Max-Planck-Institut
für Mathematik
in den Naturwissenschaften
Leipzig

Tensor-product approach to global time-space-parametric discretization of chemical master equation

(revised version: November 2012)

by

Sergey Dolgov and Boris N. Khoromskij

Preprint no.: 68 2012

Tensor-product approach to global time-space-parametric discretization of chemical master equation

Sergey Dolgov[$,†], Boris Khoromskij[†]

[†] *Max-Planck Institute for Mathematics in Sciences,*
Germany, 04103 Leipzig, Inselstraße 22
[sergey.dolgov,bokh]@mis.mpg.de

Abstract. We study the application of the novel tensor formats (TT, QTT, QTT-Tucker) to the solution of d-dimensional chemical master equations, applied mostly to gene regulating networks (signaling cascades, toggle switches, phage-λ). For some important cases, e.g. signaling cascade models, we prove good separability properties of the system operator. The time is treated as an additional variable, with the Quantized tensor representations (QTT, QTT-Tucker) employed, leading to the log-complexity in the system size. This global space-time $(d+1)$-dimensional system, approximated in the QTT or QTT-Tucker formats, is solved in the block-diagonal form by the ALS-type iterations. Another issue considered is the quantification of uncertainty, which means that some model parameters are not known exactly, but only their ranges can be estimated. It occurs frequently in real-life systems. In this case, we introduce the unknown parameters as auxiliary variables discretized on the corresponding grids, and solve the global space-parametric system at once in the tensor formats.

Keywords: multilinear algebra, tensor products, chemical master equation, parameter dependent problems
AMS Subject Classification: 65F50, 15A69, 65F10, 82C31, 80A30, 34B08,

1. Introduction

The paper is devoted to the solution of a chemical master equation in structured tensor formats. This problem arises mostly in the modeling of chemical reactions (kinetics) in genetic regulatory networks, cell systems and so on. Typically the number of molecules of a given chemical species in intra-cellular systems is up to the order of hundreds. At such concentrations, solution of systems in terms of ordinary differential equations is inappropriate, since stochastic fluctuations in numbers of molecules (of the relative order of 10^{-1}) play an important role in the evolution of the system [1, 2]. In such a case, we cannot ignore the stochasticity of the system. The stochastic description of the chemical reaction kinetics is

[$] Partially supported by RFBR grants 12-01-00546-a, 11-01-12137-ofi-m-2011, 11-01-00549-a, 12-01-33013 mol-ved-a, 12-01-31056, by Rus. Gov. Contracts П1112, 14.740.11.0345, 14.740.11.1067, 16.740.12.0727, and Priority Research Program OMN-3 at Institute of Numerical Mathematics, Russian Academy of Sciences, Russia, 119333 Moscow, Gubkina 8.

given by the chemical master equation (CME), see (2) below, which simulates the probability distribution describing the state of the system [3, 4, 5].

During the history of the chemical kinetics modeling in biology, different approaches were developed. Monte Carlo methods are based on a statistically large ensemble of realizations of the stochastic process associated with the CME. The most famous is the stochastic simulation algorithm (SSA) by Gillespie [4]. Several improvements include the advanced sampling techniques [6], τ-leaping methods [7], system-partitioning hybrid methods [8, 9]. Additionally, the chemical Fokker-Planck equation may be considered [10] to treat high-concentration systems, which can be discretized on coarser grids than the CME.

In order to analyze a stochastic reaction system, all Monte Carlo techniques require a lot of realizations. Sometimes important biological events may be very rare, and extremely large number of realizations may be required to catch the relevant statistics.

A principal alternative to the Monte Carlo-type methods is the solution of the master equation directly as a linear ODE. For many systems, it is observed that the probability distribution vanishes rapidly outside a bounded domain. Thus, it is possible to truncate the state space to a finite domain, and approximate the exact solution by the solution of the resulting truncated CME [11].

However, even the truncated state space volume is usually still very large, and grows exponentially with the number of species. This problem is called a *curse of dimensionality* since [12]. For example, a system with 10 species and typical concentrations of about 100 would be described by 100^{10} float numbers, which is infeasible on any supercomputer. So, some low-parametric approximation is needed.

As such reduced grid-based methods, we would like to mention the sparse grids technique [13], as well as the initial tensor-structured approaches: the greedy algorithm in the canonical tensor format [14, 15, 16, 17, 18], and the solver in the Tucker format, based on the manifold dynamics via the projection onto the tangent space [19]. The latter exploits the so-called *Dirac-Frenkel* principle to propagate the system on the tensor product manifold. For a detailed description we refer to [20, 21, 22].

Alternatively, one can formulate an implicit time propagation scheme (Euler, Crank-Nicolson), and apply some tensor structured solver to the linear system involved. Moreover, we may consider the time as an independent variable and solve the coupled space-time system at once, taking benefits from the reduction of complexity based on the tensor structuring. This technique can be considered as an adaptive construction of the tensor manifold without a priori knowledge on the system, in contrast to the propagation of a predefined manifold in the tensor Dirac-Frenkel scheme. The initial application of this approach to the Fokker-Planck equation was given in [23], and here we shall use it for the chemical master equation as well. For another coupled space-time technique for hyperbolic equations see [24].

A certain part of this paper contains the theoretical analysis of the tensor representation of the linear system matrix, which demonstrates a nice structure. As a bi-product, we prove TT rank estimates for Hamiltonians of the Heisenberg (XYZ) spin system models. Provided that a solution can be computed with a prescribed accuracy, we are free from an additional error analysis, which is required in Monte Carlo or hybrid methods.

As such a tensor-structured linear solver, we use the Alternating Minimal Residual (AMR) method [25] (the one-block version AMR($\mathtt{one}$)). It employs the ideas of the alternating least squares (ALS) and MALS (DMRG) methods [26, 27, 28, 29], improved by combining with the GMRES approach to determine a new correction direction. The initial verification of the AMR methods and their comparison with the DMRG was conducted on some examples of the Fokker-Planck and master equations. It was found, that the ALS and MALS techniques fail to compute the solution of nonsymmetric systems (which appear in the CME simulation), and recasting the problem to the normal equations is too expensive, whereas the AMR methods are much faster and provide the desired accuracy, see Section 5.1 for an additional discussion. Here we give a thorough study of more complicated CME models.

The paper is organized as follows. In Section 2, the chemical kinetic and master equations are presented, as well as the parameter-dependent problems, and basic properties are mentioned. Section 3 describes tensor formats, principal techniques and corresponding notations. Section 4 is devoted to some typical CME operators and their analytic tensor-structured representations. In Section 5 we present the main computational schemes for the time propagation, as well as the stationary solution. Finally, Sections 6 and 7 provide the illustrative numerical experiments and conclusion.

2. Problem statement

2.1. From a stochastic to deterministic model

Suppose that d different active chemical species $S_1, ..., S_d$ are given, and they can react in M reaction channels. Denote the vector of their concentrations in terms of number of molecules as $\mathbf{x} = (x_1, ..., x_d)$, $x_i \in (\{0\} \cup \mathbb{N})$. Each channel is specified by the *stoichiometric vector* $\mathbf{z^m} \in \mathbb{Z}^d$, and the *propensity* function $w^m(\mathbf{x}) : \ (\{0\} \cup \mathbb{N})^d \to \mathbb{R}, \ m = 1, ..., M$.

The deterministic ODE on the concentrations is written as follows:

$$\frac{dx_i}{dt} = \sum_{m=1}^{M} z_i^m w^m(\mathbf{x}), \quad i = 1, ..., d. \tag{1}$$

However, in some cases (e.g. small concentrations), the reaction occurrence is a stochastic process, and (1) does not hold any more in the deterministic sense. Instead of stochastic simulation (SSA, [4]), one may consider a *deterministic* difference equation on the joint probability density - the chemical master equation [3], which reads

$$\frac{dP(\mathbf{x}, t)}{dt} = AP = \sum_{m=1}^{M} w^m(\mathbf{x} - \mathbf{z^m})P(\mathbf{x} - \mathbf{z^m}, t) - w^m(\mathbf{x})P(\mathbf{x}, t), \tag{2}$$

where

$$P(\mathbf{x}, t) : (\{0\} \cup \mathbb{N})^d \to \mathbb{R}$$

is the joint probability of species $S_1, ..., S_d$ to be presented in the system in concentrations $x_1, ..., x_d$ at the time t.

Since $\mathbf{x}$ represents the number of molecules, its values belong to $(\{0\} \cup \mathbb{N})^d$. Thus, the discretization grid arises naturally from the model:

$$x_i \in \{x_i(j_i)\} = \{j_i\}, \quad j_i = 0, 1, ..., \quad i = 1, ..., d.$$

The chemical master equation (CME) can be considered as a discrete partial differential equation with spatial differences instead of derivatives. Note that in the exact formulation j_i are not bounded. However, very large values are unlike, and we consider the finite state space projection (FSP) [11] on a cubic grid: each j_i is considered in a finite range $j_i = 0, ..., N_i - 1$.

