Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Alternating minimal energy methods for linear systems in higher dimensions. Part I: SPD systems

by

Sergey Dolgov and Dmitry Savostyanov

Preprint no.: 10 2013

Alternating minimal energy methods for linear systems in higher dimensions. Part I: SPD systems*

Sergey V. Dolgov† and Dmitry V. Savostyanov‡

January 25, 2013

Abstract

We introduce a family of numerical algorithms for the solution of linear system in higher dimensions with the matrix and right hand side given and the solution sought in the tensor train format. The proposed methods are rank–adaptive and follow the alternating directions framework, but in contrast to ALS methods, in each iteration a tensor subspace is enlarged by a set of vectors chosen similarly to the steepest descent algorithm. The convergence is analysed in the presence of approximation errors and the geometrical convergence rate is estimated and related to the one of the steepest descent. The complexity of the presented algorithms is linear in the mode size and dimension and the convergence demonstrated in the numerical experiments is comparable to the one of the DMRG–type algorithm.

Keywords: high–dimensional problems, tensor train format, ALS, DMRG, steepest descent, convergence rate, superfast algorithms.

1 Introduction

Linear systems arising from high–dimensional problems usually can not be solved by standard numerical algorithms. If the equation is considered in d dimensions on a $n_1 \times n_2 \times \ldots \times n_d$ grid, the number of unknowns $n_1 \ldots n_d$ scales exponentially with d, and even for moderate dimension d and mode sizes n_k the numerical complexity lays far beyond the technical possibilities of modern workstations and parallel systems. To make the problem tractable, different approximations are proposed, including sparse grids [38, 3] and tensor product methods [24, 22, 23, 14]. In this paper we consider the linear system $Ax = y$, where the matrix A and right-hand-side y are given and approximate solution x is sought in the *tensor train* (TT) format. Methods based on the TT format, also known as a *linear tensor*

*Partially supported by RFBR grants 12-01-00546-a, 11-01-12137-ofi-m-2011, 11-01-00549-a, 12-01-33013, 12-01-31056, Russian Fed. Gov. contracts No. П1112, 14.740.11.0345, 16.740.12.0727 at Institute of Numerical Mathematics, Russian Academy of Sciences, and EPSRC grant EP/H003789/1 at the University of Southampton.

†Max-Planck-Institut für Mathematik in den Naturwissenschaften, Inselstr. 22-26, D-04103 Leipzig, Germany (sergey.v.dolgov@gmail.com).

‡University of Southampton, Department of Chemistry, Highfield Campus, Southampton SO17 1BJ, United Kingdom (dmitry.savostyanov@gmail.com)

network, are novel and particularly interesting among all tensor product methods due to their robustness and simplicity.

The numerical optimization on tensor networks was first considered in quantum physics community by S. White [42], who introduces the *matrix product states* (MPS) formalism to represent the ground state of a spin system together with the *density matrix renormalization group* (DMRG) optimization scheme. The tensor train format and some computational methods were independently re-discovered in the papers of Oseledets and Tyrtyshnikov (see [30] and references therein) until the results of White et. al. were popularized in the numerical mathematics community by R. Schneider [18]. The questions concerning the convergence properties of alternating schemes for different tensor product formats were immediately raised and studied. The experimental results from quantum physics show the notably fast convergence of DMRG for the ground state problem, i.e., finding the minimal eigenstate of a system, but give no theoretical justification for this observation. The *alternating least squares* (ALS) algorithm was used in multilinear analysis for the computation of *canonical* tensor decomposition since early results of Hitchcock [17] and was known for its monotone but very slow convergence. For ALS there is also a lack of convergence estimates both in the classical papers [16, 4], and in the recent ones, where ALS was applied to the Tucker model [5, 33], tensor trains [29], hierarchical Tucker format [25] and high–dimensional interpolation [34].

In recent papers by Uschmajew [41, 35] the local convergence of ALS is proven for the canonical and tensor train decompositions. This is a major theoretical breakthrough, which unfortunately does not immediately lead to practical algorithms due to the local character of convergence studied, unjustified assumptions on the structure of the Hessian, and very strong requirements on the accuracy of the initial guess. The convergence rate of ALS is difficult to estimate partly due to the complex geometrical structure of manifolds defined by tensor networks. This problem is now approached from several directions, and we might expect new results soon [12, 11].

In contrast to ALS schemes which operate on manifolds of fixed dimension, the DMRG algorithm changes the ranks of a tensor format. This allows to choose the ranks adaptively to the desired error threshold or the accuracy of the result and develop more practical algorithms which do not rely on a priori choice of ranks. The DMRG was adopted for novel tensor formats (see references above) and new problems, including adaptive high–dimensional interpolation [37] and solution of linear systems [9, 18]. The geometrical analysis, eg the convergence of the nonlinear Gauss–Seidel method, is however even more difficult when the dimensions of underlying manifolds are not fixed.

Apart of working with the tensor format structure directly, like ALS and DMRG do, standard algorithms from numerical linear algebra can be applied with tensor approximations and other tensor arithmetics. Following this paradigm, the solution of linear problems in tensor product formats was addressed in [36, 1, 6]. The usual considerations of linear algebra can be used in this case to analyze the convergence. A first notable example is the method of conjugate–gradient type for the Rayleigh quotient minimization in higher dimensions, for which the global convergence was proven by O. Lebedeva [27].

We develop a framework which combines the ALS optimization steps (ranks are fixed, convergence estimates not yet possible) with the steps when the tensor subspaces are increased and the ranks of a tensor format grow. Choosing the new vectors in accordance with standard linear algebra algorithms, we recast the classical convergence estimates for the proposed algorithm in higher dimensions. In this paper we consider the case of sym-

metrical positive definite (SPD) matrices and analyze the convergence in the A–norm, i.e. minimize the *energy function*. The *basis enrichment* choice follows the steepest descent (SD) algorithm and the convergence of the resulted method is analyzed with respect to the one of steepest descent. We show that the basis enrichment step combined with the ALS step can be seen as a certain computationally cheap approximation of the DMRG step. The complexity of the resulted method is equal to the one of ALS and is linear in the mode size and dimension. Our choice of the basis enrichment appears to be very good for practical computations, and for the considered numerical examples the proposed methods converge almost as fast as the DMRG algorithm.

Summarizing the above, the proposed algorithms have (1) proven geometrical convergence with the estimated rate, (2) practical convergence compared to the one of DMRG, (3) numerical complexity compared to the one of ALS.

The paper is organized as follows.

In Section 2 we introduce the tensor train notation and necessary definitions.

In Section 3 we introduce the basic notation for ALS and DMRG schemes. We also study how the modification of one TT–block affects the ALS problem for its neighbor and describe this in terms of the Galerkin correction method.

In Section 4 we develop the family of steepest descent methods for the problems in one, two and many dimensions. The proposed methods have an inner–outer structure, i.e., a steepest descent step in d dimensions is followed by a steepest descent step in $d-1$ dimension, etc, cf. the interpolation algorithms [32, 13]. The convergence of the recursive algorithms in higher dimensions is analyzed using the Galerkin correction framework. The effect of roundoff / approximation errors is also studied.

Since we make no assumptions on the TT–ranks of the solution, the ranks of the vectors in the proposed algorithms can grow at each iteration and make the algorithm inefficient. In Section 5 we discuss the implementation details, in particular the steps when the tensor approximation is required to reduce the ranks.

In Section 6 the model numerical experiments demonstrate the efficiency of the method proposed and compare it with other algorithms mentioned in the paper.

2 Tensor train notation and definitions

The tensor train (TT) representation of a d-dimensional tensor $x = [x(i_1, \dots, i_d)]$ is written as the following multilinear map (cf. [35])

$$\begin{aligned} x = \tau(\bar{X}) &= \tau(X^{(1)}, \dots, X^{(d)}), \\ x(i_1, \dots, i_d) &= X^{(1)}_{\alpha_0,\alpha_1}(i_1) X^{(2)}_{\alpha_1,\alpha_2}(i_2) \dots X^{(d-1)}_{\alpha_{d-2},\alpha_{d-1}}(i_{d-1}) X^{(d)}_{\alpha_{d-1},\alpha_d}(i_d), \end{aligned} \tag{1}$$

where $i_k = 1, \dots, n_k$ are the *mode* (physical) indices, $\alpha_k = 1, \dots, r_k$ are the *rank* indices, $X^{(k)}$ are the tensor train *cores* (TT–cores) and $\bar{X} = (X^{(1)}, \dots, X^{(d)})$ denote the whole tensor train. Here and later we use the Einstein summation convention [10], which assumes a summation over every pair of repeated indices. Therefore, in Eq. (1) we assume the summation over all rank indices α_k, $k = 1, \dots, d-1$. We also imply the *closed boundary conditions* $r_0 = r_d = 1$ to make the right–hand side a scalar for each $(i_1, \dots, i_d)$. Eq. (1) is written in the elementwise form, i.e., the equation is assumed over all free (unpaired) indices. It is often convenient in higher dimensions and will be used throughout the paper.

The indices can be written either in the subscript x_j or in brackets $x(j)$. For the summation, there is no difference. The subscripted indices are usually considered as *row and column* indices of a matrix, while the indices in brackets are seen as *parameters*. For example, each TT–core $X^{(k)}$ is considered as a parameter-dependent on i_k matrix with the row index α_{k-1} and the column index α_k as follows

$$X^{(k)} = [X^{(k)}_{\alpha_{k-1},\alpha_k}(i_k)] \in \mathbb{C}^{r_{k-1}\times n_k\times r_k}, \qquad X^{(k)}(i_k) \in \mathbb{C}^{r_{k-1}\times r_k}.$$

In our notation $X^{(k)}(i_k)$ is a matrix, for which standard algorithms like orthogonalization (QR) and singular value decomposition (SVD) can be applied. We will freely transfer indices from subscripts to brackets in order to make the equations easier to read or to emphasize a certain transposition of elements in tensors. It brings the notations in consistence with previous papers on the numerical tensor methods, e.g. [18, 9, 8, 35] and others.

We will reshape arrays into matrices and vectors by using the *index grouping*, i.e., combining two or more indices $\alpha, \ldots, \zeta$ in a single multi-index $\overline{\alpha\ldots\zeta}$. Following [35] we define *interface* matrices $X^{\leqslant k} \in \mathbb{C}^{n_1\ldots n_k\times r_k}$ and $X^{>k} \in \mathbb{C}^{r_k\times n_{k+1}\ldots n_d}$ as follows

$$\begin{aligned} X^{\leqslant k}(\overline{i_1 i_2\ldots i_k}, \alpha_k) &= X^{(1)}_{\alpha_1}(i_1)X^{(2)}_{\alpha_1\alpha_2}(i_2)\ldots X^{(k)}_{\alpha_{k-1},\alpha_k}(i_k),\\ X^{>k}(\alpha_k, \overline{i_{k+1}\ldots i_{d-1}i_d}) &= X^{(k+1)}_{\alpha_k,\alpha_{k+1}}(i_{k+1})\ldots X^{(d-1)}_{\alpha_{d-2},\alpha_{d-1}}(i_{d-1})X^{(d)}_{\alpha_{d-1}}(i_d), \end{aligned} \tag{2}$$

and similarly for symbols $X^{<k}$ and $X^{\geqslant k}$. Using the τ notation defined in (1) we can write $x = \tau(X^{\leqslant k}, X^{>k})$. For a tensor $x = [x(i_1,\ldots,x_d)]$ we also define the *unfolding matrix*, which consists of the entries of the original tensor as follows

$$X^{\{k\}}(\overline{i_1\ldots i_k}, \overline{i_{k+1}\ldots i_d}) = x(i_1,\ldots,i_d), \qquad X^{\{k\}} \in \mathbb{C}^{n_1\ldots n_k\times n_{k+1}\ldots n_d}.$$

For x in the TT–format (1) it holds $X^{\{k\}} = X^{\leqslant k}X^{>k}$ and therefore $\operatorname{rank} X^{\{k\}} = r_k$. In [30] the reverse is proven: for any tensor x there exists the representation (1) with TT–ranks $r_k = \operatorname{rank} X^{\{k\}}$. This gives the term *TT–rank* the definite algebraic meaning. As a result, the tensor train representation of fixed TT–ranks yields a closed manifold, and the rank-$(r_1, ..., r_{d-1})$ approximation problem is well–posed. We can also approximate a given tensor by a tensor train with quasi–optimal ranks using a simple and robust approximation (rank truncation, or *tensor rounding*) algorithm [30]. This is the case for all tensor networks without cycles, eg. Tucker [40], HT [15], QTT-Tucker [8], etc. In contrast, the MPS formalism originally assumes the *periodic boundary conditions* $\alpha_0 = \alpha_d$ and sum over these indices, which leads to $\operatorname{Tr}(X^{(1)}\ldots X^{(d)})$, where all matrices can be shifted in cycle under the trace. The optimization in such type of *tensor networks* is difficult, because they form unclosed manifolds and the best approximation does not always exist.

