A robust GMRES algorithm in Tensor Train format

Olivier Coulaud, Luc Giraud, Martina Iannacito

▶ **To cite this version:**

Olivier Coulaud, Luc Giraud, Martina Iannacito. A robust GMRES algorithm in Tensor Train format. [Research Report] RR-9484, Inria. 2022, pp.1-48. hal-03776529v2

HAL Id: hal-03776529

https://inria.hal.science/hal-03776529v2

Submitted on 21 Sep 2022 (v2), last revised 25 Oct 2022 (v3)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

A robust GMRES algorithm in Tensor Train format

Olivier Coulaud, Luc Giraud, Martina Iannacito

RESEARCH REPORT
N° 9484
September 2022
Project-Team Concace

ISSN 0249-6399 ISRN INRIA/RR--9484--FR+ENG

A robust GMRES algorithm in Tensor Train format

Olivier Coulaud*, Luc Giraud*, Martina Iannacito*

Project-Team Concace

Research Report n° 9484 — September 2022 — 48 pages

Abstract: We consider the solution of linear systems with tensor product structure using a GMRES algorithm. In order to cope with the computational complexity in large dimension both in terms of floating point operations and memory requirement, our algorithm is based on low-rank tensor representation, namely the Tensor Train format. In a backward error analysis framework, we show how the tensor approximation affects the accuracy of the computed solution. With the bacwkward perspective, we investigate the situations where the d-dimensional problem to be solved results from the concatenation of a sequence of $(d-1)$-dimensional problems (like parametric linear operator or parametric right-hand side problems), we provide backward error bounds to relate the accuracy of the d-dimensional computed solution with the numerical quality of the sequence of $(d-1)$-dimensional solutions that can be extracted form it. This enables to prescribe convergence threshold when solving the d-dimensional problem that ensures the numerical quality of the $(d-1)$-dimensional solutions that will be extracted from the d-dimensional computed solution once the solver has converged. The above mentioned features are illustrated on a set of academic examples of varying dimensions and sizes.

Key-words: GMRES, backward stability, Tensor Train format

* Inria, Inria centre at the University of Bordeaux

RESEARCH CENTRE
BORDEAUX – SUD-OUEST

200 avenue de la Vieille Tour
33405 Talence Cedex

Un algorithme GMRES robuste au format tensor train

Résumé : Nous considérons la résolution de systèmes linéaires avec une structure de produit tensoriel en utilisant un algorithme GMRES. Afin de faire face à la complexité de calcul en grande dimension, à la fois en termes d'opérations en virgule flottante et d'exigences de mémoire, notre algorithme est basé sur une représentation tensorielle à faible rang, à savoir le format Tensor Train. Dans un cadre d'analyse d'erreur inverse, nous montrons comment l'approximation tensorielle affecte la précision de la solution calculée. Dans une perspective d'erreur inverse, nous étudions les situations où le problème de dimension d à résoudre résulte de la concaténation d'une séquence de problèmes de dimension $(d-1)$ (comme les problèmes d'opérateurs linéaires paramétriques ou de second membres paramétriques), nous fournissons des bornes d'erreur inverse pour relier la précision de la solution calculée de dimension d à la qualité numérique de la séquence de solutions de dimension $(d-1)$ qui peut être extraite de celle-ci. Cela permet de prescrire un seuil de convergence lors de la résolution du problème à d dimensions qui garantit la qualité numérique des solutions à $(d-1)$ dimensions qui seront extraites de la solution calculée en d dimensions une fois que le solveur aura convergé. Les caractéristiques mentionnées ci-dessus sont illustrées sur un ensemble d'exemples académiques de dimensions et de tailles variables.

Mots-clés : GMRES, backward stabilité, format Tenseur Train

Contents

1 Introduction

In many domains in sciences and engineering, the problems to be solved can naturally be modeled mathematically as d-dimensional linear systems with tensor product structure, i.e., as

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

where $\mathbf{A}$ represents multilinear endomorphism operator on $\mathbb{R}^{n_1 \times \cdots \times n_d}$, $\mathbf{b} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is the right-hand side and $\mathbf{x} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is the searched solution. Two main approaches have emerged in the search for methods to solve high-dimensional linear systems. The first approach is based on optimization techniques mainly based on Alternating Linearised Scheme such as ALS, MALS [14] AMEN [6] and DMRG [20], that break the d-dimensional linear system into low dimension minimization sub-problems, getting an high dimensional solution through an optimization process. The second approach focuses on how to generalize to high-dimensional linear systems iterative methods, as Krylov subspace methods, among which there are conjugate gradient, Generalized Minimal RESidual (GMRES) and biconjugate gradient method [26].

Over the years, different attempts to extend iterative methods from classical matrix linear systems to high dimensional ones have been made, see [5, 17, 2]. However, solving high dimensional linear systems is challenging, since the number of variables grows exponentially with the number of dimensions of the problem. To tackle this phenomenon, known in the tensor linear algebra community as 'curse of dimensionality', there are several compression techniques, as High Order Singular Value Decomposition [4], Hierarchical-Tucker [9] and Tensor-Train (TT) [18]. These compression algorithms provide an approximation at a given accuracy of a given tensor, decreasing the memory footprint, but introducing meanwhile rounding errors. For the solution of such linear systems, the iterative methods have to rely heavily on compression techniques, to prevent memory deficiencies. Consequently, it is fundamental to take into account the effect of tensor rounding errors due to tensor recompression, when evaluating the numerical quality of the solution obtained from an iterative algorithm.

In this work, with a backward error perspective we investigate the numerical performance of the Modified Gram-Schmidt GMRES (MGS-GMRES) [23] for tensor linear systems represented through the TT-formalism. In the classical matrix context, it has been shown that MGS-GMRES is backward stable [21] in the IEEE arithmetic, where 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 pointed out numerically that the MGS-GMRES backward stability holds even when the data representation introduces component-wise or norm-wise perturbations, different from the unit round-off of the finite precision arithmetic. Differently for previously proposed versions of GMRES in tensor format [5], this paper investigates numerically, through many examples, the backward stability of MGS-GMRES for tensor linear systems, where the TT-formalism introduces representation errors bounded by the prescribed accuracy of the computed solution. In particular, we consider the situation where either the right-hand side or the multilinear operator of the d-dimensional system depends on a parameter. The tensor structure enables us to solve simultaneously for many discrete values of the parameter, by simply reformulating the problem in a space of dimension $(d+1)$. We establish theoretical backward error bounds to assess the quality of the d-dimensional solution extracted from the $(d+1)$-dimensional solution. This enables to define the convergence threshold to be used for the solution of the problem of dimension $(d+1)$ that ensures the numerical quality of the d-dimensional solution extracted for the individual problem once MGS-GMRES has converged. We verify the tightness of these bounds through numerical examples. We also investigate the memory consumption of our TT-GMRES algorithm. In particular we observe that, as it could have been expected the memory requirement grows with the number of iterations and with the accuracy of the tensor representation, i.e., how

accurate the tensor approximation is. From the memory viewpoint, MGS-GMRES in TT-format happens to be a suitable backward stable method for solving large high-dimensional systems, if the number of iterations remains reasonable or if a restart approach is considered. In our work, almost all the examples in TT-format are solved with a right preconditioned MGS-GMRES, to satisfy the prescribed tolerance in a small number of iterations. While we spend some words over the quality of the preconditioner, we do not study elaborated restarting techniques.

The remainder of this paper is organized as follows. In Section 2 we introduce the notation. Then we focus on GMRES, presenting the algorithm in a matrix computation framework. After introducing the TT representation, the MGS-GMRES algorithm in TT-format is fully described. Next, in Section 3, we present our approach to solve simultaneously multiple linear systems, which share a common structure. We provide some theoretical results about the quality of the solution extracted from the simultaneous system solution. Numerical experiments are reported in Section 4, where we first illustrate the main features of the solver and compare its robustness to the previous realization of GMRES in TT-format [5]. Then we illustrate the tightness of the bounds derived on Section 3 when solving parameter depend problems formulated in TT format. After summarizing the main results of our work in the conclusion, we investigate further the preconditioner choice and the convergence of problems with the same operator and different right-hand sides solved simultaneously in Appendix A and B respectively. The conclusive Appendix C describes in details the construction of the $(d+1)$-dimensional linear system in TT-format from systems of dimension d.

2 Preliminaries on GMRES and tensors

For ease of reading, we adopt the following notations for the different mathematical objects involved in the description. Small Latin letters stand for scalars and vectors (e.g., a), leaving the context to clarify the object nature. Matrices are denoted by capital Latin letters (e.g., A), tensor by bold small Latin letters (e.g., $\mathbf{a}$), the linear operator between two spaces are calligraphic bold capital letter (e.g., $\boldsymbol{\mathcal{A}}$) and the tensors representation of linear operators by bold capital Latin letters (e.g., $\mathbf{A}$). We adopt the 'Matlab notation' denoting by " : " all the indices along a mode. For example given a matrix $A \in \mathbb{R}^{m\times n}$, then $A(:,i)$ stands for the i-th column of A. The tensor product is denoted by $\otimes$ and Kronecker product by $\otimes_{\mathrm{K}}$, while the Euclidean dot product by $\langle\cdot,\cdot\rangle$ both for vectors and tensors, where it is generalized through the tensor contraction. We denote by $||\cdot||$ the Euclidean norm for vectors and the Frobenius norm for matrix and tensors. Let $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n_1\times\cdots\times n_d} \to \mathbb{R}^{n_1\times\cdots\times n_d}$ be a linear operator on tensor product of spaces and $\mathbf{A} \in \mathbb{R}^{(n_1\times n_1)\times\cdots\times(n_d\times n_d)}$ its tensor representation with respect to the canonical basis, then $||\mathbf{A}||_2$ is the L2 norm of the linear operator $\boldsymbol{\mathcal{A}}$. If $d = 2$, then we have the L2 norm of the matrix associated with a simpler linear operator among two linear vector spaces.

2.1 Preconditioned GMRES

For the solution of a linear system using an iterative solver, it is recommended to used stopping criterion based on a backward error [15, 10, 21]. For iterative schemes, two normwise backward errors can be considered. The iterative scheme will be stopped when the backward error will become lower than a user prescribed threshold; that is, when the current iterate can be considered as the exact solution of a perturbed problem where the relative norm of the perturbation is lower than the threshold. If we denote $Ax = b$ the linear system to be solved a first backward error on $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$ can be considered. We denote $\eta_{A,b}(x_k)$ this normwise backward error

associated with the approximate solution x_k at iteration k, that is defined by [22, 13]

$$\begin{aligned}\eta_{A,b}(x_k) &= \min_{\Delta A,\Delta b} \{\tau > 0 : \|\Delta A\| \leq \tau\|A\|,\ \|\Delta b\| \leq \tau\|b\| \\ &\qquad \text{and } (A+\Delta A)x_k = b + \Delta b\} \\ &= \frac{\|Ax_k - b\|}{\|A\|_2\|x_k\| + \|b\|}. \end{aligned} \tag{1}$$

In some circumstances, a simpler backward error criterion based on perturbations only in the right-hand side can also be considered, that leads to the second possible choice

$$\begin{aligned}\eta_b(x_k) &= \min_{\Delta b} \{\tau > 0 : \|\Delta b\| \leq \tau\|b\| \text{ and } Ax_k = b + \Delta b\} \\ &= \frac{\|Ax_k - b\|}{\|b\|}. \end{aligned} \tag{2}$$

Starting from the zero initial guess, GMRES [23] constructs a series of approximations x_k in Krylov subspaces of increasing dimension k so 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

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

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 $\operatorname{span}\{V_k\} = \mathcal{K}_k(A,b)$ and

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

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

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

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

$$\|\tilde{r}_k\| = \|\beta e_1 - \bar{H}_k y\| = \|b - Ax_k\|. \tag{3}$$

In finite precision calculation, this equality no longer holds but it has been shown that the GMRES method is backward stable with respect to $\eta_{A,b}$ [21] meaning that along the iterations $\eta_{A,b}(x_k)$ might go down-to $\mathcal{O}(u)$ where u is the unit round-off of the floating point arithmetic used to perform the calculations. An overview of GMRES is given in Algorithm 1; we refer to [23, 24] for a more detailed presentation.

Because the orthonormal basis V_k has to be stored, a restart parameter defining the maximal dimension of the search Krylov space is used to control the memory footprint of the solver. If the maximum dimension of the search space is reached without converging, the algorithm is restarted using the final iterate as the initial guess for a new cycle of GMRES. Furthermore, it is often needed to consider a preconditioned to speed-up the convergence. Using right-preconditioned GMRES consists in considering a non singular matrix M, the so-called preconditioner that approximates the inverse of A in some sense. In that case, GMRES is applied to the preconditioned system $AMt = b$. Once the solution t has been computed the solution of the original system is recovered as $x = Mt$. The right-preconditioned GMRES is sketched in Algorithm 2 for a restart parameter `m` and a convergence threshold ε .

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
  $\bar{H}_{k+1,k} = \|w\|$
  $v_{k+1} = w/\bar{H}_{k+1,k}$
  $y_k = \underset{y \in \mathbb{R}^k}{\operatorname{argmin}} \left\| \beta e_1 - \bar{H}_k y \right\|$
  $x_k = V_k y_k$
  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 refinment step with at most m GMRES iterations on AM
  t_k, `hasConverged` = GMRES(AM, r, `m`, ε)
  $x = x + Mt_k$ ▷ Update the unpreconditionned with the computed correction
end while
return: x, `hasConverged`

2.2 The Tensor Train format

Firstly, we describe the main key elements of the Tensor Train (TT) notation for tensors and linear operators between tensor product of spaces. Secondly, we present the advantages in using this formalism to solve linear systems that are naturally defined in high dimension spaces.

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 $k \in \{1,\dots,d\}$. Since 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 $n = \max_{i\in\{1,\dots,d\}}\{n_i\}$, different compression techniques were proposed over the years to reduce the memory consumption [4, 9, 18]. For the purpose of this work the most suitable tensor representation is the *Tensor Train* (TT) format [18]. The key idea of TT is expressing a tensor of order d as the contraction of d tensors of order 3. The contraction is actually the generalization to tensors of the matrix-vector product. Given $\mathbf{a} \in \mathbb{R}^{n_1\times\cdots\times n_h\times\cdots\times n_{d_1}}$ and $\mathbf{b} \in \mathbb{R}^{m_1\times\cdots\times n_h\times\cdots\times m_{d_2}}$, their *tensor contraction* with respect to mode h, denoted $\mathbf{a}\bullet_h\mathbf{b}$, provides a new tensor $\mathbf{c} \in \mathbb{R}^{n_1\times\cdots\times n_{h-1}\times n_{h+1}\times\cdots\times n_{d_1}\times m_1\times\cdots\times m_{h-1}\times m_{h+1}\times\cdots\times m_{d-1}}$ such that its $(i_1,\dots,i_{h-1},i_{h+1},\dots,i_{d_1},j_1\dots,j_{h-1},j_{h+1},\dots,j_{d_2})$ element is

$$\begin{aligned} c &= (\mathbf{a}\bullet_h\mathbf{b})(i_1,\dots,i_{h-1},i_{h+1},\dots,i_{d_1},j_1\dots,j_{h-1},j_{h+1},\dots,j_{d_2}) \\ &= \sum_{i_h=1}^{n_h} \mathbf{a}(i_1,\dots,i_h,\dots,i_{d_1})\mathbf{b}(j_1,\dots,i_h,\dots,j_{d_2}). \end{aligned}$$

The contraction between tensors is linearly extended to more modes. To shorten the notation we omit the bullet symbol and the mode indices when the modes to contract will be clear from the context.

The contraction applies also to the tensor representation of tensor linear operators for computing the operator powers. Let $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n_1\times\cdots\times n_d} \to \mathbb{R}^{n_1\times\cdots\times n_d}$ be a linear operator and let the tensor $\mathbf{A} \in \mathbb{R}^{(n_1\times n_1)\times\cdots\times(n_d\times n_d)}$ be its representation with respect to the canonical basis of $\mathbb{R}^{n_1\times\cdots\times n_d}$. Then the tensor representation with respect to this canonical basis of $\boldsymbol{\mathcal{A}}^2$ is $\mathbf{B} \in \mathbb{R}^{(n_1\times n_1)\times\cdots\times(n_d\times n_d)}$, whose element $b = \mathbf{B}(i_1,j_1,\dots,i_d,j_d)$ is

$$\begin{aligned} b &= \big(\mathbf{A}\ {}_{h_\mathrm{L}}\bullet_{h_\mathrm{R}}\ \mathbf{A}\big)(i_1,j_1,\dots,i_d,j_d) \\ &= \sum_{k_1,\dots,k_d=1}^{n_1,\dots,n_d} \mathbf{A}(i_1,k_1,\dots,i_d,k_d)\mathbf{A}(k_1,j_1,\dots,k_d,j_d) \end{aligned}$$

with $h_\mathrm{L} = \{2,4,\dots,2d\}$ and $h_\mathrm{R} = \{1,3,\dots,2d-1\}$. From this, we recursively obtain the tensor associated with $\boldsymbol{\mathcal{A}}^h$ for $h \in \mathbb{N}$.

The Tensor Train expression 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 called k-th *TT-core* for $k \in \{1,\dots,d\}$, with $r_0 = r_d = 1$. Notice 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 the notation consistency we represent them as tensor. The k-th TT-core of a tensor are denoted by the same bold letter underlined with a subscript k. The value r_k is called *k-th TT-rank*. Thanks to the TT-formalism, the $(i_1,\dots,i_d)$-th element of $\mathbf{x}$ writes

$$\mathbf{x}(i_1,\dots,i_d) = \sum_{j_0,\dots,j_d=1}^{r_0,\dots,r_d} \underline{\mathbf{x}}_1(j_0,i_1,j_1)\underline{\mathbf{x}}_2(j_1,i_2,j_2)\dots\underline{\mathbf{x}}_{d-1}(j_{d-2},i_{d-1},j_{d-1})\underline{\mathbf{x}}_d(j_{d-1},i_d,j_d).$$

Given an index i_k, we denote the i_k-th matrix slice of $\underline{\mathbf{x}}_k$ with respect to mode 2 by $\underline{X}_k(i_k)$, i.e., $\underline{X}_k(i_k) = \underline{\mathbf{x}}_k(:,i_k,:)$. Then each element of the TT-tensor $\mathbf{x}$ can be expressed as the product

of d matrices, i.e.,

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

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 $\underline{X}_d(i_d) \in \mathbb{R}^{r_{d-1} \times 1}$. Remark that $\underline{X}_1(i_1)$ and $\underline{X}_d(i_d)$ are actually vectors, but as before to have an homogeneous notation they write as matrices with a single row or column.

Storing a tensor in TT-format requires $\mathcal{O}(dnr^2)$ units of memory with $n = \max_{i \in \{1,\dots,d\}}\{n_i\}$ and $r = \max_{i \in \{1,\dots,d\}}\{r_i\}$. In this case the memory footprint growths linearly with the tensor order. However to have a significant benefit in the use of this formalism, the value r has to stay bounded and small. A first possible drawback of the TT-format appears with the addition of two TT-tensors. Indeed given two TT-tensors $\mathbf{x}$ and $\mathbf{y}$ with k-th TT-rank r_k and s_k respectively, then the k-th TT-rank of $\mathbf{x} + \mathbf{y}$ is equal to $r_k + s_k$, see [7]. So if $\mathbf{x}$ is a TT-tensor with k-th TT-rank r_k, then tensor $\mathbf{z} = 2\mathbf{x}$ has k-th TT-rank equal to r_k if we simply multiply the first TT-core by 2, but if it is computed as a sum of twice $\mathbf{x}$ then its TT-ranks double. In the following part, we discuss a studied solution to address this issue.

