An extended Krylov-like method for the solution of multi-linear systems

Hussam Al Daas, Damiano Lombardi

Received: Added at production | Revised: Added at production | Accepted: Added at production

DOI: xxx/xxxx

ARTICLE TYPE

An extended Krylov-like method for the solution of multi-linear systems

Hussam Al Daas[1] | Damiano Lombardi[2]

[1]Rutherford Appleton Laboratory, Harwell Campus, Didcot, UK
[2]Inria Paris, 2, rue Simone Iff, Paris, France

Correspondence
*Corresponding author name, This is sample corresponding address. Email: authorone@gmail.com

Summary

In the present work, numerical methods for the solution of multi-linear system are presented. Most large-scale multi-linear solvers rely on either the alternating least-squares or low rank Krylov methods. The approach we use to develop our methods lies somehow in between and can be considered as a generalisation of an alternated direction method. Given the multi-linear operator in the form of a sum of Kronecker product of matrices, we solve at each iteration a linear system for each summand. The approximate solution is then defined to be the best linear combination of these solutions, as well as the previous solution and the residual. Some convergence results are proved. Numerical experiments on two problems arising from parametric PDEs show the effectiveness of the proposed method.

KEYWORDS:
Multi-linear systems, tensor methods, parametric Partial Differential Equations

1 | INTRODUCTION

Tensor methods are currently under scrutiny to provide efficient representations of high-dimensional data. A general overview on the topic can be found in [1–4]. Beside their application in compression ([5–8]) and completion tasks ([9–11]), they can be used to obtain parsimonious approximation of solutions of high-dimensional problems ([12–18]). This kind of problems arise in many applications such as parameter dependent PDEs, uncertainty quantification, and molecular dynamics [19–21]. Discretizing these problems yields large-scale linear systems whose sizes scale exponentially with the dimension. Casting the problem as a multi-linear system, that is, representing the system entities (the matrix, the right-hand side, and the solution) in their tensor counterpart, does not change the size of the system. However, it might allow to leverage the properties of these tensors to avoid the cumbersome of the curse of dimensionality. For example, in many cases and under mild assumptions, it can be shown that the solution of the problem admits a low rank representation [22, 23]. Therefore, developing methods that exploit this property gained the attention of researchers in the last decade right after low rank tensor techniques and their arithmetic were introduced [1, 24]. Several works were presented in the literature to propose and investigate different kind of methods to solve multi-linear problems efficiently.

A popular class of methods to solve multi-linear systems are based on the Alternating Least Square (ALS) method [25–29]. This consists in solving with a fixed-point strategy the Euler-Lagrange equations obtained by minimizing the residual norm. The drawback of this method is that an estimation of the rank of the solution needs to be prescribed which might not be a simple task. The Alternating Minimal Energy Method, proposed in [20, 30] is a modified ALS method that overcomes the last issue. Methods based on the minimization of the residual energy have been recently investigated in [31]. All these methods proved to be efficient, in particular when the system to be solved is symmetric positive definite. Similarly to linear system solving, Krylov subspace methods, proposed for instance in [32–37], are an important class of solvers, well adapted for the solution of generic systems. There are also other attempts to extend several iterative linear solvers to the multi-linear context [21, 38–40]. In [41], an Alternating Direction

(AD) method with shifts is set up to efficiently solve the problem arising from the discretization of elliptic equations. The method we present in this work is closely related to the last method. However, our method does not require shifts.

The present work was motivated by the parsimonious approximation of parametric Partial Differential Equations (PDEs). In particular, we focus on the possibility of solving systems which are not necessarily symmetric and positive definite. The method we propose leverage some observations on the structure of multi-linear systems and what could be an Alternating Direction Iteration (ADI) strategy applied to this context. When generalized, this iteration can be interpreted as a first step towards an extended Krylov-like method[42] to solve multi-linear systems. One of the key properties is the possibility to take advantage of state-of-the-art solvers for linear systems and preconditioners.

The structure of the work reads as follows: Section 2 presents the notation we use and defines the framework of the multidimensional problem that we consider in this work. Afterwards, we present the multi-linear solvers based on the alternating direction method in Section 3 jointly with a convergence analysis in the coercive case. Following that, we discuss practical implementation issues and compare the computational operations required for the proposed solvers against the inexact Krylov method TTGMRES. We illustrate the efficiency of the proposed solver by presenting numerical experiments in Section 4 on two test cases that do not respect the theoretical assumptions required by our analysis. We demonstrate how the proposed solvers can achieve the same accuracy obtained by TTGMRES while keeping the intermediate tensor ranks very small compared to the ones encountered in TTGMRES. Finally, we give our conclusion in Section 5.

2 | NOTATION AND PROBLEM SETTING.

We introduce in this section the notation used throughout the manuscript. As stated in the introduction, this work is motivated by the application of tensor methods in high-dimensional function representation. In this section, we start by introducing the notation to deal with functions.

Let $d \in \mathbb{N}^*$ be the number of variables, $m_1, \dots m_d \in \mathbb{N}^*$ and $\Omega_1 \subset \mathbb{R}^{m_1}, \dots, \Omega_d \subset \mathbb{R}^{m_d}$ be open bounded sets. A function $u(x_1, \dots, x_d)$ is defined as follows:

$$u : \begin{cases} \Omega_1 \times \dots \times \Omega_d \to \mathbb{R} \\ (x_1, \dots, x_d) \mapsto u(x_1, \dots, x_d). \end{cases}$$

The function u is, in the present context, a solution of a problem (a system of parametric PDEs), that can be written in general as:

$$\mathcal{F}(u; x_1, \dots, x_d) = 0.$$

Some of the variables play the role of space-time variables and some of them might play the role of parameters. For numerous equations, the structure of the residual $\mathcal{F}$ is such that, when discretized, it leads to a multi-linear system. Let us consider, for instance, the following parametric problem: let the space variable be $x_1 \in \Omega_1$, the diffusivity be $x_2 \in \Omega_2$, the domain be $\Omega = \Omega_1 \times \Omega_2$ and the problem to be solved be:

$$-x_2 \Delta_{x_1} u = f \ \text{ in } \Omega_1 \times \Omega_2,$$
$$u = 0 \ \text{ on } \partial\Omega_1 \times \Omega_2.$$

Let us write the weak formulation for this problem. Let $v \in H_0^1(\Omega_1) \times L^2(\Omega_2)$. Then, it holds:

$$\int_\Omega x_2 \nabla_{x_1} u \cdot \nabla_{x_1} v \, dx_1 dx_2 = \int_\Omega f v \, dx_1 dx_2, \ \ \forall v \in H_0^1(\Omega_1) \times L^2(\Omega_2).$$

We discretize the weak formulation of the problem. As the functions are defined on domains which are in Cartesian product of domains this would make it possible to exploit the principle of separation of variables. Let us consider, first, a Galerkin method to discretize the first variable. Let $V_{N_1} \subset H_0^1(\Omega_1)$ be a finite dimensional subspace of $H_0^1(\Omega_1)$, dense in $H_0^1(\Omega_1)$. Then:

$$u(x_1, x_2) \approx \sum_{i_1=1}^{N_1} \hat{u}_{i_1}(x_2) v_{i_1}^{(1)}(x_1).$$

We consider, in this example, a Galerkin formulation also for the second variable. This choice is arbitrary and the discretization in different variables can be heterogeneous. Let $V_{N_2} \subset L^2(\Omega_2)$, every function $\hat{u}_{i_1}(x_2)$ can be approximated as follows:

$$\hat{u}_{i_1}(x_2) \approx \sum_{i_2=1}^{N_2} U_{i_1 i_2} v_{i_2}^{(2)}(x_2).$$