In the following, we will use a more convenient counterpart to (2) via the shift matrices. Denote

$$J^z = \begin{bmatrix} 0 & & & & \\ \vdots & \ddots & & & \\ 1 & & \ddots & & \\ 0 & \ddots & & \ddots & \\ & & 1 & \cdots & 0 \end{bmatrix} \leftarrow |z|\text{-th row}, \quad \text{if } z \leqslant 0, \text{ and } \quad J^{-z} = J^{z\top}. \tag{3}$$

Now we write (2) as a discrete difference equation:

$$\frac{dP(\mathbf{x},t)}{dt} = \sum_{m=1}^{M} (\mathbf{J}^{\mathbf{z}^m} - \mathbf{J}^0)\operatorname{diag}(w^m(\mathbf{x}))P(\mathbf{x},t), \tag{4}$$

where the multidimensional shift operator reads

$$\mathbf{J}^{\mathbf{z}} = J^{z_1} \otimes \cdots \otimes J^{z_d},$$

and $\operatorname{diag}(w)$ is a diagonal matrix with the values of $w(\mathbf{x})$ at all grid points. Note that $\mathbf{J}^0$ is just an identity matrix of proper sizes.

The operator A possesses the following properties:

1. $A \leqslant 0$, so that $P(\mathbf{x},t) \to P^*(\mathbf{x}),\ t \to \infty$ - the stationary solution, $AP^* = 0$.
2. $A^\top \cdot \mathbf{1} = 0$, so that the total probability normalization $\sum\limits_{\mathbf{x}} P(\mathbf{x},t)$ is conserved in time. One usually normalizes $\sum\limits_{\mathbf{x}} P(\mathbf{x},t) = 1$.
3. The stoichiometric vectors can be split into two parts: $\mathbf{z}^\mathbf{m} = \mathbf{z}^{\mathbf{m}+} + \mathbf{z}^{\mathbf{m}-}$, where $z_i^{m+} = \max(z_i^m, 0)$, and $z_i^{m-} = \min(z_i^m, 0)$, $i = 1, ..., d$. As will be considered in more details in Section 4.1, a reaction in the channel $\mathbf{z}^{\mathbf{m}+}$ corresponds to the increasing of the specie concentration, whereas the reaction $\mathbf{z}^{\mathbf{m}-}$ leads to the destruction of the molecules. Since the number of molecules can not become negative, the following boundary condition has to be imposed:
$$w^m(x^{0-}) = 0, \quad x_i^{0-} = 0, \text{ if } z_i^{m-} \neq 0.$$
In other words, the probability of the concentration-decreasing reaction is zero, if the corresponding species are absent. Thus, the concentrations can not be advanced into the negative domain.

2.2. Parametric problems and uncertainty quantification

In addition to the state variables x, there can be auxiliary dimensions. Especially in biological models which are under consideration here, some of the coefficients (e.g. in propensities) might be given not exactly, but in a certain range. The goal might be in finding such *parameter* values so that the model quantities are as close as possible to the observed ones.

This is the case when tensor methods are worth to be used: we introduce some discretization grid in each parameter variable in its range (e.g. uniform or Chebyshev), and add this new dimension to the model equation. We hope that the dependence on a parameter is not very strong, and the coupled system can be approximated efficiently in a tensor-structured format. Thus, the complexity of such a space-time-parametric solution is expected to be lower than of solutions computed at all parametric points independently.

Such a parameter fitting is called the *uncertainty quantification* (after the "uncertainly" defined coefficients, traces back to [30]), and was also considered in the framework of stochastic PDEs, reformulated in terms of parameter-dependent equations [31]. Specific *greedy* approach to the Chemical Master Equation was presented in [14].

The general parametric problem is formulated as follows. Suppose the matrix A depends on a parameter $A = A(y)$, and the parametric grid $y_1, ..., y_N$ is given. We assemble the global system

$$\frac{\partial P}{\partial t} = \mathcal{A}P = \begin{bmatrix} A(y_1) & & \\ & \ddots & \\ & & A(y_N) \end{bmatrix} \begin{bmatrix} P(y_1) \\ \vdots \\ P(y_N) \end{bmatrix}.$$

Despite the diagonality, since N might be large (if y is in fact a multiparameter tuple), it is more efficient to consider the structured tensor solution of the global space-parametric system. That is, we agree formally that $x_{d+1} = y$, $\tilde{x} = (x_1, ..., x_d, y)$, and formulate the corresponding larger dimensional problem with the new matrix $\mathcal{A}$, and $P(\mathbf{x}, y_i, t)$ stacked into a one large vector $P(\tilde{\mathbf{x}}, t)$.

For instance, in the CME problem, typically the parametrization is incorporated in the coefficients of the propensity functions, $w^m = w^m(\mathbf{x}, y) = w^m(\tilde{\mathbf{x}})$. Then the CME operator (4) recasts as

$$\frac{dP(\tilde{\mathbf{x}}, t)}{dt} = \mathcal{A}P(\tilde{\mathbf{x}}, t) = \sum_{m=1}^{M} \left((\mathbf{J}^{\mathbf{z}^m} - \mathbf{J}^0) \otimes I_N \right) \operatorname{diag}(w^m(\tilde{\mathbf{x}})) P(\tilde{\mathbf{x}}, t).$$

Since the range of $\mathbf{x}$ is $\left[0, ..., N_i - 1\right]^{\otimes d}$, (4) turns to the linear ODE in time of size $\prod_{i=1}^{d} N_i \leqslant N^d$. Even for moderate N and d (of order tens), the problem becomes too huge to be treated via standard methods. Even higher complexity arises in the parametric case where the range of $\tilde{\mathbf{x}}$ is $\left[0, ..., N_i - 1\right]^{\otimes d} \otimes [1, ..., N]$. Instead, we will use the tensor format approximations to reduce the number of unknowns. The simplest but robust format is the Tensor Train format (TT, Matrix Product States).

3. Tensor formats

The Matrix Product States representation appeared in the quantum physics community [32, 33, 27]. A great development was made in the modeling of quantum many-body systems. The so-called density matrix renormalization group (DMRG) [32, 33, 27] is a numerical variational technique devised to obtain the ground states of spin systems with high accuracy. It traces back to [27], and it is nowadays the most efficient method for 1-dimensional quantum systems, but its generalization to 2 or 3-dimensional *tensor network* cases is still an open question. It was then noticed, that the DMRG is a minimization method for the Rayleigh quotient in the Matrix Product States (MPS) [32], which also arise in the study of entanglement in quantum systems.

In the community of numerical analysis, the MPS was reopened by Tyrtyshnikov and Oseledets in 2009, as the so-called *tensor train format*, or simply TT-format [34, 35]. A

d-dimensional tensor $\mathbf{A}$ is said to be in the TT-format, if its elements are represented as a matrix product

$$A(i_1, \dots, i_d) = G_1(i_1)G_2(i_2)\cdots G_d(i_d), \quad i_k = 1, ..., n_k, \tag{5}$$

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 (5) scalar, *boundary conditions* $r_0 = r_d = 1$ are imposed (the so-called *open boundary conditions*). The numbers r_k, called *TT-ranks*, play the crucial role in storage and complexity estimates. Equation (5) can be recast to the sum of Kronecker products,

$$A = \sum_{\alpha_1,...,\alpha_{d-1}} G_1(:, \alpha_1) \otimes G_2(\alpha_1, :, \alpha_2) \otimes \cdots \otimes G_d(\alpha_{d-1}, :), \tag{6}$$

where the sum over α_k goes from 1 to r_k. For fixed values of $\mathbf{r} = [r_1, \dots, r_{d-1}]$ the parametric representation (5) defines a closed embedded manifold $\mathbb{TT}_{\mathbf{r}}$ [36] in the linear space of all d-tensors. It is clear, that all *TT-cores* $G_k(i_k)$ are 3-tensors of sizes $r_{k-1} \times n_k \times r_k$, 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 $\mathcal{O}(dnr^2)$. The connection between the TT and DMRG/ALS schemes was discussed, in particular, by R. Schneider et. al., see [36, 28].

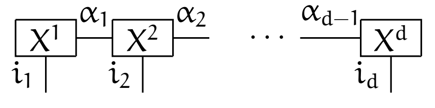
The operators (multilevel matrices) are represented in the TT format in a slightly different way. Given a matrix $\{A(i, j)\} = \{A((i_1, ..., i_d), (j_1, ..., j_d))\}$, we perform the index permutation and decompose

$$A(i_1, j_1, \dots, i_d, j_d) = G_1(i_1, j_1)G_2(i_2, j_2)\cdots G_d(i_d, j_d).$$

This format is consistent in a sense, that it reduces to the standard matrix Kronecker product in the 2D case and $r_1 = 1$ (cf. (6)).

The functional (or parametric) representation (5) is convenient in constructive representations, since the TT blocks contraction can be written as the standard product ("$\cdot$") of matrices, depending on mode indices i_k as parameters. The equations are meant to hold for all possible values of mode indices - parameters. Due to that fact, we may omit the mode indices, if we are considering the TT structure only without specifying the dependence on mode indices. The Kronecker representation (6) is used mostly for rank-1 tensors or matrices, and to describe the mode structure.

