Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Fast solution of multi-dimensional parabolic problems in the TT/QTT-format with initial application to the Fokker-Planck equation

(revised version: February 2012)

by

Sergey Dolgov, Boris N. Khoromskij, and Ivan V. Oseledets

Preprint no.: 80 2011

Fast solution of multi-dimensional parabolic problems in the TT/QTT-format with initial application to the Fokker-Planck equation

S. V. Dolgov[1], B. N. Khoromskij[2] and I. V. Oseledets[3]

[1,2] *Max-Planck-Institut für Mathematik in den Naturwissenschaften, Inselstr. 22-26, D-04103 Leipzig, Germany*

[1,3] *Institute of Numerical Mathematics, Russian Academy of Sciences, Gubkina Street, 8, Moscow, Russia*

[1] sergey.v.dolgov@gmail.com, [2] bokh@mis.mpg.de,
[3] ivan.oseledets@gmail.com, http://spring.inm.ras.ru/osel

AMS Subject Classification: 35K20, 65F50, 15A69, 65D15 33F05, 65F10, 35Q84, 82D60
Key words: parabolic problems, QTT-format, DMRG, higher dimensions, tensor methods, Fokker-Planck equation, dumbbell model

Abstract. In this paper we propose two schemes of using the so-called QTT-approximation for the solution of multidimensional parabolic problems. First, we present a simple one-step implicit time integration scheme using an ALS-type solver in the QTT-format.

As the second approach, we use the global space-time formulation, resulting in a large block linear system, encapsulating all time steps, and solve it at once in the QTT format.

We prove the QTT-rank estimate for certain classes of multivariate potentials and respective solutions in (x, t) variables. The log-linear complexity of storage and the solution time is observed in both spatial and time grid sizes.

The method is applied to the Fokker-Planck equation arising from the beads-springs models of polymeric liquids.

1. Introduction

1.1. Problem setting

This paper is devoted to the solution of the semi-discrete parabolic problem of form

$$\frac{d\psi}{dt} = -A\psi + f(t), \quad \psi(0) = \psi_0, \quad t \geqslant 0, \tag{1}$$

[1,3] This work was partially supported by RFBR grants 09-01-12058, 10-01-00757, 11-01-00549, RFBR/DFG grant 09-01-91332, Russian Federation Gov. contracts No. П1178, П1112 and П940, 14.740.11.0345, and Promotionsstipendium by Max-Plank Institute. Part of this work was done during the stay of I. V. Oseledets in Max-Plank Institute for Mathematics in Sciences, Leipzig, Germany.

where A is a discretization of a high-dimensional elliptic operator in $\mathbb{R}^d$ on a tensor grid. In this case, the vector of unknowns ψ at each time step can be naturally considered as a d-dimensional array (tensor) $\Psi(i_1, \ldots, i_d), 1 \leqslant i_k \leqslant n_k$. For simplicity, assume that all *mode sizes* n_k are equal to n. The formal number of unknowns in this case behaves as n^d and is subject to the *curse of dimensionality*.

Such problems appear in several applications. Among them is the Fokker-Planck equation (it traces back, probably, to [1]).

A particular interest is connected with the polymeric liquid models which can be described by the Fokker-Planck equation. Various solution methods, based on the Monte-Carlo approach [2], direct treatment via the spectral methods for low-dimensional examples [3, 4], greedy algorithms to compute sums of separable tensor products [5, 6, 7, 8] (the so-called *canonical* tensor format, or *PARAFAC* [9, 10, 11]) were developed in the recent years to solve such problems. All of these methods use step-by-step time integration thus leading to the linear complexity scaling in time. A *global* time scheme in the QTT format developed in this paper suggests a new approach with the possibility of *logarithmic* time complexity (see also [12] for QTT-Cayley transform, and [13, 14] for the sparse grids approach applied in t-variable).

Note, that the choice of the particular low-parametric representation is crucial. The approximation by separable function, the so-called canonical format, is a good candidate. However, its usage is restricted by several drawbacks, among which the most important is that the approximation problem in the canonical format is ill-posed [15]. On the contrary, the tensor formats presented in this paper admit stable algebraic, rounding and solution routines, thus in the case the solution has the desired structure, the method will find it in a stable way.

The solution of (1) has to be performed by a certain time-propagation scheme. For simplicity, consider only one-step schemes with a constant time step τ,

$$\psi_{k+1} = S\psi_k + \widehat{f}_k, \tag{2}$$

where S is a time-propagator, $\psi_k \approx \psi(t_k)$ is the approximate solution at time $t_k = \tau k$, and $\widehat{f}_k$ is a discretization to the right-hand side. For example, $S = I - \tau A, \widehat{f}_k = \tau f(t_k)$ corresponds to the explicit Euler, $S = (I + \tau A)^{-1}$ to the implicit Euler, and $S = (I + \frac{\tau}{2}A)^{-1}(I - \frac{\tau}{2}A)$ to the Crank-Nicolson scheme, respectively. The implicit Euler and the Crank-Nicolson schemes require the solution of a linear system at each time step. The solution is in fact a high-dimensional array (tensor) with a large number of dimensions. It can not be stored as a full array, thus it should be approximated in a certain data-sparse way. This means, that ψ_k belongs to some class $\mathcal{S}$ of structured tensors. In order to work with such low-parametric representations, one needs a fast procedure for the computation of $S\psi_k + \widehat{f}_k$ (i.e., right-hand side of (2)), and some fast *truncation* operator T_ε, which projects the result to $\mathcal{S}$ with some accuracy ε. The modified time-stepping scheme therefore reads

$$\psi_{k+1} = T_\varepsilon(S\psi_k + \widehat{f}_k).$$

In high dimensions the main problem is not the construction of the appropriate time integrator (i.e. the operator S), since a standard scheme (e.g., the Crank-Nicolson scheme) can be used. The difficulty arises due to the large data arrays representing the approximate solution even on a single time step. In fact, the number of elements to be stored increases exponentially in the dimension d. In this way, the choice of the tensor format $\mathcal{S}$ is crucial.

1.2. Tensor formats

In this paper we use the so-called *tensor train format*, or simply TT-format [16, 17]. A d-dimensional tensor $\mathbf{A}$ is said to be in the TT-format, if its elements are represented as

$$A(i_1, \ldots, i_d) = G_1(i_1) G_2(i_2) \ldots G_d(i_d), \quad 1 \leqslant i_k \leqslant n_k, \tag{3}$$

where $G_k(i_k)$ is a $r_{k-1} \times r_k$ matrix for each fixed i_k. To make the matrix-by-matrix product in (3) scalar, *boundary conditions* $r_0 = r_d = 1$ are imposed. The numbers r_k, called *TT-ranks*, play the crucial role in storage and complexity estimates. Equation (3) can be written in the index form,

$$A(i_1, \ldots, i_d) = \sum_{\alpha_1, \ldots, \alpha_{d-1}} G_1(\alpha_1, i_1, \alpha_2) G_2(\alpha_1, i_2, \alpha_2) \ldots G_d(\alpha_{d-1}, i_d),$$

where the sum over α_k goes from 1 to r_k. For fixed values of $\mathbf{r} = [r_1, \ldots, r_d]$ the parametric representation (3) defines a embedded manifold $\mathbb{TT}_{\mathbf{r}}$ [18] in the linear space of all d-tensors. It is clear, that each of *TT-cores* $G_k(i_k)$ has $r_{k-1} n_k r_k$ elements, thus if all the ranks r_k are bounded by some constant r, and the mode sizes n_k by n, the storage is estimated as $\mathcal{O}(dnr^2)$.

On the other hand, a great development was made in the modeling of quantum spin many-body systems. The so-called density matrix renormalization group (DMRG) [19, 20, 21] is a numerical variational technique devised to obtain the low energy physics of quantum many-body systems with high accuracy. It traces back to [21], and it is nowadays the most efficient method for 1-dimensional quantum systems, but its generalization to multidimensional case[1] is still an open question. It was then noticed, that DMRG is a minimization method for the Rayleigh quotient in the Matrix Product States (MPS) [19], which also arise in the study of entanglement in quantum systems. The TT and MPS approximations are very close in form. The connection between TT and DMRG/ALS schemes was discussed, in particular, by R. Schneider et. al., see [18, 22].

Traditional and commonly used tensor representations in multilinear algebra and numerical analysis include canonical and Tucker formats, see the surveys and lecture notes [23, 24, 25]. The canonical rank-R format is the representation of form

$$A(i_1, \ldots, i_d) = \sum_{\alpha=1}^{R} U_1(i_1, \alpha) \ldots U_d(i_d, \alpha),$$

while the Tucker rank-$(r_1, ..., r_d)$ format is defined by

$$A(i_1, \ldots, i_d) = \sum_{\alpha_1, \ldots, \alpha_d} G(\alpha_1, \ldots, \alpha_d) U_1(i_1, \alpha_1) \ldots U_d(i_d, \alpha_d).$$

If the tensor has the canonical representation with rank R, then there exists a TT-representation with TT-ranks bounded by R (but they can be much smaller). The tensors with bounded canonical rank do not form a manifold, and algorithms for the computation of the best fixed-rank approximation are not robust (i.e. operator T_ε is not always well-defined). Thus this format can not be used in conjunction with the time-stepping

[1] By "multidimensional" system in the quantum information theory a more complicated *tensor networks* are meant, not the approximation of multidimensional tensors.

scheme, since the truncation has to be done at each step. The Tucker format can be used for small and medium values of d. In quantum molecular dynamics simulation the Tucker format was successfully used in the MCTDH framework (see the book [26]). The disadvantage of the Tucker format is the inherent exponential scaling in the dimension. In turn, the TT-format has linear scaling in the dimension, provided that the TT-ranks are bounded. Another alternative to the Tucker format might be the $\mathcal{H}$T format [27]. Some efficient methods arise from the combination of formats, e.g. the multilevel solver for the Hartree-Fock equation in the Tucker format with the canonical representation of the core [28, 29].