We then restrict the formulation to $V_{N_1} \times V_{N_2}$ by considering test functions $\left\{ v^{(1)}_{i_1} v^{(2)}_{i_2} \right\}_{1 \leq i_1 \leq N_1, 1 \leq i_2 \leq N_2}$:

$$\sum_{i_1=1}^{N_1} \sum_{i_2=1}^{N_2} U_{i_1 i_2} \int_{\Omega_1} \nabla_{x_1} v^{(1)}_{i_1} \nabla_{x_1} v^{(1)}_{j_1} \, dx_1 \int_{\Omega_2} x_2 v^{(2)}_{i_2} v^{(2)}_{j_2} \, dx_2 = \int_{\Omega} f v^{(1)}_{j_1} v^{(2)}_{j_2} \, dx_1 dx_2, \ \forall \left\{ v^{(1)}_{j_1}, v^{(2)}_{j_2} \right\}_{1 \leq i_1 \leq N_1, 1 \leq i_2 \leq N_2} \in V_{N_1} \times V_{N_2}.$$

In the case in which f is available in separated form, the integral on the right-hand side can also be separated. The discretized weak formulation is a matrix equation for the unknown $U \in \mathbb{R}^{N_1 \times N_2}$. In general, when considering more than 2 variables, the unknown will be a tensor, and the equation (when the residual is similar in the structure to the one considered in the example) will be a multi-linear problem.

We introduce now the notation used to deal with discrete tensors. Let $d \in \mathbb{N}^*$ be the tensor order. Let $(N_1, \dots, N_d) \in \mathbb{N}^d$ be the mode sizes and a multi-index be $(i_1, \dots, i_d) \in \mathbb{N}^d$, with $1 \leq i_j \leq N_j, \ 1 \leq j \leq d$. A mode is an element $v^{(j)} \in \mathbb{R}^{N_j}$. Let a tensor $T \in \mathbb{R}^{N_1 \times \dots \times N_d}$ be the element of a multi-linear space.

A pure tensor product is defined as follows:

$$v^{(1)} \otimes \cdots \otimes v^{(d)} : \begin{cases} \mathbb{R}^{N_1} \times \cdots \times \mathbb{R}^{N_d} \to \mathbb{R}^{N_1 \times \cdots \times N_d} \\ v^{(1)}, \dots, v^{(d)} \mapsto T_{i_1, \dots, i_d} = v^{(1)}_{i_1} \cdots v^{(d)}_{i_d}. \end{cases}$$

A low rank tensor T is in the Canonical Polyadic (CP) format if it can be written as the sum of $r \in \mathbb{N}^*$ (called the CP rank) pure tensors:

$$T = \sum_{i=1}^{r} v^{(1)}_i \otimes \cdots \otimes v^{(d)}_i.$$

In what follows, we will make use of the Tensor Train (TT) format[24]:

$$T_{i_1, \dots, i_d} = g_{T,1}(i_1) G_{T,2}(i_2) \cdots G_{T,d-2}(i_{d-2}) g_{T,d}(i_d),$$

where $g_{T,1}(i_1) \in \mathbb{R}^{1 \times r_1}$, $g_{T,d}(i_d) \in \mathbb{R}^{r_{d-1} \times 1}$, $G_{T,k}(i_k) \in \mathbb{R}^{r_{k-1} \times r_k}$, and $r_k \in \mathbb{N}^*$ are the TT ranks. The Frobenius scalar product between two tensors X, Y is defined as:

$$\langle X, Y \rangle_F = \sum_{i_1=1}^{N_1} \cdots \sum_{i_d=1}^{N_d} X_{i_1, \dots, i_d} Y_{i_1, \dots, i_d}.$$

The Frobenius norm for a tensor will be denoted $\|X\|_F$ defined as:

$$\|X\|_F^2 = \sum_{i_1=1}^{N_1} \cdots \sum_{i_d=1}^{N_d} X^2_{i_1, \dots, i_d}.$$

A pure tensor operator is defined as follows. Let $A_j \in \mathbb{R}^{N_j \times N_j}$ be a set of matrices. A pure tensor operator $A = A_1 \otimes \cdots \otimes A_d$ is an operator whose action on a pure tensor product $v = v^{(1)} \otimes \cdots \otimes v^{(d)}$ reads:

$$Av : \begin{cases} \mathbb{R}^{N_1} \times \cdots \times \mathbb{R}^{N_d} \to \mathbb{R}^{N_1} \times \cdots \times \mathbb{R}^{N_d} \\ \left[A_1 \otimes \cdots \otimes A_d \right] v^{(1)} \otimes \cdots \otimes v^{d} \mapsto A_1 v^{(1)} \otimes \cdots \otimes A_d v^{(d)}. \end{cases} \tag{1}$$

In this work we consider operators which are written as a sum of $L \in \mathbb{N}^*$ pure operator tensor terms:

$$A = \sum_{l=1}^{L} A_{1,l} \otimes \cdots \otimes A_{d,l}. \tag{2}$$

Let B, X be two tensors, we will investigate a set of methods to solve the multi-linear problem:

$$AX = B. \tag{3}$$

3 | METHODS.

The methods investigated in the present work will try to leverage ideas introduced for the iterative solution of linear systems to the context of multi-linear algebra. One of the key observations which is at the origin of the first class of iterative methods, namely Jacobi and Gauss–Seidel, is that diagonal or triangular linear systems can be easily solved. This is the starting point of the present work: what are the structure of multi-linear problems that can be solved easily?

One easy to solve multi-linear problem is the following: let A be an invertible pure tensor operator (we will say a rank-1 operator), as defined in Eq.(1), and let $B = \sum_{i=1}^{r_B} b_i^{(1)} \otimes \cdots \otimes b_i^{(d)}$ be a rank r_B tensor. Then, the solution to the multi-linear system of equation $AX = B$ can be given as:

$$X = \sum_{i=1}^{r_B} A_1^{-1} b_i^{(1)} \otimes \cdots \otimes A_d^{-1} b_i^{(d)}.$$

Remark that the solution has been written by considering B, which is given, as written in CP format. Nothing changes at this level if B is written in Tucker (simple or hierarchical) or Tensor Train.

Solving the problem is equivalent to solve d linear systems with multiple right-hand sides.

3.1 | Alternated Direction Iteration (ADI).

The main steps of the method are detailed. Let $A_i = A_{1,i} \otimes \cdots \otimes A_{d,i}$ so that $A = \sum_{i=1}^{L} A_i$, and let $X^{(0)}$ be an initial (guess) solution. The approximate solution obtained at the $k-$th step of the method is denoted by $X^{(k)}$.

1. Solve L pure tensor operator systems for the tensors Y_i:

$$A_i Y_i + \sum_{l \neq i}^{L} A_l X^{(k-1)} = B, \quad 1 \leq i \leq L.$$

2. Let $(a_0, \dots a_L) \in \mathbb{R}$ be scalars. The solution $X^{(k)}$ is expressed as:

$$X^{(k)} = a_0 X^{(k-1)} + \sum_{i=1}^{L} a_i Y_i.$$

The weights are the solution of the following least square problem on the system residual:

$$(a_0^*, \dots, a_L^*) = \arg \inf_{(a_0, \dots, a_L)} \|B - AX^{(k)}\|_F^2.$$

3.1.1 | Analysis

We propose a mathematical analysis of the ADI iteration under the following hypotheses:

(H_1) : $\forall i = 1, \dots, L$, there exist real positive constants $m_i \in \mathbb{R}^+$ such that:

$$\langle U, A_i V \rangle_F \leq m_i \|U\|_F \|V\|_F, \quad \forall U, V \in \mathcal{V}_n$$

(H_2) : $\forall i = 1, \dots, L$, there exist real positive constants $\alpha_i \in \mathbb{R}^+$ such that:

$$\langle V, A_i V \rangle_F \geq \alpha_i \|V\|_F^2, \quad \forall (V \neq 0) \in \mathcal{V}_n.$$

Let the system residual at the k-th iteration be $R^{(k)}$. The following holds:

1. Proposition. If the method converges, it converges to the true solution of the multi-linear problem.

Proof. In the case of convergence, we have: $X^{(k)} = X^{(k+1)} = X$. This entails:

$$Y_i = A_i^{-1} B - A_i^{-1} \sum_{l \neq i}^{L} A_l X.$$

This can be rewritten as:

$$Y_i = A_i^{-1}(B - AX) + X.$$

The linear combination of the partial solutions reads:

$$X = \left(a_0 + \sum_{i=1}^{L} a_i \right) X + \sum_{i=1}^{L} a_i A_i^{-1} (B - AX).$$

Let us choose $(a_i)_{1 \leq i \leq L} > 0$ and $1 - a_0 - \sum_{i=1}^{L} a_i = 0$. Since $\sum_{i=1}^{L} a_i A_i^{-1}$ is invertible (hypotheses H_1, H_2), this entails $B - AX = 0$, which concludes the proof. □

The following result shows that the iteration produces a sequence of residuals whose norm is monotonically non-increasing.