The tensor train representation of the matrix is made similarly with the TT–cores depending on two parameters i_k, j_k. Hence, $x = \tau(\bar{X})$ is sought in the form (1) and A and y given in the TT–format as follows

$$\begin{aligned} A(i_1,\ldots,i_d;\, j_1,\ldots,j_d) &= A^{(1)}(i_1,j_1)\ldots A^{(d)}(i_d,j_d),\\ y(i_1,\ldots,i_d) &= Y^{(1)}(i_1)\ldots Y^{(d)}(i_d). \end{aligned} \tag{3}$$

For A and x given in the TT–format, the matrix-vector product $c = Ax$ is also a TT–format computed as follows

$$c(i_1,\ldots,i_d) = \left(A^{(1)}(i_1,j_1)\otimes X^{(1)}(j_1)\right)\ldots\left(A^{(d)}(i_d,j_d)\otimes X^{(d)}(j_d)\right),$$

where $\otimes$ denotes the tensor (Kronecker) product of two matrices defined as follows

$$A = \big[A(i,j)\big], \quad B = \big[B(p,q)\big], \qquad C = A \otimes B = \big[C(\overline{ip},\overline{jq})\big] = \big[A(i,j)B(p,q)\big].$$

We refer to [30] for more details on basic tensor operations in the TT–format.

In this paper we will use standard l_2 scalar product $(\cdot,\cdot)$ and the A–scalar product $(\cdot,\cdot)_A$ defined by a symmetrical positive definite (SPD) matrix A as follows

$$(u,v)_A = u^* A v, \qquad \|u\|_A^2 = (u,u)_A.$$

For a given nonsingular matrix U we define the A–orthogonal projector R_U as follows: for all v and all $w \in \operatorname{span} U$ it holds

$$R_U v \in \operatorname{span} U, \qquad (w, R_U v)_A = (w,v)_A, \qquad R_U = U(U^*AU)^{-1}U^*A.$$

We will use vector notations for mode indices $\mathbf{i} = (i_1,\dots,i_d)$ and rank indices $\mathbf{r} = (r_0,\dots,r_d)$. We also denote the subspace of tensor trains $\bar{X} = (X^{(1)},\dots,X^{(d)})$ with tensor ranks $\mathbf{r}$ as

$$\mathcal{T}_{\mathbf{r}} = \bigtimes_{i=1}^{d} \mathbb{C}^{r_{i-1}\times n_i \times r_i}.$$

3 Alternating minimization methods

3.1 ALS–like minimization with fixed TT–ranks

The MPS formalism was proposed in Quantum Physics, where the representation (1) was used for the minimization of the Rayleigh quotient $(x,Ax)/(x,x)$. Similarly, the solution of a linear system $Ax = y$ with $A = A^*$ can be sought through the minimization of an *energy function*

$$J(x) = \|x_* - x\|_A^2 = (x,Ax) - 2\Re(x,y) + \text{const}, \tag{4}$$

where x_* denotes the exact solution. We consider the Hermitian matrix $A = A^*$ and the right-hand side y given in the TT–format (3), and solve the minimization problem with x sought in the TT–format (1) with fixed TT–ranks $\mathbf{r}$, i.e., $\bar{X}_* = \arg\min_{\bar{X}\in\mathcal{T}_{\mathbf{r}}} J(\tau(\bar{X}))$.This heavy nonlinear minimization problem can hardly be solved unless a (very) accurate initial guess is available (see, eg. [35]). To make it tractable, we can use the alternating linear optimization framework and substitute the global minimization over the tensor train $\bar{X} \in \mathcal{T}_{\mathbf{r}}$ by the linear minimization over all cores $X^{(1)},\dots,X^{(d)}$ subsequently in a cycle. Solving the *local* problem we assume that all cores but k–th of the current tensor train $\bar{X} = (X^{(1)},\dots,X^{(d)})$ are 'frozen', and the minimization is done over $X^{(k)}$ as follows

$$\begin{aligned} &\bar{X}_{\text{new}} = (X^{(1)},\dots,X^{(k-1)},X^{(k)}_{\text{new}},X^{(k+1)},\dots,X^{(d)}), \qquad \text{where} \\ &X^{(k)}_{\text{new}} = \arg\min_{X^{(k)}} J(\tau(\bar{X})), \qquad \text{s.t.} \quad X^{(k)} \in \mathbb{C}^{r_{k-1}\times n_k \times r_k}. \end{aligned} \tag{5}$$

Clearly, the energy function does not grow during the sequence of ALS updates and the solution will converge to a local minimum.

To write each ALS step as a linear problem, let us stretch all entries of the TT–core $X^{(k)}$ in the vector $x_k(\overline{\alpha_{k-1}i_k\alpha_k}) = X^{(k)}_{\alpha_{k-1},\alpha_k}(i_k)$. From (1) we see that $x = \mathcal{X}_{\neq k}x_k$, where

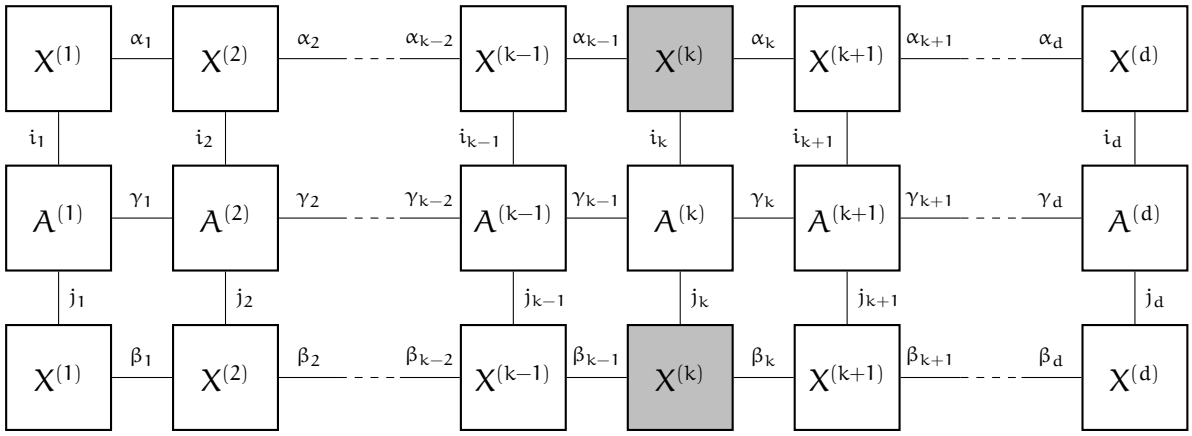

Figure 1: Tensor network corresponding to the quadratic form (Ax, x) with matrix A and vector x given in the tensor train format. The boxes are tensors with lines (legs) denoting indices. Each bond between two tensors assumes a summation over the join index.

$\mathcal{X}_{\neq k} = \mathcal{P}_{\neq k}(\bar{X})$ is the $n_1 \dots n_d \times r_{k-1} n_k r_k$ matrix defined as follows

$$\begin{aligned} \mathcal{X}_{\neq k}(\overline{i_1 \dots i_d}, \overline{\alpha_{k-1} j_k \alpha_k}) &= X^{(1)}_{\alpha_1}(i_1) \dots X^{(k-1)}_{\alpha_{k-2},\alpha_{k-1}}(i_{k-1}) \delta(i_k, j_k) X^{(k+1)}_{\alpha_k,\alpha_{k+1}}(i_{k+1}) \dots X^{(d)}_{\alpha_{d-1}}(i_d), \\ \mathcal{X}_{\neq k} = \mathcal{P}_{\neq k}(\bar{X}) &= X^{<k} \otimes I_{n_k} \otimes \left(X^{>k}\right)^\top, \end{aligned} \tag{6}$$

where $\delta(i, j)$ is the Kronecker symbol, i.e., $\delta(i, j) = 1$ if $i = j$ and $\delta(i, j) = 0$ elsewhere. If $J(\tau(\bar{X}))$ is considered as a function of x_k, it is also the second-order energy function

$$J(\tau(x)) = (A\mathcal{X}_{\neq k} x_k, \mathcal{X}_{\neq k} x_k) - 2(y, \mathcal{X}_{\neq k} x_k) = (\mathcal{X}^*_{\neq k} A \mathcal{X}_{\neq k} x_k, x_k) - 2(\mathcal{X}^*_{\neq k} y, x_k),$$

where the gradient w.r.t. x_k is zero when[1]

$$\left(\mathcal{X}^*_{\neq k} A \mathcal{X}_{\neq k}\right) x_k = \mathcal{X}^*_{\neq k} y. \tag{7}$$

The solution of the local minimization problem (5) is therefore equivalent to the solution of the original system $Ax = y$ in the *reduced basis* $\mathcal{X}_{\neq k} = \mathcal{P}_{\neq k}(\bar{X})$, defined by (6).