In this paper, along the line with the TT-format, the so-called *Quantized-TT (QTT)*-format is used. The idea is as follows. Suppose that the one-dimensional grid size is a power of 2, i.e. $n = 2^L$. Then, ψ_k can be reshaped to a $D = dL$-dimensional tensor with mode sizes equal to 2. Then, the TT-decomposition is applied to this tensor. If the TT-ranks of this D-tensor (or QTT-ranks of ψ_k) are small, then the *logarithmic complexity*, $\mathcal{O}(d \log n)$, is attained. The idea of QTT representation was first proposed in [30] for $2^L \times 2^L$ matrices, and it was generalized to the class of function-related tensors in [31], where its beneficial approximation properties were established. Moreover, the QTT-format allows simple constructive representations of basic operators (Laplacian, gradient and divergence operators) [32] on uniform tensor grids. The logarithmic dependence on the one-dimensional grid size make it a very promising tool for high-dimensional problems. There are also algorithms, which exploit the binary QTT structure heavily, such as the super-fast data-sparse Fourier transform [33]. The Cayley transform in the TT or QTT format [12] gives another approach to complex-time parabolic ("square root of hyperbolic") problems (e.g., the molecular Schrödinger equation). It is also important, that small mode sizes (2 for the QTT-format) allow the construction of a fast approximate DMRG-like solver for a QTT-structured matrix. Such a solver, the *TT-Solve* algorithm, was recently described in [34] (see also [35] for the eigenvalue solver of this type and [22], where the general ALS-type schemes are discussed). This solver is routinely utilized in our implicit time-stepping calculations.

1.3. Solution scheme

This paper aims to combine efficient operations in the TT-format with the standard time-stepping schemes for the parabolic problems. The solution scheme consists of three steps. First, the matrix of the problem A,the initial data ψ_0 and the right-hand side f is transformed into the TT-format. Sometimes such transition is obvious (for example, when the initial data is given in the canonical format). As a substep, one can try to transform the time propagator S to the TT-format, but it may not be an easy task. Second, the matrix-by-vector product $S\psi_k$ is implemented via fast TT-arithmetic (for the explicit scheme) or via the TT-Solve algorithm [34], which provides an efficient approach to solve linear systems handling both matrix and vectors in the TT-format. The third step is to project the current iterand to the set of TT-tensors with possibly smaller TT-ranks, while maintaining the prescribed accuracy ε.

On the other hand, the advantages of the QTT format motivate us to treat the problem from essentially new point of view: we consider the time as a independent dimension, introduce a discretization of the whole differential equation with all spatial and time dimension connected in one large linear system. Such scheme is usually never used in the

standard approaches to the time dependent problems, since the bidiagonality of the time-related part of the matrix admits the linear (the best possible in this case) complexity of the direct block elimination. But the QTT format promises a *logarithmic* complexity, and it could be reasonable to convert even a bidiagonal matrix to the corresponding tensor structure and use iterative methods.

1.4. Organization of the paper

The rest of the paper is organized as follows. In Section 2 the time discretization schemes and rank bounds are presented. Section 3 introduces the main application considered in this paper — the Fokker-Planck equation. We briefly discuss also other existing models and outline possible difficulties connected with tensor computations. Section 4 gives numerical experiments, manifesting the performance of the proposed methods, applied to the Fokker-Planck equation.

2. Discretization in time, solution methods and rank bounds

2.1. Time-stepping and global solution schemes

We propose and compare two time integration schemes: first, the implicit time stepping procedure with the solution of a linear system on each step with the TT-solve algorithm, second, one block system both for space and time, which is solved at once via the TT-solve. As a time-stepping scheme the Crank-Nicolson scheme is chosen. At each time step, the linear system is of form

$$\left(I + \frac{\tau}{2}A\right)\psi_{k+1} = \left(I - \frac{\tau}{2}A\right)\psi_k + \frac{\tau}{2}(f_k + f_{k+1}), \quad k = 0, ..., N_t - 1. \tag{4}$$

Provided the matrix A is nonnegative definite, it holds that the spectral norm, $\left\|\left(I + \frac{\tau}{2}A\right)^{-1}\right\| \leqslant 1$, and hence the linear system is well-posed [36]. If A is given in the TT or QTT format, then the system matrix $(I + \frac{\tau}{2}A)$ is also in the same format since the identity matrix has a perfect rank-1 structure. How to solve the linear system (4)? In order to do this, the alternating linear algorithm, TT-Solve [34] is used as a black-box approximate solver. Its complexity is linear in the dimension, polynomial in the mode size, and polynomial in the TT-ranks of the matrix and the approximate solution. It requires an initial approximation to the solution, and also the required relative accuracy (in the residual). To choose the initial approximation, the simplest scheme is to use ψ_k, the value from the previous step, as an initial guess. However, it may require a lot of inner iterations in the DMRG iterations. We found, that the usage of an explicit Euler scheme

$$\widehat{\psi}_k = (I - \tau A)\psi_k + \tau f_k.$$

is a much better choice.

To derive the second scheme, rewrite the equation (4) as follows:

$$\psi_{k+1} - \psi_k + \frac{\tau}{2}A\psi_{k+1} + \frac{\tau}{2}A\psi_k = \frac{\tau}{2}(f_k + f_{k+1}).$$

Now, it is clear how to write the global system:

$$\begin{bmatrix} I + \frac{\tau}{2}A & & & \\ -I + \frac{\tau}{2}A & I + \frac{\tau}{2}A & & \\ & \ddots & \ddots & \\ & & -I + \frac{\tau}{2}A & I + \frac{\tau}{2}A \end{bmatrix} \begin{bmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_{N_t} \end{bmatrix} = \begin{bmatrix} \left(I - \frac{\tau}{2}A\right)\psi_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} + \frac{\tau}{2} \begin{bmatrix} f_0 + f_1 \\ f_1 + f_2 \\ \vdots \\ f_{N_t - 1} + f_{N_t} \end{bmatrix}. \tag{5}$$

Such statement was also considered with the *sparse grids* approach to parameter reduction [13, 14], where the complexity of the solution has no (or logarithmic) dependence on the number of time steps. To achieve that, certain smoothness conditions have to be imposed, and $\log(h)$ term arises in the error in any case. In this work, we consider the QTT approximation of the system (5), therefore it requires

$$\mathcal{O}(\log(N_x)\log(N_t)r^2)$$

memory cells to store the data, and

$$\mathcal{O}(\log(N_x)\log(N_t)r^4)$$

operations to compute the solution (the first estimate arises from the TT structure itself [30, 31], the second one is the complexity of the TT-Solve algorithm [34]). If the QTT ranks are bounded, we have the logarithmic complexity *both* in space and time. On the other hand, we do not have to consider specially the error analysis, as the linear system is solved with controllable global accuracy in the adaptive procedure. The only issue is to estimate the rank bounds. A rigorous proof can be provided only in some special cases, but most of numerical experiments manifest reasonable rank values (e.g. independent on the number of time steps) for various relevant problems.

2.2. Time-related QTT rank bounds

The estimates stated above are productive only if the rank r remains bounded during the whole solution process. It is reasonable to expect it to be independent (or slightly dependent, say, logarithmic) on the spatial and time grid sizes. In certain cases, rank bounds can be obtained. The idea is based on the A-eigenbasis decomposition, as well as in [12]. In the following, we will always use the Euclidean (Frobenius) norm $\|\cdot\|$ for vectors and tensors.

Lemma 1. Suppose a (discrete) operator A possesses a complete set of orthogonal eigenvectors $\{\varphi_m\}$ with the QTT ranks of their ε-approximations bounded by r_m and corresponding eigenvalues $\{\lambda_m\}$, $\operatorname{Re}\lambda_m \geqslant 0$. Suppose the initial vector ψ_0 can be represented as a sum $\psi_0 = \sum_{m=1}^{N_x} \gamma_m \varphi_m$ with decaying coefficients $\{\gamma_m\}$ so that

$$\left\| \psi_0 - \sum_{m=1}^{M} \gamma_m \varphi_m \right\| \leqslant \varepsilon \|\psi_0\|$$

for some $M < N_x$ (a case of interest is of course $M \ll N_x$), and the right-hand side $f_k = 0,\ k = 0, ..., N_t$.

Then there exists the $(M+1)\varepsilon$-approximation to the global space-time solution $\begin{bmatrix}\psi_1 & \psi_2 & \cdots & \psi_{N_t}\end{bmatrix}^T$ of (5) with the ranks of the QTT blocks related to the time variable bounded by M, and of the blocks related to the spatial variables by $\sum_{m=1}^{M} r_m \leqslant M \max(r_m)$.

Proof. Denote the time transition operator $\left(I+\frac{\tau}{2}A\right)^{-1}\left(I-\frac{\tau}{2}A\right) = S$, then the solution at the time step k is given as

$$\psi_k = S^k \psi_0.$$

Clearly, the operator S has the same set of eigenvectors $\{\varphi_m\}$ as A and the eigenvalues

$$q_m = \frac{1-\frac{\tau}{2}\lambda_m}{1+\frac{\tau}{2}\lambda_m}, \quad |q_m| \leqslant 1$$

(notice that if there exists the full set of eigenvectors of A, the non-negativity condition in the Kellogg's lemma for (4) [36] coincides with our condition $\operatorname{Re}\lambda_m \geqslant 0$). Now, if the initial vector ψ_0 is projected onto the eigenbasis of A, the same decomposition persists at each time step k:

$$\psi_k = \sum_{m=1}^{N_x} q_m^k \gamma_m \varphi_m,$$

and moreover, as $|q_m| \leqslant 1$, its coefficients are not greater than γ_m, so that if ψ_0 was approximated by M terms with the accuracy ε, the accuracy $M\varepsilon + \varepsilon$ holds for ψ_k approximated by no more than M terms as well ($M\varepsilon$ arises since each of the eigenfunctions is also given by its approximation with the accuracy ε). Now the global solution reads

$$\begin{bmatrix}\psi_1 & \psi_2 & \cdots & \psi_{N_t}\end{bmatrix}^T \approx \sum_{m=1}^{M} \gamma_m \begin{bmatrix}q_m & q_m^2 & \cdots & q_m^{N_t}\end{bmatrix}^T \otimes \varphi_m,$$

i.e. the separation rank between the space and time is not greater than M. The time-related vector $\begin{bmatrix}q_m & q_m^2 & \cdots & q_m^{N_t}\end{bmatrix}^T$ is nothing else but the exponential function $\exp(t \ln q_m)$, which has all QTT ranks equal to 1 [31], so the same rank-M bound holds for combined space-time QTT-blocks.