2. For the ADI iteration defined above, it holds:

$$\|R^{(k+1)}\|_F \leq \|R^{(k)}\|_F, \ \forall k \geq 0.$$

Proof. The residual at iteration $k+1$ reads:

$$R^{(k+1)} = R^{(k)} + (1 - a_0) A X^{(k)} - \sum_{i=1}^{L} a_i A Y_i. \tag{4}$$

At every iteration, the coefficients $a_0, a_1, \dots, a_L$ are determined in such a way that the residual is minimized:

$$(a_0, a_1, \dots, a_L)^* = \arg \inf_{a_0, a_1, \dots, a_L} \frac{1}{2} \langle R^{(k+1)}, R^{(k+1)} \rangle_F.$$

The Euler Lagrange equations read:

$$\langle R^{(k+1)}, \partial_{a_i} R^{(k+1)} \rangle_F = 0, \ i = 0, \dots, L.$$

By computing the derivatives of the residual in Eq.(4) with respect to the coefficients we get:

$$\langle R^{(k+1)}, A X^{(k)} \rangle_F = 0,$$
$$\langle R^{(k+1)}, A Y_j \rangle_F = 0, \ \forall 1 \leq j \leq L.$$

By taking the scalar product of Eq.(4) with R^{k+1} it holds:

$$\|R^{(k+1)}\|_F^2 = \langle R^{(k)}, R^{(k+1)} \rangle_F,$$

because the two last terms vanish, and this, by the Cauchy-Schwarz inequality leads to:

$$\|R^{(k+1)}\|_F^2 \leq \|R^{(k+1)}\|_F \|R^{(k)}\|_F,$$

which concludes the proof. □

Under the hypothesis that the rank-1 multi-linear operators are associated to continuous and coercive forms, it is possible to state the following convergence result:

3. Let us recall the hypotheses $(H_1), (H_2)$ and show that, under these hypotheses the method actually converges. If the operators $(A_i)_{1 \leq i \leq L}$ are such that there exist constants $m_i, \alpha_i > 0$ such that:

$$\langle U, A_i V \rangle_F \leq m_i \|U\|_F \|V\|_F,$$
$$\langle U, A_i U \rangle_F \geq \alpha_i \|U\|_F^2, \forall U \in \mathcal{U}_n$$

Then, there exist a constant $C < 1$ such that: $\|R^{(k)}\|_F \leq C^k \|R^{(0)}\|_F$.

Proof. Let us start from the end of the previous proposition. It holds:

$$\|R^{(k+1)}\|_F^2 = \langle R^{(k)}, R^{(k+1)} \rangle_F.$$

For the Cauchy-Schwarz equality to hold, it must happen that there exist a real constant $\lambda \in \mathbb{R}$ such that: $R^{(k+1)} = \lambda R^{(k)}$. Let us show that this would lead to a contradiction. Indeed, consider the optimality conditions:

$$\langle R^{(k+1)}, A X^{(k)} \rangle_F = 0 = \lambda \langle R^{(k)}, A X^{(k)} \rangle_F,$$
$$\langle R^{(k+1)}, A Y_j \rangle_F = 0 = \lambda \langle R^{(k)}, A Y_j \rangle_F,$$

For the solutions Y_j it holds:

$$Y_j = X^{(k)} + A_j^{-1} R^{(k)},$$

which is substituted into the constraints leading to:

$$\langle R^{(k)}, A Y_j \rangle_F = 0 = \langle R^{(k)}, A X^{(k)} \rangle_F + \langle R^{(k)}, A A_j^{-1} R^{(k)} \rangle_F.$$

The first term vanishes because of the first set of constraints, and the second term leads to a set of equations of the form:

$$\langle R^{(k)}, A A_j^{-1} R^{(k)} \rangle_F = 0.$$

which is in clear contradiction with the hypotheses, hence proving that: $\|R^{(k+1)}\|_F < \|R^{(k)}\|_F$. Let us now quantify the rate of convergence. To this end, let us test the Eq.(4) against $R^{(k)}$:

$$\langle R^{(k+1)}, R^{(k)}\rangle_F = \|R^{(k)}\|_F^2 + (1-a_0)\langle AX^{(k)}, R^{(k)}\rangle_F - \sum_{i=1}^{L} a_i \langle AY_i, R^{(k)}\rangle_F.$$

This is rewritten as:

$$\langle R^{(k+1)}, R^{(k)}\rangle_F = \|R^{(k)}\|_F^2 + (1-a_0 - \sum_{i=1}^{L} a_i)\langle AX^{(k)}, R^{(k)}\rangle_F - \sum_{i=1}^{L} a_i \langle AA_i^{-1}R^{(k)}, R^{(k)}\rangle_F.$$

By virtue of the optimality conditions, the left hand side is the norm of the residual at $k+1$, squared. Moreover: $AX^{(k)} = B - R^{(k)}$:

$$\|R^{(k+1)}\|_F^2 = (a_0 + \sum_{i=1}^{L} a_i)\|R^{(k)}\|_F^2 + (1-a_0 - \sum_{i=1}^{L} a_i)\langle B, R^{(k)}\rangle_F - \sum_{i=1}^{L} a_i \langle AA_i^{-1}R^{(k)}, R^{(k)}\rangle_F.$$

The coercivity and continuity of the operators lead to:

$$\langle AA_i^{-1}R^{(k)}, R^{(k)}\rangle_F \geq \frac{1}{m_i}(\sum_{j=1}^{L} \alpha_j)\|R^{(k)}\|_F^2$$

We get the following inequality:

$$\|R^{(k+1)}\|_F^2 \leq \left[a_0 + \sum_{i=1}^{L} a_i \left(1 - \frac{1}{m_i}\sum_{j=1}^{L}\alpha_j\right)\right]\|R^{(k)}\|_F^2 + (1-a_0-\sum_{i=1}^{L} a_i)\langle B, R^{(k)}\rangle_F$$

The coefficients $a_0, \dots a_L$ have been optimized. They will provide a smaller or equal residual norm than the choice $(a_i)_{0\leq i\leq L} > 0$, $(1 - a_0 - \sum_{i=1}^{L} a_i) = 0$:

$$\|R^{(k+1)}\|_F^2 \leq \left[1 - \sum_{i=1}^{L} a_i + \sum_{i=1}^{L} a_i \left(1 - \frac{1}{m_i}\sum_{j=1}^{L}\alpha_j\right)\right]\|R^{(k)}\|_F^2 = \left[1 - \sum_{i=1}^{L}\frac{a_i}{m_i}\sum_{j=1}^{L}\alpha_j\right]\|R^{(k)}\|_F^2.$$

Let $C = \left[1 - \sum_{i=1}^{L}\frac{a_i}{m_i}\sum_{j=1}^{L}\alpha_j\right]^{1/2} < 1$, this concludes the proof.