One may use also the diagrammatic notations, introduced by White and then used in [28, 37, 38] as especially convenient to demonstrate tensor networks: a multiindex array (*block*) is represented by a rectangle, its indices by lines, and if a line connects two rectangles, the new array is computed by multiplying the elements of the connected blocks and summing by the common index. For example, the TT (MPS) structure looks like



Traditional and commonly used tensor representations in multilinear algebra and numerical analysis include canonical and Tucker formats, see the surveys and lecture notes [26, 39, 40]. The canonical rank-R format is the representation of form

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

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

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

If a 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, since the corresponding optimization problem is ill-posed. Thus this format can not be used in conjunction with the time-stepping schemes, since the truncation has to be done at each step. The Tucker format can be used for small and moderate values of d. 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 [41, 42]. In quantum molecular dynamics simulation, the Tucker format was successfully used in the MCTDH framework (see the book [43]). 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 [44]. It also has an analog in the physics community, the so-called *Tensor Tree Networks* (TTN) representation [45].

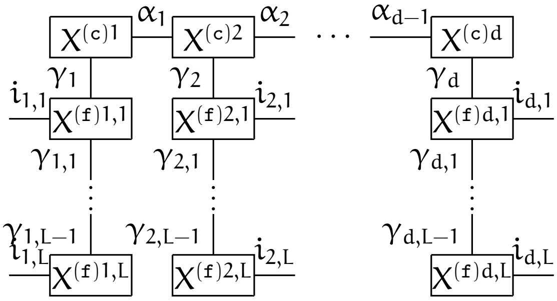
In this paper, along the line with the TT-format, the so-called *Quantized-TT (QTT)*-format is used. The idea is as follows. Suppose that the one-dimensional grid size is a power of 2, i.e. $n = 2^L$. Then a tensor can be reshaped to a $D = dL$-dimensional tensor with mode sizes equal to 2. After that, the TT-decomposition is applied to this tensor. If the TT-ranks of this D-tensor (or *QTT-ranks*) are small, then the *logarithmic complexity*, $\mathcal{O}(d \log n)$, is attained. The idea of TT applied to reshaped tensors with virtual dimensions was first proposed in [46] for $2^L \times 2^L$ matrices, and then generalized to the class of function-related tensors in [47] being named the QTT format, where its beneficial approximation properties were established. Moreover, the QTT-format allows simple constructive representations of basic operators (Laplacian, gradient and divergence operators) [48] on uniform tensor grids.

The logarithmic dependence on the one-dimensional grid size makes it a very promising tool for high-dimensional problems. For example, one may use the global space-time discretization of a differential equation, using the QTT at least in the time variable [23, 24].

There are also algorithms, which exploit the binary QTT structure heavily, such as the super-fast data-sparse Fourier transform [37].

The small mode sizes (2 for the QTT-format) motivated the construction of a fast approximate DMRG-like solver for a QTT-structured matrix. Such a solver, the *TT-Solve* algorithm, was described in [29] (see also [49] for the eigenvalue solver of this type and [28], where the general ALS-type schemes are discussed). However, for a large mode size its n^2 storage complexity becomes prohibitive, and recently the new alternating techniques have been developed [25]. See also the discussion in Section 5.1.

However, in some cases, the ranks of the straightforward QTT representation described above (referred later as *linear QTT*) grow very rapidly with the accuracy. To overcome this problem, some new tensor representation was proposed, called the *QTT-Tucker format* [38]. It exploits the QTT approximation not for the TT cores, but for the Tucker factors instead, thus keeping the entanglement of physical variables separately from the virtual ones. To get rid of the curse of dimensionality, the Tucker core is stored in the TT format. From these considerations, one winds up with the following tensor network:



(7)

The TT-Tensor

$$X^{(e)k}_{\alpha_{k-1},\alpha_k}(i_k) = X^{(c),k}(\alpha_{k-1},\alpha_k)X^{(f)k,1}(i_{k,1})\cdots X^{(f)k,L}(i_{k,L}),$$

corresponding to the quantized k-th Tucker factor, connected with one core block, is called the *extended factor* ([38], Def. 4), and the initial tensor is now the product of the extended factors,

$$X(i_1,\dots,i_d) = X^{(e)1}(i_1)\cdots X^{(e)d}(i_d).$$

In the numerical experiments (Section 6), we will investigate both the linear QTT and the QTT-Tucker formats.

4. TT representation of typical CME operators

To employ tensor decompositions, we need to present all initial data in our favorable format. Assuming the tensor separability of each propensity function w^m, we obtain immediately the rank estimate of the whole operator:

$$\operatorname{rank}(A) \leqslant \sum_{m=1}^{M} 2\cdot\operatorname{rank}(w^m). \tag{8}$$

We use here the rank-1 form of any $\mathbf{J}^{\mathbf{z}}$, TT-addition rule, and the fact that the diagonal matrix is constructed from a vector without changing the TT ranks.

4.1. Reversible monomolecular reactions

A case of special interest is the reactions of form

$$\emptyset \leftrightarrows S_i, \quad S_i \leftrightarrows S_k$$

(the first one denotes creation/destruction of a specie without involving the others), when each $\mathbf{z}^{\mathbf{m}}$ contains only $+1$ or -1 at just one position, and $\sum\limits_{m=1}^{M} z_i^m = 0,\ i = 1,\dots,d$. Such reactions appear frequently in gene regulatory networks, such as switches, cascades, etc. (see below). We have thus $M = 2d$ reactions, and (4) can be separated in two parts, corresponding respectively to the *creation* and *destruction* of a specie:

$$\frac{dP(x,t)}{dt} = \sum_{m=1}^{d}(\mathbf{J}^{m+} - \mathbf{J}^0)\operatorname{diag}(w^{m+}(\mathbf{x}))P(\mathbf{x},t) + \sum_{m=1}^{d}(\mathbf{J}^{m-} - \mathbf{J}^0)\operatorname{diag}(w^{m-}(\mathbf{x}))P(\mathbf{x},t), \tag{9}$$

w^{m+} is the propensity corresponding to creation reactions with $\mathbf{z^m} \geqslant 0$, w^{m-} corresponds to $\mathbf{z^m} \leqslant 0$, and

$$\mathbf{J}^{m+} = \underbrace{J^0 \otimes \cdots \otimes J^{-1}}_{m} \otimes J^0 \cdots \otimes J^0, \quad \mathbf{J}^{m-} = \underbrace{J^0 \otimes \cdots \otimes J^{1}}_{m} \otimes J^0 \cdots \otimes J^0.$$

Thus the shift operators in each part can be collected into rank-1 difference operators, acting only on x_m:

$$\nabla_m^- = \underbrace{J^0 \otimes \cdots \otimes (J^0 - J^{-1})}_{m} \otimes J^0 \cdots \otimes J^0, \quad \nabla_m^+ = \underbrace{J^0 \otimes \cdots \otimes (J^1 - J^0)}_{m} \otimes J^0 \cdots \otimes J^0. \tag{10}$$

Now, the CME (9) reads

$$\begin{gathered}\frac{dP(\mathbf{x},t)}{dt} = AP = (A^+ + A^-)P, \\ A^+ = -\sum_{m=1}^{d} \nabla_m^- \mathrm{diag}(w^{m+}(\mathbf{x})), \quad A^- = \sum_{m=1}^{d} \nabla_m^+ \mathrm{diag}(w^{m-}(\mathbf{x})).\end{gathered} \tag{11}$$

which has the very close form to the diffusion equation discretized using the finite difference scheme.

The tensor rank of the whole operator A is straightforwardly estimated as the sum of the ranks of the propensities, see (8). However, we would like to consider typical gene networks in more details.

4.2. Signaling cascade genetic model

A cascading process occurs when (usually) adjacent genes produce a protein which influences on the expression of the succeeding gene, see Fig. 1. This is a typical model in genetic networks; as an example, the lytic phase of the λ-phage system [50] can be considered. A mutually repressing gene pair, or gene toggle (Fig. 2), that can be found in such systems, is also a case of the cascade model, with the dimension 2.

Figure 1. Cascade signaling network

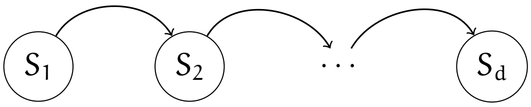
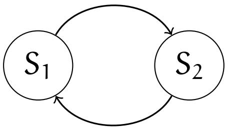


Figure 2. Toggle switch



In many such cases the m-th destruction propensity depends only on x_m, thus its rank-1 decomposition reads

$$w^{m-}(\mathbf{x}) = e_1 \otimes \cdots \otimes w^{m-}(x_m) \otimes \cdots \otimes e_d,$$

where e_i is the vector of all ones. Now, the whole destruction operator A^- in (11) has the Laplace-like form

$$A^- = D_1 \otimes J^0 \cdots \otimes J^0 + \cdots + J^0 \otimes \cdots \otimes D_d, \quad D_m = (J^1 - J^0)\mathrm{diag}(w^{m-}(x_m)),$$

which is proven to have TT rank 2 [48].

The creation part is usually more complicated (it contains feedbacks between species), and depends on several variables. In the cascade networks (including also the toggle switch), the m-th creation propensity depends on x_{m-1} (or x_{m+1}), and, probably, on x_m. Thus, the corresponding operator part sums the two-variables terms:

$$A^+ = D_1^1 \otimes J^0 \cdots \otimes J^0 + D_1^2 \otimes D_2^2 \otimes J^0 \cdots \otimes J^0 + \cdots + J^0 \cdots \otimes D_{d-1}^d \otimes D_d^d,$$

where $D_{m-1}^m = \operatorname{diag}(w^{m+}(x_{m-1}))$, $D_m^m = -(J^0 - J^{-1})$ (we assume here that w^{m+} does not depend on x_m; the generalization will be given in Remark 1). What is nice, such a sum can be represented as a rank-3 TT-tensor, as shown in Lemma 1. Thus, its storage is linear in d.

