ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Seminar for Applied Mathematics SAM

Robust solver in a quantized tensor format for three-dimensional elliptic problems

M. Rakhuba

Research Report No. 2019-30
June 2019

Seminar für Angewandte Mathematik
Eidgenössische Technische Hochschule
CH-8092 Zürich
Switzerland

Funding: ETH Grant ETH-44 17-1

ROBUST SOLVER IN A QUANTIZED TENSOR FORMAT FOR THREE-DIMENSIONAL ELLIPTIC PROBLEMS

M. RAKHUBA†

Abstract. The aim of this paper is to propose a *robust* numerical solver, which is capable of efficiently solving a three-dimensional elliptic problem in a data-sparse quantized tensor format. In particular, we use the combined Tucker and quantized tensor train format (TQTT), which allows us to use astronomically large grid sizes. However, due to ill-conditioning of discretized differential operators, such fine grids lead to numerical instabilities. The idea to obtain a robust solver is to utilize the well-known alternating direction implicit method and modify it to avoid multiplication by differential operators. So as to make the method efficient, we derive an explicit TQTT representation of the iteration matrix and QTT representations of the inverses of symmetric tridiagonal Toeplitz matrices as an auxiliary result. As an application, we consider accurate solution of elliptic problems with singular potentials arising in electronic Schroedinger's equation.

1. Introduction. The idea of quantization [22, 23, 17] is to reshape an array with 2^{L} entries into a $2 \times \cdots \times 2$ multidimensional array, and then to apply tensor decomposition to reduce the number of parameters. This approach has appeared to be fruitful to solve partial differential equations, where quantization is applied to vectors and matrices arising after the discretization on very fine uniform *virtual*[1] meshes. Although such fine meshes result in excessive resolution in parts of the domain where the solution is smooth, the underlying black-box compression of tensor representations based on singular value decomposition allows us to dramatically reduce the total number of parameters in the quantized representation. As an example, it was proven [14] that under certain smoothness assumptions finite element solutions to elliptic PDEs can be approximated with quantized tensor train (QTT) using a small number of parameters. In particular, compressed by QTT finite element (FE) solutions with underlying low-order discretization on uniform grids converge exponentially with respect to the number of effective degrees of freedom in their QTT representations.

Despite the fact that, in a variety of cases, solutions of differential equations can be approximated using the quantized approach with a small number of parameters, finding these approximations can be a challenging task. Indeed, the underlying discretization is produced on "astronomically" large virtual grids, with, e.g., 2^{50} grid points in each physical dimension, and due to ill-conditioning of discretized differential operators and round-off errors, severe instability effects occur [14, 3]. Let us support this fact with an example of the finite difference (FD) discretization of $-u''(x) = \sin \pi x$, $x \in (0,1)$ with b.c. $u(0) = u(1) = 0$ on a uniform grid with 2^{L} internal grid points. In Figure 1, we compare the relative error obtained by a standard tridiagonal matrix solver with the relative error obtained by the optimization-based algorithm [7] to find a low-rank QTT approximation to the solution. We observe that for fine grids the error for both approaches increases, highlighting ill-conditioning of the problem.

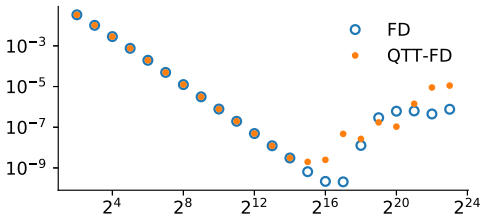


Fig. 1: ℓ_2 relative error w.r.t. #grid points 2^{L} for FD solution of 1D Poisson's equation.

The goal of this paper is to overcome the aforementioned stability issue by developing a robust and efficient numerical solver based on a quantized tensor format for elliptic problems of the form

$$\begin{aligned} -\nabla^2 u + \kappa^2 u &= f \quad \text{in} \quad \Omega, \\ u|_{\partial\Omega} &= g, \end{aligned} \tag{1.1}$$

where Ω a rectangular hexahedron in $\mathbb{R}^3$ and κ is a real, possibly large constant. The key assumption we utilize is that both the right-hand side and the solution of the discretized problem allow for low-rank quantized tensor representations. We note that problem (1.1) can be used as a building block for constructing robust numerical solvers based on iterative methods for more general problems such as those with non-constant κ. Of particular interest are problems arising in electronic structure calculations such as density functional theory, where fine grids allow to accurately approximate singularities of the solution around the location of nuclei. The fact that solutions to such equations allow low-rank

†Seminar for Applied Mathematics, ETH Zurich, Rämistrasse 101, 8092 Zurich, Switzerland
maksim.rakhuba@sam.math.ethz.ch

[1]QTT decomposition is applied to operators and functions under consideration discretized on very fine meshes, called *virtual*. This name stresses the fact that matrices and vectors arising in the discretization are never formed explicitly. No computations are performed without additional compression based on tensor decompositions.

representations has been observed numerically in a number of works [16, 27, 30, 31].

Contributions. To overcome the stability problem, we propose an algorithm based on the alternating direction implicit (ADI) iteration. The advantage of the ADI iteration is that it utilizes a nearly optimal number of iterations at the cost of only solving the linear systems arising in discretization of one-dimensional PDEs and multiplication by matrices of discretized second derivatives. The key observation we make is that one step of the ADI method can be equivalently represented avoiding multiplication by matrices that come from the discretization of derivatives (Section 2). Instead of a QTT format that corresponds to a linear tensor network, we utilize the combined Tucker and QTT decomposition (TQTT), which leads to smaller ranks both for the iteration matrix of the ADI method and the solution vector (Section 3). To make the method efficient, we derive explicit formulas for the iteration matrix of the ADI method and show that its tensor rank is bounded by 5 (Section 5). As an auxiliary result we obtain explicit QTT representations, with all ranks equal to 5, of tridiagonal Toeplitz matrices (Section 4). The efficiency of the method is certified by numerical experiments on model problems with grid sizes up to 2^{120} grid points, which for the considered examples and moderate accuracies takes less than a minute of computational time on a laptop. Numerical examples also include preliminary results of solving equations arising in electronic structure calculations.

Related work. The quantization approach was proposed in [22, 23] for matrices and in [17] for a more general setting. Since then it has been successfully applied to solve differential equations in various applications, see the surveys [11, 4, 18, 16, 19] and references therein. It was proven that, in certain cases, one can obtain exponential convergence with respect to the number of effective degrees of freedom in quantized representations [10, 14, 13]. Nevertheless, the possibility to use very fine virtual grids so as to considerably benefit from the quantization was limited due to numerical instabilities, as noted in [14]. The first attempt to address instabilities arising in discretization of elliptic PDEs was made in [26] for the one-dimensional case and generalized to two spatial dimensions in [3]. Although the solver opened up the possibility to use $\sim 2^{20}$ grid points in every space dimension (for two-dimensional problems), for finer grids instabilities still occurred. In the recent work [1] the problem of instabilities in a QTT format was formalized and a solver based on a BPX preconditioner was proposed. In order to assemble the preconditioned matrix the authors derived analytically its explicit representation for general elliptic operators and an arbitrary number of spatial variables. The solver allows for the solution of two-dimensional problems within minutes of computational time, but requires much more time for problems in three and more dimensions. The problem is that rank of the preconditioned matrix grows exponentially w.r.t. number of physical variables (although it is independent of the number of grid levels).

The ADI method used in this paper was introduced in [29] in the context of solving two-dimensional elliptic and parabolic partial differential equations. Since then, it has been used in different applications including Lyapunov and Sylvester matrix equations [32]. We also refer to the book [34] for more details regarding the theoretical aspects of the method. In the context of tensor decompositions the ADI iteration was considered in [20] without quantization.

Contributions of this paper also include explicit formulas for the inverse of certain tridiagonal Toeplitz matrices. In [15] explicit QTT representations for the Laplace operator and its inverse (for one physical dimension) were proposed. In particular, an explicit formula for the inverse of $\mathrm{tridiag}(-1, 2, -1)$ was suggested. In this paper we, however, need an inverse of a more general tridiagonal Toeplitz matrix $\mathrm{tridiag}(-1, \alpha, -1)$, $\alpha > 2$. The approach proposed in this paper allows us to obtain explicit representations with QTT ranks equal to 5, and can be easily extended to find the QTT inverse of a general tridiagonal Toeplitz matrix as is indicated in Section 3.

2. Alternating direction implicit method. In this section, we formulate the ADI method for the discretized problem and extend the result [8] of choosing iterative parameters of ADI to the case of the screened Poisson's equation. We also present derivative-free formulas for the ADI method that allow us to avoid multiplication by discretized differential operators. The possibility to use derivative-free formulas lets us to avoid instability arising due to round-off errors on very fine virtual grids.

Consider the three-dimensional screened Poisson's equation (1.1) in a cube $\Omega = (a, b)^3$ where $\kappa \geq 0$ is a constant. Let us discretize it on a uniform grid $\Omega^{(\mathrm{L})} = \{a + jh_{\mathrm{L}} \colon j = 1, \ldots, 2^{\mathrm{L}}\}^3$ with 2^{L} grid points in each spatial variable, where $h_{\mathrm{L}} = (b - a) \cdot (2^{\mathrm{L}} + 1)^{-1}$ is the grid step. The second order finite

difference (FD) discretization of (1.1) reads

$$\mathbf{\Sigma}^{(\mathrm{L})}\mathbf{u}^{(\mathrm{L})} = \mathbf{f}^{(\mathrm{L})}, \tag{2.1}$$

where

$$\mathbf{\Sigma}^{(\mathrm{L})} = \mathbf{S}^{(\mathrm{L})} \otimes \mathbf{I}^{(\mathrm{L})} \otimes \mathbf{I}^{(\mathrm{L})} + \mathbf{I}^{(\mathrm{L})} \otimes \mathbf{S}^{(\mathrm{L})} \otimes \mathbf{I}^{(\mathrm{L})} + \mathbf{I}^{(\mathrm{L})} \otimes \mathbf{I}^{(\mathrm{L})} \otimes \mathbf{S}^{(\mathrm{L})} + \kappa^2\, \mathbf{I}^{(\mathrm{L})} \otimes \mathbf{I}^{(\mathrm{L})} \otimes \mathbf{I}^{(\mathrm{L})},$$

$$\mathbf{S}^{(\mathrm{L})} = \frac{1}{h_{\mathrm{L}}^2} \begin{bmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix}_{2^{\mathrm{L}}\times 2^{\mathrm{L}}}, \qquad \mathbf{I}^{(\mathrm{L})} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}_{2^{\mathrm{L}}\times 2^{\mathrm{L}}},$$

and the vectors $\mathbf{u}^{(\mathrm{L})}, \mathbf{f}^{(\mathrm{L})} \in \mathbb{R}^{2^{\mathrm{L}}}$ are correspondingly the FD solution vector and the right-hand side f evaluated at the points of $\Omega^{(\mathrm{L})}$ with nonzero boundary conditions taken into account. Alternatively, one could utilize the finite element method (FEM) based on the tensor product of one-dimensional piecewise-linear hat basis functions. In this case, one would obtain tridiagonal Toeplitz mass matrices instead of $\mathbf{I}^{(\mathrm{L})}$, and the approach proposed in this paper would still be applicable. Nevertheless, we focus on the FD discretization for simplicity.

Let us introduce the ADI iteration to solve (2.1), first without imposing the low-rank constraints. For this purpose, we introduce the notation

$$\mathbf{\Sigma}_1 = \left(\mathbf{S}^{(\mathrm{L})} + \frac{\kappa^2}{3}\,\mathbf{I}^{(\mathrm{L})}\right) \otimes \mathbf{I}^{(\mathrm{L})} \otimes \mathbf{I}^{(\mathrm{L})},$$

$$\mathbf{\Sigma}_2 = \mathbf{I}^{(\mathrm{L})} \otimes \left(\mathbf{S}^{(\mathrm{L})} + \frac{\kappa^2}{3}\,\mathbf{I}^{(\mathrm{L})}\right) \otimes \mathbf{I}^{(\mathrm{L})},$$

$$\mathbf{\Sigma}_3 = \mathbf{I}^{(\mathrm{L})} \otimes \mathbf{I}^{(\mathrm{L})} \otimes \left(\mathbf{S}^{(\mathrm{L})} + \frac{\kappa^2}{3}\,\mathbf{I}^{(\mathrm{L})}\right),$$

so that the matrix of (2.1) can be written as $\mathbf{\Sigma}^{(\mathrm{L})} = \mathbf{\Sigma}_1 + \mathbf{\Sigma}_2 + \mathbf{\Sigma}_3$. The Kronecker product structure of each of $\mathbf{\Sigma}_j$ allows us to invert shifted matrices $\mathbf{\Sigma}_j + \sigma\mathbf{I}$ with some $\sigma \in \mathbb{R}$ using only the inverse of a tridiagonal matrix $(\mathbf{S}^{(\mathrm{L})} + (\kappa^2/3 + \sigma)\cdot \mathbf{I}^{(\mathrm{L})})$. Therefore, unless data-sparse formats are used, the linear systems with the matrix $\mathbf{\Sigma}_j + \sigma\mathbf{I}$ can be solved in linear time with respect to the number of unknowns. The ADI method takes advantage of this property and involves only linear systems with such matrices. In particular, the ADI method for three-dimensional problems proposed by Douglas [8] reads: given $\mathbf{u}_0 \in \mathbb{R}^{3\mathrm{L}}$, compute

$$\begin{aligned} (\mathbf{\Sigma}_1 + \sigma_k\mathbf{I})\,\mathbf{u}_{k+1/3} &= -(\mathbf{\Sigma}_1 + 2\mathbf{\Sigma}_2 + 2\mathbf{\Sigma}_3 - \sigma_k\mathbf{I})\,\mathbf{u}_k + 2\mathbf{f}^{(\mathrm{L})} \\ (\mathbf{\Sigma}_2 + \sigma_k\mathbf{I})\,\mathbf{u}_{k+2/3} &= \mathbf{\Sigma}_2\,\mathbf{u}_k + \sigma_k\,\mathbf{u}_{k+1/3} \\ (\mathbf{\Sigma}_3 + \sigma_k\mathbf{I})\,\mathbf{u}_{k+1} &= \mathbf{\Sigma}_3\,\mathbf{u}_k + \sigma_k\,\mathbf{u}_{k+2/3}, \qquad k = 0, 1, 2, \ldots \end{aligned} \tag{2.2}$$

where iteration parameters σ_k are called *shifts* and $\mathbf{u}_k$ is expected to approximate $\mathbf{u}^{(\mathrm{L})}$ for large enough k. Formulas (2.2) involve both the solution of linear systems with the matrices $(\mathbf{\Sigma}_j + \sigma_k\mathbf{I})$ and matrix-vector products by the matrices $\mathbf{\Sigma}_j$, $j = 1, 2, 3$. The robust inversion of the tridiagonal Toeplitz matrices arising in the iterations of the method, is considered in Section 4. By contrast, multiplication by $\mathbf{\Sigma}_j$ is unstable for fine virtual grids [1]. Fortunately, one can avoid performing multiplications by using equivalent (in exact arithmetics) derivative-free formulas as is shown in Section 2.1.