The TT-formalism enables us to express in a compact way also linear operators between tensor product of spaces. Let $\boldsymbol{\mathcal{A}} : \mathbb{R}^{n_1 \times \cdots \times n_d} \to \mathbb{R}^{n_1 \times \dots n_d}$ be a linear operator between tensor product of spaces, fixed the canonical basis 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. Henceforth a tensor associated with a linear operator over tensor product of spaces will be called a *tensor operator*. The TT-representation of tensor operator $\mathbf{A} \in \mathbb{R}^{(n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$, usually called *TT-matrix*, is

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

where $\underline{\mathbf{a}}_k \in \mathbb{R}^{r_{k-1} \times n_k \times n_k \times r_k}$, is its k-th *TT-core*, with $r_0 = r_d = 1$. So its element $a = \mathbf{A}(i_1, j_1, \dots, i_d, j_d)$ is expressed in TT-format as

$$a = \sum_{h_0, \dots, h_d = 1}^{r_0, \dots, r_d} \underline{\mathbf{a}}_1(h_0, i_1, j_1, h_1) \underline{\mathbf{a}}_2(h_1, i_2, j_2, h_2) \cdots \underline{\mathbf{a}}_{d-1}(h_{d-2}, i_{d-1}, j_{d-1}, h_{d-1}) \underline{\mathbf{a}}_d(h_{d-1}, i_d, j_d, h_d).$$

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 $\underline{\mathbf{a}}_k$ for every $i_k, j_k \in \{1, \dots, n_k\}$ and $k \in \{1, \dots, d\}$. Then the last equation is equivalently expressed as

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

As before we estimate the storage cost as $\mathrm{O}(dnmr^d)$ where $n = \max_{i \in \{1,\dots,d\}}\{n_i\}$, $m = \max_{i \in \{1,\dots,d\}}\{m_i\}$ and $r = \max_{i \in \{1,\dots,d\}}\{r_i\}$. However the k-th TT-rank of the contraction of a TT-operator and a TT-vector is equal to the product of the k-th TT-rank of the two contracted objects, see [7]. For example given the TT-operator $\mathbf{A} \in \mathbb{R}^{n_1 \times m_1 \times \cdots \times n_d \times m_d}$ and TT-tensor $\mathbf{x} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ with k-th TT-rank r_k and s_k respectively, their contraction $\mathbf{b} = \mathbf{A}\mathbf{x}$ is a TT-tensor with k-th TT-rank equal to $r_k s_k$.

The TT-rank growth is a crucial point in the implementation of algorithms using TT-tensors: it may lead to run out of memory and prevent the calculation to complete. To address this issue, a rounding algorithm to reduce the TT-rank was proposed in [18]. Given a TT-tensor $\mathbf{x}$ and a relative accuracy δ, the TT-rounding algorithm provides a TT-tensor $\tilde{\mathbf{x}}$ that is at a relative distance δ from $\mathbf{x}$, i.e., $||\mathbf{x} - \tilde{\mathbf{x}}|| \leq \delta ||\mathbf{x}||$. Given a TT-tensor $\mathbf{x} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and setting $r = \max_{i \in \{1,\dots,d\}}\{r_i\}$ and $n = \max_{i \in \{1,\dots,d\}}\{n_i\}$, the computational cost, in terms of floating point operations, of a TT-rounding over $\mathbf{x}$ is $\mathcal{O}(dnr^3)$, as stated in [18].

2.3 Preconditioned GMRES in Tensor Train format

Assume $\mathbf{A} \in \mathbb{R}^{(n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$ to 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

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{4}$$

with $\mathbf{x} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$. Notice that setting $d = 2$ we have the standard linear system from classical matrix computation. A possible way for solving (4) is using a tensor-extended version of GMRES. Since all the operations appearing in this iterative solver are feasible with the TT-formalism, we assume that all the objects are expressed in TT-format. A main drawback in this approach is due to the repetition of sums and contractions in the different loops, which leads to the TT-rank growth and a possible memory over-consumption. Therefore introducing compression steps in TT-GMRES is essential but a particular attention should be paid to the choice of the rounding parameter to ensure that the prescribed GMRES tolerance ε can be reached. Our TT-GMRES algorithm is fully presented in Algorithm 3.

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

1: **input:** $\mathbf{A}$, $\mathbf{b}$, `m`, ε, δ.
2: $\mathbf{r_0} = \mathbf{b}$, $\beta = \|\mathbf{r_0}\|$ and $\mathbf{v}_1 = (1/\beta)\mathbf{r}_0$
3: **for** $k = 1, \ldots,$ `maxit` **do**
4: $\quad \mathbf{w} = \texttt{TT-round}(\mathbf{A}\mathbf{v}_k, \delta)$ ▷ MGS variant
5: $\quad$ **for** $i = 1, \ldots, k$ **do**
6: $\quad\quad \bar{H}_{i,k} = \langle \mathbf{v}_i \, , \mathbf{w} \rangle$
7: $\quad\quad \mathbf{w} = \mathbf{w} - \bar{H}_{i,k}\mathbf{v}_i$
8: $\quad$ **end for**
9: $\quad \mathbf{w} = \texttt{TT-round}(\mathbf{w}, \delta)$
10: $\quad \bar{H}_{k+1,k} = \|\mathbf{w}\|$
11: $\quad \mathbf{v}_{k+1} = (1/\bar{H}_{k+1,k})\mathbf{w}$
12: $\quad y_k = \operatorname{argmin}_{y \in \mathbb{R}^k} \left\| \beta e_1 - \bar{H}_k y \right\|$
13: $\quad \mathbf{x}_k = \texttt{TT-round}\big(\sum_{j=1}^{k+1} y_k(j)\mathbf{v}_j, \delta\big)$
14: $\quad$ **if** $((\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}_k) < \varepsilon)$ **then**
15: $\quad\quad$ `hasConverged = True`
16: $\quad\quad$ **break**
17: $\quad$ **end if**
18: **end for**
19: **return:** $\mathbf{x} = \mathbf{x}_k$, `hasConverged`

In Algorithm 3 and 4 there is an additional input parameter δ, i.e., the rounding accuracy. The TT-rounding algorithm at accuracy δ is applied to the result of the 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, Line 13. The purpose is to balance with the rounding the rank growth due to the tensor contraction or sum that occurred in the immediate previous step. As it will be observed in the numerical experiments of Section 4, the rounding accuracy δ has to be chosen lower or equal than the GMRES target accuracy ε.

3 Solution of parametric problems in Tensor Train format

In this section, we investigate the situation where either the tensor representation of the linear operator or the right-hand side has a mode related to a parameter that is discretized. In the case

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

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

of the parametric linear operator, we are interested into the numerical quality of the computed solutions when we solve for all the parameters at once compared to the solution computed when the parametric systems are treated independently. In the case of the right-hand sides depending on a parameter, we investigate the links between the search space of TT-GMRES enabling the solution of all the right-hand sides at once and the spaces built by the GMRES solver on each right-hand side considered independently. In this subsection, tensor slices play a key role, as consequence we introduce a specific notation. Given a tensor $\mathbf{a} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ in TT-format 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_k-th slice with respect to mode k. Since henceforth we will take slice only with respect to the first mode, instead of writing $\mathbf{a}^{[1,i_1]}$ for the i_1-th slice on the first mode we will simply write $\mathbf{a}^{[i_1]}$. Similarly $\mathbf{A}^{[i_1]}$ denotes the (i_1, i_1)-th slice with respect to mode $(1, 2)$ of a tensor operator $\mathbf{A} \in \mathbb{R}^{(n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$.

3.1 Parameter dependent linear operators

This subsection focuses on a specific type of parametric tensor operators expressed as $\mathbf{A}_\alpha = \mathbf{B}_0 + \alpha \mathbf{B}_1$ with $\alpha \in \mathbb{R}$ and $\mathbf{B}_0, \mathbf{B}_1$ 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

$$\mathbf{A}_\ell \mathbf{x}_\ell = \mathbf{b}_\ell \tag{5}$$

where $\mathbf{A}_\ell = \mathbf{B}_0 + \alpha_\ell \mathbf{B}_1$, $\mathbf{b}_\ell \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, $\alpha_\ell \in [a, b]$ for every $\ell \in \{1, \ldots, p\}$. At this level, it is possible to choose between classically solving each system independently or solving them simultaneously in a higher dimensional space defining the so-called “all-in-one" system. This latter system writes

$$\mathbf{Ax} = \mathbf{b} \tag{6}$$

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

$$\mathbf{A}(h, \ell, i_1, j_1, \ldots, i_d, j_d) = \begin{cases} \mathbf{A}_\ell(i_1, j_1, \ldots, i_d, j_d) & \text{if} \quad h = \ell, \\ 0 & \text{if} \quad h \neq \ell, \end{cases} \tag{7}$$

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

$$\mathbf{b}(\ell, i_1, \ldots, i_d) = \mathbf{b}_\ell(i_1, \ldots, i_d) \tag{8}$$

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 $\mathbf{A}$ writes in a compact format as

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

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

$$\mathbf{A}^{[\ell]} = \mathbf{B}_0 + \alpha_\ell \mathbf{B}_1 = \mathbf{A}_\ell \tag{9}$$

and similarly the ℓ-th slice of $\mathbf{b}$ with respect to the first mode is $\mathbf{b}^{[\ell]} = \mathbf{b}_\ell$ by construction. So that Equation (5) also writes

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

with $x^{[\ell]} = x_\ell$. It shows that, once the "all-in-one" system, Equation (6), has been solved, the solution related to a specific parameter can be extracted as a slice of the "all-in-one" solution, obtaining an *extracted individual solution*. In other words, given the k-th iterate $\mathbf{x}_k$ of the "all-in-one" system, the extracted individual solution for the ℓ-th problem is $\mathbf{x}_k^{[\ell]}$, i.e., the ℓ-th slice with respect to the first mode defined as

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

In the following propositions, we investigate the relation between the backward error of the "all-in-one" system solution and the extracted individual one. The equalities given for the "all-in-one" system are clearly true if the tensor and the tensor operators are given in full format, but they hold also in TT-format. All the details related to the "all-in-one" construction in TT-format are given in Appendix C.

The proven bounds enable us to tune the convergence threshold when solving for multiple parameters while guaranteeing a prescribed quality for the individual extracted solutions. In particular, the bound given by Equation (10) in Proposition 3.1 shows that if a certain accuracy ε is expected for the extracted individual solution in terms of the backward error in (2), a more stringent convergence threshold should be used for the "all-in-one" system solution that should be set to $\varepsilon/\sqrt{p}$.

Proposition 3.1. *Let the "all-in-one" operator* $\mathbf{A} \in \mathbb{R}^{(n_1 \times n_1) \times \ldots (n_d \times n_d) \times (p \times p)}$ *and right-hand side* $\mathbf{b} \in \mathbb{R}^{n_1 \times \cdots \times n_d \times p}$ *be as in Equations* (7) *and* (8) *respectively, 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* (9) *and let* $\mathbf{b}_\ell \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ *be a tensor such that* $||\mathbf{b}_\ell|| = 1$*, that defines the individual linear systems*

$$\mathbf{A}_\ell \mathbf{x}_\ell = \mathbf{b}_\ell$$

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

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

$$\eta_{\mathbf{b}}\big(\mathbf{x}_k\big)\sqrt{p} \geq \eta_{\mathbf{b}_\ell}\big(\mathbf{x}_k^{[\ell]}\big) \tag{10}$$

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

Proof. For the sake of simplicity we use $\eta_{\mathbf{b}}$ and $\eta_{\mathbf{b}_\ell}$ squared throughout the proof. The quantity $\eta^2_{\mathbf{b}_\ell}(\mathbf{x}^{[\ell]})$ explicitly gets

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

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

$$\eta^2_{\mathbf{b}}(\mathbf{x}) = \frac{\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2}{\|\mathbf{b}\|^2}. \tag{11}$$

Thanks to the diagonal structure of $\mathbf{A}$ and the Frobenius norm definition, Equation (11) writes

$$\eta^2_{\mathbf{b}}(\mathbf{x}) = \frac{\sum_{\ell=1}^{n} \|\big(\mathbf{A}\mathbf{x}-\mathbf{b}\big)^{[\ell]}\|^2}{\sum_{k=1}^{p}\|\mathbf{b}^{[k]}\|^2} = \frac{\sum_{\ell=1}^{n}\|\mathbf{A}_\ell\mathbf{x}^{[\ell]}-\mathbf{b}_\ell^2\|}{\sum_{k=1}^{p}\|\mathbf{b}_k\|^2} = \frac{\sum_{\ell=1}^{p}\eta^2_{\mathbf{b}_\ell}(\mathbf{x}^{[\ell]})}{p} \tag{12}$$

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

For the backward error based on perturbation of both the linear operator and the right-hand side defined by (1), a similar result can be derived. While informative this result has a lower practical interest as the term $\rho_\ell(x)$ in (13) depends on the solution; so defining the convergence threshold for the 'all-in-one' solution to guarantee the individual backward error requires some a priori information on the solution norms.

Proposition 3.2. *With the same hypothesis and notation as for Proposition 3.1 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* $\mathbf{A}_\ell\mathbf{x}_\ell=\mathbf{b}_\ell$ *respectively, for every* $\ell\in\{1,\dots,p\}$*, we have*

$$\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x})\,\rho_\ell(\mathbf{x}) \geq \eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) \qquad \textit{where} \qquad \rho_\ell(\mathbf{x}) = \frac{\|\mathbf{A}\|_2\|\mathbf{x}\|+\sqrt{p}}{\|\mathbf{A}_\ell\mathbf{x}^{[\ell]}\|+1}. \tag{13}$$

Proof. The quantity $\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x})$ explicitly writes

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

If the previous equation is multiplied equivalently by $\eta_{\mathbf{b}}(\mathbf{x})$, it gets

$$\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}) = \frac{\|\mathbf{A}\mathbf{x}-\mathbf{b}\|}{\|\mathbf{A}\|_2\|\mathbf{x}\|+\|\mathbf{b}\|}\frac{\eta_{\mathbf{b}}(\mathbf{x})}{\eta_{\mathbf{b}}(\mathbf{x})} = \frac{\|\mathbf{b}\|}{\|\mathbf{A}\|_2\|\mathbf{x}\|+\|\mathbf{b}\|}\eta_{\mathbf{b}}(\mathbf{x}) = \frac{\sqrt{p}}{\|\mathbf{A}\|_2\|\mathbf{x}\|+\sqrt{p}}\eta_{\mathbf{b}}(\mathbf{x}) \tag{14}$$

by 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 function of $\eta_{\mathbf{b}_\ell}(\mathbf{x}^{[\ell]})$ as

$$\eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) = \frac{\|\mathbf{b}_\ell\|}{\|\mathbf{A}_\ell\|_2\|\mathbf{x}^{[\ell]}\|+\|\mathbf{b}_\ell\|}\eta_{\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) = \frac{1}{\|\mathbf{A}_\ell\|_2\|\mathbf{x}^{[\ell]}\|+1}\eta_{\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) \tag{15}$$

since $\|\mathbf{b}_\ell\| = 1$. Multiplying each side of Equation (14) by $(\|\mathbf{A}\|_2\|\mathbf{x}\|+\sqrt{p})$, it follows

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

Thanks to the result of Proposition 3.1, we have

$$\big(\|\mathbf{A}\|_2\|\mathbf{x}\|+\sqrt{p}\big)\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}) = \eta_{\mathbf{b}}(\mathbf{x})\sqrt{p} \geq \eta_{\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) = \big(\|\mathbf{A}_\ell\|_2\|\mathbf{x}^{[\ell]}\|+1\big)\eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) \tag{16}$$

from Equation (15). Dividing both sides of Equation (16) by $\|\mathbf{A}_\ell\|_2\|\mathbf{x}^{[\ell]}\|+1$, it becomes

$$\frac{\|\mathbf{A}\|_2\|\mathbf{x}\|+\sqrt{p}}{\|\mathbf{A}_\ell\mathbf{x}^{[\ell]}\|+1}\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}) \geq \eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) \tag{17}$$

since $\|\mathbf{A}_\ell\|_2\|\mathbf{x}^{[\ell]}\| \geq \|\mathbf{A}_\ell\mathbf{x}^{[\ell]}\|$ by the definition of the L2 norm. □

3.2 Parameter dependent right-hand sides

We consider a particular case of this "all-in-one" approach. We intend to solve p linear systems with the same linear operator and different right-hand sides. Given a linear tensor operator $\mathbf{A}_0 \in \mathbb{R}^{(n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$, we define the ℓ-th linear system as

$$\mathbf{A}_0 \mathbf{x}_\ell = \mathbf{b}_\ell \tag{18}$$

with $\mathbf{b}_\ell \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ for every $\ell \in \{1, \ldots, p\}$. To solve simultaneously all the right-hand sides expressed in Equation (18), we repeat the construction introduced in Subsection 3, except that $\mathbf{A}_0$ is repeated on the 'diagonal' of tensor linear operator $\mathbf{A}$ defined in Equation (7). Thanks to the tensor properties, the tensor operator $\mathbf{A} \in \mathbb{R}^{(n_1 \times n_1) \times \cdots \times (n_d \times n_d) \times (p \times p)}$ writes

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

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, that is $\mathbf{b}^{[\ell]} = b_\ell$. If the initial guess is $\mathbf{x}_0 \in \mathbb{R}^{p \times n_1 \times \cdots \times n_d}$ equal to the null tensor, then at the k-th 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

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

i.e., we seek a tensor $\mathbf{x}_k \in \mathcal{K}_k(\mathbf{A}, \mathbf{b})$ such that

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

Due to the diagonal structure of $\mathbf{A}$, the Frobenius norm of $\mathbf{r}_k = \mathbf{A}\mathbf{x}_k - \mathbf{b}$ is naturally written as follows

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

with, similarly to the previous section, $\mathbf{x}_k^{[\ell]}$ is the ℓ-th slice with respect to the first mode of $\mathbf{x}_k$. Thanks to the diagonal structure of $\mathbf{A}$, we have that the ℓ-th slice of the Krylov basis vector $\mathbf{A}^h\mathbf{b}$ with respect to the first mode is $\mathbf{A}_0^h \mathbf{b}_\ell$. Consequently the ℓ-th slices of the basis vectors of $\mathcal{K}_k(\mathbf{A}, \mathbf{b})$ span the Krylov space $\mathcal{K}_k(\mathbf{A}_0, \mathbf{b}_\ell)$. It means that the individual solutions defined by the slices $\mathbf{x}_k^{[\ell]}$ of the iterate from the "all-in-one" TT-GMRES scheme lie in the same space as the $(\mathbf{x}_\ell)_k$ generated by TT-GMRES applied to the individual systems $\mathbf{A}_0\mathbf{x}_\ell = \mathbf{b}_\ell$ with $(\mathbf{x}_\ell)_0 = 0$. While the two iterates belong to the same space, they are different since the former, $\mathbf{x}_k^{[\ell]}$, is build by minimizing the residual norm of $\mathbf{A}\mathbf{x} - \mathbf{b}$ over $\mathcal{K}_k(\mathbf{A}, \mathbf{b})$ and the latter, $(\mathbf{x}_\ell)_k$, by minimizing the residual norm of $\mathbf{A}_0\mathbf{x}_\ell = \mathbf{b}_\ell$ over $\mathcal{K}_k(\mathbf{A}_0, \mathbf{b}_\ell)$. If we neglect the effect of the rounding, one can expect that

$$\|\mathbf{b}_\ell - \mathbf{A}_0 \mathbf{x}_k^{[\ell]}\| \geq \|\mathbf{b}_\ell - \mathbf{A}_0 (\mathbf{x}_\ell)_k\|$$

Remark 3.3. *We notice that a block TT-GMRES method could also be defined for the solution of such multiple right-hand side problems. In that situation each individual residual norm would be minimized over the same space spanned by the sum of the individual Krylov space. This would be somehow the dual approach to the one described above, where we minimize the sum of the residual norms on each individual Krylov space.*

Regarding the numerical quality of the extracted solution compared to the individually computed solution, the bound stated in Proposition 3.1 is still true. As in the previous section an informative, but with lower practical interest, bound similar Proposition 3.2 can be derived.

