which the right-hand sides depend on a parameter. We describe the construction of a unique linear system in TT format when there are multiple right-hand sides depending on a parameter.

2.1. The Tensor Train format. Let $\mathbf{x}$ be a d-order tensor in $\mathbb{R}^{n_1\times\cdots\times n_d}$ and n_k the dimension of mode k for every $k \in \{1,\dots,d\}$. Storing the full tensor $\mathbf{x} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ has a memory cost of $\mathcal{O}(n^d)$ with $n = \max_{i\in\{1,\dots,d\}}\{n_i\}$, so several compression techniques have been proposed over the years to reduce the memory consumption [6, 11, 22]. For this work, the most suitable tensor representation is the *Tensor Train* (TT) format [22]. The main concept of TT is to represent a d-order tensor as the contraction of d 3-order tensors. This contraction is a generalization of the matrix–vector product to tensors.

The Tensor Train representation of $\mathbf{x} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ is

$$\mathbf{x} = \underline{\mathbf{x}}_1\underline{\mathbf{x}}_2\cdots\underline{\mathbf{x}}_d,$$

where $\underline{\mathbf{x}}_k \in \mathbb{R}^{r_{k-1}\times n_k\times r_k}$ is the kth *TT core* for $k \in \{1,\dots,d\}$, with $r_0 = r_d = 1$. Note that $\underline{\mathbf{x}}_1 \in \mathbb{R}^{r_0\times n_1\times r_1}$ and $\underline{\mathbf{x}}_d \in \mathbb{R}^{r_{d-1}\times n_d\times r_d}$ reduce essentially to matrices, but for consistency in notation, we represent them as tensors. The kth TT core of a tensor is denoted by the same bold letter underlined, with a subscript k. The value r_k is called the *kth TT rank*.

Given an index i_k, we denote the i_kth matrix slice of the kth TT core $\underline{\mathbf{x}}_k$ with respect to mode 2 by $\underline{X}_k(i_k)$, i.e., $\underline{X}_k(i_k) = \underline{\mathbf{x}}_k(:,i_k,:)$. Each element of the TT tensor $\mathbf{x}$ can be expressed as the product of d matrices, that is,

$$\mathbf{x}(i_1,\dots,i_d) = \underline{X}_1(i_1)\cdots\underline{X}_d(i_d),$$

with $\underline{X}_k(i_k) \in \mathbb{R}^{r_{k-1}\times r_k}$ for every $i_k \in \{1,\dots,n_k\}$ and $k \in \{2,\dots,d-1\}$, while $\underline{X}_1(i_1) \in \mathbb{R}^{1\times r_1}$ and $\underline{X}_d(i_d) \in \mathbb{R}^{r_{d-1}\times 1}$. It is important to note that $\underline{X}_1(i_1)$ and $\underline{X}_d(i_d)$ are actually

ETNA
Kent State University and
Johann Radon Institute (RICAM)

vectors, but for the sake of consistency, they are written as matrices with a single row or column. TT-format tensors are called TT vectors.

In order to store a tensor in TT format, $\mathcal{O}(dnr^2)$ units of memory are required, where $n = \max_{i\in\{1,\dots,d\}}\{n_i\}$ and $r = \max_{i\in\{1,\dots,d\}}\{r_i\}$. The memory footprint grows linearly with the tensor order and quadratically with the maximal TT rank. Therefore, knowing the maximal TT rank is usually sufficient to estimate the TT compression benefit. However, for more accuracy, we introduce the compression ratio measure. If $\mathbf{x} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ is a tensor in TT format, the compression ratio is the storage cost of $\mathbf{x}$ in TT format divided by the storage cost in dense format, i.e.,

$$\frac{\sum_{i=1}^{d} r_{i-1}n_i r_i}{\prod_{j=1}^{d} n_j},$$

where r_i is the ith TT rank of $\mathbf{x}$. The TT ranks, r_i, must remain small to achieve a significant benefit from this formalism. One drawback of the TT format is that it may become less efficient when adding two TT vectors. Given two TT vectors $\mathbf{x}$ and $\mathbf{y}$ with kth TT ranks r_k and s_k, respectively, the kth TT rank of $\mathbf{x}+\mathbf{y}$ is less than or equal to $r_k + s_k$ (see [9]).

The TT formalism allows for the compressed expression of linear operators between tensor product spaces. Given a linear operator $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n_1\times\cdots\times n_d} \to \mathbb{R}^{n_1\times\cdots\times n_d}$, with the canonical basis fixed for $\mathbb{R}^{n_1\times\cdots\times n_d}$, we associate with $\boldsymbol{\mathcal{A}}$ the tensor $\mathbf{A} \in \mathbb{R}^{(n_1\times n_1)\times\cdots\times(n_d\times n_d)}$ in the standard way. Therefore, a tensor associated with a linear operator between tensor product spaces will be referred to as a *tensor operator*. The TT representation of the tensor operator $\mathbf{A} \in \mathbb{R}^{(n_1\times n_1)\times\cdots\times(n_d\times n_d)}$, commonly referred to as the *TT matrix*, is expressed as

$$\mathbf{A} = \underline{\mathbf{a}}_1 \cdots \underline{\mathbf{a}}_d,$$

where $\underline{\mathbf{a}}_k \in \mathbb{R}^{r_{k-1}\times n_k\times n_k\times r_k}$ is the kth *TT core*, with $r_0 = r_d = 1$.

For every $i_k, j_k \in \{1,\dots,n_k\}$ and $k \in \{1,\dots,d\}$, let $\underline{A}_k(i_k, j_k) \in \mathbb{R}^{r_{k-1}\times r_k}$ be the (i_k, j_k)th slice with respect to mode $(2,3)$ of $\underline{\mathbf{a}}_k$. Therefore, the $(i_1, j_1, \dots, i_d, j_d)$th entry of $\mathbf{A}$ can be expressed as

$$\mathbf{A}(i_1, j_1, \dots, i_d, j_d) = \underline{A}_1(i_1, j_1)\cdots\underline{A}_d(i_d, j_d).$$

The estimated storage cost remains the same as before, namely $\mathrm{O}(dnmr^d)$, where $n = \max_{i\in\{1,\dots,d\}}\{n_i\}$, $m = \max_{i\in\{1,\dots,d\}}\{m_i\}$ and $r = \max_{i\in\{1,\dots,d\}}\{r_i\}$. It is worth noting that the kth TT rank of the contraction of a TT matrix and a TT vector is less than or equal to the product of the kth TT rank of the two contracted objects, as explained in [9]. For example, given the TT matrix $\mathbf{A} \in \mathbb{R}^{n_1\times m_1\times\cdots\times n_d\times m_d}$ and the TT vector $\mathbf{x} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ with kth TT ranks r_k and s_k, respectively, their contraction $\mathbf{b} = \mathbf{A}\mathbf{x}$ is a TT vector with kth TT rank less than or equal to $r_k s_k$.

The potential growth of TT ranks is a crucial point in the implementation of algorithms using the TT formalism, as it may lead to a shortage of memory and prevent the computation from being completed. To address this issue, a rounding algorithm was proposed in [22] to reduce the TT ranks. The TT rounding algorithm takes a TT vector $\mathbf{x}$ and a relative accuracy δ as inputs and provides a TT vector $\tilde{\mathbf{x}}$ as output. The output TT vector $\tilde{\mathbf{x}}$ is at a relative distance δ from the input TT vector $\mathbf{x}$, i.e., $\|\mathbf{x}-\tilde{\mathbf{x}}\| \le \delta\|\mathbf{x}\|$. The computational cost, in terms of floating-point operations, of a TT rounding over $\mathbf{x}$ is $\mathcal{O}(dnr^3)$, as stated in [22], if $\mathbf{x} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ is a d-order TT vector with $r = \max_{i\in\{1,\dots,d\}}\{r_i\}$ and $n = \max_{i\in\{1,\dots,d\}}\{n_i\}$.

2.2. Parameter-dependent linear operators. In this section and the following one, tensor slices play a central role, so we introduce some specific notation. Given a TT vector

ETNA
Kent State University and
Johann Radon Institute (RICAM)

$\mathbf{a} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ with TT cores $\underline{\mathbf{a}}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$, $\mathbf{a}^{[k,i_k]}$ denotes the i_kth slice with respect to mode k. Henceforth, we will only take a slice with respect to the first mode, so instead of writing $\mathbf{a}^{[1,i_1]}$ for the i_1th slice on the first mode, we will simply write $\mathbf{a}^{[i_1]}$. Similarly, $\mathbf{A}^{[i_1]}$ represents the (i_1, i_1)th slice in mode $(1, 2)$ of a tensor operator $\mathbf{A} \in \mathbb{R}^{(n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$.

This section focuses on a specific type of parametric tensor operator expressed as $\mathbf{A}_\alpha = \mathbf{B}_0 + \alpha \mathbf{B}_1$, where $\alpha \in \mathbb{R}$ and $\mathbf{B}_0$ and $\mathbf{B}_1$ are two tensor operators of $\mathbb{R}^{(n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$. Assuming that α takes p different real values in the interval $[a, b]$, we define p linear systems of the form

$$\mathbf{A}_\ell \mathbf{y}_\ell = \mathbf{b}_\ell, \tag{2.1}$$

where $\mathbf{A}_\ell = \mathbf{B}_0 + \alpha_\ell \mathbf{B}_1$, $\mathbf{b}_\ell \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $\alpha_\ell \in [a, b]$ for every $\ell \in \{1, \ldots, p\}$. At this level, one can choose between either solving each system independently or solving them simultaneously in a higher-dimensional space. This latter choice will be referred to as the "all-in-one" approach. The "all-in-one" linear system can be expressed as

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{2.2}$$

where $\mathbf{A} \in \mathbb{R}^{(p \times p) \times (n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$ is a tensor operator such that

$$\mathbf{A}(h, \ell, i_1, j_1, \ldots, i_d, j_d) = \begin{cases} \mathbf{A}_\ell(i_1, j_1, \ldots, i_d, j_d) & \text{if} \quad h = \ell, \\ 0 & \text{if} \quad h \neq \ell, \end{cases} \tag{2.3}$$

and the right-hand side is $\mathbf{b} \in \mathbb{R}^{p \times n_1 \times \cdots \times n_d}$ defined as

$$\mathbf{b}(\ell, i_1, \ldots, i_d) = \mathbf{b}_\ell(i_1, \ldots, i_d) \tag{2.4}$$

for $i_k, j_k \in \{1, \ldots, n_k\}$, $k \in \{1, \ldots, d\}$ and $\ell, h \in \{1, \ldots, p\}$. The tensor operator $\mathbf{A}$ is written in a compact form as

$$\mathbf{A} = \mathbb{I}_p \otimes \mathbf{B}_0 + \operatorname{diag}(\alpha_1, \ldots, \alpha_p) \otimes \mathbf{B}_1.$$

The (ℓ, ℓ)th slice of $\mathbf{A}$ with respect to modes $(1, 2)$ is denoted by

$$\mathbf{A}^{[\ell]} = \mathbf{B}_0 + \alpha_\ell \mathbf{B}_1 = \mathbf{A}_\ell, \tag{2.5}$$

and, similarly, the ℓth slice of $\mathbf{b}$ with respect to the first mode is $\mathbf{b}^{[\ell]} = \mathbf{b}_\ell$ by construction. Consequently, equation (2.1) can also be written as

$$\mathbf{A}^{[\ell]} \mathbf{x}^{[\ell]} = \mathbf{b}^{[\ell]},$$

with $\mathbf{x}^{[\ell]} = \mathbf{y}_\ell$. This implies that, after solving the "all-in-one" system defined by equation (2.2), a specific parameter's solution can be obtained by selecting a slice from the "all-in-one" solution along the parameter mode (first mode). This slice of the "all-in-one" solution is called the *extracted solution*. In other words, if an iterative solution is computed at iteration k, the extracted solution for the ℓth problem, $\mathbf{x}_k^{[\ell]}$, is the ℓth slice with respect to mode 1 of the kth iterate of the "all-in-one" system, $\mathbf{x}_k$, i.e.,

$$\mathbf{x}_k^{[\ell]} = \mathbf{x}_k(\ell, i_1, \ldots, i_d).$$

Section 4.3 examines the connection between the numerical quality of the extracted solution and the individual solution.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