2.1. Derivative-free ADI formulas. To obtain derivative-free formulas for (2.2), let us first equivalently rewrite (2.2) as

$$\mathbf{u}_{k+1} = \mathbf{u}_k - 2\sigma_k^2\,\mathbf{B}(\sigma_k)\left((\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2 + \mathbf{\Sigma}_3)\mathbf{u}_k - \mathbf{f}^{(\mathrm{L})}\right), \tag{2.3}$$

where

$$\mathbf{B}(\sigma) = (\mathbf{\Sigma}_3 + \sigma\,\mathbf{I})^{-1}\,(\mathbf{\Sigma}_2 + \sigma\,\mathbf{I})^{-1}\,(\mathbf{\Sigma}_1 + \sigma\,\mathbf{I})^{-1},$$

To avoid multiplication by $\mathbf{\Sigma}_j$, $j = 1, 2, 3$, we observe that

$$(\mathbf{\Sigma}_j + \sigma\,\mathbf{I})^{-1}\,\mathbf{\Sigma}_j = (\mathbf{\Sigma}_j + \sigma\,\mathbf{I})^{-1}\,(\mathbf{\Sigma}_j + \sigma\,\mathbf{I} - \sigma\,\mathbf{I}) = \mathbf{I} - \sigma\,(\mathbf{\Sigma}_j + \sigma\,\mathbf{I})^{-1}. \tag{2.4}$$

Then, thanks to the commutativity of $\mathbf{\Sigma}_i$ and $\mathbf{\Sigma}_j$ for $i, j = 1, 2, 3$ and using (2.4), we may write

$$\mathbf{B}(\sigma)\,(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2 + \mathbf{\Sigma}_3) = \frac{1}{2}\sum_{\substack{i,j=1\\ i\neq j}}^{3}(\mathbf{\Sigma}_i + \sigma\,\mathbf{I})^{-1}\,(\mathbf{\Sigma}_j + \sigma\,\mathbf{I})^{-1} - 3\sigma\prod_{j=1}^{3}(\mathbf{\Sigma}_j + \sigma\,\mathbf{I})^{-1}. \tag{2.5}$$

Introducing the matrix $\mathbf{T}(\sigma)$:

$$\mathbf{T}(\sigma) = \mathbf{I} - 2\sigma^2\,\mathbf{B}(\sigma)\,(\mathbf{\Sigma}_1 + \mathbf{\Sigma}_2 + \mathbf{\Sigma}_3)$$

we can write the iterative process (2.3) as

$$\mathbf{u}_{k+1} = \mathbf{T}(\sigma_k)\,\mathbf{u}_k - 2\sigma_k^2\,\mathbf{B}(\sigma_k)\mathbf{f}^{(\mathtt{L})}, \tag{2.6}$$

which we will use later for the low-rank version of the iteration. Thanks to (2.5), applications of the matrix $\mathbf{T}(\sigma)$ can be represented without multiplications by $\mathbf{\Sigma}_j$. Thus, every step of (2.3) can be written only in terms of multiplications by the inverse of a tridiagonal Toeplitz matrix $(\mathbf{S}^{(\mathtt{L})} + (\kappa^2/3 + \sigma)\cdot\mathbf{I}^{(\mathtt{L})})$ that arises in $(\mathbf{\Sigma}_j + \sigma\,\mathbf{I})^{-1}$. Before we proceed to the derivation of explicit low-rank representations for the inverse of tridiagonal Toeplitz matrices in the QTT format, let us discuss the choice of shift parameters σ_k.

2.2. Choice of shifts σ_k. Parameters σ_k determine the convergence rate of the iterative process (2.3). Since we are aiming at running the iteration on very fine grids, we want to obtain convergence to the desired tolerance in at worst $\mathcal{O}(\mathtt{L}^\beta)$ steps for some $\beta \geq 0$, i.e., polylogarithmically in the total number of virtual grid points $2^{3\mathtt{L}}$. The choice of parameters from [8] leads to $\mathcal{O}(\mathtt{L})$ iterations to achieve a given tolerance for $\kappa = 0$. The author is not aware[2] of any general result for the choice of shifts which is based on spectral bounds for $\mathbf{\Sigma}_1, \mathbf{\Sigma}_2, \mathbf{\Sigma}_3$. In this section, we, therefore, adapt the result from [8] to the case $\kappa \geq 0$. We note that the generalization presented below follows from [8] with only minor modifications.

Let us introduce the error $\mathbf{e}_k = \mathbf{u}^{(\mathtt{L})} - \mathbf{u}_k$, where $\mathbf{u}_k$ is a sequence generated by (2.3). Then,

$$\mathbf{e}_{k+1} = \mathbf{T}(\sigma_k)\,\mathbf{e}_k,$$

If $\sigma_k = \sigma_0 > 0$ for all k, then [8] (for $\kappa = 0$)

$$\|\mathbf{e}_{k+1}\| \leq \rho\,\|\mathbf{e}_k\|, \quad 0 < \rho < 1,$$

and hence $\mathbf{u}_k \to \mathbf{u}^{(\mathtt{L})}$ as $k \to \infty$. In this case, however, convergence is slow even if parameter σ_0 is chosen optimally. To get faster convergence one has to consider nonconstant shifts σ_k, which are usually chosen cyclically every N iterations, i.e.,

$$\sigma_k = \sigma_{k\,(\mathrm{mod}\,N)}, \quad k = 0, 1, 2, \ldots. \tag{2.7}$$

To determine a sequence that allows us to achieve the desired accuracy in $\mathcal{O}(\mathtt{L})$ iterations, we follow [8] and investigate decay of individual components of the error using the eigenbasis of $\mathbf{\Sigma}$. Without loss of generality assume that $k \in \{0, 1, \ldots, N-1\}$. The matrix $\mathbf{S}^{(\mathtt{L})}$ has eigenvectors $\mathbf{v}_p = \{\sin\pi pj(2^{\mathtt{L}} + 1)^{-1}\}_{j=1}^{2^{\mathtt{L}}}$, $p = 1, \ldots, 2^{\mathtt{L}}$. Thus, the symmetric matrix $\mathbf{\Sigma}$ has eigenvectors $\mathbf{v}_p \otimes \mathbf{v}_q \otimes \mathbf{v}_r$, $p, q, r = 1, \ldots, 2^{\mathtt{L}}$ that form a basis in $\mathbb{R}^{3\mathtt{L}}$. Let us decompose $\mathbf{e}_0$ using these eigenvectors with some coefficients $\mathcal{C}_{pqr}$:

$$\mathbf{e}_0 = \sum_{p,q,r=1}^{2^{\mathtt{L}}} \mathcal{C}_{pqr}\,\mathbf{v}_p \otimes \mathbf{v}_q \otimes \mathbf{v}_r.$$

[2] In [34] E. Wachspress also indicated the unawareness of a general result for three-dimensional problems.

After N iterations, we obtain

$$\mathbf{e}_N = \left(\prod_{k=0}^{N-1} \mathbf{T}(\sigma_k)\right) \mathbf{e}_0 = \sum_{p,q,r=1}^{2^{\mathtt{L}}} \mathcal{C}_{pqr} \left(\prod_{k=0}^{N-1} \rho_{pqr}(\sigma_k)\right) \mathbf{v}_p \otimes \mathbf{v}_q \otimes \mathbf{v}_r, \tag{2.8}$$

where $\rho_{pqr}(\sigma)$ are the eigenvalues of the matrix $\mathbf{T}(\sigma)$:

$$\rho_{pqr}(\sigma) = 1 - \frac{2\left[\lambda_p(\sigma) + \lambda_q(\sigma) + \lambda_r(\sigma)\right]}{\left[1+\lambda_p(\sigma)\right]\left[1+\lambda_q(\sigma)\right]\left[1+\lambda_r(\sigma)\right]},$$

expressed in terms of the eigenvalues $\lambda_p(\sigma)$ of the matrix $\sigma^{-1} \cdot (\mathbf{S}^{(\mathtt{L})} + \kappa^2/3\ \mathbf{I}^{(\mathtt{L})})$:

$$\lambda_p(\sigma) = \frac{4}{\sigma h_{\mathtt{L}}^2}\left(\sin^2 \frac{\pi p}{2(2^{\mathtt{L}}+1)} + \frac{\kappa^2 h_{\mathtt{L}}^2}{12}\right), \quad p = 1, \ldots, 2^{\mathtt{L}}.$$

Note that $|\rho_{pqr}(\sigma)| < 1$ for any $\sigma > 0$ since

$$0 < \frac{a+b+c}{(1+a)(1+b)(1+c)} < 1 \quad \forall\, a, b, c > 0.$$

The idea to choose σ_k, $k = 0, 1, \ldots, N-1$ is to split eigenvectors $\mathbf{v}_p \otimes \mathbf{v}_q \otimes \mathbf{v}_r$, $p, q, r = 1, \ldots, 2^{\mathtt{L}}$ into N groups such that the corresponding error component in (2.8) in each of the groups is sufficiently decreased after N iterations. To ensure that, we use the following lemma.

LEMMA 2.1 ([8]). *Let*

$$\rho(a, b, c) = 1 - \frac{2(a+b+c)}{(1+a)(1+b)(1+c)},$$

let also $\nu \geq 1$ *and*

$$\mu = \frac{3\nu}{1 + 3\nu^2 + \nu^3}, \tag{2.9}$$

then

$$\widehat{\rho}\,(\mu, \nu) = \max_{\substack{\mu \leq a \leq \nu \\ 0 \leq b, c \leq \nu}} \left|\rho(a, b, c)\right| = 1 - \frac{6\nu}{(1+\nu)^3} = 1 - \frac{2\mu}{1+\mu}.$$

With the help of Lemma 2.1 we can prove the following result.

PROPOSITION 2.1. *Let parameters* σ_k *of the iterative process* (2.3) *be chosen cyclically in accordance with* (2.7) *and such that*

$$\sigma_k = \frac{4}{\mu}\left(\frac{\nu}{\mu}\right)^k \frac{\sin^2 \dfrac{\pi}{2(2^{\mathtt{L}}+1)} + \dfrac{\kappa^2 h_{\mathtt{L}}^2}{12}}{h_{\mathtt{L}}^2}, \quad k = 0, 1, \ldots, N-1,$$

for some $\nu \geq 1$, μ *as in* (2.9) *and with*

$$N = \left\lceil 1 + \log\left(\frac{\cos^2 \frac{\pi}{2(2^{\mathtt{L}}+1)} + \frac{\kappa^2 h_{\mathtt{L}}^2}{12}}{\sin^2 \frac{\pi}{2(2^{\mathtt{L}}+1)} + \frac{\kappa^2 h_{\mathtt{L}}^2}{12}}\right) \Big/ \log\left(\frac{\nu}{\mu}\right)\right\rceil = \mathcal{O}(\mathtt{L}).$$

Then after

$$n = \left\lceil \frac{\log \varepsilon^{-1}}{\log\left[(1+\mu)/(1-\mu)\right]} \right\rceil$$

cycles, for $\varepsilon < 1$ *we get*

$$\|\mathbf{u}^{(\mathtt{L})} - \mathbf{u}_{nN-1}\| \leq \varepsilon \|\mathbf{u}^{(\mathtt{L})} - \mathbf{u}_0\|.$$

Moreover, the parameter choice $\nu = \nu_* \approx 1.778$, $\mu = \mu_* \approx 0.3312$ *minimizes* nN *– the total number of iterations.*

Proof. Denote

$$\zeta(\sigma) = \frac{4}{\sigma h_{\mathtt{L}}^2}, \quad \xi_p = \sin^2 \frac{\pi p}{2(2^{\mathtt{L}}+1)} + \frac{\kappa^2 h_{\mathtt{L}}^2}{12}, \tag{2.10}$$

so that $\lambda_p(\sigma) = \zeta(\sigma)\xi_p$. We set

$$\xi^{(0)} = \sin^2 \frac{\pi}{2(2^{\mathtt{L}}+1)} + \frac{\kappa^2 h_{\mathtt{L}}^2}{12}$$

and

$$\begin{aligned} \zeta(\sigma_k)\, \xi^{(k)} &= \mu, \\ \zeta(\sigma_k)\, \xi^{(k+1)} &= \nu, \end{aligned}$$

resulting into

$$\begin{aligned} \zeta(\sigma_k) &= \mu \left(\frac{\mu}{\nu}\right)^k \left(\sin^2 \frac{\pi}{2(2^{\mathtt{L}}+1)} + \frac{\kappa^2 h_{\mathtt{L}}^2}{12}\right)^{-1}, \\ \xi^{(k)} &= \left(\frac{\nu}{\mu}\right)^k \left(\sin^2 \frac{\pi}{2(2^{\mathtt{L}}+1)} + \frac{\kappa^2 h_{\mathtt{L}}^2}{12}\right), \end{aligned} \tag{2.11}$$

which ensures

$$\mu \le \zeta(\sigma_k)\xi \le \nu, \quad \xi \in [\xi^{(k)}, \xi^{(k+1)}].$$

The sequence is interrupted right after $\xi^{(N-1)}$ becomes larger than $\xi_{2^{\mathtt{L}}}$:

$$\xi^{(N-1)} \ge \sin^2 \frac{\pi 2^{\mathtt{L}}}{2(2^{\mathtt{L}}+1)} + \frac{\kappa^2 h_{\mathtt{L}}^2}{12} \equiv \cos^2 \frac{\pi}{2(2^{\mathtt{L}}+1)} + \frac{\kappa^2 h_{\mathtt{L}}^2}{12}.$$

For small enough $h_{\mathtt{L}}$, it leads to

$$N \ge 1 + \log \left(\frac{\cos^2 \frac{\pi}{2(2^{\mathtt{L}}+1)} + \frac{\kappa^2 h_{\mathtt{L}}^2}{12}}{\sin^2 \frac{\pi}{2(2^{\mathtt{L}}+1)} + \frac{\kappa^2 h_{\mathtt{L}}^2}{12}} \right) \Big/ \log \left(\frac{\nu}{\mu}\right) = 1 + \frac{2\mathtt{L} - \log_2 \left(\frac{\pi^2}{4} + \frac{\kappa^2 (b-a)^2}{12} \right)}{\log_2 \left(\frac{\nu}{\mu}\right)} + \mathcal{O}(2^{-\mathtt{L}}) = \mathcal{O}(\mathtt{L}). \tag{2.12}$$