Proposition 3.4. *Under the hypothesis of Proposition 3.2, if* $\mathbf{A} = \mathbb{I}_p \otimes \mathbf{A}_0$*, then for* $\eta_{\mathbf{A},\mathbf{v}}(\mathbf{x})$ *and* $\eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}^{[\ell]})$ *associated with the linear systems* $\mathbf{Ax} = \mathbf{b}$ *and* $\mathbf{A}_0\mathbf{x}_\ell = \mathbf{b}_\ell$ *respectively, for every* $\ell \in \{1, \dots, p\}$ *the following inequality holds*

$$\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x})\,\psi_\ell(\mathbf{x}) \geq \eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}^{[\ell]}) \qquad \textit{where} \qquad \psi_\ell(\mathbf{x}) = \frac{\|\mathbf{x}\| + \sqrt{p}/\|\mathbf{A}_0\|_2}{\|\mathbf{x}^{[\ell]}\| + 1/\|\mathbf{A}_0\|_2}. \tag{19}$$

Proof. The result follows from the thesis of Proposition 3.2, since $\|\mathbf{A}\|_2 = \|\mathbf{A}_0\|_2$ □

Corollary 3.5. *Given a sequence of iterative solutions* $\{\mathbf{x}_k\}_{k\in\mathbb{N}}$ *and a value* ν*, 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*

$$\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}_k)\,\rho^*(\mathbf{x}_k) \geq \eta_{\mathbf{A}_\ell,\mathbf{b}_\ell}(\mathbf{x}_k^{[\ell]}) \qquad \textit{where} \qquad \rho^*(\mathbf{x}_k) = \frac{\|\mathbf{A}\|_2\|\mathbf{x}_k\| + \sqrt{p}}{2 - \nu} \tag{20}$$

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

The thesis of Corollary 3.5 is independent of the structure of the operator and consequently remains valid in this multiple right-hand side structure described above.

4 Numerical experiments

In this section we investigate the numerical behaviour of the TT-GMRES solver for linear problems with increasing dimension as it naturally arises in some partial differential equation (PDE) studies. We start by illustrating how the TT-operators of our numerical examples are directly constructed in TT-format, thanks to their peculiarity. For all the examples, we illustrate numerical concerns related to the algorithm convergence and computational costs, with a focus on memory growth and memory saving.

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

$$\begin{aligned}\mathbf{A} =& M_1 \otimes R_2 \otimes R_3 \otimes \cdots \otimes R_{d-2} \otimes R_{d-1} \otimes R_d \\ &+ L_1 \otimes M_2 \otimes R_3 \otimes \cdots \otimes R_{d-2} \otimes R_{d-1} \otimes R_d \\ &+ \cdots + L_1 \otimes L_2 \otimes L_3 \otimes \cdots \otimes L_{d-2} \otimes M_{d-1} \otimes R_d \\ &+ L_1 \otimes L_2 \otimes L_3 \otimes \cdots \otimes L_{d-2} \otimes L_{d-1} \otimes M_d \end{aligned} \tag{21}$$

with $L_k, M_k, R_k \in \mathbb{R}^{n_k \times m_k}$ for every $k \in \{1, \dots, d\}$. As relevant property, these linear operators are expressed in TT-format with TT-rank 2, i.e.,

$$\mathbf{A} = \begin{bmatrix} L_1 & M_1 \end{bmatrix} \otimes \begin{bmatrix} L_2 & M_2 \\ 0 & R_2 \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} L_{d-1} & M_{d-1} \\ 0 & R_{d-1} \end{bmatrix} \otimes \begin{bmatrix} M_d \\ R_d \end{bmatrix} \tag{22}$$

as proved in [16, Lemma 5.1]. Remarking that the general expression of 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 1-dimensional Laplacian using the central-point finite difference scheme with discretization step $h = \frac{1}{n+1}$, i.e.,

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

Then the TT-expression of $\mathbf{\Delta}_d$ is

$$\mathbf{\Delta}_d = \begin{bmatrix} \mathbb{I}_n & \Delta_1 \end{bmatrix} \otimes \begin{bmatrix} \mathbb{I}_n & \Delta_1 \\ \mathbb{0} & \mathbb{I}_n \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} \mathbb{I}_n & \Delta_1 \\ \mathbb{0} & \mathbb{I}_n \end{bmatrix} \otimes \begin{bmatrix} \Delta_1 \\ \mathbb{I}_n \end{bmatrix}. \tag{23}$$

To solve linear systems efficiently, we consider an approximation of the inverse of the discrete Laplacian operator, $\mathbf{M}$, as a preconditioner [11, 12]. This operator writes

$$\mathbf{M} = \sum_{k=-q}^{q} c_k \exp(-t_k\Delta_1) \otimes \cdots \otimes \exp(-t_k\Delta_1) \tag{24}$$

where $c_k = \xi t_k$, $t_k = \exp(k\xi)$ and $\xi = \frac{\pi}{q}$. Thanks to the previously stated property of sum of TT-tensors, we conclude that the TT-ranks of $\mathbf{M}$ will be at least $2q+1$. In Section 4 we consider the linear system $\mathbf{Ax} = \mathbf{b}$ and to speed up its convergence we apply the preconditioner TT-matrix $\mathbf{M}$, effectively solving $\mathbf{AMt} = \mathbf{b}$. 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 rounding compression is further discussed in Appendix A.

To evaluate the converge of the TT-GMRES at the k-th iteration, we display in Section 4 the stopping criterion $\eta_{\mathbf{AM},\mathbf{b}}$, that is

$$\eta_{\mathbf{AM},\mathbf{b}}(\mathbf{t}_k) = \frac{\|\mathbf{AMt}_k - \mathbf{b}\|}{\|\mathbf{AM}\|_2\|\mathbf{t}_k\| + \|\mathbf{b}\|} \tag{25}$$

with $\mathbf{t}_k$ the preconditioned approximated solution at the k-th iteration. We compute exactly the norm of residual, of the right-hand side and of the iterative preconditioned approximated solution. The L2-norm of the preconditioner operator $\mathbf{AM}$ is instead computed by the following sampling approximation. Let $\mathcal{W}$ be a set of normalized TT-vectors generated randomly from a normal distribution, then $\|\mathbf{AM}\|_2$ is approximated by the maximum of the norm of the image of the elements of $\mathcal{W}$ through $\mathbf{AM}$, i.e.,

$$\|\mathbf{AM}\|_2 \approx \max_{\mathbf{w}\in\mathcal{W}} \|\mathbf{AMw}\|.$$

Similarly, the L2-norm of $\mathbf{A}$ is also approximated by $\max\{\|\mathbf{Aw}\| s.t. \mathbf{w} \in \mathcal{W}\}$. Because we are interested in the magnitude of these norms, we keep this norm estimation process simple and only compute 10 random TT-vectors of $\mathcal{W}$.

In order to investigate main numerical features of the GMRES implementation described in the previous section we consider two classical PDEs that are the Poisson and convection-diffusion equation.

The Poisson problem writes

$$\begin{cases} -\Delta u = f \quad \text{in} \quad \Omega = [0,1]^3, \\ u = 0 \quad \text{in} \quad \partial\Omega, \end{cases} \tag{26}$$

where $f : \mathbb{R}^3 \to \mathbb{R}$ is such that the analytical solution of this Poisson problem is $u : [0,1]^3 \to \mathbb{R}$ defined as $u(x,y,z) = (1-x^2)(1-y^2)(1-z^2)$. Let set a grid of n points per mode over Ω, the discretization of the Laplacian over the Cartesian grid is the linear operator $-\boldsymbol{\Delta}_d$ defined in Equation (23) with $d = 3$. Let $\mathbf{b} \in \mathbb{R}^{n\times n\times n}$ be the discrete right-hand side in TT-format such that $\mathbf{b}(i_1, i_2, i_3) = f(x_{i_1}, y_{i_2}, z_{i_3})$.

The convection-diffusion problem, identical to the one considered in [5], writes

$$\begin{cases} -\Delta u + 2y(1-x^2)\dfrac{\partial u}{\partial x} - 2x(1-y^2)\dfrac{\partial u}{\partial y} = 0 \quad \text{in} \quad \Omega = [-1,1]^3\,, \\ \\ u_{\{y=1\}} = 1 \qquad \text{and} \qquad u_{\partial\Omega\setminus\{y=1\}} = 0\,. \end{cases} \tag{27}$$

Setting a grid of n points per mode over $[-1,1]^3$, the Laplacian is discretized as in Equation (23) with $d = 3$. Let $\boldsymbol{\nabla}_x$ be discretization of the first derivative of u with respect to mode 1 defined as $\boldsymbol{\nabla}_x = \nabla_1 \otimes \mathbb{I}_n \otimes \mathbb{I}_n$, similarly $\boldsymbol{\nabla}_y$ is the discrete first derivative with respect to mode 2 written as $\boldsymbol{\nabla}_y = \mathbb{I}_n \otimes \nabla_1 \otimes \mathbb{I}_n$, where ∇_1 is the order-2 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 two components of v are descretized 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 $\mathbf{V}_2 = \text{diag}(-2x) \otimes \text{diag}(1-y^2) \otimes \mathbb{I}_n$. Then the discrete diffusion term $\mathbf{D}$ writes

$$\begin{aligned} \mathbf{D} &= \mathbf{V}_1 \bullet \boldsymbol{\nabla}_x + \mathbf{V}_2 \bullet \boldsymbol{\nabla}_y \\ &= \text{diag}(1-x^2)\nabla_1 \otimes \text{diag}(2y) \otimes \mathbb{I}_n + \text{diag}(-2x) \otimes \text{diag}(1-y^2)\nabla_1 \otimes \mathbb{I}_n\,. \end{aligned} \tag{28}$$

The final operator passed to the TT-GMRES algorithm is $\mathbf{A} = -\boldsymbol{\Delta}_3 + \mathbf{D}$, the right-hand side is the TT-tensor $\mathbf{b} \in \mathbb{R}^{n\times n\times n}$ and the initial guess is the zero TT-tensor $\mathbf{x}_0$. To ensure a fast convergence, similarly to [5], we consider a right preconditioner $\mathbf{M}$ from Equation (24) for this test example.

4.1 Main features and robustness properties

In this section, we first illustrate in Section 4.1.1 the major differences between our GMRES implementation and the one proposed in [5] that mostly hightlights the robustness of our variant. We motivate the need of effective preconditioners in Section 4.1.2 and illustrate the performance and the main features of preconditioned GMRES in Section 4.1.3. All the experiments were performed using `python 3.6.9` and with the tensor toolbox `ttpy 1.2.0` [19].

4.1.1 Comparison with previous tensor GMRES algorithm

In this section we describe the TT-GMRES introduced in [5], that we refer to as relaxed TT-GMRES, that attempts to use advanced features enabled by the inexact GMRES theory [3, 8, 25, 27]. In particular, these inexact GMRES theoretical results show that some perturbations can be introduced in the linear operator when enlarging the Krylov space so that the magnitude of these perturbations can grow essentially as the inverse of the true residual norm of the current iterate. In that context the accuracy of computation of the linear operator can be relaxed, that motivated the use of this terminology in [3, 8]. The inexact GMRES theory assumes exact arithmetic so that Equation (3) holds. In practice, this equality becomes invalid as soon as some loss of orthogonality appears in the Arnoldi basis so that

$$\|\tilde{r}_k\| = \|\beta e_1 - \bar{H}_k y\| \neq \|r_k\| = \|b - Ax_k\|; \tag{29}$$

that is, the norms of the least squares residual and the true residual differ.

In a TT-computational context these inexact Krylov results motivated the heuristic presented in [5], that consists in transferring the perturbation policy from the matrix to the output of the matrix-vector product. More precisely, the variable perturbation magnitude is implemented by varying the rounding threshold δ applied to the tensor resulting from the matrix-vector product along the iterations. Furthermore, the magnitude of the rounding δ is computed using the least squares residual norm rather than the true residual norm for practical computational reasons. A possible consequence of this choice is that δ is somehow artificially increased.

Although the rounding are performed exactly at the same step in the two algorithms, there are two differences between our TT-GMRES and the relaxed TT-GMRES [5]. The first difference is related to the rounding threshold policy that is variable (or relaxed to use the terminology of the pioneer paper on inexact GMRES [3]) and constant in our case. We simply define the value of δ essentially to the value of the target accuracy in terms of backward error (1) ((25) when a preconditioner is used). The second difference is related to the stopping criterion that is defined in terms of backward error (1) in our case ((25) when a preconditioner is used) while it is based on a scaled least squares residual defined by Equation (30) in [5]:

$$\tilde{\eta}_{\mathbf{b}}(\mathbf{x}_k) = \frac{\|\tilde{r}_k\|}{\|\mathbf{b}\|}. \tag{30}$$

Because in practice the true residual differs from the least squares residual, this latter is monotonically decreasing towards zero, such a stopping criterion can lead to an earlier stop.

We choose this stopping criterion based on backward error because it is the one for which, in the matrix framework, GMRES is backward stable in finite precision [21]. Through intensive numerical experiments [1], we observed that our TT-GMRES inherits the same backward stability property. Indeed if δ is the rounding accuracy and $\mathbf{x}_k$ the GMRES solution at iteration k, then $\eta_{\mathbf{A},\mathbf{b}}(\mathbf{x}_k)$ is $\mathcal{O}(\delta)$ as δ is the dominating part of the rounding error occurring during the numerical calculation. Consequently assuming $\delta \leq \varepsilon$, our GMRES variant is able to ensure a ε-backward stable solution. This property is well illustrated in Figure 1 in the case of preconditionned GMRES. The 3d convection-diffusion problem with 63 discretization points is solved using 3 different rounding accuracies, i.e., $\delta \in \{10^{-3}, 10^{-5}, 10^{-8}\}$, and a maximum of 100 iterations. For each value of δ, the backward error $\eta_{\mathbf{AM},\mathbf{b}}(\mathbf{t}_k)$ decreases and stagnates around δ.

The second significant difference between the two GMRES variants is the choice of the rounding threshold along the iterations that is constant for us and varies as the inverse $\|\tilde{\mathbf{r}}_k\|$ in [5]. This variation of the rounding is illustrated in Figure 2. We solve with the two different algorithms the same convection-diffusion problem with 63 discretization points in each space dimension. We select three different rounding accuracies $\delta \in \{10^{-3}, 10^{-5}, 10^{-8}\}$ and perform 100 iterations of

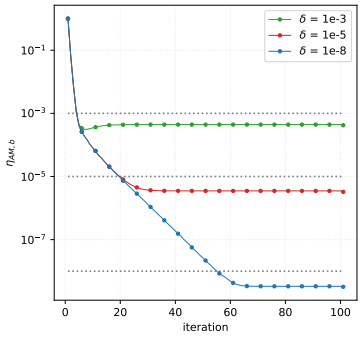


Figure 1: Convergence history of TT-GMRES on a 3-d convection diffusion problem, $n = 64$, for three different rounding accuracies δ

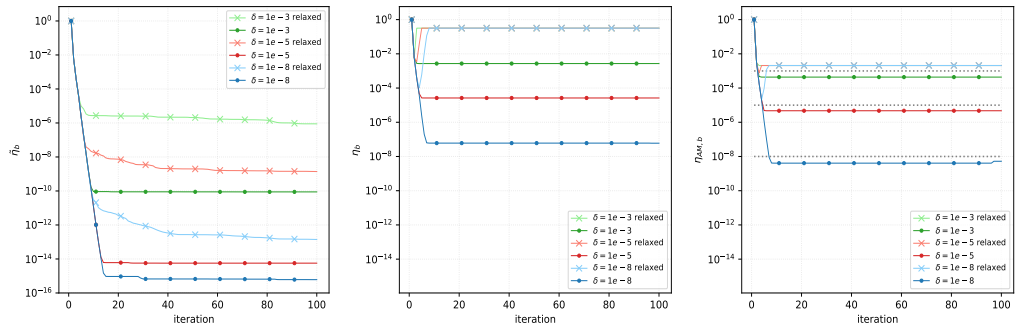
full GMRES (i.e., no restart). In Figure 2f we see the extreme growth of the rounding threshold, when it is scaled by the norm of $\tilde{r}_k$, the least-squares residual norm that becomes smaller and smaller. When the rounding accuracy becomes significantly large, the TT-ranks in relaxed TT-GMRES are cut to 1, losing almost all the information carried in the tensor. Figure 2a shows the scaled residual used as stopping criterion in [5]. We observe that if δ is not relaxed along the iterations, the value of $\tilde{\eta}_{\mathbf{b}}$ decreases extremely quickly, reaching 10^{-10} for $\delta = 10^{-3}$ and at least 10^{-14} for the other rounding accuracies. On the other hand if the rounding accuracy is relaxed during the iterations, we see that in all the cases $\tilde{\eta}_{\mathbf{b}}$ reaches at least 10^{-6}. However, the comparison of Figure 2a and Figure 2b illustrates the numerical difference of the least squares residual norm and the true residual norms given by Equation (29). This comparison reveals that $\tilde{\eta}_{\mathbf{b}}(\mathbf{x}_k)$ with the relaxed δ converges, but $\eta_{\mathbf{b}}(\mathbf{x}_k)$, that is also a backward error as defined in (2), does not. It means that the solutions computed using the relaxed δ are meaningless in terms of backward error accuracy. Similar conclusions can be drawn from Figure 2c that presents the history of $\eta_{\mathbf{AM},\mathbf{b}}$ for the two algorithms. When the rounding accuracy is kept constant, we recover a backward stable behaviour similar to the one proved for finite precision calculation in classical linear system solution in matrix format. Indeed $\eta_{\mathbf{AM},\mathbf{b}}$ always reaches and stagnates around the selected constant value of δ. On the contrary, when δ is relaxed at each iteration, the quantity $\eta_{\mathbf{AM},\mathbf{b}}$ stagnates quickly slightly above 10^{-3}, whatever the starting value of δ. From these two figures, we conclude that relaxing the rounding accuracy and using $\tilde{\eta}_{\mathbf{b}}$ as stopping criterion, together or independently, do not provide any insight on the quality of the computed solution.

Obviously the choice of relaxing the rounding accuracy has a powerful effect on the rank of the last Krylov basis vector and on the solution, as illustrated by Figure 2d and 2e. Indeed in the case of the last Krylov basis vector its TT-rank oscillates around 1 for all the iterations, after the 15-th one approximately. Similarly the solution TT-rank stays equal to 1, after increasing at the very first steps. Unfortunately the computed solutions are numerically meaningless.

In the following, we consider calculation with convergence threshold and rounding accuracy equal to 10^{-5}, that is, $\delta = \varepsilon = 10^{-5}$, with a maximum of 500 iterations and restart $m = 25$.