Lemma 1. Given the matrices $E_k,\ F_k^k,\ F_k^{k+1} \in \mathbb{R}^{N_k \times N_k}$. The cascadic sum

$$H = F_1^1 \otimes \left(\bigotimes_{k=2}^{d} E_k\right) + \sum_{i=2}^{d} \left(\bigotimes_{k=1}^{i-2} E_k\right) \otimes F_{i-1}^i \otimes F_i^i \otimes \left(\bigotimes_{k=i+1}^{d} E_k\right) \tag{12}$$

possesses an explicit exact rank-3 TT decomposition $H = H^1(i_1, j_1) \cdots H^d(i_d, j_d)$, where

$$H^1 = \begin{bmatrix} E_1 & F_1^2 & F_1^1 \end{bmatrix}, \quad H^k = \begin{bmatrix} E_k & F_k^{k+1} & 0 \\ 0 & 0 & F_k^k \\ 0 & 0 & E_k \end{bmatrix}, \quad H^{d-1} = \begin{bmatrix} F_{d-1}^d & 0 \\ 0 & F_{d-1}^{d-1} \\ 0 & E_{d-1} \end{bmatrix}, \quad H^d = \begin{bmatrix} F_d^d \\ E_d \end{bmatrix}.$$

For the Tucker decomposition the same rank-3 bound holds.

Proof. We begin to split the dimensions recursively, extracting the linearly independent elements, in the same way as in the TT-SVD algorithm [35]. So, the first step reads [1]

$$H = \begin{bmatrix} E_1 & F_1^2 & F_1^1 \end{bmatrix} \begin{bmatrix} F_2^3 F_3^3 \cdots E_d + \cdots + E_2 \cdots F_{d-1}^d F_d^d \\ F_2^2 E_3 \cdots E_d \\ E_2 \cdots E_d \end{bmatrix}.$$

The first term here is exactly the first TT block of the decomposition. Now, suppose we have the following form

$$H_k = \begin{bmatrix} F_k^{k+1} F_{k+1}^{k+1} \cdots E_d + \cdots + E_2 \cdots F_{d-1}^d F_d^d & F_k^k E_{k+1} \cdots E_d & E_k \cdots E_d \end{bmatrix}^\top. \tag{13}$$

We split the k-th dimension of each row in the same manner,

$$H_k = \begin{bmatrix} E_k & F_k^{k+1} & 0 \\ 0 & 0 & F_k^k \\ 0 & 0 & E_k \end{bmatrix} \begin{bmatrix} F_{k+1}^{k+2} F_{k+2}^{k+2} \cdots E_d + \cdots + E_{k+1} \cdots F_{d-1}^d F_d^d \\ F_{k+1}^{k+1} E_{k+2} \cdots E_d \\ E_{k+1} \cdots E_d \end{bmatrix},$$

and derive the k-th TT block,

$$\begin{bmatrix} E_k & F_k^{k+1} & 0 \\ 0 & 0 & F_k^k \\ 0 & 0 & E_k \end{bmatrix}, \quad k = 2, ..., d-2. \tag{14}$$

[1] In the proof of the lemma, by the indexless quantities we mean not the full tensors (e.g. like in the representation (6)), but the parametric matrices, constructed by fixing mode indices in (5). No ambiguity arises since we are not considering the mode structure of these matrices, but only their appearance in the TT representation.

The rest dimensions are presented in the same form as (13), so we can continue the splitting. The last two blocks are splitted as follows

$$\begin{bmatrix} F^{d}_{d-1} F^{d}_{d} \\ F^{d-1}_{d-1} E_{d} \\ E_{d-1} E_{d} \end{bmatrix} = \begin{bmatrix} F^{d}_{d-1} & 0 \\ 0 & F^{d-1}_{d-1} \\ 0 & E_{d-1} \end{bmatrix} \begin{bmatrix} F^{d}_{d} \\ E_{d} \end{bmatrix},$$

giving the $(d-1)$-th and d-th TT blocks, respectively. We see, that all the TT ranks are equal to 3, except the $(d-1)$-th, which is equal to 2, that confirms the claim of the lemma. To obtain the Tucker rank estimate, it is sufficient to note that each TT block contains only 3 independent elements, and follow the TT-to-Tucker procedure described in [38]. □

Remark 1. In (12), each summand is a rank-1 tensor. However, we can straightforwardly generalize it to the case, when the neighboring terms are summed from several components:

$$F^{k}_{k-1} \otimes F^{k}_{k} \to \sum_{\alpha_k=1}^{r_k} F^{k}_{k-1,\alpha_k} \otimes F^{k}_{k,\alpha_k}$$

(i.e., the rank of each propensity is not equal to 1). In this case, we can collect respectively the row and column vectors

$$F^{k}_{k-1} = \begin{bmatrix} F^{k}_{k-1,1} & \cdots & F^{k}_{k-1,r_k} \end{bmatrix}, \quad F^{k}_{k} = \begin{bmatrix} F^{k}_{k,1} \\ \vdots \\ F^{k}_{k,r_k} \end{bmatrix},$$

and the constructions (14) will be considered as block matrices, with the sizes (i.e. the TT ranks) $(2+r_k) \times (2+r_{k+1})$. Counting the linearly independent elements in each TT block, we conclude that the k-th Tucker rank is bounded by $1 + r_k + r_{k+1}$.

Remark 2. The sum (12) can be considered in the canonical format as well. However, its rank is bounded by $\mathcal{O}(d)$. In Lemma 1 we establish a refined result, that the TT ranks may be bounded by a constant independently on d, which is especially interesting in high-dimensional cases.

Corollary 1. If the destruction propensity w^{m-} depends only on x_m, and the creation propensity w^{m+} depends only on x_{m-1}, the whole CME operator admits an explicit exact TT decomposition of rank 4, $A = A^1(i_1, j_1) \cdots A^d(i_d, j_d)$, where

$$A^1 = \begin{bmatrix} J^0 \\ D^2_1 \\ D^1_1 \\ D_1 \end{bmatrix}^{\top}, \quad A^k = \begin{bmatrix} J^0 & D^{k+1}_{k} & 0 & D_k \\ 0 & 0 & D^{k}_{k} & 0 \\ 0 & 0 & J^0 & 0 \\ 0 & 0 & 0 & J^0 \end{bmatrix}, \quad A^{d-1} = \begin{bmatrix} D^{d}_{d-1} & J^0 & D_{d-1} \\ 0 & 0 & D^{d-1}_{d-1} \\ 0 & J^0 & 0 \\ 0 & 0 & J^0 \end{bmatrix}, \quad A^d = \begin{bmatrix} D^{d}_{d} \\ D_d \\ J^0 \end{bmatrix}.$$

The same rank-4 bound holds for the Tucker decomposition.

Proof. The rank-2 decomposition of the destruction operator was already discussed. For the creation part we use Lemma 1, by setting $E_k = J^0$, $F^{k}_{k-1} = \mathrm{diag}(w^{m+}(x_{m-1}))$, and $F^{k}_{k} = -(J^0 - J^{-1})$.

The straightforward TT-addition gives the rank-5 structure, but due to the fact that J^0 encounters both A^+ and A^-, the rank can be reduced as follows. The first block reads

$$\begin{bmatrix} J^0 & D_1^2 & D_1^1 & J^0 & D_1 \end{bmatrix} = \begin{bmatrix} J^0 & D_1^2 & D_1^1 & D_1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Multiplying the latter mode-independent term with the second rank-5 block, obtain

$$\begin{bmatrix} J^0 & D_2^3 & 0 & J^0 & D_2 \\ 0 & 0 & D_2^2 & 0 & 0 \\ 0 & 0 & J^0 & 0 & 0 \\ 0 & 0 & 0 & 0 & J^0 \end{bmatrix} = \begin{bmatrix} J^0 & D_2^3 & 0 & D_2 \\ 0 & 0 & D_2^2 & 0 \\ 0 & 0 & J^0 & 0 \\ 0 & 0 & 0 & J^0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

We see that after the reduction of linear dependency, the same scalar factor arose, as in the first step. So we can continue the process and come to the form claimed by the corollary. □

4.3. Application to spin models

The similar Hamiltonians arise also in the one-dimensional spin systems modeling with nearest neighbor interactions. For example, the Heisenberg (XYZ) model with open boundary conditions [51, 52], acting on $\bigotimes_{i=1}^{d} \mathbb{C}^2$, reads

$$\begin{aligned} H &= j_x H_{xx} + j_y H_{yy} + j_z H_{zz} + \lambda H_x, \\ H_{\mu\nu} &= \sum_{i=2}^{d} \left(\bigotimes_{k=1}^{i-2} E_k \right) \otimes P_\mu \otimes P_\nu \otimes \left(\bigotimes_{k=i+1}^{d} E_k \right), \\ H_\mu &= \sum_{i=1}^{d} \left(\bigotimes_{k=1}^{i-1} E_k \right) \otimes P_\mu \otimes \left(\bigotimes_{k=i+1}^{d} E_k \right), \end{aligned}$$

where E_k are the identity matrices, P_μ are the Pauli matrices ($\mu = x, y, z$):

$$P_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad P_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad P_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix},$$

and j_μ, λ are scalars. Lemma 1 and Remark 1 can be applied straightforwardly, giving the following rank estimate:

Lemma 2. The Heisenberg (XYZ) Hamiltonian admits an explicit rank-7 TT (or Tucker) representation.