Finally for given ν and μ, the shifts are defined from (2.10) and (2.11):

$$\sigma_k = \frac{4}{\mu} \left(\frac{\nu}{\mu}\right)^k \frac{\sin^2 \frac{\pi}{2(2^{\mathtt{L}}+1)} + \frac{\kappa^2 h_{\mathtt{L}}^2}{12}}{h_{\mathtt{L}}^2} = \frac{1}{\mu} \left(\frac{\nu}{\mu}\right)^k \left(\frac{\pi^2}{(b-a)^2} + \frac{\kappa^2}{3} \right) + \mathcal{O}(2^{-2\mathtt{L}}). \tag{2.13}$$

Using Lemma 2.1 and accounting for the fact that after N iterations the error in every component is decreased at least by a factor of $\widehat{\rho}(\mu, \nu)$, we obtain

$$\left\| \prod_{k=0}^{N-1} \mathbf{T}(\sigma_k) \right\| \le \widehat{\rho}\,(\mu, \nu), \tag{2.14}$$

where shifts σ_k, $k = 0, 1, \ldots, N-1$ are defined in (2.13). If shift parameters are chosen cyclically according to (2.7), then

$$\left\| \prod_{k=0}^{nN-1} \mathbf{T}(\sigma_k) \right\| \le \widehat{\rho}\,(\mu, \nu)^n, \quad n = 1, 2, \ldots.$$

To achieve $\widehat{\rho}\,(\mu, \nu)^n \le \varepsilon$, we need to make

$$n \ge \frac{\log \varepsilon^{-1}}{\log \widehat{\rho}\,(\mu, \nu)^{-1}},$$

outer iterations, which leads to the total number of iterations equal to

$$nN \sim \frac{\log \varepsilon^{-1}}{\log \widehat{\rho}\,(\mu,\nu)^{-1} \log \frac{\nu}{\mu}}.$$

Maximization of the denominator in the latter expression over $\nu \geq 1$ yields optimal parameters ν, μ:

$$\nu_* \approx 1.778, \quad \mu_* = \frac{3\nu_*}{1+3\nu_*^2+\nu_*^3} \approx 0.3312, \tag{2.15}$$

and

$$\widehat{\rho}\,(\mu_*,\nu_*) \approx 0.5023. \qquad \square$$

Thus, as required, we obtain an iterative method that allows us to achieve the accuracy ε in overall $\mathcal{O}(\mathtt{L}\log\varepsilon^{-1})$ iterations. We note, however, that the presented analysis is quite crude and in numerical experiments we observe even faster convergence of the method.

3. Quantized tensor representations. In this section, we introduce the quantized tensor train (QTT) format and the format that is a combination of the Tucker decomposition and the QTT [5] (we refer to it as TQTT), which we use for three-dimensional problems.

3.1. QTT decomposition. To introduce the QTT decomposition of a matrix of order $2^{\mathtt{L}}$, we encode its row and column indices i, j by using their binary representation

$$i = \overline{i_1,\ldots,i_{\mathtt{L}}}, \quad j = \overline{j_1,\ldots,j_{\mathtt{L}}}, \quad i_\ell, j_\ell \in \{0,1\}, \quad \ell = 1,\ldots,\mathtt{L},$$

where we use the notation

$$\overline{i_1,\ldots,i_{\mathtt{L}}} \equiv 2^{\mathtt{L}-1}i_1 + 2^{\mathtt{L}-2}i_2 + \cdots + 2i_{\mathtt{L}-1} + i_{\mathtt{L}}.$$

We say that matrix $\mathbf{A} \in \mathbb{R}^{2^{\mathtt{L}}\times 2^{\mathtt{L}}}$ is represented using the QTT decomposition if

$$\mathbf{A}_{ij} = \sum_{\alpha_1=0}^{r_0}\sum_{\alpha_1=1}^{r_1}\cdots\sum_{\alpha_\ell=1}^{r_\ell}\left(\mathbf{g}^{(1)}_{\alpha_0\alpha_1}\right)_{i_1,j_1}\left(\mathbf{g}^{(2)}_{\alpha_1\alpha_2}\right)_{i_2,j_2}\ldots\left(\mathbf{g}^{(\mathtt{L})}_{\alpha_{L-1}\alpha_L}\right)_{i_L,j_L} \qquad i,j = 1,\ldots,2^{\mathtt{L}}, \tag{3.1}$$

where $\mathbf{g}^{(\ell)}_{\alpha_{\ell-1}\alpha_\ell}$ are 2×2 matrices for each $\ell = 1,\ldots,L$ and $\alpha_{\ell-1} = 1,\ldots,r_{\ell-1}$, $\alpha_\ell = 1,\ldots,r_\ell$, $r_0 = r_{\mathtt{L}} = 1$. For our purposes it is, however, more convenient to rewrite (3.1) as:

$$\mathbf{A} = \sum_{\alpha_1=0}^{r_0}\sum_{\alpha_1=1}^{r_1}\cdots\sum_{\alpha_\ell=1}^{r_\ell}\mathbf{g}^{(1)}_{\alpha_0\alpha_1}\otimes\mathbf{g}^{(2)}_{\alpha_1\alpha_2}\otimes\cdots\otimes\mathbf{g}^{(L-1)}_{\alpha_{\mathtt{L}-2}\alpha_{L-1}}\otimes\mathbf{g}^{(\mathtt{L})}_{\alpha_{L-1}\alpha_L}. \tag{3.2}$$

Representation (3.2) resembles multiplication of $\mathtt{L}$ block matrices, where multiplication between blocks is replaced by the Kronecker product. This concept is known under the name *strong Kronecker product*, and we denote it by "$\bowtie$", following [15]. For example,

$$\begin{bmatrix}\mathbf{B}_{11} & \mathbf{B}_{12}\\ \mathbf{B}_{21} & \mathbf{B}_{22}\end{bmatrix} \bowtie \begin{bmatrix}\mathbf{C}_{11} & \mathbf{C}_{12}\\ \mathbf{C}_{21} & \mathbf{C}_{22}\end{bmatrix} = \begin{bmatrix}\mathbf{B}_{11}\otimes\mathbf{C}_{11}+\mathbf{B}_{12}\otimes\mathbf{C}_{21} & \mathbf{B}_{11}\otimes\mathbf{C}_{12}+\mathbf{B}_{12}\otimes\mathbf{C}_{22}\\ \mathbf{B}_{21}\otimes\mathbf{C}_{11}+\mathbf{B}_{22}\otimes\mathbf{C}_{21} & \mathbf{B}_{21}\otimes\mathbf{C}_{12}+\mathbf{B}_{22}\otimes\mathbf{C}_{22}\end{bmatrix}.$$

The strong Kronecker product allows us to conveniently write the TT-decomposition (3.2) as follows [15]:

$$\mathbf{A} = \mathrm{G}_1 \bowtie \mathrm{G}_2 \bowtie \cdots \bowtie \mathrm{G}_{\mathtt{L}}, \tag{3.3}$$

where

$$\mathrm{G}_1 = \begin{bmatrix}\mathbf{g}^{(1)}_{11} & \ldots & \mathbf{g}^{(1)}_{1r_1}\end{bmatrix}, \quad \mathrm{G}_\ell = \begin{bmatrix}\mathbf{g}^{(\ell)}_{11} & \cdots & \mathbf{g}^{(\ell)}_{1r_\ell}\\ \vdots & \ddots & \vdots\\ \mathbf{g}^{(\ell)}_{r_{\ell-1}1} & \cdots & \mathbf{g}^{(\ell)}_{r_{\ell-1}r_\ell}\end{bmatrix} \ \ell = 2,\ldots,L-1, \quad \mathrm{G}_{\mathtt{L}} = \begin{bmatrix}\mathbf{g}^{(\mathtt{L})}_{11}\\ \vdots\\ \mathbf{g}^{(\mathtt{L})}_{r_L1}\end{bmatrix}.$$

Note that the number of block rows and block columns in G_ℓ indicates the values of $r_{\ell-1}$ and r_ℓ respectively.

EXAMPLE 3.1. *Consider a matrix* $\mathbf{L} \in \mathbb{R}^{8\times 8}$ *of the form*

$$\mathbf{L} = \mathbf{M}\otimes\mathbf{N}\otimes\mathbf{N} + \mathbf{N}\otimes\mathbf{M}\otimes\mathbf{N} + \mathbf{N}\otimes\mathbf{N}\otimes\mathbf{M}, \quad \mathbf{M},\mathbf{N}\in\mathbb{R}^{2\times 2}.$$

Using the strong Kronecker product notation we can find its QTT representation as follows

$$\mathbf{L} = \begin{bmatrix}\mathbf{M} & \mathbf{N}\end{bmatrix} \bowtie \begin{bmatrix}\mathbf{N}\otimes\mathbf{N} \\ \mathbf{M}\otimes\mathbf{N} + \mathbf{N}\otimes\mathbf{M}\end{bmatrix} = \begin{bmatrix}\mathbf{M} & \mathbf{N}\end{bmatrix} \bowtie \begin{bmatrix}\mathbf{N} & \\ \mathbf{M} & \mathbf{N}\end{bmatrix} \bowtie \begin{bmatrix}\mathbf{N} \\ \mathbf{M}\end{bmatrix}.$$

Noticing that the block matrix $\begin{bmatrix}\mathbf{N} & \\ \mathbf{M} & \mathbf{N}\end{bmatrix}$ *has* 2 *block rows and* 2 *block columns, we conclude that* $r_1 = r_2 = 2$.

The QTT decomposition of a vector $\mathbf{x}\in\mathbb{R}^{2^{\mathtt{L}}}$ is defined analogously to (3.3) with the only difference that $\mathbf{g}_{ij}^{(\ell)}$ are in $\mathbb{R}^2$ instead of $\mathbb{R}^{2\times 2}$. Note also that in practice vectors can rarely be represented in tensor formats with small ranks exactly. Therefore, one is usually concerned with obtaining a low-rank QTT approximation to a given vector.

3.2. Combined Tucker and QTT decomposition. We are particularly interested in approximating matrices and vectors arising from the discretization of the three-dimensional problem (1.1) with the physical coordinates denoted by x, y, z. By enumerating grid points of a $2^{\mathtt{L}}\times 2^{\mathtt{L}}\times 2^{\mathtt{L}}$ grid using a single index i and introducing its binarization, we write

$$i = \overline{i_1^{(x)},\ldots,i_{\mathtt{L}}^{(x)},i_1^{(y)},\ldots,i_{\mathtt{L}}^{(y)},i_1^{(z)},\ldots,i_{\mathtt{L}}^{(z)}}, \quad i_\ell^{(x)},i_\ell^{(y)},i_\ell^{(z)}\in\{0,1\}, \quad \ell = 1,\ldots,\mathtt{L}, \tag{3.4}$$

where $i^{(\alpha)} = \overline{i_1^{(\alpha)},\ldots,i_{\mathtt{L}}^{(\alpha)}}$, $\alpha = x, y, z$ enumerates points of a one-dimensional grid in the physical coordinate α. Let $\mathbf{A}\in\mathbb{R}^{2^{3\mathtt{L}}\times 2^{3\mathtt{L}}}$ be a matrix of a discretization of a linear operator, discretized on a tensor product $2^{\mathtt{L}}\times 2^{\mathtt{L}}\times 2^{\mathtt{L}}$ grid with the enumeration of grid points as in (3.4). Then, we may write its QTT decomposition as

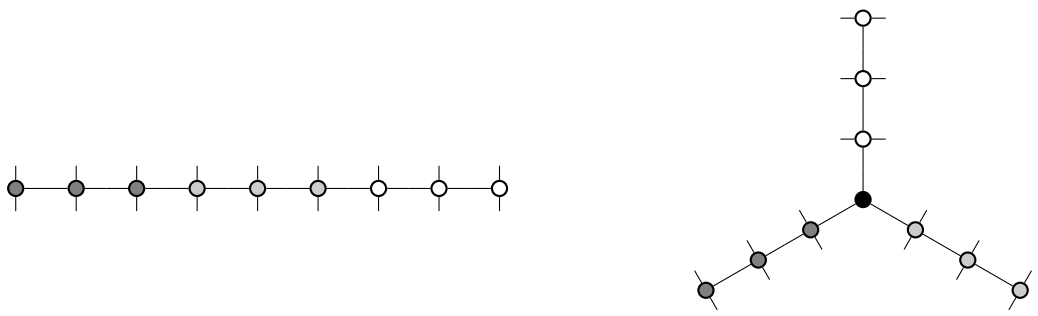
$$\mathbf{A} = \underbrace{\left(\mathrm{G}_1 \bowtie \cdots \bowtie \mathrm{G}_{\mathtt{L}}\right)}_{x\text{-coordinate}} \bowtie \underbrace{\left(\mathrm{G}_{\mathtt{L}+1} \bowtie \cdots \bowtie \mathrm{G}_{2\mathtt{L}}\right)}_{y\text{-coordinate}} \bowtie \underbrace{\left(\mathrm{G}_{2\mathtt{L}+1} \bowtie \cdots \bowtie \mathrm{G}_{3\mathtt{L}}\right)}_{z\text{-coordinate}}. \tag{3.5}$$

where the first $\mathtt{L}$ cores depend on $i_1^{(x)},\ldots,i_{\mathtt{L}}^{(x)}$, the second $\mathtt{L}$ cores on $i_1^{(y)},\ldots,i_{\mathtt{L}}^{(y)}$ and the third $\mathtt{L}$ cores on $i_1^{(z)},\ldots,i_{\mathtt{L}}^{(z)}$. Decomposition (3.5) can be conveniently visualized using tensor network diagrams. For this, we use the following graphical representations. We denote a matrix by –●– where two edges illustrate dependency of the matrix entries on two indices. The matrix-matrix multiplication is denoted as –●–●–. Similarly, we represent an ℓ-dimensional tensor using ℓ outgoing edges. In particular a three-dimensional tensor is denoted by –●– By analogy with matrix multiplication, one can represent a one index contraction of two three-dimensional arrays as –●–●–. Note that in QTT decomposition of matrices (3.3), block matrices G_ℓ can be naturally represented as 4-dimensional arrays of size $r_{\ell-1}\times 2\times 2\times r_\ell$. Thus, the graphical representation of (3.5) is a linear network (see Figure 2a). We put emphasis on the fact that the modes corresponding to the coordinate y are squeezed in between those of x and z. This leads to larger rank values corresponding to the y-coordinate for both matrices and vectors [5].