The spatial QTT-blocks consist of the corresponding cores of the eigenfunctions $\{\varphi_m\}_{m=1}^M$, thus the ranks of the eigenfunctions are summed up, giving the second estimate of the lemma. □

Remark 1. For the heat-transfer equation

$$\begin{aligned}&\frac{\partial \psi}{\partial t} = -\Delta\psi \quad \text{in } [0,1]^d \times [0,\infty),\\ &\psi(x,0) = \psi_0(x)\end{aligned}$$

with the Dirichlet boundary conditions the eigenbasis is the sine-basis

$$\varphi_m(i) = \prod_{q=1}^{d} \sin\left(\frac{\pi m_q i_q}{N_x+1}\right),$$

with the ranks of the exact QTT decomposition not greater than 2 [31] (see also [37] concerning the explicit rank-2 representation). On the other hand, the decay of the Fourier coefficients γ_m is governed by the smoothness of a function:

$$\psi_0 \in C^p \Rightarrow |\gamma_m| \leqslant \mathcal{O}(m^{-p-1}),$$

so that $M = \mathcal{O}(\varepsilon^{-1/p})$, and the QTT-ranks of the space-time solution are bounded by $2M$.

Remark 2. Lemma 1 imposes some requirements on eigenfunctions of the spatial operator A, but not on the tensor structure of the operator itself. In general, the TT or QTT ranks of the matrix can be small, but the ranks of the solution may be large, or vise versa. As for the Fokker-Planck equation, especially with a non-scalar diffusion tensor $\{D_{ij}\}$, the eigenfunctions might have very large QTT-ranks, leading to the principal difficulty of solving such problems by tensor methods, despite that for certain models the TT-ranks of the stiffness matrix are shown to be bounded by 2, see Section 3.3.

2.3. Rank bound for the Gaussian stationary solution

The multidimensional Gaussian function is a prototype function for the solution of the Fokker-Planck equation. Thus, it is interesting to obtain rank bounds for this function. Let

$$f(x) = \prod_{k=1}^{d} \exp\left(-\frac{x_k^2}{2p_k^2}\right). \tag{6}$$

Since the multidimensional Gaussian function reduces to a product of one-dimensional ones, its separation ranks (and hence all TT-ranks) are equal to 1. The QTT-ranks of each 1D Gaussian can be estimated via the polynomial approximation of a regular function admitting analytical extension to the Bernstein ellipse $\mathcal{E}_\rho = \left\{w \in \mathbb{C} : \ |1+w| + |1-w| \leqslant \rho + \frac{1}{\rho}\right\}$:

$$\|f(w) - T_n(w)\|_\infty \leqslant C \log(N) \frac{M}{1-\rho} \rho^{-n}, \quad w \in \mathcal{E}_\rho, \ \rho > 1, \quad M = \max_{w \in \mathcal{E}_\rho} |f(w)|,$$

where T_n is a best polynomial interpolation of degree n on a grid with N points, and C does not depend on N, n, M, ρ [38, 39]. The polynomial of degree n on a uniform grid has the QTT ranks bounded by $n+1$, see [37]. It gives the exponential convergence of an approximation with its rank, but the constant M can be very large; for the Gaussian function we have $M = f(i\frac{a}{\rho}) = \exp(\frac{a^2}{2p^2\rho^2})$.

To deduce more elegant results we use another approach, presented in the following

Lemma 2. Suppose uniform grid points $-a = x_0 < x_1 < \cdots < x_N = a$, $x_i = -a + hi$, $N = 2^L$ are given on an interval $[-a, a]$, and the vector g is defined by its elements $g_i = e^{-\frac{x_i^2}{2p^2}}$, $i = 0, ..., N-1$. Suppose in addition that $\int_a^\infty e^{-\frac{x^2}{2p^2}} \leqslant \frac{\varepsilon}{2} < 1$. Then for all sufficiently small $\varepsilon > 0$ there exists the QTT approximation g_r with the ranks bounded as

$$r(g_r) \leqslant c \frac{a}{p} \sqrt{\log\left(\frac{1}{\varepsilon} \frac{p}{1+a}\right)},$$

and the accuracy

$$|g - g_r| \leqslant \left(\frac{r}{2a} + 1\right)\varepsilon = \left(c\frac{1}{p}\sqrt{\log\left(\frac{1}{\varepsilon}\frac{p}{1+a}\right)} + 1\right)\varepsilon,$$

where c does not depend on a, p, ε or N.

Proof. This lemma is based on the Fourier transform of the Gaussian function. Indeed, consider the approximation via the partial Fourier sum

$$e^{-\frac{x^2}{2p^2}} = \sum_{m=0}^{M} \alpha_m \cos\left(\frac{\pi m x}{a}\right) + \eta \quad \text{on } [-a, a],$$

where $|\eta| = \left|\sum\limits_{m=M+1}^{\infty} \alpha_m \cos\left(\frac{\pi m x}{a}\right)\right| < \varepsilon$. There are no sin functions in this sum, as the Gaussian function is even with respect to 0. If we discretize now this sum on a uniform grid, all vectors generated by cos functions will have exact QTT-representations with all ranks equal to 2, see [31]. So it is enough to provide an estimate on M.

The Fourier coefficients are computed as

$$\alpha_m = \frac{\int\limits_{-a}^{a} e^{-\frac{x^2}{2p^2}} \cos\left(\frac{\pi m x}{a}\right) dx}{\int\limits_{-a}^{a} \cos^2\left(\frac{\pi m x}{a}\right) dx},$$

where all denominators are equal to a if $m > 0$, and $2a$ if $m = 0$. Let us denote them as

$$|C_m|^2 = \begin{cases} 2a, & \text{if } m = 0, \\ a, & \text{otherwise.} \end{cases}$$

In the nominator, we note that the cos function is bounded by 1, and

$$\int\limits_{-\infty}^{\infty} e^{-\frac{x^2}{2p^2}} dx = \int\limits_{-a}^{a} e^{-\frac{x^2}{2p^2}} dx + 2\int\limits_{a}^{\infty} e^{-\frac{x^2}{2p^2}} dx \leqslant \int\limits_{-a}^{a} e^{-\frac{x^2}{2p^2}} dx + \varepsilon,$$

so we approximate

$$\alpha_m = \left(\int\limits_{-\infty}^{\infty} e^{-\frac{x^2}{2p^2}} \cos\left(\frac{\pi m x}{a}\right) dx - \xi_m\right) / |C_m|^2, \quad 0 < \xi_m < \varepsilon.$$

We deduce the integral over the whole axis from the continuous Fourier transform: indeed, it is known, that the Fourier image of the Gaussian function is another Gaussian function:

$$\int\limits_{-\infty}^{\infty} e^{-\frac{x^2}{2p^2}} e^{i\omega x} dx = \int\limits_{-\infty}^{\infty} e^{-\frac{x^2}{2p^2}} \cos(\omega x) dx = p e^{-\frac{\omega^2 p^2}{2}},$$

where i is the imaginary unity. So, plugging here $\omega = \frac{\pi m}{a}$ in, we get the statement for α_m:

$$\alpha_m = \left(p e^{-\frac{\pi^2 m^2 p^2}{2a^2}} - \xi_m \right) / |C_m|^2$$

Now we truncate α_m on a value $m = M$ so that $\alpha_M \leqslant \varepsilon$, hence for M:

$$M \leqslant \frac{\sqrt{2}}{\pi} \frac{a}{p} \log^{0.5} \left(\frac{p}{(1 + |C_M|^2)\varepsilon} \right) = \frac{\sqrt{2}}{\pi} \frac{a}{p} \log^{0.5} \left(\frac{p}{1 + a} \frac{1}{\varepsilon} \right),$$

which gives the first result of the lemma (up to rank 2 of each cosine function). Note that due to such very fast decay of the Fourier coefficients, the threshold $\alpha_M \leqslant \varepsilon$ implies $\sum_{m=M+1}^{\infty} \alpha_m \leqslant \varepsilon$ as well.

To obtain the expression for the error, recall that

$$g(x) = e^{-\frac{x^2}{2p^2}} = \sum_{m=0}^{M} \frac{1}{|C_m|^2} \left(p e^{-\frac{\pi^2 m^2 p^2}{2a^2}} - \xi_m \right) \cos\left(\frac{\pi m x}{a} \right) + \eta,$$

$$g_r(x) = \sum_{m=0}^{M} \frac{1}{|C_m|^2} p e^{-\frac{\pi^2 m^2 p^2}{2a^2}} \cos\left(\frac{\pi m x}{a} \right).$$

Now taking into account bounds $|\xi_m| < \varepsilon$, $|\cos\left(\frac{\pi m x}{a}\right)| < 1$, $|\eta| < \varepsilon$, $r = 2M$ we get the estimate for $|g - g_r|$. □

Remark 3. Requiring that $e^{-\frac{a^2}{2p^2}} \leqslant \varepsilon$ (it is to be imposed naturally in order to compute a physically relevant solution without significant boundary effects) we obtain $a = \sqrt{2} p \log^{0.5}(1/\varepsilon)$, so that $r(g_r) \leqslant c \log(1/\varepsilon)$, i.e. the result that could be obtained via the polynomial approximation. But here we have the *uniform* estimate with respect to all parameters except the accuracy.

In addition, it is worth no note, that the usage of grid information was connected only with the index splitting in the QTT (in the representation of cosine functions), and the number of summands were estimated on the continuous level. In practical computations, the grid effects in the case $p \to h$ (i.e. when there are few grid points in the Gaussian peak) make the ranks significantly smaller than those provided by Lemma 2.