□

3.2 | Extension with residual combination and analogy with the extended Krylov methods.

The basic iteration presented and analyzed in the previous section can be extended by observing that the iterate $X^{(k+1)}$ has been computed by considering a linear combination of tensors: the previous iterate and pre-conditioned residual. A natural extension consists in adding to the linear combination the residual itself and the operators applied to it:

1. Solve L pure tensor operator systems for the tensors Y_i:

$$A_i Y_i + \sum_{l\neq i}^{L} A_l X^{(k)} = B, \quad 1 \leq i \leq L.$$

2. Let $(a_0, \dots, a_{2L+1}) \in \mathbb{R}$ be scalars. The solution $X^{(k+1)}$ is expressed as:

$$X^{(k+1)} = a_0 X^{(k)} + \sum_{i=1}^{L} a_i Y_i + \sum_{j=L+1}^{2L} a_j A_j R^{(k)} + a_{2L+1} R^{(k)}.$$

The weights are the solution of the following least square problem on the system residual:

$$(a_0^*, \dots a_{2L+1}^*) = \arg \inf_{(a_0, \dots a_{2L+1})} \|B - AX^{(k+1)}\|_F^2.$$

Remark that, since $Y_i = X^{(k)} + A_i^{-1}R^{(k)}$ we have:

$$X^{(k+1)} = \left[a_0 + \sum_{j=L+1}^{2L} a_i\right] X^{(k)} + \sum_{i=1}^{L} a_i A_j^{-1} R^{(k)} + \sum_{j=L+1}^{2L} a_j A_i R^{(k)} + a_{2L+1} R^{(k)}.$$

If we just considered the residual we would get the equivalent of a Richardson iteration, multiplying the residual by the operators is a step towards building Krylov spaces, and the pre-conditioned residuals come from the ADI basic iteration. In terms of analysis, the results presented in the previous section hold, since the residual produced by the extended iteration will be smaller or equal than the one produced by the basic ADI iteration, by construction.

3.3 | Practical implementation and acceleration.

The computational cost of the proposed iteration is discussed. Let the solution $X^{(k)}$ and the residual $R^{(k)}$ be tensors written in CP format, of rank $n_s, n_r \in \mathbb{N}^*$ respectively. When solving the set of problems for Y_i, the rank of each of them is $n_r + n_s$. Then, the solution at the iteration X^{k+1} would have a rank equal to: $n_s+L(n_r+n_s)$ for the basic ADI iteration and $n_s+n_r+L(n_r+n_s)+Ln_r$ for the extended one. When iterating this would lead to a computational burden, both in terms of memory and number of operations. In order to reduce the cost, we propose to recompress the residual and the solution at each iteration. The truncation is performed adaptively in the following way. Let $\alpha_s^{(k)}, \alpha_r^{(k)} \in \mathbb{R}^+$. The truncation of $X^{(k)}, R^{(k)}$ is denoted by $\tilde{X}^{(k)}, \tilde{R}^{(k)}$:

$$\|X^{(k)} - \tilde{X}^{(k)}\|_F \leq \alpha_s^{(k)} \|X^{(k)}\|_F,$$
$$\|R^{(k)} - \tilde{R}^{(k)}\|_F \leq \alpha_r^{(k)} \|R^{(k)}\|_F.$$

The values of $\alpha_{s,r}^{(k)}$ are changing with the iteration. In particular, as the residual decreases, the accuracy of the solution representation is increased. From a practical point of view we consider the values as function of the norm of the residual itself. Let $\alpha_{s,r} \in \mathbb{R}^+$:

$$\alpha_{s,r}^{(k)} = \alpha_{s,r} \|R^{(k)}\|_F$$

This condition, per se, does not prevent eventual stagnation. To avoid this, if two successive iterates have a residual that decrease less than a prescribed tolerance, we divide the values of α_s, α_r by a factor 2:

$$\|R^{(k)}\|_F \geq (1-\delta)\|R^{(k-1)}\|_F \ \Rightarrow \ \alpha_{s,r} \leftarrow \frac{1}{2}\alpha_{s,r}.$$

In the present work we considered $\delta = 10^{-3}$.

3.4 | Summary of the method.

In this section we present the pseudocode of the algorithm and compare the computation costs of one iteration of TTGMRES and the proposed algorithms. An iteration of the MADI solver consists of preparing the right-hand sides R_k for $k = 1, \dots, r_A$ which can be computed by applying the operator $\left(A - A_{1,k} \otimes \cdots \otimes A_{d,k}\right)$ to the last approximate solution X,

$$R_k = \left(A - A_{1,k} \otimes \cdots \otimes A_{d,k}\right) X.$$

Afterwards, for $k = 1, \dots, r_A$, we solve the linear system of equations with the pure tensor operator $A_{1,k} \otimes \cdots \otimes A_{d,k}$,

$$A_{1,k} \otimes \cdots \otimes A_{d,k} S_k = R_k.$$

Note that since X is assumed to be represented in low rank form, R_k can be also give in a low rank form. Therefore, the last linear system can be solved efficiently using classic linear solver by solving for each mode individually. The extended version of the proposed algorithm computes $S_{r_A+k} = A_{1,k} \otimes \cdots \otimes A_{d,k} R_k$. Finally to obtain the new approximate solution, the method solves the least squares problem:

$$X = arg \min_{X=\sum_{k=0}^{p} \alpha_k S_k} \|AX - B\|_F,$$

where S_0 is the previous approximate solution and $p = 2r_A$ for the extended algorithm and $p = r_A$ otherwise. This problem can be solved by solving the $p \times p$ linear system

$$H\alpha = f,$$

where $H(i,j) = (AS_i)^T AS_j$ and $f(i) = (AS_i)^T B$. Note that the low rank structure of AS_i for $i = 1, \dots, p$ and B allows to compute H and f efficiently. Algorithm 1 presents a pseudocode of the extended multidimensional ADI method. A comparison of the computational costs of the operations performed in one iteration of our proposed methods and TTGMRES are presented in Table 1. Note that the costs of the matrix multiplication, preconditioning, and inner product depend on the ranks of the intermediate tensors arising in each method. It is known that the ranks of the Krylov basis vectors increase rapidly in low-rank GMRES methods. The intermediate tensor arising in our proposed methods are only the approximations of the solution and

Method	$\langle u, v\rangle_F$	$v = (A_1 \otimes \cdots \otimes A_k)u$	$v = (A_1^{-1} \otimes \cdots \otimes A_k^{-1})u$
MADI	$(\frac{(r_A+1)(r_A+2)}{2} + r_A + 1)$	r_A	r_A
MEADI	$((r_A + 1)(2r_A + 1) + 2r_A + 1)$	$2r_A$	$2r_A$
TTGMRES	$k + 1$	r_A	1

TABLE 1 Computational operations comparison. $\langle u, v\rangle_F$ represents the inner product of two tensors. $v = (A_1 \otimes \cdots \otimes A_k)u$ represents the product of a rank-1 pure operator with a tensor. k is the iteration number. $v = (A_1^{-1} \otimes \cdots \otimes A_k^{-1})u$ represents the solution of a linear system where the matrix is given as a nonsingular rank-1 pure operator. The ranks can be different in each method which affects the costs.

the residuals. Therefore, we expect that the ranks of these tensors would be moderate and of the order of the rank of the actual solution. Our numerical results align with our expectations. Using tensors in TT format, the costs of matrix-tensor product and preconditioning increase quadratically with the TT-rank while the inner product cost increases cubically. Hence, the cost of a TTGMRES iteration is expected to be significantly larger than the one of an multi dimensional ADI iteration.