4.1.2 Poisson problem

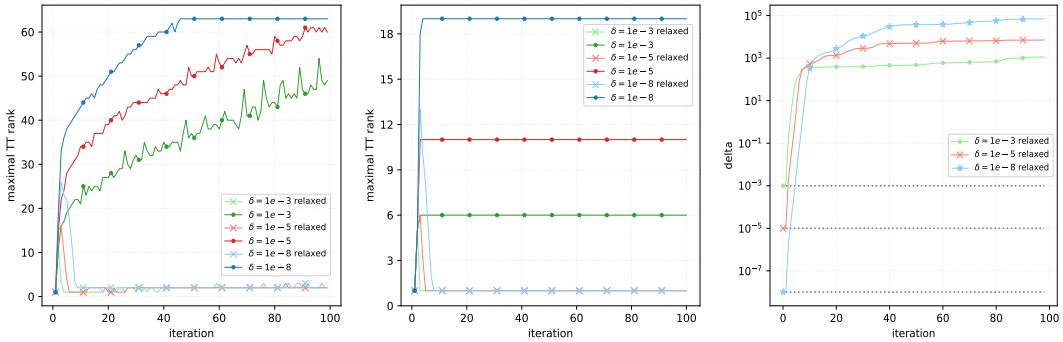
We consider restarted TT-GMRES for the solution of the 3-d Poisson problem with $n \in \{63, 127, 255\}$. Figure 3a shows that the algorithm is able to converge to the prescribed tolerance $\varepsilon = 10^{-5}$ with a number of iterations that increases with the number of discretization points. This high number



(a) Convergence history with $\tilde{\eta}_{\mathbf{b}}$ with least squares residual

(b) Convergence history with $\eta_{\mathbf{b}}$ with true residual

(c) Convergence history $\eta_{\mathbf{AM},\mathbf{b}}$ with true residual



(d) Max TT-rank of the last Krylov vector

(e) Max TT-rank of the iterative solution

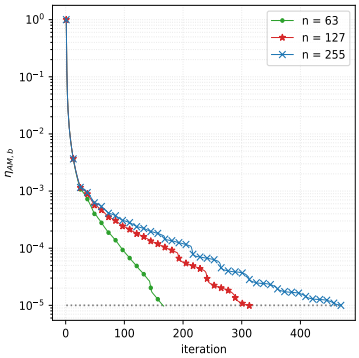
(f) History of the relaxed δ values

Figure 2: TT-GMRES and relaxed TT-GMRES for the solution of 3-d convection diffusion problem with $n = 63$

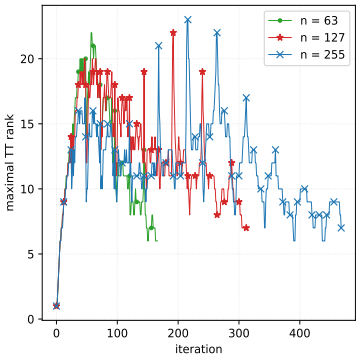
of steps to solve a quite simple PDE motivates the need of a preconditioner. Indeed in general the larger the number of TT-GMRES iterations, the larger the TT-rank growth for the Krylov basis vectors; consequently, the higher the computational cost per iteration. For that example, it can be seen in Figure 3b, that the rank of the current iterate grows significantly during the first iterations (first 100 iterations for $n = 63$ and the first 200 steps for $n \in \{127, 255\}$) before decreasing in a non monotonic way. We infer that this particular behavior is related to the separable nature of the analytical solution, which is

$$u(x, y, z) = \left[\text{diag}(1 - x^2)\right] \otimes \left[\text{diag}(1 - y^2)\right] \otimes \left[\text{diag}(1 - z^2)\right]$$

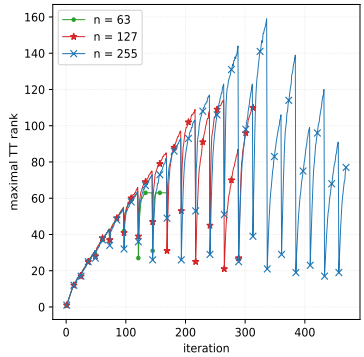
with rank 1 and as consequence its TT-rank is also bounded by 1. After some iterations TT-GMRES seems to capture the main structure of the solution, being able to almost halve the TT-ranks, as it is visible in Figure3b. Another quantity monitored during the iterations is the



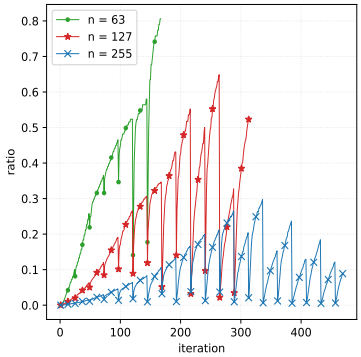
(a) Convergence history



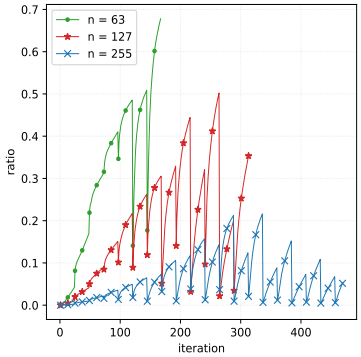
(b) Maximal TT-rank of the iterative solution



(c) Maximal TT-rank of the last Krylov vector



(d) Compression ratio for the last Krylov vector

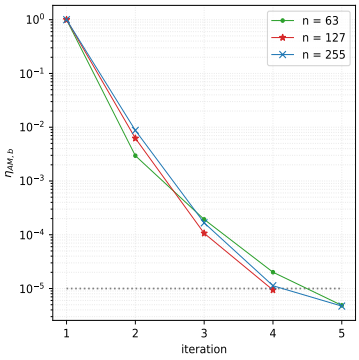


(e) Compression ratio for the entire Krylov basis

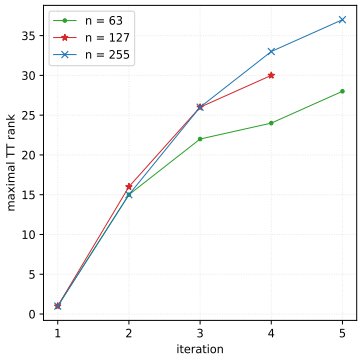
Figure 3: 3-d Poisson problem using $\delta = \varepsilon = 10^{-5}$

growth of the last Krylov vector TT-ranks. In Figure 3c the maximum TT-rank of the last Krylov vector presents a steep increase during a first phase, followed by slight decreasing phase. The behavior of the maximum TT-rank establishes the trend in the compression ratio of the last vector and of the entire basis. Indeed the curves of Figures 3d and 3e are the same of Figure 3c scaled by a constant, equal to n^3 for the first and kn^3 for the second where k is equal to the current iteration in the restart. Lastly in Figure 3c mainly during the second phase, there are many consecutive drops in the maximum TT-ranks which appear with a specific frequency. They are due to the restart after every other 25-th iteration. In fact at restart the new Krylov vector is equal to the normalized rounded residual, whose basic starting TT-ranks is the one of $\mathbf{x}$, equal to 21 at maximum. Lastly notice that in the worst case storing the last Krylov vector and the entire Krylov basis request approximately 80% for $n = 63$ of the memory that would be used for storing entirely them. Furthermore, this ratio decreases when the number of points per mode increases (i.e., $n \in \{63, 127, 255\}$), that is an appealing feature of the TT-format that allows the solution of larger problems for a given memory budget compared to the situation where the full tensors would have to be stored.

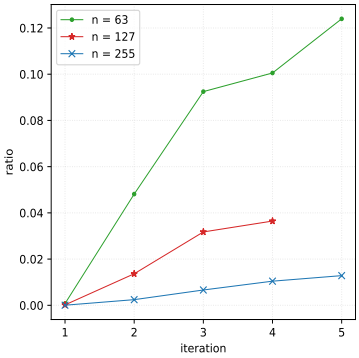
4.1.3 Convection-diffusion



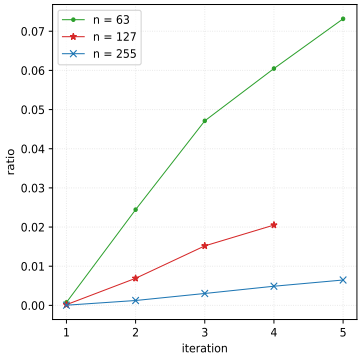
(a) Convergence history



(b) Maximal TT-rank of the last Krylov vector



(c) Compression ratio for the last Krylov vector



(d) Compression ratio for the entire Krylov basis

Figure 4: 3-d Convection diffusion using $\delta = \varepsilon = 10^{-5}$

We test three different grid dimensions, i.e., $n \in \{63, 127, 255\}$, with preconditioner $\mathbf{M}$ from Equation (24) with $q \in \{16, 32\}$. Indeed without it, even with the smallest dimension, TT-GMRES does not converge to the prescribed tolerance $\varepsilon = 10^{-5}$ in a reasonable number of iterations. However using the preconditioner defined in Equation (24), an approximated solution is found in 5 or less iterations, as displayed in Figure 4a. The preconditioner in this case has an extremely strong effect, from which the TT-rank growth and the memory consumption benefit. In Figure 4b the maximum TT-rank exceeds in the worst case the value 35, but to fully interpret this information the compression ratio must be taken into consideration. In fact, Figure 4c shows that in the worst case to store the last Krylov vector in TT-format we use approximately 12% of the memory we would need to store the full tensor. Similarly in Figure 4d we see that storing in TT-format the entire Krylov basis request in the worst case only 7% of the memory that would be used to store the full tensors basis. Although not reported in this document, a more stringent accuracy would require a smaller rounding threshold and consequently a larger memory to store the TT-vectors.

4.2 Solution of parameter dependent linear operators

This section focuses on 4-d PDEs, namely parametric convection-diffusion and stationary heat equations. The domain of both problems is obtained as a Cartesian product of a 3-d space domain and a further parameter space. The common idea for these PDEs is solving for all discrete parameter values simultaneously, getting an "all-in-one" solution. The structure of the operators enables us to check numerically the quality of the theoretical bounds stated in Section3.

4.2.1 Parametric convection diffusion

The parametric convection diffusion problem is a variation of Problem (27), defined as

$$\begin{cases} -\alpha\Delta u + 2y(1-x^2)\dfrac{\partial u}{\partial x} - 2x(1-y^2)\dfrac{\partial u}{\partial y} = 0 \quad \text{in} \quad \Omega = [-1,1]^3\,, \\ \\ u_{\{y=1\}} = 1 \qquad \text{and} \qquad u_{\partial\Omega\setminus\{y=1\}} = 0\,. \end{cases} \tag{31}$$

As in Section 4.1.3 let define a grid of n points along each direction of Ω, then the final discrete operator of this PDE is $\mathbf{A}_\alpha = \alpha\boldsymbol{\Delta}_3 + \mathbf{D}$ with $\alpha \in [1, 10]$ and $\mathbf{D}$ defined in Equation (28). Similarly, the right-hand side $\mathbf{c}_\alpha \in \mathbb{R}^{n\times n\times n}$ depends on the parameter $\alpha \in [1, 10]$ because of the boundary conditions. To solve for multiple discrete values of α, getting an "all-in-one" problem and solution, we tensorize $\boldsymbol{\Delta}_3$ and $\mathbf{D}$ by a diagonal matrices, adding a fourth dimension. The tensor operator for the simultaneous solution is $\mathbf{A} \in \mathbb{R}^{(p\times p)\times(n\times n)\times(n\times n)\times(n\times n)}$ defined as

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

where $A = \mathrm{diag}(\alpha_1, \dots \alpha_p)$ with $\alpha_i \in [1, 10]$ logarithmically distributed for $i \in \{1, \dots, p\}$. The right-hand side of the "all-in-one" problem is $\mathbf{b} \in \mathbb{R}^{p\times n\times n\times n}$ such that

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

using the slice notation introduced in Section 3. By construction $\|b\| = \sqrt{p}$, i.e., the discrete "all-in-one" problem fits into the hypothesis of Proposition 3.2 and 3.4. Remark that the "all-in-one" linear operator is directly constructed as TT-matrix from the TT-matrix of the single linear system, while the "all-in-one" right-hand side is constructed as full tensor and then converted into a TT-vector.

TT-GMRES is used for solving the "all-in-one" linear system for $n \in \{63, 127, 255\}$ and $p = 20$, with the preconditioner defined in Equation (24) with $q \in \{16, 32\}$. Figure 5a shows that the algorithm converges in less than 20 iterations for the first two values of n and in less than 25 for $n = 255$; that is, no restart is needed. For the computational side, Figure 5b displays the maximal TT-rank of the last Krylov vector, which in the worst case in lower than 100. This result translates in terms of memory by a need of slightly more than 4% of the memory that would be required to store the full Krylov vector in the worst case, as highlighted by Figure 5c. Looking at the cost of storing the entire Krylov basis in Figure 5d, we see that TT-format requires around 2% of the memory necessary to store the entire Krylov basis in full tensor format.

We now investigate the tightness of the bound given in Proposition 3.2 and 3.4. Figure 7 shows the quality of the bound for $\eta_{\mathbf{b}_\ell}$ for $\ell \in \{1, \dots, p\}$. For all the values of n, the $\eta_{\mathbf{b}_1}$ curve dominates the other during the first half of the iterations. In the optimal case, the difference between $\eta_{\mathbf{b}_\ell}$ and $\eta_{\mathbf{b}}$ is lower than one order of magnitude. To plot the $\eta_{\mathbf{AM},\mathbf{b}}$ bound from Proposition 3.2, we define a vector $\upsilon_\ell \in \mathbb{R}^k$ whose i-th component corresponds to the value of the coefficient ρ_ℓ from Equation (13) evaluated for the solution at the i-th iteration, i.e.,

$$\upsilon_\ell(i) = \rho_\ell(\mathbf{x}^{[i]}) \qquad \text{for every} \qquad i \in \{1, \dots, k\}$$

with k equal to the iteration number. Let ℓ_m and ℓ_M the parameter index for which the norm of v_ℓ is minimal and maximal respectively, i.e.,

$$\ell_m = \underset{q\in\{1,\dots,p\}}{\operatorname{argmin}} \|v_q\| \quad \text{and} \quad \ell_M = \underset{q\in\{1,\dots,p\}}{\operatorname{argmax}} \|v_q\| \tag{32}$$

which in our specific case are equal to 1 and 14 respectively. In Figure 7 we display in $\eta_{\mathbf{AM},\mathbf{b}}(\mathbf{t}_k)$ scaled by ρ_ℓ (see Equation (13) from Proposition 3.2) and by ρ^* (see Equation (20) from Corollary 3.5) versus $\eta_{\mathbf{A}_\ell\mathbf{M},\mathbf{b}_\ell}(\mathbf{x}_k^{[\ell]})$ for $\ell \in \{1, 14\}$ and for all the values of n.

The three scaled curves overlap from the third iterations for all the grid dimensions, meaning that the approximation of the scaling coefficient given by ρ^* is extremely valid in this example. We see that the orange curve corresponding to $\eta_{\mathbf{A}_5\mathbf{M},\mathbf{b}_5}$ and the blue one for $\eta_{\mathbf{A}_{20}\mathbf{M},\mathbf{b}_{20}}$ intersect frequently, with a difference of one order at most. Moreover the difference between $\eta_{\mathbf{A}_5\mathbf{M},\mathbf{b}_5}$ and $\eta_{\mathbf{AM},\mathbf{b}}$ scaled by ρ_5 is lower than one order of magnitude in the optimal case, while in the worst case it is not larger than two orders. Therefore we conclude that for this PDE the bound of the "all-in-one" for the individual solution is quite tight. Notice that to estimate ρ^* no extra computation is required, while the norm of $\mathbf{A}_\ell\mathbf{x}^{[\ell]}$ has to be computed to get the value of $\rho_\ell(\mathbf{x}^{[\ell]})$.

4.2.2 Heat equation with parametrized diffusion coefficient

We consider the heat equation with parametrized diffusion coefficient studied in [17] and defined as

$$\begin{cases} -\nabla\cdot(\sigma_\theta(x,y,z)\nabla u(x,y,z)) &= 1 \quad \text{in} \quad \Omega = [-1,1]^3\,, \\ u &= 0 \quad \text{in} \quad \partial\Omega. \end{cases} \tag{33}$$

where the coefficient σ_θ being a piece-wise constant function such that

$$\sigma_\theta(x,y,z) = \begin{cases} 1+\theta & \text{in} \quad [-0.5,0.5]^3, \\ 1 & \text{elsewhere,} \end{cases}$$

with $\theta \in [0, 10]$. The function σ_θ, rewritten as $\sigma_\theta(x,y,z) = 1 + \theta\mathbb{1}_\Xi(x,y,z)$ where $\mathbb{1}_\Xi$ is the indicator function of Ξ, provides a linear dependency on θ for the PDE. If Ξ_x is the projection of set Ξ over the x-axis and similarly for Ξ_y and Ξ_z, then $\sigma_\theta(x,y,z) = 1+\theta\mathbb{1}_{\Xi_x}(x)\mathbb{1}_{\Xi_y}(y)\mathbb{1}_{\Xi_z}(z)$. The problem stated in Equation (33) writes equivalently

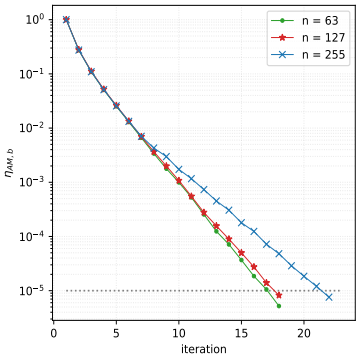
$$\begin{cases} -\Delta u(x,y,z) - \theta\nabla\cdot\Big(\mathbb{1}_{\Xi_x}(x)\mathbb{1}_{\Xi_y}(y)\mathbb{1}_{\Xi_z}(z)\nabla u(x,y,z)\Big) &= 1 \quad \text{in} \quad \Omega = [-1,1]^3\,, \\ u &= 0 \quad \text{in} \quad \partial\Omega. \end{cases} \tag{34}$$

After setting a grid on n points along each direction on Ω, the first term $\mathbf{B}_0$ of the operator in (34) is discretized by the 3-d Laplacian $\mathbf{\Delta}_3$. For the second term $\mathbf{B}_1$, notice that the indicator function $\mathbb{1}_\Xi$ is trivially not differentiable on Ξ boundaries. So it is approximated on the grid points, paying attention to not set them on $\partial\Xi$. The final expression of $\mathbf{B}_1$ is

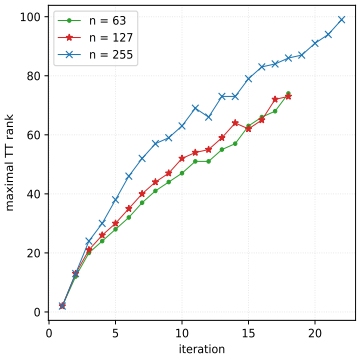
$$\mathbf{B}_1 = D_x\Delta_1\otimes D_y\otimes D_z + D_x\otimes D_y\Delta_1\otimes D_z + D_x\otimes D_y\otimes D_z\Delta_1$$

where Δ_1 is the 1-d discrete Laplacian, $D_x = \operatorname{diag}(\mathbb{1}_{\Xi_{x_i}}) \in \mathbb{R}^{n\times n}$ and similarly for D_y and D_x. Remark that $\mathbf{B}_1$ is a Laplacian-like operator, which is expressed in TT-format according to Equation (21) and (22). The final discrete TT-operator of Problem (33) is

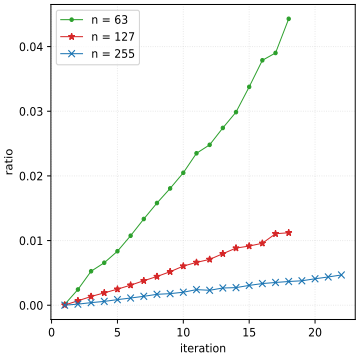
$$\mathbf{A}_\theta = \mathbf{B}_0 + \theta\mathbf{B}_1.$$