3. Fokker-Planck equation

3.1. Non-stationary Fokker-Planck equation

The main application in this paper is the numerical solution of the non-stationary Fokker-Planck equation. In general form, this equation reads

$$\begin{aligned} &\tfrac{\partial \psi}{\partial t} = -A\psi, \quad \psi(0) = \psi_0, \quad \text{where} \\ &A\psi = -\sum_{i,j=1}^{d} \frac{\partial}{\partial q_i} \cdot \frac{\partial}{\partial q_j} D_{ij} \psi + \sum_{i=1}^{d} \frac{\partial}{\partial q_i} \cdot \nu_i \psi, \quad \text{and} \\ &q_i \in \mathbb{R}^n, \quad n = 1, 2, 3. \end{aligned} \tag{7}$$

Such equations arise in many stochastic models like polymeric liquids with Brownian motion [2, 3, 5, 40], chemical master equations [41] and so on. The solution is assumed

to vanish at infinity, $\psi(q,t) \to 0$, $|q| \to \infty$, thus equation (7) is posed in a cylinder $[-a,a]^d \times [0,\infty)$ for sufficiently large a, and the Dirichlet boundary conditions are employed. Equation (7) has a zero forcing term, thus the time evolution eventually converges to the null-space of A, i.e. to the solution of the stationary equation

$$A\psi_* = 0. \tag{8}$$

In the most of the paper, we focus on the simplified case of a scalar diffusion tensor, $D_{ij} = \varepsilon\delta_{ij}$, so that

$$A\psi = -\varepsilon\Delta\psi + \mathrm{div}(\psi v), \quad v = (v_1, ..., v_d)^T. \tag{9}$$

Nevertheless, in polymeric liquid models a general equation is written in the form [2]

$$\frac{\partial\psi}{\partial t} + \sum_{i=1}^{d} \frac{\partial}{\partial q_i} \cdot \left(K q_i - \frac{1}{4} \sum_{j=1}^{d} D_{ij} \frac{\partial \phi}{\partial q_j} \right) \psi - \frac{1}{4} \sum_{i,j=1}^{d} D_{ij} \frac{\partial}{\partial q_i} \cdot \frac{\partial \psi}{\partial q_j} = 0, \tag{10}$$

where the diffusion tensor is not diagonal, and since each q_i is itself a low-dimensional vector, representing the position of the i-th particle in space, K is a square matrix of size 1,2, or 3, respectively. In Section 3.3 and in the last numerical example we discuss, why the straightforward treatment of such problem in a format with separated variables is more difficult, than in the case (9).

In applications, the stationary solution, normalized as

$$\int \psi_* dq_1 \dots dq_d = 1, \tag{11}$$

has a meaning of the *probability density*, i.e. has to be non-negative. The direct solution of equation (8) by some approximate solver (i.e., by the TT-Solve algorithm) may not preserve non-negativity. A time-stepping scheme, however, after appropriate discretization of equation (9) will preserve it. Another important feature of the non-stationary problem (7) is that the solution may have non-trivial dependence on t, for example the norm of the solution may grow in some time interval $[0; T]$, and only after that it starts to stabilize (so-called "shocks").

The analytical solution of equations (7), (9), (8) is usually not known. However, if v is a potential field, i.e.

$$v = \mathrm{grad}(\phi), \quad \phi : \mathbb{R}^{dn} \to \mathbb{R},$$

then the analytical solution of the stationary problem is given as [1],

$$\psi = C e^{-\frac{\phi}{\varepsilon}},$$

where the constant C is defined to satisfy the normalization condition (11). From practical point of view the numerical solution of the Fokker-Planck equation with the potential field is of course not interesting. However, it is very convenient to test the numerical algorithms for the solution of the Fokker-Planck equation on such kind of input data (for example, the grid approximation properties).

A particular model, also referred to as the Hookean spring model [5], is defined by

$$\phi = \frac{|q|^2}{2p^2} = \sum_{k=1}^{d} \frac{|q_k|^2}{2p^2}, \quad v_k = \frac{q_k}{p^2} \in \mathbb{R}^n. \tag{12}$$

In this case, the stationary solution is the Gaussian function,

$$\psi = Ce^{-\frac{|q|^2}{2p^2\varepsilon}} = C\prod_{k=1}^{d} e^{-\frac{|q_k|^2}{2p^2\varepsilon}}.$$

Its TT and QTT rank bounds were obtained in Section 2.3.

3.2. Discretization in space

Let q_i be the vector of coordinates of a point in space, describing the position a spring. It can be either one, two or three-dimensional, $q_i = x_i$, $q_i = (x_i, y_i)$, and $q_i = (x_i, y_i, z_i)$, respectively. In this section, each $q_i = x_i$ is a one-dimensional coordinate vector, though the scheme presented below can be used for higher-dimensional models as well.

To discretize (9) in spatial variables, we will use a simple finite difference scheme. Take sufficiently large $a > 0$ and introduce a uniform tensor grid in $[-a, a]^d$ with $n = 2^L$ points in each direction, with a step size $h = \frac{2a}{N}$:

$$x(i) = -a + ih, \quad i = 0, \ldots, N-1.$$

The Laplace operator is discretized via a standard second-order finite difference scheme. The corresponding matrix has the form

$$\Delta_d = \Delta_1 \otimes I \otimes \ldots \otimes I + \ldots + I \otimes \ldots \otimes \Delta_1,$$

where

$$\Delta_1 = \frac{1}{h^2}\mathtt{tridiag}[1, -2, 1],$$

and I is a $N \times N$ identity matrix. The QTT-representation of such matrix with ranks bounded by 4 was obtained in [32]. Thus, the storage of the discrete Laplace operator is no more than $16D$ elements and is negligible.

The convection term is more interesting. Since it is a divergence of a vector-valued function it is convenient to use the central difference for the derivatives in each direction. The corresponding matrix T is then represented as

$$T = C_1\Lambda_1 + C_2\Lambda_2 + \ldots + C_d\Lambda_d, \tag{13}$$

where $C_k\Lambda_k$ is the discretization of the term

$$\frac{\partial}{\partial x_k}v_k,$$

and v_k is the k-the component of the vector field v. The multiplication by v_k reduces to the multiplication by the diagonal matrix Λ_k obtained from v_k by taking values on a tensor grid. The diagonal elements of Λ_k are naturally indexed by a multiindex $(i_1, \ldots, i_d)$,

$$\Lambda_k(i_1, \ldots, i_d, i_1, \ldots, i_d) = v_k(x(i_1), x(i_2), \ldots, x_d(i_d)).$$

The matrix C_k is a central-difference operator in the k-th mode, i.e.

$$C_k = I \otimes \ldots \otimes \underbrace{C}_{k} \otimes \ldots \otimes I, \tag{14}$$

and C is a one-dimensional central difference operator, i.e.,

$$C = \frac{1}{h}\mathtt{tridiag}\left[-\frac{1}{2}, 0, \frac{1}{2}\right].$$

The QTT-ranks of the matrix (13) depend on the QTT-ranks of the components of the vector field v_k taken on a considered grid. Then the following simple Lemma holds.

Lemma 3. Suppose that the QTT-ranks of the functions v_k on a tensor grid are bounded by r. Then, the QTT-ranks of the matrix T (13) are bounded by $5rd$.

Proof. Since the QTT ranks of v_k are bounded by r, the QTT-ranks of the diagonal matrices Λ_k are also bounded by r. The QTT-ranks of the matrices C_k are bounded by 5, which can be shown analogously to the Laplace operator. The statement of Lemma follows from (13) using the multiplicativity and addititivity properties of the QTT-ranks. □

The proof of Lemma 3 is constructive and provides a way to compute the QTT-representation of the matrix T using the QTT-representation of the vector field v. The QTT-representation of the vector v can be obtained either by using known analytical representations [37, 31] (e.g. in the case where v_k is a sum of exponential, trigonometric and/or polynomial functions), or by using adaptive cross approximations in the TT-format [42, 43], when these functions are given only pointwise.

Remark 4. In case of the Hookean potential (12) the matrix T has all TT ranks equal to 2, the same as the Laplace matrix. Indeed, each Λ_k has the form $I \otimes \cdots \otimes I \otimes \mathrm{diag}(v_k) \otimes I \otimes \cdots \otimes I$ (which follows from the separability of the potential), and is similar to (14). So the matrix T reads

$$T = C\mathrm{diag}(v_1) \otimes I \otimes \ldots \otimes I + \ldots + I \otimes \ldots \otimes C\mathrm{diag}(v_d),$$

which is the same form as the Laplacian and as it was proven in [37, 32] to have TT ranks equal to 2.

3.3. Possible generalizations and related difficulties

In the numerical examples below we will focus on the advantages of the QTT structuring in the 3D problem, arising from the so-called *dumbbell* model, i.e., 2 beads connected by one spring, and the extension of the spring in a 3-dimensional space with the coordinate vector $q = (x, y, z)$. The solution of such model can be approximated in the QTT format with moderate ranks (30-40), which depend only slightly on the spatial and time grid sizes. So, we will demonstrate the log-volume complexity.

There are more complicated models, with two main directions of generalization.

3.3.1. Multi-bead model. This is the multi-spring model with d springs, resulting in the $3d$ Fokker-Planck equation, with d varying from $3-4$ to several tens (sometimes, simplified $2d$ or $1d$ models are considered). Moreover, the operator of second-order derivatives is usually not diagonal like in (9). The most frequently used diffusion tensor in bead-spring

models (appears if the coordinates q are chosen so that $q_i = r_{i+1} - r_i$, where r_i is the position of i-th bead) reads

$$D_{ij} = \begin{cases} 2, & i = j, \\ -1, & |i-j| = 1, \\ 0, & \text{otherwise}, \end{cases} \tag{15}$$

and the velocity is given as

$$v_i = Kq_i - \alpha \sum_{j=1}^{d} D_{ij} \frac{\partial \phi}{\partial q_j}, \quad i = 1, ..., d.$$

This results in a certain form of the solution, being the Gaussian function, but aligned along the diagonals of a $3d$-dimensional hypercube (e.g. rotated on the angle $\approx \pi/4$ in the plane x_i, y_i, see Fig. 1).

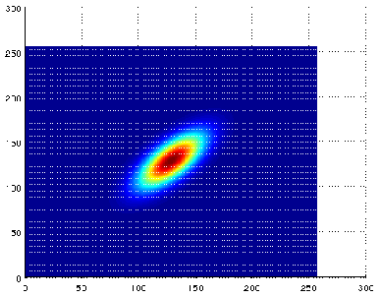

Figure 1. A x_1, y_1 projection of (approximately) stationary solution in a 4-spring model.

Such structure of the solution results in a large separation rank of the corresponding variables (which is natural, since this pattern is close to a diagonal matrix with nonzero entries, being known to have the maximal possible ranks equal to its size). An even worse situation occurs if such dimensions stay far from each other in a tensor train; then all intermediate blocks have to be contaminated by large-rank factors, connecting just two badly separable dimensions. In this cases the QTT-ranks in the middle of the train achieve values of several hundreds, which makes this method uncompetitive with, for example, Monte-Carlo techniques. A special permutation of dimensions, or usage of so-called normal coordinates (to align the solution along the coordinate axes) might be advised, but it requires additional study. In the last subsection we present an example showing that the TT ranks grow linearly with the magnitude of K, making "naive" tensor solution difficult even for small external velocities.