2.3. Parameter-dependent right-hand sides. This section considers a specific case of the "all-in-one" approach. The goal is to solve p linear systems that share the same linear operator but have different right-hand sides. If $\mathbf{A}_0 \in \mathbb{R}^{(n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$ is a linear tensor operator, the ℓth linear system is defined as

$$\mathbf{A}_0 \mathbf{y}_\ell = \mathbf{b}_\ell, \tag{2.6}$$

where $\mathbf{b}_\ell \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ for every $\ell \in \{1, \ldots, p\}$. To simultaneously solve all the right-hand sides expressed in equation (2.6), we repeat the construction introduced in Section 2.2, except that $\mathbf{A}_0$ is repeated on the "diagonal" of the tensor linear operator $\mathbf{A}$ defined in equation (2.3). Thanks to the tensor properties, the tensor operator $\mathbf{A} \in \mathbb{R}^{(p \times p) \times (n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$ can be written as

$$\mathbf{A} = \mathbb{I}_p \otimes \mathbf{A_0}, \tag{2.7}$$

so that $\mathbf{A}^{[\ell]} = \mathbf{A}_0$ for every $\ell \in \{1, \ldots, p\}$. The right-hand side $\mathbf{b}$ is defined similarly to the previous section, i.e., $\mathbf{b}^{[\ell]} = \mathbf{b}_\ell$.

The case of multiple right-hand sides can be formulated and solved either as an "all-in-one" problem or as a block problem, as explained in Section 4.2. Furthermore, in Section 4.4, the quality of individual solutions is linked with the numerical quality of the "all-in-one" solution in this specific case.

3. Preliminaries on GMRES and block GMRES. This section provides an overview of the GMRES algorithm, and its matrix and tensor variants. In classical matrix computation, Section 3.1 describes the main properties of the GMRES algorithm. Section 3.2 presents the block variant of GMRES, which is used to solve linear systems with multiple right-hand sides in matrix format. Finally, Section 3.3 outlines the TT-GMRES algorithm.

3.1. Preconditioned GMRES in matrix computation. When using an iterative solver to compute the solution of a linear system, it is recommended to use a stopping criterion based on a backward error [12, 17, 24]. The iterative scheme should be stopped when the backward error becomes smaller than a user-prescribed threshold. This means that the current iterate can be considered as the exact solution of a perturbed problem where the relative norm of the perturbation is smaller than the threshold. Two norm-wise backward errors can be considered for iterative schemes. Let $Ax = b$ be the linear system to be solved. We can consider a norm-wise backward error on $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$. The norm-wise backward error associated with the approximate solution x_k at iteration k is denoted as $\eta_{A,b}(x_k)$ [15]. The following equality was proved in [26]:

$$\begin{aligned} \eta_{A,b}(x_k) &= \min_{\Delta A,\, \Delta b} \{\tau > 0 : \|\Delta A\| \leq \tau \|A\|\,,\, \|\Delta b\| \leq \tau \|b\| \text{ and } (A + \Delta A)x_k = b + \Delta b\} \\ &= \frac{\|Ax_k - b\|}{\|A\|_2 \|x_k\| + \|b\|}. \end{aligned} \tag{3.1}$$

In certain situations, a simpler backward error criterion based solely on perturbations in the right-hand side can also be considered, leading to the second possible choice:

$$\begin{aligned} \eta_b(x_k) &= \min_{\Delta b} \{\tau > 0 : \|\Delta b\| \leq \tau \|b\| \text{ and } Ax_k = b + \Delta b\} \\ &= \frac{\|Ax_k - b\|}{\|b\|}. \end{aligned} \tag{3.2}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Starting from the zero initial guess, GMRES [29] constructs a series of approximations x_k in Krylov subspaces of increasing dimension k such that the residual norm of the sequence of iterates is decreasing over these nested spaces. More specifically,

$$x_k = \underset{x \in \mathcal{K}_k(A,b)}{\operatorname{argmin}} \|b - Ax\|,$$

with

$$\mathcal{K}_k(A,b) = \operatorname{span}\{b, Ab, \ldots, A^{k-1}b\}$$

being the k-dimensional Krylov subspace spanned by A and b. In practice, a matrix $V_k = [v_1, \ldots, v_k] \in \mathbb{R}^{n \times k}$ with orthonormal columns and an upper Hessenberg matrix $\bar{H}_k \in \mathbb{R}^{(k+1)\times k}$ are iteratively constructed using the Arnoldi procedure such that one has $\operatorname{span}\{V_k\} = \mathcal{K}_k(A,b)$ and

$$AV_k = V_{k+1}\bar{H}_k, \quad \text{with} \quad V_{k+1}^T V_{k+1} = I_{k+1}.$$

This relation is often referred to as the Arnoldi relation. As a result, $x_k = V_k y_k$ with

$$y_k = \underset{y \in \mathbb{R}^k}{\operatorname{argmin}} \left\|\beta e_1 - \bar{H}_k y\right\|,$$

where $\beta = \|b\|$ and $e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^{k+1}$. In exact arithmetic, the following equality holds between the least-squares residual and the true residual

$$\|\tilde{r}_k\| = \|\beta e_1 - \bar{H}_k y\| = \|b - Ax_k\|.$$

In finite-precision calculation, the equality may no longer hold. However, it has been demonstrated that the GMRES method is backward stable with respect to $\eta_{A,b}$ [24]. This means that during the iterations $\eta_{A,b}(x_k)$ may decrease to $\mathcal{O}(u)$, where u is the unit round-off of the floating-point arithmetic used for the calculations. Algorithm 1 provides an overview of GMRES. For a more detailed presentation, see [28, 29].

To control the memory footprint of the solver, a restart parameter is used to define the maximal dimension of the search Krylov space since the orthonormal basis V_k must be stored. If the algorithm fails to converge after reaching the maximum dimension of the search space, it is restarted using the final iterate as the initial guess for a new cycle of GMRES. Furthermore, it is often necessary to consider a preconditioning to speed up convergence. Using right-preconditioned GMRES consists in considering a non-singular matrix M, the so-called preconditioner, which approximates the inverse of A in some sense. In this case, the preconditioned system $AMt = b$ is solved using GMRES. The solution t is then used to compute the solution of the original system, that is, $x = Mt$. Algorithm 2 outlines the right-preconditioned GMRES for a restart parameter $\mathtt{m}$ and a convergence threshold ε.

3.2. Block GMRES in matrix computation. Block GMRES is a variant of GMRES that can be used to solve a linear system with multiple right-hand sides. The system is represented as $AX = B$ where $B = [b^{[1]}, \ldots, b^{[p]}] \in \mathbb{R}^{n \times p}$ and $X = [x^{[1]}, \ldots, x^{[p]}] \in \mathbb{R}^{n \times p}$. The algorithm uses a block variant of the Arnoldi relationship to build the search space, which is defined by the sum of the Krylov subspace associated with each of the right-hand sides, assuming the initial guess is zero. To simplify the explanation, we assume that the block of right-hand sides is full rank and that there is no partial convergence during the iterations. For a complete description of the latter situation and an efficient approach to deal with it, refer to [27]. The search space is

$$\mathcal{K}_k(A,B) = \bigoplus_{i=1}^{p} \mathcal{K}_k(A, b^{[i]}).$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Algorithm 1 x, `hasConverged` = GMRES(A, b, `m`, ε)

input: A, b, `m`, ε
$r_0 = b$, $\beta = \|r_0\|$ and $v_1 = r_0/\beta$
for $k = 1, \ldots,$ `m` **do**
  $w = Av_k$
  for $i = 1, \ldots, k$ **do** ▷ MGS variant
    $\bar{H}_{i,k} = \langle v_i, w \rangle$
    $w = w - \bar{H}_{i,k} v_i$
  end for
  $\bar{H}_{k+1,k} = \|w\|$
  $v_{k+1} = w/\bar{H}_{k+1,k}$
  $y_k = \operatorname{argmin}_{y \in \mathbb{R}^k} \|\beta e_1 - \bar{H}_k y\|$
  $x_k = V_k y_k$
  if $(\eta_{A,b}(x_k) < \varepsilon)$ **then**
    `hasConverged = True`
    break
  end if
end for
return: $x = x_k$, `hasConverged`

Algorithm 2 x, `hasConverged` = Right-GMRES(A, M, b, x_0, `m`, ε)

input: A, M, b, `m`, ε
`hasConverged = False`
$x = x_0$
while `not(hasConverged)` **do**
  $r = b - Ax$ ▷ Iterative refinement step with at most m GMRES iterations on AM
  t_k, `hasConverged` = GMRES(AM, r, `m`, ε)
  $x = x + Mt_k$ ▷ Update the unpreconditioned solution with the computed correction
end while
return: x, `hasConverged`

In this space, the kth iterate is defined as the minimum Frobenius norm of the block residual, which is given by

$$X_k = \underset{X \in \mathcal{K}_k(A,B)}{\operatorname{argmin}} \|B - AX\|_F.$$

The residual norm of each individual right-hand side is minimized over the sum of the Krylov spaces associated with all right-hand sides. Therefore, if the ith column of the residual block is considered, the kth iterate associated with the ith right-hand side is

$$x_k^{[i]} = \underset{x \in \mathcal{K}_k(A,B)}{\operatorname{argmin}} \|b^{[i]} - Ax^{[i]}\|.$$

For a more detailed discussion on the block GMRES variant, see [27, 28].

3.3. Preconditioned GMRES in Tensor Train format. Let $\mathbf{A} \in \mathbb{R}^{(n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$ be a tensor operator and $\mathbf{b} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ a tensor, then the general tensor linear system is

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{3.3}$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where $\mathbf{x} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$. It is important to note that if we set $d = 1$, we obtain the standard linear system from classical matrix computation. To solve equation (3.3), we can use a tensor-extended version of GMRES. Since all operations involved in this iterative solver are feasible with the TT formalism, we assume that all objects are expressed in TT format. One major limitation of this approach is the repetition of additions and contractions in the various loops. This results in the growth of TT rank and potential memory overconsumption. Therefore, it is crucial to introduce compression steps in TT-GMRES. However, special attention must be paid to the selection of the TT rounding parameter to ensure that the prescribed GMRES tolerance ε can be achieved. The complete TT-GMRES algorithm is presented in Algorithm 3.