Proof. Assembling the blocks

$$F_{k-1}^k = \begin{bmatrix} j_x P_x & j_y P_y & j_z P_z \end{bmatrix}, \quad F_k^k = \begin{bmatrix} P_x & P_y & P_z \end{bmatrix}^\top,$$

we reduce the problem to that is described by Lemma 1 and Remark 1, with $r_k = 3$. Taking into account the Laplacian-like structure of λH_x of rank 2, obtain the estimate $(2 + 3) + 2 = 7$. □

If some of j_x, j_y, j_z are equal to zero, the reduced models appear, for example, the Heisenberg (XY) Hamiltonian with $j_z = 0$ and TT rank 6, or Ising (ZZ) model with $j_x = j_y = 0$ and TT rank 5.

5. Computational scheme

5.1. Solving the dynamical problem

To solve equation (2), we employ the implicit Crank-Nicolson time integration scheme

$$\left(I_x - \frac{\tau}{2}A\right) P(t_{p+1}, x) = \left(I_x + \frac{\tau}{2}A\right) P(t_p, x), \quad p = 0, ..., N_t - 1, \tag{15}$$

where $\tau = t_{p+1} - t_p$ and N_t are chosen so that approximately a steady state is reached, and I_x is the identity matrix of the same sizes as A. One issue in the high dimensional cascade modeling is the long-time integration, such that the total time $T = \tau N_t$ is of the order of $100 - 1000$, and the successive integration using Formula (15) requires too many steps.

Instead, we will consider time as an additional independent variable and formulate the global space-time $(d+1)$-dimensional linear system, as it was done in [23]:

$$\left[(I_t - J_t^{-1}) \otimes I_x - \tfrac{\tau}{2}(I_t + J_t^{-1}) \otimes A\right] P(t) = e_1 \otimes \left(P(t_0) + \tfrac{\tau}{2}AP(t_0)\right), \tag{16}$$

where $J_t^{-1} \in \mathbb{R}^{N_t \times N_t}$ is the down shift matrix according to (3), I_t is the identity matrix of size N_t, and e_1 is its first column. This system can be approximated either in the linear QTT or the QTT-Tucker formats (7), with the quantization in the time variable as well, leading to a logarithmic dependence on the number of steps $\log(N_t)$. The computational benefits of this scheme w.r.t. the traditional time stepping one (15) were demonstrated in [23].

However, as we have addressed in [23], it might be more efficient not to solve (16) for the whole time range $[0, T]$, but split it to subintervals $[0, T_0]$, $[T_0, 2T_0], ..., [T - T_0, T]$, and perform such a restarted solution. In the numerical experiments for a 20-dimensional cascade (Section 6.1) we have found an optimal value $T_0 \sim 15$, whereas $T = 400$. In other words, we cast (16) to the block-diagonal form,

$$I_P \otimes \left[(I_{t'} - J_{t'}^{-1}) \otimes I_x - \tfrac{\tau}{2}(I_{t'} + J_{t'}^{-1}) \otimes A\right] P(t) = \left\{e_1 \otimes \left(P(t_p) + \tfrac{\tau}{2}AP(t_p)\right)\right\}_{p=0}^{P-1}, \tag{17}$$

where $t_p = pT_0$, $p = 0, ..., P-1$, $P = T/T_0$, and $I_{t'}, J_{t'}^{-1} \in \mathbb{R}^{N_t/P \times N_t/P}$.

As the linear solver to (16) in the structured tensor format, we employ the alternating-direction approach. Up to now, there are three (at least, most robust) realizations of this technique in the TT format:

- ALS (Alternating Linear Scheme) [28],
- MALS/DMRG (Modified ALS) [28, 29], and
- AMR (Alternating Minimal Residual) [25].

The simplest Alternating Linear Scheme (ALS), and its Modified version (MALS, a.k.a. DMRG in quantum physics community), were brought to the numerical math community in [28], and [29] contains further improvements. The main ideas of these methods are given below. For more details we refer to [28, 29].

In the ALS, we perform a succeeding optimization for each TT block, i.e. we fix all the blocks from a previous iteration, then reduce and solve the problem for the entries of a certain block (only one!), and proceed to the next block. The ALS method is known to have the following drawbacks: first, the solution ranks are fixed to those of the initial guess and cannot be updated during the iterations, and second, it might take a lot of iterations ($\sim 10^4 - 10^5$) to achieve a reasonable accuracy.

The MALS calculates in each step the elements not of one block, but of *two* succeeding blocks, contracted into a larger *supercore* with two mode indices. After its elements are found, the supercore is separated to the two updated TT cores by a simple matrix SVD. This step allows to determine the TT ranks *adaptively* during the iterations. Moreover, its convergence is usually much faster in practice than of the previous method. The cyclic sweep through all TT cores (now in fact the pairs!) is done in the same manner as in the ALS. The MALS method was used in [23] to solve the global space-time system (16).

However, since paired supercores are involved, the asymptotic complexity per iteration of the MALS is higher than of the simpler ALS.

Lemma 3. [28, 29] Suppose that the d-dimensional matrix A with the mode sizes n is given in the TT format with the ranks bound r_A, the R.H.S is given in the TT format with the ranks bound r_y, and the solution ranks are bounded by r. Then, one iteration of the ALS method requires

$$\mathcal{O}(dnr^2 + dr^2r_A + drr_y)$$

memory, and

$$\mathcal{O}(dnr^3r_A + dn^2r^2r_A^2 + dnr^2r_y)$$

operations.
One iteration of the MALS method needs

$$\mathcal{O}(dn^2r^2 + dr^2r_A + drr_y)$$

memory, and performs

$$\mathcal{O}(dn^2r^3r_A + dn^3r^2r_A^2 + dn^2r^2r_y + dn^3r^3)$$

operations.

In [38] it was shown that the TT versions of these solvers can be naturally exploited inside the corresponding algorithms for the QTT-Tucker format. Along the line, the complexity increase up to n^3 of the MALS was discussed, since there the Tucker ranks play the role of the mode sizes in a certain step. In the linear QTT $n = 2$, and a slightly higher cost of one iteration is fully compensated by a significantly reduced number of iterations of the MALS vs ALS, but if $n \sim 20$, it is already much more difficult to work with.

Another issue is that the theoretical convergence analysis of the standard ALS-type methods is hard to be provided. There are only local estimates [53], which might be too restrictive in practice. The situation becomes much more pessimistic, if the convergence does not take place at all even for the Modified ALS algorithm, as for the $20d$ example here (Section 6.1).

To get rid of these problems, the family of AMR methods was developed [25]. The linear system solution is restricted to the elements of one TT core as in the ALS, but in addition, a special rank adaptation technique is performed: we compute principal components of the residual (in practice, only a certain part of the residual may be used, giving the cost reduction, but without a serious corruption of the convergence), and then add the corresponding TT-tensor to the approximant. Since the TT-addition sums the ranks of the addends, this is the way to increase the ranks during the computations (to reduce the ranks, it is sufficient to perform the simple TT-rounding). Moreover, the next core optimization mimics now the Galerkin correction, using the residual components as the basis. This is why the name Alternating Minimal Residual has appeared, and this method is indeed much more robust in practice than (M)ALS, especially for nonsymmetric matrices, which is the case in the Chemical Master Equation.

Since all the operations involve only one block entries in each step, the method has the same asymptotic complexity as the ALS, see Lemma 3. It is the method that will be used in the numerical tests below.

5.2. Computing the stationary state

Since the CME operator $A \leqslant 0$ and does not depend on time, the dynamical problem converges eventually to a single steady state [2]. Sometimes we need only the stationary distribution, but not the transient processes. In this case, the global formulation (16) is not efficient.

Contrarily, we employ the simplest implicit Euler iteration:

$$(I - \tau A)\, P(t_{p+1}, x) = P(t_p, x), \quad p = 1, ..., N_t, \tag{18}$$

solving the linear system in the left hand side via the alternating method. Note, that here the time step τ might be quite large. The intermediate solutions do not approximate the transient processes accurately, but the method is convergent to the stationary state, if $A \leqslant 0$, which is the case in our CME model. As an additional cost reduction, we can use the following trick. After each step (18), compute the closeness of the approximant to the kernel: $\epsilon = \frac{\|AP\|}{\|P\|}$. If ϵ is large, we do not need to solve (18) very precisely. When ϵ diminishes, the accuracy may be improved. Practically, a rule of the form $\varepsilon = 10^{-1} \cdot \epsilon$, where ε is the tensor rounding and solution accuracy, is used. This approach decreases the complexity of intermediate iterations significantly.

In the following we will refer to this method as `Euler` $\circ$ `AMR`, after its two-level iterative scheme:

- outer Euler iterations in the full tensor space, and
- inner AMR iterations in the space of TT-elements to find a solution to (18).