Algorithm 1 Extended Multi dimensional ADI

```
Require: The tensor operator $A$ given as in 2, the tensor right-hand side $B$, convergence threshold $\varepsilon$, and an approximate solution $X_0$ in the TT format
Ensure: Approximate solution $X$ to 3

for $k = 1$ to $r_A$ do
    Set $R_k = B - \sum_{i=1, i\neq k}^{r_A} A_{1,i} \otimes \cdots \otimes A_{d,i} X_0$
    Solve $A_{1,k} \otimes \cdots \otimes A_{d,k} S_k = R_k$
    Set $S_{r_A+k} = A_{1,k} \otimes \cdots \otimes A_{d,k} R_k$
    Set $S_0 = X_0$
end for
Solve the least squares problem $\alpha = arg\min \|A \sum_{i=0}^{2r_A} \alpha_i S_i - B\|_F$
Set $X = \sum_{i=0}^{2r_A} \alpha_i S_i$
```

4 | NUMERICAL EXPERIMENTS.

In this section several numerical experiments are proposed to assess the proposed method. Our numerical experiments are performed using Matlab. The TT operations are performed by using the Matlab code from https://github.com/SAMSI-RandTensors/randomizedTT. In particular the rounding of the sum of TT tensors which is heavily required in both TTGMRES and our methods use the randomized algorithm proposed in[8].

4.1 | The cookies problem.

The first problem we consider is the so called cookies problem[19, 43] described as follows:

$$\begin{aligned} -\nabla \cdot (\sigma(x, y; \rho)\nabla(u(x, y; \rho))) &= f(x, y) \quad \text{in } \Omega, \\ u(x, y; \rho) &= 0 \qquad\quad \text{on } \delta\Omega, \end{aligned} \tag{5}$$

where Ω is $(-1, 1) \times (-1, 1)$, $\delta\Omega$ is the boundary of Ω and σ is defined as:

$$\sigma(x, y; \rho) = \begin{cases} 1 + \rho_i & \text{if } (x, y) \in D_i \\ 1 & \text{elsewhere,} \end{cases}$$

where D_i for $i = 1, \dots, p$ are disjoint disks distributed in Ω such that their centers are equidistant and ρ_i is selected from a set of samples $J_i \subset \mathbb{R}$ for $i = 1, \dots, p$. To solve this problem, for each combination of values $(\rho_1, \dots, \rho_p)$, one can solve the linear system $\left(A_{1,1} + \sum_{i=1}^{p} \rho_i A_{1,i+1}\right) u = f$, where $A_{1,1} \in \mathbb{R}^{I_1 \times I_1}$ is the discretization of the operator $-\nabla \cdot (\nabla \circ)$ in Ω, $A_{1,i+1}$ is the discretization of $-\nabla \cdot (\chi_{D_i} \nabla \circ)$ in Ω, where χ_S is the indicator function of the set S, and f is the discretization of the function f. The number of linear systems to solve in that case is the product of the cardinalities of the sets $(J_i)_{1 \le i \le p}$.

The first step in order to discretize the problem is to consider its weak formulation. Let $v \in H_0^1(\Omega)$, it holds:

$$\int\limits_{\Omega} \sigma \nabla u \cdot \nabla v \, dx = \int\limits_{\Omega} f v \, dx, \ \ \forall v \in H_0^1(\Omega).$$

We replace σ by its expression, obtaining:

$$\int\limits_{\Omega} \nabla u \cdot \nabla v \, dx + \sum_{i=1}^{p} \rho_i \int\limits_{\Omega} \chi_i \nabla u \cdot \nabla v \, dx = \int\limits_{\Omega} f v,$$

We are interested in approximating the solution u as function of the space x and the parameters $\left\{\rho_i\right\}_{1 \le i \le p}$, so that: $u = u(x, \rho_1, \dots, \rho_p)$. We choose to discretize this problem in space by means of finite elements, and for the parameters we adopt a collocation approach. In order to setup a Galerkin method, we consider $\left\{v_j\right\}_{1 \le j \le N_1} \in V_{N_1} \subset H_0^1(\Omega)$. We write:

$$u(x, \rho_1, \dots, \rho_p) = \sum_{k_1=1}^{n} \hat{u}_{k_1}(\rho_1, \dots \rho_p) v_{k_1}(x).$$

By restricting the test space to V_{N_1}, the discrete in space formulation reads:

$$\sum_{k_1=1}^{n} \hat{u}_{k_1} \left[\int\limits_{\Omega} \nabla v_{k_1} \cdot \nabla v_l \, dx + \sum_{i=1}^{p} \rho_i \int\limits_{\Omega} \chi_i \nabla v_{k_1} \cdot \nabla v_l \, dx \right] = \int\limits_{\Omega} f v_l, \ \ \forall v_l \in V_{N_1}.$$

Every $\hat{u}_{k_1}$ depends upon the p variables ρ_i. Once we discretize completely the problem, we obtain:

$$\hat{u}_{k_1 k_2 \dots k_{p+1}} = \hat{u}_{k_1}(\rho_1^{(k_2)}, \dots, \rho_p^{k_{(p+1)}}), \quad 1 \le k_i \le N_i, \text{ for } i = 1, \dots, p+1.$$

The fully discrete formulation hence reduces to:

$$\sum_{k_1=1}^{N_1} \hat{u}_{k_1 k_2 \dots k_{p+1}} \left[\int\limits_{\Omega} \nabla v_{k_1} \cdot \nabla v_l \, dx + \sum_{i=1}^{p} \rho_i^{k_{i+1}} \int\limits_{\Omega} \chi_i \nabla v_{k_1} \cdot \nabla v_l \, dx \right] = \int\limits_{\Omega} f v_l, \ \ \forall v_l \in V_{N_1}.$$

This is a multi-linear problem to be solved for the tensor $\hat{u}_{k_1 \dots k_{p+1}}$. Let:

$$\begin{aligned}
K_1 &\in \mathbb{R}^{N_1 \times N_1}, \ \ K_1(k,l) = \int\limits_{\Omega} \nabla v_k \cdot \nabla v_l \, dx, \text{ for } 1 \le k, l \le N_1 \\
K_i &\in \mathbb{R}^{N_1 \times N_1}, \ \ K_i(k,l) = \int\limits_{\Omega} \chi_{i-1} \nabla v_k \cdot \nabla v_l \, dx, \text{ for } i = 2, \dots, p+1, 1 \le k, l \le N_1 \\
R_i &\in \mathbb{R}^{N_i \times N_i}, \ \ R_i(k_i, l_i) = \rho_{i-1}^{(k_i)} \delta_{k_i, l_i}, \text{ for } i = 2, \dots, p+1, 1 \le k_i, l_i \le N_i
\end{aligned}$$

For each value $\rho = (\rho_1, \dots, \rho_p)$, we need to solve the linear system

$$\left(K_1 + \rho_1 K_2 + \dots + \rho_p K_{p+1}\right) u_{x,\rho} = f.$$

The set of linear systems can be cast as one linear system involving all possible values of ρ

$$\left(\sum_{i=1}^{p+1} A_{1,i} \otimes \dots \otimes A_{p+1,i}\right) u = f \otimes 1_{N_2} \otimes \dots \otimes 1_{N_{p+1}},$$

where $A_{1,i} = K_i$, $A_{i,j} = I_{N_j}$ if $i \neq j$, $A_{i,i} = R_i$ for $i = 2, \dots, p+1$, and 1_{N_i} is the vector of ones of length N_i.

In this numerical test, we discretize the problem using $\mathbb{P}1$ finite elements in the 2D domain $(-1, 1) \times (-1, 1)$. The number of regions $p = 4$. The number of degrees of freedom in space $N_1 = 2{,}855$ and the number of samples for each parameter is $N_i = 100$ for $i = 2, \dots, 5$ taken from an equidistant sampling of $[1, 10]$.