Algorithm 3 $\mathbf{x}$, `hasConverged` = TT-GMRES($\mathbf{A}$, $\mathbf{b}$, `m`, ε, δ)

input: $\mathbf{A}$, $\mathbf{b}$, `m`, ε, δ
$\mathbf{r_0} = \mathbf{b}$, $\beta = \|\mathbf{r_0}\|$ and $\mathbf{v}_1 = (1/\beta)\mathbf{r}_0$
for $k = 1, \ldots,$ `maxit` **do**
    $\mathbf{w} = \texttt{TT-round}(\mathbf{A}\mathbf{v}_k, \delta)$ ▷ MGS variant
    for $i = 1, \ldots, k$ **do**
        $\bar{H}_{i,k} = \langle \mathbf{v}_i, \mathbf{w} \rangle$
        $\mathbf{w} = \mathbf{w} - \bar{H}_{i,k}\mathbf{v}_i$
    end for
    $\mathbf{w} = \texttt{TT-round}(\mathbf{w}, \delta)$
    $\bar{H}_{k+1,k} = \|\mathbf{w}\|$
    $\mathbf{v}_{k+1} = (1/\bar{H}_{k+1,k})\mathbf{w}$
    $y_k = \operatorname{argmin}_{y \in \mathbb{R}^k} \|\beta e_1 - \bar{H}_k y\|$
    $\mathbf{x}_k = \texttt{TT-round}\big(\sum_{j=1}^{k+1} y_k(j)\mathbf{v}_j, \delta\big)$
    if $(\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}_k) < \varepsilon)$ **then**
        `hasConverged = True`
        break
    end if
end for
return: $\mathbf{x} = \mathbf{x}_k$, `hasConverged`

Algorithm 4 $\mathbf{x}$, `hasConverged` = TT-Right-GMRES($\mathbf{A}$, $\mathbf{M}$, $\mathbf{b}$, $\mathbf{x_0}$, `m`, ε, δ)

input: $\mathbf{A}$, $\mathbf{M}$, $\mathbf{b}$, `m`, ε, δ
`hasConverged = False`
$\mathbf{x} = \mathbf{x_0}$
while `not (hasConverged)` **do**
    $\mathbf{r} = \texttt{TT-round}(\mathbf{b} - \mathbf{A}\mathbf{x}, \delta)$ ▷ Iterative refinement step with at most m GMRES iterations on $\mathbf{AM}$
    $\mathbf{t}_k$, `hasConverged` = TT-GMRES($\mathbf{AM}$, $\mathbf{r}$, `m`, ε, δ)
    $\mathbf{x} = \texttt{TT-round}(\mathbf{x} + \mathbf{M}\mathbf{t}_k, \delta)$ ▷ Update the unpreconditioned solution with the computed correction
end while
return: $\mathbf{x}$, `hasConverged`

Algorithms 3 and 4 introduce an additional input parameter, δ, which represents the TT rounding threshold. The TT rounding algorithm at accuracy δ is applied to the result of the

ETNA
Kent State University and
Johann Radon Institute (RICAM)

contraction between $\mathbf{A}$ and the last Krylov basis vector computed in line 4, to the new Krylov basis vector after orthogonalization in line 9, and to the updated iterative solution in line 13. The purpose of this step is to balance the rank growth that the tensor contraction and addition of the previous steps may cause. Notice that the MGS orthogonalization kernel plays a key role in the rank growth. When an orthogonalization kernel is applied to low-rank TT vectors, it produces a set of TT vectors of larger TT ranks, to satisfy the orthogonality constraint. This growth can be balanced by the use of the TT rounding operation, which reduces the TT ranks and affects the orthogonality of the final basis. In [4], the authors study in detail this phenomenon for several orthogonalization kernels applied to TT vectors. As shown in the numerical experiments in Section 5, the TT rounding accuracy (δ) must be less than or equal to the GMRES target accuracy (ε).

4. Some comments and observations on GMRES in tensor format. Section 4.1 compares the TT-GMRES algorithm in the variable accuracy framework described in Section 3.3 with the TT-GMRES algorithm from [7]. Furthermore, the connection between the TT-GMRES algorithm from [7] and the inexact GMRES theory is proved. In Section 4.2, two possible approaches for solving multiple right-hand sides are compared: the TT-GMRES algorithm applied with the "all-in-one" construction described in Section 2.3, and the block GMRES variant. Finally, Sections 4.3 and 4.4 describe backward error bounds in TT format for parametric linear systems and multiple right-hand sides, as introduced in Sections 4.3 and 4.4, respectively.

4.1. TT-GMRES with variable rounding versus inexact GMRES. In this section, we will first recall some of the existing results from the literature on GMRES and inexact GMRES, and then we will draw some connections with TT-GMRES. In exact arithmetic, two important properties hold for GMRES: the Arnoldi basis is perfectly orthogonal (i.e., $V_k^T V_k = I_k$), and, as a corollary, the least-squares residual norm is equal to the linear system residual norm. However, in finite precision computation, it is known that these two equalities do not hold any more [24]. Despite this, GMRES is backward stable, that is, $\eta_{A,b}(x_k) \approx \mathcal{O}(u)$ when $\kappa_2(V_k) > 4/3$; cf. [24]. The $\kappa_2(V_k)$ can be used to detect the stagnation of the backward error. Indeed, the $\eta_{A,b}(x_k)$ will be $\mathcal{O}(u)$ within the iteration at which $\kappa_2(V_k)$ is greater than $4/3$. Still, in exact arithmetic, the idea of relaxing the accuracy when performing the matrix–vector product in the Arnoldi procedure was first observed experimentally in [3]. The inaccuracy in the matrix–vector product is modelled by introducing a perturbation matrix E_k (i.e., $w = (A + E_k)v_k$) whose relative norm defines the amount of inaccuracy. Later, a series of papers [10, 30, 32] provided theoretical justification, showing that in exact arithmetic the norm of E_k can grow as the inverse of the residual norm of the linear system times a prescribed threshold η, while still ensuring that the attainable GMRES residual norm will reach this threshold η.

These latter results motivated a heuristic proposed in the TT-GMRES algorithm described in [7] and [25]. The heuristic increases the TT rounding threshold proportionally to the inverse of the residual least-squares norm. The TT rounding computes a TT vector, which can be viewed as a perturbation of the original vector with a relative perturbation norm bounded by the threshold. Upon initial examination, an issue with this TT-GMRES algorithm is that the perturbation is applied to $\mathbf{w}$, the outcome of the matrix–vector product, rather than the linear operator involved to compute it. However, we show below that the TT rounding can also be interpreted as a perturbation on the linear operator, which partially justifies the proposed heuristic. In Algorithm 3, during step 4, the computation $\mathbf{w} = \texttt{TT-round}(\mathbf{A}\mathbf{v}_k, \delta)$ can be written as the application of an inexact version of the operator applied to $\mathbf{v}_k$, that is,

$$\mathbf{w} = \mathbf{A}\mathbf{v}_k + \Delta\mathbf{w} = (\mathbf{A} + \mathbf{E}_k)\mathbf{v}_k,$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

where $\Delta\mathbf{w}$ is a tensor whose norm is bounded by $\|\Delta\mathbf{w}\| \leq \delta\|\mathbf{A}\mathbf{v}_k\|$ by a property of the TT rounding. In fact, if we think of $\mathbf{R}$ as the Householder reflector [4] that maps the normalized TT vector $\Delta\hat{\mathbf{w}} = \Delta\mathbf{w}/\|\Delta\mathbf{w}\|$ to $\mathbf{v}_k$, we do have

$$\mathbf{w} = (\mathbf{A} + \mathbf{E}_k)\mathbf{v}_k,$$

where $\mathbf{E}_k = \|\Delta\mathbf{w}\|\mathbf{R}$, so that $\|\mathbf{E}_k\| = \|\Delta\mathbf{w}\| \leq \delta\|\mathbf{A}\mathbf{v}_k\| \leq \delta\|\mathbf{A}\|$.

If a sufficiently good preconditioner is used, the linear operator seen by TT-GMRES is such that $\|\mathbf{A}\| \approx 1$. This results in selecting the TT rounding threshold δ_k at the kth GMRES iteration as

$$\delta_k = \mathcal{O}(\|\beta e_1 - \bar{H}_y y_k\|^{-1}),$$

as originally proposed in the first paper on inexact GMRES, then known as relaxed GMRES [3].

We have shown that a variable TT rounding strategy can be viewed as an inexact matrix–vector product in the inexact GMRES framework, but some other gaps need to be filled to fully assess the robustness of TT-GMRES. In particular, we lack the analysis related to the other roundings performed in Lines 9 and 13 of Algorithm 3. These missing theoretical pieces will be the subject of future work.

4.2. GMRES in tensor format versus GMRES and block GMRES in matrix computation. This section investigates the relationship between the iterates computed by GMRES applied on a $(d+1)$-mode tensor space to solve all the d-mode right-hand sides at once versus GMRES applied to the individual d-mode right-hand sides individually. Specifically, the iterates computed by the two approaches belong to the same Krylov space but are characterized by a different optimality condition for the residual norm minimization. The presentation uses classical $\mathbb{R}^n$ vector spaces to maintain simple notation. Using rigorous notation to describe the underlying principle in tensor spaces would make the notation very heavy, which might obscure the ideas.

The problem of solving a linear system $AX = B$, where A, X and B are matrices of compatible dimensions such that $X = [x^{[1]}, \ldots, x^{[p]}]$ and $B = [b^{[1]}, \ldots, b^{[p]}]$, can be recast by the Kronecker product in a tensor-like structure as

$$(\mathbb{I}_p \otimes A) \begin{bmatrix} x^{[1]} \\ \vdots \\ x^{[p]} \end{bmatrix} = \begin{bmatrix} b^{[1]} \\ \vdots \\ b^{[p]} \end{bmatrix}.$$

Based on this structure of the linear operator, we can observe that the individual iterate is $x_k^{[i]} \in \mathcal{K}_k(A, b^{[i]})$ and the residual norm associated with the global iterate is

$$\|r_k\| = \left(\sum_{i=1}^{p} \|r_k^{[i]}\|^2 \right)^{1/2},$$

where $r_k^{[i]}$ is the residual associated with the individual iterate extracted from the global iterate. The coordinates of the individual iterates in the Krylov basis will be the same for all the iterates, and these iterates minimize the 2-norm of the sum of the squares of the individual square residual norms.

If GMRES is run for each right-hand side separately, the iterate at step k would also belong to $\mathcal{K}_k(A, b^{[i]})$, but it would only minimize its own residual norm, which would consequently be lower than the one in the tensor case.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

Another option is to use a block-GMRES algorithm defined on d mode tensors. As already mentioned in Section 3.2, at step k, each individual residual norm is minimized over the sum of the individual Krylov spaces. That is a dual situation in the solution of a $(d+1)$-mode computation, where each solution is sought in its own Krylov space by minimizing the sum of the squares of all residual norms.

4.3. Backward error bounds for parametric operators. The purpose of the following propositions is to examine the relationship between the backward error of the "all-in-one" system solution and the extracted individual one. The equalities provided for the "all-in-one" system are true when the tensor and the tensor operators are given in full format, but they also hold in TT format. For further details on the "all-in-one" construction in TT format, refer to [5, Appendix C].

The bounds that will be proven allow us to adjust the convergence threshold when solving for multiple parameters, while ensuring a specific quality for the individual extracted solutions. Specifically, the bound presented in equation (4.1) of Proposition 4.1 indicates that if a certain accuracy ε is expected for the extracted individual solution in terms of the backward error in (3.2), a more stringent convergence threshold should be used for the "all-in-one" system solution. This threshold should be set to $\varepsilon/\sqrt{p}$.