One should note, that a certain type of potentials, e.g. the exponential repulsive potential from [2], contains terms of the form $\exp(-(x_1 + x_2)^2/2p^2)$ in the multi-spring models, which is also diagonally-aligned, and the smaller is p, the closer is its pattern to the identity matrix. Thus such potentials are completely intractable via the methods based on the separation of variables without rather sophisticated modification of variables (including fictitious variables).

3.3.2. More accurate modeling of each spring. In this paper and in [2] the main part of the potential at infinity is Hookean, which admits the springs to be infinitely

extensible, which is not relevant in certain flow regimes. To fix the maximal length of a spring, one uses mostly the *Inverse Langevin* spring force

$$\frac{\partial \phi}{\partial q_i}(q_i) = \frac{\sqrt{b_i}}{3} L^{-1}\left(\frac{|q_i|}{\sqrt{b_i}}\right)\frac{q_i}{|q_i|},$$

where the *Langevin* function reads $L(t) = \coth(t) - 1/t,\ t \in [0, \infty)$, so that the force has a vertical asymptote at $|q_i| = \sqrt{b_i}$, and thus, the maximal length of i-th spring is limited by $\sqrt{b_i}$. In practice one uses not the inverse Langevin function itself but its certain approximation:

- FENE $\phi = \sum_{i=1}^{d} -\frac{b_i}{2}\ln\left(1 - \frac{|q_i|^2}{b_i}\right)$, or
- CPAIL $\phi = \sum_{i=1}^{d} \frac{|q_i|^2}{6} - \frac{b_i}{3}\ln\left(1 - \frac{|q_i|^2}{b_i}\right)$,

see [3, 5, 4]. Such potentials separate well in variables q_i (in fact, these potentials are sums of univariate functions, which have the TT ranks equal to 2, see the consideration in Remark 4), but the velocities are infinite on the boundaries, making numerical treatment of the problem a complicated subject.

For a one-spring 2D model use of the Gaussian grids allows to work with small-sized stiffness matrices, admitting the direct solution in the implicit time stepping scheme [4]. But in a multi-spring model it can not be done even for a 10-point grid in each coordinate, thus one has to deal both with the rotating solutions and ill-conditioned matrices.

First attempt to implement a tensor-structured solution method for a multi-spring problem traces back to [6], where a greedy algorithm, giving the approximate solution as a sum of rank-1 tensors (the so-called *canonical* tensor format) was presented. However, it was tested only for qualitative estimates with the accuracy 0.1. Even for the "small" case of two one-dimensional springs it generates 7 canonical terms and achieves the L_2 accuracy 0.01. From the considerations in the beginning of this section it becomes clear that even for the orthogonal SVD-based decompositions the separation rank of the solution in this case can increase rapidly with the accuracy. So, due to the slow convergence in the ranks of this method (it can happen with greedy approximation approach even if a low-rank orthogonal approximation exists, see, e.g., [44, 45, 7, 8]), and well-known drawbacks of the canonical decomposition, the efficient use of tensor solvers in this approach will be limited by the accuracy demands and restrictions on the rank parameters.

4. Numerical experiments

All numerical experiments are conducted in Matlab R2009b on a Linux x86_64 computer with 2.0 GHz Intel Xeon E5504 CPU.

4.1. Heat equation

As a "sanity" test we consider the heat equation

$$\begin{aligned} &\tfrac{\partial u}{\partial t} - \Delta u = f \quad \text{in } \Omega = [0,1]^2, \\ &u|_{\partial\Omega} = 0, \\ &u(0) = g. \end{aligned}$$

The solution structure from the tensor point of view can differ significantly with the input data. We illustrate it on the following two choices of g and f.

4.1.1. Smooth data. First, we consider the case of a smooth analytic solution and investigate the convergence of the solution scheme with the refinement of the grids. We choose $f = 0$, $g(x_1, x_2) = \sin(\pi x_1)\sin(\pi x_2)$. The analytical solution at the time t is $u^*(x_1, x_2) = g(x_1, x_2)\exp(-2\pi^2 t)$. The time interval is fixed to $[0, 1/2]$, and the residual tolerance for the TT-solve algorithm is 10^{-6}. The relative Frobenius-norm error at $t = 1/2$ versus grid sizes in space and time is given in Table 1. The convergence is a bit faster than

Table 1. $\frac{\|u-u^*\|_F}{\|u^*\|_F}$ versus the spatial N_x and time N_t grid sizes

N_t \ N_x	2^8	2^9	2^{10}	2^{11}
2^7	4.7598e-03	4.8515e-03	4.8746e-03	4.8803e-03
2^9	1.8271e-04	2.7475e-04	2.9786e-04	3.0363e-04
2^{11}	1.0380e-04	1.1745e-05	1.1363e-05	1.7152e-05
2^{13}	1.2171e-04	2.9652e-05	6.5401e-06	7.8656e-07

the theoretical bound $\mathcal{O}(N_t^{-2}+N_x^{-2})$ of the Crank-Nicolson - Finite Difference scheme. The solution time of the block scheme (5) is almost independent on N_t, N_x (about $100-200$ milliseconds for any test in Table 1), since the QTT ranks are uniformly bounded by a small constant.

4.1.2. Irregular data. Now, as the input, we take $f(x_1, x_2) = g = 1$, $x_1, x_2 \in (0, 1)$, i.e. the functions, which have the discontinuity at the boundary. The time step should be small enough to resolve transitional processes near the boundary, otherwise, as the second-order scheme is not monotonous, the oscillations occur, which usually have large tensor ranks.

The time interval is fixed to $[0, 1]$, the spatial grid size in each direction is $N_x = 256$, and the QTT approximation tolerance is 10^{-6}. We compare the time stepping solution scheme (4) with the block approach (5), see Table 2. For the number of time steps varying from 2^8 to 2^{16}, we present the computational times of both approaches, measure of closeness of the final solution to the stationary one (the norm of the time derivative), and the QTT-rank of the block solution. We notice, that the solution time even decreases

Table 2. CPU times (sec.), relative residuals provided by the final solution, and the average QTT rank in the block and time stepping schemes.

	Block solution			Time stepping	
N_t	CPU time	$\frac{\|-\Delta u(1)-f\|}{\|f\|}$	rank	CPU time	$\frac{\|-\Delta u(1)-f\|}{\|f\|}$
2^8	76.96	1.53e+03	37.17	1310.4	1.53e+03
2^{10}	83.23	7.34e-03	41.09	4547.1	7.38e-03
2^{12}	69.32	1.44e-04	39.40	3845.9	1.69e-03
2^{14}	60.40	3.67e-05	35.91	6232.4	1.58e-04
2^{16}	61.37	5.49e-05	32.98	12707	7.24e-05

with the number of time steps in the block algorithm (and grows slower than linear in the time stepping case), since the smaller the time step, the smoother the solution is. Moreover, it leads also to a smaller condition number of the matrices in the linear systems involved. Notice, that the decay of the *discrete* spatial residual shown here is not due to the $\mathcal{O}(N_t^{-2})$-approximation of the time scheme, but to the better resolution of fast harmonics.

As for the rank, it is stable with respect to the number of time steps, which confirms Lemma 1, and manifests also a slight decrease due to the improving smoothness of the solution. Despite that this example is a bit unrealistic from the practical point of view, it demonstrates the advantages of the logarithmic scaling of our block scheme, especially if we have to choose extremely small time steps.

4.2. Dumbbell example

4.2.1. Problem formulation. As a model example, consider the *dumbbell model* discretized on large time-space grids. It is a three-dimensional Fokker-Planck equation of form (7), (9) with

$$v = Kq - \frac{1}{2}\mathrm{grad}(\phi), \quad \varepsilon = \frac{1}{2},$$

where

$$K = \beta \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad q = \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in [-a, a]^3, \tag{16}$$

specifies the *external* velocity, and the potential energy ϕ is given as [2]

$$\phi = \frac{1}{2}(x^2 + y^2 + z^2) + \frac{\alpha}{p^3} e^{-(x^2+y^2+z^2)/(2p^2)}, \tag{17}$$

including the Hookean potential of one spring, and the repulsion potential of the beads. In this case the velocity reads

$$v = Kq - \frac{1}{2}q + \frac{\alpha}{2p^5} e^{-(x^2+y^2+z^2)/(2p^2)} q.$$

It follows immediately, that its TT ranks are bounded by 3: indeed,

$$e^{-(x^2+y^2+z^2)/(2p^2)} = e^{-x^2/(2p^2)} e^{-y^2/(2p^2)} e^{-z^2/(2p^2)}$$

is a rank-1 function, as well as separate x, y, z, so the rank of the total sum is not greater than 3. Notice that the QTT ranks are greater: the QTT ranks of each coordinate vector are equal to 2, the rank of the Gaussian function depends on the accuracy as $\mathcal{O}(|\log(\varepsilon)|^{1/2})$, which is given by Lemma 2.

The following functional (the so-called Kramer expression) of the solution is interesting:

$$\boldsymbol{\tau}(t) = \int \psi(t) \left(q \otimes \mathrm{grad}(\phi)\right) dq.$$

In particular, the following functions

$$\eta(t) = -\frac{\boldsymbol{\tau}_{12}}{\beta}, \quad \Psi(t) = -\frac{\boldsymbol{\tau}_{11} - \boldsymbol{\tau}_{22}}{\beta^2}. \tag{18}$$

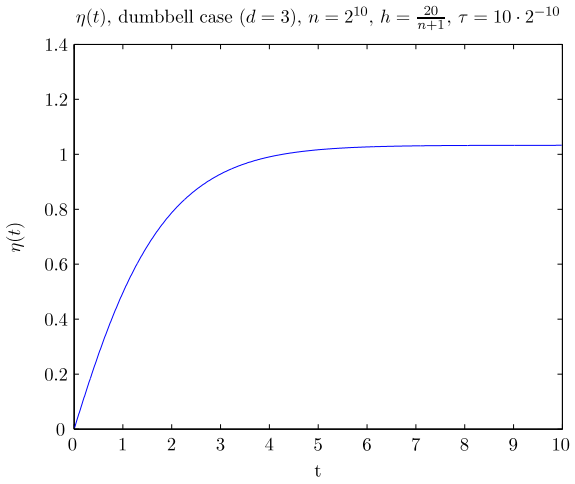

Figure 2. Dependence of η on t.