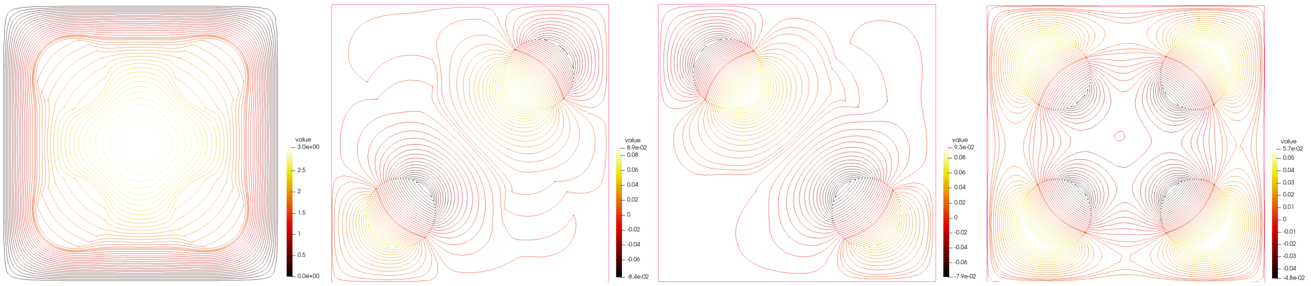

FIGURE 1 First four fibers of the TT-core associated to the space variable, reconstructed by using the finite element space, for the test-case presented in Section4.1. Contour plots with 40 values between the minimum and the maximum.

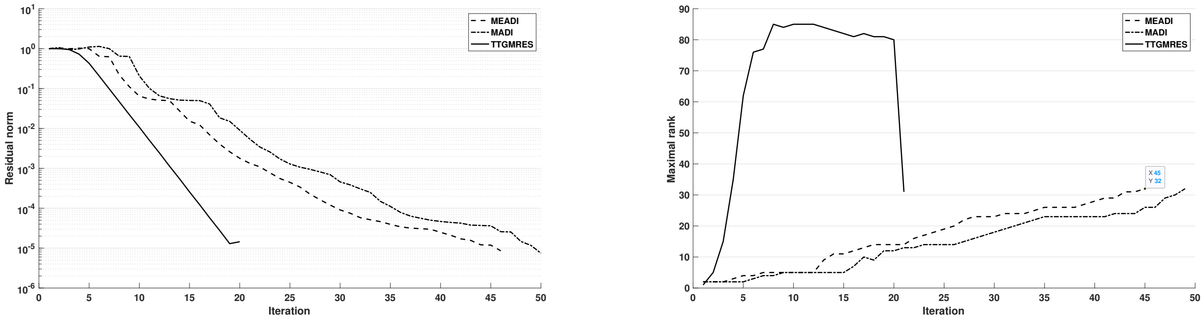

FIGURE 2 TTGMRES, MEADI, and MADI residual histories and maximum TT-rank histories for the cookies problem. The maximum TT-rank of the final approximated solution is appended to the maximum TT-ranks for each method.

Figure 2 depicts the convergence history and the maximum TT-rank of intermediate tensors of our proposed methods compared to TTGMRES. Although TTGMRES converges faster, the TT-ranks of the Krylov basis vectors increase rapidly compared to the very moderate increase in our proposed methods. The largest TT-rank obtained in our methods corresponds to the TT-rank of the solution, while it is three times larger for TTGMRES. Therefore, a TTGMRES has much more costly iterations and storage requirement than compared to our proposed methods. Figure 1 illustrates the first four columns of the first TT-core of the approximate solution obtained by using our solver. Each column stands for a basis vector of the low dimensional subspace approximating the solution manifold of (5).

4.2 | Heat equation.

We consider in this section a parametric heat equation. Let $\Omega_1 \subset \mathbb{R}^2$ be the L-shaped space domain shown in Fig. 3. A point is denoted by $x^{(1)} = (x_1^{(1)}, x_2^{(1)}) \in \Omega_1$. The time domain is $\Omega_2 = [0, \tau]$. Let the parameters α, ν be respectively in $\Omega_3 = [0.1, 1]$ and $\Omega_4 = [1, 10]$. The problem reads:

$$\begin{aligned} \partial_t u - \nu \Delta u = 0, &\qquad \text{in } \Omega_1, \\ \partial_n u + \alpha u = 0, &\qquad \text{on } \partial\Omega_1 \setminus \Gamma_1, \\ u = 0 &\qquad \text{on } \Gamma_1, \\ u(x, 0) = u_0(x), & \end{aligned} \tag{6}$$

where we considered the initial condition to be given by the function: $u_0 = x_1^{(1)}$.

The first step towards the problem discretization consists in writing the weak formulation of the problem. Let

$$\mathcal{V} = \left\{ v \in H^1(\Omega_1) | v = 0 \text{ on } \Gamma_1 \right\}.$$

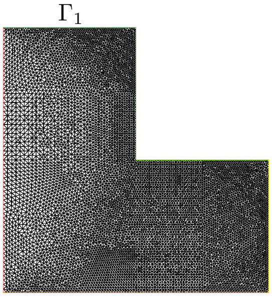

FIGURE 3 Mesh used for the test case presented in Section 4.2, consisting of 7,887 degrees of freedom.

Then, it holds:

$$\int_{\Omega_1} \partial_t u v \, dx^{(1)} + \nu \int_{\Omega_1} \nabla u \cdot \nabla v \, dx^{(1)} + \alpha\nu \int_{\partial\Omega_1 \setminus \Gamma_1} uv \, ds^{(1)} = 0, \quad \forall v \in \mathcal{V}. \tag{7}$$

We now introduce the discretization for the different domains. Concerning the space we used a Galerkin approximation by means of P1 finite elements, denoting the test functions $v_{N_1} \in V_{N_1} \subseteq \mathcal{V}$, where the last inclusion holds also in the sense of density. Concerning the time, we used a classical finite difference scheme, and for the parameters a collocation method (reducing simply to point values in the present case). Let the solution u be approximated by u_N, whose expression reads:

$$u_N(x, t, \alpha, \nu) = \sum_{i_1=1}^{N_1} \hat{u}_{i_1}(t, \alpha, \nu) v_{i_1}(x). \tag{8}$$

This is injected into Eq.(7) leading to the semi-discrete formulation:

$$\sum_{i_1=1}^{N_1} M(j_1, i_1)\partial_t \hat{u}_{i_1} + \nu K(j_1, i_1)\hat{u}_{i_1} + \nu\alpha R(j_1, i_1)\hat{u}_{i_1} = 0,$$

where M, K, R are the mass, stiffness and Robin matrices respectively, whose expression is given by:

$$M(j_1, i_1) = \int_{\Omega^{(1)}} v_{i_1} v_{j_1} \, dx^{(1)},$$

$$K(j_1, i_1) = \int_{\Omega^{(1)}} \nabla v_{i_1} \cdot \nabla v_{j_1} \, dx^{(1)},$$

$$R(j_1, i_1) = \int_{\partial\Omega_1 \setminus \Gamma_1} v_{i_1} v_{j_1} \, ds^{(1)}.$$