As an alternative, one may try to solve directly the null-space problem $AP = 0$. However, the ALS and DMRG eigensolvers are designed to minimize the Rayleigh quotient and require a symmetric matrix. For the CME, one should solve $A^\top AP = 0$, which was found to be completely uncompetitive with the `Euler` $\circ$ `AMR` approach, in terms of both the cost (the conditioning and TT ranks of the matrix are squared) and accuracy (in the low-rank projected system, the lowest eigenvalue is not exactly zero; since lots of other small eigenvalues are possible, the DMRG solver may formally decrease the Rayleigh quotient below the tolerance, but the corresponding eigenvector will be completely irrelevant). A development of efficient alternating methods for nonsymmetric eigenvalue problems is a matter of future research.

[2] Note the difference with the initial reaction ODE (1), which is nonlinear and in general has a nontrivial attractor; this is not the case for the *linear* CME, this is one reason why our approach is more reliable than the direct stochastic simulation

6. Numerical experiments

The experiments were conducted on a Linux x86_64 machine with Intel Xeon E5504 processor at 2.00 GHz with the cache size 4096 KB/core. The Alternating Minimal Residual method (TT and QTT-Tucker versions) for the linear solution and approximation (for fast MatVecs) was implemented in MATLAB as a part of the TT-Toolbox[3] with the most time consuming routines called externally from C/FORTRAN MEX files.

6.1. 20-dimensional signaling cascade

First, we test the simplest but high-dimensional cascade problem from [13, 14], for which we have provided a theoretical analysis of the operator separability. The model parameters are fixes as follows:

- $d = 20$, hence $M = 40$;
- for $m = 1$: $w^{m+} = 0.7$, $z^m = -e_m$: generation of the first protein;
- for $m > 1$: $w^{m+} = \dfrac{x_{m-1}}{5 + x_{m-1}}$, $z^m = -e_m$: succeeding creation reactions;
- for $m = 1, ..., 20$: $w^{m-} = 0.07 \cdot x_m$, $z^m = e_m$: destruction reactions.

The notations $m+$, $m-$, as well as the operator assembly are according to (11), and e_m is the m-th identity vector (the practical operator construction can be done using the Corollary 1 directly, with the TT-rank bound 4).

The computational scheme specifications are the following

- Computational domain $\mathbf{x} \in [0, ..., 63]^{\otimes d}$. The PDF value at $x_i = 63$ is below the machine precision.
- Linear QTT format for space and time. Since $N = 64$, and the function decays rapidly, the QTT-rank overhead w.r.t. the TT or Tucker formats is small; moreover, the alternating scheme is even more efficient in the linear QTT, since the size of one block is $2r^2$, instead of $\tilde{r}^3$ in the QTT-Tucker format, and the ranks r and $\tilde{r}$ are almost the same. However, we will benefit from the QTT-Tucker representation for the larger grid tests below.
- We solve the dynamical problem until $T = 400$, restarting the global space-time solver as proposed in Section 5.1. We perform an additional test to find an optimal restarting parameter T_0.
- As the initial condition, we choose $P(\mathbf{x}, 0) = e_1 \otimes \cdots \otimes e_1$, i.e. all species are presented in zero concentrations with probability 1.
- Tensor rounding and solution accuracy $\varepsilon = 10^{-5}$.

[3] http://github.com/oseledets/TT-Toolbox

As a resulting quantity, we compute the mean concentrations of all species in time,

$$\langle x_i \rangle(t) = \frac{\sum\limits_{\mathbf{x}} x_i P(\mathbf{x}, t)}{\sum\limits_{\mathbf{x}} P(\mathbf{x}, t)}, \quad i = 1, ..., d,$$

which are shown in Fig. 3. Additionally, the convergence of the transient solution to the steady state is shown in Fig. 4. One of the interesting features in the cascade systems is the intrinsic delay between the equal concentrations of different species, which can be observed in Fig. 3. To keep the time solution history accurately is important to measure such delays. Fig. 4 shows that actually $T \sim 300$ is enough to approximate the stationary solution with the desired accuracy. In the rest of the time interval, some perturbations occur, since the solution is computed with almost random noise of the magnitude ε.

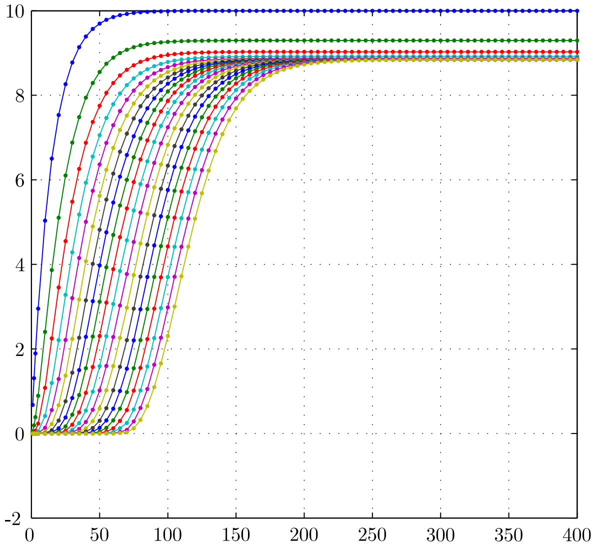


Figure 3. Mean concentrations $\langle x_i \rangle(t)$.

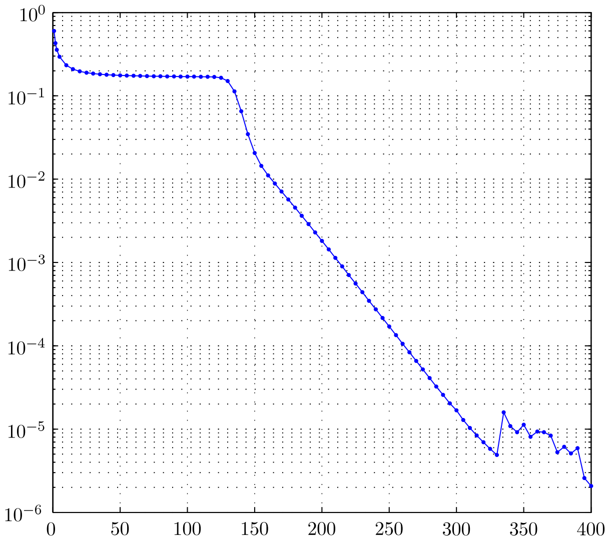


Figure 4. Closeness to the kernel $\dfrac{||AP||}{||P||}(t)$

To demonstrate the performance of the global space-time scheme, we present the CPU times of the solver with different numbers of time steps N_t in each interval $[(p-1)T_0, pT_0]$ (Fig. 5), $p = 1, ..., T/T_0$, and the time interval widths T_0 (Fig. 6). We see, that the computational time grows logarithmically with the time grid size. The fastest method was obtained by setting $T_0 = 15$. For smaller T_0, the solution in each interval is cheap, but the amount of intervals is large. For large T_0 vise versa, the conditioning (and TT ranks) of each system is high, and it takes more time for the method to converge.

6.2. A toggle switch in E.Coli with uncertainly defined coefficients

Now, consider the solution in presence of uncertainty (see Section 2.2). In this test, we simulate the synthetic genetic bistable toggle switch developed in *Escherichia coli* [54]. The CME model reads

- $d = 2$, $M = 4$:
- $w^{1+} = \dfrac{\alpha_1}{1 + x_2^{\beta}}$, $z = -e_1$: generation of S_1;

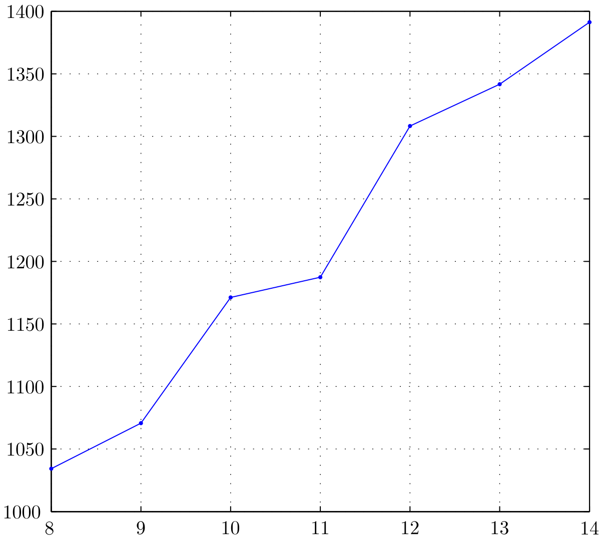


Figure 5. CPU time (sec.) versus $\log_2(N_t)$, $T_0 = 15$.

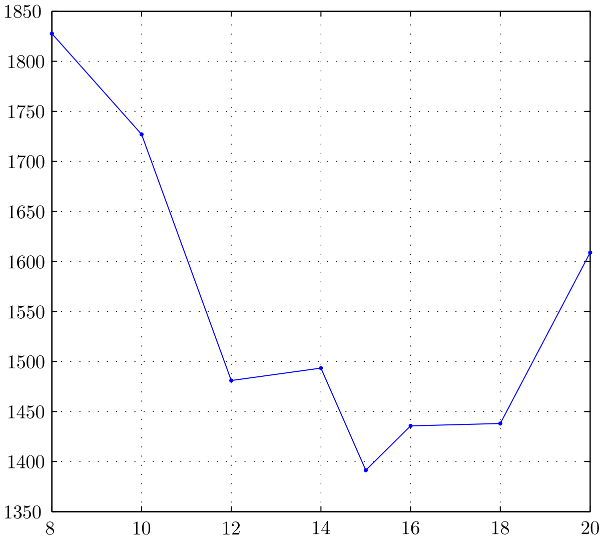


Figure 6. CPU time (sec.) versus T_0, $N_t = 2^{14}$.