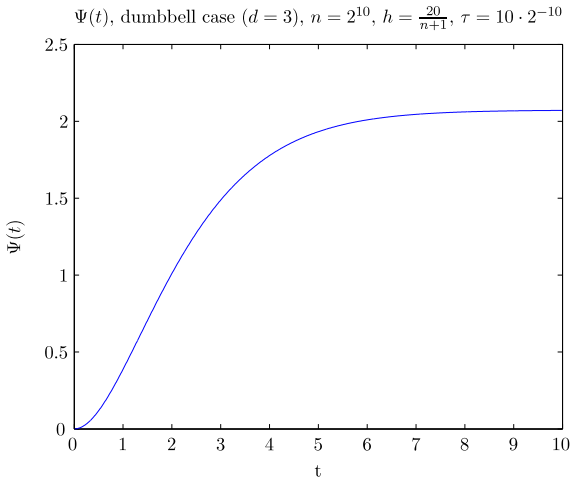

Figure 3. Dependence of Ψ on t.

will be computed, see Fig. 2, 3.

In the following, we present the results on the numerical complexity and accuracy. All the computations were performed using the TT-solver from the TT Toolbox 2.1 with the QTT format for data. The following parameters were fixed:

- $\beta = 1$, $\alpha = 0.1$, $p = 0.5$,
- computational domain $\Omega = [-10, 10]^3$,
- the problem was solved on the time interval $[0, 10]$, and the final solution was taken at the time $T = 10$,
- relative tensor rounding (for approximations) and residual (for TT-solve) accuracy $\varepsilon = 10^{-6}$.

In the results below, the Q-dimension is shown both for spatial d_x and time d_t discretizations. The grid size h and time step τ are computed as follows,

$$h = \frac{20}{2^{d_x} + 1}, \quad \tau = \frac{10}{2^{d_t}}.$$

4.2.2. Numerical discretization properties. First, consider the grid approximation with respect to h. In this test, the time dimension is fixed to $d_t = 8$ (so $\tau \approx 0.039$), and the spatial dimension varies from 4 to 9. The quantities of interest, stationary η and Ψ are shown in Table 3. The relevant digits are emphasized with the boldface.

The behavior of errors $\eta_{d_x}(T) - \eta_9(T)$ and $\Psi_{d_x}(T) - \Psi_9(T)$ with respect to d_x is shown in Figures 4, 5. Coefficients of the linear fit equations show that the approximation is even better than h^2, being in the order of $h^{2.5}$ for η, and $h^{2.8}$ for Ψ. The difference between $\eta_8(T)$ and $\eta_9(T)$ is about $1.38 \cdot 10^{-5}$ and is already governed by tensor roundings rather than the grid approximation.

4.2.3. Complexity tests on the timestepping solution scheme. Now we present the CPU times of the step-by-step time integration via the Crank-Nicolson scheme (4). The

Table 3. The functions $\eta(T)$ and $\Psi(T)$ versus the spatial dimension d_x.

d_x	$\eta(T)$	$\Psi(T)$
4	**1.03**88419	**2.0**84148
5	**1.032**0981	**2.0**69552
6	**1.032**6395	**2.07**0764
7	**1.032**7829	**2.071**111
8	**1.0328**263	**2.071**209
9	**1.0328**125	**2.071**143

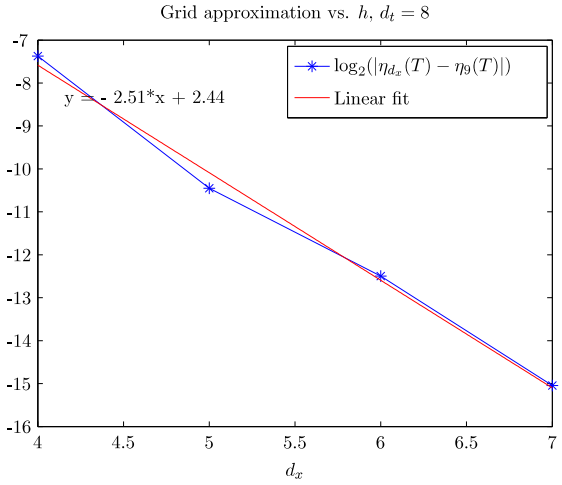

Figure 4. Grid approximation of η versus h.

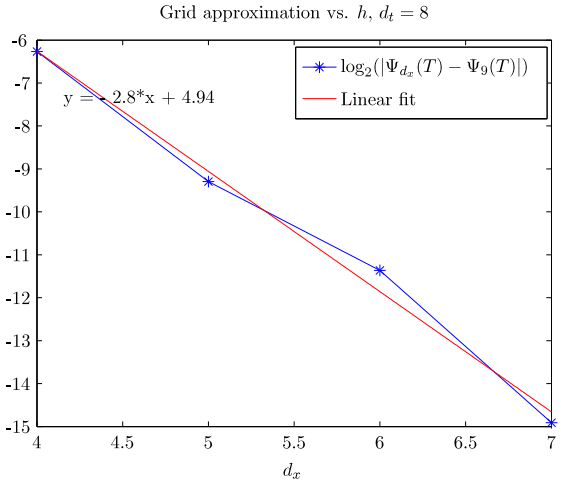

Figure 5. Grid approximation of Ψ versus h.

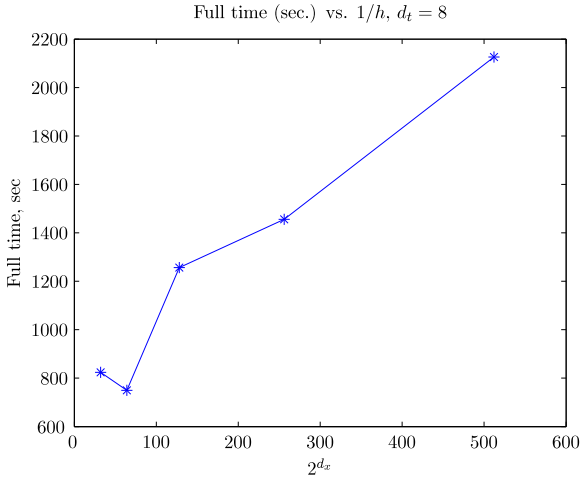

Figure 6. Full solution time (sec.) versus h, timestepping scheme.

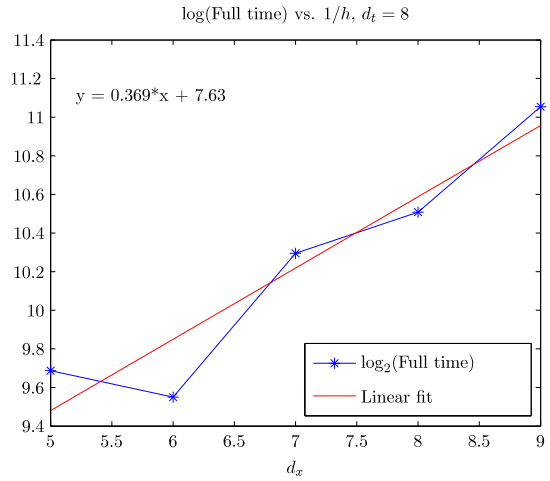

Figure 7. Full solution time in log-scale versus h, timestepping scheme.

timings of the solution process are shown in Figures 6, 7. The second one is provided to check the asymptotic complexity: the first coefficient in the linear fit equation ($0.369 < 1$) indicates that even a sublinear complexity with respect to the number of grid points was achieved. A time decrease from $d_x = 5$ to $d_x = 6$ is due to lower QTT ranks, as the finer grid resolves the solution structure better and provides a smaller grid Reynolds number,

thus getting rid of some rank-increasing oscillations.

Now, fix the spacial quantized dimension to be $d_x = 9$ and check the approximation and solution time with respect to the time Q-dimension (Figures 8, 9). From Figure 8

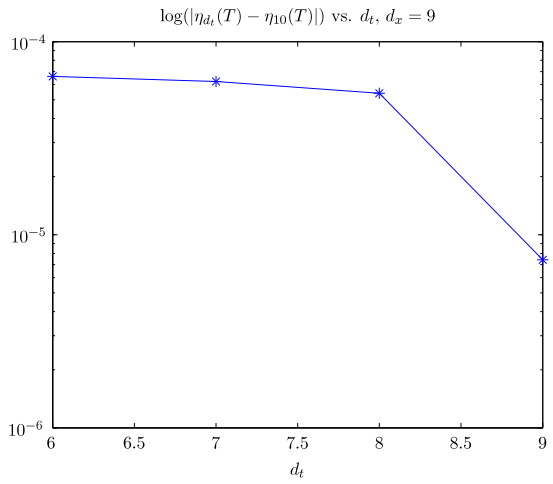

Figure 8. Approximation of η versus τ.

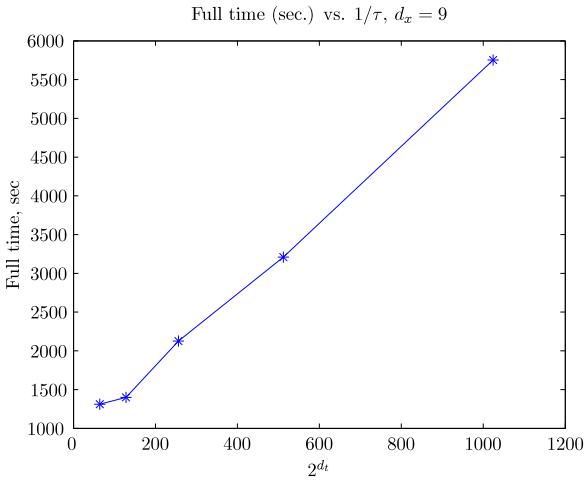

Figure 9. Full solution time versus τ.

we see, that the difference between η, computed on 2^{d_t} time steps and $2^{10} = 1024$ time steps is about 10^{-4} if $d_t < 9 = d_x$ and does not decrease significantly. That is due to an oscillating behavior of the solution computed with $h < \tau$. But if $\tau \leqslant h$, the error drops down to the value 10^{-5}, i.e. $\mathcal{O}(\tau^2)$, as it follows from the properties of the Crank-Nicolson scheme. The total CPU time grows linearly with the number of time steps, as expected.

4.2.4. Block solution scheme: implementation details and results. From our results, it becomes clear that the smaller time step we take, the more accurate solution we get. One might think of the extremely high number of time steps (e.g. $2^{12}-2^{16}$), especially for high absolute times T (the accuracy of the Crank-Nicolson scheme is estimated as $\mathcal{O}(\tau^2 T)$, thus if $T = 100$, 10^4 time steps might be reasonable). But to implement it in a time-stepping scheme requires too much time.