PROPOSITION 4.1. *Given the "all-in-one" operator* $\mathbf{A} \in \mathbb{R}^{(p\times p)\times(n_1\times n_1)\times\cdots\times(n_d\times n_d)}$ *and the right-hand side* $\mathbf{b} \in \mathbb{R}^{p\times n_1\times\cdots\times n_d}$*, as defined in equations* (2.3) *and* (2.4)*, we consider the "all-in-one" system*

$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

Let $\mathbf{A}_\ell \in \mathbb{R}^{(n_1\times n_1)\times\cdots\times(n_d\times n_d)}$ *be the tensor operator as in equation* (2.5) *and let* $\mathbf{b}_\ell \in \mathbb{R}^{n_1\times\cdots\times n_d}$ *be a tensor such that* $\|\mathbf{b}_\ell\| = 1$*, which defines the individual linear systems*

$$\mathbf{A}_\ell \mathbf{y}_\ell = \mathbf{b}_\ell,$$

where $\mathbf{A}_\ell = \mathbf{A}^{[\ell]}$ *and* $\mathbf{b}_\ell = \mathbf{b}^{[\ell]}$ *for every* $\ell \in \{1,\dots,p\}$.

If $\mathbf{x}_k$ *represents the "all-in-one" iterate, we have*

$$\eta_{\mathbf{b}}(\mathbf{x}_k)\sqrt{p} \geq \eta_{\mathbf{b}_\ell}(\mathbf{x}_k^{[\ell]}) \tag{4.1}$$

for $\ell \in \{1,\dots,p\}$.

Proof. For the sake of simplicity, we use $\eta_{\mathbf{b}}$ and $\eta_{\mathbf{b}_\ell}$ squared throughout the proof and discard the subscript of the kth "all-in-one" iterate. The quantity $\eta^2_{\mathbf{b}_\ell}(\mathbf{x}^{[\ell]})$ is explicitly expressed as

$$\eta^2_{\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) = \frac{\left\|\mathbf{A}_\ell\mathbf{x}^{[\ell]} - \mathbf{b}_\ell\right\|^2}{\|\mathbf{b}_\ell\|^2},$$

while $\eta^2_{\mathbf{b}}(\mathbf{x})$ is written as

$$\eta^2_{\mathbf{b}}(\mathbf{x}) = \frac{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2}{\|\mathbf{b}\|^2}. \tag{4.2}$$

Owing to the diagonal structure of $\mathbf{A}$ and the definition of the Frobenius norm, equation (4.2) can be simplified to

$$\eta^2_{\mathbf{b}}(\mathbf{x}) = \frac{\sum_{\ell=1}^n \left\|(\mathbf{A}\mathbf{x}-\mathbf{b})^{[\ell]}\right\|^2}{\sum_{k=1}^p \left\|\mathbf{b}^{[k]}\right\|^2} = \frac{\sum_{\ell=1}^n \left\|\mathbf{A}_\ell\mathbf{x}^{[\ell]} - \mathbf{b}_\ell\right\|^2}{\sum_{k=1}^p \|\mathbf{b}_k\|^2} = \frac{\sum_{\ell=1}^p \eta^2_{\mathbf{b}_\ell}(\mathbf{x}^{[\ell]})}{p},$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

since $\|\mathbf{b}\|^2 = \sum_{k=1}^{n} \|\mathbf{b}_k\|^2 = p$. Taking the square root of both sides of this equation yields the desired result. □

For the backward error based on perturbation of both the linear operator and the right-hand sides, defined by equation (3.1), a similar result can be derived.

PROPOSITION 4.2. *Based on the hypothesis and notation of Proposition 4.1 for* $\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x})$ *and* $\eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}^{[\ell]})$ *associated with the linear systems* $\mathbf{A}\mathbf{x} = \mathbf{b}$ *and* $\mathbf{A}_\ell\mathbf{y}_\ell = \mathbf{b}_\ell$*, respectively, for every* $\ell \in \{1, \ldots, p\}$*, then we have*

$$\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}_k)\,\rho_\ell(\mathbf{x}_k) \geq \eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}_k^{[\ell]}) \quad \textit{where} \quad \rho_\ell(\mathbf{x}_k) = \frac{\|\mathbf{A}\|_2\|\mathbf{x}_k\| + \sqrt{p}}{\left\|\mathbf{A}_\ell\mathbf{x}_k^{[\ell]}\right\| + 1}, \tag{4.3}$$

with $\mathbf{x}_k$ *being the* k*th "all-in-one" iterate and* $\mathbf{x}_k^{[\ell]}$ *being its* ℓ*th slice with respect to mode* 1.

Proof. The subscript of the kth "all-in-one" iterate is dropped for simplicity. The backward error $\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x})$ is explicitly written as

$$\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}) = \frac{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|}{\|\mathbf{A}\|_2\|\mathbf{x}\| + \|\mathbf{b}\|}.$$

Multiplying the previous equation by $\eta_{\mathbf{b}}(\mathbf{x})$ yields

$$\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}) = \frac{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|}{\|\mathbf{A}\|_2\|\mathbf{x}\| + \|\mathbf{b}\|}\,\frac{\eta_{\mathbf{b}}(\mathbf{x})}{\eta_{\mathbf{b}}(\mathbf{x})} = \frac{\|\mathbf{b}\|}{\|\mathbf{A}\|_2\|\mathbf{x}\| + \|\mathbf{b}\|}\,\eta_{\mathbf{b}}(\mathbf{x}) = \frac{\sqrt{p}}{\|\mathbf{A}\|_2\|\mathbf{x}\| + \sqrt{p}}\,\eta_{\mathbf{b}}(\mathbf{x}) \tag{4.4}$$

according to the definition of $\eta_{\mathbf{b}}(\mathbf{x})$, and $\|\mathbf{b}\| = \sqrt{p}$. Similarly, $\eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}^{[\ell]})$ is expressed in terms of $\eta_{\mathbf{b}_\ell}(\mathbf{x}^{[\ell]})$ as

$$\eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) = \frac{\|\mathbf{b}_\ell\|}{\|\mathbf{A}_\ell\|_2\left\|\mathbf{x}^{[\ell]}\right\| + \|\mathbf{b}_\ell\|}\,\eta_{\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) = \frac{1}{\|\mathbf{A}_\ell\|_2\left\|\mathbf{x}^{[\ell]}\right\| + 1}\,\eta_{\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) \tag{4.5}$$

since $\|\mathbf{b}_\ell\| = 1$. By multiplying each side of equation (4.4) by $(\|\mathbf{A}\|_2\|\mathbf{x}\| + \sqrt{p}\,)$, it follows that

$$(\|\mathbf{A}\|_2\|\mathbf{x}\| + \sqrt{p}\,)\eta_{\mathbf{A},\mathbf{b}} = \eta_{\mathbf{b}}\sqrt{p}.$$

Owing to the result of Proposition 4.1, we have

$$(\|\mathbf{A}\|_2\|\mathbf{x}\| + \sqrt{p}\,)\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}) = \eta_{\mathbf{b}}(\mathbf{x})\sqrt{p} \geq \eta_{\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) = (\|\mathbf{A}_\ell\|_2\left\|\mathbf{x}^{[\ell]}\right\| + 1\,)\eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) \tag{4.6}$$

from equation (4.5). Dividing both sides of equation (4.6) by $\|\mathbf{A}_\ell\|_2\left\|\mathbf{x}^{[\ell]}\right\| + 1$, we obtain

$$\frac{\|\mathbf{A}\|_2\|\mathbf{x}\| + \sqrt{p}}{\left\|\mathbf{A}_\ell\mathbf{x}^{[\ell]}\right\| + 1}\,\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}) \geq \eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}^{[\ell]})$$

because $\|\mathbf{A}_\ell\|_2\left\|\mathbf{x}^{[\ell]}\right\| \geq \left\|\mathbf{A}_\ell\mathbf{x}^{[\ell]}\right\|$ according to the definition of the L2 norm. □

The calculation of $\rho_\ell(\mathbf{x}_k)$ in equation (4.3) requires a little extra cost.

COROLLARY 4.3. *Let* $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ *be a sequence of iterative solutions and* ν *a real value. If there exists a* $k_\ell^* \in \mathbb{N}$ *such that* $|\,\|\mathbf{A}_\ell\mathbf{x}_k^{[\ell]}\| - 1| \leq \nu$ *for every* $k \geq k_\ell^*$*, then*

$$\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}_k)\,\rho^*(\mathbf{x}_k) \geq \eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}_k^{[\ell]}) \quad \textit{where} \quad \rho^*(\mathbf{x}_k) = \frac{\|\mathbf{A}\|_2\|\mathbf{x}_k\| + \sqrt{p}}{2 - \nu} \tag{4.7}$$

for every $\ell \in \{1, \ldots, p\}$ *and for every* $k \in \mathbb{N}$ *such that* $k \geq k^{**}$*, where* $k^{**} = \max k_\ell^*$.

This corollary provides a bound depending only on the "all-in-one" iterative solution, and it holds true for all the d-dimensional problems.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

4.4. Backward error bounds for parametric right-hand sides. If the initial guess for $\mathbf{x}_0 \in \mathbb{R}^{p\times n_1\times\cdots\times n_d}$ is the null tensor and $\mathbf{b}$ is defined as in (2.4), then at the kth iteration TT-GMRES minimizes with respect to $\mathbf{x}_k$ the norm of the residual $\mathbf{r}_k = \mathbf{A}\mathbf{x}_k - \mathbf{b}$ on the space

$$\mathcal{K}_k(\mathbf{A},\mathbf{b}) = \operatorname{span}\{\mathbf{b}, \mathbf{A}\mathbf{b}, \mathbf{A}^2\mathbf{b}, \ldots, \mathbf{A}^{k-1}\mathbf{b}\}.$$

In other words, we seek a tensor $\mathbf{x}_k \in \mathcal{K}_k(\mathbf{A},\mathbf{b})$ such that

$$\mathbf{x}_k = \underset{\mathbf{x}\in\mathcal{K}_k(\mathbf{A},\mathbf{b})}{\operatorname{argmin}} \|\mathbf{A}\mathbf{x}-\mathbf{b}\|.$$

The Frobenius norm of $\mathbf{r}_k = \mathbf{A}\mathbf{x}_k - \mathbf{b}$, due to the diagonal structure of $\mathbf{A}$ defined by (2.7), is naturally written as follows:

$$\|\mathbf{r}_k\|^2 = \sum_{\ell=1}^{p} \|\mathbf{b}_\ell - \mathbf{A}_0\mathbf{x}_k^{[\ell]}\|^2. \tag{4.8}$$

PROPOSITION 4.4. *Given the "all-in-one" operator* $\mathbf{A} \in \mathbb{R}^{(p\times p)\times(n_1\times n_1)\times\cdots\times(n_d\times n_d)}$ *and the right-hand side* $\mathbf{b} \in \mathbb{R}^{p\times n_1\times\cdots\times n_d}$*, as defined in equations* (2.7) *and* (2.4)*, we consider the "all-in-one" system*

$$\mathbf{A}\mathbf{x} = \mathbf{b}.$$

Let $\mathbf{b}_\ell \in \mathbb{R}^{n_1\times\cdots\times n_d}$ *be a tensor such that* $\|\mathbf{b}_\ell\| = 1$*, which defines the individual linear systems*

$$\mathbf{A}_0\mathbf{y}_\ell = \mathbf{b}_\ell,$$

where $\mathbf{b}_\ell = \mathbf{b}^{[\ell]}$ *for every* $\ell \in \{1,\ldots,p\}$.

If $\mathbf{x}_k$ *represents the "all-in-one" iterate, we have*

$$\eta_{\mathbf{b}}(\mathbf{x}_k)\sqrt{p} \geq \eta_{\mathbf{b}_\ell}(\mathbf{x}_k^{[\ell]})$$

for $\ell \in \{1,\ldots,p\}$.

Proof. The proof is based on equation (4.8) using similar arguments to those for the proof of Proposition 4.1. □

Similarly to Proposition 4.2, an informative bound of lower practical interest can be derived.