The tensor train representation (1) is non-unique. Indeed, two representations $\bar{X}$ and $\bar{Y}$ map to one tensor $\tau(\bar{X}) = \tau(\bar{Y})$ as soon as

$$Y^{(k)}(i_k) = H^{-1}_{k-1} X^{(k)}(i_k) H_k, \qquad k = 1, \dots, d,$$

where $H_0 = H_d = 1$ and $H_k \in \mathbb{C}^{r_k \times r_k}$, $k = 1, \dots, d-1$, are arbitrary nonsingular matrices. Given a vector in the TT–format $x = \tau(\bar{X})$, any transformation $\mathtt{H} = (H_0, \dots, H_d)$ does not change the energy level since $J(\tau(\bar{X})) = J(\tau(\bar{Y}))$ but gives us some flexibility for the choice of the reduced basis since $\mathcal{P}_{\neq k}(\bar{X}) \neq \mathcal{P}_{\neq k}(\bar{Y})$. The proper choice of the *representation* $\bar{X}$ essentially defines the reduced basis and affects the properties of the *local problem* (7). A prominent transformation $\mathtt{H}$ is the TT–orthogonalization algorithm proposed in [30]. It chooses matrices H_k applying the QR factorization to the reshaped TT–cores, i.e., matrices of size $r_{k-1} \times n_k r_k$ and / or $r_{k-1} n_k \times r_k$. The transformation $\mathtt{H}$ given by the TT–orthogonalization implies the left–orthogonality constrains on TT–cores $Y^{(1)}, \dots, Y^{(k-1)}$ and right–orthogonality

[1] For illustration see Fig. 1, for the detailed derivation see [18, 9].

on $Y^{(k+1)}, \dots, Y^{(d)}$, which results in the orthogonality of the interfaces $Y^{<k}$ and $Y^{>k}$ and hence the reduced basis $\mathcal{Y}_{\neq k} = \mathcal{P}_{\neq k}(\bar{Y})$. Such a *normalization* step will be assumed in many algorithms throughout the paper; in most cases we will do this without introduction of a new representation $\bar{Y}$ just by 'claiming' the necessary orthogonalization pattern of the TT representation we use. If the reduced basis method is applied and such a representation $\bar{X}$ is chosen so that $\mathcal{X}_{\neq k} = \mathcal{P}_{\neq k}(\bar{X}) = \mathcal{P}$ is orthogonal, the spectrum of the reduced matrix $\mathcal{P}^* A \mathcal{P}$ lies between the minimum and maximum eigenvalues of the matrix A. Indeed, using the Rayleigh quotient [19], we write

$$\lambda_{\min}(\mathcal{P}^* A \mathcal{P}) = \min_{\|v\|=1} (\mathcal{P}v, A\mathcal{P}v) = \min_{u \in \operatorname{span} \mathcal{P}, \|u\|=1} (u, Au) \geqslant \min_{\|u\|=1} (u, Au) = \lambda_{\min}(A),$$

and similarly for the maximum values. It follows that the reduced matrix is conditioned not worse than the original, $\operatorname{cond}(\mathcal{X}^*_{\neq k} A \mathcal{X}_{\neq k}) \leqslant \operatorname{cond}(A)$. Therefore, the orthogonality of TT–cores ensures the stability of local problems and we will silently assume this for all reduced problems in this paper.

To conclude this part, let us calculate the complexity of the local problem (7). As was pointed out in [9], either a direct elimination, or an iterative linear solver with fast matrix-by-vector products (*matvecs*) may be applied. If the direct solution method is used, the costs which are required to form the $r_{k-1} n_k r_k \times r_{k-1} n_k r_k$ matrix of the local problem (7) are smaller than the complexity of the Gaussian elimination, i.e., the overall cost is $\mathcal{O}(n^3 r^6)$.[2] If an iterative method is used to solve the local problem, one multiplication $\mathcal{X}^*_{\neq k} A \mathcal{X}_{\neq k}$ requires $\mathcal{O}(n r_A r^3 + n^2 r_A^2 r^2)$ operations, where r and r_A denote the TT–rank of the current solution x and the matrix A, respectively. Careful implementation of the matvec is essential to reach this complexity, see [9] for details. The complexity of the normalization step is only $\mathcal{O}(dnr^3)$ operations and can be neglected.

3.2 DMRG–like minimization and adaptivity of TT–ranks

In practical numerical work the TT–ranks of the solution are usually not known in advance, which puts a restriction on the use of the methods with fixed TT–ranks. The underestimation of TT–ranks leads to a low accuracy of the solution, while the overestimation results in a large computational overhead. This motivates the development of methods which can choose and modify the TT–ranks on–the–fly adaptively to the desired accuracy level. A prominent example of such method is the Density Matrix Renormalization Group (DMRG) algorithm [42], developed in the quantum physics community for the solution of a ground state problem. DMRG performs similarly to the ALS but at each step combines two succeeding blocks $X^{(k)}$ and $X^{(k+1)}$ into one *superblock*

$$w_k(\overline{\alpha_{k-1} i_k i_{k+1} \alpha_{k+1}}) = W^{(k)}_{\alpha_{k-1}, \alpha_{k+1}}(\overline{i_k i_{k+1}}), \qquad W^{(k)}(\overline{i_k i_{k+1}}) = X^{(k)}(i_k) X^{(k+1)}(i_{k+1}), \quad (8)$$

and make the minimization over w_k. Classical DMRG minimizes the Rayleigh quotient, our version minimizes the energy function $J(x)$, see [18, 9]. Similarly to (6),(7) we write the local DMRG problem $B w_k = g_k$ as follows

$$\begin{gathered} \mathcal{P} = \mathcal{P}_{\notin\{k,k+1\}}(\bar{X}) = X^{<k} \otimes I_{n_k} \otimes I_{n_{k+1}} \otimes \left(X^{>k+1}\right)^\top \in \mathbb{C}^{n_1 \dots n_d \times r_{k-1} n_k n_{k+1} r_{k+1}}, \\ B = \mathcal{P}^* A \mathcal{P}, \qquad g_k = \mathcal{P}^* y \in \mathbb{C}^{r_{k-1} n_k n_{k+1} r_{k+1}}. \end{gathered} \quad (9)$$

[2] We will always assume that $n_1 = \dots = n_d = n$ and $r_1 = \dots = r_{d-1} = r$ in the complexity estimates.

When the w_k is computed, new TT–blocks are obtained by the low–rank decomposition, i.e. the right-hand side of (8) is computed and the k-th rank is updated adaptively to the chosen accuracy. The minimization over $\mathcal{O}(n^2r^2)$ components of w_k leads to complexity $\mathcal{O}(n^3)$, and seriously increases the computational time for systems with large mode sizes.

3.3 One–block enrichment as a Galerkin reduction of the two-dimensional system

Suppose that we have just solved (7) and updated the TT-block $X^{(k)}$. Before we move to the next step, we would like to improve the reduced basis $\mathcal{P}_{\neq k+1}(\bar{X})$ by adding a few vectors to it. Denote the current solution vector by $t = \tau(\bar{T})$ and suppose we add a step $s = \tau(\bar{S})$. Then the updated solution $x = t+s$ has the TT–representation $x = \tau(\bar{X})$ defined as follows

$$\begin{aligned} X^{(1)}(i_1) &:= \begin{bmatrix} T^{(1)}(i_1) & S^{(1)}(i_1) \end{bmatrix}, \\ X^{(p)}(i_p) &:= \begin{bmatrix} T^{(p)}(i_p) & 0 \\ 0 & S^{(p)}(i_p) \end{bmatrix}, \quad X^{(d)}(i_d) := \begin{bmatrix} T^{(d)}(i_d) \\ S^{(d)}(i_d) \end{bmatrix}, \end{aligned} \tag{10}$$

where $p = 2, \dots, d-1$. We will denote this tensor train as $\bar{X} = \bar{T} + \bar{S}$.[3] The considered update affects the solution process in two ways: first, naturally, adds a certain correction to the solution, and second, enlarge the reduced basis that we will use at the *next step* of the ALS minimization. Indeed, it can easily be seen from definition (2) that

$$X^{<k} = \begin{bmatrix} T^{<k} & S^{<k} \end{bmatrix}, \qquad \left(X^{>k}\right)^\top = \begin{bmatrix} \left(T^{>k}\right)^\top & \left(S^{>k}\right)^\top \end{bmatrix}.$$

From (6) we conclude that $\mathcal{P}_{\neq k}(\bar{T} + \bar{S}) = \begin{bmatrix} T^{<k} & S^{<k} \end{bmatrix} \otimes I \otimes \begin{bmatrix} \left(T^{>k}\right)^\top & \left(S^{>k}\right)^\top \end{bmatrix}$ and hence

$$\mathcal{X}_{\neq k} = \mathcal{P}_{\neq k}(\bar{T} + \bar{S}) = \begin{bmatrix} \mathcal{P}_{\neq k}(\bar{T}) & S^{<k} \otimes I \otimes \left(T^{>k}\right)^\top & T^{<k} \otimes I \otimes \left(S^{>k}\right)^\top & \mathcal{P}_{\neq k}(\bar{S}) \end{bmatrix}. \tag{11}$$

The clever choice of $s = \tau(\bar{S})$ allows to add the essential vectors to span $\mathcal{P}_{\neq k}(\bar{T} + \bar{S})$ and therefore improve the convergence of ALS.

A random choice of $s \in \mathcal{T}_{\mathbf{r}}$ with some small TT–ranks $\mathbf{r}$ (cf. *random kick* proposed in [37, 29]) may lead to a slow convergence. It also introduces an unwanted perturbation of the solution. A more robust idea is to choose s in accordance to some one-step iterative method, for instance, take $s \approx z = y - At$ and construct a steepest descent or minimal residual method with approximations. This choice allows to derive the convergence estimate similarly to the classical one and will be discussed in Sec. 4.

To stay within methods of linear complexity, we restrict ourselves to zero shifts $s = 0$ with a simple TT–structure $\bar{S} = (0, \dots, 0, S^{(k)}, 0, \dots, 0)$. The tensor train $\bar{X} = \bar{T} + \bar{S}$ has the following structure[4]

$$X^{(k)}(i_k) = \begin{bmatrix} T^{(k)}(i_k) & S^{(k)}(i_k) \end{bmatrix}, \qquad X^{(k+1)}(i_{k+1}) = \begin{bmatrix} T^{(k+1)}(i_{k+1}) \\ 0 \end{bmatrix}, \tag{12}$$

[3] Due to the non–uniqueness of the TT–format other representations (probably with smaller TT–ranks) can exist for $x = t + s$.

[4] We give a description for the *forward* sweep, i.e. the one with increasing $k = 1, \dots, d$. For the *backward* sweep the construction is done analogously.

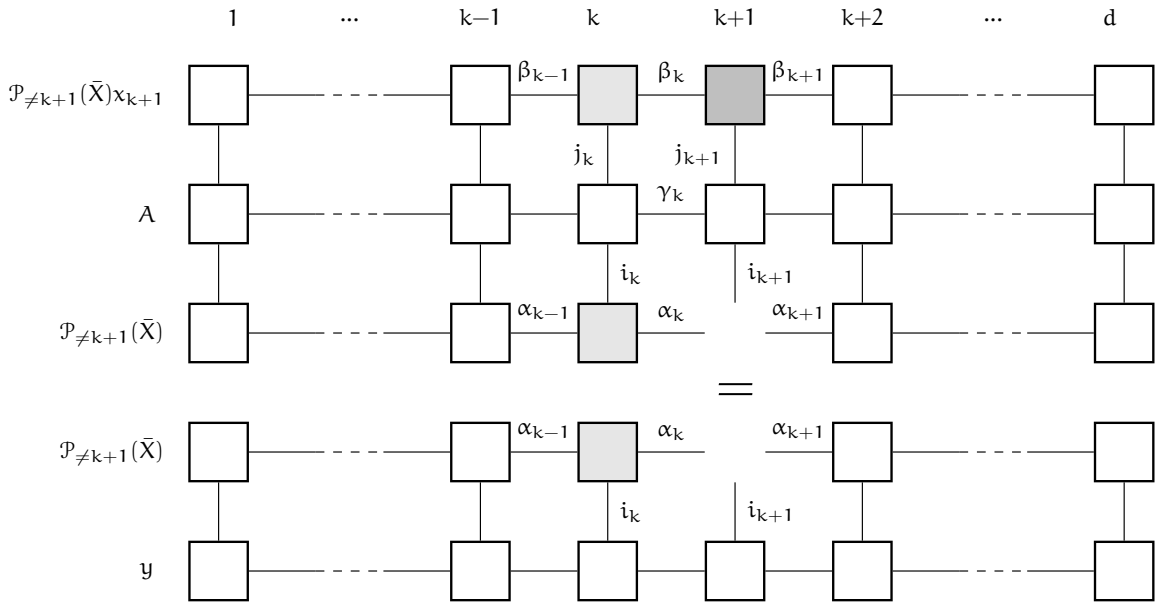

Figure 2: Linear system $Ax = y$ in the reduced basis $\mathcal{P}_{\neq k+1}(\bar{X})$ shown by tensor networks. The reduced system has $r_k n_{k+1} r_{k+1}$ unknowns, shown by the dark box. Gray boxes show the $X^{(k)}$ which is updated by $S^{(k)}$ to improve the convergence. White boxes contribute to the local matrix B and right-hand side g of the 2D system (9).

