

**Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig**

**Alternating minimal energy methods for linear
systems in higher dimensions. Part II: Faster
algorithm and application to nonsymmetric
systems**

(revised version: April 2013)

by

Sergey Dolgov and Dmitry Savostyanov

Preprint no.: 36

2013



Alternating minimal energy methods for linear systems in higher dimensions. Part II: Faster algorithm and application to nonsymmetric systems^{*}

Sergey V. Dolgov[†] and Dmitry V. Savostyanov[‡]

April 11, 2013

Abstract

In this paper we accomplish the development of the fast rank–adaptive solver for tensor–structured symmetric positive definite linear systems in higher dimensions. In [9] this problem is approached by alternating minimization of the energy function, which we combine with steps of the basis expansion in accordance with the steepest descent algorithm. In this paper we combine the same steps in such a way that the resulted algorithm works with one or two neighboring cores at a time. The recurrent interpretation of the algorithm allows to prove the global convergence and to estimate the convergence rate. We also propose several strategies, both rigorous and heuristic, to compute new subspaces for the basis enrichment in a more efficient way. We test the algorithm on a number of high–dimensional problems, including the non-symmetrical Fokker–Planck and chemical master equations, for which the efficiency of the method is not fully supported by the theory. In all examples we observe a convincing fast convergence and high efficiency of the proposed method.

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

1 Introduction

In this paper we develop the results of [9]. We consider tensor–structured linear systems, which arise naturally from high–dimensional problems, e.g. PDEs. The number

^{*}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 and EPSRC grant EP/H003789/1 at the University of Southampton. This work was initiated when D.S. was with the Institute of Numerical Mathematics RAS, Moscow.

[†]Max-Planck-Institut für Mathematik in den Naturwissenschaften, Inselstr. 22-26, D-04103 Leipzig, Germany, and Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina str. 8, 119333 Moscow, Russia (sergey.v.dolgov@gmail.com)

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

of unknowns grows exponentially w.r.t. the number of dimensions d , which makes standard algorithms inefficient even for moderate d . This problem is known as the *curse of dimensionality*, and is attacked by different low-parametric approximations, e.g. *sparse grids* [32, 3] and *tensor product methods* [20, 18, 12]. A particularly simple, elegant and efficient representation of high-dimensional data is a linear tensor network, also called the *matrix product states* (MPS) and *tensor train* (TT) format.

The MPS approach was originally proposed in the quantum physics community to represent the quantum states of many-body systems [10, 19]. This representation was rediscovered as the TT format by Oseledets and Tyrtysnikov [24], who were looking for a proper method to generalize a low-rank decomposition of matrices to high-dimensional arrays (tensors). The MPS approach came with the *alternating least squares* (ALS) and *density matrix renormalization group* (DMRG) [36, 26] algorithms for the ground state problem. The ALS considers the minimization of the Rayleigh quotient over the vectors with a fixed tensor structure, while DMRG does the same allowing the rank of the solution to change. Experiments from quantum physics point out that the convergence of the DMRG is usually notably fast, while the one of the ALS can be rather poor.

The general numerical linear algebra context in which the TT format is introduced allows to think more widely about the power of tensor representations. For instance, we can apply DMRG-like techniques to high-dimensional problems other than just the ground state problem, e.g. interpolation of high-dimensional data [25, 31], solution of linear systems [14, 8], fast linear algebra in tensor formats [23]. We can also consider better alternatives to the DMRG, which follow the same *alternating linear scheme* (ALS) framework, but are numerically more efficient. A tempting goal is to obtain an algorithm which has the DMRG-like convergence and the ALS-like numerical complexity. In [9] we present such an algorithm for a solution of symmetric positive definite (SPD) linear systems in higher dimensions.

The central idea in [9] is to support the alternating steps, i.e. optimization in a fixed tensor manifold, by steps which *expand* the basis in accordance with some classical iterative algorithms. A *steepest descent* (SD) algorithm is a natural choice for SPD problems. The *enrichment* step uses the essential information about the *global* residual of the large high-dimensional system on the local optimization step, that helps to escape the spurious local minima introduced by the nonlinear tensor formulation and ensure the global convergence. The convergence rate of the whole method can then be established adapting a classical theory. In contrast, optimization in the fixed tensor manifolds can be analyzed via the Gauss-Seidel theory and only *local* convergence estimates are available [28], which hold only in a (very) small vicinity of the exact solution.

The global enrichment step used in algorithms “ $t + \text{ALS}(z)$ ” and “ $\text{ALS}(t + z)$ ” in [9] modifies all components of the tensor train format simultaneously. There is nothing particularly wrong with this, but it is interesting to mix the same steps differently to obtain the algorithm which works with only one or two neighboring components at once, similarly to the DMRG technique. In this paper we develop such a method, namely the *alternating minimal energy* (AMEn) algorithm. We prove the global convergence of AMEn and estimate the convergence rate w.r.t. the one of the steepest descent algorithm. We also propose several methods to compute the required local component of the global residual, using either the SVD-based approximation, or incomplete Cholesky decomposition, or low-rank ALS approximation.

The rest of the paper is organized as follows. In Section 2 we introduce necessary

definitions and notations. In Section 3 we propose the AMEn algorithm, then we compare it with similar algorithms from [9] and prove the convergence theorem. In Section 4 we discuss efficient methods to compute the required component of the residual. In Section 5 we test the algorithm on a number of high-dimensional problems, including the non-symmetrical Fokker–Planck and chemical master equations, for which the efficiency of the method is not fully supported by the theory. In all examples we observe a convincing fast convergence and high efficiency of the proposed method, as well as the advantages of the AMEn algorithm over the previously proposed ones.

2 Notations and definitions

This paper is based on the notations of [9], which we recall briefly here.

We consider linear systems $Ax = y$ in d -dimensional space, i.e. assume that a vector x has d indices i_1, \dots, i_d , and $i_k = 1, \dots, n_k$, $k = 1, \dots, d$. Such arrays are referred to as d -tensors $x = [x(i_1, \dots, i_d)]$. For the purposes of this paper it is convenient to consider a *vectorization* of a tensor

$$x = \text{vec } x, \quad x(\overline{i_1 \dots i_d}) = x(i_1, \dots, i_d),$$

where $\overline{i_1 \dots i_d}$ denotes a single index combined from i_1, \dots, i_d ¹. This *index grouping* is widely used throughout the paper. In the following we do not distinguish between x and x .

The tensor train (TT) representation of x is written as the following multilinear map²,

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

where i_k are referred to as *mode* (physical) indices, $\alpha_k = 1, \dots, r_k$ are the *rank* indices, $X^{(k)}$ are the tensor train *cores* (TT-cores) and $\bar{X} = (X^{(1)}, \dots, X^{(d)})$ denotes the tensor train. We follow the Einstein summation convention, which assumes a summation over every pair of repeated indices. All equations are supposed to hold for all possible values of free (unpaired) indices.

The τ mapping is defined also for a subset of TT-cores (a *subtrain*) and maps it to the *interface* matrix of size $n_1 \dots n_k \times r_k$, defined as follows,

$$\begin{aligned} X^{\leq k} &= \tau(X^{(1)}, \dots, X^{(k)}), \\ X^{\leq k}(\overline{i_1 i_2 \dots i_k}, \alpha_k) &= X_{\alpha_1}^{(1)}(i_1) X_{\alpha_1 \alpha_2}^{(2)}(i_2) \dots X_{\alpha_{k-1}, \alpha_k}^{(k)}(i_k), \end{aligned} \quad (2)$$

¹The multi-index can be defined following the *big-endian* convention $\overline{i_1 \dots i_d} = i_d + (i_{d-1} - 1)n_d + \dots + (i_1 - 1)n_2 \dots n_d$ or *little-endian* convention $\overline{i_1 \dots i_d} = i_1 + (i_2 - 1)n_1 + \dots + (i_d - 1)n_1 \dots n_{d-1}$. The big-endian notation is similar to numbers written in the positional system, while the little-endian notation is used in numerals in the Arabic scripts and is consistent with the Fortran style of indexing. The definition of the Kronecker (tensor) product \otimes should be also consistent with the chosen endianness. The orthodox definition in linear algebra assumes the big-endianness, while the development of the efficient program code usually makes us think in the little-endian way. The rest of the paper can be read without a particular care of the endianness. It is enough to remember that $z = x \otimes y$ means $z(\overline{ij}) = x(i)y(j)$.

²Note that τ maps tensor train cores to a vectorized representation of a d -tensor, not to the d -tensor itself, cf. *quantized* tensor train (QTT) [17]. In this paper we do not distinguish between them and keep the notation simple.

and similarly for symbols $X^{<k}$, $X^{>k}$ and $X^{\geq k}$. For x given by (1) we have $X^{[k]} = X^{\leq k} X^{>k}$.

Note that the definition of τ allows us to write

$$x = \tau(X^{\leq k}, X^{>k}) = \tau(X^{<k}, X^{(k)}, X^{>k}) = \tau(X^{<k}, X^{(k)}, X^{(k+1)}, X^{\geq k+2}),$$

where two last mappings depict the decompositions used in ALS and DMRG algorithms proposed by S. White et al. [36, 26] for the ground state problem in Quantum Physics. The original DMRG algorithm is formulated via the minimization of the Rayleigh quotient, where the heavily nonlinear high-dimensional optimization is reduced to the sequence of numerically tractable optimizations over the elements of each core.

Similarly, we consider the solution of a linear equation $Ax = y$ through the minimization of the *energy* function

$$J(x) = \|x_* - x\|_A^2 = (x, Ax) - 2\Re(x, y) + \text{const}, \quad (3)$$

where $x_* = A^{-1}y$ is the exact solution, and $\|u\|_A^2 = (u, u)_A = (u, Au)$ denotes the A -norm of a vector u . Following the *alternating linear scheme* (ALS), the high-dimensional minimization is reduced to the minimization w.r.t. all cores one-by-one. Each *local* minimization is equivalent to the solution of a linear system, which is tractable due to a moderate size. The high-dimensional linear system can be split into a sequence of one-dimensional systems due to the *linearity* of the tensor train format $\tau(X^{(1)}, \dots, X^{(d)})$ w.r.t. each TT-core $X^{(k)}$. This linearity writes as the following matrix-by-vector product

$$\begin{aligned} x &= \tau(\bar{X}) = \tau(X^{<k}, X^{(k)}, X^{>k}) = \mathcal{P}_{\neq k}(\bar{X})x^{(k)}, \\ \mathcal{P}_{\neq k}(\bar{X}) &= X^{<k} \otimes I_{n_k} \otimes (X^{>k})^\top, \end{aligned} \quad (4)$$

where the vectorized TT-core $x^{(k)}$ is a reshape of the three-dimensional array into a vector

$$x^{(k)} = \text{vec } X^{(k)}, \quad x^{(k)}(\overline{\alpha_{k-1} i_k \alpha_k}) = X_{\alpha_{k-1}, \alpha_k}^{(k)}(i_k). \quad (5)$$

The elementwise definition of the *frame matrix* $\mathcal{X}_{\neq k} = \mathcal{P}_{\neq k}(\bar{X})$ is the following

$$\mathcal{X}_{\neq k}(\overline{i_1 \dots i_d}, \overline{\alpha_{k-1} j_k \alpha_k}) = X_{\alpha_1}^{(1)}(i_1) \dots X_{\alpha_{k-2}, \alpha_{k-1}}^{(k-1)}(i_{k-1}) \delta(i_k, j_k) X_{\alpha_k, \alpha_{k+1}}^{(k+1)}(i_{k+1}) \dots X_{\alpha_{d-1}}^{(d)}(i_d), \quad (6)$$

where $\delta(i, j)$ is the Kronecker symbol, i.e., $\delta(i, j) = 1$ if $i = j$ and $\delta(i, j) = 0$ elsewhere. Similarly we define frame matrices $\mathcal{P}_{\leq k}(\bar{X})$, $\mathcal{P}_{\geq k}(\bar{X})$, $\mathcal{P}_k(\bar{X})$. For example, $\mathcal{P}_{\leq k}(\bar{X})$ writes

$$\begin{aligned} \mathcal{X}_{\leq k} &= \mathcal{P}_{\leq k}(\bar{X}) = X^{\leq k} \otimes I_{n_{k+1} \dots n_d}, \\ \mathcal{X}_{\leq k}(\overline{i_1 \dots i_d}, \overline{\alpha_k j_{k+1} \dots j_d}) &= X_{\alpha_1}^{(1)}(i_1) \dots X_{\alpha_{k-1}, \alpha_k}^{(k)}(i_k) \delta(i_{k+1}, j_{k+1}) \dots \delta(i_d, j_d). \end{aligned} \quad (7)$$

The A -orthogonal projector on the subspace span U is defined as follows

$$R_U = U(U^*AU)^{-1}U^*A. \quad (8)$$

It is easy to check that $R_U^2 = R_U$ and $R_U U = U$. Also, for any v such that $(U, v)_A = 0$ it holds $R_U v = 0$, hence the name A -orthogonal.

3 Alternating minimal energy methods

3.1 AMEn and ALS

One of the main results of the previous paper [9] is the algorithm $\text{ALS}(t+z)$. Each iteration of this algorithm consists of one ‘global’ basis enrichment step which changes all the cores, followed by d update steps over all the cores subsequently. Classical optimization algorithms for tensor networks, e.g. ALS and DMRG, follow the alternating linear framework, i.e. update one or two neighboring cores at a time. We would like to keep the enrichment as well as other steps within the same idea, and propose another version of the method as Alg. 1, which we will refer to as the *alternating minimal energy algorithm* (AMEn). The difference between two algorithms is illustrated by a simple three-dimensional example in Fig. 1.