To overcome this issue we use the combined Tucker and QTT decomposition (TQTT for short), which is similar to the one[3] proposed in [5]. To define TQTT let us first define the Tucker decomposition [33]. A matrix $\mathbf{A}$ is said to be represented using Tucker decomposition with the multilinear rank $\{R_1, R_3, R_3\}$ if

$$\mathbf{A} = \sum_{\alpha_1=1}^{R_1}\sum_{\alpha_2=1}^{R_3}\sum_{\alpha_3=1}^{R_3} \mathcal{G}_{\alpha_1\alpha_2\alpha_3}\,\mathbf{U}_{\alpha_1}\otimes\mathbf{V}_{\alpha_2}\otimes\mathbf{W}_{\alpha_3}, \tag{3.6}$$

[3]The only difference is that we do not apply TT-decomposition to the Tucker core, as in the three-dimensional case it leads to a larger number of parameters.



(a) (b)

Fig. 2: Graphical tensor network representation of a matrix given by: (a) QTT decomposition and (b) combined Tucker and QTT decompositions (TQTT). In this example, $\mathtt{L} = 3$, nodes ●, ●, ○ correspond to coordinates x, y, z respectively. Node ● corresponds to the three-dimensional core arising in the Tucker decomposition. Note that in (a) modes corresponding to y-coordinate are "squeezed" between those of x and z, while in (b) modes corresponding to x, y, z are located with no preferred direction.

where $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$ is called *Tucker core* and $\mathbf{U}_{\alpha_1}, \mathbf{V}_{\alpha_2}, \mathbf{W}_{\alpha_3} \in \mathbb{R}^{2^{\mathtt{L}} \times 2^{\mathtt{L}}}$. The block matrices

$$\mathrm{U} = \begin{bmatrix} \mathbf{U}_1 & \dots & \mathbf{U}_{R_1} \end{bmatrix}, \quad \mathrm{V} = \begin{bmatrix} \mathbf{V}_1 & \dots & \mathbf{V}_{R_2} \end{bmatrix}, \quad \mathrm{W} = \begin{bmatrix} \mathbf{W}_1 & \dots & \mathbf{W}_{R_3} \end{bmatrix},$$

are called the *Tucker factors*. We use the following notation to compactly write the Tucker decomposition (3.6) of $\mathbf{A}$

$$\mathbf{A} = [\![\mathcal{G}; \mathrm{U}, \mathrm{V}, \mathrm{W}]\!].$$

We can now apply the QTT decomposition[4] to the block matrices $\mathrm{U}, \mathrm{V}, \mathrm{W}$ so that

$$\mathbf{A} = [\![\mathcal{G}; \mathrm{U}_1 \bowtie \cdots \bowtie \mathrm{U}_{\mathtt{L}}, \mathrm{V}_1 \bowtie \cdots \bowtie \mathrm{V}_{\mathtt{L}}, \mathrm{W}_1 \bowtie \cdots \bowtie \mathrm{W}_{\mathtt{L}}]\!], \tag{3.7}$$

where $\mathrm{U}_{\mathtt{L}}, \mathrm{V}_{\mathtt{L}}, \mathrm{W}_{\mathtt{L}}$ have R_1, R_2, R_3 block columns correspondingly. We call the decomposition (3.7) combined Tucker and QTT decomposition (TQTT). The graphical version of this decomposition is presented in Figure 2b.

EXAMPLE 3.2. *Let* $\mathbf{A} \in \mathbb{R}^{2^{3\mathtt{L}} \times 2^{3\mathtt{L}}}$ *be given as*

$$\mathbf{A} = \mathbf{Q} \otimes \mathbf{P} \otimes \mathbf{P} + \mathbf{P} \otimes \mathbf{Q} \otimes \mathbf{P} + \mathbf{P} \otimes \mathbf{P} \otimes \mathbf{Q}, \quad \mathbf{Q}, \mathbf{P} \in \mathbb{R}^{2^{\mathtt{L}} \times 2^{\mathtt{L}}}. \tag{3.8}$$

We can represent $\mathbf{A}$ *in Tucker format with the factors*

$$\mathrm{U} = \mathrm{V} = \mathrm{W} = \begin{bmatrix} \mathbf{Q} & \mathbf{P} \end{bmatrix} \tag{3.9}$$

and the core $\mathcal{G} \in \mathbb{R}^{2 \times 2 \times 2}$

$$\mathcal{G}_{\alpha_1 \alpha_2 \alpha_3} = \begin{cases} 1, & \{\alpha_1 \alpha_2 \alpha_3\} \in \{\{122\}, \{212\}, \{221\}\}. \\ 0, & \text{otherwise.} \end{cases}$$

Suppose additionally, that both $\mathbf{Q}$ *and* $\mathbf{P}$ *allow QTT representation of rank one, i.e.,*

$$\mathbf{Q} = \mathbf{q}^{\otimes \mathtt{L}}, \quad \mathbf{P} = \mathbf{p}^{\otimes \mathtt{L}}, \quad \mathbf{q}, \mathbf{p} \in \mathbb{R}^{2 \times 2}.$$

Then to obtain the TQTT decomposition of $\mathbf{A}$ *we only need to find QTT decomposition of the factors* (3.9)*:*

$$\begin{bmatrix} \mathbf{Q} & \mathbf{P} \end{bmatrix} = \begin{bmatrix} \mathbf{q} & \mathbf{p} \end{bmatrix} \bowtie \begin{bmatrix} \mathbf{q} & \\ & \mathbf{p} \end{bmatrix} \bowtie \cdots \bowtie \begin{bmatrix} \mathbf{q} & \\ & \mathbf{p} \end{bmatrix} \bowtie \begin{bmatrix} \mathbf{q} & \\ & \mathbf{p} \end{bmatrix}.$$

Note that the last core has 2 *block columns.*

[4]Since block matrices are decomposed, the last core of QTT decomposition depends on the number of blocks.

4. Explicit QTT representation of inverses of symmetric tridiagonal Toeplitz matrices. The proposed derivative-free ADI method based on (2.5) requires solution of discretized one-dimensional elliptic problems. Specifically, we are interested in solving linear systems with the matrices of the form

$$\mathbf{S}^{(\mathsf{L})} = \begin{bmatrix} \alpha & -1 & & \\ -1 & \alpha & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & \alpha \end{bmatrix}_{2^{\mathsf{L}} \times 2^{\mathsf{L}}}, \quad \alpha > 2, \tag{4.1}$$

where, for example,

$$\alpha = 2 + \kappa^2 \left(2^{\mathsf{L}} + 1\right)^{-2} \tag{4.2}$$

arises in the FD discretization of $-u'' + \kappa^2 u = f$ with Dirichlet b.c. or

$$\alpha = \frac{2 + \frac{2}{3}\kappa^2 \left(2^{\mathsf{L}} + 1\right)^{-2}}{1 - \frac{1}{6}\kappa^2 \left(2^{\mathsf{L}} + 1\right)^{-2}} = 2 + \frac{\kappa^2 \left(2^{\mathsf{L}} + 1\right)^{-2}}{1 - \frac{1}{6}\kappa^2 \left(2^{\mathsf{L}} + 1\right)^{-2}}$$

arises in the FE discretization using piecewise linear basis functions. Due to the ill-conditioning of $\mathbf{S}^{(\mathsf{L})}$, we are not able to apply solvers such as AMEn [7] directly to this matrix. A possible way could be to use a robust solver for one-dimensional PDEs proposed in [1], which in turn requires modifications for large κ. Here we will, however, find an explicit representation of the inverse of $\mathbf{S}^{(\mathsf{L})}$, avoiding more expensive optimization-based algorithms. Moreover, thanks to the usage of the explicit representation of $\mathbf{S}^{(\mathsf{L})^{-1}}$, we will be able to find explicit representations of matrices $\mathbf{T}(\sigma)$ and $\mathbf{B}(\sigma)$ from (2.6) with small ranks.

The proposed approach to derive the QTT representation of $\mathbf{S}^{(\mathsf{L})^{-1}}$ is based on the explicit formula [21]

$$\left(\mathbf{S}^{(\mathsf{L})^{-1}}\right)_{ij} = \frac{1}{\sinh\theta \, \sinh(2^{\mathsf{L}}+1)\theta} \begin{cases} \sinh i\theta \, \sinh(2^{\mathsf{L}}+1-j)\theta, & 1 \le i \le j \le 2^{\mathsf{L}} \\ \sinh j\theta \, \sinh(2^{\mathsf{L}}+1-i)\theta, & 1 < j < i \le 2^{\mathsf{L}} \end{cases}, \tag{4.3}$$

where $\cosh\theta = \alpha/2$. The key observation we make is that using basic properties of hyperbolic functions we can write $\mathbf{S}^{(\mathsf{L})^{-1}}$ equivalently as

$$\left(\mathbf{S}^{(\mathsf{L})^{-1}}\right)_{ij} = \frac{\cosh\left(2^{\mathsf{L}}+1-|i-j|\right)\theta - \cosh\left(2^{\mathsf{L}}+1-(i+j)\right)\theta}{2\,\sinh\theta\,\sinh(2^{\mathsf{L}}+1)\theta}. \tag{4.4}$$

One can show[5] that the QTT ranks of the matrix

$$\left\{\cosh\left(2^{\mathsf{L}}+1-|i-j|\right)\theta\right\}_{i,j=1}^{2^{\mathsf{L}}} \tag{4.5}$$

are bounded by 3 and the ranks of the matrix

$$\left\{\cosh\left(2^{\mathsf{L}}+1-(i+j)\right)\theta\right\}_{i,j=1}^{2^{\mathsf{L}}} \tag{4.6}$$

by 2. As a result, the matrix $\mathbf{S}^{(\mathsf{L})^{-1}}$ can be explicitly represented with the ranks bounded by 5 since TT ranks of a sum are bounded from above by a sum of the ranks [24].

Note, however, that (4.3) and (4.4) cannot be used for large values of $(2^{\mathsf{L}}+1)\theta$ because of the exponential growth of hyperbolic functions. In particular, if α is chosen as in (4.2) and $0 < \kappa\left(2^{\mathsf{L}}+1\right)^{-1} \ll 1$, we get[6]

$$(2^{\mathsf{L}}+1)\theta \approx \kappa,$$

[5] We will show these facts for matrices that differ from (4.5) and (4.6) only by a constant multiplier, which does not affect rank values.

[6] For $\alpha = 2 + \kappa^2\left(2^{\mathsf{L}}+1\right)^{-2}$ we have $\sinh\theta = \sqrt{\cosh^2\theta - 1} = \sqrt{\alpha^2/4 - 1} \approx \kappa\left(2^{\mathsf{L}}+1\right)^{-1}$ if $\kappa\left(2^{\mathsf{L}}+1\right)^{-1} \ll 1$.

and $\sinh(2^{\mathrm{L}}+1)\theta \approx \exp(\kappa)/2$. Since in the ADI method shifts σ_k can be large, we represent (4.4) by dividing numerator and denominator by $\exp(2^{\mathrm{L}}+1)\theta$ to avoid the exponential growth:

$$\left(\mathbf{S}^{(\mathrm{L})^{-1}}\right)_{ij} = \frac{e^{-|i-j|\theta} + e^{|i-j|\theta-2(2^{\mathrm{L}}+1)\theta} - e^{-(i+j)\theta} - e^{(i+j)\theta-2(2^{\mathrm{L}}+1)\theta}}{2\,\sinh\theta\,(1+e^{-2(2^{\mathrm{L}}+1)\theta})}. \tag{4.7}$$

In order to find an explicit representation of $\mathbf{S}^{(\mathrm{L})^{-1}}$ using the latter formula, we introduce the following auxiliary matrices:

$$\begin{aligned} \mathbf{E}_\ell^{(\nwarrow)} &= \begin{bmatrix} 1 & \epsilon_{\ell-1} \\ \epsilon_{\ell-1} & \epsilon_\ell \end{bmatrix}, \quad \mathbf{E}_\ell^{(\nearrow)} = \begin{bmatrix} \epsilon_{\ell-1} & 1 \\ \epsilon_\ell & \epsilon_{\ell-1} \end{bmatrix}, \\ \mathbf{E}_\ell^{(\swarrow)} &= \begin{bmatrix} \epsilon_{\ell-1} & \epsilon_\ell \\ 1 & \epsilon_{\ell-1} \end{bmatrix}, \quad \mathbf{E}_\ell^{(\searrow)} = \begin{bmatrix} \epsilon_\ell & \epsilon_{\ell-1} \\ \epsilon_{\ell-1} & 1 \end{bmatrix}, \end{aligned} \tag{4.8}$$

where

$$\epsilon_\ell = e^{-2^\ell\theta}. \tag{4.9}$$

We also introduce

$$\mathbf{I} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{J} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathbf{P} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{4.10}$$

The following proposition is the main result of this section.

PROPOSITION 4.1. *Let* $\mathbf{S}^{(\mathrm{L})} = \mathrm{tridiag}(-1, \alpha, -1)$, $\alpha > 2$ *be tridiagonal Toeplitz matrix, then* $\mathbf{S}^{(\mathrm{L})^{-1}}$ *allows for the QTT representation with ranks* $5, 5, \ldots, 5$

$$\mathbf{S}^{(\mathrm{L})^{-1}} = \mathrm{T}_{\mathrm{L}} \bowtie \cdots \bowtie \mathrm{T}_2 \bowtie \mathrm{T}_1,$$

where

$$\mathrm{T}_{\mathrm{L}} = 2\begin{bmatrix} \mathbf{I} & \mathbf{J}+\epsilon_1\epsilon_L\mathbf{J}^\top & \epsilon_1\epsilon_L\mathbf{J}+\mathbf{J}^\top & -\mathbf{E}_{\mathrm{L}}^{(\nwarrow)} & -\mathbf{E}_{\mathrm{L}}^{(\searrow)} \end{bmatrix},$$

$$\mathrm{T}_\ell = 2\begin{bmatrix} \mathbf{I} & \mathbf{J}+\epsilon_1\epsilon_{\mathrm{L}+1}{\epsilon_\ell}^{-1}\mathbf{J}^\top & \epsilon_1\epsilon_{\mathrm{L}+1}{\epsilon_\ell}^{-1}\mathbf{J}+\mathbf{J}^\top & & \\ & \mathbf{E}_\ell^{(\swarrow)} & & & \\ & & \mathbf{E}_\ell^{(\nearrow)} & & \\ & & & \mathbf{E}_\ell^{(\nwarrow)} & \\ & & & & \mathbf{E}_\ell^{(\searrow)} \end{bmatrix}, \quad \ell = 2, \ldots, \mathrm{L}-1$$

$$\mathrm{T}_1 = \frac{2^{-\mathrm{L}}}{\sinh\theta\,(1+\epsilon_1\epsilon_{\mathrm{L}+1})} \begin{bmatrix} (1+\epsilon_1\epsilon_{\mathrm{L}+1})\,\mathbf{I} + \epsilon_0\,(1+\epsilon_{\mathrm{L}+1})\mathbf{P} \\ \epsilon_0\,\mathbf{E}_1^{(\swarrow)} \\ \epsilon_0\,\mathbf{E}_1^{(\nearrow)} \\ \epsilon_1\,\mathbf{E}_1^{(\nwarrow)} \\ \epsilon_1\,\mathbf{E}_1^{(\searrow)} \end{bmatrix}.$$

and $\cosh\theta = \alpha/2$. *Parameters* ϵ_ℓ *are defined in* (4.9) *and the auxiliary matrices – in* (4.8) *and* (4.10).