and $X^{(p)} = T^{(p)}$ for other p. Note that since $s = 0$, the enrichment step does not affect the energy $J(\tau(\bar{X})) = J(\tau(\bar{T}))$. Therefore, we can choose $S^{(k)}$ freely and develop (probably, heuristic) approaches to improve the convergence of our scheme. The reduced basis $\mathcal{X}_{\neq k+1} = \mathcal{P}_{\neq k+1}(\bar{T} + \bar{S})$ depends on the choice of $S^{(k)}$ as follows (cf. (6))

$$\begin{aligned}
\mathcal{X}_{\neq k+1}(\overline{i_1 \dots i_d}, \overline{\alpha_k j_{k+1} \alpha_{k+1}}) &= X^{<k}(\overline{i_1 \dots i_{k-1}}, \alpha_{k-1}) X^{(k)}_{\alpha_{k-1},\alpha_k}(i_k)\delta(i_{k+1}, j_{k+1}) X^{>k+1}(\alpha_{k+1}, \overline{i_{k+2} \dots i_d}),\\
\mathcal{X}_{\neq k+1} &= X^{<k}_{\alpha_{k-1}} \otimes X^{(k)}_{\alpha_{k-1}} \otimes I_{n_{k+1}} \otimes \left(X^{>k+1}\right)^\top,\\
X^{(k)}_{\alpha_{k-1}} &= \begin{bmatrix} T^{(k)}_{\alpha_{k-1}} & S^{(k)}_{\alpha_{k-1}} \end{bmatrix},
\end{aligned} \tag{13}$$

where $X^{<k}_{\alpha_{k-1}}$ is a column of $X^{<k}$ and $X^{(k)}_{\alpha_{k-1}}$ is the $n_k \times r_k$ matrix which is the *slice* of 3-tensor $X^{(k)} = [X^{(k)}(\alpha_{k-1}, i_k, \alpha_k)]$ corresponding to the fixed α_{k-1}, and similarly for $S^k(\alpha_{k-1})$ and $T^{(k)}(\alpha_{k-1})$. Below we will write the local system (7) at the step $k + 1$ and see how it is affected by the choice of $S^{(k)}$.

The two-dimensional system defined by (9) is shown by gray boxes in the Fig. 2. It appears here as the local problem in the DMRG method, but in the same framework we may consider the whole initial system with $d = 2$, and $k = 1$, depending on what type of analysis we would like to perform.

Now the reduced system for the elements of $x_{k+1}(\overline{\beta_k j_{k+1} \beta_{k+1}}) = X^{(k+1)}(\beta_k, j_{k+1}, \beta_{k+1})$ writes

$$\begin{aligned}
X^{(k)}_{\alpha_k,a} B_{ab,a'b'} X^{(k)}_{a',\beta_k} X^{(k+1)}_{\beta_k,b'} &= X^{(k)}_{\alpha_k,a} G_{a,b},\\
\left[X^{(k)} \otimes I\right]^* B \left[X^{(k)} \otimes I\right] x_{k+1} &= \left[X^{(k)} \otimes I\right]^* g,
\end{aligned} \tag{14}$$

where the following multi-indices are introduced for brevity of notation

$$\begin{aligned}&\overline{\alpha_{k-1}i_k} = a, \quad \overline{\beta_{k-1}j_k} = a', \quad a, a' = 1, \ldots, r_{k-1}n_k,\\ &\overline{i_{k+1}\alpha_{k+1}} = b, \quad \overline{j_{k+1}\beta_{k+1}} = b', \quad b, b' = 1, \ldots, r_{k+1}n_{k+1},\end{aligned}$$

and $X^{(k)} \in \mathbb{C}^{r_{k-1}n_k \times r_k}$, $I = I_{r_{k+1}n_{k+1}}$. The system (14) has $r_k n_{k+1} r_{k+1}$ unknowns. At the same time it is the reduction of a 2D system $Bw = g$ which has $r_{k-1}n_k n_{k+1} r_{k+1}$ unknowns. Therefore, the choice of the enrichment $S^{(k)}$ (as a part of $X^{(k)}$) can be considered as a cheaper approximation of the 2D system solution. Taking into account the structure of $X^{(k)}$ from (13) we rewrite (14) as follows

$$\begin{bmatrix} T & S \end{bmatrix}^* B \begin{bmatrix} T & S \end{bmatrix} x_{k+1} = \begin{bmatrix} T & S \end{bmatrix}^* g, \qquad T = T^{(k)} \otimes I, \quad S = S^{(k)} \otimes I. \tag{15}$$

The system (15) is difficult to analyze. However, we may propose a certain approximation to its solution, and estimate the quality of the solution to the whole system (15) via the properties of the approximation. Namely, let us consider the zero–padded TT–core $X^{(k+1)}$ in (12) as the *initial guess*, i.e., some information about the solution x_{k+1} that we want to use. For instance, we can apply the block Gauss–Seidel step, restricting the unknown block to the form

$$X^{(k+1)}(i_{k+1}) = \begin{bmatrix} T^{(k+1)}(i_{k+1}) \\ V(i_{k+1}) \end{bmatrix}, \qquad \begin{aligned} t(\overline{\alpha'_k i_{k+1} \alpha_{k+1}}) &= T^{(k+1)}_{\alpha'_k \alpha_{k+1}}(i_{k+1}),\\ v(\overline{\alpha''_k i_{k+1} \alpha_{k+1}}) &= V_{\alpha''_k \alpha_{k+1}}(i_{k+1}).\end{aligned} \tag{16}$$

Then (15) writes as the following overdetermined system

$$\begin{bmatrix} T^* \\ S^* \end{bmatrix} B \begin{bmatrix} Tt + Sv \end{bmatrix} = \begin{bmatrix} T^* \\ S^* \end{bmatrix} g,$$

and following the Gauss–Seidel step we solve it considering only the lower part

$$(S^* B S) v = S^* (g - B T t). \tag{17}$$

Equation (17) is a Galerkin reduction method with the basis S applied to the system $Bw = g$ with the initial guess (8), and TT–cores $X^{(k)}$ and $X^{(k+1)}$ defined by (12). After (17) is solved, the updated superblock $W^{(k)}_{\text{new}}$ writes as follows

$$\begin{aligned} X^{(k)}(i_k) &= \begin{bmatrix} T^{(k)}(i_k) & S^{(k)}(i_k) \end{bmatrix}, \qquad X^{(k+1)}_{\text{new}}(i_{k+1}) = \begin{bmatrix} T^{(k+1)}(i_{k+1}) \\ V(i_{k+1}) \end{bmatrix},\\ W^{(k)}_{\text{new}}(\overline{i_k i_{k+1}}) &= T^{(k)}(i_k) T^{(k+1)}(i_{k+1}) + S^{(k)}(i_k) V(i_{k+1})\\ &= W^{(k)}(\overline{i_k i_{k+1}}) + S^{(k)}(i_k) V(i_{k+1}),\end{aligned} \tag{18}$$

which allows to consider the proposed method as a solver for the 2D system, which performs the *low–rank correction* for the superblock rather than recompute it from scratch.

Equations (14) and (17) can be considered as certain approximate approaches to the solution of the 2D system (9). Different such approaches can be collected into Table 1, sorted from the highest to the lowest accuracy.

Method	$X^{(k)}$		$X^{(k+1)}$		Complexity
	$T^{(k)}$	$S^{(k)}$	$T^{(k+1)}$	V	
DMRG (9)	optimize		optimize		$\mathcal{O}(r^3 n^3)$
AMEn (14)	keep	choose	optimize		$\mathcal{O}(r^2 n)$
DMRG correction	keep	optimize	keep	optimize	$\mathcal{O}(\rho^3 r n)$
Galerkin correction (17)	keep	choose	keep	optimize	$\mathcal{O}(\rho r n)$

Table 1: Comparison of different solution methods for a two–dimensional system (9) with blocks given by (18). We may *keep* the block from the previous iteration, *choose* it arbitrary (eg., using quasi–optimal or heuristic choice) or *optimize* solving the reduced system. In the complexity estimates, r is typical rank of $\bar{X}$ and ρ is typical rank of $\bar{S}$.

4 Steepest descent schemes

4.1 Steepest descent with perturbation

Given the initial guess t, the *steepest descent* (SD) step minimizes the energy function (4) over vectors $x = t + s\alpha$, where the step is chosen as follows

$$s = -\operatorname{grad} J(t) = y - At = z,$$
$$\alpha = \arg\min J(t + z\alpha) = \frac{(z,z)}{(z,Az)}.$$

The solution after the SD step satisfies the so-called *Galerkin condition* $(z, y - Ax) = 0$. The progress of the SD step can be analyzed in terms of A–norms of errors $c = x_* - t$ and $d = x_* - x$ as follows

$$x = t + z\frac{\|z\|^2}{\|z\|_A^2}, \qquad d = c - z\frac{\|z\|^2}{\|z\|_A^2} = (I - R_z)c.$$

This gives interpretation in terms of projections and proves the monotone decrease of the energy function $J_A(x) = \|d\|_A^2 \leqslant \|c\|_A^2 = J_A(t)$. To estimate the convergence rate, we write

$$\|d\|_A^2 = (c, (I - R_z)^* A (I - R_z)c) = (c, A(I - R_z)c) = \omega_z^2 \|c\|_A^2, \quad \omega_z^2 = \frac{(c, (I - R_z)c)_A}{(c,c)_A}.$$

The convergence rate ω_z is therefore a square root of the Rayleigh quotient for $I - R_z$ in the A–scalar product. It can be bounded using the Kantorovich inequality [20] as follows

$$\omega_z^2 = 1 - \frac{(z,z)}{(z,Az)}\frac{(z,z)}{(z,A^{-1}z)} \leqslant \left(\frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}}\right)^2, \tag{19}$$

where $\lambda_{\max}$ and $\lambda_{\min}$ denote the largest and smallest eigenvalues of A, respectively.

The residual $z = y - At$ of the steepest descent method can not be computed exactly for high–dimensional problems. Suppose that it is approximated by $\tilde{z}$ and the perturbed SD step is applied as follows

$$x = t + \tilde{z}\frac{\|\tilde{z}\|^2}{\|\tilde{z}\|_A^2}, \qquad \tilde{d} = c - \tilde{z}\frac{\|\tilde{z}\|^2}{\|\tilde{z}\|_A^2} = (I - R_{\tilde{z}})c + R_{\tilde{z}}(c - \tilde{c}), \tag{20}$$

where $A\tilde{c} = \tilde{z}$. We further restrict ourselves to the perturbations of the following form

$$z = \tilde{z} + \delta z, \quad (\tilde{z}, \delta z) = 0, \qquad \|\delta z\|_A \leqslant \varepsilon \|\tilde{z}\|_A \leqslant \varepsilon \|z\|_A, \tag{21}$$

which will appear naturally in our algorithms for higher dimensions. For such perturbations the second term vanishes, $R_{\tilde{z}}(c - \tilde{c}) = 0$, and the perturbation of the SD step writes through the perturbation of A–orthogonal projectors as follows

$$\tilde{d} - d = -(R_{\tilde{z}} - R_z)\, c.$$

A comprehensive overview of the perturbation theory for projections, pseudo–inverses and least square problems can be found in [39]. Rather than adapting their results to the case of A–orthogonal projectors, we will develop a more accurate estimate for $\tilde{d} - d$ using specifically the perturbations (21).