PROPOSITION 4.5. *Under the hypothesis of Proposition 4.2, if* $\mathbf{A} = \mathbb{I}_p \otimes \mathbf{A}_0$*, then for* $\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x})$ *and* $\eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}^{[\ell]})$ *associated with the linear systems* $\mathbf{A}\mathbf{x} = \mathbf{b}$ *and* $\mathbf{A}_0\mathbf{y}_\ell = \mathbf{b}_\ell$*, respectively, the following inequality holds:*

$$\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}_k)\,\psi_\ell(\mathbf{x}_k) \geq \eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}_k^{[\ell]}) \quad \textit{where} \quad \psi_\ell(\mathbf{x}_k) = \frac{\|\mathbf{A}_0\|_2\|\mathbf{x}_k\| + \sqrt{p}}{\left\|\mathbf{A}_0\mathbf{x}_k^{[\ell]}\right\| + 1}$$

for every $\ell \in \{1,\ldots,p\}$.

Proof. This result follows from the thesis of Proposition 4.2, since $\|\mathbf{A}\|_2 = \|\mathbf{A}_0\|_2$. □

The bound of Corollary 4.3 remains valid in the multiple right-hand-side structure described in this section. The thesis does not depend on the repetition of the same of the operator.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

5. Numerical experiments. This section investigates the numerical behavior of the TT-GMRES solver for linear problems with increasing dimension, as it naturally arises in some partial differential equation (PDE) studies. The TT operators of our numerical examples are directly constructed in TT format, thanks to their peculiarity. In this section, we present numerical aspects related to convergence of the algorithm and the computational cost, with a focus on memory growth and memory savings. All experiments were conducted using `Python 3.6.9` and the tensor toolbox `ttpy 1.2.0` [23].

The problem we will address involves *Laplace-like* operators. A Laplace-like tensor operator, $\mathbf{A} \in \mathbb{R}^{n_1 \times m_1 \times \cdots \times n_d \times m_d}$, is the sum of operators written as

$$\begin{aligned}\mathbf{A} = {} & M_1 \otimes R_2 \otimes R_3 \otimes \cdots \otimes R_{d-2} \otimes R_{d-1} \otimes R_d \\ & + L_1 \otimes M_2 \otimes R_3 \otimes \cdots \otimes R_{d-2} \otimes R_{d-1} \otimes R_d \\ & + \cdots + L_1 \otimes L_2 \otimes L_3 \otimes \cdots \otimes L_{d-2} \otimes M_{d-1} \otimes R_d \\ & + L_1 \otimes L_2 \otimes L_3 \otimes \cdots \otimes L_{d-2} \otimes L_{d-1} \otimes M_d,\end{aligned}$$

where $L_k, M_k, R_k \in \mathbb{R}^{n_k \times m_k}$ for every $k \in \{1, \ldots, d\}$. These linear operators are expressed in TT format with TT rank 2, that is,

$$\mathbf{A} = \begin{bmatrix} L_1 & M_1 \end{bmatrix} \otimes \begin{bmatrix} L_2 & M_2 \\ 0 & R_2 \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} L_{d-1} & M_{d-1} \\ 0 & R_{d-1} \end{bmatrix} \otimes \begin{bmatrix} M_d \\ R_d \end{bmatrix},$$

as proven in [18, Lemma 5.1]. The expression for the discrete d-dimensional Laplacian on a uniform grid of n points in each direction is

$$\mathbf{\Delta}_d = \Delta_1 \otimes \mathbb{I}_n \otimes \cdots \otimes \mathbb{I}_n + \cdots + \mathbb{I}_n \otimes \mathbb{I}_n \otimes \cdots \otimes \Delta_1,$$

where $\mathbb{I}_n$ is the identity matrix of size n, and $\Delta_1 \in \mathbb{R}^{n \times n}$ is the discrete one-dimensional Laplacian using the central-point finite difference scheme with discretization step $h = 1/(n+1)$, that is,

$$\Delta_1 = \frac{1}{h^2} \begin{bmatrix} -2 & 1 & 0 & \ldots & 0 \\ 1 & -2 & 1 & \ldots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 1 & -2 & 1 \\ 0 & 0 & \ldots & 1 & -2 \end{bmatrix}.$$

The TT expression of $\mathbf{\Delta}_d$ is

$$\mathbf{\Delta}_d = \begin{bmatrix} \mathbb{I}_n & \Delta_1 \end{bmatrix} \otimes \begin{bmatrix} \mathbb{I}_n & \Delta_1 \\ \mathbb{O} & \mathbb{I}_n \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} \mathbb{I}_n & \Delta_1 \\ \mathbb{O} & \mathbb{I}_n \end{bmatrix} \otimes \begin{bmatrix} \Delta_1 \\ \mathbb{I}_n \end{bmatrix}. \tag{5.1}$$

To efficiently solve linear systems, we use an approximation of the inverse of the discrete Laplacian operator, $\mathbf{M}$, as a preconditioner [13, 14]. This operator can be written as

$$\mathbf{M} = \sum_{k=-q}^{q} c_k \exp(-t_k \Delta_1) \otimes \cdots \otimes \exp(-t_k \Delta_1), \tag{5.2}$$

where $c_k = \xi t_k$, $t_k = \exp(k\xi)$ and $\xi = \pi/q$. The TT ranks of $\mathbf{M}$ will be at least $2q+1$, based on the previously stated property of the sum of TT tensors.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

To examine the primary numerical characteristics of the TT-GMRES implementations discussed in the previous sections, we analyze the classical convection–diffusion equation, which is the same as that examined in [7] and is expressed as

$$\begin{cases} -\Delta u + 2y(1-x^2)\dfrac{\partial u}{\partial x} - 2x(1-y^2)\dfrac{\partial u}{\partial y} = 0 \quad \text{in} \quad \Omega = [-1,1]^3, \\ u_{\{y=1\}} = 1 \qquad \text{and} \qquad u_{\partial\Omega\setminus\{y=1\}} = 0\,. \end{cases} \tag{5.3}$$

We set a grid of n points per mode over $[-1,1]^3$ and discretize the Laplacian as shown in equation (5.1) with $d=3$. The discretization of the first derivative of u with respect to mode 1, $\boldsymbol{\nabla}_x$, is defined as $\boldsymbol{\nabla}_x = \nabla_1 \otimes \mathbb{I}_n \otimes \mathbb{I}_n$. Similarly, the discrete first derivative with respect to mode 2, $\boldsymbol{\nabla}_y$, is written as $\boldsymbol{\nabla}_y = \mathbb{I}_n \otimes \nabla_1 \otimes \mathbb{I}_n$, where ∇_1 is the order-two central finite difference matrix, i.e.,

$$\nabla_1 = \frac{1}{2h}\begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ -1 & 0 & 1 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & -1 & 0 & 1 \\ 0 & 0 & \dots & -1 & 0 \end{bmatrix}.$$

Let $v : [-1,1]^3 \to \mathbb{R}^2$ be a function such that $v(x,y,z) = \big(2y(1-x^2), -2x(1-y^2)\big)$. The components of v are discretized over the Cartesian grid set on $[-1,1]^3$, defining two tensors $\mathbf{V}_1, \mathbf{V}_2 \in \mathbb{R}^{(n\times n)\times(n\times n)\times(n\times n)}$ such that $\mathbf{V}_1 = \text{diag}(1-x^2) \otimes \text{diag}(2y) \otimes \mathbb{I}_n$ and $\mathbf{V}_2 = \text{diag}(-2x) \otimes \text{diag}(1-y^2) \otimes \mathbb{I}_n$. The diffusion term $\mathbf{D}$ discretized is expressed as

$$\begin{aligned} \mathbf{D} &= \mathbf{V}_1 \bullet \boldsymbol{\nabla}_x + \mathbf{V}_2 \bullet \boldsymbol{\nabla}_y \\ &= \text{diag}(1-x^2)\nabla_1 \otimes \text{diag}(2y) \otimes \mathbb{I}_n + \text{diag}(-2x) \otimes \text{diag}(1-y^2)\nabla_1 \otimes \mathbb{I}_n. \end{aligned} \tag{5.4}$$

The operator passed to the TT-GMRES algorithm is $\mathbf{A} = -\boldsymbol{\Delta}_3 + \mathbf{D}$. The right-hand side is represented by the TT vector $\mathbf{b} \in \mathbb{R}^{n\times n\times n}$ and the initial guess is the zero TT vector $\mathbf{x}_0$.

To ensure rapid convergence, we use the right preconditioner $\mathbf{M}$ from equation (5.2), as in [7], for this test example. The preconditioner TT matrix $\mathbf{M}$ is always computed by a number of addends q equal to a quarter of the grid step dimension. To keep the TT rank of the preconditioner small, we choose to round it to 10^{-2}. The choice of the number of addends and of the TT rounding compression are further discussed in [5, Appendix A].

When a right preconditioner is used, TT-GMRES actually solves the linear system $\mathbf{AMt} = \mathbf{b}$. To evaluate the convergence of the right-preconditioned TT-GMRES, we display the convergence history of $\eta_{\mathbf{AM},\mathbf{b}}$, which is defined as

$$\eta_{\mathbf{AM},\mathbf{b}}(\mathbf{t}_k) = \frac{\|\mathbf{AMt}_k - \mathbf{b}\|}{\|\mathbf{AM}\|_2\|\mathbf{t}_k\| + \|\mathbf{b}\|},$$

with $\mathbf{t}_k$ being the preconditioned approximate solution at the kth iteration. We compute the norm of the residual, the norm of the right-hand side, and the norm of the iterative preconditioned approximate solution. The L2-norm of the preconditioner operator $\mathbf{AM}$ is computed using a sampling approximation. Let $\mathcal{W}$ be a set of normalized TT vectors, randomly generated from a normal distribution. A lower bound for $\|\mathbf{AM}\|_2$ can be found as the maximum norm of the image of the elements of $\mathcal{W}$ through $\mathbf{AM}$, i.e.,

$$\tau_{\mathbf{AM}} = \max_{\mathbf{w}\in\mathcal{W}} \|\mathbf{AMw}\| \le \|\mathbf{AM}\|_2\,.$$

ETNA
Kent State University and
Johann Radon Institute (RICAM)

As a consequence, the backward error will be estimated as

$$\eta_{\mathbf{AM},\mathbf{b}}(\mathbf{t}_k) \leq \frac{\|\mathbf{AMt}_k - \mathbf{b}\|}{\tau_{\mathbf{AM}}\|\mathbf{t}_k\| + \|\mathbf{b}\|}.$$

In the following numerical experiments, we report on this quantity, where $\tau_{\mathbf{AM}}$ is computed using 10 elements of $\mathcal{W}$.

5.1. Main features and robustness properties. A link between the TT-GMRES variant proposed in [7] and inexact TT-GMRES is established in Section 5.1.1. It is shown that a robust stopping criterion based on the backward error with perturbation on both the linear operator and the right-hand side is suitable for the inexact TT-GMRES algorithm. Additionally, the backward stability of inexact TT-GMRES is experimentally investigated in Section 5.1.2.