To analyse the convergence of the $\text{ALS}(t+z)$ algorithm, we can see it as a method which implements the (approximate) steepest descent step followed by a sequence of optimization steps for the energy function. The approximate steepest descent step $x = t + h\tilde{z}$ with $\tilde{z} \approx z = y - At$ and optimal h gives [9, Thm. 1] the convergence rate $\omega_z = \omega_z + \mathcal{O}(\varepsilon)$, where ω_z denotes the progress of the exact steepest descent step. A fortiori, the *global convergence* of the $\text{ALS}(t+z)$ is proven with the convergence rate not slower than ω_z . The AMEn algorithm does not have a global enrichment step and the convergence can not be proven in one line. However, the convergence analysis is possible if Alg. 1 is seen as a recurrent method. Though the theoretical estimates do not provide a clear distinction which method is preferable, in numerical experiments in Sec. 5 we will observe that the AMEn technique delivers more accurate solution, while the average convergence rate is almost the same as of the $\text{ALS}(t+z)$ method.

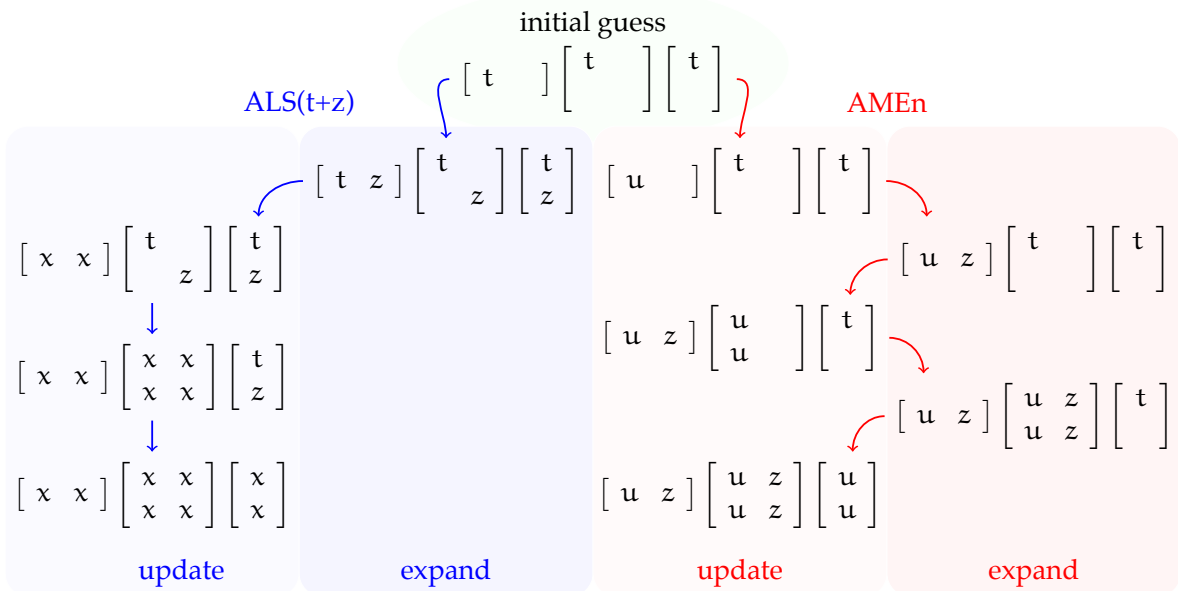


Figure 1: Schematic representation of $\text{ALS}(t+z)$ and AMEn algorithms

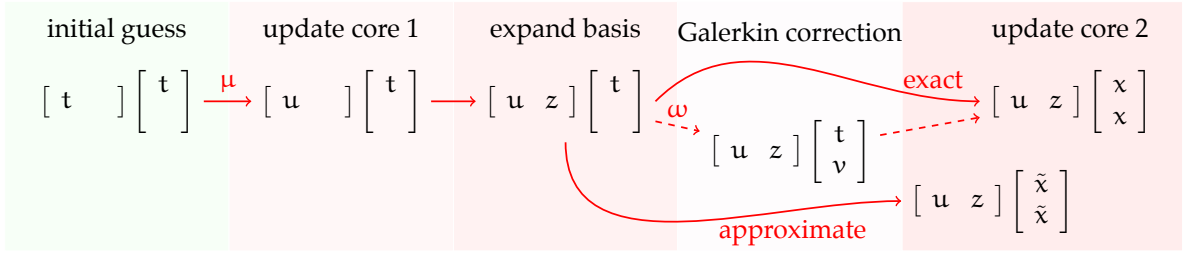


Figure 2: Schematic representation of the AMEn algorithm

3.2 AMEn in two dimensions

The idea of the convergence analysis is introduced by a two-dimensional example, see Fig. 2. In two dimensions Alg. 1 can be seen as a sequence of the following operations.

1. Start from the initial guess

$$\mathbf{t} = \begin{bmatrix} \mathbb{T}^{(1)} & \end{bmatrix} \begin{bmatrix} \mathbb{T}^{(2)} \end{bmatrix}$$

2. Update the first TT-core, minimizing the energy function over the entries of $\mathbb{T}^{(1)}$

$$\mathbf{u}^{(1)} = \arg \min_{\mathbb{T}^{(1)}} J \left(\begin{bmatrix} \mathbb{T}^{(1)} & \end{bmatrix} \begin{bmatrix} \mathbb{T}^{(2)} \end{bmatrix} \right), \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}^{(1)} & \end{bmatrix} \begin{bmatrix} \mathbb{T}^{(2)} \end{bmatrix}$$

3. Expand the basis in the first core using the first TT-core of the residual.

$$\mathbf{y} - \mathbf{A}\mathbf{u} = \mathbf{z} \approx \tilde{\mathbf{z}} = \tau(\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}), \quad \mathbf{u} = \begin{bmatrix} \mathbf{u}^{(1)} & \mathbf{z}^{(1)} \end{bmatrix} \begin{bmatrix} \mathbb{T}^{(2)} \end{bmatrix}$$

4. Perform the Galerkin correction step by minimizing the energy function over the bottom part of second TT-core.

$$\mathbf{v}^{(2)} = \arg \min_{\mathbb{V}^{(2)}} J \left(\begin{bmatrix} \mathbf{u}^{(1)} & \mathbf{z}^{(1)} \end{bmatrix} \begin{bmatrix} \mathbb{T}^{(2)} \\ \mathbb{V}^{(2)} \end{bmatrix} \right), \quad \mathbf{v} = \begin{bmatrix} \mathbf{u}^{(1)} & \mathbf{z}^{(1)} \end{bmatrix} \begin{bmatrix} \mathbb{T}^{(2)} \\ \mathbf{v}^{(2)} \end{bmatrix}$$

5. Minimize the energy function over all entries of the second TT-core

$$\mathbf{x}^{(2)} = \arg \min_{\mathbb{X}^{(2)}} J \left(\begin{bmatrix} \mathbf{u}^{(1)} & \mathbf{z}^{(1)} \end{bmatrix} \mathbb{X}^{(2)} \right), \quad \mathbf{x} = \begin{bmatrix} \mathbf{u}^{(1)} & \mathbf{z}^{(1)} \end{bmatrix} \mathbf{x}^{(2)}$$

The ALS update steps 2 and 5 reduce the energy function by a factor $\mu_1^2 \leq 1$ and $\mu_2^2 \leq 1$, respectively, which can be rigorously estimated only locally, i.e. in a very small vicinity of a true solution. The basis enrichment step 3 does not provide any progress, because it does not change the solution vector, but only its TT-representation. The Galerkin correction step 4 does not technically present in the algorithm. If we omit it, the update step 5 will deliver the same TT-core, optimizing the energy function over the positions occupied by both upper and bottom parts of the second core. Without actually affecting the result of the computations, step 4 is essential to analyse the convergence of the whole method, since the progress of the Galerkin correction step can be estimated w.r.t. the one of the steepest descent. This idea is formally expressed as the following theorem.

Theorem 1. In the notations set above for the two-dimensional linear system $Ax = y$ one iteration of AMEn Alg. 1 provides the following progress

$$\frac{J(x)}{J(t)} = \frac{\|x_* - x\|_A^2}{\|x_* - t\|_A^2} = \frac{\|x_* - u\|_A^2}{\|x_* - t\|_A^2} \frac{\|x_* - x\|_A^2}{\|x_* - u\|_A^2} = \mu^2 \omega_x^2,$$

$$\mu \leq 1, \quad \omega_x \leq \omega_z \leq \omega_{\tilde{z}} = \omega_z + \mathcal{O}(\varepsilon),$$

where $\mathcal{X} = \mathcal{P}_1(\bar{X}) = X^{(1)} \times I_{n_2}$, $\mathcal{Z} = \mathcal{P}_1(\bar{Z}) = Z^{(1)} \times I_{n_2}$, and ω_S is defined by

$$\omega_S^2 = 1 - \frac{(x_* - u, R_S(x_* - u))_A}{(x_* - u, x_* - u)_A}. \quad (9)$$

Proof. The minimization in step 5 is written as $x_*^{(2)} = \arg \min_{x^{(2)}} J(\mathcal{X}x^{(2)})$, $x^{(2)} = \text{vec } X^{(2)}$, cf. (4), and the gradient is zero when the Galerkin conditions $(\mathcal{X}^* A \mathcal{X})x_*^{(2)} = \mathcal{X}^* y$ are met. The solution after one AMEn iteration writes as follows

$$x = \mathcal{X}x_*^{(2)} = \mathcal{X}(\mathcal{X}^* A \mathcal{X})^{-1} \mathcal{X}^* A x_* = R_{\mathcal{X}} x_*, \quad (10)$$

where $R_{\mathcal{X}}$ is the A -orthogonal projector on \mathcal{X} , cf. (8). Since $u \in \text{span } \mathcal{X}$ we have

$$x_* - x = (I - R_{\mathcal{X}})(x_* - u), \quad \frac{J(x)}{J(u)} = \frac{\|x_* - x\|_A^2}{\|x_* - u\|_A^2} = 1 - \frac{(x_* - u, R_{\mathcal{X}}(x_* - u))_A}{(x_* - u, x_* - u)_A} = \omega_{\mathcal{X}}^2. \quad (11)$$

Similarly, the progress of the Galerkin correction step 4 is estimated as follows (see [9]),

$$x_* - v = (I - R_{\mathcal{Z}})(x_* - u), \quad \frac{J(v)}{J(u)} = \frac{\|x_* - v\|_A^2}{\|x_* - u\|_A^2} = 1 - \frac{(x_* - u, R_{\mathcal{Z}}(x_* - u))_A}{(x_* - u, x_* - u)_A} = \omega_{\mathcal{Z}}^2. \quad (12)$$

It is not easy to estimate $\omega_{\mathcal{X}}$ directly. However, since $Z^{(1)}$ is a part of $X^{(1)}$, we have $\text{span } \mathcal{Z} \subset \text{span } \mathcal{X}$ and therefore $\omega_{\mathcal{X}} \leq \omega_{\mathcal{Z}}$. Similarly, since $\tilde{z} \in \text{span } \mathcal{Z}$ it holds $\omega_{\mathcal{Z}} \leq \omega_{\tilde{z}}$, where $\omega_{\tilde{z}}$ denotes a progress of the perturbed steepest descent step. The final estimate for $\omega_{\tilde{z}}$ is obtained in [9, Thm 1] with a precise derivation of the asymptotic $\mathcal{O}(\varepsilon)$ term. To finish the proof we note that $J(u)/J(t) = \mu^2 \leq 1$ by construction of step 2. \square

Remark 1. A convergence rate ω_z of the steepest descent algorithm is estimated using the Kantorovich inequality as follows

$$\omega_z \leq \frac{\lambda_{\max}(A) - \lambda_{\min}(A)}{\lambda_{\max}(A) + \lambda_{\min}(A)} = \Omega(A) < 1,$$

where $\lambda_{\min}(A)$ and $\lambda_{\max}(A)$ denote the smallest and largest eigenvalues of A . For any ω_z we can choose such a threshold level ε that $\omega_{\tilde{z}} < 1$, which guarantees the global convergence of the AMEn algorithm.

Algorithm 1 AMEn algorithm

Require: System $Ax = y$ in the TT-format, initial guess $t = \tau(\bar{T})$

Ensure: Updated solution $x = \tau(\bar{X})$

- 1: **for** $k = 1, \dots, d$ **do** {Cycle over TT-cores}
 - 2: Update $U^{(k)} = \arg \min_{S^{(k)}} J(\tau(X^{(1)}, \dots, X^{(k-1)}, S^{(k)}, T^{(k+1)}, \dots, T^{(d)}))$,
 - 3: For local problem (14) approximate $z_k = y_k - A_k u_k \approx \tilde{z}_k = \tau(Z_k^{(k)}, \dots, Z_k^{(d)})$
 - 4: Expand the basis $X^{(k)} = \begin{bmatrix} U^{(k)} & Z_k^{(k)} \end{bmatrix}$, $S^{(k+1)} := \begin{bmatrix} T^{(k+1)} \\ 0 \end{bmatrix}$
 - 5: Recover the TT-orthogonality of $\bar{X}_k = (X^{(1)}, \dots, X^{(k)}, S^{(k+1)}, T^{(k+2)}, \dots, T^{(d)})$.
 - 6: **end for**
 - 7: **return** $x = \tau(X^{(1)}, \dots, X^{(d)})$
-

3.3 AMEn in higher dimensions

In higher dimensions, the AMEn Alg. 1 can be described using the same scheme. We start from an initial guess $t = \tau(T^{(1)}, T^{\geq 2})$. In step 2 we update the first core obtaining $u = \tau(U^{(1)}, T^{\geq 2})$. In step 3 we approximate the residual $y - Au = z \approx \tilde{z} = \tau(Z^{(1)}, Z^{\geq 2})$ and expand the first core by $Z^{(1)}$. These operations are numerically tractable, see Sec. 4 for details. Minimization problem in step 5 leads to the following linear system

$$(\mathcal{X}_1^* A \mathcal{X}_1) x^{\geq 2} = \mathcal{X}_1^* y, \quad \mathcal{X}_1 = \mathcal{P}_1(\bar{X}) = X^{(1)} \otimes I_{n_2 \dots n_d}, \quad x^{\geq 2} = \tau(X^{(2)}, \dots, X^{(d)}), \quad (13)$$

which has $r_1 n_2 \dots n_d$ unknowns and is still too large to be solved directly. Note that $\mathcal{X}_1 = X^{(1)} \otimes I$ is a rank-1 multilevel matrix, hence $B = \mathcal{X}_1^* A \mathcal{X}_1$ has the same TT-ranks as A , but its TT-representation is shorter by one core. Similarly, $\mathcal{X}_1^* y$ represents a smaller right-hand side in the TT format, and the solution $x^{\geq 2}$ is sought in the TT format as well. Therefore, the linear problem in d dimensions is reduced to the one in $d - 1$ dimensions, i.e. to the minimization over the remaining subtrain, and the same algorithm is applied recurrently.