Theorem 1. For $\tilde{z}$ given by (21) the progress of the perturbed SD step (20) writes as follows

$$\|\tilde{d}\|_A \leqslant \omega_{\tilde{z}} \|c\|_A, \qquad \omega_{\tilde{z}} = \omega_z + \varepsilon\sqrt{2(1-\omega_z^2)} + \frac{1}{2\sqrt{2}}\varepsilon^3 \operatorname{cond}^2(A),$$

where ω_z is the progress of the unperturbed SD step given by (19).

Proof. For $z = \tilde{z} + \delta z$ the following simple identity can be verified from definition

$$R_z - R_{\tilde{z}} = \frac{\tilde{z}\,\delta z^*}{\|\tilde{z}\|_A^2}(I - R_z^*)A + (I - R_{\tilde{z}})\frac{\delta z\, z^*}{\|z\|_A^2}A.$$

The perturbation of the SD step $\tilde{d} - d = (R_z - R_{\tilde{z}})c$ writes

$$\tilde{d} - d = \tilde{p}\frac{\|z\|^2}{\|z\|_A^2} + \tilde{z}\frac{(p, z)}{\|\tilde{z}\|_A^2},$$

where $p = (I - R_z)\delta z$ and $\tilde{p} = (I - R_{\tilde{z}})\delta z$. Obviously, $\|\tilde{p}\|_A \leqslant \|\delta z\|_A \leqslant \varepsilon \|z\|_A$. To estimate the A–norm of the second term, we write

$$\begin{aligned}(z, p) = z^*(I - R_z)\delta z = z^*\delta z - \frac{z^* z z^* A \delta z}{\|z\|_A^2} &= \|\delta z\|^2 - \frac{\|z\|^2}{\|z\|_A^2}(z, \delta z)_A \\ &= (A^{-1}\delta z - \gamma z, \delta z)_A,\end{aligned}$$

where $\gamma = \|z\|^2/\|z\|_A^2$. Then $|(p, z)| \leqslant \|A^{-1}\delta z - \gamma z\|_A \|\delta z\|_A$ and

$$\begin{aligned}\|A^{-1}\delta z - \gamma z\|_A^2 &= \|A^{-1}\delta z\|_A^2 - 2\gamma(\delta z, z) + \gamma^2\|z\|_A^2 = \gamma^2\|z\|_A^2 + \|\delta z\|_{A^{-1}}^2 - 2\gamma\|\delta z\|^2 \\ &\leqslant \gamma^2\|z\|_A^2 + \|\delta z\|_{A^{-1}}^2.\end{aligned}$$

Since $\tilde{p}$ and $\tilde{z}$ are A–orthogonal, we write

$$\|\tilde{d} - d\|_A^2 = \|\tilde{p}\|_A^2\gamma^2 + \frac{|(p, z)|^2}{\|\tilde{z}\|_A^2} \leqslant \varepsilon^2\|z\|_A^2\gamma^2 + \varepsilon^2(\gamma^2\|z\|_A^2 + \|\delta z\|_{A^{-1}}^2) = \varepsilon^2\left(2\frac{\|z\|^4}{\|z\|_A^2} + \|\delta z\|_{A^{-1}}^2\right).$$

Finally, we estimate

$$\frac{\|\tilde{d} - d\|_A}{\|c\|_A} \leqslant \varepsilon\sqrt{2}\frac{\|z\|^2}{\|z\|_A\|c\|_A} + \varepsilon\frac{\|\delta z\|_{A^{-1}}^2\|z\|_A}{2\sqrt{2}\|z\|^2\|z\|_{A^{-1}}} \leqslant \varepsilon\sqrt{2(1-\omega_z^2)} + \varepsilon^3\frac{\operatorname{cond}^2(A)}{2\sqrt{2}},$$

where the last inequality is based on

$$\|u\|_{A^{-1}} \leqslant \lambda_{\min}^{-1/2}\|u\| \leqslant \lambda_{\min}^{-1}\|u\|_A, \qquad \|u\|_{A^{-1}} \geqslant \lambda_{\max}^{-1/2}\|u\| \geqslant \lambda_{\max}^{-1}\|u\|_A, \qquad \mathrm{cond}(A) = \frac{\lambda_{\max}}{\lambda_{\min}}.$$

Since $\|\tilde{d}\|_A \leqslant \|\tilde{d} - d\|_A + \|d\|_A$, we obtain the statement of the theorem. □

Remark 1. If $\omega_z < 1$ there exists $\varepsilon_* > 0$ such that for all $0 < \varepsilon < \varepsilon_*$ it holds $\omega_{\tilde{z}} < 1$. This critical value $\varepsilon_*(\kappa, \omega)$ is the real positive root of the cubic equation $\omega_{\tilde{z}}(\varepsilon) = 1$, where $\kappa = \mathrm{cond}(A)$ and ω act as parameters. The minimal value of $\varepsilon_*(\kappa, \omega)$ for $\omega \leqslant (\kappa-1)/(\kappa+1)$ and $\kappa \to \infty$ behaves as $\varepsilon_* = \kappa^{-1} + \mathcal{O}(\kappa^{-3/2})$.

4.2 Steepest descent in two dimensions

Consider the two–dimensional linear system $Ax = y$ written in the elementwise notation as follows[5]

$$A(\overline{i_1 i_2}, \overline{j_1 j_2})x(\overline{j_1 j_2}) = y(\overline{i_1 i_2}), \qquad i_1, j_1 = 1, \ldots, n_1, \quad i_2, j_2 = 1, \ldots, n_2.$$

As previously, we assume A and y to be given, and x to be sought in the following low-rank decomposition format

$$\begin{aligned} A(\overline{i_1 i_2}, \overline{j_1 j_2}) &= A_\gamma^{(1)}(i_1, j_1) A_\gamma^{(2)}(i_2, j_2), \quad A^{(p)} = [A_\gamma^{(p)}(i_p, j_p)] \in \mathbb{C}^{n_p \times n_p \times r_A}, \\ y(\overline{i_1 i_2}) &= y_\beta^{(1)}(i_1) y_\beta^{(2)}(i_2), \quad Y^{(p)} = [y_\beta^{(p)}(i_p)] \in \mathbb{C}^{n_p \times r_y}, \\ x(\overline{j_1 j_2}) &= x_\alpha^{(1)}(j_1) x_\alpha^{(2)}(j_2), \quad X^{(p)} = [x_\alpha^{(p)}(j_p)] \in \mathbb{C}^{n_p \times r_x}, \end{aligned} \tag{22}$$

where $p = 1, 2$. Given the initial guess t in the same format, we compute the low–rank approximation of the residual $\tilde{z} \approx z = y - At$ as follows

$$\tilde{z}(\overline{i_1 i_2}) = z_\zeta^{(1)}(i_1) z_\zeta^{(2)}(i_2), \qquad Z^{(1)} = [z_\zeta^{(1)}(i_1)] \in \mathbb{C}^{n_1 \times r_z}, \quad Z^{(2)} = [z_\zeta^{(2)}(i_2)] \in \mathbb{C}^{n_2 \times r_z}.$$

Following the perturbed SD algorithm, we can write the updated solution $x = t + \tilde{z}\alpha$ in a form

$$x(\overline{j_1 j_2}) = \begin{bmatrix} T^{(1)}(j_1) & Z^{(1)}(j_1) \end{bmatrix} \begin{bmatrix} T^{(2)}(j_2) \\ Z^{(2)}(j_2)\alpha \end{bmatrix}, \tag{23}$$

and optimize by the step size α. Recalling the considerations from Section 3.3, we can consider more efficient optimization steps listed in Table 1. For example, the solution of DMRG system (9) corresponds to the exact solution of the considered 2D system. We will particularly consider the Galerkin correction framework, i.e., will optimize over the bottom block of $X^{(2)}$, denoted as V in (18). This is the cheapest method in Table 1, and all other methods have better convergence properties.

In the proposed method we choose the step $x = t + Z\nu$ where

$$Z = Z^{(1)} \otimes I_{n_2} \in \mathbb{C}^{n_1 n_2 \times r_\zeta n_2}, \quad \nu(\overline{\zeta i_2}) = V_\zeta(i_2),$$

[5] We consider x and y as vectors and at the same time as two-dimensional arrays $x = [x(j_1, j_2)]$ and $y = [y(i_1, i_2)]$ with the same entries. We will switch freely between these representations without change of a notation.

and without the loss of generality assume the orthogonality of Z. Minimization of the energy function $J(x)$ over v leads to the set of Galerkin conditions $Z^*(y - Ax) = 0$ and the step writes as follows

$$x = t + Zv, \qquad (Z^*AZ)v = Z^*\tilde{z}. \tag{24}$$

Note that if we restrict ourselves to the perturbations $z = \tilde{z} + \delta z$ such that $Z^*\delta z = 0$, it holds

$$v = (Z^*AZ)^{-1}Z^*\tilde{z} = (Z^*AZ)^{-1}Z^*z.$$

Then the accuracy of the proposed method can be estimated similarly to the standard SD step $d = c - Zv = (I - R_Z)c$,and the progress of this step writes

$$\|d\|_A^2 = \omega_Z^2\|c\|_A^2, \qquad \omega_Z^2 = \frac{(c, (I - R_Z)c)_A}{(c, c)_A}. \tag{25}$$

Since $\tilde{z} \in \operatorname{span} Z$ it follows that $\omega_Z \leqslant \omega_{\tilde{z}}$, i.e., the convergence of the proposed method (24) is not slower than the one of the perturbed SD step (20) estimated in Thm. 1.

Remark 2. When $\operatorname{span} Z = \mathbb{C}^{n_1 n_2}$ we converge in one iteration, i.e. $\omega_Z = 0$. For large Z s.t. $z \in \operatorname{span} Z$ we can expect $\omega_Z \ll \omega_z$. In general, however, the inequality $\omega_Z \leqslant \omega_z$ is sharp. To show this, consider $Z = \begin{bmatrix} z & s \end{bmatrix}$ with $(z, s) = 0$. It is easy to show that

$$1 - \omega_Z^2 = \frac{\|z\|^4\|s\|_A^2}{\|z\|_{A^{-1}}^2\left(\|s\|_A^2\|z\|_A^2 - |(s,z)_A|^2\right)}, \qquad \frac{1 - \omega_z^2}{1 - \omega_Z^2} = 1 - \frac{|(s,z)_A|^2}{\|s\|_A^2\|z\|_A^2} \leqslant 1,$$

which proves $\omega_Z \leqslant \omega_z$. However, the ratio can be equal to one when $(s, z)_A = 0$ and $(s, z) = 0$ simultaneously. It can happen, eg. if s is an eigenvector of A. Similarly, if there is a k–dimensional invariant subspace of A which is orthogonal to z, we can form $Z = \begin{bmatrix} z & s_1 \dots s_k \end{bmatrix}$ from the basis vectors of this subspace and have the same convergence $\omega_z = \omega_Z$ as the SD step does.

To find the correction term v we have to solve the reduced linear system size $r_z n_2$, which writes as follows

$$Bv = g, \qquad B = Z^*AZ, \qquad g = Z^*z. \tag{26}$$

Suppose that n_2 is still too large for the system to be solved exactly and we find the approximate solution $v \approx v_* = B^{-1}g$. The simplest idea is to solve the reduced problem by the standard SD method. The following theorem estimates the progress of such 'lazy' approach.

Theorem 2. Consider the system $Ax = y$ with the initial guess t and error $c = x_* - t$. After one outer step of SD (24) and one inner step of SD applied to the reduced problem (26), the error $d = x_* - x$ writes as follows

$$\begin{aligned} d &= \left((I - R_Z) + Z(I - Q_g)Z^*R_Z\right)c, \\ &= \left((I - R_Z) + (I - R_{\tilde{z}})R_Z\right)c, \\ \|d\|_A^2 &= \left(\omega_Z^2 + (1 - \omega_Z)^2\omega_g^2\right)\|c\|_A^2, \end{aligned} \tag{27}$$

where Q_g is the B–orthogonal projector on g, and $\omega_g \leqslant \omega_{\tilde{z}}$.