5.1.1. Comparison of inexact GMRES and classical GMRES in TT format. This section presents a comparison of the numerical behavior of TT-GMRES with constant TT rounding as described in Algorithm 3 and its inexact variant introduced in [7]. In the inexact variant, the TT rounding threshold at step 4 of Algorithm 3 is increased as $\|\tilde{\mathbf{r}}_k\|^{-1}$, where $\tilde{\mathbf{r}}_k = \|\beta e_1 - \bar{H}_k y_k\|$. The numerical behavior is evaluated through the convergence history of the norm-wise backward error of the preconditioned system, $\eta_{\mathbf{AM},\mathbf{b}}(\mathbf{t}_k)$, as defined by (3.1). Figure 5.1 displays the convergence history of the norm-wise backward error for the TT-GMRES with constant TT rounding accuracy and the inexact TT-GMRES with varying TT rounding accuracy (referred to as "inexact" in the legend of the curves). We consider an initial rounding accuracy $\delta \in \{10^{-3}, 10^{-5}, 10^{-8}\}$ and perform 50 iterations of full GMRES (i.e., without restart). The test example is a 3D convection diffusion problem with $n = 63$ discretization points in each mode, with preconditioner $\mathbf{M}$ from equation (5.2) with $q \in \{16, 32\}$.

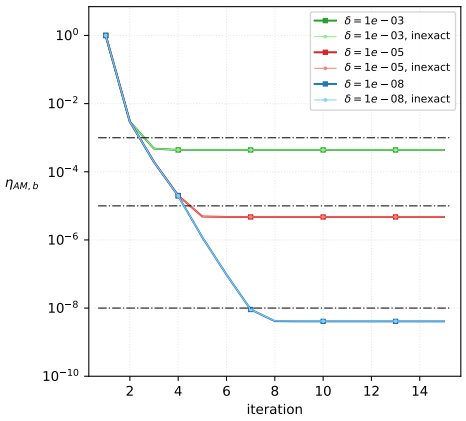


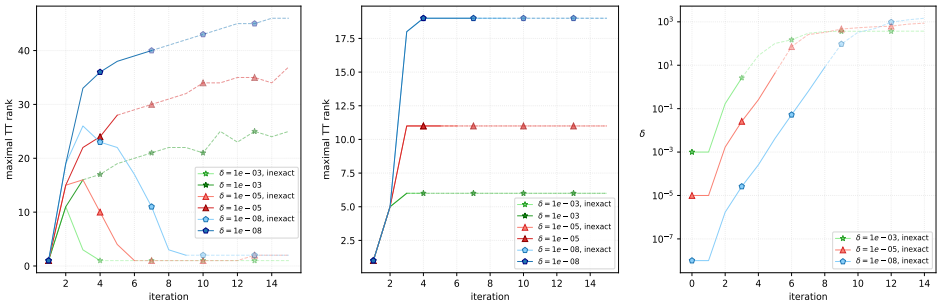
FIG. 5.1. *Convergence history of $\eta_{\mathbf{AM},\mathbf{b}}$ for TT-GMRES and inexact TT-GMRES applied to 3D convection diffusion problem with $n = 63$.*

The primary observation is that TT-GMRES and its inexact variant exhibit very similar convergence behavior, since all convergence histories of $\eta_{\mathbf{AM},\mathbf{b}}$ overlap. Upon examining the convergence history of $\eta_{\mathbf{AM},\mathbf{b}}$, we observe that TT-GMRES with constant TT rounding accuracy inherits the backward stability property of GMRES in matrix computation. Specifically, for each value of δ, the backward error $\eta_{\mathbf{AM},\mathbf{b}}(\mathbf{t}_k)$ decreases and stagnates around δ.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

If δ represents the TT rounding accuracy and $\mathbf{t}_k$ represents the GMRES solution at iteration k, then $\eta_{\mathbf{AM},\mathbf{b}}(\mathbf{t}_k)$ is $\mathcal{O}(\delta)$ since δ is the dominant part of the TT rounding error that occurs during the numerical calculation. Therefore, assuming $\delta \approx \varepsilon$, TT-GMRES can ensure a δ backward stable solution.

Inexact TT-GMRES also succeeds in reducing the backward error to a value close to δ, indicating that it might also be backward stable. The main advantage of this approach is demonstrated in Figure 5.2a, where increasing the TT rounding threshold throughout the iterations results in a significant decrease in the maximum TT rank of the Arnoldi basis vector (i.e., in the memory footprint). As illustrated in Figure 5.2b, the iterative solutions obtained from the two TT-GMRES variants have the same TT rank at each iteration. It is important to note that the TT rank is displayed as a dashed line once the TT-GMRES variants have reached their attainable accuracy. Finally, Figure 5.2c illustrates the evolution of δ_k during the iterations and highlights the significant difference between the TT rounding accuracy initial value and its final one.



(a) Max TT rank of the last Krylov vector.

(b) Max TT rank of the iterative solution.

(c) History of the relaxed δ values.

FIG. 5.2. *Memory request of TT-GMRES and relaxed TT-GMRES applied to the 3D convection–diffusion problem with* $n = 63$.

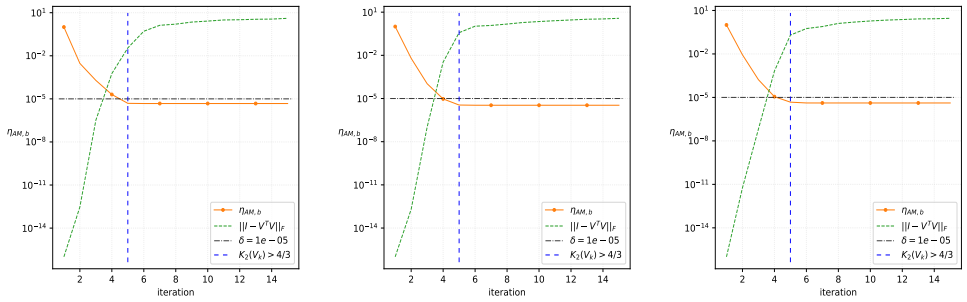
In the next section, we will evaluate some other numerical properties of inexact TT-GMRES, which are identical to those known and theorized for classical GMRES in matrix computation.

5.1.2. Inexact TT-GMRES backward stability: an experimental illustration. As already stated, GMRES is backward stable in matrix computation [24]. Specifically, it is known that when the condition number of the Arnoldi basis $\kappa(V_k)$ exceeds $4/3$, the backward error $\eta_{A,b}$ is close to the machine precision of the working arithmetic. We demonstrate numerically that this property also applies to inexact TT-GMRES. Let $\mathcal{V}_k = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ be the set of TT vectors of the Arnoldi TT basis. The condition number of $\mathcal{V}_k$, $\kappa(\mathcal{V}_k)$, is computed as the condition number of the R factor of the MGS-QR factorization of $\mathcal{V}_k$. Refer to [4] for a description of the MGS-QR factorization of a set of TT vectors.

We test three different grid dimensions for the 3D convection–diffusion problem, namely $n \in \{63, 127, 255\}$, with preconditioner $\mathbf{M}$ from equation (5.2) with $q \in \{16, 32\}$ and a single TT rounding threshold. The convergence history of $\eta_{\mathbf{AM},\mathbf{b}}$ is shown in Figure 5.3. The horizontal dashed-dotted black line represents the TT rounding initial accuracy δ, and the vertical dashed blue line indicates the iteration where $\kappa(\mathcal{V}_k)$ becomes larger than $4/3$.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

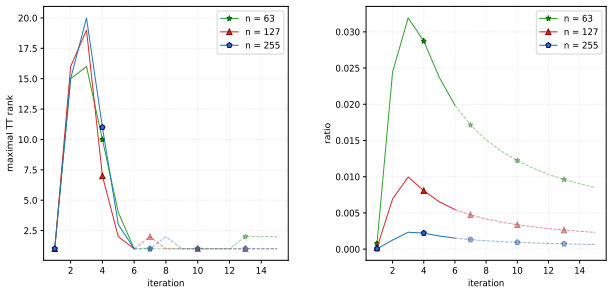
Let $V_k^T V_k$ denote the Gram matrix associated with the Arnoldi basis set, $\mathcal{V}_k$. The loss of orthogonality of the Arnoldi basis, computed as $\|\mathbb{I}_k - V_k^T V_k\|$, is displayed with a dashed green curve in Figure 5.3. Similarly to the theoretical matrix computation result for GMRES, the backward error $\eta_{\mathbf{AM},\mathbf{b}}$ of inexact TT-GMRES reaches an attainable accuracy of $\mathcal{O}(\delta)$ when $\kappa(\mathcal{V}_k) \geq 4/3$ for the three examples. This shows that $\eta_{\mathbf{AM},\mathbf{b}} \leq \delta$ is a robust stopping criterion for inexact TT-GMRES.



(a) $n = 63$. (b) $n = 127$. (c) $n = 255$.

FIG. 5.3. *Convergence history of $\eta_{\mathbf{AM},\mathbf{b}}$ versus loss of orthogonality for the 3D convection–diffusion problem using $\delta = 10^{-5}$.*

Finally, we illustrate once again the memory benefits of the inexact TT-GMRES variant in Figure 5.4. Figure 5.4a shows the maximum TT rank of the last Krylov vector in the basis, while Figure 5.4b shows the memory gain compared to storing the tensor in full format for the entire Arnoldi basis. In the latter plot, for the largest example (i.e., $n = 255$), less than 0.03% of the memory required for a full tensor GMRES computation is necessary when using the inexact TT-GMRES. This illustrates that the curse of dimensionality can be overcome by such a linear solver.



(a) Maximal TT rank of the last Krylov vector.

(b) Compression ratio for the entire Krylov basis.

FIG. 5.4. *The 3D convection–diffusion problem using $\delta = 10^{-5}$.*

We consider only the inexact TT-GMRES variant in the following experiments reported in this paper, since this variant experimentally shows similar numerical behavior to the TT-GMRES with remarkable memory advantages.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

5.2. Solution of parameter-dependent linear operators. This section focuses on four-dimensional PDEs, namely parametric convection–diffusion. The domain of the problem is obtained as a Cartesian product of a three-dimensional space domain and an additional parameter space. The main idea behind this section is to solve for all discrete parameter values simultaneously, resulting in an "all-in-one" solution. The structure of the operator allows for numerical evaluation of the theoretical bounds stated in Section 4.3.

The parametric convection–diffusion problem is defined as

$$\begin{cases} -\alpha\Delta u + 2y(1-x^2)\dfrac{\partial u}{\partial x} - 2x(1-y^2)\dfrac{\partial u}{\partial y} = 0 \quad \text{in} \quad \Omega = [-1,1]^3, \\ u_{\{y=1\}} = 1 \quad \text{and} \quad u_{\partial\Omega\setminus\{y=1\}} = 0. \end{cases}$$

If a grid of n points along each direction of Ω is defined, the final discrete operator of this PDE is $\mathbf{A}_\alpha = \alpha\boldsymbol{\Delta}_3 + \mathbf{D}$, where $\alpha \in [1, 10]$ and $\mathbf{D}$ is defined in equation (5.4). Similarly, the right-hand side $\mathbf{c}_\alpha \in \mathbb{R}^{n\times n\times n}$ depends on the parameter $\alpha \in [1, 10]$ due to the boundary conditions. To solve for multiple discrete values of α, we can tensorize $\boldsymbol{\Delta}_3$ and $\mathbf{D}$ by a diagonal matrix, adding a fourth dimension. This allows us to solve for all the parameter values simultaneously using the tensor operator $\mathbf{A} \in \mathbb{R}^{(p\times p)\times(n\times n)\times(n\times n)\times(n\times n)}$ such that

$$\mathbf{A} = A \otimes \boldsymbol{\Delta}_d + \mathbb{I}_p \otimes \mathbf{D},$$

where $A = \operatorname{diag}(\alpha_1, \dots \alpha_p)$ and $\alpha_i \in [1, 10]$ logarithmically distributed for $i \in \{1, \dots, p\}$. The "all-in-one" problem's right-hand side is represented by $\mathbf{b} \in \mathbb{R}^{p\times n\times n\times n\times n}$, where

$$\mathbf{b}^{[\ell]} = \frac{1}{\|\mathbf{c}_{\alpha_\ell}\|}\mathbf{c}_{\alpha_\ell} \quad \text{for} \quad \ell \in \{1, \dots, p\},$$

using the slice notation introduced in Section 2.1. By construction, $\|\mathbf{b}\| = \sqrt{p}$, which implies that the discrete "all-in-one" problem fits the hypothesis of Propositions 4.2 and 4.5. Note that the "all-in-one" linear operator is constructed directly as a TT matrix from the TT matrix of the single linear system. On the other hand, the "all-in-one" right-hand side is first constructed as a full tensor and then converted into a TT vector.