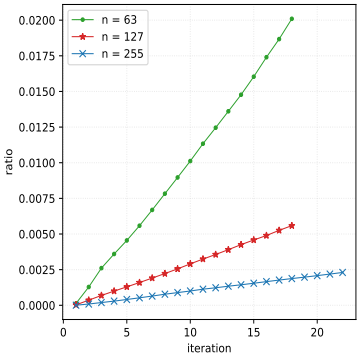
(a) Convergence history



(b) Maximal TT-rank of the last Krylov vector



(c) Compression ratio for the last Krylov vector



(d) Compression ratio for the entire Krylov basis

Figure 5: 4-d Parametric convection diffusion using $\delta = \varepsilon = 10^{-5}$

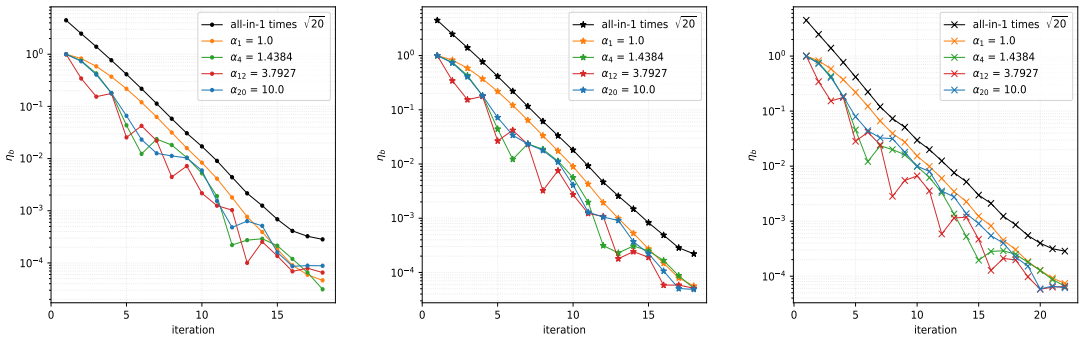
The right-hand side is $\mathbf{c} \in \mathbb{R}^{n \times n \times n}$ such that $\mathbf{c}(i_1, i_2, i_3) = 1$ for $i_k \in \{1, \ldots, n\}$ for $k \in \{1, 2, 3\}$. To study the quality of the bounds expressed in Proposition 3.1 and 3.2, the tensor $\mathbf{c}$ is normalized, i.e., it is scaled by $1/n^3$. Since we want to solve for p values of θ in $[0, 10]$ simultaneously, i.e., we want to solve p-times the discrete Problem (33) for different values of θ, we tensorize $\mathbf{B}_0$ and $\mathbf{B}_1$ by a diagonal matrices, adding a fourth dimension. The tensor discrete operator $\mathbf{A} \in \mathbb{R}^{(p \times p) \times (n \times n) \times (n \times n) \times (n \times n)}$ of the "all-in-one" problem writes

$$\mathbf{A} = \mathbb{I}_p \otimes \mathbf{B}_0 + \Theta \otimes \mathbf{B}_1$$

where $\Theta = \text{diag}(\theta_1, \ldots, \theta_p)$ for $\theta_i \in [0, 10]$ uniformly distributed for $i \in \{1, \ldots, p\}$. The right-hand side of the "all-in-one" problem is

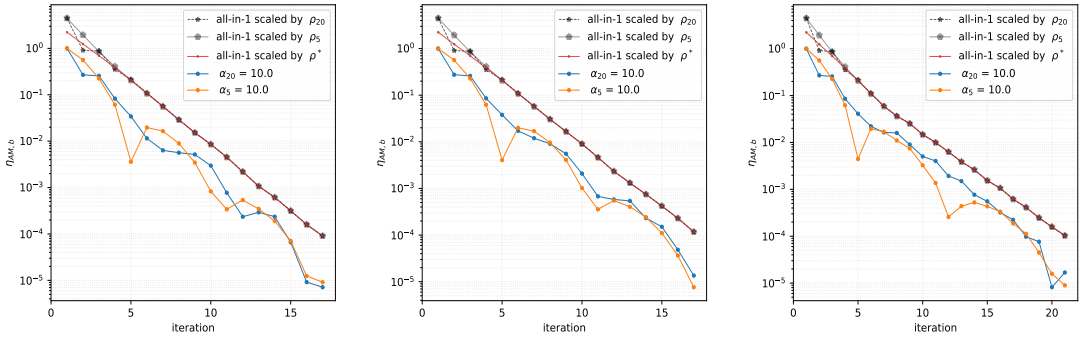
$$\mathbf{b} = \mathbb{I}_p \otimes \mathbf{c}.$$

Remark that since $\|\mathbf{c}\| = 1$ by construction, then $\|\mathbf{b}\| = \sqrt{p}$. We perform experiments with full TT-GMRES (i.e., no restart) for $n \in \{63, 127\}$ and $p = 20$, with the preconditioner defined in Equation (24) with $q \in \{16, 32\}$. Figure 8a shows that TT-GMRES converges to the prescribed



(a) Convergence history in $\eta_{\mathbf{b}}$ for $n = 63$ (b) Convergence history in $\eta_{\mathbf{b}}$ for $n = 127$ (c) Convergence history in $\eta_{\mathbf{b}}$ for $n = 255$

Figure 6: 4-d Parametric convection diffusion $\eta_{\mathbf{b}}$ bound using $\delta = \varepsilon = 10^{-5}$

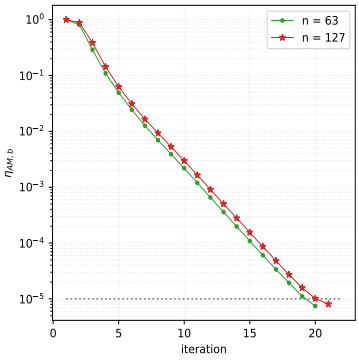


(a) Convergence history in $\eta_{\mathbf{AM},\mathbf{b}}$ for $n = 63$ (b) Convergence history in $\eta_{\mathbf{AM},\mathbf{b}}$ for $n = 127$ (c) Convergence history in $\eta_{\mathbf{AM},\mathbf{b}}$ for $n = 255$

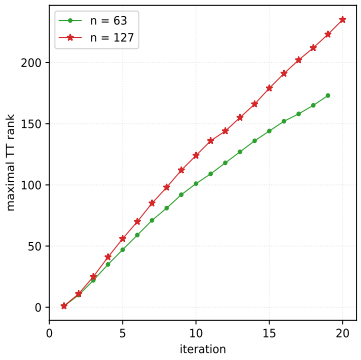
Figure 7: 4-d Parametric convection diffusion $\eta_{\mathbf{AM},\mathbf{b}}$ bound using $\delta = \varepsilon = 10^{-5}$

tolerance in approximately 20 iterations. From the point of view of the memory consumption, in Figure 8*b* we see that for $n = 63$ the maximum TT-rank is lower 200, while for $n = 127$ it is lower than 250. In terms of memory saving, Figure 8c shows that in the worst case we are using only 10% and less than 5% of the memory necessary to store one full tensor of the Krylov basis and the entire full basis respectively.

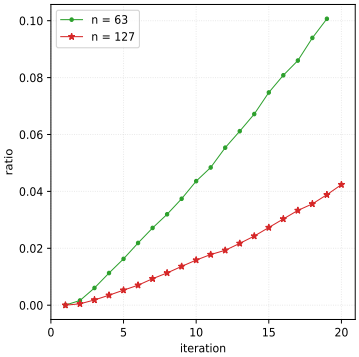
In Figure 9 we have the relation of $\eta_{\mathbf{b}}$ and $\eta_{\mathbf{b}_\ell}$ for $\ell \in \{1, \ldots, p\}$. All the curves present the same shape, with the one associated with $\theta_1 = 0$ being the most peculiar one. We see that in the optimal case the distance between the "all-in-one" curve and the individual ones is lower than one order of magnitude, while in the worst case, realized by $\theta_1 = 0$, the difference is approximately almost of two orders. A similar argument holds for $\eta_{\mathbf{AM},\mathbf{b}}$ bound. As in Section 4.2.1, we compute ℓ_m and ℓ_M, as defined in Equation (32), which are equal to $\ell_m = 20$ and $\ell_M = 1$ respectively. In Figure 10 we see that the two curves $\eta_{\mathbf{A}_\ell\mathbf{M},\mathbf{b}_\ell}$ have a starting and ending overlapping part, while in the internal part they differ by less than one order of magnitude. The three scaled curves for $\eta_{\mathbf{AM},\mathbf{b}}$ overlap from the third iteration. As in the previous studied case, ρ^* from Corollary 3.5 provides a good approximation of the scaling coefficient. In the optimal case the distance is of one order of magnitude approximately, while in the worst one a little more than one order.



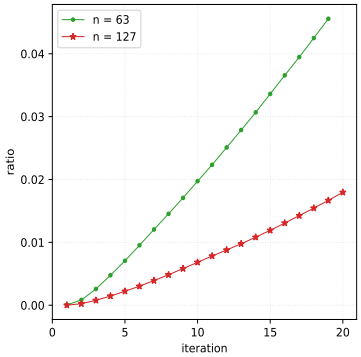
(a) Convergence history



(b) Maximal TT-rank of the last Krylov vector



(c) Compression ration for the last Kyrolv vector



(d) Compression ratio for the entire Krylov basis

Figure 8: 4-d Heterogeneous convection diffusion using $\delta = \varepsilon = 10^{-5}$

4.3 Solution of parameter dependent right-hand sides

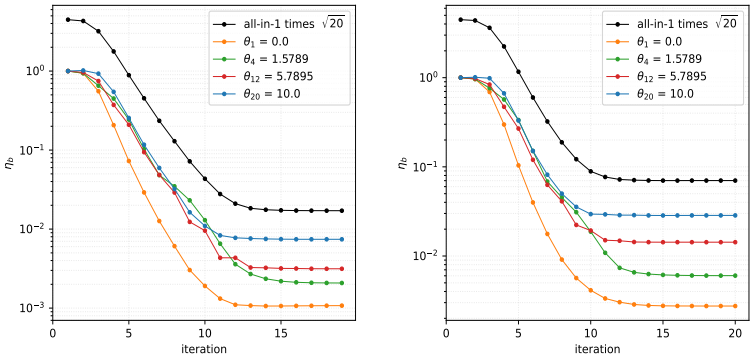
The aim of this section is to investigate the numerical properties of some examples in the context of the multiple right-hand side solution, following the tensorized approach described in 3.

4.3.1 Poisson problem

In this subsection we solve simultaneously multiple Poisson problems stated in Equation (26) with modified right-hand sides. Let $-\boldsymbol{\Delta}_3$ be the discretization of the Laplacian over a Cartesian grid of n points per mode for the domain $\Omega = [0,1]^3$. Let $\mathbf{b} \in \mathbb{R}^{n\times n\times n}$ be the right-hand side discretization defined in Section 4.1.2. We define the individual linear system as

$$-\boldsymbol{\Delta}_3 \mathbf{u}_\ell = \mathbf{b} + \mathbf{e}^{[\ell]}$$

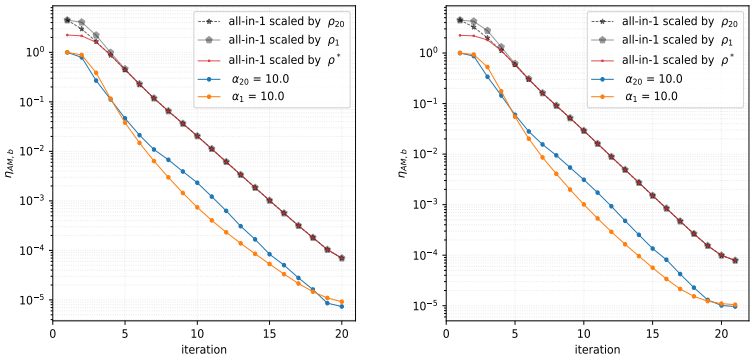
where $\mathbf{e}^{[\ell]} \in \mathbb{R}^{n\times n\times n}$ is the ℓ-th slice with respect to the last mode of $\mathbf{e} \in \mathbb{R}^{n\times n\times n\times p}$ a realization of the normal distribution $\mathcal{N}(0,1)$. Since the aim is solving simultaneously the p problems, as in



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

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

Figure 9: 4-d Heterogeneous convection diffusion $\eta_{\mathbf{b}}$ bound using $\delta = \varepsilon = 10^{-5}$



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

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

Figure 10: 4-d Heterogeneous convection diffusion $\eta_{\mathbf{AM},\mathbf{b}}$ bound using $\delta = \varepsilon = 10^{-5}$

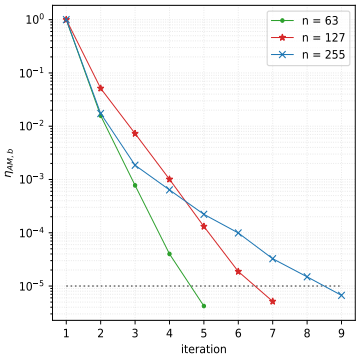
Section 3, we define the “all-in-one" tensor linear operator $\mathbf{A} \in \mathbb{R}^{(n\times n)\times(n\times n)\times(n\times n)}$

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

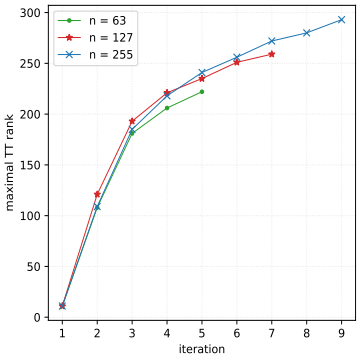
while the “all-in-one" right-hand side is $\mathbf{c} \in \mathbb{R}^{n\times n\times n\times p}$ such that

$$\mathbf{c} = \mathbb{I}_p \otimes \mathbf{b} + \mathbf{e}.$$

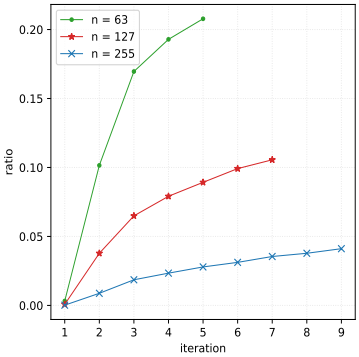
We consider the solution of the problem with $n \in \{63, 127, 255\}$ and $p = 20$. To speed up the convergence we introduce the preconditioner defined in (24) with $q \in \{16, 32\}$. Notice that theoretically the TT-rank of $\mathbf{c}$ may become extremely large, leading to a memory over-consumption and higher computational costs. To face this drawback, we impose a small TT-rank to $\mathbf{e}^{[\ell]}$, so that the TT-rank of $\mathbf{c}$ ends up being 11 at maximum. To study the bounds stated in Section 3, we need to comply with the hypothesis so that we scale each individual right-hand side by its norm, so that $\|\mathbf{c}\| = \sqrt{p}$.



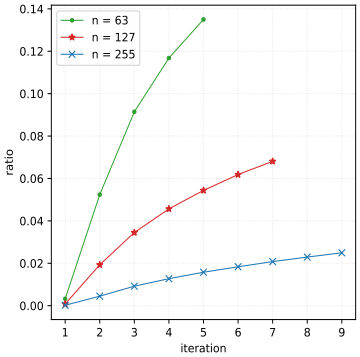
(a) Convergence history



(b) Maximal TT-rank of the last Krylov vector



(c) Compression ration for the last Kyrolv vector



(d) Compression ratio for the entire Krylov basis

Figure 11: 4-d multiple right-hand side Poisson problem using $\delta = \varepsilon = 10^{-5}$

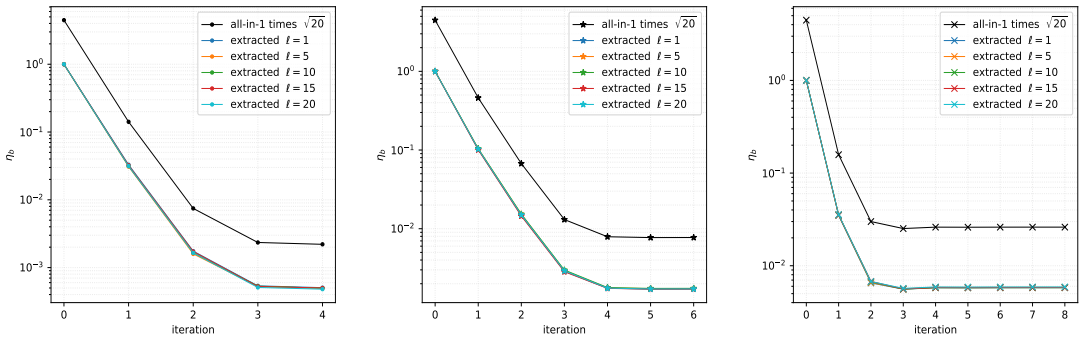
As we can see in Figure 11a, TT-GMRES converges in 5 iterations for $n = 63$, in 7 for $n = 127$ and in 9 for $n = 255$. Figure 11b shows that the TT-rank of the last Krylov vector becomes quickly large, with maximum values ranging from 200 to 300. However looking at Figures 11c and 11d, the compression ratio for a single basis vector and for the entire basis remains extremely small, from 0.05 to 0.2 for the first one and from 0.02 and 0.14 for the entire basis, meaning that the TT approach is still effective from the memory point of view. As in the parametric operator case, we study the bounds expressed in Propositions 3.1 and 3.4. In Figure 12, we see that the bound for $\eta_{\mathbf{b}}$ is always quite tight, around 1 order of magnitude approximately. To use the result of Proposition 3.4, we set k equal to the number of iterations and for every $\ell \in \{1, \dots, p\}$, we define the vector $\gamma_\ell \in \mathbb{R}^k$ such that

$$\gamma_\ell(i) = \psi_\ell(\mathbf{x}^{(i)}) \quad \text{for every} \qquad i \in \{1, \dots, k\}.$$

We define ℓ_m and ℓ_M as the indexes which realize the minimum and the maximum of γ_ℓ norm, i.e.,

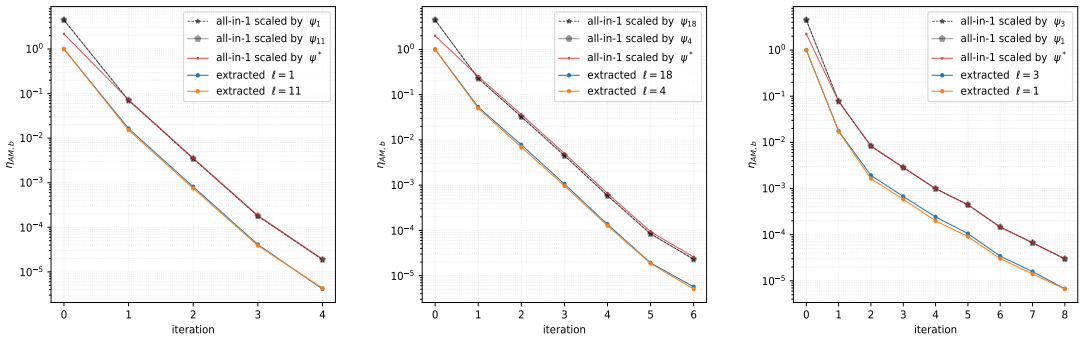
$$\ell_m = \underset{q \in \{1,\dots,p\}}{\operatorname{argmin}} \|\gamma_q\| \quad \text{and} \quad \ell_M = \underset{q \in \{1,\dots,p\}}{\operatorname{argmax}} \|\gamma_q\|. \tag{35}$$

In this specific case for each grid point step, the value of ℓ_m and ℓ_M is reported in Figure 13. The same Figure shows that the bound in this specific case is quite good, with approximately less of 1 order of magnitude of difference, in the optimal and in the worst case. Moreover the three scaled "all-in-one" curves overlap from the second iteration, suggesting again that ρ^* from Corollary 3.5 is a good approximation of the scaling factors.