Let us make several remarks about the computational aspects of formulas in Proposition 4.1.

REMARK 4.1. *To avoid division of two small numbers, parameters* $(\epsilon_1\epsilon_{\mathrm{L}+1}{\epsilon_\ell}^{-1})$ *in* T_ℓ *should be calculated as* $\exp\left((-2-2^{\mathrm{L}+1}+2^\ell)\theta\right)$.

REMARK 4.2. *Let* $\alpha = 2+\delta^2$, *where* $\delta \geq 0$ *is some small number. If* θ *is calculated using* $\theta = \mathrm{arccosh}((2+\delta^2)/2)$, *we will numerically obtain* $\theta = 0$, *as* $\delta^2/2$ *reaches machine epsilon. Therefore, we use Taylor expansion at* $\delta = 0$ *instead:*

$$\theta = \delta - \frac{1}{24}\,\delta^3 + \frac{3}{640}\,\delta^5 - \frac{5}{7168}\,\delta^7 + \mathcal{O}\left(\delta^9\right).$$

To prove Proposition 4.1, we will first prove two auxiliary lemmas. For convenience, we independently find representations of the Toeplitz and Hankel parts of (4.7).

LEMMA 4.1. *For any constant* $\theta \in \mathbb{R}$*, the Hankel matrix*

$$\mathbf{H}^{(\mathtt{L})} = \left\{ e^{-(i+j)\theta} + e^{(i+j)\theta - 2(2^{\mathtt{L}}+1)\theta} \right\}_{i,j=1}^{2^{\mathtt{L}}}$$

has a QTT representation with ranks $2, 2, \ldots, 2$*:*

$$\mathbf{H}^{(\mathtt{L})} = \mathrm{H}_{\mathtt{L}} \bowtie \cdots \bowtie \mathrm{H}_2 \bowtie \mathrm{H}_1,$$

where

$$\mathrm{H}_1 = \begin{bmatrix} \epsilon_1 \mathbf{E}_1^{(\nwarrow)} \\ \epsilon_1 \mathbf{E}_1^{(\searrow)} \end{bmatrix}, \quad \mathrm{H}_\ell = \begin{bmatrix} \mathbf{E}_\ell^{(\nwarrow)} & \\ & \mathbf{E}_\ell^{(\searrow)} \end{bmatrix} \ \ell = 2, \ldots, \mathtt{L}-1, \quad \mathrm{H}_{\mathtt{L}} = \begin{bmatrix} \mathbf{E}_{\mathtt{L}}^{(\nwarrow)} & \mathbf{E}_{\mathtt{L}}^{(\searrow)} \end{bmatrix}. \tag{4.11}$$

Proof. First, let us find recursive formulas for matrices

$$\mathbf{E}_1^{(\ell)} = \left\{ e^{-(i+j)\theta} \right\}_{i,j=1}^{2^\ell} \quad \text{and} \quad \mathbf{E}_2^{(\ell)} = \left\{ e^{(i+j)\theta - 2^{\ell+1}\theta} \right\}_{i,j=1}^{2^\ell}, \quad \ell = 1, \ldots, L. \tag{4.12}$$

For $\ell = 2, \ldots, L$, we have

$$\mathbf{E}_1^{(\ell)} = \left[\begin{array}{c|c} \mathbf{E}_1^{(\ell-1)} & \mathbf{E}_1^{(\ell-1)} e^{-2^{(\ell-1)}\theta} \\ \hline \mathbf{E}_1^{(\ell-1)} e^{-2^{(\ell-1)}\theta} & \mathbf{E}_1^{(\ell-1)} e^{-2^\ell \theta} \end{array} \right] = \mathbf{E}_\ell^{(\nwarrow)} \otimes \mathbf{E}_1^{(\ell-1)}, \tag{4.13}$$

$$\mathbf{E}_2^{(\ell)} = \left[\begin{array}{c|c} \mathbf{E}_2^{(\ell-1)} e^{-2^\ell \theta} & \mathbf{E}_2^{(\ell-1)} e^{-2^{(\ell-1)}\theta} \\ \hline \mathbf{E}_2^{(\ell-1)} e^{-2^{(\ell-1)}\theta} & \mathbf{E}_2^{(\ell-1)} \end{array} \right] = \mathbf{E}_\ell^{(\searrow)} \otimes \mathbf{E}_2^{(\ell-1)}, \tag{4.14}$$

where

$$\mathbf{E}_\ell^{(\nwarrow)} = \begin{bmatrix} 1 & e^{-2^{(\ell-1)}\theta} \\ e^{-2^{(\ell-1)}\theta} & e^{-2^\ell \theta} \end{bmatrix}, \quad \mathbf{E}_\ell^{(\searrow)} = \begin{bmatrix} e^{-2^\ell \theta} & e^{-2^{(\ell-1)}\theta} \\ e^{-2^{(\ell-1)}\theta} & 1 \end{bmatrix}.$$

Note that from (4.12)

$$\mathbf{E}_1^{(1)} = e^{-2\theta}\, \mathbf{E}_1^{(\nwarrow)}, \qquad \mathbf{E}_2^{(1)} = \mathbf{E}_1^{(\searrow)}.$$

Finally,

$$\mathbf{H}^{(\mathtt{L})} = \mathbf{E}_1^{(\ell)} + e^{-2\theta}\, \mathbf{E}_2^{(\ell)} = \mathbf{E}_L^{(\nwarrow)} \otimes \cdots \otimes \left(e^{-2\theta}\, \mathbf{E}_1^{(\nwarrow)} \right) + \mathbf{E}_L^{(\searrow)} \otimes \cdots \otimes \left(e^{-2\theta}\, \mathbf{E}_1^{(\searrow)} \right) =$$

$$\begin{bmatrix} \mathbf{E}_L^{(\searrow)} & \mathbf{E}_L^{(\nwarrow)} \end{bmatrix} \bowtie \begin{bmatrix} \mathbf{E}_{L-1}^{(\nwarrow)} & \\ & \mathbf{E}_{L-1}^{(\searrow)} \end{bmatrix} \bowtie \cdots \bowtie \begin{bmatrix} \mathbf{E}_2^{(\nwarrow)} & \\ & \mathbf{E}_2^{(\searrow)} \end{bmatrix} \bowtie \begin{bmatrix} e^{-2\theta}\, \mathbf{E}_1^{(\nwarrow)} \\ e^{-2\theta}\, \mathbf{E}_1^{(\searrow)} \end{bmatrix},$$

which completes the proof. ∎

LEMMA 4.2. *For any constant* $\theta \in \mathbb{R}$*, the Toeplitz matrix*

$$\mathbf{K}^{(\mathtt{L})} = \left\{ e^{-|i-j|\theta} + e^{|i-j|\theta - 2(2^{\mathtt{L}}+1)\theta} \right\}_{i,j=1}^{2^{\mathtt{L}}} \tag{4.15}$$

has a QTT representation with ranks $3, 3, \ldots, 3$*:*

$$\mathbf{K}^{(\mathtt{L})} = \mathrm{K}_{\mathtt{L}} \bowtie \cdots \bowtie \mathrm{K}_2 \bowtie \mathrm{K}_1,$$

where

$$\mathrm{K}_1 = \begin{bmatrix} (1+\epsilon_1\epsilon_{L+1})\, \mathbf{I} + \epsilon_0\, (1+\epsilon_{L+1}) \mathbf{P} \\ \epsilon_0\, \mathbf{E}_1^{(\swarrow)} \\ \epsilon_0\, \mathbf{E}_1^{(\nearrow)} \end{bmatrix}$$

$$\mathrm{K}_\ell = \begin{bmatrix} \mathbf{I} & \mathbf{J} + \epsilon_1 \epsilon_{L+1} \epsilon_\ell^{-1}\, \mathbf{J}^\top & \epsilon_1 \epsilon_{L+1} \epsilon_\ell^{-1}\, \mathbf{J} + \mathbf{J}^\top \\ & \mathbf{E}_\ell^{(\swarrow)} & \\ & & \mathbf{E}_\ell^{(\nearrow)} \end{bmatrix}, \qquad \ell = \overline{2, L-1} \tag{4.16}$$

$$\mathrm{K}_L = \begin{bmatrix} \mathbf{I} & \mathbf{J} + \epsilon_1 \epsilon_L \mathbf{J}^\top & \epsilon_1 \epsilon_L \mathbf{J} + \mathbf{J}^\top \end{bmatrix}.$$

Proof. Let us introduce the notation

$$\begin{aligned}\mathbf{X}^{(\ell)} &= \left\{e^{-(i-j)\theta-2^\ell\theta}\right\}_{i,j=1}^{2^\ell}, \quad \ell = 1,\dots,\mathtt{L}.\\ \mathbf{K}^{(\ell)} &= \left\{e^{-|i-j|\theta}+e^{|i-j|\theta-2(2^\mathtt{L}+1)\theta}\right\}_{i,j=1}^{2^\ell}\end{aligned} \tag{4.17}$$

Then we have

$$\begin{aligned}\mathbf{K}^{(\ell)} &= \begin{bmatrix}\mathbf{K}^{(\ell-1)} & \\ & \mathbf{K}^{(\ell-1)}\end{bmatrix} + \begin{bmatrix} & \mathbf{X}^{(\ell-1)}\\ \mathbf{X}^{(\ell-1)\top} & \end{bmatrix} + \underbrace{e^{-2(2^\mathtt{L}+1)\theta+2^\ell\theta}}_{\epsilon_1\cdot\epsilon_{L+1}\cdot\epsilon_\ell^{-1}}\begin{bmatrix} & \mathbf{X}^{(\ell-1)\top}\\ \mathbf{X}^{(\ell-1)} & \end{bmatrix} =\\ &= \mathbf{I}\otimes\mathbf{K}^{(\ell-1)} + \left(\mathbf{J}+\epsilon_1\epsilon_{L+1}\epsilon_\ell^{-1}\mathbf{J}^\top\right)\otimes\mathbf{X}^{(\ell-1)} + \left(\epsilon_1\epsilon_{L+1}\epsilon_\ell^{-1}\mathbf{J}+\mathbf{J}^\top\right)\otimes\mathbf{X}^{(\ell-1)\top} =\\ &= \begin{bmatrix}\mathbf{I} & \mathbf{J}+\epsilon_1\epsilon_{L+1}\epsilon_\ell^{-1}\mathbf{J}^\top & \epsilon_1\epsilon_{L+1}\epsilon_\ell^{-1}\mathbf{J}+\mathbf{J}^\top\end{bmatrix} \bowtie \begin{bmatrix}\mathbf{K}^{(\ell-1)}\\ \mathbf{X}^{(\ell-1)}\\ \mathbf{X}^{(\ell-1)\top}\end{bmatrix}, \quad \ell = 2,\dots,L.\end{aligned} \tag{4.18}$$

Since

$$\mathbf{X}^{(\ell)} = \left[\begin{array}{c|c}\mathbf{X}^{(\ell-1)}\,e^{-2^{(\ell-1)}\theta} & \mathbf{X}^{(\ell-1)}\,e^{-2^\ell\theta}\\ \hline \mathbf{X}^{(\ell-1)} & \mathbf{X}^{(\ell-1)}\,e^{-2^{(\ell-1)}\theta}\end{array}\right] = \mathbf{E}_\ell^{(\swarrow)}\otimes\mathbf{X}^{(\ell-1)}, \quad \mathbf{E}_\ell^{(\swarrow)} = \begin{bmatrix}e^{-2^{(\ell-1)}\theta} & e^{-2^\ell\theta}\\ 1 & e^{-2^{(\ell-1)}\theta}\end{bmatrix},$$

and

$$\mathbf{X}^{(\ell)\top} = \mathbf{E}_\ell^{(\swarrow)\top}\otimes\mathbf{X}^{(\ell-1)\top} = \mathbf{E}_\ell^{(\nearrow)}\otimes\mathbf{X}^{(\ell-1)\top},$$

we obtain

$$\begin{bmatrix}\mathbf{K}^{(\ell)}\\ \mathbf{X}^{(\ell)}\\ \mathbf{X}^{(\ell)\top}\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \mathbf{J}+\epsilon_1\epsilon_{L+1}\epsilon_\ell^{-1}\mathbf{J}^\top & \epsilon_1\epsilon_{L+1}\epsilon_\ell^{-1}\mathbf{J}+\mathbf{J}^\top\\ & \mathbf{E}_\ell^{(\swarrow)} & \\ & & \mathbf{E}_\ell^{(\nearrow)}\end{bmatrix} \bowtie \begin{bmatrix}\mathbf{K}^{(\ell-1)}\\ \mathbf{X}^{(\ell-1)}\\ \mathbf{X}^{(\ell-1)\top}\end{bmatrix}, \quad \ell = 2,\dots,\mathtt{L}-1. \tag{4.19}$$

Using (4.15) and (4.17), for $\ell = 1$ we have

$$\begin{aligned}\mathbf{K}^{(1)} &= \begin{bmatrix}1 & e^{-\theta}\\ e^{-\theta} & 1\end{bmatrix} + e^{-2(2^\mathtt{L}+1)\theta}\begin{bmatrix}1 & e^{\theta}\\ e^{\theta} & 1\end{bmatrix} = (1+\epsilon_1\epsilon_{L+1})\,\mathbf{I} + \epsilon_0\,(1+\epsilon_{L+1})\mathbf{P},\\ \mathbf{X}^{(1)} &= \begin{bmatrix}e^{-2\theta} & e^{-3\theta}\\ e^{-\theta} & e^{-2\theta}\end{bmatrix} = \epsilon_0\,\mathbf{E}_1^{(\swarrow)},\\ \mathbf{X}^{(1)\top} &= \epsilon_0\,\mathbf{E}_1^{(\nearrow)}.\end{aligned} \tag{4.20}$$

To complete the proof, we apply (4.18) for $\ell = \mathtt{L}$, use (4.19) for $\ell = \mathtt{L}-1,\dots,2$ and finally utilize expressions from (4.20). □

Proof of Proposition 4.1. Note that

$$\mathbf{S}^{(\mathtt{L})^{-1}} = \frac{1}{2\,\sinh\theta\,(1+e^{-2(2^\mathtt{L}+1)\theta})}\left(\mathbf{K}^{(\mathtt{L})}+\mathbf{H}^{(\mathtt{L})}\right)$$

Then, using the explicit representation of a sum of two TT matrices [24], we obtain

$$\mathbf{K}^{(\mathtt{L})}+\mathbf{H}^{(\mathtt{L})} = \begin{bmatrix}\mathrm{K}_\mathtt{L} & \\ & \mathrm{H}_\mathtt{L}\end{bmatrix}\bowtie\cdots\bowtie\begin{bmatrix}\mathrm{K}_1 & \\ & \mathrm{H}_1\end{bmatrix}.$$

Lemmas 4.1 and 4.2 yield explicit formulas for K_ℓ, H_ℓ, $\ell = 1,\dots,\mathtt{L}$, which completes the proof. □

The proposed approach can be used for a general tridiagonal Toeplitz matrix. In case of a non-symmetric Toeplitz matrix, formula (4.3) is multiplied by a rank-1 term $\mathfrak{c}^{i-j}$ with some constant $\mathfrak{c}$ [9], which does not change the rank of the representation. To keep the presentation short we will address explicit formulas in the general case in future work.