As an alternative, the *block* linear system solution (5) is proposed. The only problem is that the longer time interval one consider, the more significantly different snapshots have to be stored in one QTT tensor, resulting in the larger QTT ranks to keep for the same accuracy. If we need only the final (stationary) solution, it is better to split the global time interval on several subintervals, and implement the block solution method with restarts. We summarize it in Algorithm 1.

In our dumbbell example, we split the global interval $[0; 10]$ on 8 equal subintervals (i.e., $[0; 1.25]$, $[1.25; 2.5]$ and so on), and solve the global system (5) on each of them. The time dimensions d_t presented below correspond to the discretization of *each* block system on each time subinterval ($d_{t,m}$ in Algorithm 1), thus, the equivalent number of time steps in the timestepping procedure is 8 times larger.

In Figure 10, we present the computational times on each interval with respect to the Q-time dimension d_t. The spacial dimension d_x is fixed to $d_x = 8$. In Figure 11 the total CPU times of the solutions (i.e. to compute the solution at the time $T = 10$) are given with respect to d_t and d_x. Notice, that the difference between d_t shown here and in the previous experiment with the step-by-step procedure (the corresponding equivalent value

Algorithm 1: Restarted block solution algorithm.

Input: The time interval splitting $[0, T] = \bigcup_{m=1}^{M} [T_{m-1}, T_m]$, initial solution $\psi(0)$ at $T_0 = 0$, matrix of spatial discretization A, time dimensions for each subinterval $\{d_{t,m}\}_{m=1}^{M}$.

Output: The solution history $\psi(t)$, and/or the final solution $\psi(T)$ in the QTT format.

1: **for** $m = 1, ..., M$ **do**
2: Assemble the block linear system (5), using $\psi_0 = \psi(t_{m-1})$, $N_t = 2^{d_{t,m}}$, $\tau = \frac{T_m - T_{m-1}}{N_t}$, and the matrix A.
3: Solve the linear system, obtain the part of the solution history $\psi(t)$ on the subinterval $[T_{m-1}, T_m]$.
4: Extract the final solution $\psi(t_m)$ from the block QTT-tensor $\psi(t)$.
5: **end for**

in the time stepping scheme is presented in brackets as d_t^{ts}). We see, that the average

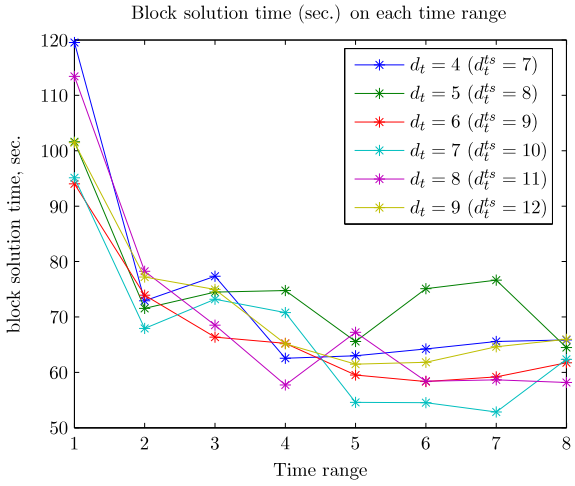

Figure 10. Block solution time (sec.) versus the time range and d_t, $d_x = 8$.

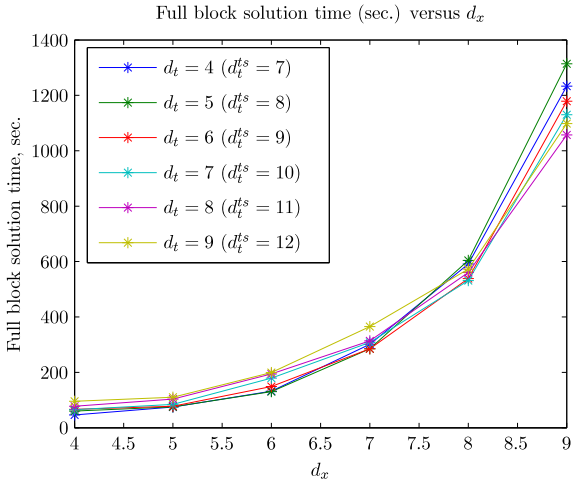

Figure 11. Full block solution time (sec.) versus d_t and d_x.

time is almost independent from the number of time steps (oscillations are due to the randomizations in the TT-solve procedure and slightly different TT ranks, see Figure 10). The main trend is a decrease of the time spent on a particular time subinterval with its absolute position, which follows from the convergence of the solution to the stationary one: on the first subinterval we have significantly varying snapshots in the block TT-storage, whereas on the last ones there are almost equal to the stationary solution.

Approximate linear increase of the QTT-ranks with the spacial dimension d_x causes the *polylog* complexity with respect to the spacial grid size (Figure 11). In this case, we observe a total *cubic-log*, $\mathcal{O}(d_x^3) = \mathcal{O}(\log^3(N_x))$, experimental complexity: one logarithm comes from the dimension of a tensor, and two more come from the approximately logarithmic grow of the QTT-ranks with the spatial grid size. But, nevertheless, these times are significantly smaller than ones from the time stepping procedure. From Figure 11 we see, that the block solution algorithm with the parameters $d_x = 9$, $d_t^{ts} = 12$ has taken about 1100 seconds. Extrapolating the results presented on Figure 9, we can conclude, that the step-by-step procedure would require about $22000 \gg 1100$ seconds for the same grid

sizes.

The QTT-ranks are observed to be almost independent on the number of time steps, thus the assumption of uniformly bounded ranks is fulfilled. Figure 11 shows the linear dependence of the time on the time dimension d_t, that is, the logarithmic dependence on the number of time steps. We conclude, that the total numerically observed complexity is

$$\mathcal{O}(r^2 \log(N_t) \log(N_x)) = \mathcal{O}(\log(N_t) \log^3(N_x)).$$

4.3. Multi-spring example

To illustrate the difficulty of the computation, described in Section 3.3, we solve the 4-spring model, with each spring moving in a 3-dimensional space, thus giving the 12-dimensional equation (10), where $d = 4$ is the number of springs, D_{ij} is given by (15), and K by (16). For the spring potential, we choose here the Hookean model $\phi = \sum_{i=1}^{d} \frac{1}{2}|x_i|^2$.

The equation (10) is discretized using the finite difference scheme (see Section 3.2) and solved in the QTT-format. As the initial guess, the product of Gaussian functions (6) with unit dispersion ($p_k = 1$) was set. The dependence of the average QTT and TT (i.e. only the ranks between "real" dimensions are considered) ranks of the $\varepsilon = 10^{-4}$-approximation versus the external velocity β is given in Figure 12.

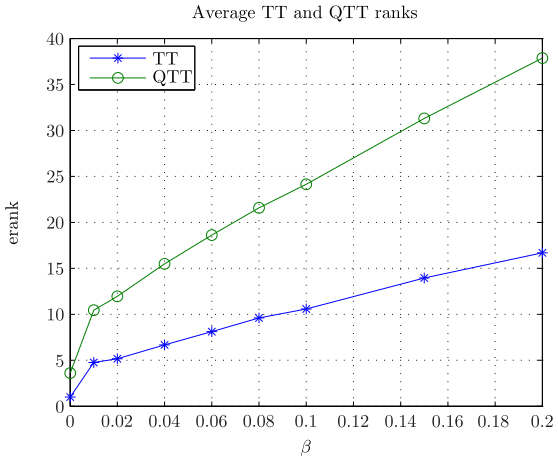

Figure 12. TT and QTT ranks of the stationary solution versus β in a 4-spring model.

We see, that the both ranks grow linearly (except the degenerate case $\beta = 0$) with the velocity, however, already for $\beta = 0.2$, the QTT rank is equal to 38 (but $\beta \sim 1$ is of usual choice, see the previous one-spring experiments). Notice, that we present the *average* rank. The maximal QTT rank in the middle of the tensor train representing the solution is equal to 84 in the case $\beta = 0.2$. We recall that in the course of the TT-solve procedure, we have to solve linear systems of size $4r^2$ at each step, which in this case equals to 28000 (which is less than, say, 256^{12}, but still is a computationally demanding problem to solve).

Moreover, the larger the dimension of a model is, the larger timescale T we need for a solution to be close enough to the stationary one. In the current example, $T = 100$ is a good choice (compare with $T = 10$ in the 3D example), and the total computational time is about 20000 seconds (for $\beta = 0.1$). Nevertheless, if we were able to compute accurately

the transformation to the normal coordinates, the TT ranks of the nearby stationary solution should be close to 1, which would make the problem tractable.

To conclude, it is worth to see the influence of the repulsion potential, presented in (17) and taken from [2], which is absent in the Hookean model (we recall the discussion in Section 3.3 concerning bad separability of the repulsion potential). To that end, we compute the viscosity functions according to (18) from the solution at $\mathsf{T} = 100$.

Potential	η	Ψ
Hookean + repulsion, $\beta = 1$	7.932184	60.348455
Hookean, $\beta = 0.02$	7.9998	60.4180
Hookean, $\beta = 0.08$	7.9927	60.8285
Hookean, $\beta = 0.2$	7.9730	60.4315

We see that 2 digits are fixed in all cases. So, with 99% accuracy, the viscosity is governed by the Hookean model and is stable with respect to the velocity in the considered range of β.

5. Conclusion

A tensor structured scheme for the numerical solution of the parabolic PDEs was presented. Provided that the input and output data of a model possess bounded QTT-ranks, the method manifests polylog complexity in both the spatial grid size and the number of time steps. An application to the Fokker-Planck equation in polymeric fluid modelling was considered. For the simplest cases the theoretical results on the tensor properties of solution and Hamiltonians were established, and numerical experiments were conducted to confirm the theoretical performance of our solvers.

Nevertheless, some difficulties may arise if the proposed method is applied to a class of more complex models straightforwardly. Our analysis indicates that the structure of the general Fokker-Planck Hamiltonian might imply the substantial increase of the tensor ranks, requiring certain modifications, to be studied in a future work.

References

[1] *Fokker A. D.* Die mittlere Energie rotierender elektrischer Dipole im Strahlungsfeld // *Annalen der Physik*. 1914. V. 348, № 5. P. 810–820. doi: 10.1002/andp.19143480507. http://dx.doi.org/10.1002/andp.19143480507.