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

Figure 12: 4-d Poisson problem $\eta_{\mathbf{b}}$ bound using $\delta = \varepsilon = 10^{-5}$



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

Figure 13: 4-d multiple right-hand side Poisson problem $\eta_{\mathbf{AM},\mathbf{b}}$ bound using $\delta = \varepsilon = 10^{-5}$

4.3.2 Convection-diffusion problem

As previously, the aim of this subsection is to illustrate the solution of multiple convection-diffusion problem (27), with different right-hand sides. Let $\mathbf{A}_0$ be the discretization of (27) operator over a Cartesian grid of n points per mode for the domain $\Omega = [0, 1]^3$. Let $\mathbf{b} \in \mathbb{R}^{n \times n \times n \times n}$ be the right-hand side discretization defined in Section 4.1.3. We define the individual linear system 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 $\ell \in \{1,\dots,p\}$. Since the aim is solving simultaneously the p problems, as in Section 3, we define the "all-in-one" tensor linear operator $\mathbf{A} \in \mathbb{R}^{(n\times n)\times(n\times n)\times(n\times n)}$

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

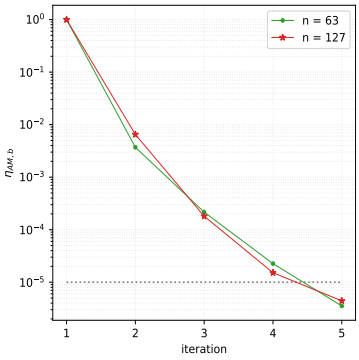
while the "all-in-one" right-hand side is $\mathbf{c} \in \mathbb{R}^{n\times n\times n\times p}$ such that

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

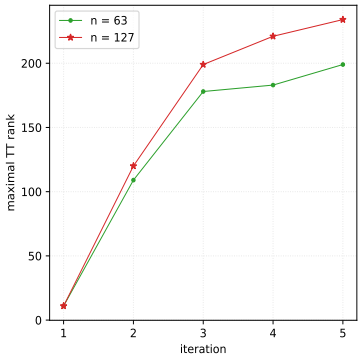
for every $i_k \in \{1,\dots,n_k\}$, $\ell \in \{1,\dots,p\}$ for $k \in \{1,\dots,3\}$. The problem is solved for $n \in \{63,127\}$ and $p = 20$. As in all the previous cases of study, we use the preconditioner stated in (32) with $q \in \{16,32\}$ and we impose a small TT-rank to $\mathbf{e}_\ell$, so that the TT-rank of $\mathbf{c}$ ends up being 11 at maximum.

Figure 14a illustrates the convergence history in 5 iterations for both the grid dimensions. If we compare this Figure with Figure 4a, we observe that the curves are very similar. Generally speaking, the number of iterations for GMRES to converge, neglecting the effect of the rounding, is equal to the number of eigenvectors which span the subspace where the right-hand side lives. This implies that if all the right-hand sides belong to the same linear subspace, the number of iterations necessary to converge is the same, implying that under this hypothesis solving for 1 or p right-hand sides requires the same number of iterations. This point is further discussed in Appendix B. From the point of view of the memory consumption, the comparison of Figure 4b and 14b shows that the solution for 20 right-hand sides leads to TT-rank significantly larger, from 25 to 30 in the single right-hand side solution versus more than 200 for the 20 right-hand side "all-in-one" system. However if we had solved 20 systems independently, summing all the TT-ranks, we could have reached a maximum of 500 up to 700. This becomes more interesting if we compare the compression ratios for the last Krylov vector, looking at Figure 4c and 14c. We have a ratio from 0.02 up to 0.12 for a single right-hand side solution versus 0.1 up to 0.17 for the simultaneous one, which shows that these ratios are extremely closed, considering that in the second case we are solving in a higher dimension. A similar argument holds for the ratio of compression of the entire Krylov basis. In Figure 14d, the ratio is between 0.06 and 0.12, while in Figure 4d it is between 0.01 and 0.07.

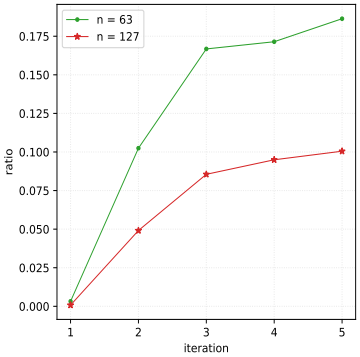
In Figure 15, we present the bound for $\eta_{\mathbf{b}}$ stated in Proposition 3.1. We see that it is quite tight during the first iterations and gets more loose at the end, setting at more than 1 order of difference. As in the previous subsection, we compute ℓ_m and ℓ_M according to Equation (35), deciding which curves are plotted in Figure 16. The resulting bound, displayed in Figure 16, is quite tight, being of slightly less than 1 order of magnitude approximately, with the three scaled curves overlapping from the second iteration.



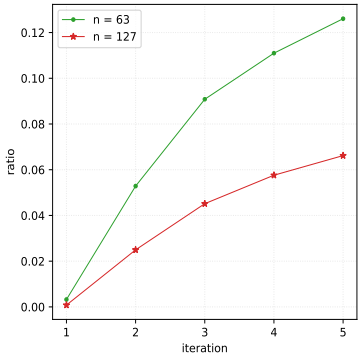
(a) Convergence history



(b) Maximal TT-rank of the last Krylov vector

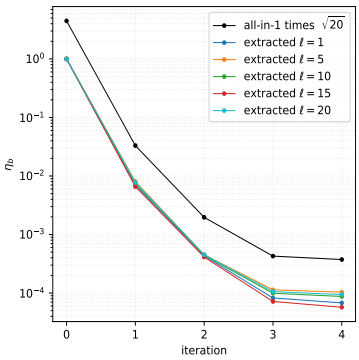


(c) Compression ration for the last Kyrolv vector

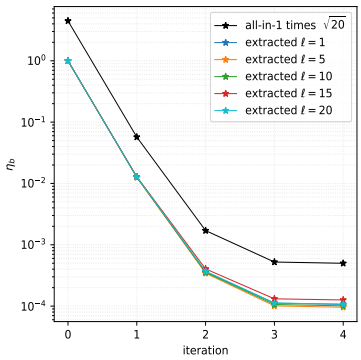


(d) Compression ratio for the entire Krylov basis

Figure 14: 4-d multiple right-hand sides convection-diffusion problem using $\delta = \varepsilon = 10^{-5}$

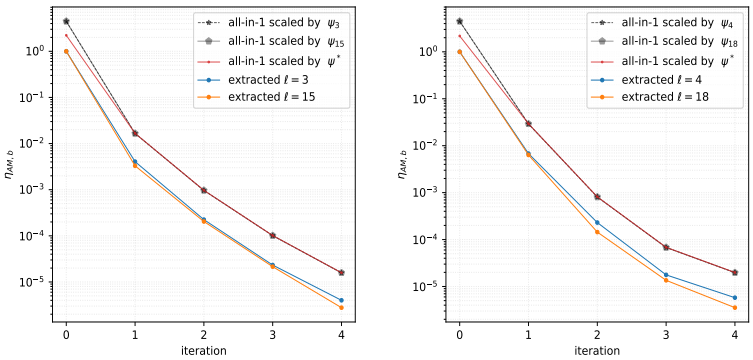


(a) Convergence history in $\eta_{\mathbf{b}}$ for $n = 63$



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

Figure 15: 4-d convection-diffusion problem $\eta_{\mathbf{b}}$ bound using $\delta = \varepsilon = 10^{-5}$



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

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

Figure 16: 4-d multiple right-hand side convection-diffusion problem $\eta_{\mathbf{AM},\mathbf{b}}$ bound using $\delta = \varepsilon = 10^{-5}$

5 Concluding remarks

In this work we proposed a GMRES algorithm for solving high-dimensional liner systems expressed in TT-format and we investigated numerically its backward stability. The examples presented in Section 4 suggest that the backward stability properties observed in the matrix framework still hold true in the tensor context where the recompression (TT-rounding operation) is the dominating part of the computational round-off. Several of these examples enable us to evaluate the tightness of the proposed backward bounds, theoretically proved in Section 2. The existence of these bounds together with the memory requirement illustrate the capabilities of the simultaneous approach when solving parametric tensor linear systems. In Section 4.1.1 we highlight the differences between our algorithm and its previously presented implementation [5], stressing that our approach guarantees the backward stability property of the computed solution. The proposed TT-GMRES algorithm still carries some intrinsic drawbacks. The memory requirement increases with the number of iterations, making crucial the use of an efficient preconditioner. Therefore the development of effective preconditioner for multilinear operators is a challenging open question.

Acknowledgment

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] Emmanuel Agullo, Olivier Coulaud, Luc Giraud, Martina Iannacito, Gilles Marait, and Nick Schenkels. The backward stable variants of GMRES in variable accuracy. Research Report RR-9483, Inria Bordeaux Sud-Ouest, 2022.

[2] Jonas Ballani and Lars Grasedyck. A projection method to solve linear systems in tensor format. *Numerical Linear Algebra with Applications*, 20(1):27–43, 2013.

[3] A. Bouras and V. Frayssé. Inexact matrix-vector products in Krylov methods for solving linear systems: a relaxation strategy. *SIAM Journal on Matrix Analysis and Applications*, 26(3):660–678, 2005.

[4] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.

[5] S. V. Dolgov. TT-GMRES: solution to a linear system in the structured tensor format. *Russian Journal of Numerical Analysis and Mathematical Modelling*, 28(2):149–172, 2013.

[6] Sergey V. Dolgov and Dmitry V. Savostyanov. Alternating minimal energy methods for linear systems in higher dimensions. *SIAM Journal on Scientific Computing*, 36(5):A2248–A2271, 2014.

[7] Patrick Gelß. *The Tensor-Train Format and Its Applications*. PhD thesis, Freien Universitat Berlin, 2017.

[8] L. Giraud, S. Gratton, and J. Langou. Convergence in backward error of relaxed GMRES. *SIAM Journal Scientific Computing*, 29(2):710–728, 2007.

[9] Lars Grasedyck. Hierarchical singular value decomposition of tensors. *SIAM Journal on Matrix Analysis and Applications*, 31(4):2029–2054, 2010.

[10] A. Greenbaum. *Iterative methods for solving linear systems.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1997.

[11] W. Hackbusch and B. N. Khoromskij. Low-rank Kronecker-product approximation to multi-dimensional nonlocal operators. part I. separable approximation of multi-variate functions. *Computing*, 76(3):177–202, Jan 2006.

[12] W. Hackbusch and B. N. Khoromskij. Low-rank kronecker-product approximation to multi-dimensional nonlocal operators. part II. HKT representation of certain operators. *Computing*, 76(3):203–225, Jan 2006.

[13] Nicholas J. Higham. *Accuracy and stability of numerical algorithms.* SIAM, 2005. Second edition.

[14] Sebastian Holtz, Thorsten Rohwedder, and Reinhold Schneider. The Alternating Linear Scheme for Tensor Optimization in the Tensor Train Format. *SIAM Journal on Scientific Computing*, 34(2):A683–A713, jan 2012.

[15] J. Drkošová and M. Rozložník and Z. Strakoš and A. Greenbaum. Numerical stability of the GMRES method. *BIT Numerical Mathematics*, 35:309–330, 1995.

[16] Vladimir A. Kazeev and Boris N. Khoromskij. Low-rank explicit QTT representation of the Laplace operator and its inverse. *SIAM Journal on Matrix Analysis and Applications*, 33(3):742–758, 2012.

[17] Daniel Kressner and Christine Tobler. Low-rank tensor Krylov subspace methods for parametrized linear systems. *SIAM Journal on Matrix Analysis and Applications*, 32(4):1288–1316, 2011.

[18] I. V. Oseledets. Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011.

[19] I. V. Oseledets. ttpy, 2015. https://github.com/oseledets/ttpy.

[20] Ivan V. Oseledets. DMRG approach to fast linear algebra in the TT-format. *Computational Methods in Applied Mathematics*, 11(3):382–393, 2011.

[21] Christopher C. Paige, Miroslav Rozložník, and Zdenvek Strakoš. Modified Gram-Schmidt (MGS), least squares, and backward stability of MGS-GMRES. *SIAM Journal on Matrix Analysis and Applications*, 28(1):264–284, 2006.

[22] J. L. Rigal and J. Gaches. On the compatibility of a given solution with the data of a linear system. *Journal of the ACM*, 14(3):543–548, July 1967.

[23] Youcef Saad and Martin H. Schultz. GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.

[24] Yousef Saad. *Iterative methods for sparse linear systems*, volume 82. SIAM, 2003.

[25] V. Simoncini and D. B. Szyld. Theory of inexact Krylov subspace methods and applications to scientific computing. *SIAM Journal Scientific Computing*, 25:454–477, 2003.

[26] Tobler, Christine. *Low-rank tensor methods for linear systems and eigenvalue problems.* PhD thesis, ETH Zurich, 2012.

[27] J. van den Eshof and G. L. G. Sleijpen. Inexact Krylov subspace methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 26(1):125–153, 2004.

Appendices

A Preconditioner parameter study

In this first appendix the preconditioner firstly introduced in (24) is further investigated. In particular we focus on the effect on the convergence of the number of addends and on the compression accuracy chosen to compute it. As in Subsection 2.2, let $\mathbf{M} \in \mathbb{R}^{(n\times n)\times\cdots\times(n\times n)}$ be the d-order TT-matrix that approximate the inverse of the discrete Laplacian $\mathbf{\Delta}_d$, cf. [11, 12], defined as

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

where $c_k = \eta t_k$, $t_k = \exp(k\eta)$ and $\eta = \pi/q$. As already mentioned, since $\mathbf{M}$ is a sum of tensors, its TT-rank is greater or equal than $2q+1$. The magnitude of TT-ranks of $\mathbf{M}$ conditions the TT-ranks of the Krylov basis vectors and of the final solution. Therefore it is convenient to keep $\mathbf{M}$ TT-rank significantly small, either by reducing the number of addends, i.e., choosing a low value for q, or by compressing $\mathbf{M}$ with an accuracy τ. With the help of a Poisson problem, we illustrate the trade off between number of addends and compression accuracy which leads to the optimal convergence. We consider the Poisson problem, written as

$$\begin{cases} -\Delta u = 1 & \text{in} \quad \Omega = [0,1]^3 \\ u = 0 & \text{in} \quad \partial\Omega \end{cases}$$

so that the effect of the preconditioner is as much evident as possible. Let $-\mathbf{\Delta}_3$ be the discretization of the Laplacian operator over a grid of $n = 63$ points per mode. Similarly let $\mathbf{b} \in \mathbb{R}^{n\times n\times n}$ be a tensor with all the entries equal to 1. Then, setting $\mathbf{A} = -\mathbf{\Delta}_3$, TT-GMRES solves the tensor linear system $\mathbf{Au} = \mathbf{b}$, preconditioning it on the right as

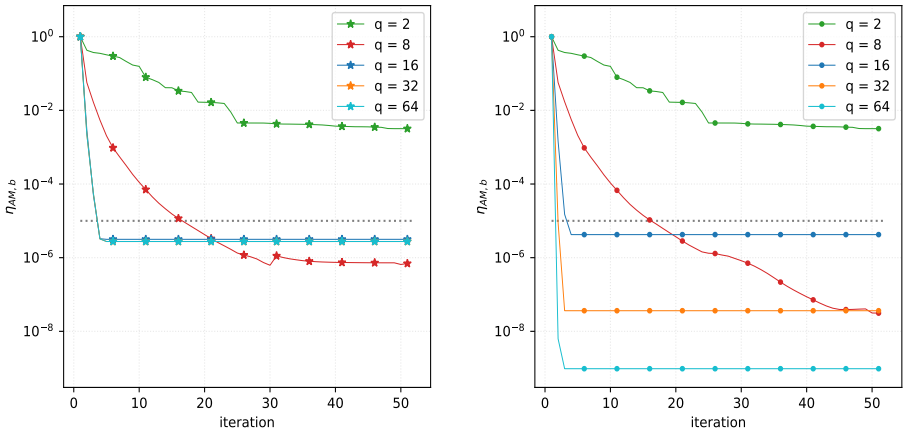
$$\mathbf{AM}_{q,\tau}\mathbf{t} = \mathbf{b}$$

for $q \in \{2, 8, 16, 32, 64\}$ and $\tau \in \{10^{-2}, 10^{-8}\}$. The parameters of TT-GMRES are tolerance $\varepsilon = 10^-16$, rounding accuracy $\delta = 10^{-5}$, dimension of the Krylov space $m = 25$ and a maximum of 2 restart. We set TT-GMRES tolerance equal to the machine precision so that the algorithm performs all the 50 iterations. In Table 1, we report the maximal TT-rank of $\mathbf{M}_{q,\tau}$ and the $\|\mathbf{AM}_{q,\tau}\|_2$ rounded at the third digits for all the combinations of q and τ. Remark that fixed a value for q the L2 norm of the preconditioned linear system is the same up to the third digits for both the values of τ. This seems to suggest that the number of addends plays a key role in determining the quality of the preconditioner, while the rounding accuracy τ affects more significantly the TT-rank, removing kind of unnecessary information. Indeed for $\tau = 10^{-2}$ and $q \geq 8$, the maximal value of the TT-rank is always 5, but depending for an increasing number of addends, the L2 norm gets closer to 1. Similarly for $\tau = 10^{-8}$ and $q \geq 32$, the maximal TT-rank is 15 and the rounded L2 norm is equal to 1.

	$\tau = \mathbf{10^{-2}}$					$\tau = \mathbf{10^{-8}}$				
q	**2**	**8**	**16**	**32**	**64**	**2**	**8**	**16**	**32**	**64**
Max TT-rank of $\mathbf{M}$	2	5	5	5	5	2	7	13	15	15
L2 norm of $\mathbf{AM}_{q,\tau}$	0.012	0.276	0.949	1.00	1.00	0.012	0.276	0.949	1.00	1.00

Table 1: Preconditioner properties for grid step $n = 63$.

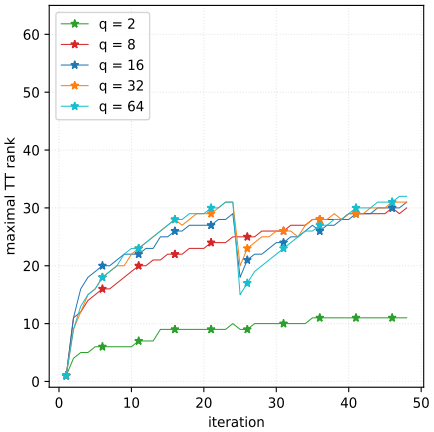
Looking at the convergence history in Figures 17a and 17b, a value of $q \geq 16$ is already sufficient to reach in a very low number of iterations the bound 10^{-5}, due to the TT-GMRES rounding value δ. Figure 17b shows clearly that keeping more information in the preconditioner, TT-GMRES may reach very low levels. However in Figure 17d we observe the side effect of more information. The TT-rank of the last Krylov vector increases significantly for very accurate preconditioner. More precisely, comparing Figures 17c and 17d, the TT-rank of the last Krylov vector doubles if the preconditioner is more accurately rounded. Notice also that in Figure 17c, the TT-rank for $q \in \{16, 32, 64\}$ is almost the same. For the solution viewpoint, the rounding accuracy chosen for the preconditioner has not a big impact on its TT-rank. Indeed, as plotted in Figure 17e and 17f, for both the values of τ and for all $q \geq 8$, the TT-rank of the solution is equal to 5, while only for $q = 2$ it increases, meaning that only 5 addends are not sufficient to speed up the discrete Laplacian convergence.