The convergence rate of AMEn is defined by a recurrent application of the result of Thm. 1. We need to consider a sequence of the reduced problems

$$A_k x^{\geq k} = y_k, \quad A_k = \mathcal{X}_{<k}^* A \mathcal{X}_{<k}, \quad y_k = \mathcal{X}_{<k}^* y, \quad \mathcal{X}_{<k} = \mathcal{P}_{<k}(\bar{X}). \quad (14)$$

The initial guess is $t^{\geq k} = \tau(T^{(k)}, T^{\geq k+1})$, the update step for the core $T^{(k)}$ gives $u_k = \tau(U^{(k)}, T^{\geq k+1})$, further steps of AMEn return the solution $x^{\geq k} = \tau(X^{(k)}, \dots, X^{(d)})$, and the true solution is defined by $x_\star^{\geq k} = A_k^{-1} y_k$. Similarly to Thm. 1 we have

$$\frac{\|x_\star^{\geq k} - u_k\|_{A_k}^2}{\|x_\star^{\geq k} - t^{\geq k}\|_{A_k}^2} = \mu_k^2 \leq 1, \quad \frac{\|x_\star^{\geq k} - x^{\geq k}\|_{A_k}^2}{\|x_\star^{\geq k} - u_k\|_{A_k}^2} = \omega_k^2 = 1 - \frac{(c_k, R_x^{\langle A_k \rangle} c_k)_{A_k}}{\|c_k\|_{A_k}^2}, \quad (15)$$

where $c_k = x_\star^{\geq k} - u_k$, $\mathcal{X} = \mathcal{P}_1(X^{(k)}, \dots, X^{(d)})$, and $R_x^{\langle A_k \rangle}$ denotes the A_k -orthogonal projector to \mathcal{X} . With these definitions in hand we write the following theorem.

Theorem 2. The AMEn Alg. 1 converges globally with the following convergence rate

$$\begin{aligned} \frac{\|x_\star - x\|_{A}^2}{\|x_\star - t\|_{A}^2} &= \mu_1^2 \left(\omega_1^2 + (1 - \omega_1^2) \mu_2^2 \left(\omega_2^2 + (1 - \omega_2^2) \mu_3^2 \left(\omega_3^2 + \dots + \mu_{d-1}^2 \omega_{d-1}^2 \right) \dots \right) \right) \\ &= \sum_{k=1}^{d-1} \omega_k^2 \prod_{j=1}^{k-1} (1 - \omega_j^2) \prod_{j=1}^k \mu_j^2 = \phi_d^2. \end{aligned} \quad (16)$$

Proof. In two dimensions, the theorem reduces to Thm. 1, which proves the base of recursion. We suppose that (16) holds in $d - 1$ dimension and prove it recurrently for d dimensions.

If $\mathbf{t} = \tau(\mathbf{T}^{(1)}, \mathbf{T}^{\geq 2})$ is the initial guess, $\mathbf{u} = \tau(\mathbf{U}^{(1)}, \mathbf{T}^{\geq 2})$ appears after the update of the first core, and $\mathbf{x} = \tau(\mathbf{X}^{(1)}, \mathbf{X}^{\geq 2})$ is the result returned by one iteration of AMEn, the error $\mathbf{x}_* - \mathbf{x}$ is written as follows

$$\mathbf{x}_* - \mathbf{x} = \mathbf{x}_* - \tau(\mathbf{X}^{(1)}, \mathbf{X}_*^{\geq 2}) + \tau(\mathbf{X}^{(1)}, \mathbf{X}_*^{\geq 2}) - \tau(\mathbf{X}^{(1)}, \mathbf{X}^{\geq 2}) = (\mathbf{x}_* - \mathcal{X}_1 \mathbf{x}_*^{\geq 2}) + \mathcal{X}_1 (\mathbf{x}_*^{\geq 2} - \mathbf{x}^{\geq 2}),$$

where the first term is the error of the first ('outer') AMEn step provided the solution $\mathbf{x}_*^{\geq 2}$ of the reduced problem (13) is computed exactly, and the second term is the error of other ('inner') AMEn steps, which we find using the assumption of the recurrence. Using (11) we show that these terms are Λ -orthogonal,

$$\begin{aligned} \mathbf{x}_* - \mathbf{x} &= (\mathbf{I} - \mathbf{R}_{\mathcal{X}_1}) \mathbf{x}_* + \mathcal{X}_1 (\mathbf{x}_*^{\geq 2} - \mathbf{x}^{\geq 2}), \\ \|\mathbf{x}_* - \mathbf{x}\|_{\Lambda}^2 &= \|(\mathbf{I} - \mathbf{R}_{\mathcal{X}_1}) \mathbf{x}_*\|_{\Lambda}^2 + \|\mathcal{X}_1 (\mathbf{x}_*^{\geq 2} - \mathbf{x}^{\geq 2})\|_{\Lambda}^2. \end{aligned} \quad (17)$$

The first term writes by Thm. 1 as follows,

$$\frac{\|(\mathbf{I} - \mathbf{R}_{\mathcal{X}_1}) \mathbf{x}_*\|_{\Lambda}^2}{\|\mathbf{x}_* - \mathbf{t}\|_{\Lambda}^2} = \frac{\|(\mathbf{I} - \mathbf{R}_{\mathcal{X}_1})(\mathbf{x}_* - \mathbf{u})\|_{\Lambda}^2}{\|\mathbf{x}_* - \mathbf{t}\|_{\Lambda}^2} = \frac{\|\mathbf{x}_* - \mathbf{u}\|_{\Lambda}^2}{\|\mathbf{x}_* - \mathbf{t}\|_{\Lambda}^2} \frac{\|(\mathbf{I} - \mathbf{R}_{\mathcal{X}_1})(\mathbf{x}_* - \mathbf{u})\|_{\Lambda}^2}{\|\mathbf{x}_* - \mathbf{u}\|_{\Lambda}^2} = \mu_1^2 \omega_1^2.$$

For the second term we need the following norm equivalence,

$$\|\mathcal{X}_1 (\mathbf{x}_*^{\geq 2} - \mathbf{x}^{\geq 2})\|_{\Lambda}^2 = (\mathbf{x}_*^{\geq 2} - \mathbf{x}^{\geq 2}, \mathcal{X}_1^* \mathbf{A} \mathcal{X}_1 (\mathbf{x}_*^{\geq 2} - \mathbf{x}^{\geq 2})) = \|\mathbf{x}_*^{\geq 2} - \mathbf{x}^{\geq 2}\|_{\mathbf{B}}^2,$$

where $\mathbf{B} = \mathcal{X}_1^* \mathbf{A} \mathcal{X}_1$. In the right-hand side we see the error norm of the AMEn algorithm applied to the linear problem (13) with $d - 1$ cores. According to our assumption, it writes by (16) as follows,

$$\frac{\|\mathbf{x}_*^{\geq 2} - \mathbf{x}^{\geq 2}\|_{\mathbf{B}}^2}{\|\mathbf{x}_*^{\geq 2} - \mathbf{t}^{\geq 2}\|_{\mathbf{B}}^2} = \sum_{k=2}^{d-1} \omega_k^2 \prod_{j=1}^{k-1} (1 - \omega_j^2) \prod_{j=1}^k \mu_j^2 = \Phi_{d-1}^2, \quad (18)$$

and for the second term we obtain

$$\frac{\|\mathcal{X}_1 (\mathbf{x}_*^{\geq 2} - \mathbf{x}^{\geq 2})\|_{\Lambda}^2}{\|\mathbf{x}_* - \mathbf{t}\|_{\Lambda}^2} = \Phi_{d-1}^2 \frac{\|\mathbf{x}_*^{\geq 2} - \mathbf{t}^{\geq 2}\|_{\mathbf{B}}^2}{\|\mathbf{x}_* - \mathbf{t}\|_{\Lambda}^2} = \Phi_{d-1}^2 \mu_1^2 \frac{\|\mathcal{X}_1 (\mathbf{x}_*^{\geq 2} - \mathbf{t}^{\geq 2})\|_{\Lambda}^2}{\|\mathbf{x}_* - \mathbf{u}\|_{\Lambda}^2}.$$

In the numerator we simplify $\mathcal{X}_1 \mathbf{x}_*^{\geq 2} = \mathbf{R}_{\mathcal{X}_1} \mathbf{x}_*$, cf. (10), and

$$\mathcal{X}_1 \mathbf{t}^{\geq 2} = \tau(\mathbf{X}^{(1)}, \mathbf{t}^{\geq 2}) = \tau \left(\begin{bmatrix} \mathbf{U}^{(1)} & \mathbf{Z}^{(1)} \end{bmatrix} \begin{bmatrix} \mathbf{t}^{\geq 2} \end{bmatrix} \right) = \tau(\mathbf{U}^{(1)}, \mathbf{t}^{\geq 2}) = \mathbf{u} = \mathbf{R}_{\mathcal{X}_1} \mathbf{u}.$$

Finally, we write last part of the second term as follows, cf. (11),

$$\frac{\|\mathcal{X}_1 (\mathbf{x}_*^{\geq 2} - \mathbf{t}^{\geq 2})\|_{\Lambda}^2}{\|\mathbf{x}_* - \mathbf{u}\|_{\Lambda}^2} = \frac{\|\mathbf{R}_{\mathcal{X}_1} (\mathbf{x}_* - \mathbf{u})\|_{\Lambda}^2}{\|\mathbf{x}_* - \mathbf{u}\|_{\Lambda}^2} = 1 - \omega_1^2.$$

Substituting both terms into (17), we complete the proof by

$$\frac{\|\mathbf{x}_* - \mathbf{x}\|_{\Lambda}^2}{\|\mathbf{x}_* - \mathbf{t}\|_{\Lambda}^2} = \frac{\|(\mathbf{I} - \mathbf{R}_{\mathcal{X}_1}) \mathbf{x}_*\|_{\Lambda}^2}{\|\mathbf{x}_* - \mathbf{t}\|_{\Lambda}^2} + \frac{\|\mathcal{X}_1 (\mathbf{x}_*^{\geq 2} - \mathbf{x}^{\geq 2})\|_{\Lambda}^2}{\|\mathbf{x}_* - \mathbf{t}\|_{\Lambda}^2} = \mu_1^2 \omega_1^2 + \mu_1^2 (1 - \omega_1^2) \Phi_{d-1}^2 = \Phi_d^2, \quad (19)$$

where Φ_{d-1} evaluates the convergence rate of the AMEn in $d - 1$ dimensions by (18). \square

Remark 2. From the recurrence relation (19) it is clear that if the convergence of the reduced problem $\phi_{d-1} < 1$, then the convergence rate of the full problem

$$\phi_d^2 = \mu_1^2 \omega_1^2 + \mu_1^2 (1 - \omega_1^2) \phi_{d-1}^2 < 1.$$

By Remark 1, we can always choose the approximation threshold ε to ensure $\phi_2 < 1$ for the AMEn algorithm in two dimensions. Therefore, we can always choose ε to provide $\phi_d < 1$, which guarantees the global convergence of Alg. 1.

Similarly to Thm. 1 we can estimate ω_k in (15) as follows,

$$\omega_k^2 = 1 - \frac{(\mathbf{c}_k, \mathbf{R}_{\mathcal{X}}^{(A_k)} \mathbf{c}_k)_{A_k}}{\|\mathbf{c}_k\|_{A_k}^2} \leq 1 - \frac{(\mathbf{c}_k, \mathbf{R}_{\mathcal{Z}}^{(A_k)} \mathbf{c}_k)_{A_k}}{\|\mathbf{c}_k\|_{A_k}^2} \leq 1 - \frac{(\mathbf{c}_k, \mathbf{R}_{\tilde{\mathcal{Z}}_k}^{(A_k)} \mathbf{c}_k)_{A_k}}{\|\mathbf{c}_k\|_{A_k}^2} = \omega_{\tilde{\mathcal{Z}}_k}^2,$$

where $\tilde{\mathcal{Z}}_k \approx \mathcal{Z}_k = A_k \mathbf{c}_k = \mathbf{y}_k - A_k \mathbf{u}_k$, $\tilde{\mathcal{Z}}_k = \tau(\mathbf{Z}^{(k)}, \mathbf{Z}^{\geq k+1})$ and $\mathcal{Z} = \mathcal{P}_1(\mathbf{Z}^{(k)}, \mathbf{Z}^{\geq k+1})$. In the right-hand side we see the convergence rate of the perturbed steepest descent method applied to the reduced problem with the matrix A_k . It is estimated [9, Thm 1] as follows

$$\omega_{\tilde{\mathcal{Z}}_k} = \omega_{\mathcal{Z}_k} + \mathcal{O}(\varepsilon), \quad \omega_{\mathcal{Z}_k} \leq \frac{\lambda_{\max}(A_k) - \lambda_{\min}(A_k)}{\lambda_{\max}(A_k) + \lambda_{\min}(A_k)} = \Omega(A_k) < 1,$$

where ε denotes the relative accuracy of $\tilde{\mathcal{Z}}_k$. It can be shown that if all $\mathcal{X}_{<k}$ are orthogonal we have

$$\begin{aligned} \lambda_{\min}(A_k) &\geq \lambda_{\min}(A_{k-1}), & \lambda_{\max}(A_k) &\leq \lambda_{\max}(A_{k-1}), \\ \Omega(A_{d-1}) &\leq \Omega(A_{d-2}) \leq \dots \leq \Omega(A_1) = \Omega(A) = \frac{\lambda_{\max}(A) - \lambda_{\min}(A)}{\lambda_{\max}(A) + \lambda_{\min}(A)}, \end{aligned} \quad (20)$$

where the last term estimates the convergence rate of SD algorithm applied to $A\mathbf{x} = \mathbf{y}$.