Proof. If ν is the obtained (approximate) solution of (26), the progress of the step (24) is

$$\begin{aligned} d = x_* - x = c - Z\nu = (I - R_Z)c + Z(\nu_* - \nu), \\ \|d\|_A^2 = \|x_* - x\|_A^2 = \|(I - R_Z)c\|_A^2 + \|Z(\nu_* - \nu)\|_A^2 = \omega_Z^2\|c\|_A^2 + \|\nu_* - \nu\|_B^2, \end{aligned} \tag{28}$$

where in the last line we use the A–orthogonality of the two terms. The initial guess for ν is zero, and after one step of the SD applied to (26) the error is

$$\nu_* - \nu = (I - Q_g)(\nu_* - 0) = (I - Q_g)B^{-1}g = (I - Q_g)(Z^*AZ)^{-1}Z^*z.$$

The first line of the theorem now follows by the definition of R_Z. To prove the second line it is enough to note that

$$ZQ_gZ^* = \frac{Zgg^*Z^*AZZ^*}{g^*Z^*AZg} = \frac{\tilde{z}\tilde{z}^*AZZ^*}{\tilde{z}^*A\tilde{z}} = R_{\tilde{z}}ZZ^*,$$

and $ZZ^*R_Z = R_Z$. The progress of the inner SD step is $\|\nu_* - \nu\|_B = \omega_g\|\nu_*\|_B$, where

$$\|\nu_*\|_B^2 = \|B^{-1}g\|_B^2 = \|Z^*\tilde{z}\|_{B^{-1}}^2 = \|Z^*z\|_{B^{-1}}^2 = (z, Z(Z^*AZ)^{-1}Z^*z) = (c, R_Zc)_A = (1-\omega_Z^2)\|c\|_A^2.$$

Substituting these estimates to (28) we obtain the second claim of the theorem.

Now we prove that $\omega_g \leqslant \omega_{\tilde{z}}$. Similarly to (19) we have

$$\omega_g^2 = \frac{(\nu_*, (I - Q_g)\nu_*)_B}{(\nu_*, \nu_*)_B} = 1 - \frac{\|g\|^4}{(g, Bg)(g, B^{-1}g)}.$$

Since Z is orthogonal, $\|g\| = \|Z^*z\| = \|\tilde{z}\|$. It also holds that $(g, Bg) = (Zg, AZg) = (\tilde{z}, A\tilde{z})$. Finally we show that

$$(g, B^{-1}g) = (\tilde{z}, Z(Z^*AZ)^{-1}Z^*\tilde{z}) = (A^{-1}\tilde{z}, R_zA^{-1}\tilde{z})_A \leqslant \|A^{-1}\tilde{z}\|_A^2 = (\tilde{z}, A^{-1}\tilde{z}),$$

which completes the proof. □

The second term of (28) can be written also as follows

$$\begin{aligned} Z(\nu_* - \nu) = Z(I - Q_g)\nu_* = Z\left(I - \frac{gg^*B}{g^*Bg}\right)B^{-1}g = ZB^{-1}Z^*\tilde{z} - \frac{\|g\|^2}{\|g\|_B^2}\tilde{z} \\ = ZB^{-1}Z^*z - \frac{\|\tilde{z}\|^2}{\|\tilde{z}\|_A^2}z = R_Zc - R_{\tilde{z}}c, \end{aligned}$$

which gives $d = (I - R_{\tilde{z}})c$. This shows that the combination of one outer and one inner SD step is equivalent to the SD step with perturbation (20). This is also easily seen from the structure of our inner–outer method itself. Indeed, in the outer step we add components $Z^{(1)}$ to the basis set and in the inner step we add components of the inner residual $g = Z^*\tilde{z} = z_2$, where z_2 contains the elements of $Z^{(2)}$ stretched into one vector. Therefore, the described inner–outer scheme is equivalent to one 'global' SD step.

The idea behind Theorem 2 is of course not to prove a slightly worse estimate in a more complicated way. In the recursive algorithm the second term in (28) will be obtained by the SD step followed by further optimization which will decrease the error of the reduced problem and consequently the total error. The SD step is therefore required as an initial guess for which we can provide a theoretical estimate of convergence. The practical convergence that we expect is of course better than the upper estimate in (27).

Algorithm 1 $x = t + \mathrm{ALS}(z)$

Require: System $Ax = y$ and initial guess t in the TT–format (1), approximate residual $\tilde{z} = \tau(\bar{Z}) \in \mathcal{T}_{\mathbf{r}}$.

Ensure: Updated solution $x = t + \nu$, $\nu = \tau(\bar{V}) \in \mathcal{T}_{\mathbf{r}}$.

1: **for** $k = d, \ldots, 1$ **do** {Cycle over TT–cores}

2: Find $V^{(k)} = \arg\min_{Z^{(k)}} J(t + \tau(Z^{(1)}, \ldots, Z^{(k-1)}, Z^{(k)}, V^{(k+1)}, \ldots, V^{(d)}))$

3: **end for**

4: **return** $\nu = \tau(V^{(1)}, \ldots, V^{(d)})$

Remark 3. Regarding the spectrum of reduced problems, the following two–side inequality is proved in [28]

$$(U^* A U)^{-1} \leqslant U^* A^{-1} U \leqslant \frac{(\lambda_{\min} + \lambda_{\max})^2}{4\lambda_{\min}\lambda_{\max}} (U^* A U)^{-1},$$

where U is unitary matrix and $B \geqslant C$ means that $B - C$ is positive definite. The last inequality used in Theorem 2 follows from the left part of this inequality (which is itself rather elementary).

4.3 Greedy descent method

In higher dimensions we can further improve the steepest descent step by an ALS cycle over the step vector, as shown by Alg. 1. This algorithm searches for $\max_{s \in \mathcal{T}_{\mathbf{r}}} J(t + s)$ using the ALS optimization and therefore can be considered as a *greedy* algorithm. The application of greedy algorithms to optimization in tensor formats was rigorously studied in [12, 2, 26].

Alg. 1 starts from the SD step with perturbation, and then the energy function is additionally improved by an alternative minimization cycle. The combined progress is therefore not worse than the one of the SD step, $\|d\|_A \leqslant \omega_{\tilde{z}} \|c\|_A$, given by Thm. 1. Another estimate is proven by the following theorem.

Theorem 3. Consider the system $Ax = y$ with the initial guess t and error $c = x_* - t$. The step described by Alg. 1 returns the solution $x = t + \nu$ such that the error $d = x_* - x$ is bounded as follows

$$\begin{aligned} \|d\|_A^2 &\leqslant \nu_1^2 \Big(\omega_1^2 + (1 - \omega_1^2)\nu_2^2 \Big(\omega_2^2 + (1 - \omega_2^2)\nu_3^2 \big(\omega_3^2 + \ldots + \nu_{d-1}^2 \omega_{d-1}^2 \big) \ldots \Big) \Big) \|c\|_A^2 \\ &= \left(\sum_{k=1}^{d-1} \omega_k^2 \prod_{j=1}^{k-1} (1 - \omega_j^2) \prod_{j=1}^{k} \nu_j^2 \right) \|c\|_A^2, \qquad (29) \\ \omega_k^2 &= \omega_{\tilde{Z}_{\leqslant k}}^2 = 1 - \frac{(c, R_{Z_{\leqslant k}} c)_A}{(c, c)_A}, \qquad \nu_k \leqslant 1. \end{aligned}$$

Proof. In 2D the statement of the theorem reads $\|d\|_A^2 \leqslant \nu_1^2 \omega_1^2 \|c\|_A^2$. It is easy to see that the ALS update over $Z^{(2)}$ gives exactly the two–dimensional SD step (24) with the progress $\omega_Z = \omega_{\tilde{Z}_1} = \omega_1$ given by (25). The ALS update over $Z^{(1)}$ further improves the energy function by the factor $\nu_1^2 \leqslant 1$, which proves the statement of the theorem for $d = 2$. The base of the recursion is proved.

After a *microstep* when $Z^{(k+1)}$ is optimized and becomes $V^{(k+1)}$, the solution writes as follows

$$x_k = t + \mathcal{Z}_{\leqslant k} v_{>k}, \qquad \mathcal{Z}_{\leqslant k} \in \mathbb{C}^{n_1 \dots n_d \times r_k n_{k+1} \dots n_d}, \quad v_{>k} \in \mathbb{C}^{r_k n_{k+1} \dots n_d},$$

where $v_{>k} = \tau(V^{(k+1)}, \dots, V^{(d)})$, i.e. $v_{>k}(\overline{\alpha_k j_{k+1} \dots j_d}) = V^{(k+1)}_{\alpha_k \alpha_{k+1}}(j_{k+1}) \dots V^{(d)}_{\alpha_{d-1}}(j_d)$, and

$$\begin{aligned} \mathcal{Z}_{\leqslant k} &= \mathcal{P}_{\leqslant k}(\bar{Z}) = Z^{\leqslant k} \otimes I_{n_{k+1}} \otimes \dots \otimes I_{n_d}, \\ \mathcal{Z}_{\leqslant k}(\overline{i_1 \dots i_d}, \overline{\alpha_k j_{k+1} \dots j_d}) &= Z^{(1)}_{\alpha_1}(i_1) Z^{(2)}_{\alpha_1 \alpha_2}(i_2) \dots Z^{(k)}_{\alpha_{k-1} \alpha_k}(i_k) \delta(i_{k+1}, j_{k+1}) \dots \delta(i_d, j_d). \end{aligned} \tag{30}$$

This equation is similar to the two–dimensional SD step (24) and allows to estimate the progress of Alg. 1 using the result of Thm. 2 recursively. Following (28), the progress can be written as follows

$$\frac{\|x_* - x_k\|_A^2}{\|x_* - t\|_A^2} = \left(\omega_k^2 + (1 - \omega_k^2) \frac{\|v_{>k,*} - v_{>k}\|_{A_k}^2}{\|v_{>k,*} - 0\|_{A_k}^2} \right), \tag{31}$$

where $A_k = \mathcal{Z}^*_{\leqslant k} A \mathcal{Z}_{\leqslant k}$, $z_k = \mathcal{Z}^*_{\leqslant k} \tilde{z}$ and $v_{>k,*}$ is the exact solution of the reduced problem $A_k v_{>k} = z_k$. Note that $z_k = \mathcal{Z}^*_{\leqslant k} \tau(Z^{(1)}, \dots, Z^{(d)}) = \tau(Z^{(k+1)}, \dots, Z^{(d)})$, so the inner SD steps will share the TT–factors of the same residual $\tilde{z}$.