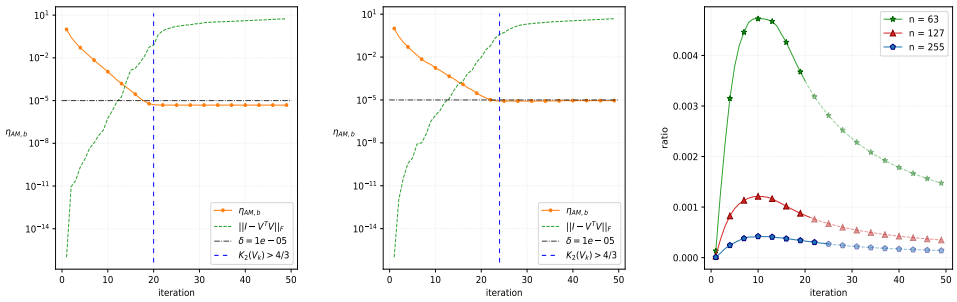
TT-GMRES is utilized to solve the "all-in-one" linear system for $n \in \{63, 127, 255\}$ and $p = 20$. The preconditioner $\overline{\mathbf{M}}$, defined in equation (5.2) with value $q \in \{16, 32\}$, is tensorized with the identity

$$\mathbf{M} = \mathbb{I}_p \otimes \overline{\mathbf{M}}. \tag{5.5}$$

Figures 5.5a and 5.5b show the convergence history of $\eta_{\mathbf{AM},\mathbf{b}}$ and the loss of orthogonality for $n = 127$ and 255. The vertical dashed blue line indicates the iteration $\tilde{k}$ such that $\kappa_2(\mathcal{V}_{\tilde{k}})$ is larger than $4/3$, where $\mathcal{V}_k$ is the set of the TT vectors of the basis of the Krylov space. Figure 5.5c displays the compression ratio for the entire Krylov basis. These findings are consistent with those presented in Section 5.1.2 and confirm the observations made in that section. In conclusion, inexact TT-GMRES appears to be δ backward stable and enables substantial memory savings. Hence, the larger the problem, the greater the savings.

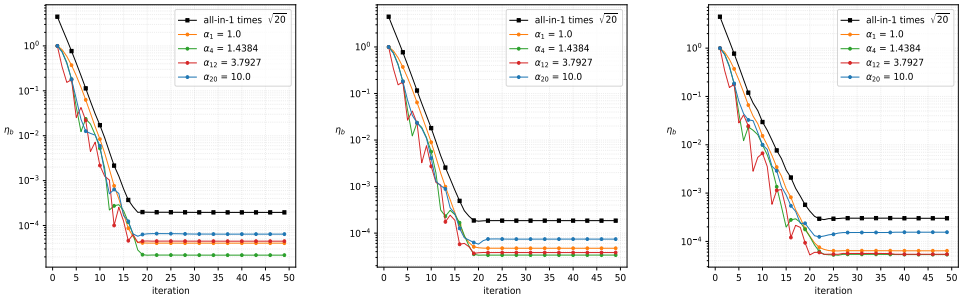
First, we examine the tightness of the bound presented in Proposition 4.1. Figure 5.6 shows the convergence history of η_b. The $\eta_{\mathbf{b}_1}$ curve dominates the others during the first half of the iterations for all values of n. In the optimal case, the difference between $\eta_{\mathbf{b}_\ell}$ and $\eta_{\mathbf{b}}$ is less than one order of magnitude. Although the individual linear systems do not converge similarly, the bound is quite tight during convergence and slightly more pessimistic once convergence is reached. It is also noticeable that the convergence history is monotonic for the "all-in-one" residual, as expected, but not for the individual ones.

ETNA
Kent State University and
Johann Radon Institute (RICAM)



(a) $n = 127$. (b) $n = 255$. (c) Compression ratio of the Krylov basis.

FIG. 5.5. *Convergence history of $\eta_{\mathbf{AM},\mathbf{b}}$ versus loss of orthogonality and compression ratio for the 4D parametric convection–diffusion problem using $\delta = 10^{-5}$.*



(a) $n = 63$. (b) $n = 127$. (c) $n = 255$.

FIG. 5.6. *Convergence history of the $\eta_{\mathbf{b}}$ bound for 4D convection–diffusion parametric operators using $\delta = 10^{-5}$.*

To plot the bound for $\eta_{\mathbf{AM},\mathbf{b}}$ from Proposition 4.2, we define a vector $\upsilon_\ell \in \mathbb{R}^w$. The kth component of υ_ℓ corresponds to the value of the coefficient ρ_ℓ from equation (4.3) evaluated for the solution at the kth iteration, i.e.,

$$\upsilon_\ell(k) = \rho_\ell(\mathbf{t}_k) \quad \text{for every} \quad k \in \{1, \dots, w\},$$

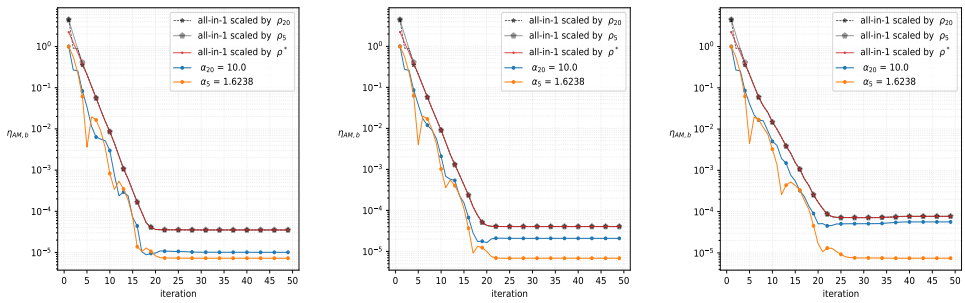
where w is the number of iterations considered. Let ℓ_m and ℓ_M be the parameter indices for which the norm of υ_ℓ is minimal and maximal, respectively, that is,

$$\ell_m = \operatorname*{argmin}_{\ell \in \{1,\dots,p\}} \|\upsilon_\ell\| \quad \text{and} \quad \ell_M = \operatorname*{argmax}_{\ell \in \{1,\dots,p\}} \|\upsilon_\ell\|. \tag{5.6}$$

In our specific case, these indices are equal to 1 and 14, respectively. Figure 5.7 displays $\eta_{\mathbf{AM},\mathbf{b}}(\mathbf{t}_k)$ scaled by ρ_ℓ (see equation (4.3) from Proposition 4.2) and by ρ^* (see equation (4.7) from Corollary 4.3) versus $\eta_{\mathbf{A}_\ell\overline{\mathbf{M}},\mathbf{b}_\ell}(\mathbf{t}_k^{[\ell]})$ for $\ell \in \{1, 14\}$ and for all the values of n. The three scaled curves overlap starting from the third iteration for all the grid dimensions, indicating that the scaling coefficient approximation given by ρ^* is highly accurate in this example. It is observed that the orange curve corresponding to $\eta_{\mathbf{A}_5\overline{\mathbf{M}},\mathbf{b}_5}$ and the blue one for $\eta_{\mathbf{A}_{20}\overline{\mathbf{M}},\mathbf{b}_{20}}$ frequently intersect with a difference of at most one order. Furthermore, the difference between $\eta_{\mathbf{A}_5\overline{\mathbf{M}},\mathbf{b}_5}$ and $\eta_{\mathbf{AM},\mathbf{b}}$ scaled by ρ_5 is less than one order of magnitude in the optimal case, and

ETNA
Kent State University and
Johann Radon Institute (RICAM)

not larger than two in the worst case. Thus, we conclude that the bound of the "all-in-one" for the individual solution is quite tight for this PDE. Note that no extra computation is required to estimate ρ^*, while the norm of $\mathbf{A}_\ell \overline{\mathbf{M}} \mathbf{t}_k^{[\ell]}$ has to be computed to obtain the value of $\rho_\ell(\mathbf{t}_k)$.



(a) $n = 63$. (b) $n = 127$. (c) $n = 255$.

FIG. 5.7. *Convergence history of the $\eta_{\mathbf{AM},\mathbf{b}}$ bound for 4D convection–diffusion parametric operators using $\delta = 10^{-5}$.*

5.3. Solution of parameter-dependent right-hand sides. This subsection aims to illustrate the solution of multiple convection–diffusion problems (5.3) with different right-hand sides. The discretization of the equation (5.3) operator over a Cartesian grid of n points per mode for the domain $\Omega = [0,1]^3$ is denoted $\mathbf{A}_0$. The right-hand-side discretization defined in Section 5.1.2 is represented by $\mathbf{b} \in \mathbb{R}^{n\times n\times n}$. The individual linear system is defined as

$$\mathbf{A}_0 \mathbf{u}_\ell = \mathbf{b} + \mathbf{e}_\ell,$$

where $\mathbf{e}_\ell \in \mathbb{R}^{n\times n\times n}$ is a realization of the normal distribution $\mathcal{N}(0,1)$ for every $\ell \in \{1,\dots,p\}$. To solve the p problems simultaneously, we define the "all-in-one" tensor linear operator $\mathbf{A} \in \mathbb{R}^{(p\times p)\times(n\times n)\times(n\times n)}$ as

$$\mathbf{A} = \mathbb{I}_p \otimes (-\boldsymbol{\Delta}_3),$$

while the "all-in-one" right-hand side is $\mathbf{c} \in \mathbb{R}^{p\times n\times n\times n}$ such that

$$\mathbf{c}(\ell, i_1, i_2, i_3) = \mathbf{b}(i_1, i_2, i_3) + \mathbf{e}_\ell(i_1, i_2, i_3)$$

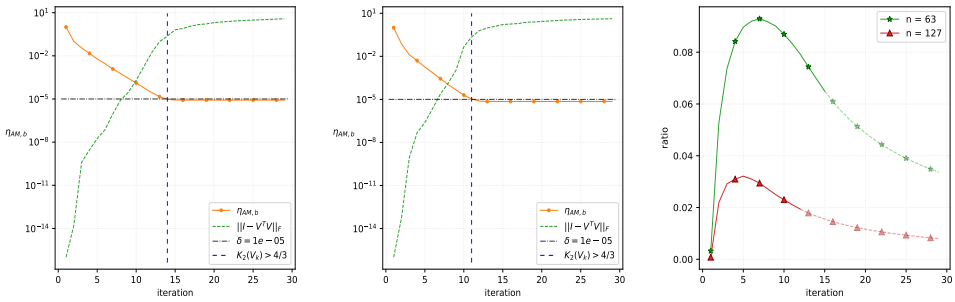
for every $i_k \in \{1,\dots,n_k\}$ and, $\ell \in \{1,\dots,p\}$ for $k \in \{1,\dots,3\}$. The problem is solved for $n \in \{63, 127\}$ and $p = 20$. The preconditioner stated in (5.5) with $q \in \{7, 10\}$ is used, and a small TT rank is imposed to $\mathbf{e}_\ell$, resulting in a maximum TT rank of 11 for $\mathbf{c}$.