Remark 3. The requirement for $\mathcal{X}_{<k}$ to be orthogonal for $k = 1, \dots, d$ is equivalent to the *TT-orthogonality* of the tensor train \bar{X} , see [24] for details. As a counterpart of (20), we may say that this requirement prevents the condition numbers of reduced matrices in (14) from increasing, which is essential for numerical stability. By construction, $\mathcal{X}^{(k)} = [\mathbf{U}^{(k)} \quad \mathbf{Z}^{(k)}]$ does not provide the TT-orthogonality of \bar{X} . An additional step is required to *recover* the orthogonality, i.e. make the TT-core $\mathcal{X}^{(k)}$ column-orthogonal. It is done via a QR decomposition

$$\begin{bmatrix} \mathbf{U}^{(k)} & \mathbf{Z}^{(k)} \end{bmatrix} \begin{bmatrix} \mathbf{T}^{(k+1)} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{U}}^{(k)} & \hat{\mathbf{Z}}^{(k)} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{uu} & \mathbf{R}_{uz} \\ & \mathbf{R}_{zz} \end{bmatrix} \begin{bmatrix} \mathbf{T}^{(k+1)} \\ \mathbf{0} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{U}}^{(k)} & \hat{\mathbf{Z}}^{(k)} \end{bmatrix} \begin{bmatrix} \mathbf{R}_{uu} \mathbf{T}^{(k+1)} \\ \mathbf{0} \end{bmatrix}.$$

We denote the result after orthogonalization by the same symbol $\mathcal{X}^{(k)} = \hat{\mathcal{X}}^{(k)} = [\hat{\mathbf{U}}^{(k)} \quad \hat{\mathbf{Z}}^{(k)}]$. All considerations in Theorem 2 remain valid, since all estimates are based on the subspaces, which are unaffected by the QR decomposition. Therefore, we imply the TT-orthogonality silently to simplify the discussion, and the actual operation is fast and does not influence the analysis.

In [9, Thm. 3], the convergence rate of the *greedy descent* algorithm $\mathbf{t} + \text{ALS}(\mathbf{z})$ is given by exactly the same formula as (16), but the values ω_k are defined differently. In the AMEn method, ω_k is given by (15) and relates to the convergence of the reduced problem (14).

For the algorithm $t + \text{ALS}(z)$ it is defined as $\omega_k = \omega_{\mathcal{Z}_{\leq k}}$ in terms of (9). Since $\text{span } \mathcal{Z}_{\leq k+1} \in \text{span } \mathcal{Z}_{\leq k}$, for the greedy algorithm it holds $\omega_{k+1} \leq \omega_k$. For the AMEn algorithm, we can prove this only for upper bounds $\Omega(A_k)$ as shown in (20).

Considering the ‘width’ of $\mathcal{Z}_{\leq k}$ and $\mathcal{X}_{\leq k}$, we can expect that

$$\omega_k(\text{AMEn}) \ll \omega_k(t + \text{ALS}(z)), \quad \mu_k(\text{AMEn}) \leq \mu_k(t + \text{ALS}(z)),$$

which is observed in numerical experiments. It is not clear however whether this heuristic statement holds in general.

4 Fast approximation of the residual

In this section we discuss how to compute the approximation on step 3 in AMEn Alg. 1 efficiently.

4.1 SVD-based approximation

In steepest descent schemes proposed in [9], the low-rank approximation of the residual is computed once per iteration, and a standard SVD-based TT-rounding procedure from [24] can be used. In AMEn Alg. 1 we can not approximate each z_k individually by the TT-SVD, since it makes the total complexity quadratic in the dimension d . To keep the complexity linear in d , we have to investigate the tensor structure of z_k .

Looking at the TT representation of the reduced system (14) and recalling that $\mathcal{X}_{<k} = X^{<k} \otimes I_{n_k \dots n_d}$ has a rank-one structure, we conclude that $y_k = \mathcal{X}_{<k}^* y$ inherits the blocks $k+1, \dots, d$ from y as follows,

$$y_k(\overline{\alpha_{k-1} i_k}, \dots, i_d) = Y_k^{(k)}(\overline{\alpha_{k-1} i_k}) Y^{(k+1)}(i_{k+1}) \dots Y^{(d)}(i_d),$$

where $Y_k^{(k)} = \mathcal{P}_{<k}^*(X^{(1)}, \dots, X^{(k)}) Y^{\leq k}$. The similar representation holds for the local matrix $A_k = \tau(A_k^{(k)}, A^{(k+1)}, \dots, A^{(d)})$. Therefore the local residual $z_k = y_k - A_k u_k$ writes

$$z_k(\overline{\alpha_{k-1} i_k}, \dots, i_d) = \hat{Z}_k^{(k)}(\overline{\alpha_{k-1} i_k}) \cdot \hat{Z}^{(k+1)}(i_{k+1}) \dots \hat{Z}^{(d)}(i_d), \quad (21)$$

$$\hat{Z}_k^{(k)}(\overline{\alpha_{k-1} i_k}) = \left[Y_k^{(k)}(\overline{\alpha_{k-1} i_k}) - A_k^{(k)}(\overline{\alpha_{k-1} i_k}, \overline{\beta_{k-1} j_k}) \otimes U^{(k)}(\overline{\beta_{k-1} j_k}) \right],$$

$$\hat{Z}^{(p)}(i_p) = \left[Y^{(p)}(i_p) \quad A^{(p)}(i_p, j_p) \otimes T^{(p)}(j_p) \right], \quad p = 2, \dots, d-1,$$

$$\hat{Z}^{(d)}(i_d) = \left[Y^{(d)}(i_d) \quad A^{(d)}(i_d, j_d) \otimes T^{(d)}(j_d) \right].$$

The TT decomposition of the exact residual $z_k = \tau(\hat{Z}_k^{(k)}, \hat{Z}^{(k+1)}, \dots, \hat{Z}^{(d)})$ has only one block which actually depends on the information gained in the step k , the others can be precomputed before the iteration. In the approximate residual $\tilde{z}_k = \tau(Z_k^{(k)}, \dots, Z_k^{(d)})$ all blocks depend on the recently computed $U^{(k)}$, but only one block $Z_k^{(k)}$ is actually required. This means that if we keep all TT-cores $Z^{(p)}$ right-orthogonal, we can compute $Z_k^{(k)}$ by the SVD compression of $\hat{Z}_k^{(k)}$ only. Therefore, the SVD-based approximation of the residual involves the information from only one core, and the complexity of each enrichment step does not grow with d . The overall complexity is therefore linear in d , as required.

4.2 Cholesky-based approximation

The singular value decomposition provides the optimal approximation accuracy for a prescribed rank, but is numerically expensive. Each TT-core $\hat{Z}^{(k)}$ has the sizes $R_{k-1} \times n_k \times R_k$, where $R_k = r_k r_k(A) + r_k(y)$, and the QR and SVD operations have the complexity $\mathcal{O}(R^3) = \mathcal{O}(r^6)$, which may be inefficient. Since a very precise approximation of the residual is not always required, we may avoid expensive QR and SVD steps by considering the Unfinished Cholesky algorithm (see, e.g. [30]) applied to the Gram matrix of the first unfolding of z_k . A careful implementation allows to reduce the complexity to $\mathcal{O}(r^5)$.

Given (21), its first unfolding reads

$$Z_k^{(k)}(\overline{\alpha_{k-1} i_k}, \overline{i_{k+1}}, \dots, i_d) = \hat{Z}_k^{(k)}(\overline{\alpha_{k-1} i_k}) \cdot \hat{Z}^{(k+1)}(i_{k+1}) \dots \hat{Z}^{(d)}(i_d),$$

and the Gram matrix $G_k = Z_k^{(k)}(Z_k^{(k)})^*$ computes as follows,

$$G_k(\overline{\alpha_{k-1} i_k}, \overline{\beta_{k-1} j_k}) = \Gamma_k^{(k)}(\overline{\alpha_{k-1} i_k}, \overline{\beta_{k-1} j_k}) \cdot \Gamma^{(k+1)} \dots \Gamma^{(d)},$$

where $\Gamma_k^{(k)}(\overline{\alpha_{k-1} i_k}, \overline{\beta_{k-1} j_k}) = \hat{Z}_k^{(k)}(\overline{\alpha_{k-1} i_k}) \otimes \hat{Z}_k^{(k)}(\overline{\beta_{k-1} j_k})$, and $\Gamma^{(p)} = \hat{Z}^{(p)}(i_p) \otimes \hat{Z}^{(p)}(i_p)$ for $p = k+1, \dots, d$.

Similarly to the SVD-based method, we can precompute $\Gamma^{(p)}$ before the iteration. The enrichment vectors are then calculated as the factors $Z_k^{(k)}$ in the Unfinished Cholesky decomposition

$$G_k(\overline{\alpha_{k-1} i_k}, \overline{\beta_{k-1} j_k}) \approx Z_k^{(k)}(\overline{\alpha_{k-1} i_k}) D \left(Z_k^{(k)}(\overline{\beta_{k-1} j_k}) \right)^*.$$

4.3 ALS-based approximation

To reduce the complexity even further, we can approximate $\tilde{z} \approx z = y - Au$ using the auxiliary ALS iteration. We start from some low-rank initial guess $\tilde{z} = \tau(\bar{Z})$ and minimize $\|\tilde{z} - z\|$ under the constraint $\tilde{z} = \mathcal{Z}_{\neq k} z^{(k)}$, where $z^{(k)} = \text{vec } Z^{(k)}$. For a unitary $\mathcal{Z}_{\neq k}$ this leads to the extremal condition $z^{(k)} = \mathcal{Z}_{\neq k}^* z$.

Until the convergence of the fixed-rank ALS is not proved, this approach is heuristic. However, as was observed in numerical experiments, it provides the enrichment basis almost of the same quality as the SVD-based method, although much faster. It is enough to conduct two alternating methods simultaneously step by step, which means that only one ALS update is performed for \tilde{z} between the subsequent AMEn iterations.

The algorithm is organized as follows. Given some low-rank approximation \tilde{z} , we assume that $\text{span}(Z^{>k})$ is a good approximation basis for z_k as well (which appears to hold in practice). That is, the enrichment is computed as a projection

$$\text{vec } Z_k^{(k)} = \mathcal{P}_{\neq 1}^*(Z^{(k)}, \dots, Z^{(d)}) z_k. \quad (22)$$

Now, we need to update \tilde{z} for the forthcoming iterations. The current solution approximant is $u = \tau(X^{<k}, U^{(k)}, T^{>k})$, so the TT-core $Z^{(k)}$ writes

$$\begin{aligned} z^{(k)} &= \mathcal{Z}_{\neq k}^*(y - Au) = \mathcal{Z}_{\neq k}^* y - \mathcal{Z}_{\neq k}^* A \mathcal{P}_{\neq k}(\bar{U}) u^{(k)}, \\ \bar{U} &= (X^{(1)}, \dots, X^{(k-1)}, U^{(k)}, T^{(k+1)}, \dots, T^{(d)}). \end{aligned} \quad (23)$$

Note that $Z^{(k)}$ serves only as an update of the global residual approximation \tilde{z} and cannot be used as an enrichment directly.

Similarly to the previous sections, one may avoid $\mathcal{O}(d^2)$ cost of (23),(22) by performing all calculations involving the same TT blocks (e.g. $\hat{Z}^{(p)}$) only once during the AMEn iteration. The resulting complexity is therefore that of the fixed-rank ALS, $\mathcal{O}(\rho n^2 r^3)$, where $\rho = r(\tilde{z})$ is the TT-rank of \tilde{z} . In practice, it is usually enough to take $\rho \leq 5$.

Finally, let us note that the minimization $\|\tilde{z} - z\|$ is equivalent to the maximization of (\tilde{z}, z) . In other words, the AMEn method solves approximately the following *minimax* problem,

$$\max_{\tilde{z}} \min_x (\tilde{z}, y - Ax) = \max_{\tilde{z}} \min_x (\tilde{z}, x_* - x)_A,$$

by performing the subsequent ALS updates for \tilde{z} in the rank- ρ TT format, and x in the TT format with a *varying* rank r . This allows us to establish a connection between the AMEn method and the greedy approximations, in particular, the Minimax Proper Generalized Decomposition [22]. However, the greedy techniques usually perform the optimization over rank-1 separable tensors. As a some improvement one may mention the *orthogonal* greedy method, which orthogonalizes the residual to the basis of R current canonical factors of the solution (i.e. selects R scalars). The AMEn approach may be considered as a next milestone in the family of adaptive tensor-structured linear solvers. By updating a larger portion of solution data at a time, it appears to be more robust and accurate, as was demonstrated in [9].

5 Numerical experiments

In these experiments, we compare the MATLAB versions of AMEn algorithms proposed above (SVD, Chol, ALS) with the ALS($t + z$) method from the previous work [9], as well as the DMRG method from [8]. The AMEn and DMRG methods were implemented within the framework of the TT-Toolbox³ 2.2 (routines `amen_solve2` and `dmg_solve3`, respectively), and the computations were done at the Linux machine with 2.6 GHz AMD Opteron CPU, and MATLAB R2012a.