[2] *Venkiteswaran G., Junk M.* A QMC approach for high dimensional Fokker-Planck equations modelling polymeric liquids // *Math. Comput. Simul.* 2005. V. 68. P. 43–56. doi: 10.1016/j.matcom.2004.09.002.

[3] *Lozinski A., Chauviére C.* A fast solver for Fokker-Planck equation applied to viscoelastic flows calculations: 2D FENE model // *Journal of Computational Physics*. 2003. V. 189, № 2. P. 607 - 625. http://www.sciencedirect.com/science/article/pii/S0021999103002481.

[4] *Chauviére C., Lozinski A.* Simulation of dilute polymer solutions using a Fokker-Planck equation // *Computers & Fluids*. 2004. V. 33. P. 687-696.

[5] *Figueroa, Leonardo E. and Süli, Endre*. Greedy approximation of high-dimensional Ornstein-Uhlenbeck operators with unbounded drift: Preprint. arxiv:1103.0726v1: Oxford, 2011. http://arxiv.org/abs/1103.0726v1.

[6] *Ammar A., Mokdad B., Chinesta F., Keunings R.* 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*. 2006. V. 139, № 3. P. 153 - 176. http://www.sciencedirect.com/science/article/pii/S0377025706001662.

[7] *Le Bris C., Leliévre T., Maday Y.* Results and Questions on a Nonlinear Approximation Approach for Solving High-dimensional Partial Differential Equations // *Constructive Approximation*. 2009. V. 30. P. 621-651. http://dx.doi.org/10.1007/s00365-009-9071-1.

[8] *Cancés E., Ehrlacher V., Leliévre T.* Convergence of a greedy algorithm for high-dimensional convex nonlinear problems // *Mathematical Models and Methods in Applied Sciences*. 2011. http://arxiv.org/abs/1004.0095.

[9] *Hitchcock F. L.* The expression of a tensor or a polyadic as a sum of products // *J. Math. Phys*. 1927. V. 6, № 1. P. 164–189.

[10] *Harshman R. A.* Foundations of the Parafac procedure: models and conditions for an explanatory multimodal factor analysis // *UCLA Working Papers in Phonetics*. 1970. V. 16. P. 1-84.

[11] *Caroll J. D., Chang J. J.* Analysis of individual differences in multidimensional scaling via n-way generalization of Eckart-Young decomposition // *Psychometrika*. 1970. V. 35. P. 283-319. doi: 10.1007/BF02310791.

[12] *Gavrilyuk I., Khoromskij B.* Quantized-TT-Cayley transform for computing the dynamics and the spectrum of high-dimensional Hamiltonians // *Comput. Methods in Appl. Math*. 2011. V. 11, № 3. P. 273-290.

[13] *Griebel M., Oeltz D.* A sparse grid space-time discretization scheme for parabolic problems // *Computing*. 2007. V. 81. P. 1-34. http://dx.doi.org/10.1007/s00607-007-0241-3.

[14] *von Petersdorff T., Schwab C.* Numerical solution of parabolic equations in high dimensions // *ESAIM: Mathematical Modelling and Numerical Analysis*. 2004. V. 38, № 01. P. 93-127. doi: 10.1051/m2an:2004005. http://dx.doi.org/10.1051/m2an:2004005.

[15] *de Silva V., Lim L.-H.* Tensor rank and the ill-posedness of the best low-rank approximation problem // *SIAM J. Matrix Anal. Appl*. 2008. V. 30, № 3. P. 1084–1127. doi: 10.1137/06066518x.

[16] *Oseledets I. V., Tyrtyshnikov E. E.* Breaking the curse of dimensionality, or how to use SVD in many dimensions // *SIAM J. Sci. Comput*. 2009. V. 31, № 5. P. 3744-3759. doi: 10.1137/090748330.

[17] *Oseledets I. V.* Tensor-train decomposition // *SIAM J. Sci. Comput.* 2011. V. 33, № 5. P. 2295-2317. doi: 10.1137/090752286.

[18] *Holtz S., Rohwedder T., Schneider R.* On manifolds of tensors of fixed TT–rank // *Numer. Math.* 2011. doi: 10.1007/s00211-011-0419-7.

[19] *Östlund S., Rommer S.* Thermodynamic limit of Density Matrix Renormalization // *Phys. Rev. Lett.* 1995. V. 75, № 19. P. 3537–3540. doi: 10.1103/PhysRevLett.75.3537. `http://link.aps.org/doi/10.1103/PhysRevLett.75.3537`.

[20] *White S. R., Huse D.* Numerical renormalization-group study of low-lying eigenstates of the antiferromagnetic S=1 Heisenberg chain // *Phys. Rev. B.* 1993. V. 48, № 6. P. 3844–3852. doi: 10.1103/PhysRevB.48.3844. `http://prb.aps.org/abstract/PRB/v48/i6/p3844_1`.

[21] *White S. R.* Density-matrix algorithms for quantum renormalization groups // *Phys. Rev. B.* 1993. V. 48, № 14. P. 10345–10356. doi: 10.1103/PhysRevB.48.10345.

[22] *Rohwedder T., Holtz S., Schneider R.* The alternation least squares scheme for tensor optimisation in the TT-format: Preprint DFG-Schwerpunktprogramm 1234 71: 2010.

[23] *Kolda T. G., Bader B. W.* Tensor decompositions and applications // *SIAM Review.* 2009. V. 51, № 3. P. 455–500. doi: 10.1137/07070111X.

[24] *Khoromskij B. N.* Tensor-structured numerical methods in scientific computing: survey on recent advances // *Chemometr. Intell. Lab. Syst.* 2012. V. 110, № 1. P. 1-19. doi: 10.1016/j.chemolab.2011.09.001.

[25] *Khoromskij B. N.* Introduction to tensor numerical methods in scientific computing: Preprint, Lecture Notes 06-2011: University of Zürich, 2010. `http://www.math.uzh.ch/fileadmin/math/preprints/06_11.pdf`.

[26] *Meyer H.-D., Gatti F., Worth G. A.* Multidimensional Quantum Dynamics: MCTDH Theory and Applications. — Weinheim: Wiley-VCH, 2009.

[27] *Hackbusch W., Kühn S.* A new scheme for the tensor representation // *J. Fourier Anal. Appl.* 2009. V. 15, № 5. P. 706–722.

[28] *Khoromskij B. N., Khoromskaia V., Flad. H.-J.* Numerical solution of the Hartree–Fock equation in multilevel tensor-structured format // *SIAM J. Sci. Comput.* 2011. V. 33, № 1. P. 45-65. doi: 10.1137/090777372.

[29] *Khoromskij B. N., Khoromskaia V.* Multigrid accelerated tensor approximation of function related multidimensional arrays // *SIAM J. Sci. Comput.* 2009. V. 31, № 4. P. 3002-3026. doi: 10.1137/080730408.

[30] *Oseledets I. V.* Approximation of $2^d \times 2^d$ matrices using tensor decomposition // *SIAM J. Matrix Anal. Appl.* 2010. V. 31, № 4. P. 2130-2145. doi: 10.1137/090757861.

[31] *Khoromskij B. N.* $\mathcal{O}(d \log N)$–Quantics approximation of N–d tensors in high-dimensional numerical modeling // *Constr. Appr.* 2011. V. 34, № 2. P. 257-280. doi: 10.1007/s00365-011-9131-1.

[32] *Kazeev V., Khoromskij B. N.* Explicit low-rank QTT representation of Laplace operator and its inverse: Preprint 75. — Leipzig: MPI MIS, 2010. www.mis.mpg.de/preprints/2010/preprint2010_75.pdf.

[33] *Dolgov S. V., Khoromskij B. N., Savostyanov D. V.* Multidimensional Fourier transform in logarithmic complexity using QTT approximation: Preprint 18. — Leipzig: MPI MIS, 2011. http://www.mis.mpg.de/preprints/2011/preprint2011_18.pdf.

[34] *Dolgov S. V., Oseledets I. V.* Solution of linear systems and matrix inversion in the TT-format: Preprint 19. — Leipzig: MPI MIS, 2011. http://www.mis.mpg.de/preprints/2011/preprint2011_19.pdf.

[35] *Khoromskij B. N., Oseledets I. V.* DMRG+QTT approach to computation of the ground state for the molecular Schrödinger operator: Preprint 69. — Leipzig: MPI MIS, 2010. www.mis.mpg.de/preprints/2010/preprint2010_69.pdf.

[36] *Kellogg R. B.* An Alternating Direction Method for Operator Equations // *SIAM*. 1964. V. 12, № 4. P. 848-854. doi: DOI:10.1137/0112072. http://dx.doi.org/10.1137/0112072.

[37] *Oseledets I. V.* Constructive representation of functions in tensor formats: Preprint 2010-04. — Moscow: INM RAS, 2010. http://pub.inm.ras.ru.

[38] *Bernstein S.* Leçons sur les propriétés extrémales et la meilleure approximation des fonctions analytiques d´une variable réelle. — Paris: Gauthier-Villars, 1926.

[39] *Tadmor E.* The exponential accuracy of Fourier and Chebychev differencing methods // *SIAM J. Numer. Anal.* 1986. V. 23. P. 1-23.

[40] *Barrett J. W., Süli E.* Finite element approximation of kinetic dilute polymer models with microscopic cut-off // *ESAIM: Mathematical Modelling and Numerical Analysis*. 2011. V. 45. P. 39-89.

[41] *Gillespie D. T.* A rigorous derivation of the chemical master equation // *Physica A: Statistical Mechanics and its Applications*. 1992. V. 188, №1-3. P. 404 - 425. http://www.sciencedirect.com/science/article/pii/037843719290283V.

[42] *Oseledets I. V., Tyrtyshnikov E. E.* TT-cross algorithm for the approximation of multidimensional arrays // *Linear Algebra Appl.* 2010. V. 432, № 1. P. 70-88. doi: 10.1016/j.laa.2009.07.024.

[43] *Savostyanov D. V., Oseledets I. V.* Fast adaptive interpolation of multi-dimensional arrays in tensor train format // *Proceedings of nDS-2011 Conference*. IEEE, 2011.

[44] *Temlyakov V.* Greedy Approximation. — Cambridge University Press, 2011.

[45] *Binev P., Cohen A., Dahmen W. et al.* Convergence Rates for Greedy Algorithms in Reduced Basis Methods // *SIAM Journal on Mathematical Analysis*. 2011. V. 43, № 3. P. 1457-1472. doi: 10.1137/100795772. http://link.aip.org/link/?SJM/43/1457/1.