5. TQTT representation of iteration matrices. Recall that one step of the iterative process can be written as (2.6):

$$\mathbf{u}_{k+1} = \mathbf{T}(\sigma_k)\,\mathbf{u}_k - 2\sigma_k^2\,\mathbf{B}(\sigma_k)\mathbf{f}^{(\mathrm{L})},$$

where

$$\mathbf{T}(\sigma) = \mathbf{I} - 2\sigma^2\,\mathbf{B}(\sigma)\,(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_3),$$
$$\mathbf{B}(\sigma) = (\boldsymbol{\Sigma}_3 + \sigma\,\mathbf{I})^{-1}\,(\boldsymbol{\Sigma}_2 + \sigma\,\mathbf{I})^{-1}\,(\boldsymbol{\Sigma}_1 + \sigma\,\mathbf{I})^{-1}.$$

We are now in the position to find the explicit TQTT representations of $\mathbf{T}(\sigma)$ and $\mathbf{B}(\sigma)$, since they can be expressed in terms of QTT decompositions of tridiagonal Toeplitz matrix inverse:

$$\mathbf{R}^{(\mathrm{L})} = \sigma\left[\mathbf{S}^{(\mathrm{L})} + \left(\frac{\kappa^2}{3} + \sigma\right)\mathbf{I}^{(\mathrm{L})}\right]^{-1}. \tag{5.1}$$

Indeed, using (2.5), we may write the iteration matrix $\mathbf{T}(\sigma)$ in the form

$$\begin{aligned}\mathbf{T}(\sigma) =& \mathbf{I}^{(\mathrm{L})}\otimes\mathbf{I}^{(\mathrm{L})}\otimes\mathbf{I}^{(\mathrm{L})} + 6\,\mathbf{R}^{(\mathrm{L})}\otimes\mathbf{R}^{(\mathrm{L})}\otimes\mathbf{R}^{(\mathrm{L})} - \\ & 2\left(\mathbf{I}^{(\mathrm{L})}\otimes\mathbf{R}^{(\mathrm{L})}\otimes\mathbf{R}^{(\mathrm{L})} + \mathbf{R}^{(\mathrm{L})}\otimes\mathbf{I}^{(\mathrm{L})}\otimes\mathbf{R}^{(\mathrm{L})} + \mathbf{R}^{(\mathrm{L})}\otimes\mathbf{R}^{(\mathrm{L})}\otimes\mathbf{I}^{(\mathrm{L})}\right).\end{aligned} \tag{5.2}$$

The following lemma shows that $\mathbf{T}(\sigma)$ allows explicit TQTT representation with TQTT ranks, all bounded by 5.

PROPOSITION 5.1. *For any $\sigma\in\mathbb{R}$ the iteration matrix $\mathbf{T}(\sigma)$ defined in (5.2) allows for the TQTT representation with Tucker ranks $2,2,2$ and QTT ranks of Tucker factors $5,5,\ldots,5,2$:*

$$\mathbf{T}(\sigma) = \left[\!\!\left[\mathcal{T};\ \mathrm{T}_{\mathrm{L}}\bowtie\cdots\bowtie\mathrm{T}_2\bowtie\widehat{\mathrm{T}}_1,\ \mathrm{T}_{\mathrm{L}}\bowtie\cdots\bowtie\mathrm{T}_2\bowtie\widehat{\mathrm{T}}_1,\ \mathrm{T}_{\mathrm{L}}\bowtie\cdots\bowtie\mathrm{T}_2\bowtie\widehat{\mathrm{T}}_1\right]\!\!\right],$$

where

$$\widehat{\mathrm{T}}_1 = \begin{bmatrix} & 2^{1-\mathrm{L}}\mathbf{I} \\ \sigma h_{\mathrm{L}}^2\,\mathrm{T}_1 & \mathbf{O} \\ & \mathbf{O} \\ & \mathbf{O}\end{bmatrix},\quad \mathbf{I} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix},\quad \mathbf{O} = \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}, \tag{5.3}$$

tensor $\mathcal{T}\in\mathbb{R}^{2\times2\times2}$:

$$\mathcal{T}_{\alpha_1\alpha_2\alpha_3} = \begin{cases} 6, & \{\alpha_1\alpha_2\alpha_3\} = \{111\}\\ 1, & \{\alpha_1\alpha_2\alpha_3\} = \{222\}\\ -2, & \{\alpha_1\alpha_2\alpha_3\}\in\{\{112\},\{121\},\{211\}\}\\ 0, & \text{otherwise},\end{cases} \tag{5.4}$$

and the block matrices $\mathrm{T}_1,\ldots,\mathrm{T}_{\mathrm{L}}$ are defined in Proposition 4.1 with θ such that

$$\cosh\theta = 1 + \frac{h_{\mathrm{L}}^2}{2}\left(\frac{\kappa^2}{3} + \sigma\right).$$

Proof. First, using (5.2) we notice that $\mathbf{T}(\sigma)$ allows the following Tucker decomposition:

$$\mathbf{T}(\sigma) = \left[\!\!\left[\mathcal{T};\ \begin{bmatrix}\mathbf{R}^{(\mathrm{L})} & \mathbf{I}^{(\mathrm{L})}\end{bmatrix},\ \begin{bmatrix}\mathbf{R}^{(\mathrm{L})} & \mathbf{I}^{(\mathrm{L})}\end{bmatrix},\ \begin{bmatrix}\mathbf{R}^{(\mathrm{L})} & \mathbf{I}^{(\mathrm{L})}\end{bmatrix}\right]\!\!\right],$$

with the tensor $\mathcal{T}\in\mathbb{R}^{2\times2\times2}$ defined in (5.4). According to (5.1) the matrix $\mathbf{R}^{(\mathrm{L})}$ can be written as

$$\mathbf{R}^{(\mathrm{L})} = \sigma h_{\mathrm{L}}^2\begin{bmatrix}\beta & -1 & & \\ -1 & \beta & \ddots & \\ & \ddots & \ddots & -1\\ & & -1 & \beta\end{bmatrix}^{-1}_{2^{\mathrm{L}}\times2^{\mathrm{L}}},\quad \beta = 2 + h_{\mathrm{L}}^2\left(\frac{\kappa^2}{3}+\sigma\right).$$

Consequently, it can be represented using Proposition 4.1:

$$\mathbf{R}^{(\mathtt{L})} = \mathrm{T}_{\mathtt{L}} \bowtie \cdots \bowtie \mathrm{T}_2 \bowtie (\sigma h_{\mathtt{L}}^2 \mathrm{T}_1). \tag{5.5}$$

Since the first principal block of T_ℓ is $2\mathbf{I}$, for all $\ell = 2, \ldots, \mathtt{L}$, we can represent the identity matrix $\mathbf{I}^{(\mathtt{L})}$ as

$$\mathbf{I}^{(\mathtt{L})} = \mathrm{T}_{\mathtt{L}} \bowtie \cdots \bowtie \mathrm{T}_2 \bowtie \begin{bmatrix} 2^{1-\mathtt{L}}\mathbf{I} \\ \mathbf{O} \\ \mathbf{O} \\ \mathbf{O} \\ \mathbf{O} \end{bmatrix}.$$

As a result, we have

$$\begin{bmatrix} \mathbf{R}^{(\mathtt{L})} & \mathbf{I}^{(\mathtt{L})} \end{bmatrix} = \mathrm{T}_{\mathtt{L}} \bowtie \cdots \bowtie \mathrm{T}_2 \bowtie \widehat{\mathrm{T}}_1,$$

where $\widehat{\mathrm{T}}_1$ is defined in (5.3), which completes the proof. ∎

Matrix $\mathbf{B}(\sigma)$ can be written as

$$\mathbf{B}(\sigma) = \sigma^{-3}\, \mathbf{R}^{(\mathtt{L})} \otimes \mathbf{R}^{(\mathtt{L})} \otimes \mathbf{R}^{(\mathtt{L})},$$

and has, therefore, Tucker ranks $1, 1, 1$ as the following proposition suggests.

PROPOSITION 5.2. *For any $\sigma \in \mathbb{R}$ the matrix $\mathbf{B}(\sigma)$ defined in (5.2) allows TQTT representation with Tucker ranks $1, 1, 1$ and QTT ranks of Tucker factors $5, 5, \ldots, 5, 1$:*

$$\mathbf{B}(\sigma) = \left[\!\!\left[1;\, \mathrm{T}_{\mathtt{L}} \bowtie \cdots \bowtie \mathrm{T}_2 \bowtie \left(h_{\mathtt{L}}^2 \mathrm{T}_1\right),\, \mathrm{T}_{\mathtt{L}} \bowtie \cdots \bowtie \mathrm{T}_2 \bowtie \left(h_{\mathtt{L}}^2 \mathrm{T}_1\right),\, \mathrm{T}_{\mathtt{L}} \bowtie \cdots \bowtie \mathrm{T}_2 \bowtie \left(h_{\mathtt{L}}^2 \mathrm{T}_1\right) \right]\!\!\right],$$

where block matrices $\mathrm{T}_1, \ldots, \mathrm{T}_{\mathtt{L}}$ are defined in Proposition 4.1 with θ:

$$\cosh\theta = 1 + \frac{h_{\mathtt{L}}^2}{2}\left(\frac{\kappa^2}{3} + \sigma\right).$$

Proof. The proof follows directly from the explicit representation (5.5) of $\mathbf{R}^{(\mathtt{L})}$. ∎

6. Rank-truncated ADI method in TQTT format. In Section 5, we have derived explicit TQTT representations of $\mathbf{T}(\sigma)$ and $\mathbf{B}(\sigma)$. One step of the iteration (2.6) requires matrix-vector multiplications with these matrices and one linear combination of vectors. Each of these operations lead to rank increase. In particular, a matrix-vector multiplication leads to TQTT representation with the product of ranks, while linear combination of two vectors leads to rank summation [5]. To avoid the rank growth one should apply rank truncation after each iteration, i.e.,

$$\begin{aligned} \widetilde{\mathbf{u}}_{k+1} &= \mathbf{T}(\sigma_k)\, \mathbf{u}_k - 2\sigma_k^2\, \mathbf{B}(\sigma_k)\mathbf{f}^{(\mathtt{L})}, \\ \mathbf{u}_{k+1} &= \mathfrak{T}_\varepsilon(\widetilde{\mathbf{u}}_{k+1}), \end{aligned}$$

where $\mathfrak{T}_\varepsilon$ reduces rank of the representation while maintaining relative accuracy ε in ℓ_2 norm. One could also consider the so-called *hard rank thresholding* when the rank truncation is performed by the maximum rank value. It allows us to avoid the rank growth and, hence, the complexity. However, it can also lead to divergence of the method if rank is chosen to be too small.

Rank growth. Denoting the maximal TQTT rank of a vector $\mathbf{u}$ by $r_{\mathrm{TQTT}}(\mathbf{u})$, we can obtain rank bound for $\widetilde{\mathbf{u}}_{k+1}$:

$$r_{\mathrm{TQTT}}(\widetilde{\mathbf{u}}_{k+1}) \le 5\left(r_{\mathrm{TQTT}}(\mathbf{u}_k) + r_{\mathrm{TQTT}}(\mathbf{f}^{(\mathtt{L})})\right),$$

as according to Propositions 5.1 and 5.2 both $\mathbf{T}(\sigma)$ and $\mathbf{B}(\sigma)$ allow for explicit TQTT representations with the maximal rank 5. If needed, to reduce the complexity, $\mathbf{T}(\sigma_k)\, \mathbf{u}_k$ and $\mathbf{B}(\sigma_k)\mathbf{f}^{(\mathtt{L})}$ can be rounded independently before their linear combination is assembled. The overall algorithm is summarized in Algorithm 6.1.