5.1 SPD example: Poisson equation

First, we consider the same symmetric positive definite example as in [9]. This is the high-dimensional Poisson equation,

$$-\Delta u(x) = e, \quad x \in \Omega = [0, 1]^d, \quad u|_{\partial\Omega} = 0,$$

where Δ is the finite difference Laplacian discretization on a uniform grid with 64 points in each direction, i.e., the total size of the system is 64^d , and e is the vector of all ones. Different greedy-type methods were compared in [9], as well as the ALS($t + z$) method. Now we focus on non-greedy techniques, including AMEn+SVD and AMEn+ALS, see Fig. 3. The TT-rank of the enrichment \tilde{z} is $\rho = 4$, Frobenius-norm threshold for the solution $\text{tol} = 10^{-5}$, and the problem dimension $d = 16$.

We see that all methods except the DMRG demonstrate comparable performances. Even though in the beginning of the iterations the ALS($t + z$) method seems to be the fastest, it approaches the same CPU times as the AMEn methods when the rank increases.

³<http://github.com/oseledets/TT-Toolbox>

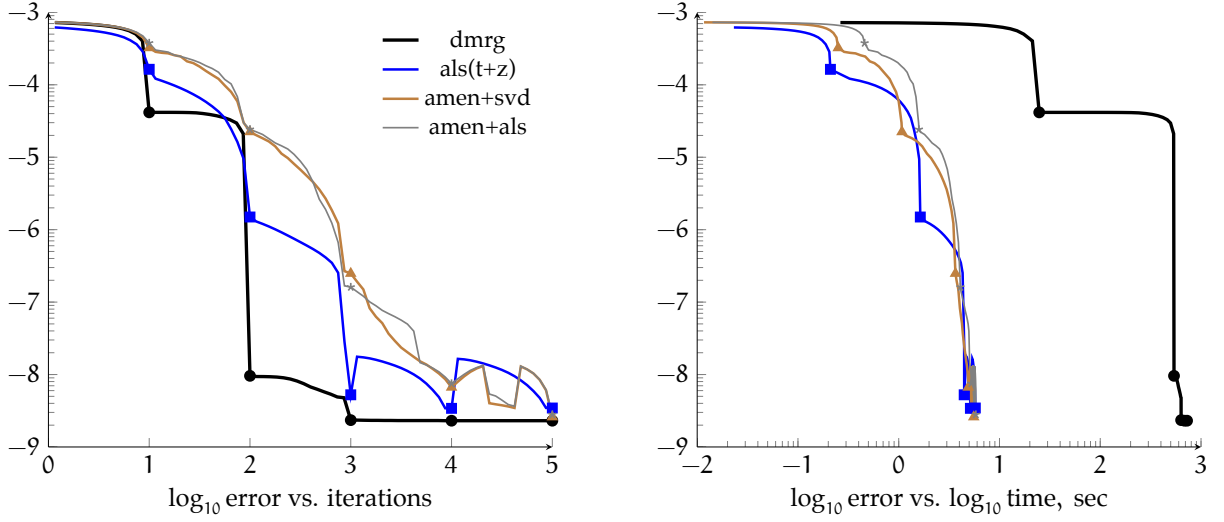


Figure 3: A -norm error in different methods versus iterations (left), and CPU time (right). Poisson equation, $d = 16$

In this example, the desired accuracy level is reached by all algorithms. However, it might be not the case, as we will see in the following.

5.2 Nonsymmetric example: Chemical Master Equation

The second example is the Chemical Master Equation [34], applied to the d -dimensional cascade gene regulatory model [13, 1]. This is the huge-sized ODE

$$\frac{d\psi(t)}{dt} = A\psi(t) \in \mathbb{R}^{n^d},$$

where $\psi(t) = \{\psi(\mathbf{i}, t)\}$, $\mathbf{i} = (i_1, \dots, i_d) \in [0, \dots, 63]^{\otimes d}$, so that $n = 64$, and the operator is formulated as follows,

$$\begin{aligned} A &= A_1 + \dots + A_d, \\ A_1\psi(\mathbf{i}, t) &= \alpha_0 \cdot (\psi(\mathbf{i} - \mathbf{e}_1, t) - \psi(\mathbf{i}, t)) + \delta \cdot ((i_1 + 1)\psi(\mathbf{i} + \mathbf{e}_1, t) - i_1\psi(\mathbf{i}, t)), \\ A_k\psi(\mathbf{i}, t) &= \frac{\beta i_{k-1}}{\beta i_{k-1} + \gamma} (\psi(\mathbf{i} - \mathbf{e}_k, t) - \psi(\mathbf{i}, t)) + \delta \cdot ((i_k + 1)\psi(\mathbf{i} + \mathbf{e}_k, t) - i_k\psi(\mathbf{i}, t)) \end{aligned}$$

for $k = 2, \dots, d$, where $\mathbf{e}_k \in \mathbb{R}^d$ is the k -th identity vector. The particular model parameters were fixed to the values

$$\alpha_0 = 0.7, \quad \delta = 0.07, \quad \beta = 1, \quad \gamma = 5.$$

The Chemical Master Equation serves as an accurate model for gene transcription, protein production and other biological processes. However, its straightforward solution becomes impossible rapidly with increasing number of species d . Existing techniques include the Monte-Carlo-type methods (so-called SSA [11] and its descendants), as well as more tensor-related ones: Sparse Grids [13], greedy approximations in the canonical tensor format [1] and tensor manifold dynamics [15]. The first two approaches only relax

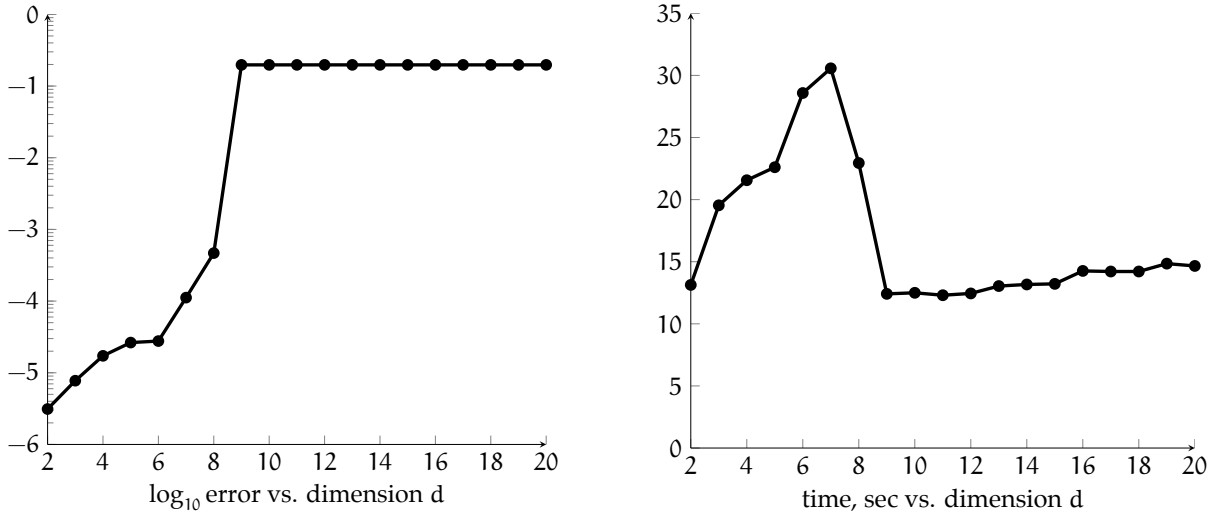


Figure 4: Error (left) and CPU time (right) of the DMRG method vs. the dimension

the curse of dimensionality to some extent; typical examples involve up to 10 dimensions and may take from 15 minutes to many hours on high-performance machines. Tensor-product low-rank approaches seem to be more promising. Unfortunately, we cannot estimate a possible potential of greedy or manifold dynamics methods, whereas up to now our alternating linear solution techniques appear to be more efficient. For more intensive study of the CME applications of the AMEn and DMRG methods see [6] and [16], respectively. Note that for systems with moderate dimensions and smaller time steps, the DMRG method can be of a good use for such problems, as was demonstrated in [16]. However, as we will see, the AMEn algorithm appears to perform better than DMRG for more complicated problems.

Two specific tricks allow to take more benefits from the tensor structuring. First, we employ the Crank-Nicolson discretization in time, but instead of the step-by-step propagation, consider the time as a $(d + 1)$ -th variable and formulate one global system encapsulating all time steps [7],

$$\left(I - \frac{\tau}{2}\mathcal{A}\right)\psi(t_{m+1}) = \left(I + \frac{\tau}{2}\mathcal{A}\right)\psi(t_m), \quad m = 0, \dots, N_t - 1,$$

where $t_m = \tau m$, τ is the time step size, and the initial state is $\psi(0) = \bigotimes_{k=1}^d e_1$, $e_1 \in \mathbb{R}^{64}$ is the first identity vector. In particular, we choose $N_t = 2^{12}$, $T = 10$, and $\tau = T/N_t$. Such a time interval is not enough to reach the stationary solution, but the transient process is also of interest. As a result, we end up with a $(d + 1)$ -dimensional system of size $n^d \cdot N_t$.

Second, we prepare all the initial data and seek the solution not in the $(d+1)$ -dimensional TT-format directly, but in the so-called Quantized TT format [17]: we reshape additionally all tensors to the sizes $2 \times 2 \times \dots \times 2$, and apply the $(d \log_2(n) + \log_2(N_t))$ -dimensional TT decomposition, but with each mode size reduced to 2.

However, the matrix is strongly nonsymmetric, which makes difficulties for the DMRG approach. We fix the truncation tolerance for the solution to $\text{tol} = 10^{-5}$, and track the Frobenius-norm error of the DMRG solution w.r.t. the reference one, obtained by the AMEn+SVD method with tolerance 10^{-8} , versus the dimension d , see Fig. 4. Since the

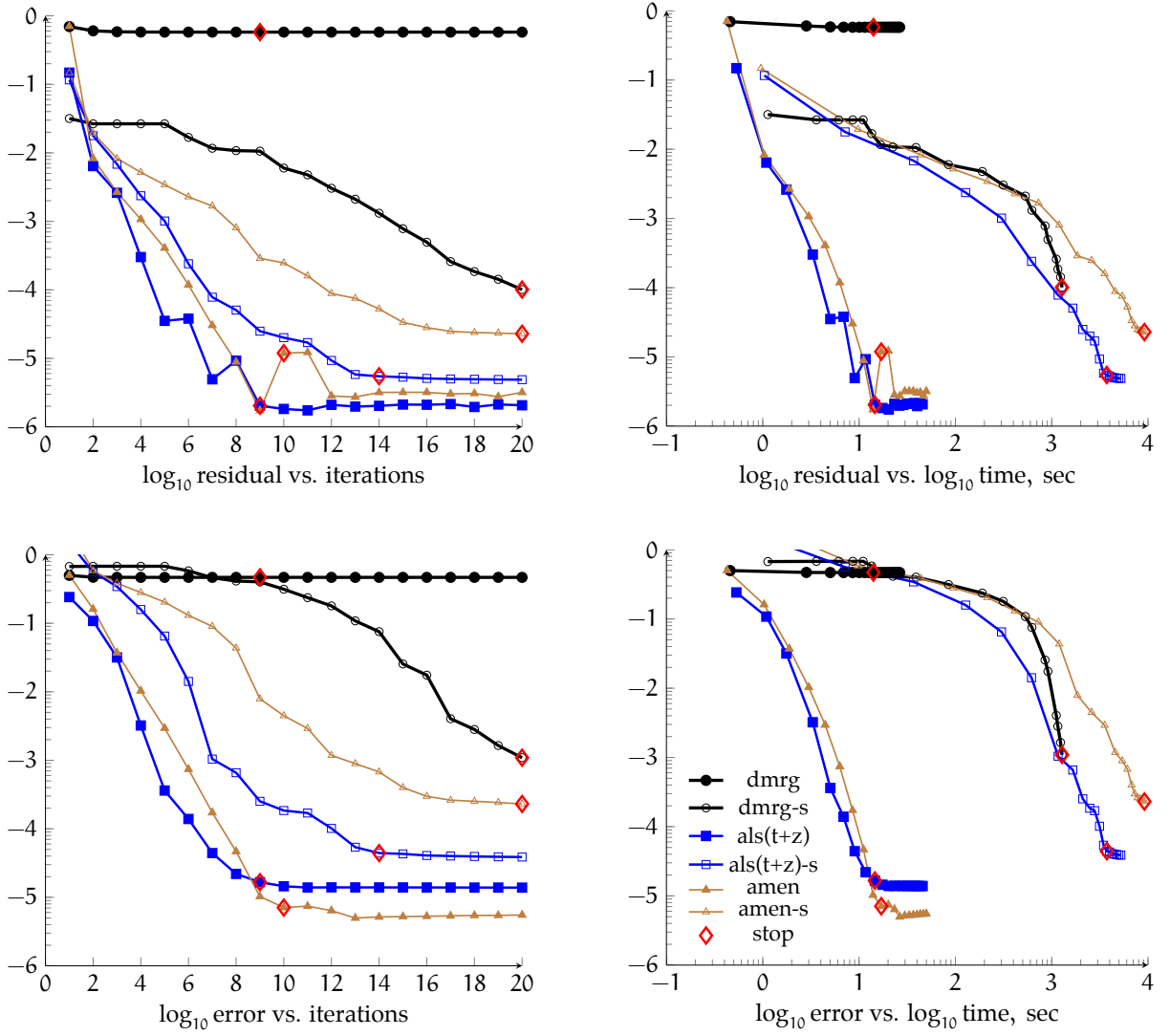


Figure 5: Residual (top) and error (bottom) in different methods vs. iterations (left), and CPU time (right), 20-dimensional problem

DMRG technique takes into account only local information on the system, its accuracy deteriorates rapidly with the increasing dimension. A stagnation in a local minimum is also reflected by a sharp drop of the CPU time, since the method skips the “converged” TT blocks. This makes the DMRG unreliable for high-dimensional problems, even if the QTT format allows to get rid of large mode sizes.