We now discretize in time and parameters:

$$\hat{u}_{i_1 i_2 i_3 i_4} = \hat{u}_{i_1}(t_{i_2}, \alpha_{i_3}, \nu_{i_4}).$$

When discretizing in time we use a first order implicit Euler scheme, and, with a slight abuse of notation, we denote $\hat{u}_{i_1,\dots,i_4}$ the approximation of $\hat{u}_{i_1}(t_{i_2}, \alpha_{i_3}, \nu_{i_4})$:

$$\sum_{i_1=1}^{N_1} M(j_1, i_1) \frac{\hat{u}_{i_1 i_2+1\ i_3 i_4} - \hat{u}_{i_1 i_2\ i_3 i_4}}{\Delta t} + \nu_{i_4} K(j_1, i_1)\hat{u}_{i_1 i_2+1\ i_3 i_4} + \nu_{i_4}\alpha_{i_3} R_{j_1 i_1}\hat{u}_{i_1 i_2+1\ i_3 i_4} = 0,$$

with the initial condition:

$$\sum_{i_1=1}^{N_1} M(j_1, i_1)\hat{u}_{i_1\ 1\ i_3 i_4} = \int_{\Omega_1} u_0 v_{j_1} \, dx^{(1)}.$$

which is the discrete formulation for the fourth order tensor $\hat{u}$. Remark that the equation in its discrete formulation can be written as a multi-linear problem, whose expression is:

$$\left[(M \otimes T \otimes I_{N_3} \otimes I_{N_4}) + (K \otimes I_{N_2} \otimes I_{N_3} \otimes D_\nu) + (R \otimes I_{N_2} \otimes D_\alpha \otimes D_\nu)\right] \hat{u} = b,$$

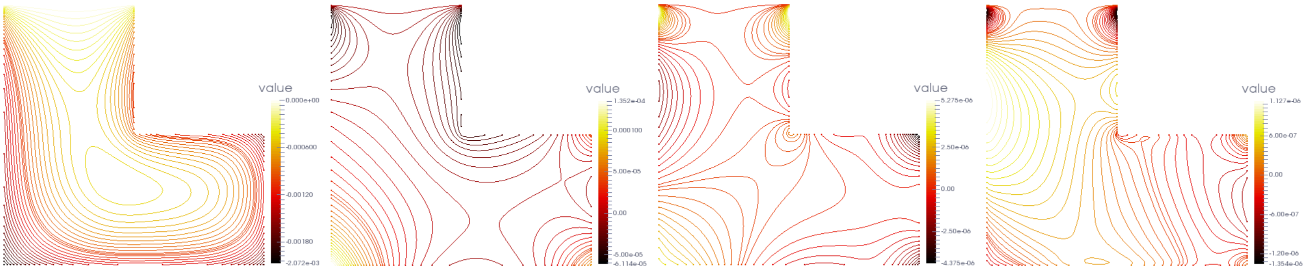

FIGURE 4 First four fibers of the TT-core associated to the space variable, reconstructed by using the finite element space, for the test-case presented in Section4.2. Contour plots with 40 values between the minimum and the maximum.

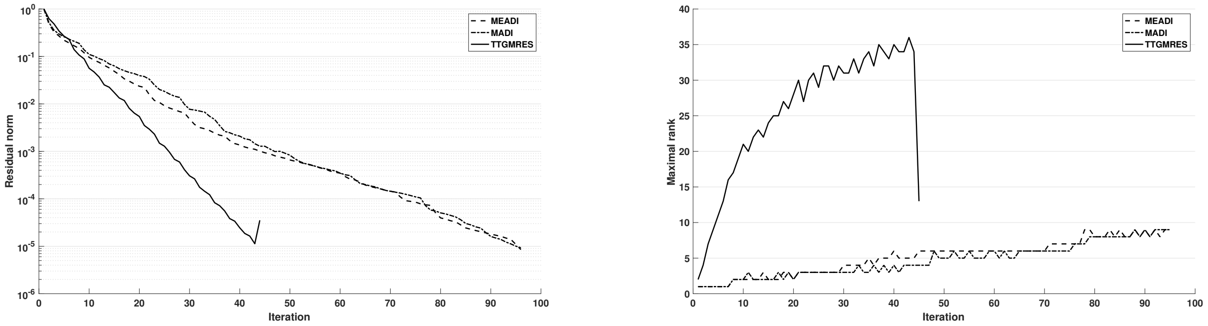

FIGURE 5 TTGMRES, MEADI, and MADI residual histories and maximum TT rank histories for the heat equation. The maximum TT-rank of the final approximated solution is appended to the maximum TT-ranks for each method.

where T is bidiagonal matrix with ones on the diagonal and minus ones on the subdiagonal; D_α and D_ν are diagonal matrices with the ith diagonal element is α_i and ν_i, respectively; and b is $u_0 \otimes e_{N_2} \otimes 1_{N_3} \otimes 1_{N_3}$, where u_0 is the discrete initial state and e_m is the first canonical basis vector of length m. We choose to represent the solution tensor $\hat{u}$ by means of a Tensor Train format, which reads:

$$\hat{u}_{i_1 i_2 i_3 i_4} = \sum_{k_1=1}^{r_1} \sum_{k_2=1}^{r_2} \sum_{k_3=1}^{r_3} U^{(1)}_{i_1\ k_1} U^{2}_{k_1 i_2 k_2} U^{(3)}_{k_2 i_3 k_3} U^{(4)}_{k_3 i_4},$$

for $r_1, r_2, r_3 \in \mathbb{N}^*$, to be determined along with the solution cores $U^{(1,2,3,4)}$. We discuss now the multi-linear problem solution. Remark that the matrix R is not invertible, hence the last rank-one operator cannot be inverted as it is. To overcome this problem we introduce a shift of the form: $R = R + I_{N_1} - I_{N_1}$, leading to:

$$\left[(M \otimes T \otimes I_{N_3} \otimes I_{N_4}) + (K \otimes I_{N_2} \otimes I_{N_3} \otimes D_\nu) + ((R + I_{N_1}) \otimes I_{N_2} \otimes D_\alpha \otimes D_\nu) - (I_{N_1} \otimes I_{N_2} \otimes D_\alpha \otimes D_\nu)\right] \hat{u} = b,$$

At the expense of one extra term in the operator, we can apply straightforwardly the method detailed in the previous section. Note that the linear system is highly nonsymmetric due to the presence of the time marching matrix T.

We use $\mathcal{P}1$ finite elements for the spatial variable resulting in 7,887 degrees of freedom. Ten time steps are used so that $N_2 = 10$, and 1,000 samples for each parameter ν and α are taken from $[1, 10]$ so that $N_3 = N_4 = 1{,}000$. Similarly to the previous experiment, we compare our proposed methods to TTGMRES. Figure 5 depicts the history of the convergence and the maximum TT-ranks of the intermediate tensors arising in all algorithms. Here again, TTGMRES converges faster with a rapidly increasing TT-ranks leading to costly iterations and high memory requirements. The maximum TT-rank obtained in TTGMRES is more than three times the maximum TT-rank of all intermediate tensors arising in our method. Figure 4 plots the first four columns of the first core of the obtained solution from our solver. These solution represent four basis vectors in the solution manifold of (6).

5 | CONCLUSION

In the present work we proposed a method to solve multi-linear systems based on the principle of Alternating Direction. This method could be consider, to some extent, as a first step towards the definition of more general extended Krylov methods for multi-linear systems. The main advantages of the proposed approach are twofold: first, we can apply it to systems which are not necessarily symmetric; second, we can exploit preconditioners specifically tailored for the problems at hand. The strategy of acceleration of the iteration, based on adapting the compression tolerances dynamically, proved to be effective and made the approach competitive in terms of computational cost and memory when compared to the Krylov based TTGMRES method. However, some hyper-parameters appearing in the method definition are arbitrary and might affect the performances both in terms of computational cost and memory use. The investigation of this point and the definition of a more general extended Krylov approach will be the object of further works.

Acknowledgments:

Damiano Lombardi acknowledges support from ANR grant ADAPT 18-CE46-0001.

References

1. Hackbusch W, and Khoromskij BN. Tensor-product approximation to operators and functions in high dimensions. Journal of Complexity. 2007;**23**(4-6):697–714.

2. Kolda TG, and Bader BW. Tensor decompositions and applications. SIAM review. 2009;**51**(3):455–500.

3. Grasedyck L, Kressner D, and Tobler C. A literature survey of low-rank tensor approximation techniques. GAMM-Mitteilungen. 2013;**36**(1):53–78.