Algorithm 6.1 Rank-truncated derivative-free ADI method in TQTT format for (2.1)

Require: Right-hand size $\mathbf{f}^{(\mathtt{L})}$ and initial guess $\mathbf{u}_0$ in TQTT format, shift κ, truncation parameter ε, tolerance parameter δ for one cycle of ADI
Ensure: $\mathbf{u}$ – low-rank TQTT approximation to $\mathbf{u}^{(\mathtt{L})}$

Calculate shifts σ_k, $k = 0, 1, \ldots, N-1$ using (2.13) with μ, ν from (2.15) and N to be the smallest integer satisfying (2.12)
Set $\mathbf{u} \coloneqq \mathbf{u}_0$
Set err $\coloneqq 1$
for $m = 0, 1, 2, \ldots$ until converged **do**
    $\mathbf{w} \coloneqq \mathbf{u}$
    for $k = 0, 1, \ldots, N-1$ **do**
        Set $\widehat{\mathbf{u}} \coloneqq \mathbf{u}$
        Calculate $\mathbf{T}(\sigma_k)$ and $\mathbf{B}(\sigma_k)$ as suggested in Propositions 5.1 and 5.2
        $\mathbf{v}_1 \coloneqq \mathbf{T}(\sigma_k)\,\mathbf{u}$, $\mathbf{v}_2 \coloneqq \sigma_k^2\,\mathbf{B}(\sigma_k)\mathbf{f}^{(\mathtt{L})}$ using TQTT arithmetics or optimization-based algorithm
        (Optional): $\mathbf{v}_1 \coloneqq \mathfrak{T}_\varepsilon(\mathbf{v}_1)$, $\mathbf{v}_2 \coloneqq \mathfrak{T}_\varepsilon(\mathbf{v}_2)$
        $\mathbf{v} \coloneqq \mathbf{v}_1 - 2\mathbf{v}_2$ using TQTT arithmetics
        $\mathbf{u} \coloneqq \mathfrak{T}_\varepsilon(\mathbf{v})$
        err_cyc $\coloneqq \|\mathbf{u} - \widehat{\mathbf{u}}\|/\|\mathbf{u}\|$
        if err_cyc $\le \delta \cdot$ err **then** ▷ Reduces total number of iterations
            break
    err $\coloneqq \|\mathbf{u} - \mathbf{w}\|/\|\mathbf{u}\|$
    if err $\le C\varepsilon$ **then** ▷ $C = 2$ is a practical choice
        break

Stopping criterion. Since we avoid calculating actions of discretized second derivatives to vectors, we do not have the access to a residual calculation so as to choose when to interrupt the iteration. Therefore, as is indicated in Algorithm 6.1, we evaluate error between the inner loops, i.e. $\|\mathbf{u}_{(n+1)N} - \mathbf{u}_{nN}\|/\|\mathbf{u}_{(n+1)N}\|$. This quantity can indeed be used as a stopping criterion: due to (2.14), we have (assuming no truncation error is introduced in each iteration):

$$\|\mathbf{u}_{(n+1)N} - \mathbf{u}_{nN}\| \ge \|\mathbf{u}_{nN} - \mathbf{u}^{(\mathtt{L})}\| - \|\mathbf{u}_{(n+1)N} - \mathbf{u}^{(\mathtt{L})}\| \ge \left(1 - \widehat{\rho}\,(\mu, \nu)\right) \|\mathbf{u}_{nN} - \mathbf{u}^{(\mathtt{L})}\|.$$

For optimal parameters μ_*, ν_* (2.15)

$$\|\mathbf{u}_{nN} - \mathbf{u}^{(\mathtt{L})}\| \le \frac{1}{1 - \widehat{\rho}\,(\mu_*, \nu_*)} \|\mathbf{u}_{(n+1)N} - \mathbf{u}_{nN}\| \approx 2\|\mathbf{u}_{(n+1)N} - \mathbf{u}_{nN}\|. \tag{6.1}$$

Tolerance in the stopping criterion of the outer iteration is chosen to be $C\varepsilon$, where ε is a truncation parameter. This is done due to possible stagnation of convergence as error approaches ε. It can also be beneficial to run the method first with larger ε, and then to start with the obtained solution as an initial guess for the desired ε. Note also, that there is an additional stopping criterion in the inner loop. It is used to reduce the total number of iterations, as the bound (2.14) is suboptimal.

Complexity. Complexity of one iteration of the method is dominated by the truncation operation and, thus, is

$$\mathcal{O}(\mathtt{L} r_{\mathrm{QTT}}^3 + r_{\mathrm{T}}^4),$$

where

$$r_{\mathrm{QTT}} = r_{\mathrm{QTT}}(\mathbf{u}_k) + r_{\mathrm{QTT}}(\mathbf{f}^{(\mathtt{L})}), \quad r_{\mathrm{T}} = r_{\mathrm{T}}(\mathbf{u}_k) + r_{\mathrm{T}}(\mathbf{f}^{(\mathtt{L})}),$$

with $r_{\mathrm{T}}(\mathbf{u})$, $r_{\mathrm{QTT}}(\mathbf{u})$ being correspondingly maximal ranks of the Tucker core and QTT modes of the TQTT representation of a vector $\mathbf{u}$.

7. Numerical experiments. The implementation is done using an open source TT-Toolbox [25] library, which contains the implementation of the two-level QTT Tucker format [5]. In this format, Tucker core is additionally decomposed using the TT decomposition. Assuming that Tucker ranks are R_1, R_2, R_3, this results in storing two additional matrices of sizes $R_1 \times \min\{R_1, R_2\}$ and $\min\{R_2, R_3\} \times$

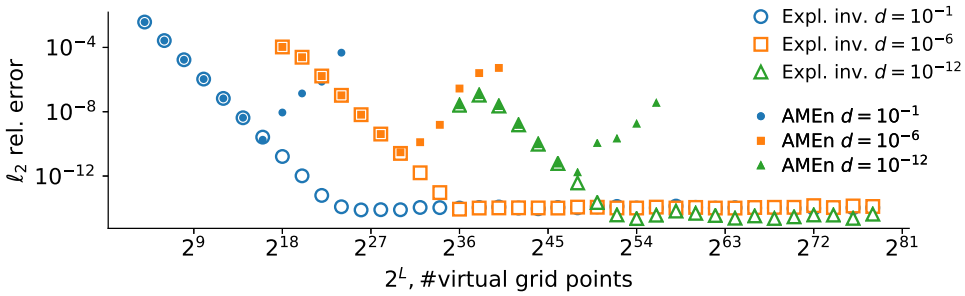


Fig. 3: Relative ℓ_2 error w.r.t. number of virtual grid points 2^{L} for the singularly perturbed problem (7.1) with the perturbation parameter d. Unfilled markers correspond to solutions obtained using the proposed explicit inversion formulas (Prop. 4.1), while filled markers correspond to solutions obtained by solving linear systems using the AMEn solver.

R_3, which are the first and the last cores of the TT-decomposition respectively. Their presence does not affect the overall scaling of the proposed algorithm as compared with the TQTT format and the influence on performance is negligibly small.

Numerical tests were performed on Intel Core i7 2.8 GHz processor with 16GB of RAM.

7.1. Singularly perturbed problem in one dimension. To verify that the explicit inversion formulas, derived in Proposition 4.1, are robust when applied to (5.1) for large σ, we consider the following one-dimensional singularly perturbed problem

$$\begin{aligned} -d^2 u'' + u &= 1, \quad \text{in} \quad (0,1), \\ u(0) &= u(1) = 0, \end{aligned} \tag{7.1}$$

where d is a small perturbation parameter. The exact solution to (7.1) can be found analytically:

$$u(x) = 1 - \frac{e^{-x/d} + e^{(x-1)/d}}{1 + e^{-1/d}}. \tag{7.2}$$

We discretize (7.1) using the finite difference method on a uniform grid $\omega^{(\mathrm{L})} = \{jh_{\mathrm{L}} \colon j = 1, \ldots, 2^{\mathrm{L}}\}$, $h_{\mathrm{L}} = (2^{\mathrm{L}} + 1)^{-1}$. To represent (7.2) on $\omega^{(\mathrm{L})}$ in the QTT format we utilize the fact that the arising exponents allow explicit rank-1 representations:

$$\begin{aligned} \left\{ e^{-x_i/d} \right\}_{i=1}^{2^{\mathrm{L}}} &= \begin{bmatrix} 1 \\ e^{-2^{\mathrm{L}-1} h_{\mathrm{L}}/d} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ e^{-2^{\mathrm{L}-2} h_{\mathrm{L}}/d} \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} 1 \\ e^{-2^{0} h_{\mathrm{L}}/d} \end{bmatrix} e^{-h_{\mathrm{L}}/d}, \\ \left\{ e^{(x_i-1)/d} \right\}_{i=1}^{2^{\mathrm{L}}} &= \begin{bmatrix} e^{-2^{\mathrm{L}-1} h_{\mathrm{L}}/d} \\ 1 \end{bmatrix} \otimes \begin{bmatrix} e^{-2^{\mathrm{L}-2} h_{\mathrm{L}}/d} \\ 1 \end{bmatrix} \otimes \cdots \otimes \begin{bmatrix} e^{-2^{0} h_{\mathrm{L}}/d} \\ 1 \end{bmatrix} e^{-h_{\mathrm{L}}/d}. \end{aligned}$$

The discretized problem reads

$$\left(\begin{bmatrix} 2 & -1 & & \\ -1 & 2 & \ddots & \\ & \ddots & \ddots & -1 \\ & & -1 & 2 \end{bmatrix} + \left(\frac{h_{\mathrm{L}}}{d} \right)^2 \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \right) \mathbf{u}^{(\mathrm{L})} = \left(\frac{h_{\mathrm{L}}}{d} \right)^2 \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}_{2^{\mathrm{L}}}. \tag{7.3}$$

To solve it, we apply two strategies. The first strategy is to assemble the FD matrix from (7.3) using explicit formulas [15] and to solve the linear system using the alternating minimal energy solver

(AMEn) [7]. The second strategy is to multiply the explicit inverse of the system matrix (Proposition 4.1) by a vector of the right-hand side. In Figure 3, we plot relative ℓ_2 error w.r.t. the exact solution against the number of virtual grid points $2^{\mathtt{L}}$ for these two strategies and different perturbation parameters d. We observe that with the proposed method we are able to use a large number of virtual grid points without stability problems. By contrast, solutions obtained using AMEn start being unreliable beginning from a certain number of grid points (depending on the accuracy parameter). We note, however, that this is not an issue of the AMEn solver itself, but is due to the ill-posedness of the problem.

7.2. Poisson's equation in three space dimensions. As a next example, consider the three-dimensional Poisson's equation

$$\begin{aligned} -\nabla^2 u &= f \quad \text{in} \quad \Omega = (0,1)^3, \\ u|_{\partial\Omega} &= 0. \end{aligned} \tag{7.4}$$

First, let us consider the case with a known exact solution. We choose f so that the exact solution is

$$u(x,y,z) = \frac{\sin \pi x \, \sin \pi y \, \sin \pi z}{1+x+y+z}. \tag{7.5}$$

Problem (7.4) is discretized as is suggested in (2.1). The right-hand side is assembled using the cross approximation algorithm [28], implemented as a function `multifuncrs` in the TT-Toolbox, and then transformed into the TQTT format. We set the tolerance parameter $\varepsilon = 10^{-9}$ for all the tested methods: the proposed ADI method, AMEn solver (`amen_solve2` function) for QTT and the solver for TQTT based on the density matrix renormalization group (DMRG) approach [5] (`dmrg_rake_solve2` function of TT-Toolbox). To improve convergence of AMEn and DMRG solvers we increased the parameter `max_full_size` to `5000`. The implementation of the proposed ADI algorithm is according to Algorithm 6.1. Matrix-vector products are performed by rounding explicit representations instead of optimization-based algorithms since the considered right-hand side function has low TQTT ranks.

Figure 4 illustrates the convergence to the exact solution and running times for the proposed method, AMEn solver and DMRG solver. The latter two solvers are applied to the explicit representation of the discretized equation assembled according to [15]. The proposed ADI solver is robust on the whole range of considered grid levels `L`. By contrast, due to ill-conditioning of the problem both AMEn and DMRG struggle to approximate the solution on fine grids.

We also consider the case

$$f \equiv 1,$$

when the exact solution is not known analytically. In Figure 5 we present internal convergence of the aforementioned methods and their running times. For DMRG and AMEn solvers internal convergence is measured using residual, while the convergence of the ADI method is according to (6.1). Remarkably, the convergence of the ADI method on this example does not depend on the number of grid levels `L`.

7.3. Screened Poisson's equation with singular right-hand side. In this section, we consider a model problem arising in electronic structure computations. In particular, we solve the screened Poisson equation with the singular right-hand side, which emerges in iterative processes in Schrödinger-type equations to calculate electron structure [12, 31]:

$$\begin{aligned} -\nabla^2 u + u &= 2r^{-1}e^{-r} \quad \text{in} \quad \Omega = (-40,40)^3, \\ u|_{\partial\Omega} &= 0, \end{aligned} \tag{7.6}$$

where $r = \sqrt{x^2+y^2+z^2}$. Thanks to the exponential decay of e^{-r}, the solution to (7.6) can be accurately approximated by

$$u \approx e^{-r}.$$

Indeed, the pointwise error introduced on the boundary $\partial\Omega$ is bounded by $\exp(-40) \approx 4.2 \cdot 10^{-19}$.

The discretization with $2^{3\mathtt{L}}$ internal grid points used in Section 2 allows us to avoid evaluation of the right-hand size at $(0,0,0)$, where r^{-1} is unbounded. Note that since the right-hand side of (7.6)

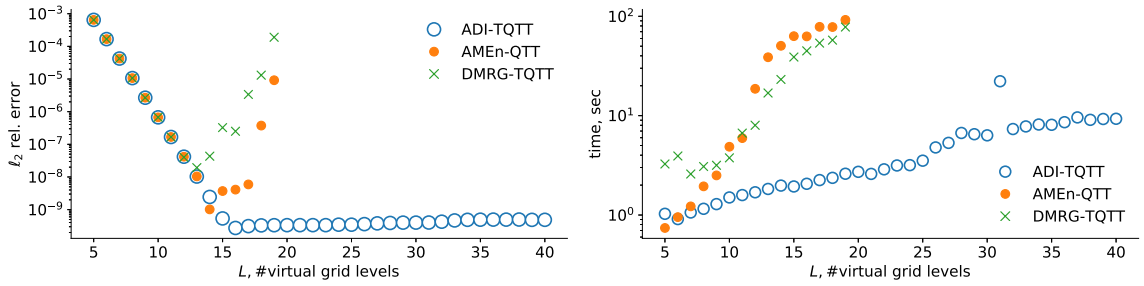


Fig. 4: Relative ℓ_2 error (left) and running times (right) w.r.t. number of virtual grid levels $\mathtt{L}$ for solving (7.4) using different methods. RHS f is chosen so that (7.5) is the exact solution; $\varepsilon = 10^{-9}$.

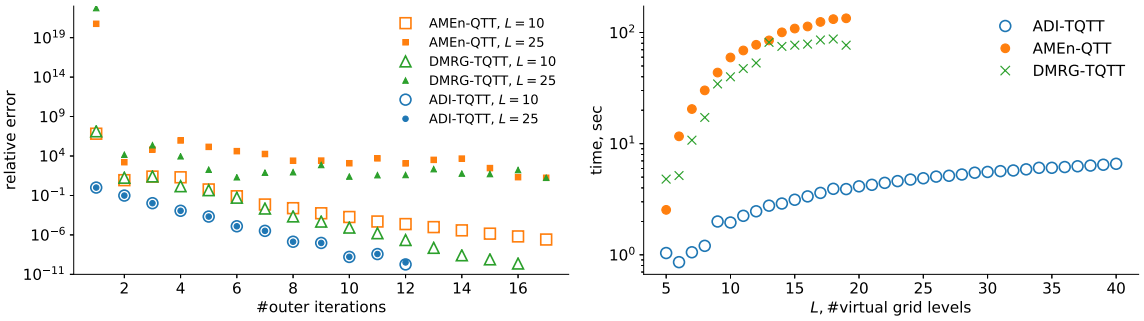


Fig. 5: Relative error in the considered methods w.r.t. number of outer iterations (left) and running times w.r.t. number of virtual grid levels $\mathtt{L}$ (right) for solving (7.4) with $f \equiv 1$.

is singular, fine virtual meshes must be used to obtain accurate solutions. In order to get a TQTT representation of the discretized $r^{-1}e^{-r}$, we use exponential sums to approximate r^{-1}. In particular, we obtain them by applying the trapezoidal rule to the following integral [2]

$$r^{-\beta} = \frac{1}{\Gamma(\beta)} \int_{-\infty}^{\infty} e^{-re^t + \beta t}\, dt, \tag{7.7}$$

for $\beta = 1/2$.