Now, we fix the dimension $d = 20$, and compare both the error and residual accuracies of all methods, as well as the computational times. In all cases, the Frobenius-norm tolerance was set to $\tau_{01} = 10^{-5}$, and the enrichment rank to $\rho = 4$.

First of all, since our methods are proven to converge in the SPD case, we shall examine both the initial and symmetrized systems (Fig. 5). A well-known way to treat a general problem via a symmetric method is the normal, or symmetrized formulation, $A^*Ax = A^*y$. However, both the condition number and the TT ranks of A^*A are the squared ones of A , and this approach should be avoided when possible.

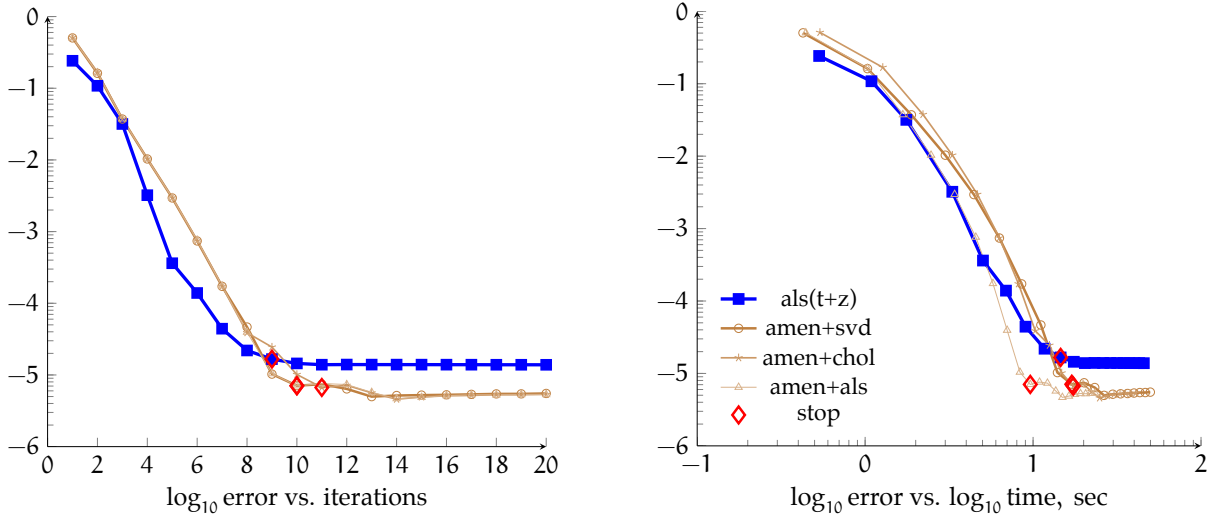


Figure 6: Error in different methods vs. iterations (left), and CPU time (right), 20-dimensional problem

Three particular techniques are considered: the DMRG method, the AMEn+SVD (marked as “amen” in Fig. 5) and the ALS($t+z$) one. The symmetrized versions are denoted by the “-s” tag.

In addition, note that the convergence of the methods may be checked locally due to the zero total correction after the enrichment in Alg. 1: before recomputing the $(k+1)$ -th block, calculate the local residual provided by the previous solution $T^{(k+1)}$. If it is below the threshold τ_{01} for all k , the method may be considered as converged, and stopped. Occurrences of this fact are marked by red rectangles (“stop”).

We observe that the symmetrization allows the DMRG method to converge at least to the accuracy 10^{-3} , but increases the CPU time by a factor greater than 100 due to the squaring of the TT ranks and condition number of the matrix. Contrarily, for the AMEn and ALS($t+z$) methods the symmetrization is completely inefficient and redundant: despite pessimistic theoretical estimates, the nonsymmetric algorithms converge rapidly to an accurate solution approximation.

Though the non-symmetrized methods may admit oscillations in the residual, the Frobenius-norm error threshold is almost satisfied in both AMEn and ALS($t+z$) methods. Nevertheless, the AMEn algorithm appears to be more accurate thanks to the enrichment update in each step. Also, its local stopping criterion is trustful: it fires just after the real error becomes smaller than the tolerance, which is not the case for other methods.

Since both AMEn-type methods in this test exploit the SVD-based residual approximation, they demonstrate almost the same CPU times. However, using the additional techniques from Section 4 we can reduce the complexity while maintaining almost the same accuracy, see Fig. 6. While the AMEn+Chol method still operates with the exact residual, the AMEn+ALS only needs to compute scalar products of the true residual and its low-rank approximation, which makes it more efficient than the AMEn+SVD method, as well as the ALS($t+z$) one.

Finally, we test the performance of the two AMEn realizations with respect to the enrichment rank ρ (TT-rank of \tilde{z}), see Fig. 7. As expected, the higher ρ is, the more accurate

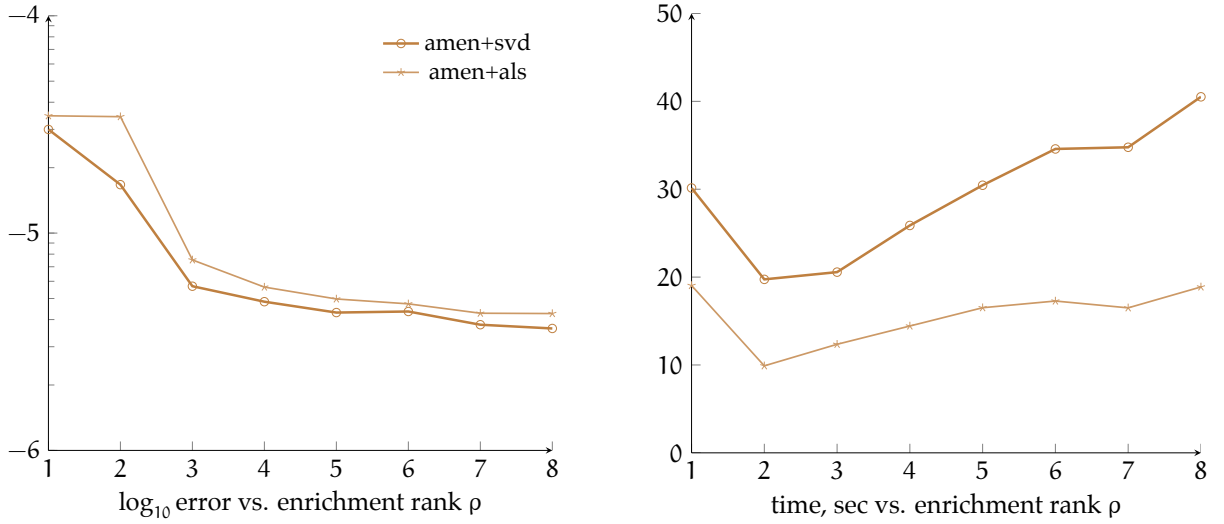


Figure 7: Error (left) and CPU time (right) in AMEn methods vs. enrichment rank, 20-dimensional problem

solution can be computed. On the other hand, it is not necessary to pick very large ranks, since the corresponding accuracy improvement does not overcome the significant increase in CPU time.

5.3 Fokker-Planck equation for complex fluid dynamics

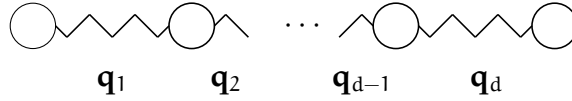


Figure 8: Bead-spring model of a polymer in a fluid

Another example of high-dimensional problems arising in the context of probability distribution modeling, is the Fokker-Planck equation (see e.g. [27]). As a particular application, consider the 8-dimensional Fokker-Planck equation of the polymer micro-model arising in the non-Newtonian fluid dynamics [4, 21, 2]. The polymer molecules in a solution are subject to the Brownian motion, and are often modeled as bead-spring chains (see Fig. 8). The spring extensions, being the degrees of freedom of the dynamical system, become the coordinates in the Fokker-Planck equation.

We consider the case of 4 two-dimensional *finitely extensible nonlinear elastic* (FENE) springs in the shear flow regime according to [2],

$$\frac{\partial \psi(\mathbf{q}, t)}{\partial t} = \sum_{i=1}^d \nabla_{\mathbf{q}_i} \cdot \left(\mathbf{K} \mathbf{q} \psi(\mathbf{q}, t) - \sum_{j=1}^d D_{ij} (\mathbf{F}_j(\mathbf{q}) \psi(\mathbf{q}, t) - \nabla_{\mathbf{q}_j} \psi(\mathbf{q}, t)) \right), \quad (24)$$

where $\mathbf{q} = (\mathbf{q}_1, \dots, \mathbf{q}_8) = (q_{1,1}, q_{1,2}, \dots, q_{4,2})$ is the stacked spring extension vectors ($q_{p,k}$ is

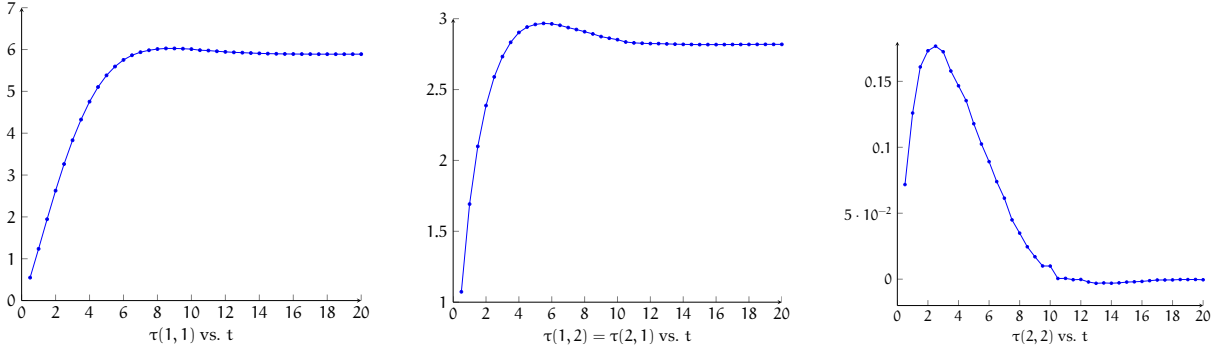


Figure 9: Components of the stress tensor vs. time

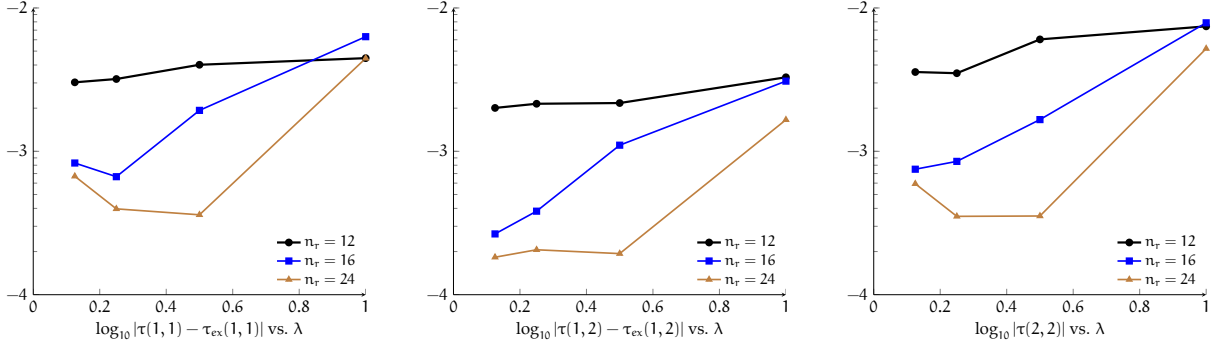


Figure 10: Accuracy of the stress tensor vs. time step and grid size

the displacement of the p -th spring in the k -th direction),

$$\mathbf{D} = \frac{1}{4} \text{tridiag}\{-1, 2, -1\} \otimes \mathbf{I}_2 = \frac{1}{4} \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & -1 & 2 & -1 \\ & & -1 & 2 \end{bmatrix} \otimes \begin{bmatrix} 1 & \\ & 1 \end{bmatrix}$$

is a spring interaction tensor,

$$\mathbf{K} = \mathbf{I}_4 \otimes (\nabla_x \mathbf{u}) = \text{diag}\{1, 1, 1, 1\} \otimes \begin{bmatrix} 0 & 0.8 \\ 0 & 0 \end{bmatrix}$$

is a flow velocity gradient (shear flow case), and

$$\mathbf{F}_j(\mathbf{q}) = \frac{\mathbf{q}_j}{1 - |\mathbf{q}_j|^2/b}, \quad |\mathbf{q}_j|^2 = q_{j,1}^2 + q_{j,2}^2, \quad b = 5,$$

is the FENE spring force. Note that the singularity in \mathbf{F}_j limits the maximal length of a spring to \sqrt{b} . Moreover, the probability density ψ at the point $|\mathbf{q}_j| = \sqrt{b}$ (and any with larger modulus) is zero. Therefore, the domain shrinks to the product of balls $\mathbb{B} = \mathbb{B}_{\sqrt{b}}^{\otimes d}$.