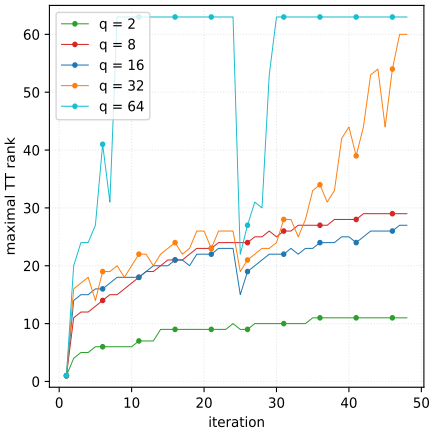
(a) Convergence history for $\tau = 10^{-2}$ and rounding $\delta = 10^{-5}$

(b) Convergence history for $\tau = 10^{-8}$ and rounding $\delta = 10^{-5}$

Figure 17: 3-d Poisson problem, comparing preconditioners

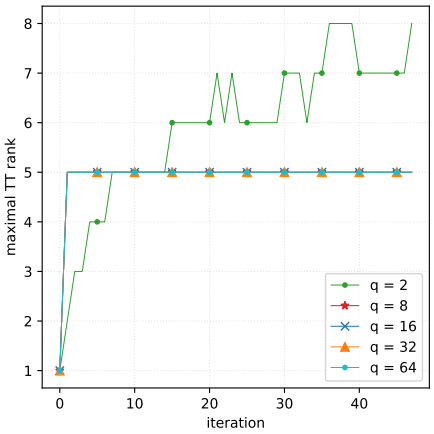


(c) Maximal TT-rank of the last Krylov vector for $\tau = 10^{-2}$

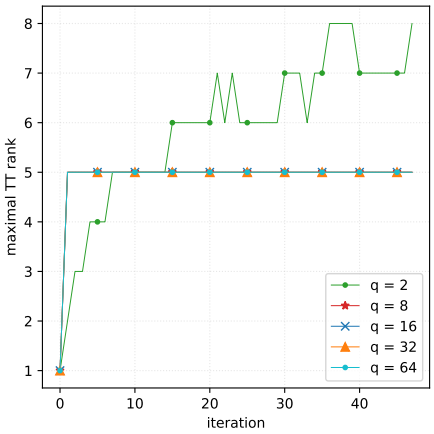


(d) Maximal TT-rank of the last Krylov vector for $\tau = 10^{-8}$

Figure 17: 3-d Poisson problem, comparing preconditioners



(e) Maximal TT-rank of the iterative solution for $\tau = 10^{-2}$



(f) Maximal TT-rank of the iterative solution for $\tau = 10^{-8}$

Figure 17: 3-d Poisson problem, comparing preconditioners

B Multiple right-hand sides: a focus on eigenvectors

In this appendix we study further the convergence of a multiple right-hand side problem. Indeed comparing the convergence history of the convection-diffusion problem, see Subsection 4.1.3 and of the multiple right-hand side convection-diffusion problem discussed in 4.3.2, notice that the number of iterations necessary to converge is equal, 5 in both cases. This appendix explains the causes of the phenomenon.

As we already explained, given a (tensor) linear system $\mathbf{Ax} = \mathbf{b}$ and the null tensor as initial guess, at the j-th iteration GMRES minimizes the norm of the residual $\mathbf{r}^{(j)} = \mathbf{Ax} - \mathbf{b}$ on the Krylov space of dimension j defined as

$$\mathcal{K}_j(\mathbf{A}, \mathbf{b}) = \text{span}\{\mathbf{b}, \mathbf{Ab}, \dots, \mathbf{A}^{j-1}\mathbf{b}\}$$

where $\mathbf{A}^h$ is obtained from h contractions over the indexes $(1, 3, \dots, 2d-1, 2d+1)$ of tensor operator $\mathbf{A}$. If $\mathbf{b}$ is equal to $\mathbf{e}_i$ an eigenvector of the tensor operator $\mathbf{A}$, then the Krylov space writes

$$\begin{aligned}\mathcal{K}_j(\mathbf{A}, \mathbf{b}) &= \text{span}\{\mathbf{b}, \mathbf{Ab}, \dots, \mathbf{A}^{j-1}\mathbf{b}\} \\ &= \text{span}\{\mathbf{e}_i, \lambda_i\mathbf{e}_i, \dots, \lambda_i^{j-1}\mathbf{e}_i\} \\ &= \text{span}\{\mathbf{e}_i\}\end{aligned}$$

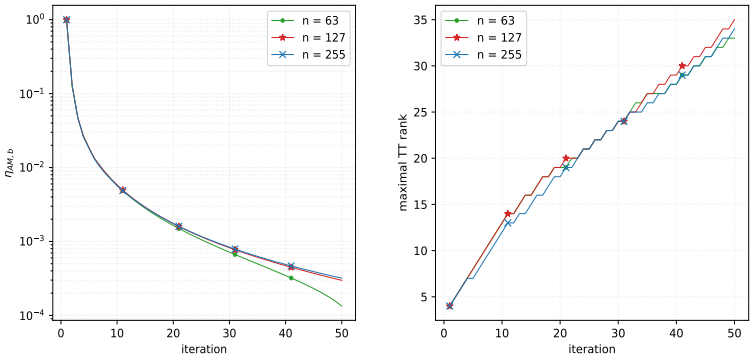
where λ_i is the i-th eigenvalue of $\mathbf{A}$. The Krylov space dimension is equal to 1, i.e., the number of eigenvector $\mathbf{e}_i$ necessary to express the right-hand side. Theoretically the number of iterations necessary to converge, i.e., the dimension of the Krylov space where the exact solution lives, is By linearity equal to the number of eigenvector necessary to express the right-hand side as their linear combination. In the problems presented in 4.3.1 and 4.3.2, we add a random generated tensor to the chosen right-hand side. Comparing the results a single right-hand side and multiple ones for the convection-diffusion problem, we may conclude that the introduced error has not increased the number of eigenvectors, for the tolerance chosen.

Let now consider a more peculiar problem with two right-hand side, living in subspaces generated by different eigenvector. More in details one right-hand side belongs to the subspace generated by a single eigenvector, while the other to the subspace generated by k different eigenvector. Thanks to our previous argument, theoretically the two systems converge independently with a different number of iterations, one for the first and k for the second. When we solve the two systems together, we expect the "all-in-one" system to converge as the slowest one, i.e., as the slowest converging one. Let $\mathbf{e_1}, \dots, \mathbf{e}_{k+1}$ be the first k different eigenvectors of the 3-dimensional discrete Laplacian $-\mathbf{\Delta}_3$. We consider the two following linear systems

$$-\mathbf{\Delta}_3\mathbf{x}_1 = \mathbf{e}_1 \tag{36}$$

$$-\mathbf{\Delta}_3\mathbf{x}_2 = \sum_{\ell=2}^{k+1}\mathbf{e}_\ell. \tag{37}$$

As described in Subesection 3, we define the "all-in-one" linear system with the 'diagonal' tensor operator $\mathbf{A}$ and the "all-in-one" right-hand side $\mathbf{b}$. TT-GMRES is used to solve this "all-in-one" system for $k = 10$, for a grid step dimension equal to $n \in \{63, 127, 255\}$, without preconditioner, with tolerance ε and rounding accuracy δ equal to 10^{-5}, no restart and a maximum of 50 iterations. Figure 18a shows the convergence history of the problem with the three different grid dimensions. Since there is no preconditioner the convergence is kind of slow, if compared with Figure 11a. At the same time, the TT-ranks growth not too quickly, because both there is not a random generated error and there are just two right-hand side. The residual curve associated

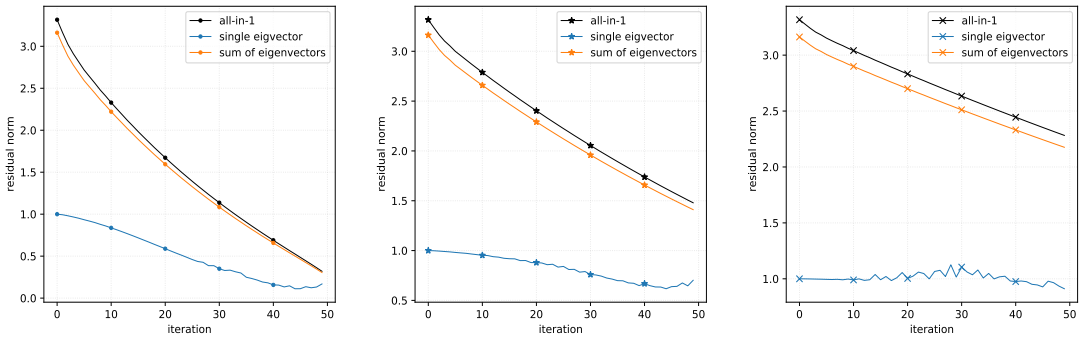


(a) Convergence history

(b) Maximal TT-rank of the last Krylov vector

Figure 18: 4-d multiple right-hand side problem with eigenvector

with Equation 36 in blue is almost flatten for all the grid dimensions, suggesting that GMRES almost immediately minimized it completely. On the other side the orange curve of problem (37) residual decreases slowly, meaning that minimizing its norm requires more steps. Lastly in all the plots the “all-in-one" residual curve follows exactly the eigenvector sum problem residual, confirming that the general convergence of the “all-in-one" system is decided by the slowest converging system, that is our intuition, confirmed in for all the grid dimensions in Figure 19. As last remark the “all-in-one" doesn't converge in 10 iterations because of the rounding effect, which slows down the convergence.



(a) Residual norm for $n = 63$

(b) Residual norm for $n = 127$

(c) Residual norm for $n = 255$

Figure 19: 4-d multiple right-hand side problem, residual comparison

C Further details on the “all-in-one" system

This appendix describes in details the construction in TT-format of the “all-in-one" system. As conclusion we provides a corollary to Proposition 3.4.

As previously stated, given a tensor $\mathbf{a} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ in TT-format with TT-cores $\underline{\mathbf{a}}_k \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$, we denote by $\mathbf{a}^{[k,i_k]}$ the i_k-th slice with respect to mode k, which in TT-format

writes as

$$\mathbf{a}^{[k,i_k]} = \underline{\mathbf{a}}_1 \cdots \underline{\mathbf{a}}_{k-1} \underline{A}_k(i_k) \underline{\mathbf{a}}_{k+1} \cdots \underline{\mathbf{a}}_d$$

with $\underline{A}_k(i_k) \in \mathbb{R}^{r_{k-1} \times r_k}$. Since henceforth we will take slice only with respect to the first mode, instead of writing $\mathbf{a}^{[1,i_1]}$ for the i_1-th slice on the first mode we will simply write $\mathbf{a}^{[i_1]}$. Similarly $\mathbf{A}^{[\ell]}$ denotes the (ℓ,ℓ)-th slice of $\mathbf{A} \in \mathbb{R}^{(n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$ with respect to the first two modes.

We start constructing the elements of the "all-in-one" system from the p individual right-hand sides. Let $\mathbf{b}_\ell \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a TT-vector for every $\ell \in \{1, \ldots, p\}$ with TT-cores $\underline{\mathbf{b}}_{\ell,k} \in \mathbb{R}^{s_{\ell,k} \times n_k \times s_{\ell,k+1}}$ for every $k \in \{1, \ldots, d\}$ with $s_1 = s_{d+1} = 1$, i.e.,

$$\mathbf{b}_\ell = \underline{\mathbf{b}}_{\ell,1} \cdots \underline{\mathbf{b}}_{\ell,d} \tag{38}$$

and its $(i_1, \ldots, i_d)$ element writes

$$\mathbf{b}_\ell(i_1, \ldots, i_d) = \underline{B}_{\ell,1}(i_1) \cdots \underline{B}_{\ell,d}(i_d)$$

with $\underline{B}_{\ell,k}(i_k) \in \mathbb{R}^{s_{\ell,k} \times s_{\ell,k+1}}$, $\underline{B}_{\ell,1}(i_1) \in \mathbb{R}^{1 \times s_{\ell,2}}$ and $\underline{B}_{\ell,d}(i_d) \in \mathbb{R}^{s_{\ell,d} \times 1}$. For simplicity we impose $s_k = s_{\ell,k}$ for every $\ell \in \{1, \ldots, p\}$ and $k \in \{1, \ldots, d\}$.

Remark .1. *This assumption on the TT-rank of $\mathbf{b}_\ell$ is not binding. Indeed setting $s_k = \max_{h \in \{1,\ldots,p\}} s_{h,k}$, then each core tensor $\underline{\mathbf{b}}_{h,k}$ of mode sizes $(s_{h,k}, n_h, s_{h,k+1})$ can be extended with zeros to (s_k, n_h, s_{k+1})*

We want to construct a tensor $\mathbf{b} \in \mathbb{R}^{p \times n_1 \times \cdots \times n_d}$ such that its ℓ-th slice with respect to the first mode of $\mathbf{b}$ is $\mathbf{b}_\ell$, i.e., $\mathbf{b}^{[\ell]} = \mathbf{b}_\ell$. As consequence, the k-th TT-core of $\mathbf{b}$ is $\underline{\mathbf{b}}_k \in \mathbb{R}^{ps_k \times n_k \times ps_{k+1}}$ such that

$$\underline{B}_k(i_k) = \begin{bmatrix} \underline{B}_{1,k}(i_k) & & \\ & \ddots & \\ & & \underline{B}_{p,k}(i_k) \end{bmatrix} \in \mathbb{R}^{ps_k \times ps_{k+1}}$$

for $k \in \{1, \ldots, d-1\}$, while $\underline{\mathbf{b}}_d \in \mathbb{R}^{ps_{d-1} \times n_d \times 1}$ and $\underline{\mathbf{b}}_0 \in \mathbb{R}^{1 \times p \times ps_1}$ are

$$\underline{B}_d(i_d) = \begin{bmatrix} \underline{B}_{1,d}(i_d) \\ \vdots \\ \underline{B}_{p,d}(i_d) \end{bmatrix} \in \mathbb{R}^{ps_d \times 1} \qquad \text{and} \qquad \underline{B}_0(\ell) = \begin{bmatrix} 0 \cdots 1 \cdots 0 \end{bmatrix} \in \mathbb{R}^{1 \times ps_1}$$

with the ℓ-th component of $\underline{B}_0(\ell)$ beign the only non-zero element. The TT-expression of $\mathbf{b}$ is

$$\mathbf{b} = \underline{\mathbf{b}}_0 \underline{\mathbf{b}}_1 \cdots \underline{\mathbf{b}}_d. \tag{39}$$

By construction we have that the ℓ-th slice of $\mathbf{b}$ with respect to mode 1 is

$$\begin{aligned} \mathbf{b}^{[\ell]} &= \underline{B}_0(\ell) \underline{\mathbf{b}}_1 \cdots \underline{\mathbf{b}}_d \\ &= \underline{\mathbf{b}}_{\ell,1} \cdots \underline{\mathbf{b}}_{\ell,d} \\ &= \mathbf{b}_\ell. \end{aligned}$$

We illustrate now the construction of the "all-in-one" system tensor linear operator. Let $\mathbf{C}, \mathbf{G} \in \mathbb{R}^{(n_1 \times n_1) \times \cdots \times (n_d \times n_d)}$ be two TT-matrix with k-th TT-core $\underline{\mathbf{c}}_k \in \mathbb{R}^{r_k \times n_k \times n_k \times r_{k+1}}$ and $\underline{\mathbf{g}}_k \in \mathbb{R}^{q_k \times n_k \times n_k \times q_{k+1}}$ for $k \in \{1, \ldots, d\}$ with $q_1 = r_1 = r_{d+1} = q_{d+1} = 1$, whose TT-expression is

$$\mathbf{C} = \underline{\mathbf{c}}_1 \cdots \underline{\mathbf{c}}_d \qquad \text{and} \qquad \mathbf{G} = \underline{\mathbf{g}}_1 \cdots \underline{\mathbf{g}}_d. \tag{40}$$

Given a diagonal matrix $D = \mathrm{diag}(\alpha_1, \ldots, \alpha_p)$, we define $\mathbf{A} \in \mathbb{R}^{(p\times p)\times(n_1\times n_1)\times\cdots\times(n_d\times n_d)}$ as

$$\mathbf{A} = \mathbb{I}_p \otimes \mathbf{C} + D \otimes \mathbf{G} \tag{41}$$

Then the expression of $\underline{\mathbf{a}}_k \in \mathbb{R}^{(r_k+q_k)\times n_k\times n_k\times(r_{k+1}+q_{k+1})}$ the k-th TT-core of $\mathbf{A}$ is

$$\underline{A}_k(i_k, j_k) = \begin{bmatrix} \underline{C}_k(i_k, j_k) & \mathbb{0} \\ \mathbb{0} & \underline{G}_k(i_k, j_k) \end{bmatrix} \qquad \text{and} \qquad \underline{A}_d(i_d, j_d) = \begin{bmatrix} \underline{C}_d(i_d, j_d) \\ \underline{G}_d(i_d, j_d) \end{bmatrix} \tag{42}$$

for every $i_k, j_k \in \{1, \ldots, n_k\}$ and $k \in \{1, \ldots, d\}$. The first TT-core $\underline{\mathbf{a}}_0 \in \mathbb{R}^{1\times p\times p\times 2}$ writes

$$\underline{A}_0(\ell, m) = \delta_{\ell,m} a_\ell \qquad \text{with} \qquad a_\ell = \begin{bmatrix} 1 & \alpha_\ell \end{bmatrix} \tag{43}$$

with $\delta_{\ell,m}$ the Kronecker delta, for $\ell, m \in \{1, \ldots, p\}$. The final TT-expression of $\mathbf{A}$ is

$$\mathbf{A} = \underline{\mathbf{a}}_0 \underline{\mathbf{a}}_1 \cdots \underline{\mathbf{a}}_d.$$

Remark now that $\mathbf{A}^{(\ell,m)}$ the (ℓ, m)-th slice with respect to mode 1 of $\mathbf{A}$ is

$$\mathbf{A}^{[\ell,m]} = \underline{A}_0(\ell, m)\underline{\mathbf{a}}_1 \cdots \underline{\mathbf{a}}_d = \delta_{\ell,m} a_\ell \underline{\mathbf{a}}_1 \cdots \underline{\mathbf{a}}_d.$$

If $\ell \neq m$, then $\mathbf{A}^{[\ell,m]} = \mathbb{0}$. On the other side, if ℓ and m are equal, then

$$\mathbf{A}^{[\ell,\ell]} = \mathbb{I}(\ell, \ell) a_\ell \, \underline{\mathbf{a}}_1 \cdots \underline{\mathbf{a}}_d = \underline{\mathbf{c}}_1 \cdots \underline{\mathbf{c}}_d + \alpha_\ell \, \underline{\mathbf{g}}_1 \cdots \underline{\mathbf{g}}_d = \mathbf{C} + \alpha_\ell \, \mathbf{G}.$$

Let consider $\mathbf{A}$ and $\mathbf{b}$ as defined in Equation (41) and (39), given $\mathbf{x} \in \mathbb{R}^{p\times n_1\times\cdots\times n_d}$ and define the new vector