4. Hackbusch W. Numerical tensor calculus. Acta numerica. 2014;**23**:651–742.

5. Austin W, Ballard G, and Kolda TG. Parallel tensor compression for large-scale scientific data. In: 2016 IEEE international parallel and distributed processing symposium (IPDPS). IEEE; 2016. p. 912–922.

6. Falcó A, Hackbusch W, and Nouy A. Tree-based tensor formats. SeMA Journal. 2021;**78**(2):159–173.

7. Al Daas H, Ballard G, and Benner P. Parallel Algorithms for Tensor Train Arithmetic. arXiv. https://arxiv.org/abs/2011.06532v2, preprint arXiv:2011.06532; 2021.

8. Al Daas H, Ballard G, Cazeaux P, Hallman E, Miedlar A, Pasha M, et al.. Randomized algorithms for rounding in the Tensor-Train format. arXiv. https://arxiv.org/abs/, preprint arXiv:; 2021.

9. Kressner D, Steinlechner M, and Vandereycken B. Low-rank tensor completion by Riemannian optimization. BIT Numerical Mathematics. 2014;**54**(2):447–468.

10. Rauhut H, Schneider R, and Stojanac Ž. Tensor completion in hierarchical tensor representations. In: Compressed sensing and its applications. Springer; 2015. p. 419–450.

11. Haberstich C, Nouy A, and Perrin G. Active learning of tree tensor networks using optimal least-squares. arXiv preprint arXiv:210413436. 2021;.

12. Dolgov SV, Khoromskij BN, and Oseledets IV. Fast solution of parabolic problems in the tensor train/quantized tensor train format with initial application to the Fokker–Planck equation. SIAM Journal on Scientific Computing. 2012;**34**(6):A3016–A3038.

13. Zhang Z, Yang X, Oseledets IV, Karniadakis GE, and Daniel L. Enabling high-dimensional hierarchical uncertainty quantification by ANOVA and tensor-train decomposition. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 2014;**34**(1):63–76.

14. Dolgov S, Khoromskij BN, Litvinenko A, and Matthies HG. Polynomial chaos expansion of random coefficients and the solution of stochastic partial differential equations in the tensor train format. SIAM/ASA Journal on Uncertainty Quantification. 2015;**3**(1):1109–1135.

15. Nouy A. Low-rank tensor methods for model order reduction. arXiv preprint arXiv:151101555. 2015;.

16. Bachmayr M, and Dahmen W. Adaptive low-rank methods: problems on Sobolev spaces. SIAM Journal on Numerical Analysis. 2016;**54**(2):744–796.

17. Corona E, Rahimian A, and Zorin D. A tensor-train accelerated solver for integral equations in complex geometries. Journal of Computational Physics. 2017;**334**:145–169.

18. Loukrezis D, Römer U, Casper T, Schöps S, and De Gersem H. High-dimensional uncertainty quantification for an electrothermal field problem using stochastic collocation on sparse grids and tensor train decompositions. International Journal of Numerical Modelling: Electronic Networks, Devices and Fields. 2018;**31**(2):e2222.

19. Kressner D, and Tobler C. Krylov subspace methods for linear systems with tensor product structure. SIAM J Matrix Anal Appl. 2009/10;**31**(4):1688–1714. Available from: https://doi.org/10.1137/090756843.

20. Dolgov SV, and Savostyanov DV. Alternating minimal energy methods for linear systems in higher dimensions. SIAM Journal on Scientific Computing. 2014;**36**(5):A2248–A2271.

21. Weinhandl R, Benner P, and Richter T. Low-rank linear fluid-structure interaction discretizations. ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik. 2020 2021/10/10;**100**(11):e201900205.

22. Grasedyck L. Existence and computation of low Kronecker-rank approximations for large linear systems of tensor product structure. Computing. 2004;**72**(3-4):247–265.

23. Dahmen W, DeVore R, Grasedyck L, and Süli E. Tensor-Sparsity of Solutions to High-Dimensional Elliptic Partial Differential Equations. Foundations of Computational Mathematics. 2016;**16**(4):813–874.

24. Oseledets IV. Tensor-train decomposition. SIAM Journal on Scientific Computing. 2011;**33**(5):2295–2317.

25. De Lathauwer L, and Nion D. Decompositions of a higher-order tensor in block terms—Part III: Alternating least squares algorithms. SIAM journal on Matrix Analysis and Applications. 2008;**30**(3):1067–1083.

26. Espig M, Hackbusch W, and Khachatryan A. On the convergence of alternating least squares optimisation in tensor format representations. arXiv preprint arXiv:150600062. 2015;.

27. Comon P, Luciani X, and De Almeida AL. Tensor decompositions, alternating least squares and other tales. Journal of Chemometrics: A Journal of the Chemometrics Society. 2009;**23**(7-8):393–405.

28. Dolgov S, and Scheichl R. A Hybrid Alternating Least Squares–TT-Cross Algorithm for Parametric PDEs. SIAM/ASA Journal on Uncertainty Quantification. 2019;**7**(1):260–291.

29. Xiao C, Yang C, and Li M. Efficient Alternating Least Squares Algorithms for Low Multilinear Rank Approximation of Tensors. Journal of Scientific Computing. 2021;**87**(3):1–25.

30. Dolgov SV. Alternating minimal energy approach to odes and conservation laws in tensor product formats. arXiv preprint arXiv:14038085. 2014;.

31. Liang M, and Dai L. Alternating Minimization Methods for Solving Multilinear Systems. Mathematical Problems in Engineering. 2021;**2021**.

32. Bader BW. Tensor-Krylov methods for solving large-scale systems of nonlinear equations. SIAM journal on numerical analysis. 2005;**43**(3):1321–1347.

33. Kressner D, and Tobler C. Krylov subspace methods for linear systems with tensor product structure. SIAM journal on matrix analysis and applications. 2010;**31**(4):1688–1714.

34. Kressner D, and Tobler C. Low-rank tensor Krylov subspace methods for parametrized linear systems. SIAM Journal on Matrix Analysis and Applications. 2011;**32**(4):1288–1316.

35. Savas B, and Eldén L. Krylov-type methods for tensor computations I. Linear Algebra and its Applications. 2013;**438**(2):891–918.

36. Huang B, Xie Y, and Ma C. Krylov subspace methods to solve a class of tensor equations via the Einstein product. Numerical Linear Algebra with Applications. 2019;**26**(4):e2254.

37. Beik FP, and Najafi-Kalyani M. A preconditioning technique in conjunction with Krylov subspace methods for solving multilinear systems. Applied Mathematics Letters. 2021;**116**:107051.

38. Ding W, and Wei Y. Solving Multi-linear Systems with M M-Tensors. Journal of Scientific Computing. 2016;**68**(2):689–715.

39. Cui LB, Li MH, and Song Y. Preconditioned tensor splitting iterations method for solving multi-linear systems. Applied Mathematics Letters. 2019;**96**:89–94.

40. Zhang Y, Liu Q, and Chen Z. Preconditioned Jacobi type method for solving multi-linear systems with M-tensors. Applied Mathematics Letters. 2020;**104**:106287.

41. Rakhuba M. Robust Alternating Direction Implicit Solver in Quantized Tensor Formats for a Three-Dimensional Elliptic PDE. SIAM Journal on Scientific Computing. 2021;**43**(2):A800–A827.

42. Simoncini V. A new iterative method for solving large-scale Lyapunov matrix equations. SIAM J Sci Comput. 2007;**29**(3):1268–1288.

43. Tobler C. Low-rank Tensor Methods for Linear Systems and Eigenvalue Problems. ETH Zurich; 2012. Available from: http://sma.epfl.ch/~anchpcommon/students/tobler.pdf.

□