A quantity of interest is the average polymeric contribution to the stress tensor,

$$\tau(t) = \sum_{i=1}^d \langle \mathbf{q}_i \mathbf{F}_i(\mathbf{q})^\top - \mathbf{I}_2 \rangle = \sum_{i=1}^d \int_{\mathbb{B}} \psi(\mathbf{q}, t) \mathbf{q}_i \mathbf{F}_i(\mathbf{q})^\top d\mathbf{q} - d \cdot \mathbf{I}_2, \quad (25)$$

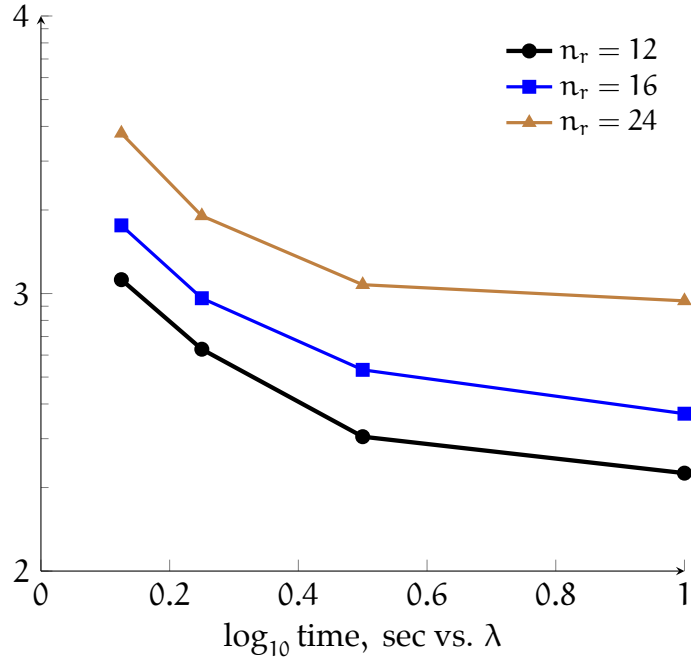


Figure 11: CPU time vs. time step and grid size

with the normalization assumption $\int \psi(\mathbf{q}, t) d\mathbf{q} = 1$.

To recast the problem domain into a hypercube, the polar coordinates are employed, $\mathbf{q} \rightarrow (r_1, \theta_1, \dots, r_4, \theta_4) \in ([0, \sqrt{b}) \otimes [0, 2\pi])^{\otimes 4}$. The discretization is done via the spectral elements method (see e.g. [33]). We will vary the number of spectral elements in each radial direction n_r , but the number of angular elements (in θ_i) is fixed to $2n_r$. With typical values $n_r \sim 20$, we end up with tensors of size $\sim 10^{12}$ and dense populated matrices, which are intractable in the full format. Since the spectral differentiation matrices are found to be incompressible in the QTT format, the 8-dimensional TT representation is used.

We would like to compute the stationary state of (24), so we use the simple implicit Euler (inverse power) method as the time discretization,

$$(M + \lambda A)\psi(t_{m+1}) = M\psi(t_m), \quad t_m = \lambda m, \quad m = 0, \dots, N_t - 1,$$

where M is the mass matrix, A is the stiffness matrix. The time integration was performed until $T = \lambda N_t = 20$, which is enough to approximate the steady state with a satisfactory accuracy, and the (unnormalized) initial state was chosen $\psi(\mathbf{q}, 0) = \bigotimes_{i=1}^4 (1 - |\mathbf{q}_i|^2/b)^{b/2}$, which corresponds to the zero velocity gradient $\mathbf{K} = 0$. Since λ is not a “time step” but a parameter of the inverse power method, we will check the performance w.r.t. λ as well.

In the previous example we have observed that the AMEn+SVD method is in fact superfluous, since the AMEn+ALS method delivers the same accuracy with lower cost. Both mode sizes (up to 48) and TT-ranks (up to 73) in this example are relatively large, so we will consider only the AMEn+ALS. We set the Frobenius-norm threshold to $\text{tol} = 10^{-4}$, and the enrichment rank $\rho = 3$. The initial guess for the AMEn+ALS method is taken from the previous Euler step.

First, let us track the evolution of the stress tensor components (25) versus Euler iterations, see Fig. 9. We see that the stress does really stabilize in the chosen time range.

Moreover, the last component tends to zero, and can therefore be used as an in-hand measure of the accuracy. In addition, we compare $\tau(1, 1)$ and $\tau(1, 2)$ with the reference values computed with $n_r = 28$ and $\tau_{01} = 10^{-5}$, see Fig. 10. For all λ except 1 (which is too large), and $n_r = 24$, the accuracy attained is of the order $10^{-4} \div 10^{-3}$. Note that typical accuracies of greedy or MC methods for many-spring models are of the order 10^{-1} [2, 35].

Finally, the computational times can be seen in Fig. 11. As expected, the complexity increases quadratically with the number of spectral elements n_r . An interesting feature is that the total CPU time decays with increasing λ . It points out that the performance of the AMEn method depends weakly on λ , and henceforth on the matrix spectrum. On the contrary, the quality of the initial guess (in terms of both ranks and accuracy) is crucial. This may motivate attempts to relate the AMEn methods to Newton or Krylov iterations in a future research.

6 Conclusion

In this paper we develop a new version of the fast rank-adaptive solver for tensor-structured symmetric positive definite linear systems in higher dimensions. Similarly to the algorithms from [9], the proposed AMEn method combines the one-dimensional local updates with the steps where the basis is expanded using the information about the global residual of the high-dimensional problem. However, in AMEn the same steps are ordered in such a way that only one or two neighboring cores are modified at once. Both methods from [9] and the AMEn converge globally, and the convergence rate is established w.r.t. the one of the steepest descent algorithm. The practical convergence in the numerical experiments is significantly faster than the theoretical estimate. The AMEn algorithm appears to be more accurate in practical computations than the previously known methods, especially if local problems are solved roughly.

The asymptotic complexity of the AMEn is linear in the dimension and mode size, similarly to the algorithms from [9]. The complexity w.r.t. the rank parameter is sufficiently improved taking into the account that a limiting step is the approximation of the residual, where the high accuracy is not always essential for the convergence of the whole method. We propose several cheaper alternatives to the SVD-based TT-approximation, namely the Cholesky decomposition and the inner ALS algorithm. The ALS approach provides a significant speedup, while maintaining almost the same convergence of the algorithm.

Finally, we apply the developed AMEn algorithm to general (non-SPD) systems, which arise from high-dimensional Fokker-Planck and chemical master equations. Theoretical convergence analysis can be made similarly to the FOM method, which is rather pessimistic and puts very strong requirements on the matrix spectrum. In numerical experiments we observe a surprisingly fast convergence, even for strongly non-symmetric systems. Here the AMEn demonstrates a significant advantage over the DMRG technique, which is known to stagnate, especially in high dimensions, see [23, 5] and Fig. 5.

There are many directions of a further research based on the ideas of [9] and this paper.

First, the ideas developed in this paper can be generalized to other problems, e.g. finding the ground state of a many-body quantum system or a particular state close to a prescribed energy. The combination of update and basis enrichment steps looks very promising for a wide class of problems, as soon as the corresponding classical iterative algorithms can be adapted to provide a proper basis expansion in higher dimensions. A huge work

is done in the community of greedy approximation methods, where the cornerstone is a subsequent rank-one update of the solution.

Second, there is a certain mismatch between the theoretical convergence estimates, which are at the level of the one-step steepest descent algorithm, and the practical convergence pattern, which looks more like the one of the GMRES. This indicates that there are further possibilities to improve our understanding of the convergence of the AMEn and similar methods. Our rates can benefit from sharp estimates of the progress of the one-dimensional update steps, which at the moment are available only in a small vicinity of a true solution, which is hard to satisfy in practice, see [28]. The superlinear convergence observed in numerical experiments inspires us to look for possible connections with the theory of Krylov-type iterative methods and a family of Newton methods.

Finally, we look forward to solving more high-dimensional problems, and are sure that they will bring new understanding of the advantages and drawbacks of the proposed method, and new questions and directions for a future research.

References

- [1] A. Ammar, E. Cueto, and F. Chinesta, *Reduction of the chemical master equation for gene regulatory networks using proper generalized decompositions*, Int. J. Numer. Meth. Biomed. Engng, 00 (2011), pp. 1–15.
- [2] A. Ammar, B. Mokdad, F. Chinesta, and R. Keunings, *A new family of solvers for some classes of multidimensional partial differential equations encountered in kinetic theory modeling of complex fluids*, Journal of Non-Newtonian Fluid Mechanics, 139 (2006), pp. 153 – 176.
- [3] Hans-Joachim Bungartz and Michael Griebel, *Sparse grids*, Acta Numerica, 13 (2004), pp. 147–269.
- [4] Cédric Chauvière and Alexei Lozinski, *Simulation of dilute polymer solutions using a Fokker-Planck equation*, Computers & Fluids, 33 (2004), pp. 687–696.
- [5] 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.
- [6] 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.
- [7] S. V. Dolgov, Boris N. Khoromskij, and Ivan V. Oseledets, *Fast solution of multi-dimensional parabolic problems in the tensor train/quantized tensor train-format with initial application to the Fokker-Planck equation*, SIAM J. Sci. Comput., 34 (2012), p. A3016–A3038.
- [8] 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.
- [9] S. V. Dolgov and D. V. Savostyanov, *Alternating minimal energy methods for linear systems in higher dimensions. Part I: SPD systems*, arXiv preprint 1301.6068, 2013.

- [10] M. Fannes, B. Nachtergaele, and R.F. Werner, *Finitely correlated states on quantum spin chains*, *Communications in Mathematical Physics*, 144 (1992), pp. 443–490.
- [11] D.T. Gillespie, *A general method for numerically simulating the stochastic time evolution of coupled chemical reactions*, *Journal of computational physics*, 22 (1976), pp. 403–434.
- [12] W. Hackbusch, *Tensor spaces and numerical tensor calculus*, Springer–Verlag, Berlin, 2012.
- [13] Markus Hegland, Conrad Burden, Lucia Santoso, Shev MacNamara, and Hilary Booth, *A solver for the stochastic master equation applied to gene regulatory networks*, *Journal of Computational and Applied Mathematics*, 205 (2007), pp. 708 – 724.
- [14] 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.
- [15] Tobias Jahnke and Wilhelm Huisinga, *A dynamical low-rank approach to the chemical master equation*, *Bulletin of Mathematical Biology*, 70 (2008), pp. 2283–2302.
- [16] V. Kazeev, M. Khammash, M. Nip, and C. Schwab, *Direct solution of the chemical master equation using quantized tensor trains*, Research Report 04, SAM, ETH Zürich, 2013.
- [17] B. N. Khoromskij, $\mathcal{O}(d \log n)$ –Quantics approximation of N – d tensors in high-dimensional numerical modeling, *Constr. Appr.*, 34 (2011), pp. 257–280.
- [18] ———, *Tensor-structured numerical methods in scientific computing: Survey on recent advances*, *Chemometr. Intell. Lab. Syst.*, 110 (2012), pp. 1–19.
- [19] A. Klümper, A. Schadschneider, and J. Zittartz, *Matrix product ground states for one-dimensional spin-1 quantum antiferromagnets*, *Europhys. Lett.*, 24 (1993), pp. 293–297.
- [20] T. G. Kolda and B. W. Bader, *Tensor decompositions and applications*, *SIAM Review*, 51 (2009), pp. 455–500.
- [21] A. Lozinski and C. Chauvière, *A fast solver for Fokker-Planck equation applied to viscoelastic flows calculations: 2D FENE model*, *Journal of Computational Physics*, 189 (2003), pp. 607 – 625.
- [22] A. Nouy, *A priori model reduction through proper generalized decomposition for solving time-dependent partial differential equations*, *Computer Methods in Applied Mechanics and Engineering*, 199 (2010), pp. 1603–1626.
- [23] I. V. Oseledets, *DMRG approach to fast linear algebra in the TT-format*, *Comput. Meth. Appl. Math*, 11 (2011), pp. 382–393.
- [24] ———, *Tensor-train decomposition*, *SIAM J. Sci. Comput.*, 33 (2011), pp. 2295–2317.
- [25] I. V. Oseledets and E. E. Tyrtyshnikov, *TT-cross approximation for multidimensional arrays*, *Linear Algebra Appl.*, 432 (2010), pp. 70–88.
- [26] S. Östlund and S. Rommer, *Thermodynamic limit of density matrix renormalization*, *Phys. Rev. Lett.*, 75 (1995), pp. 3537–3540.

- [27] Hannes Risken, *The Fokker-Planck Equation: Methods of Solutions and Applications*, 2nd ed., Springer Verlag, Berlin, Heidelberg, 1989.
- [28] 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.
- [29] Y. Saad, *Iterative methods for sparse linear systems*, SIAM, 2003.
- [30] D. V. Savostyanov, *Fast revealing of mode ranks of tensor in canonical format*, Numer. Math. Theor. Meth. Appl., 2 (2009), pp. 439–444.
- [31] 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 Multi-dimensional Systems (nDS), IEEE, 2011.
- [32] 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.
- [33] Lloyd N. Trefethen, *Spectral methods in MATLAB*, SIAM, Philadelphia, 2000.
- [34] N. G. van Kampen, *Stochastic processes in physics and chemistry*, North Holland, Amsterdam, 1981.
- [35] G. Venkiteswaran and M. Junk, *A QMC approach for high dimensional Fokker-Planck equations modelling polymeric liquids*, Math. Comput. Simul., 68 (2005), pp. 43–56.
- [36] S. R. White, *Density-matrix algorithms for quantum renormalization groups*, Phys. Rev. B, 48 (1993), pp. 10345–10356.

A FOM theory

As was observed in the numerical experiments, the AMEn method works successfully even being applied directly to non-symmetric systems. Though we cannot support this behavior with sharp estimates, one may proceed similarly to Section 3, and establish a formal theory, relating the AMEn to the Full Orthogonalization Method.