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

We want to prove that $\mathbf{r}^{[\ell]}$ the ℓ-the slice with respect to the first mode of $\mathbf{r}$ is equal to the difference of the ℓ-th slices, i.e.,

$$\mathbf{r}^{[\ell]} = \mathbf{A}^{[\ell,\ell]}\mathbf{x}^{[\ell]} - \mathbf{b}_\ell = (\mathbf{C} + \alpha_\ell \mathbf{G})\mathbf{x}^{[\ell]} - \mathbf{b}_\ell \tag{44}$$

since the (ℓ, ℓ)-th slice of $\mathbf{A}$ is $\mathbf{C} + \alpha_\ell \mathbf{G}$ for every $\ell \in \{1, \ldots, p\}$. Remark that the ℓ-th slice of $\mathbf{b}$ is $\mathbf{b}_\ell$ by construction. As consequence, the Equation (44) is true if we show that the ℓ-th slice of the contraction between $\mathbf{A}$ and $\mathbf{x}$ is equal to the contraction of their ℓ-th slices, i.e.,

$$(\mathbf{A}\mathbf{x})^{[\ell]} = \mathbf{A}^{[\ell,\ell]}\mathbf{x}^{[\ell]}.$$

Lemma .2. *Given* $\mathbf{A}$, $\mathbf{C}$, $\mathbf{G}$ *as in Equations* (40) *and* (41)*, let* $\mathbf{x} \in \mathbb{R}^{p\times n_1\times\cdots\times n_d}$ *be a* $(d+1)$*-order tensor. Then the* ℓ*-th slice of* $\mathbf{A}\mathbf{x}$ *is equal to the product of their* ℓ*-th slices, i.e.*

$$(\mathbf{A}\mathbf{x})^{[\ell]} = \mathbf{A}^{[\ell,\ell]}\mathbf{x}^{[\ell]} = (\mathbf{C} + \alpha_\ell \mathbf{G})\mathbf{x}^{[\ell]}.$$

Defined and $\mathbf{w} = (\mathbf{C} + \alpha_\ell \mathbf{G})\mathbf{x}^{[\ell]}$*, then* $\mathbf{y}^{[\ell]}$ *the* ℓ*-th slice of* $\mathbf{y}$ *with respect to mode* 1 *is equal to* $\mathbf{w}$*, i.e.,*

$$\mathbf{y}^{[\ell]} = \mathbf{w}.$$

Proof. Let $\underline{\mathbf{x}}_k \in \mathbb{R}^{t_k\times n_k\times t_{k+1}}$ be the k-th TT-core of $\mathbf{x}$ for $k \in \{1, \ldots, d\}$ with $t_{d+1} = 1$ and $\mathbf{x}_0 \in \mathbb{R}^{1\times p\times t_1}$, getting

$$\mathbf{x} = \underline{\mathbf{x}}_0 \underline{\mathbf{x}}_1 \cdots \underline{\mathbf{x}}_d \tag{45}$$

Set $\mathbf{y} = \mathbf{A}\mathbf{x}$, then by the property of TT-contraction, we get

$$\begin{aligned}\mathbf{y}(\ell, i_1, \ldots, i_d) &= \sum_{j_0, j_1, \ldots, j_d = 1}^{p, n_1, \ldots, n_d} \underline{A}_0(\ell, j_0)\underline{A}_1(i_1, j_1)\cdots\underline{A}_d(i_d, j_d)\underline{X}_0(j_0)\underline{X}_1(j_1)\cdots\underline{X}_d(j_d) \\ &= \sum_{j_0, j_1, \ldots, j_d = 1}^{p, n_1, \ldots, n_d} \big(\underline{A}_0(\ell, j_0) \otimes_{\mathrm{K}} \underline{X}_0(j_0)\big)\big(\underline{A}_1(i_1, j_1) \otimes_{\mathrm{K}} \underline{X}_1(j_1)\big)\cdots\big(\underline{A}_d(i_d, j_d) \otimes_{\mathrm{K}} \underline{X}_d(j_d)\big) \\ &= \underline{Y}_0(\ell)\underline{Y}_1(i_1)\cdots\underline{Y}_d(i_d)\end{aligned}$$

where $\underline{Y}_k(i_k) \in \mathbb{R}^{r_k t_k \times r_{k+1} t_{k+1}}$ with $r_1 = 1$, $Y_d(i_d) \in \mathbb{R}^{r_d t_d \times 1}$ and $\underline{Y}_0(\ell) \in \mathbb{R}^{1 \times t_1}$ defined as

$$\begin{aligned}\underline{Y}_0(i_0) &= \sum_{j_0=1}^{p} \underline{A}_0(\ell, j_0) \otimes_{\mathrm{K}} \underline{X}_0(j_0) \\ \underline{Y}_k(i_k) &= \sum_{j_k=1}^{n_k} \underline{A}_k(i_k, j_k) \otimes_{\mathrm{K}} \underline{X}_k(j_k) \qquad \text{for} \qquad k \in \{2, \ldots, d\}.\end{aligned} \tag{46}$$

Remark now that in the expression $\underline{Y}_0(\ell)$, the quantity $\underline{A}_0(\ell, j_0)$ is actually a vector of 2 elements times δ_{ℓ, j_0}, so we replace the Kronecker product with a simple scalar-matrix product, writing

$$\begin{aligned}\underline{Y}_0(\ell) &= \sum_{j_0=1}^{p} \delta_{\ell, j_0} a_\ell \otimes_{\mathrm{K}} \underline{X}_0(j_0) \\ &= \underline{A}_0(\ell, \ell) \otimes_{\mathrm{K}} \underline{X}_0(\ell).\end{aligned} \tag{47}$$

Let $\mathbf{x}^{[\ell]}$ be the ℓ-th slice with respect to the first mode of $\mathbf{x}$, whose TT-expression is

$$\mathbf{x}^{[\ell]} = \underline{X}_0(\ell)\underline{\mathbf{x}}_1 \cdots \underline{\mathbf{x}}_d.$$

We define $\mathbf{x}^{[\ell]}$ TT-cores to get the clean expression as

$$\begin{aligned}\underline{\mathbf{x}}_{\ell,1} &= \underline{X}_0(\ell)\mathbf{x}_1 \ \in \mathbb{R}^{1 \times n_1 \times t_2} \\ \underline{\mathbf{x}}_{\ell,k} &= \underline{\mathbf{x}}_k \in \mathbb{R}^{t_k \times n_k \times t_{k+1}} \qquad \text{for} \qquad k \in \{2, \ldots, d\}\end{aligned}$$

getting

$$\mathbf{x}^{[\ell]} = \underline{\mathbf{x}}_{\ell,1} \cdots \underline{\mathbf{x}}_{\ell,d}.$$

To compute $\mathbf{w} = (\mathbf{C} + \alpha_\ell \mathbf{G})\mathbf{x}^{[\ell]}$, we need to clarify the structure of the k-th TT-core of $\mathbf{H} = (\mathbf{C} + \alpha_\ell \mathbf{G})$, given by the TT-sum rule. Therefore the k-th TT-core is $\underline{\mathbf{h}}_k \in \mathbb{R}^{(r_k + q_k) \times n_k \times n_k \times (r_{k+1} + q_{k+1})}$ such that

$$\underline{H}_k(i_k, j_k) = \begin{bmatrix} \underline{C}_k(i_k, j_k) & \mathbb{0} \\ \mathbb{0} & \underline{G}_k(i_k, j_k) \end{bmatrix} \qquad \text{and} \qquad \underline{H}_d(i_d, j_d) = \begin{bmatrix} \underline{C}_d(i_d, j_d) \\ \underline{G}_d(i_d, j_d) \end{bmatrix} \tag{48}$$

for $i_k, j_k \in \{1, \ldots, n_k\}$ and $k \in \{2, \ldots, d\}$ with $r_{d+1} + q_{d+1} = 1$. The first TT-core $\underline{\mathbf{h}}_1 \in \mathbb{R}^{1 \times n_1 \times n_1 \times (r_2 + q_2)}$ is

$$\underline{H}_1(i_1, j_1) = \begin{bmatrix} \underline{C}_1(i_1, j_1) \\ \alpha_\ell \underline{G}_1(i_1, j_i) \end{bmatrix}$$

for $i_1, j_1 \in \{1, \ldots, n_1\}$.

Compute the $(i_1, \ldots, i_d)$ element of $\mathbf{w} = \mathbf{H}\mathbf{x}^{[\ell]} = (\mathbf{C} + \alpha_\ell \mathbf{G})\mathbf{x}^{[\ell]}$ as

$$\begin{aligned}
\mathbf{w}(i_1, \ldots, i_d) &= \sum_{j_1,\ldots,j_d}^{n_1 \ldots n_d} \underline{H}_1(i_1, i_1) \cdots \underline{H}_d(i_d, j_d) \underline{X}_{\ell,1}(j_1) \cdots \underline{X}_{\ell,d}(j_d) \\
&= \sum_{j_1,\ldots,j_d}^{n_1,\ldots,n_d} \big(\underline{H}_1(i_1, j_1)\big) \otimes_{\mathrm{K}} \underline{X}_{\ell,1}(j_1)\big) \cdots \big(\underline{H}_d(i_d, j_d) \otimes_{\mathrm{K}} \underline{X}_{\ell,d}(j_d)\big) \\
&= \underline{W}_1(i_1) \underline{W}_2(i_2) \cdots \underline{W}_d(i_d)
\end{aligned}$$

where $\underline{W}_k(i_k) \in \mathbb{R}^{r_k t_k \times r_{k+1} t_{k+1}}$, $W_d \in \mathbb{R}^{r_d t_d \times 1}$ and $\underline{W}_1 \in \mathbb{R}^{1 \times r_2 t_2}$ are defined as

$$\underline{W}_1(i_1) = \sum_{j_1=1}^{n_1} \underline{H}_1(i_1, j_1) \otimes_{\mathrm{K}} \underline{X}_0(\ell) \underline{X}_1(j_1) \tag{49}$$

$$\underline{W}_k(i_k) = \sum_{j_k=1}^{n_k} \underline{W}_k(i_k, j_k) \otimes_{\mathrm{K}} \underline{X}_k(j_k) \qquad \text{for} \qquad k \in \{2, \ldots, d\}. \tag{50}$$

Comparing Equation (48) and (42), we get $\underline{\mathbf{h}}_k = \underline{\mathbf{a}}_k$ for $k \in \{2, \ldots, d\}$. As consequence $\underline{W}_k(i_k)$ writes as

$$\underline{W}_k(i_k) = \sum_{j_k=1}^{n_k} \underline{H}_k(i_k, j_k) \otimes_{\mathrm{K}} \underline{X}_k(j_k) = \sum_{j_k=1}^{n_k} \underline{A}_k(i_k, j_k) \otimes_{\mathrm{K}} \underline{X}_k(j_k) = \underline{Y}_k(i_k).$$

If we show that $\underline{Y}_0(\ell) Y_1(i_1)$ is equal to $\underline{W}_1(i_1)$, then the thesis holds true. Remark now that $\underline{H}_1(i_1, j_1)$ can be expressed equivalently as

$$\underline{H}_1(i_1, j_1) = \begin{bmatrix} 1 & \alpha_\ell \end{bmatrix} \begin{bmatrix} \underline{C}_1(i_1, j_1) & \mathbb{0} \\ \mathbb{0} & \underline{G}_1(i_1, j_1) \end{bmatrix} = \underline{A}_0(\ell, \ell) \underline{A}_1(i_1, j_1).$$

from Equation (42) and (43). Thanks to this last equation, we have that

$$\begin{aligned}
\Big(\underline{H}_1(i_1, j_1)\Big) \otimes_{\mathrm{K}} \Big(\underline{X}_0(\ell) \underline{X}_1(j_1)\Big) &= \Big(\underline{A}_0(\ell, \ell) \underline{A}_1(i_1, j_1)\Big) \otimes_{\mathrm{K}} \Big(\underline{X}_0(\ell) \underline{X}_1(j_1)\Big) \\
&= \Big(\underline{A}_0(\ell, \ell) \otimes_{\mathrm{K}} \underline{X}_0(\ell)\Big) \Big(\underline{A}_1(i_1, j_1)) \otimes_{\mathrm{K}} \underline{X}_1(j_1)\Big)
\end{aligned}$$

by the mixed-product property of the Kronecker product. Summing over index j_1 the previous equation leads to

$$\begin{aligned}
\underline{W}_1(i_1) &= \sum_{j_1=1}^{n_1} \Big(\underline{H}_1(i_1, j_1)\Big) \otimes_{\mathrm{K}} \Big(\underline{X}_0(\ell) \underline{X}_1(j_1)\Big) \\
&= \sum_{j_1=1}^{n_1} \Big(\underline{A}_0(\ell, \ell) \otimes_{\mathrm{K}} \underline{X}_0(\ell)\Big) \Big(\underline{A}_1(i_1, j_1)) \otimes_{\mathrm{K}} \underline{X}_1(j_1)\Big) \\
&= \underline{Y}_0(\ell) \underline{Y}_1(i_1),
\end{aligned}$$

from Equations (46) and (47) i.e., the thesis. □

By the result of Lemma .2, we get that the ℓ-th slice of $\mathbf{A}\mathbf{x}$ writes

$$(\mathbf{A}\mathbf{x})^{[\ell]} = \mathbf{C}\mathbf{x}^{[\ell]}.$$

Therefore the ℓ-th slice of $\mathbf{r}$ is

$$\mathbf{r}^{[\ell]} = \mathbf{C}\mathbf{x}^{[\ell]} - \mathbf{b}^{[\ell]}.$$

As conclusive result of this construction, we want to show that

$$||\mathbf{r}||^2 = \sum_{\ell=1}^{p} ||\mathbf{r}^{[\ell]}||^2.$$

Lemma .3. *Given $\mathbf{s} \in \mathbb{R}^{n_0 \times n_1 \times \cdots \times n_d}$ and its i_0-th slice with respect to the first mode $\mathbf{s}^{(i_0)}$ then*

$$||\mathbf{s}||^2 = \sum_{i_0=1}^{n_0} ||\mathbf{s}^{[i_0]}||^2.$$

Proof. Let $\mathbf{s} \in \mathbb{R}^{n_0 \times n_1 \times \cdots \times n_d}$ be a $(d+1)$-order tensor expressed in TT-format with TT-cores $\underline{\mathbf{s}}_k \in \mathbb{R}^{r_k \times n_k \times r_{k+1}}$ with $r_0 = r_{d+1} = 1$, such that

$$\mathbf{s} = \underline{\mathbf{s}}_0 \cdots \underline{\mathbf{s}}_d.$$

Let define the TT-expression of $\mathbf{s}^{(i_0)} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is

$$\mathbf{s}^{[i_0]} = \underline{S}_0(i_0)\underline{\mathbf{s}}_1 \cdots \underline{\mathbf{s}}_d.$$

To have a correct TT-representation of $\mathbf{s}^{[i_0]}$ we define its TT-cores $\underline{\mathbf{s}}_{i_0,k} \in \mathbb{R}^{r_{k-1} \times n_k \times r_k}$ as follows

$$\begin{aligned} \underline{\mathbf{s}}_{i_0,1} &= \underline{S}_0(i_0)\underline{\mathbf{s}}_1 \ \in \mathbb{R}^{1 \times n_1 \times r_2} \\ \underline{\mathbf{s}}_{i_0,k} &= \underline{\mathbf{s}}_k \ \in \mathbb{R}^{r_k \times n_k \times r_{k+1}} \qquad \text{for} \qquad k \in \{2, \dots, d\}. \end{aligned}$$

Compute now the norm of $\mathbf{s}$ as

$$\begin{aligned} ||\mathbf{s}||^2 &= \sum_{i_0,j_1,\dots,j_d=1}^{n_0,n_1\dots,n_d} \underline{S}_0(i_0)\underline{S}_1(j_1)\cdots\underline{S}_d(j_d)\underline{S}_0(i_0)\underline{S}_1(j_1)\cdots\underline{S}_d(j_d) \\ &= \sum_{i_0,j_1,\dots,j_d=1}^{n_0,n_1,\dots,n_d} \Big(\underline{S}_0(i_0) \otimes_{\mathrm{K}} \underline{S}_0(i_0)\Big)\Big(\underline{S}_1(j_1) \otimes_{\mathrm{K}} \underline{S}_1(j_1)\Big)\cdots\Big(\underline{S}_d(j_d) \otimes_{\mathrm{K}} \underline{S}_d(j_d)\Big). \end{aligned}$$

Applying the mixed-product property of the Kronecker product to the first two matrix product, this last equation writes as

$$\begin{aligned} ||\mathbf{s}||^2 &= \sum_{i_0,j_1,\dots,j_d=1}^{n_0,n_1,\dots,n_d} \Big(\underline{S}_0(i_0) \otimes_{\mathrm{K}} \underline{S}_0(i_0)\Big)\Big(\underline{S}_1(j_1) \otimes_{\mathrm{K}} \underline{S}_1(j_1)\Big)\cdots\Big(\underline{S}_d(j_d) \otimes_{\mathrm{K}} \underline{S}_d(j_d)\Big) \\ &= \sum_{i_0=1}^{n_0} \sum_{j_1,\dots,j_d=1}^{n_1,\dots,n_d} \Big(\underline{S}_0(i_0)\underline{S}_1(j_1) \otimes_{\mathrm{K}} \underline{S}_0(i_0)\underline{S}_1(j_1)\Big)\cdots\Big(\underline{S}_d(j_d) \otimes_{\mathrm{K}} \underline{S}_d(j_d)\Big) \\ &= \sum_{i_0=1}^{n_0} \sum_{j_1,\dots,j_d=1}^{n_1,\dots,n_d} \Big(\underline{S}_{i_0,1}(j_1) \otimes_{\mathrm{K}} \underline{S}_{i_0,1}(j_1)\Big)\cdots\Big(\underline{S}_{i_0,d}(j_d) \otimes_{\mathrm{K}} \underline{S}_{i_0,d}(j_d)\Big) \\ &= \sum_{i_0=1}^{n_0} ||\mathbf{s}^{[i_0]}||^2. \end{aligned}$$

i.e., the thesis. □

By the result of Lemma .3, we have

$$||\mathbf{r}||^2 = \sum_{\ell=1}^{p} ||\mathbf{r}^{[\ell]}||^2.$$

Once the "all-in-one" system has been completely described in its construction in TT-format, we present a further result related to Proposition 3.2.

Corollary .4. *Under the hypothesis of Corollary 3.5, if there exists a* $k^\dagger \in \mathbb{N}$ *such that* $\|\mathbf{x}^{[k]}\| \leq \|\mathbf{A}^{-1}\|\sqrt{p}$ *for every* $k \geq k^\dagger$*, then*

$$\eta_{\mathbf{A},\mathbf{v}}(\mathbf{x}^{[k]})\,\rho^\dagger \geq \eta_{\mathbf{A}_\ell,\mathbf{v}_\ell}(\mathbf{x}_\ell^{[k]}) \qquad \textit{where} \qquad \rho^\dagger = \frac{\sqrt{p}}{2-\nu}(1+\kappa_2(\mathbf{A})) \tag{51}$$

for every $\ell \in \{1,\ldots,p\}$ *and for every* $k \in \mathbb{N}$ *such that* $k \geq k^\ddagger$ *where* $k^\ddagger = \max\{k^{**}, k^\dagger\}$ *with* k^{**} *given in Corollary 3.5.*

RESEARCH CENTRE
BORDEAUX – SUD-OUEST

200 avenue de la Vieille Tour
33405 Talence Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399