- $w^{1-} = x_1$, $z = e_1$: destruction of S_1;
- $w^{2+} = \dfrac{\alpha_2}{1 + \dfrac{x_1}{(1+y/K)^\eta}}$, $z = -e_2$: generation of S_2;
- $w^{2-} = x_2$, $z = e_2$: destruction of S_2;
- $\alpha_1 = 156.25$, $\alpha_2 = 15.6$, $\beta = 2.5$, $\eta = 2.0015$, $K = 2.9618 \cdot 10^{-5}$.

The parameter y is the concentration of the IPTG catalyst, and is varying from 10^{-6} to 10^{-2}. The main feature of this system is the stabilization in the so-called *low* state (low concentration of S_2) or *high* state depending on the concentration y, see Figure 11. As a result, we end up with the 3-dimensional $\tilde{x} = (x_1, x_2, y)$ problem. The system matrix does not possess an exact TT decomposition, due to the form of w^{2+}, but still admits an accurate ε-approximation with TT ranks of the order of 10.

In our modeling, we introduce the exp-uniform grid in the parameter and seek for the steady state using the outer Euler iterations $\texttt{Euler} \circ \texttt{AMR}$ (see Section 5.2) till $T = 1000$, so that the stationarity accuracy is below the tensor rounding tolerance $\varepsilon = 10^{-5}$. The time step τ is varied from 1 to 10.

This is one of the most illustrative examples when the null-space formulation $AP = 0$ is completely useless: since the stiffness matrix is diagonal w.r.t. the parametric points, the alternating method is likely to converge to some identity vector in y. The solution at that corresponding parameter point could be computed accurately (hence formally $||AP||$ is small), but the other points will be lost.

We present the comparison of the linear and the QTT-Tucker formats for solution of the system (18). The concentration x_1 is quite high, so we choose the spatial domain $[0, ..., 511]^{\otimes 2}$, and from 2^7 to 2^{13} grid points in y.

First of all, check the computational times versus the parametric grid size (Figure 7) and the time step τ (Figure 8). The growth is asymptotically logarithmic with N_y which confirms

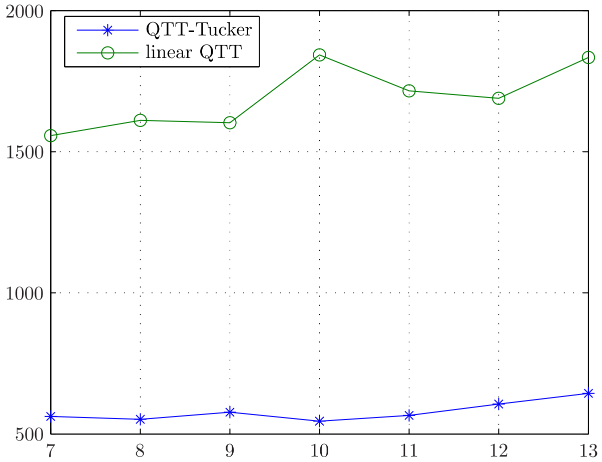


Figure 7. CPU time vs. $\log_2(N_y)$, $\tau = 5$.

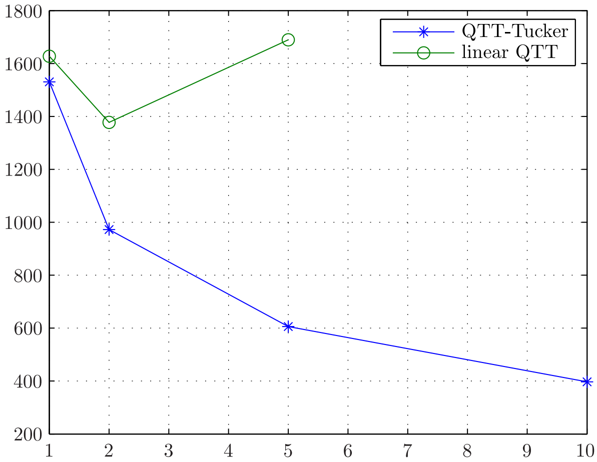


Figure 8. CPU time vs. τ, $N_y = 2^{12}$.

the advantages of the QTT format. Additionally, we see that the QTT-Tucker method is about 4 times faster; moreover, it allows to treat the case $\tau = 10$, whereas the linear QTT solver failed to converge due to a high condition number of the matrix involved.

In the second figure series, we test the accuracy of computing the average concentration $\langle x_2 \rangle$ at the parameter point $y^* = 3 \cdot 10^{-5}$; this point corresponds to a transient region (see Fig. 11), which is interesting to track. Since the grid does not contain this point exactly, the nearest-neighbor interpolation is used. As a reference value x_2^* we use that is obtained on the grid $N_y = 2^{13}$. The accuracies of computing x_2 are presented in Figures 9 and 10. In both cases, the asymptotic is better than the theoretical $\mathcal{O}(h_y)$. Though it is possible to consider more sophisticated interpolation techniques, the quantization approach allows to consider such fine grids and achieve accurate results without excessive computational cost.

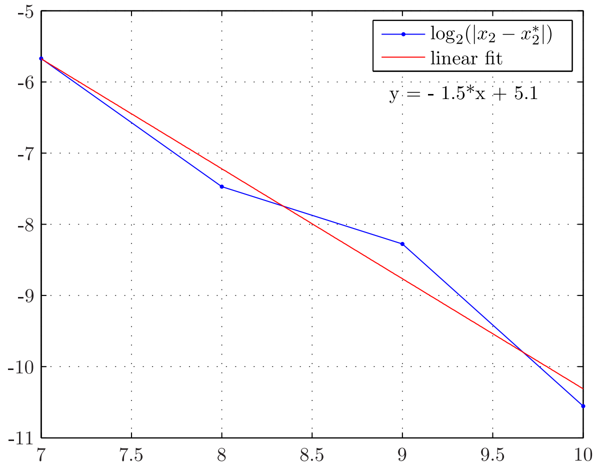


Figure 9. Interpolation accuracy of $x_2(y^*)$ vs. $\log_2(N_y)$. $\tau = 5$, linear QTT

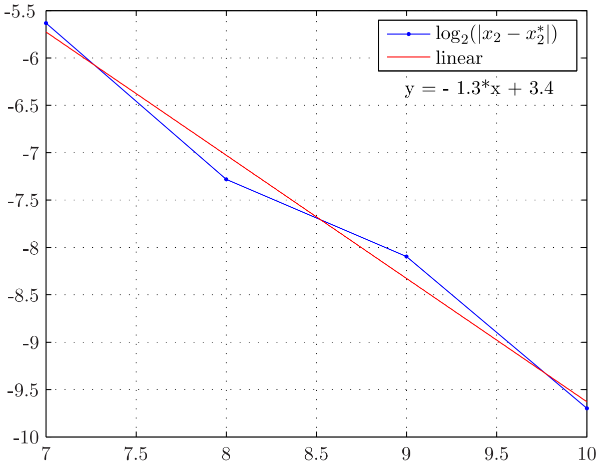


Figure 10. Interpolation accuracy of $x_2(y^*)$ vs. $\log_2(N_y)$. $\tau = 5$, QTT-Tucker.

Finally, we plot the average concentrations of both reacting proteins versus the concentration of the catalyst, see Figure 11. The quantity $\langle x_2 \rangle$ can be used also as a measure of the fraction of the cells ensemble being presented in the high (low) state. Indeed, it demonstrates the asymptotic: with y tending to ∞, the cells tend to occupy the high state. As a result,

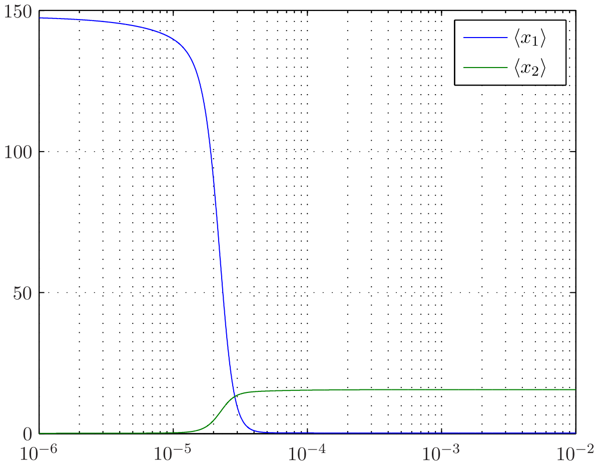


Figure 11. Average concentrations $\langle x_1 \rangle$, $\langle x_2 \rangle$ versus y

normalizing $\langle x_2 \rangle$ to its maximal value, obtain the fraction of high-state cells. It is the plot that was given in [54], Fig. 5(b), and we observe a good agreement with the experimental results.