Figure 5.8 displays results that confirm, on another example, the observations made in Section 5.1.2 regarding the backward stability and the memory savings of inexact TT-GMRES.

Figure 5.9 illustrates the bound presented in Proposition 4.4 for $\eta_{\mathbf{b}}$. Since all right-hand sides converge simultaneously, the bound is not very tight, and the gap is mostly due to $\sqrt{p}$. The convergence of the individual right-hand sides is monotonic, although this is not guaranteed by any theoretical argument.

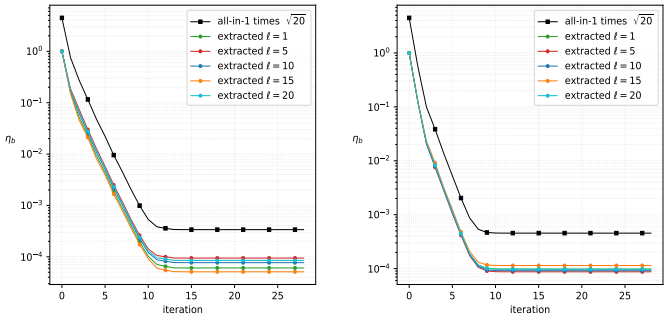
Figure 5.10 shows the bound described in Proposition 4.5, which is quite tight during the first iterations and becomes looser towards the end, differing by more than one order of magnitude. As in the previous section, we calculate ℓ_m and ℓ_M using equation (5.6) to determine which curves to plot in Figure 5.10. The resulting bound, displayed in Figure 5.10, is quite tight, differing by slightly less than one order of magnitude. As previously, the three scaled curves overlap from the second iteration.

ETNA
Kent State University and
Johann Radon Institute (RICAM)



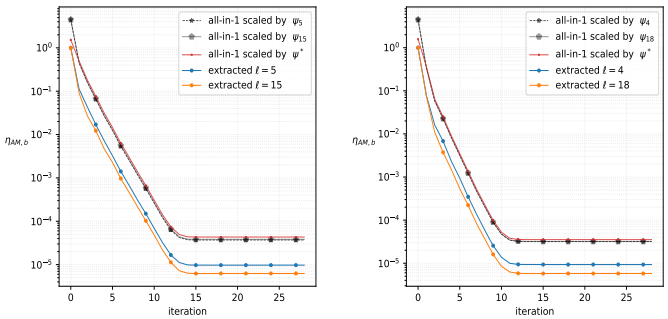
(a) $n = 63$. (b) $n = 127$. (c) Compression ratio of the Krylov basis.

FIG. 5.8. *Convergence history of $\eta_{\mathbf{AM,b}}$ versus loss of orthogonality and compression ratio for the 4D multiple right-hand sides convection–diffusion problem using $\delta = 10^{-5}$.*



(a) Convergence history in $\eta_{\mathbf{b}}$ for $n = 63$. (b) Convergence history in $\eta_{\mathbf{b}}$ for $n = 127$.

FIG. 5.9. *The 4D convection–diffusion problem $\eta_{\mathbf{b}}$ bound using $\delta = 10^{-5}$ and $\varepsilon = 10^{-16}$.*



(a) Convergence history in $\eta_{\mathbf{AM,b}}$ for $n = 63$. (b) Convergence history in $\eta_{\mathbf{AM,b}}$ for $n = 127$.

FIG. 5.10. *The 4D multiple right-hand-sides convection–diffusion problem $\eta_{\mathbf{AM,b}}$ bound using $\delta = 10^{-5}$ and $\varepsilon = 10^{-16}$.*

ETNA
Kent State University and
Johann Radon Institute (RICAM)

6. Concluding remarks. This work addresses the efficient solution of linear systems with tensor product structure using a GMRES algorithm based on low-rank Tensor Train representation. Focusing on mitigating the computational complexity in terms of computation and memory requirements for high-dimensional linear systems, our contributions unfold two key aspects. First, we establish a connection between GMRES in tensor format and its classical matrix counterparts, elucidating the relationship with inexact GMRES theory and a heuristic proposed for GMRES in Tensor Train format. Second, we provide backward error bounds that relate the accuracy of the $(d+1)$-dimensional computed solution to the numerical quality of the sequence of d-dimensional solutions extracted from it. This allows for the prescription of a convergence threshold for the $(d+1)$-dimensional problem that ensures the desired numerical quality of the d-dimensional solutions upon convergence.

Our results are substantiated by academic examples of different dimensions and sizes, which demonstrate the practical applicability and theoretical foundation of our approach. We especially focus on the demonstrated effectiveness of inexact GMRES in the Tensor Train format. Numerically, we observe that it inherits properties established for GMRES in the matrix case. Filling the gap completely, proving the δ backward stability of inexact GMRES in Tensor Train format represents a direction for future research. Furthermore, the inexact TT-GMRES algorithm still carries some intrinsic drawbacks. The use of an efficient preconditioner is crucial to quickly reach the attainable accuracy, as the memory requirement increases with the number of iterations. Therefore, the development of effective preconditioners for multilinear operators remains a challenging open question.

Finally, the theoretical and numerical examples presented in this work focus on the case of a low-rank TT operator that depends on a single parameter. The low-rank assumption is fundamental to ensure the applicability of iterative schemes such as TT-GMRES. If the considered low-rank operator depends linearly on multiple parameters, such as the stationary heat equation with heat conductivity coefficient piecewise constant on several discs from [19], the backward error bounds presented in Sections 4.3 and 4.4 can be generalized. The generalization to multiple parameters is straightforward for $\eta_{\mathbf{b}}$, while more tedious computations are required for $\eta_{\mathbf{A},\mathbf{b}}$. In this framework, it is also possible to develop bounds where only certain parameters of interest are included as variables, while the remaining parameters are kept fixed. The study of these particular bounds could be the subject of future research.

Acknowledgements. The experiments presented in this paper were carried out using the PlaFRIM experimental testbed, supported by Inria, CNRS (LABRI and IMB), Université de Bordeaux, Bordeaux INP and Conseil Régional d'Aquitaine (see https://www.plafrim.fr).

REFERENCES

[1] E. AGULLO, O. COULAUD, L. GIRAUD, M. IANNACITO, G. MARAIT, AND N. SCHENKELS, *The backward stable variants of GMRES in variable accuracy*, Tech. Report RR-9483, Inria Bordeaux Sud-Ouest, Bordeaux, 2022.

[2] J. BALLANI AND L. GRASEDYCK, *A projection method to solve linear systems in tensor format*, Numer. Linear Algebra Appl., 20 (2013), pp. 27–43.

[3] A. BOURAS AND V. FRAYSSÉ, *Inexact matrix–vector products in Krylov methods for solving linear systems: a relaxation strategy*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 660–678.

[4] O. COULAUD, L. GIRAUD, AND M. IANNACITO, *On some orthogonalization schemes in tensor train format*, Tech. Rep. RR-9491, Inria Bordeaux Sud-Ouest, Bordeaux, 2022.

[5] ———, *A robust GMRES algorithm in tensor train format*, Tech. Report RR-9484, Inria Bordeaux Sud-Ouest, Bordeaux, 2022.

[6] L. DE LATHAUWER, B. DE MOOR, AND J. VANDEWALLE, *A multilinear singular value decomposition*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1253–1278.

ETNA
Kent State University and
Johann Radon Institute (RICAM)

[7] S. V. DOLGOV, *TT-GMRES: solution to a linear system in the structured tensor format*, Russian J. Numer. Anal. Math. Modelling, 28 (2013), pp. 149–172.
[8] S. V. DOLGOV AND D. V. SAVOSTYANOV, *Alternating minimal energy methods for linear systems in higher dimensions*, SIAM J. Sci. Comput., 36 (2014), pp. A2248–A2271.
[9] P. GELSS, *The Tensor-Train Format and Its Applications*, PhD. Thesis, Freie Universität Berlin, Berlin, 2017.
[10] L. GIRAUD, S. GRATTON, AND J. LANGOU, *Convergence in backward error of relaxed GMRES*, SIAM J. Sci. Comput., 29 (2007), pp. 710–728.
[11] L. GRASEDYCK, *Hierarchical singular value decomposition of tensors*, SIAM J. Matrix Anal. Appl., 31 (2009/10), pp. 2029–2054.
[12] A. GREENBAUM, *Iterative Methods for Solving Linear Systems*, SIAM, Philadelphia, 1997.
[13] W. HACKBUSCH AND B. N. KHOROMSKIJ, *Low-rank Kronecker-product approximation to multi-dimensional nonlocal operators. I. Separable Approximation of multi-variate functions*, Computing, 76 (2006), pp. 177–202.
[14] ———, *Low-rank Kronecker-product approximation to multi-dimensional nonlocal operators. II. HKT representation of certain operators*, Computing, 76 (2006), pp. 203–225.
[15] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[16] S. HOLTZ, T. ROHWEDDER, AND R. SCHNEIDER, *The alternating linear scheme for tensor optimization in the tensor train format*, SIAM J. Sci. Comput., 34 (2012), pp. A683–A713.
[17] J. DRKOŠOVÁ, M. ROZLOŽNÍK, Z. STRAKOŠ, AND A. GREENBAUM, *Numerical stability of the GMRES method*, BIT Numer. Math., 35 (1995), pp. 309–330.
[18] V. A. KAZEEV AND B. N. KHOROMSKIJ, *Low-rank explicit QTT representation of the Laplace operator and its inverse*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 742–758.
[19] D. KRESSNER AND C. TOBLER, *Low-rank tensor Krylov subspace methods for parametrized linear systems*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 1288–1316.
[20] R. ORÚS, *A practical introduction to tensor networks: matrix product states and projected entangled pair states*, Ann. Physics, 349 (2014), pp. 117–158.
[21] I. V. OSELEDETS, *DMRG approach to fast linear algebra in the TT-format*, Comput. Methods Appl. Math., 11 (2011), pp. 382–393.
[22] ———, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317.
[23] ———, *ttpy*, Software package, 2015. https://github.com/oseledets/ttpy
[24] C. C. PAIGE, M. ROZLOŽNÍK, AND Z. STRAKOŠ, *Modified Gram–Schmidt (MGS), least squares, and backward stability of MGS-GMRES*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 264–284.
[25] D. PALITTA AND P. KÜRSCHNER, *On the convergence of Krylov methods with low-rank truncations*, Numer. Algorithms, 88 (2021), pp. 1383–1417.
[26] J.-L. RIGAL AND J. GACHES, *On the compatibility of a given solution with the data of a linear system*, J. Assoc. Comput. Mach., 14 (1967), pp. 543–548.
[27] M. ROBBÉ AND M. SADKANE, *Exact and inexact breakdowns in the block GMRES method*, Linear Algebra Appl., 419 (2006), pp. 265–285.
[28] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, Philadelphia, 2003.
[29] Y. SAAD AND M. H. SCHULTZ, *GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[30] V. SIMONCINI AND D. B. SZYLD, *Theory of inexact Krylov subspace methods and applications to scientific computing*, SIAM J. Sci. Comput., 25 (2003), pp. 454–477.
[31] C. TOBLER, *Low-rank tensor methods for linear systems and eigenvalue problems*, PhD. Thesis, ETH Zürich, Zürich, 2012.
[32] J. VAN DEN ESHOF AND G. L. G. SLEIJPEN, *Inexact Krylov subspace methods for linear systems*, SIAM J. Matrix Anal. Appl., 26 (2004), pp. 125–153.