A.1 Galerkin projection and angles between subspaces

Like in the SPD case, we begin the analysis from the two-dimensional case. Given a linear system $Ax = y$ and some basis V , the projection method is performed as follows,

$$x = Vw = V(V^*AV)^{-1}V^*y, \quad y - Ax = (I - AV(V^*AV)^{-1}V^*)y. \quad (26)$$

Given an initial guess t , we assume $t \in \text{span}(V)$, and $z = y - At \in \text{span}(V)$. Then it holds also $y - Ax = (I - AV(V^*AV)^{-1}V^*)z$.

So, (26) performs an oblique projection of the residual. Its analysis is often conducted with the help of the orthogonal projection,

$$((AV)^*AV)w = (AV)^*z, \quad y - Ax = (I - AV(V^*A^*AV)^{-1}(AV)^*)z = (I - P_{AV})z,$$

i.e. the residual minimization on V . The case $V = z$ is known as the MINRES method. Its convergence was analysed in e.g. [29],

$$\|y - Ax\| = \sin(\widehat{Az, z})\|z\|, \quad \cos(\widehat{Az, z}) = \frac{|(Az, z)|}{\|Az\|\|z\|},$$

i.e. $\widehat{Az, z}$ is the acute angle between Az and z . The worst convergence rate is estimated as

$$\omega_{\text{MR}} = \max_z \sin(\widehat{Az, z}), \quad \sqrt{1 - \omega_{\text{MR}}^2} = \min_{z \neq 0} \frac{|(Az, z)|}{\|Az\|\|z\|},$$

and for a positive definite matrix is guaranteed to be less than 1. The same approach may be used for the block case as well,

$$\|y - Ax\| = \sin(\widehat{AV, z})\|z\|, \quad \cos(\widehat{AV, z}) = \max_{q \neq 0} \frac{|(AVq, z)|}{\|AVq\|\|z\|}.$$

Obviously, if $z \in \text{span}(V)$, it holds $\sin(\widehat{AV, z}) \leq \sin(\widehat{Az, z}) < 1$.

Unfortunately, for the oblique projection (26) one cannot guarantee the monotonous convergence in general. However, assuming a certain well-conditioning of the system, we may relate the old and new residuals by a factor smaller than 1 as well.

Lemma 1. Given a column-orthogonal matrix V , initial guess $t \in \text{span}(V)$. Assume the smallest eigenvalue $\mu = \lambda_{\min}(V^*AV + V^*A^*V)/2 > 0$, and $\|z - VV^*z\| \leq \varepsilon\|z\|$. Denote $\frac{\mu}{\|AV\|} = \sqrt{1 - \omega_V^2}$. Then, the progress of (26) is bounded by

$$\|y - Ax\| \leq \left(\varepsilon + \frac{\omega_V}{\sqrt{1 - \omega_V^2}} \sqrt{1 - \varepsilon^2} \right) \|z\|.$$

Proof. First of all, notice that $y - Ax$ is orthogonal to V ,

$$\begin{aligned} y - Ax &= (VV^*)(I - AV(V^*AV)^{-1}V^*)z + (I - VV^*)(I - AV(V^*AV)^{-1}V^*)z \\ &= (I - VV^*)z - (I - VV^*)AV(V^*AV)^{-1}V^*z. \end{aligned}$$

Then, $\|y - Ax\| \leq \varepsilon\|z\| + \sin(\widehat{V, AVw})\|AVw\|$, where $w = (V^*AV)^{-1}V^*z$. For the angle we can derive the following chain of inequalities,

$$\begin{aligned} \cos(\widehat{V, AVw}) &= \max_{\|q\|=1} \frac{|(Vq, AVw)|}{\|AVw\|} \geq \min_{w \neq 0} \frac{\|V^*AVw\|}{\|AVw\|} \\ &\geq \min_{\|w\|=1} \frac{\|w^*V^*AVw\|}{\|AVw\|} \geq \min_{\|w\|=1} \frac{\|w^*V^*AVw\|}{\|AV\|}, \end{aligned} \quad (27)$$

from which we get $\sin(\widehat{V, AVw}) \leq \omega_V$. On the other hand,

$$\min_{\|w\|=1} \frac{\|w^*V^*AVw\|}{\|AV\|} = \frac{\lambda_{\min}((V^*AV + V^*A^*V)/2)}{\|AV\|} \leq \frac{\sigma_{\min}(V^*AV)}{\|AV\|},$$

so that $\frac{\|AV\|}{\sigma_{\min}(V^*AV)} \leq \frac{1}{\sqrt{1-\omega_V^2}}$. Therefore, the residual estimates as follows,

$$\sin(\widehat{V, AVw})\|AVw\| \leq \omega_V\|AV\| \frac{1}{\sigma_{\min}(V^*AV)} \|V^*z\| \leq \frac{\omega_V}{\sqrt{1-\omega_V^2}} \sqrt{1-\varepsilon^2}\|z\|.$$

□

Remark 4. It holds

$$\sqrt{1-\omega_{MR}^2} = \min_{\|z\|=1} \frac{|z^*Az|}{\|Az\|} \leq \min_{\|w\|=1} \frac{\|w^*V^*AVw\|}{\|AVw\|} \leq \cos(\widehat{V, AVw}),$$

since the minimization over Vw is a restriction w.r.t. the minimization over z in the full space. Hence, $\sin(\widehat{V, AVw}) \leq \omega_{MR}$. However, $\sin(\widehat{V, AVw})/\sqrt{1-\omega_V^2} \geq \tan(\widehat{V, AVw})$ might be greater than ω_{MR} , and even greater than 1.

Remark 5. If V contains the m -th Krylov subspace, we obtain the so-called FOM method. The progress of the FOM can be related to that of the GMRES as follows [29],

$$\omega_m^F = \frac{\omega_m^G}{\sqrt{1 - (\omega_m^G/\omega_{m-1}^G)^2}},$$

where ω_k^G, ω_k^F are the progresses of the k -step GMRES and FOM, resp. Note the similar term $\omega_V/\sqrt{1-\omega_V^2}$ in Lemma 1.

Remark 6. The condition $\|z - VV^*z\| \leq \varepsilon\|z\|$ may reflect the residual approximation, i.e. $z \approx \tilde{z} \in \text{span}(V)$, but $z \notin \text{span}(V)$. Both SVD- and ALS-based approximations (see Section 4) fit to this scheme: the SVD approximation reads $\tilde{z} = \tilde{U}\tilde{U}^*z$, where \tilde{U} is the singular vectors, and the ALS approximation reads $\tilde{z} = \tilde{z}_{\neq k}\tilde{z}_{\neq k}^*$.

A.2 Recurrent residual accumulation

Lemma 1 applies immediately to the two-dimensional AMEn method, by setting $V = \mathcal{X}_1$. Despite the generally pessimistic estimate, it occurs in practice that $\mathcal{X}_1^* A \mathcal{X}_1$ is nonsingular, and moreover, $\omega_{\mathcal{X}_1}$ is rather small such that $\|y - Ax\| < \|z\|$ and converges rapidly.

A nice property of Theorem 2 is that it itself does not rely on a particular form of \mathcal{X}_k . We only needed that the Galerkin conditions $A_{k+1} x^{\geq k} = y_{k+1}$ make the error $\|x_{\star}^{\geq k} - x^{\geq k}\|_{A_k}$ strictly smaller than $\|x_{\star}^{\geq k} - u^{\geq k}\|_{A_k}$.

Here, we write the similar result in terms of residuals.

Lemma 2. Suppose in the k -th step of the multidimensional AMEn method, the ALS step provides the residual decrease

$$\|y_k - A_k u^{\geq k}\| = \mu_k \|y_k - A_k t^{\geq k}\|, \quad \mu_k \leq 1,$$

and the exact computation of the rest cores $x_{\star}^{\geq k} = A_{k+1}^{-1} y_{k+1}$ after the enrichment provides the residual decrease

$$\|y_k - A_k x^{\geq k}\| = \omega_k \|y_k - A_k u^{\geq k}\|, \quad \omega_k < 1,$$

where $\omega_k \leq \omega_{\mathcal{X}_k}$ from Lemma 1 with $\mathcal{X}_k = \mathcal{P}_1(X^{(k)}, \dots, X^{(d)})$. Then, the total convergence rate of the AMEn method is bounded by

$$\|y - Ax\| \leq \sum_{k=1}^{d-1} \omega_k \mu_k \prod_{m=1}^{k-1} \frac{\mu_m}{\sqrt{1 - \omega_m^2}} \cdot \|y - At\|.$$

Proof. As previously, we assume that $X^{(d)}$ is computed exactly. Then,

$$\|y_{d-1} - A_{d-1} x^{\geq d-1}\| = \omega_{d-1} \|y_{d-1} - A_{d-1} u^{\geq d-1}\|.$$

The base of the recursion is proved.

Suppose the theorem holds for $A_2 x^{\geq 2} = y_2$, i.e.

$$\|y_2 - A_2 x^{\geq 2}\| \leq \sum_{k=2}^{d-1} \omega_k \mu_k \prod_{m=2}^{k-1} \frac{\mu_m}{\sqrt{1 - \omega_m^2}} \cdot \|y_2 - A_2 t^{\geq 2}\| = \Omega \|y_2 - A_2 t^{\geq 2}\|, \quad (28)$$

and write the total progress for the whole system. We have

$$y - A\tau(X^{(1)}, X^{\geq 2}) = y - A\tau(X^{(1)}, X_{\star}^{\geq 2}) + A\tau(X^{(1)}, X_{\star}^{\geq 2}) - A\tau(X^{(1)}, X^{\geq 2}). \quad (29)$$

The exact solution for the second block is the oblique projection (26), hence

$$y - A\tau(X^{(1)}, X_{\star}^{\geq 2}) = (I - A\mathcal{X}_1(\mathcal{X}_1^* A \mathcal{X}_1)^{-1} \mathcal{X}_1^*) y.$$

The last two terms in (29) are similar to that in Theorem 2,

$$A\tau(X^{(1)}, X_{\star}^{\geq 2}) - A\tau(X^{(1)}, X^{\geq 2}) = A\mathcal{X}_1(x_{\star}^{\geq 2} - x^{\geq 2}),$$

but now it is not orthogonal to $(I - A\mathcal{X}_1(\mathcal{X}_1^* A \mathcal{X}_1)^{-1} \mathcal{X}_1^*)$. Therefore, we can only use the triangle inequality,

$$\|y - A\tau(X^{(1)}, X^{\geq 2})\| \leq \|(I - A\mathcal{X}_1(\mathcal{X}_1^* A \mathcal{X}_1)^{-1} \mathcal{X}_1^*) y\| + \|A\mathcal{X}_1(x_{\star}^{\geq 2} - x^{\geq 2})\|.$$

The first term is the residual after the Galerkin solution, which is bounded by $\omega_1 \|y - Au\|$. For the second term, we have the recursion assumption (28), that is

$$\|\mathcal{X}_1^* A \mathcal{X}_1 (x_*^{\geq 2} - x^{\geq 2})\| \leq \Omega \|\mathcal{X}_1^* A \mathcal{X}_1 (x_*^{\geq 2} - t^{\geq 2})\|.$$

However, the only way to relate $\|Aw\|$ and $\|\mathcal{X}_1^* Aw\|$ is to use the angle between \mathcal{X}_1 and $A\mathcal{X}_1$, employing (27),

$$\begin{aligned} \|\mathcal{X}_1^* A \mathcal{X}_1 (x_*^{\geq 2} - x^{\geq 2})\| &\geq \sqrt{1 - \omega_1^2} \|A \mathcal{X}_1 (x_*^{\geq 2} - x^{\geq 2})\|, \\ \|A \mathcal{X}_1 (x_*^{\geq 2} - x^{\geq 2})\| &\leq \frac{\Omega}{\sqrt{1 - \omega_1^2}} \|\mathcal{X}_1^* A \mathcal{X}_1 (x_*^{\geq 2} - t^{\geq 2})\|. \end{aligned}$$

Since $\mathcal{X}_1^* A \mathcal{X}_1 x_*^{\geq 2} = \mathcal{X}_1^* y$, and $\mathcal{X}_1 t^{\geq 2} = u$, it holds

$$\|A \mathcal{X}_1 (x_*^{\geq 2} - x^{\geq 2})\| \leq \frac{\Omega}{\sqrt{1 - \omega_1^2}} \|\mathcal{X}_1^* z\| \leq \frac{\Omega}{\sqrt{1 - \omega_1^2}} \|z\|.$$

Therefore, for the total residual we have

$$\|y - A\tau(X^{(1)}, X^{\geq 2})\| \leq \left(\omega_1 + \frac{1}{\sqrt{1 - \omega_1^2}} \Omega \right) \|y - Au\|.$$

Plugging in the ALS update, the final estimate for (29) now writes as follows,

$$\|y - A\tau(X^{(1)}, X^{\geq 2})\| \leq \mu_1 \left(\omega_1 + \frac{1}{\sqrt{1 - \omega_1^2}} \Omega \right) \|y - At\|,$$

which finishes the recursion. □

Contrarily to the symmetric positive definite case, where the total progress of the AMEn method was deteriorating with d , but less than 1 in any case, here we may have a situation when the progress bound given by Lemma 2 is greater than 1. Up to this moment, the only available estimate is $\omega_k \leq \omega_1^F$, since we enrich the basis by $Z_k^{(k)}$, i.e. the first Krylov vector only. In principle, it is possible to include a larger approximate Krylov basis into the enrichment, i.e.

$$\left[Z_k^{(k)} \quad Q^{[1](k)} \quad \dots \quad Q^{[m-1](k)} \right],$$

where $q^{[p]} = \tau(Q^{[p](k)}, \dots, Q^{[p](d)}) \approx A_k^p z_k$, $p = 1, \dots, m-1$. However, this was not found to be reasonable in practical experiments. In all considered cases, the decays ω_k and μ_k provided by the single enrichment $Z_k^{(k)}$ appeared to be sufficiently small to ensure the convergence, fast enough to overcome the work required to prepare several Krylov vectors.