To prove the recursion step, assume that the theorem holds in the dimension $d - 1$, write (31) with $k = 1$ and apply (29) for the second term as follows

$$\frac{\|v_* - v\|_B^2}{\|v_* - 0\|_B^2} \leqslant \left(\sum_{k=1}^{d-2} \hat{\omega}_k^2 \prod_{j=1}^{k-1} (1 - \hat{\omega}_j^2) \prod_{j=1}^{k} \hat{v}_j^2 \right), \quad \hat{\omega}_k^2 = 1 - \frac{(v_*, Q_{\mathcal{G}_{\leqslant k}} v_*)_B}{(v_*, v_*)_B},$$

where $B = Z^* A Z$, $g = Z^* z$, $Z = \mathcal{Z}_1 = Z^{(1)} \otimes I \otimes \dots \otimes I$, v_* is the exact solution of $Bv = g$, $Q_{\mathcal{G}_{\leqslant k}}$ is the B–orthogonal projector on $\mathcal{G}_{\leqslant k}$ and $\mathcal{G}_{\leqslant k} = \mathcal{P}_{\leqslant k}(\bar{G})$ is defined for $\tau(\bar{G}) = g$ similarly to (30). Since $Z \mathcal{G}_{\leqslant k} = \mathcal{Z}_{\leqslant k+1}$, and $\|v_*\|_B = \|c\|_A$ we have

$$(v_*, Q_{\mathcal{G}_{\leqslant k}} v_*)_B = (z, Z \mathcal{G}_{\leqslant k} (\mathcal{G}^*_{\leqslant k} B \mathcal{G}_{\leqslant k})^{-1} \mathcal{G}^*_{\leqslant k} Z^* z) = (c, R_{\mathcal{Z}_{\geqslant k+1}} c)_A,$$

and $\hat{\omega}_k = \omega_{k+1}$. Similarly $v_{k+1} = \hat{v}_k$ now defines the progress of the ALS microstep over the components of $G^{(k)} = Z^{(k+1)}$. Updating $Z^{(1)}$ by the ALS step we reduce the error by the factor v_1 and write the total progress as follows

$$\frac{\|x_* - x\|_A^2}{\|x_* - t\|_A^2} \leqslant v_1^2 \left(\omega_1^2 + (1 - \omega_1^2) \sum_{k=2}^{d} \omega_k^2 \prod_{j=2}^{k-1} (1 - \omega_j^2) \prod_{j=2}^{k} v_j^2 \right),$$

which completes the proof. □

Remark 4. Under the conditions of the theorem it holds $\|d\|_A \leqslant \omega_{d-1} \|c\|_A$. Indeed, after the first ALS microstep the solution has the form $x_{d-1} = t + \mathcal{Z}_{\leqslant d-1} v_d$, see (30). Comparing this to the steepest descent in 2D (24) we follow (25) and claim the convergence rate ω^2_{d-1} for x_{d-1} and consequently for the result of Alg. 1 due to the monotone convergence of the ALS.

Remark 5. If ALS steps occasionally give no progress, i.e. $v_k = 1$, the progress ω of Alg. 1 given by (29) satisfies

$$1 - \omega^2 = (1 - \omega_1^2) \dots (1 - \omega_{d-1}^2) = \prod_{k=1}^{d-1} (1 - \omega_k^2) \leqslant 1 - \omega_{d-1}^2.$$

Algorithm 2 $x = \mathrm{ALS}(t + z)$

1: Set $\bar{X} = (X^{(1)}, \ldots, X^{(d)}) = \bar{T} + \bar{Z}$
2: **for** $k = d, \ldots, 1$ **do** {Cycle over TT–cores}
3: Find $X^{(k)}_{\mathrm{new}} = \arg\min_{X^{(k)}} J(\tau(X^{(1)}, \ldots, X^{(k)}, X^{(k+1)}_{\mathrm{new}}, \ldots, X^{(d)}_{\mathrm{new}}))$
4: **end for**
5: **return** $x = \tau(X^{(1)}_{\mathrm{new}}, \ldots, X^{(d)}_{\mathrm{new}})$

It follows that in this case $\omega^2 \geqslant \omega^2_{d-1}$, and the convergence estimate given by the previous remark is better than the one given by the theorem. If a sensible estimates for ν_k are available, we can plug them in (29) to estimate the combined progress of the SD and ALS steps.

4.4 Non-greedy combination of the steepest descent and ALS

Alg. 1 is a greedy–type algorithm. Such algorithms are likely to have a slow convergence or stagnate at some error level. To improve the practical convergence we can apply the ALS optimization to the whole solution vector $x = t + z\alpha$, as shown by Alg. 2.

Just like Alg. 1, the non-greedy Alg. 2 starts from the steepest descent step and then improves the energy function by a number of ALS updates. Therefore, the progress of Alg. 2 is estimated by the one of the SD algorithm, $\|d\|_A \leqslant \omega_{\tilde{z}}\|c\|_A$. The better estimate of Remark 4 also applies to Alg. 2, i.e. $\|d\|_A \leqslant \omega_{d-1}\|c\|_A$. This follows from the fact that the optimization over $X^{(d)}$ gives better energy function than the optimization over the lower part of this TT–block $V^{(d)}$, performed in greedy Alg. 1. However, we cannot generalize the result of Thm. 3 for Alg. 2, since the non-greedy ALS update destroys the $\bar{T}+\bar{Z}$ structure of the interfaces. The practically observed convergence of this method is nevertheless much better than that of the greedy descent method. More rigorous analysis of the convergence of ALS schemes can probably provide much better estimates for the convergence rate of the proposed algorithm.

In the sequel we will develop a version of the algorithm which mixes the ALS and SD steps, following (14), cf. line 'AMEn' in Table 1. For this algorithm it is possible to analyze the convergence recurrently similarly to Theorem (29). The mixed AMEn version also has better convergence properties for the practical problems considered in [7].

5 Practical implementation of tensor truncations

Throughout the paper, we considered vectors, perturbed due to the tensor approximation. Now we highlight the practical features of this operation.

The TT–rounding procedure [30] performs the recursive SVD-based truncations, which reduce the TT–ranks. The truncation of the k-th unfolding writes as follows,

$$X^{\{k\}}(\overline{i_1 \ldots i_k}, \overline{i_{k+1} \ldots i_d}) = U(\overline{i_1 \ldots i_k}, \alpha)\sigma(\alpha)V^*(\alpha, \overline{i_{k+1} \ldots i_d}),$$

where matrices U and V are orthogonal. The approximation algorithm returns

$$\tilde{X}^{\{k\}} = \tilde{U}\tilde{U}^* X^{\{k\}}, \quad \delta X^{\{k\}} = (I - \tilde{U}\tilde{U}^*)X^{\{k\}},$$

where $\tilde{U}$ contains the r first (dominant) vectors of U. It follows by the construction of the TT–SVD algorithm that $(\tilde{X}^{\{k\}})^*(\delta X^{\{k\}}) = 0$, and therefore $(\tilde{x}, \delta x) = (\tau(\tilde{X}), \tau(\delta X)) = 0$. We rely on this property for the residual approximation (21) in the accuracy analysis of the perturbed steepest descent method, see Theorem 1. The block version of the same orthogonality condition is used in the derivation of the two-dimensional steepest descent progress (25).

The SVD algorithm truncates a vector in the Frobenius norm, i.e. chooses the approximation rank considering a sum of squared smallest singular values. To satisfy the accuracy assumption in (21) we need to perform the accuracy control in the A-norms, $\|\delta z\|_A \leqslant \varepsilon \|z\|_A$. An optimal approximation in the A-norms is a difficult problem. We can either truncate in the Frobenius norm and rely on the norm equivalence $\|x\|\lambda_{\min}^{1/2} \leqslant \|x\|_A \leqslant \|x\|\lambda_{\max}^{1/2}$, or follow the cheap heuristic strategy proposed in [9]. In the inner steps of the TT-rounding procedure, after the SVD is computed, we throw away the smallest singular values one by one, while the *local* error / residual is below the tolerance, i.e.

$$\begin{aligned} \|X^{(k)} - U\Sigma V^*\|_{\mathcal{P}^*_{\neq k} A \mathcal{P}_{\neq k}} &\leqslant \varepsilon \|X^{(k)}\|_{\mathcal{P}^*_{\neq k} A \mathcal{P}_{\neq k}}, \quad \text{or} \\ \|\mathcal{P}^*_{\neq k} A \mathcal{P}_{\neq k}(X^{(k)} - U\Sigma V^*)\| &\leqslant \varepsilon \|\mathcal{P}^*_{\neq k} A \mathcal{P}_{\neq k} X^{(k)}\|. \end{aligned}$$

The basis enrichment step developed in our paper can only increase the TT–ranks of the solution. To make the procedure computationally feasible, we need to introduce a truncation step, which will reduce the solution ranks. To do this, we apply the TT–rounding procedure between the iterations, which perturbs the solution and can increase the energy function. Therefore, the truncation accuracy has to be chosen accurately to provide the convergence of the methods with approximation.

Assume that a step of the proposed method has the following progress,

$$\|x_* - x\|_A \leqslant \Omega \|x_* - t\|_A.$$

The progress after the approximation $\|x - \tilde{x}\|_A \leqslant \varepsilon_x \|x\|_A$ reads

$$\|x_* - \tilde{x}\|_A = \|x_* - x + x - \tilde{x}\|_A \leqslant \Omega \|x_* - t\|_A + \varepsilon_x \|x\|_A.$$

While the energy function is large, the first term dominates for sufficiently small ε_x. In the end of the process, the perturbation error is comparable to the progress of the method, and the algorithm stagnates. We will see this in numerical examples.

6 Numerical experiments

Let us verify the methods proposed on a model example of symmetric positive definite system:

$$-\Delta x = e, \qquad x \in \Omega = [0:1]^d, \qquad x|_{\partial\Omega} = 0,$$

where Δ is the standard finite difference Laplacian discretization on a uniform grid with the mode size 64 in each direction, i.e., the linear system has 64^d unknowns. The right–hand side e is the vector of all ones. Such a system arises naturally in the heat transfer simulation, or to precondition more complex elliptic problems. Note that the matrix and the right–hand side have exact low–rank representations, see [21, 31].

For different d we compare the following methods in Fig. 3:

- the DMRG method presented in [9] ("dmrg");
- the 2D SD method (24) in a form $x = t + \mathcal{Z}_{\leqslant d-1}\nu_d$ ("$x = t + Z\nu$");
- the greedy algorithm 1 ("$x = t + \mathrm{ALS}(z)$");
- the non-greedy algorithm 2 ("$x = \mathrm{ALS}(t + z)$");
- wherever possible, the standard (vectorized) steepest descend ("sd").

The TT–rank of the enrichment $\tilde{z}$ was chosen $\rho = 5$, and the solution after each step was approximated with the relative truncation tolerance $\varepsilon_x = 10^{-4}$ in the Frobenius norm.

The convergence of the considered methods is compared in Fig. 3. A one–dimensional sweep is considered as one iteration, the progress of micro-iterations is also shown whenever possible. We can make the following remarks based on the experimental results.

- ALS steps sufficiently improves the convergence of all considered methods, i.e. the pessimistic assumptions of Remark 5 do not hold. A refined analysis of ALS convergence rates ν_k is still an open question.
- The convergence of non-greedy Alg. 2 is comparable to the one of the DMRG iteration-wise. However, the complexity of each DMRG iteration is cubic in the mode size, while the proposed methods have linear complexity. This is clearly demonstrated in the right column, where the convergence is shown w.r.t. the computational time. The proposed methods time–wise are up to 100 times faster than the DMRG for this problem.
- The one-step steepest descend method shows the slowest convergence, which is a direct consequence of the narrow (one vector) direction subspace. This indicates that the upper bounds of the convergence rate established in the paper might be seriously overestimated.

7 Conclusion and future work

In this paper we equip the ALS scheme with a basis enrichment step, which is chosen in accordance with the steepest descent algorithm. The resulted method demonstrates the convergence almost as good as the one of DMRG, while has the linear in the mode size and dimension complexity of ALS. Moreover, the global convergence rate is established similarly to the one of the steepest descent. Up to the best of our knowledge, this is the first result on the global convergence of a numerically efficient solution method for linear systems in higher dimensions. The proposed algorithm combines the advances of optimization methods in tensor formats (ALS, DMRG) with the ones of classical methods of numerical analysis.

The proposed family of methods includes the algorithm with greedy–type step, for which the theoretical results obtained in the framework of greedy algorithms can be applied. However, other algorithms developed in the non-greedy style also have proven convergence rate and manifest much better convergence in numerical experiments.