We compare the proposed method with the one based on matrix diagonalization with the help of discrete Fourier transform, which is referred to as FFT solver in plots. Thanks to the exponential decay of the exact solution, we replace zero Dirichlet with periodic boundary conditions. For periodic boundary conditions, the matrix of the linear system can be diagonalized using the discrete Fourier transform, which is available for both QTT and TQTT formats [6, 5] (function `rake_fft` in the TT-Toolbox). Thus, the solution can be obtained by two Fourier transforms and one elementwise division by a vector $\boldsymbol{\lambda}$ of the eigenvalues:

$$\boldsymbol{\lambda}_{\overline{ijk}} = \lambda_i + \lambda_j + \lambda_k + 1, \quad \lambda_i = \frac{4}{h_{\mathtt{L}}^2} \sin^2 \frac{\pi(i-1)}{2^{\mathtt{L}}}, \quad i, j, k = 1, \ldots, 2^{\mathtt{L}}.$$

The elementwise inverse of $\boldsymbol{\lambda}$ is obtained by using the trapezoidal rule for (7.7) and $\beta = 1$. QTT representations of arising exponents are obtained using the cross interpolation approach [28] with initial guesses obtained from approximation on nearby grid points of the trapezoidal discretization. The number of terms was adapted to the rounding parameter ε to speed up computations.

In Figure 6, we present approximation errors for the both methods for different values of rounding parameter ε. We observe that the FFT-TQTT method also allows producing accurate results for a wide

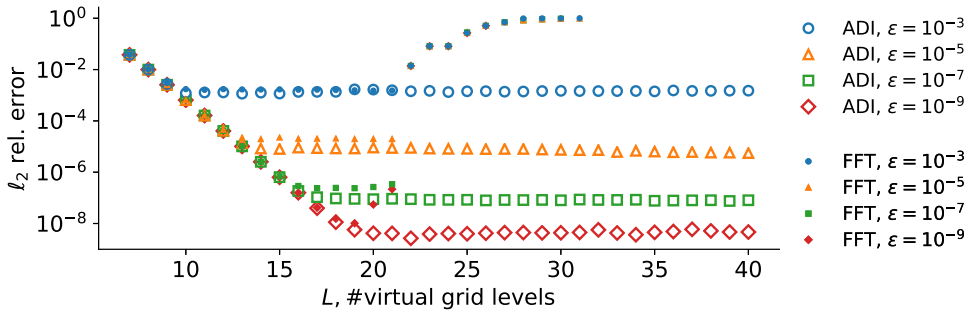


Fig. 6: Relative error of FD approximations to u for (7.6) against the number of virtual grid levels $\mathtt{L}$, obtained using the ADI method and the method based on direct diagonalization using discrete Fourier transform. Both methods utilize the TQTT format.

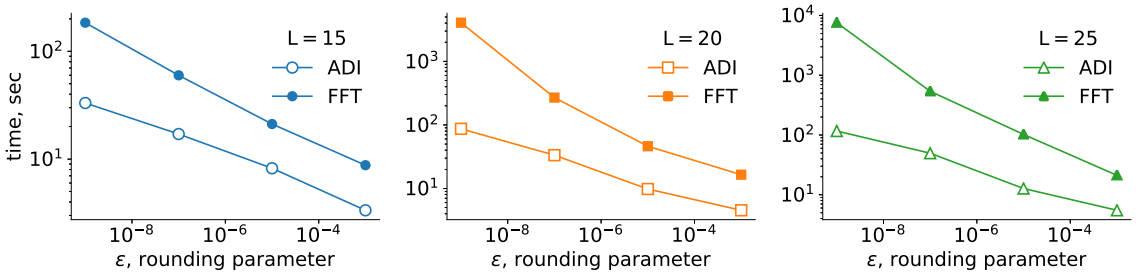


Fig. 7: Running times to solve (7.6) against the rounding parameter ε for the different number of virtual grid levels: $\mathtt{L} = 15, 20, 25$.

range of grid levels. Nonetheless, beginning from $\mathtt{L} = 20$, the error of the FFT-TQTT method starts increasing. We connect the observed growth with inaccuracies in obtaining the QTT representation of exponents in exponential sums for the elementwise inverse of $\boldsymbol{\lambda}$. Even though we used initial guess for the cross interpolation procedure, we were not able to improve the approximation quality even at the cost of significantly increasing the number of internal iterations and accuracy of the cross interpolation method. Note that this problem did not appear in exponential sums for the right-hand side $r^{-1}e^{-r}$.

In Figures 7 and 8, we present running times for the both methods for several values of grid levels $\mathtt{L}$. We observe that the proposed ADI-TQTT method is consistently faster than the FFT-TQTT for all range of accuracies. This holds also for $\mathtt{L} = 15, 20$ when there are no instabilities in assembling elementwise inverse of $\boldsymbol{\lambda}$. As an example, for $\mathtt{L} = 20$ and $\varepsilon = 10^{-7}$, the proposed method takes approximately 30 sec, while the FFT-TQTT approach takes approximately 5 minutes of running. The difference becomes even more pronounced for higher accuracies and number of grid levels.

We also note that even though we were able to accurately approximate the solution already for $\mathtt{L} = 20$ grid levels, usage of $\mathtt{L} > 20$ may be useful in quantum chemistry applications for large molecules and molecule clusters. We plan to investigate this question for more complex equations and physical systems in future work.

8. Conclusions and future work. We proposed a robust solver in a quantized tensor format based on the alternating direction implicit method for the screened Poisson's equation. The proposed method is capable of producing accurate solutions for a wide range of virtual grid levels. With the provided implementation, we were able to solve the considered equations for $\mathtt{L} = 40$ and moderate accuracies within a minute of computational time on a laptop.

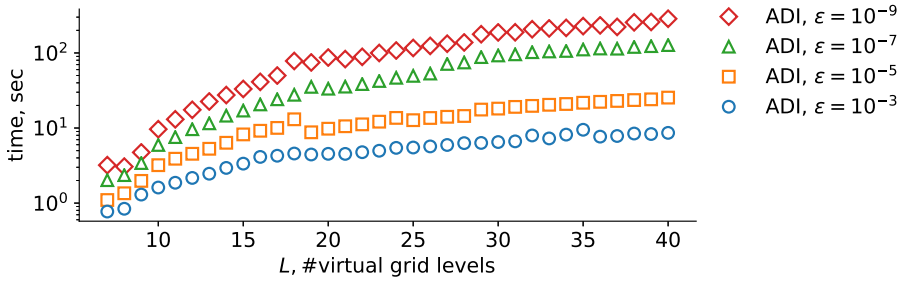


Fig. 8: Running times to solve (7.6) against number of virtual grid levels L for different rounding parameters ε using the proposed ADI method.

It is of interest to use the proposed solver as a building block to solve Schrödinger-type equations. For example, one can utilize it to solve Hartree-Fock or density functional theory equations, which are three-dimensional non-linear eigenvalue problems. Application of the solver to the multidimensional Schödinger's equation is also possible, assuming that shift parameters for the ADI iteration are available.

As an auxiliary result, we derived explicit formulas for the QTT representation of the inverses of tridiagonal Toeplitz matrices. Although we derived it only for the particular case of a tridiagonal Toeplitz matrix, the proposed strategy is straightforwardly generalizable. The consideration of general tridiagonal Toeplitz matrices, as well as matrices arising from the discretization with other boundary conditions is within the scope of future work.

Acknowledgements. The author is thankful to Carlo Marcati for his helpful comments on an early draft of the manuscript.

The work was supported by ETH Grant ETH-44 17-1.

REFERENCES

[1] M. Bachmayr and V. Kazeev, *Stability of low-rank tensor representations and structured multilevel preconditioning for elliptic PDEs*, arXiv preprint arXiv:1802.09062, (2018).
[2] G. Beylkin and L. Monzón, *Approximation by exponential sums revisited*, Appl. Comput. Harm. Anal., 28 (2010), pp. 131–149, https://doi.org/10.1016/j.acha.2009.08.011.
[3] A. V. Chertkov, I. V. Oseledets, and M. V. Rakhuba, *Robust discretization in quantized tensor train format for elliptic problems in two dimensions*, arXiv preprint 1612.01166, 2016, http://arxiv.org/abs/1612.01166.
[4] A. Cichocki, N. Lee, I. Oseledets, A.-H. Phan, Q. Zhao, D. P. Mandic, et al., *Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions*, Foundations and Trends in Machine Learning, 9 (2016), pp. 249–429, https://doi.org/10.1561/2200000059.
[5] S. Dolgov and B. Khoromskij, *Two-level QTT-Tucker format for optimized tensor calculus*, SIAM J. on Matrix An. Appl., 34 (2013), pp. 593–623, https://doi.org/10.1137/120882597.
[6] S. V. Dolgov, B. N. Khoromskij, and D. V. Savostyanov, *Superfast Fourier transform using QTT approximation*, J. Fourier Anal. Appl., 18 (2012), pp. 915–953, https://doi.org/10.1007/s00041-012-9227-4.
[7] S. V. Dolgov and D. V. Savostyanov, *Alternating minimal energy methods for linear systems in higher dimensions*, SIAM J. Sci. Comput., 36 (2014), pp. A2248–A2271, https://doi.org/10.1137/140953289.
[8] J. Douglas, *Alternating direction methods for three space variables*, Numer. Math., 4 (1962), pp. 41–63.
[9] M. Dow, *Explicit inverses of Toeplitz and associated matrices*, ANZIAM J., 44 (2008), pp. 185–215.
[10] L. Grasedyck, *Polynomial approximation in hierarchical Tucker format by vector-tensorization*, DFG-SPP1324 Preprint 43, Philipps-Univ., Marburg, 2010, http://www.dfg-spp1324.de/download/preprints/preprint043.pdf.
[11] L. Grasedyck, D. Kressner, and C. Tobler, *A literature survey of low-rank tensor approximation techniques*, GAMM-Mitt., 36 (2013), pp. 53–78, https://doi.org/10.1002/gamm.201310004.
[12] R. J. Harrison, G. I. Fann, T. Yanai, Z. Gan, and G. Beylkin, *Multiresolution quantum chemistry: Basic theory and initial applications*, J. Chem. Phys., 121 (2004), pp. 11587–11598.
[13] V. Kazeev, I. Oseledets, M. Rakhuba, and C. Schwab, *QTT-finite-element approximation for multiscale problems I: model problems in one dimension*, Adv. Comp. Math., (2016), https://doi.org/10.1007/s10444-016-9491-y, http://www.sam.math.ethz.ch/reports/2016/06.
[14] V. Kazeev and C. Schwab, *Quantized tensor-structured finite elements for second-order elliptic PDEs in two*

dimensions, Numer. Math., 138 (2018), pp. 133–190.
[15] 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, https://doi.org/10.1137/100820479.
[16] V. Khoromskaia and B. N. Khoromskij, *Tensor numerical methods in quantum chemistry*, Walter de Gruyter GmbH & Co KG, 2018.
[17] B. N. Khoromskij, $\mathcal{O}(d \log n)$*–Quantics approximation of* N*–*d *tensors in high-dimensional numerical modeling*, Constr. Approx., 34 (2011), pp. 257–280, https://doi.org/10.1007/s00365-011-9131-1.
[18] B. N. Khoromskij, *Tensor numerical methods for multidimensional PDEs: theoretical analysis and initial applications*, ESAIM: Proc., 48 (2015), pp. 1–28, https://doi.org/10.1051/proc/201448001.
[19] B. N. Khoromskij, *Tensor numerical methods in scientific computing*, vol. 19, Walter de Gruyter GmbH & Co KG, 2018.
[20] T. Mach and J. Saak, *Towards an ADI iteration for tensor structured equations*, MPI Magdeburg Preprint, 12 (2011).
[21] G. Meurant, *A review on the inverse of symmetric tridiagonal and block tridiagonal matrices*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 707–728.
[22] I. V. Oseledets, *Approximation of matrices with logarithmic number of parameters*, Doklady Math., 428 (2009), pp. 23–24, https://doi.org/10.1134/S1064562409050056.
[23] I. V. Oseledets, *Approximation of* $2^d \times 2^d$ *matrices using tensor decomposition*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2130–2145, https://doi.org/10.1137/090757861.
[24] I. V. Oseledets, *Tensor-train decomposition*, SIAM J. Sci. Comput., 33 (2011), pp. 2295–2317, https://doi.org/10.1137/090752286.
[25] I. V. Oseledets, S. Dolgov, V. Kazeev, D. Savostyanov, O. Lebedeva, P. Zhlobich, T. Mach, and L. Song, *TT-Toolbox*, 2011, https://github.com/oseledets/TT-Toolbox. https://github.com/oseledets/TT-Toolbox.
[26] I. V. Oseledets, M. V. Rakhuba, and A. V. Chertkov, *Black-box solver for multiscale modelling using the QTT format*, in Proc. ECCOMAS, Crete Island, Greece, 2016, https://www.eccomas2016.org/proceedings/pdf/10906.pdf.
[27] I. V. Oseledets, D. V. Savostyanov, and E. E. Tyrtyshnikov, *Cross approximation in tensor electron density computations*, Numer. Linear Algebra Appl., 17 (2010), pp. 935–952, https://doi.org/10.1002/nla.682.
[28] I. V. Oseledets and E. E. Tyrtyshnikov, *TT-cross approximation for multidimensional arrays*, Linear Algebra Appl., 432 (2010), pp. 70–88, https://doi.org/10.1016/j.laa.2009.07.024.
[29] D. W. Peaceman and H. H. Rachford, Jr, *The numerical solution of parabolic and elliptic differential equations*, J. SIAM, 3 (1955), pp. 28–41.
[30] M. V. Rakhuba and I. V. Oseledets, *Fast multidimensional convolution in low-rank tensor formats via cross approximation*, SIAM J. Sci. Comput., 37 (2015), pp. A565–A582, https://doi.org/10.1137/140958529.
[31] M. V. Rakhuba and I. V. Oseledets, *Grid-based electronic structure calculations: the tensor decomposition approach*, J. Comp. Phys., (2016), https://doi.org/10.1016/j.jcp.2016.02.023, http://arxiv.org/abs/1508.07632.
[32] V. Simoncini and D. B. Szyld, *Recent computational developments in krylov subspace methods for linear systems*, Numer. Linear Algebra Appl., 14 (2007), pp. 1–59.
[33] L. Tucker, *Implications of factor analysis of three-way matrices for measurement of change*, Problems in measuring change, (1963), pp. 122–137.
[34] E. Wachspress, *The ADI model problem*, Springer, 2013.