The results of this paper can be developed in the following directions. First, the analysis for the non–symmetric systems can be made similarly to this paper, substituting the

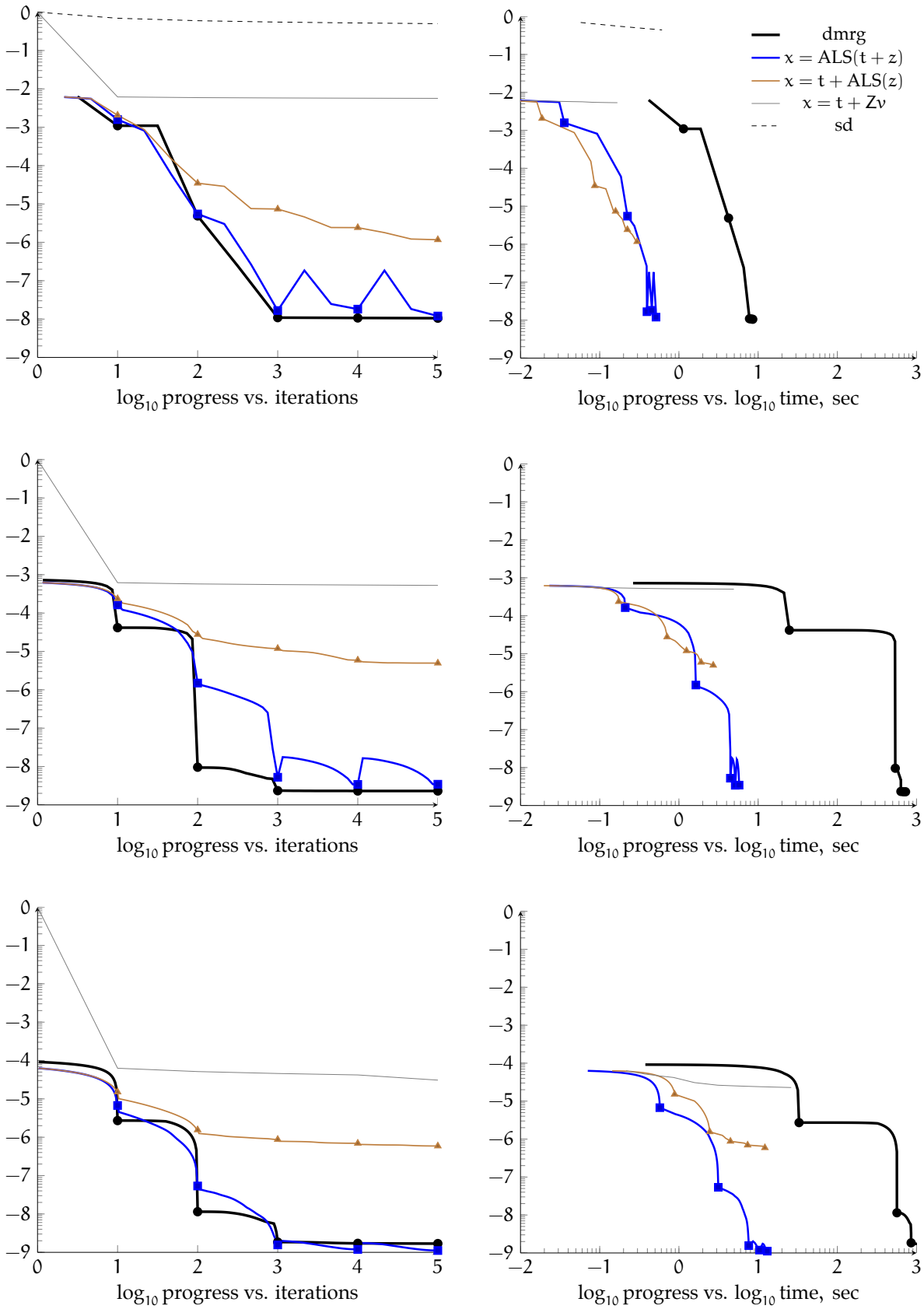

Figure 3: A–norm of the error in different methods versus iterations (left), and CPU time (right). Dimension of the problem is d = 3 (top), d = 16 (middle), d = 64 (bottom).

steepest descent algorithm by the minimal residual method. The second Krylov vector is required in MINRES–type algorithms, which have to be approximated and the convergence of perturbed method should be discussed similarly to the Theorem 1. Second, the complexity of the proposed methods w.r.t. tensor ranks should be studied and improved using faster (eg, cross) approximation schemes. Finally, we will develop and analyze the AMEn method for which the enrichment steps are mixed with ALS optimization, i.e., there is no explicit steepest descent step.

The proposed algorithms are already applied to the solution of the chemical master equation in dimensions up to twenty [7], and more practical applications will follow soon.

References

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

[2] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, and P. Wojtaszczyk, *Convergence rates for greedy algorithms in reduced basis methods*, SIAM J. Math. Anal., 43 (2011), pp. 1457–1472.

[3] Hans-Joachim Bungatrz and Michael Griebel, *Sparse grids*, Acta Numerica, 13 (2004), pp. 147–269.

[4] J. D. Caroll and J. J Chang, *Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart–Young decomposition*, Psychometrika, 35 (1970), pp. 283–319.

[5] S. R. Chinnamsetty, M. Espig, W. Hackbusch, B. N. Khoromskij, and H. J. Flad, *Tensor product approximation with optimal rank in quantum chemistry*, J. Chem. Phys., 127 (2007), pp. 84–110.

[6] S. V. Dolgov, *TT-GMRES: on solution to a linear system in the structured tensor format*, arXiv preprint 1206.5512 (To appear in: Rus. J. of Num. An. and Math. Model.), 2012.

[7] S. V. Dolgov and B. N. Khoromskij, *Tensor-product approach to global time-space-parametric discretization of chemical master equation*, Preprint 68, MPI MIS, 2012.

[8] ——, *Two-level Tucker-TT-QTT format for optimized tensor calculus*, Preprint 19, MPI MIS, 2012.

[9] S. V. Dolgov and I. V. Oseledets, *Solution of linear systems and matrix inversion in the TT-format*, SIAM J. Sci. Comput., 34 (2012), pp. A2718–A2739.

[10] A. Einstein, *Die Grundlage der allgemeinen Relativitätstheorie*, Annalen der Physik, 354 (1916), pp. 769–822.

[11] M. Espig, W. Hackbusch, and A. Khachatryan, *On the convergence of alternating least squares optimisation in tensor format representations*, in preparation, MPI MIS.

[12] A. Falcó and A. Nouy, *Proper orthogonal decomposition for nonlinear convex problems in tensor Banach spaces*, Numer. Math., 121 (2012), pp. 503–530.

[13] S. A. Goreinov, I. V. Oseledets, and D. V. Savostyanov, *Wedderburn rank reduction and Krylov subspace method for tensor approximation. Part 1: Tucker case*, SIAM J. Sci. Comput., 34 (2012), pp. A1–A27.

[14] W. Hackbusch, *Tensor spaces and numerical tensor calculus*, Springer–Verlag, Berlin, 2012.

[15] W. Hackbusch and S. Kühn, *A new scheme for the tensor representation*, J. Fourier Anal. Appl., 15 (2009), pp. 706–722.

[16] R. A. Harshman, *Foundations of the PARAFAC procedure: models and conditions for an explanatory multimodal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970), pp. 1–84.

[17] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys, 6 (1927), pp. 164–189.

[18] S. Holtz, T. Rohwedder, and R. Schneider, *The alternating linear scheme for tensor optimization in the tensor train format*, SIAM J. Sci. Comput., 34 (2012), pp. A683–A713.

[19] R. A. Horn and C.R. Johnson, *Matrix analysis*, Cambridge university press, 1985.

[20] L. V. Kantorovich, *Funktsionallniy analyz i prikladnaya matematika*, Uspehi Mat. Nauk, 3 (1945), pp. 89–185.

[21] V. A. Kazeev and B. N. Khoromskij, *Low-rank explicit QTT representation of the Laplace operator and its inverse*, SIAM J. Matrix Anal. Appl., 33 (2012), pp. 742–758.

[22] B. N. Khoromskij, $\mathcal{O}(\mathrm{d}\log \mathrm{n})$*–Quantics approximation of* N*–*d *tensors in high-dimensional numerical modeling*, Constr. Appr., 34 (2011), pp. 257–280.

[23] ——, *Tensor-structured numerical methods in scientific computing: Survey on recent advances*, Chemometr. Intell. Lab. Syst., 110 (2012), pp. 1–19.

[24] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, SIAM Review, 51 (2009), pp. 455–500.

[25] D. Kressner and C. Tobler, *Preconditioned low-rank methods for high-dimensional elliptic PDE eigenvalue problems*, Computational Methods in Applied Mathematics, 11 (2011), pp. 363–381.

[26] C. Le Bris, T. Leliévre, and Y. Maday, *Results and questions on a nonlinear approximation approach for solving high-dimensional partial differential equations*, Constr. Approx., 30 (2009), pp. 621–651.

[27] O. S. Lebedeva, *Tensor conjugate-gradient-type method for Rayleigh quotient minimization in block QTT-format*, Russ. J. Numer. Anal. Math. Modelling, 26 (2011), p. 465–489.

[28] A. W. Marshall and L. Olkin, *Matrix version of the Cauchy and Kantorovich inequalities*, Aequationes Mathematicae, 40 (1990), pp. 89–93.

[29] I. V. Oseledets, *DMRG approach to fast linear algebra in the TT–format*, Comput. Meth. Appl. Math, 11 (2011), pp. 382–393.

[30] ——, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317.

[31] ——, *Constructive representation of functions in low-rank tensor formats*, Constr. Appr., (2012). accepted.

[32] I. V. Oseledets, D. V. Savostianov, and E. E. Tyrtyshnikov, *Tucker dimensionality reduction of three-dimensional arrays in linear time*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 939–956.

[33] I. V. Oseledets, D. V. Savostyanov, and E. E. Tyrtyshnikov, *Linear algebra for tensor problems*, Computing, 85 (2009), pp. 169–188.

[34] I. V. Oseledets and E. E. Tyrtyshnikov, *TT-cross approximation for multidimensional arrays*, Linear Algebra Appl., 432 (2010), pp. 70–88.

[35] T. Rohwedder and A. Uschmajew, *Local convergence of alternating schemes for optimization of convex problems in the TT format*, SIAM J Num. Anal., ((2013)). to appear.

[36] D. V. Savostyanov, *Polilinear approximation of matrices and integral equations*, PhD thesis, INM RAS, Moscow, 2006. (in Russian).

[37] D. V. Savostyanov and I. V. Oseledets, *Fast adaptive interpolation of multi-dimensional arrays in tensor train format*, in Proceedings of 7th International Workshop on Multidimensional Systems (nDS), IEEE, 2011.

[38] S. A. Smolyak, *Quadrature and interpolation formulas for tensor products of certain class of functions*, Dokl. Akad. Nauk SSSR, 148 (1964), pp. 1042–1053. Transl.: Soviet Math. Dokl. 4:240-243, 1963.

[39] G. W. Steward, *On the perturbation of pseudo–inverses, projections and linear least squares problems*, SIAM Review, 19 (1977), pp. 634–662.

[40] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279–311.

[41] A. Uschmajew, *Local convergence of the alternating least squares algorithm for canonical tensor approximation*, SIAM J Matr. Anal. Appl., 33 (2012), pp. 639–652.

[42] S. R. White, *Density-matrix algorithms for quantum renormalization groups*, Phys. Rev. B, 48 (1993), pp. 10345–10356.