6.3. λ-phage

The last example is the simulation of the life cycle of the bacteriophage-λ [13, 19]. The first paper [13] considers the sparse grids approach. The second one is more tensor related and uses the so-called *Dirac-Frenkel* principle for the dynamical low-rank approximation in the Tucker format (DLRA). However, the simulation was done for a small time interval ($T = 10$) only, which is far from providing the stationary solution. The reason is that the stationary concentration of the second specie is too high ($\sim 10^4$) to be efficiently treated by the algorithms proposed. In this article, we present an efficient computation of the stationary solution on very large grids with the use of the QTT format and the alternating linear solver.

The model parameters read:

- $d = 5$, $M = 10$:
- $w^{1+} = \dfrac{a_1 b_1}{b_1 + x_2}$, $z = -e_1$: generation of S_1; $a_1 = 0.5$, $b_1 = 0.12$.
- $w^{1-} = c_1 \cdot x_1$, $z = e_1$: destruction of S_1; $c_1 = 0.0025$.
- $w^{2+} = \dfrac{(a_2 + x_5) b_2}{b_2 + x_1}$, $z = -e_2$: generation of S_2; $a_2 = 1$, $b_2 = 0.6$.
- $w^{2-} = c_2 \cdot x_2$, $z = e_2$: destruction of S_2; $c_2 = 0.0007$.
- $w^{3+} = \dfrac{a_3 b_3 x_2}{b_3 \cdot x_2 + 1}$, $z = -e_3$: generation of S_3; $a_3 = 0.15$, $b_3 = 1$.
- $w^{3-} = c_3 \cdot x_3$, $z = e_3$: destruction of S_3; $c_3 = 0.0231$.
- $w^{4+} = \dfrac{a_4 b_4 x_3}{b_4 \cdot x_3 + 1}$, $z = -e_4$: generation of S_4; $a_4 = 0.3$, $b_4 = 1$.

- $w^{4-} = c_4 \cdot x_4$, $z = e_4$: destruction of S_4; $\quad c_4 = 0.01$.
- $w^{5+} = \dfrac{a_5 b_5 x_3}{b_5 \cdot x_3 + 1}$, $z = -e_5$: generation of S_5; $\quad a_5 = 0.3$, $b_5 = 1$.
- $w^{5-} = c_5 \cdot x_5$, $z = e_5$: destruction of S_5; $\quad c_5 = 0.01$.

The matrix assembly is done by summing the rank-2 Laplace-like TT decomposition of the destruction part and 5 creation parts, obtaining the total TT rank bound 7.

First of all, we give a comparison with the dynamical Tucker approximation algorithm from [19]. That paper reports that the grid of sizes $15 \times 40 \times 10 \times 10 \times 10$ was used; however, to employ the QTT format, we restrict ourselves to the powers of two: $16 \times 64 \times 16 \times 16 \times 16$. The time interval is $T = 10$, and since this is a transient process, accurate time integration is necessary. To achieve that, we solve the coupled space-time system as in the first example, Section 6.1, for different inner time intervals T_0 and time grid sizes N_t. As the initial state, the ($n = 3$, $p = [0.05, ..., 0.05]$)-multinomial distribution was chosen:

$$P(x, 0) = \frac{3!}{x_1! \cdots x_5! \cdot (3 - |x|)!} 0.05^{|x|} (1 - 5 \cdot 0.05)^{3-|x|} \cdot \theta(3 - |x|),$$

where $|x| = x_1 + \cdots + x_5$, and $\theta(\xi)$ is the Heaviside function. This function can be constructed straightforwardly as a full-format $4 \times 4 \times 4 \times 4 \times 4$-tensor thanks to the zeroing Heaviside function if any of x_i is greater than 3. After that, the TT decomposition (with ranks 4) is computed, and each block is expanded by zeros to the appropriate grid size. In the end, the TT representation is reapproximated into the QTT one.

The timings of the linear QTT solution are presented in Table 1. With such small QTT ranks (~ 30), the CPU times are almost independent on the time grid size, and even tend to decrease, since the finer discretization provides more accurate and smooth solution. The same situation may be observed with respect to T_0 as well. In all cases the computation is much faster than 5 minutes of the DLRA, and a fortiori than $\sim$ 3 hours of the SSA, reported in [19].

Table 1. CPU times (sec.) versus T_0 and N_t

$T_0 \setminus N_t$	256	512	1024	2048
1	20.66	23.89	21.18	20.38
2	18.49	19.52	19.44	17.70
5	16.33	16.35	17.55	16.01
10	25.34	15.51	12.49	11.23

To check the accuracy, we compare the marginal probability densities with those are computed in the full format on the grid $16 \times 64 \times 10 \times 10 \times 10$. The corresponding Crank-Nicolson propagation matrices of size 1024000 were assembled in the MATLAB `sparse` format, with the use of `gmres` as the iterative solver. The integration was conducted with the time step 0.01, which required 1071 seconds of CPU time. The marginal distributions are shown in Figures 12-16, and the 2-norm errors are presented in Table 2. One should note, that the time step 0.01 used in the full-format simulations, as well as the FSP truncation at lower grid sizes yield the error in the full solution $\mathcal{O}(1e-4)$. That is, the QTT solution with $N_t = 256$, $T_0 = 2$ (the time step ~ 0.01) appears to be closer to the full-format one, than with the finer discretization. However, in all cases the accuracy $\mathcal{O}(1e-4)$ is achieved.

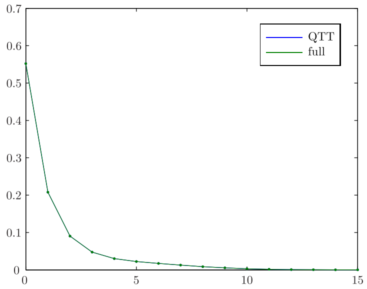


Figure 12. $P_1(x_1)$

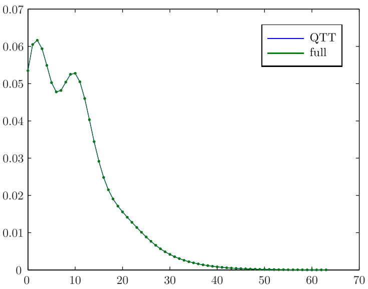


Figure 13. $P_2(x_2)$

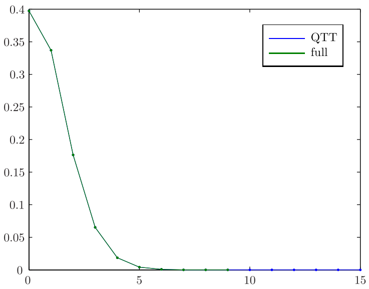


Figure 14. $P_3(x_3)$

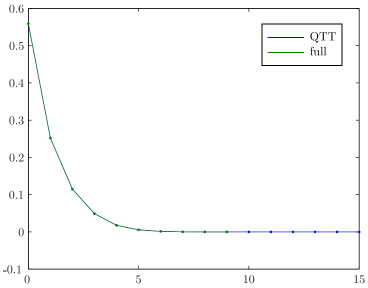


Figure 15. $P_4(x_4)$

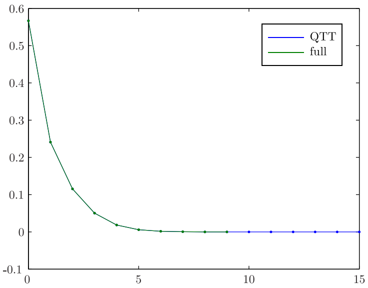


Figure 16. $P_5(x_5)$

Table 2. $||P_{qtt} - P_{full}||/||P_{full}||$ versus T_0 and N_t

$T_0 \setminus N_t$	256	512	1024	2048
1	7.971e-5	7.096e-5	7.277e-5	8.543e-5
2	4.780e-5	6.700e-5	4.356e-5	7.643e-5
5	6.387e-5	8.733e-5	2.102e-4	2.769e-4
10	1.435e-4	2.296e-4	2.933e-4	3.963e-4

To integrate the system until the stationary solution is a much more difficult problem. First, we set the grid sizes to $128 \times 65536 \times 64 \times 64 \times 64$, in accordance to very high ($\sim 4 \cdot 10^4$) concentrations of the second specie, see Fig. 18. Second, the relaxation time is large, $T \sim 2 \cdot 10^4$. In our simulation, we use the exponential splitting of the time interval:

$$t_p = \exp(0.05 \cdot p), \quad p = 1, ..., 200.$$

To obtain an accurate time history, in each subinterval $[t_{p-1}, t_p]$ we solve the coupled space-time system with an additional splitting into 1024 time steps (encapsulated in the QTT format). The convergence history and the cumulative CPU times are shown in Figure 17. We see, that it takes about an hour of computational time to recover the whole time history with the stationarity accuracy $\sim 10^{-7}$. Despite the large grids, this is the case, when the QTT-Tucker does not outperform the linear QTT format due to large core ranks (i.e. the physical dimensions are strongly connected), and its larger (cubic) asymptotic leads to a larger time. In this example, the QTT-Tucker solver takes about 4000 seconds.

If we are not interested in the transient processes, we may use the $\mathtt{Euler} \circ \mathtt{AMR}$ iterations over the time points t_p. The total CPU time in this case is 669 seconds in the linear QTT format, or 413 seconds in the QTT-Tucker format, and the relative accuracies of the mean concentrations at the final time point w.r.t. the finer time splitting are the following:

S_1	S_2	S_3	S_4	S_5
4.6e-5	1.7e-6	6.2e-8	3.1e-7	1.7e-7.

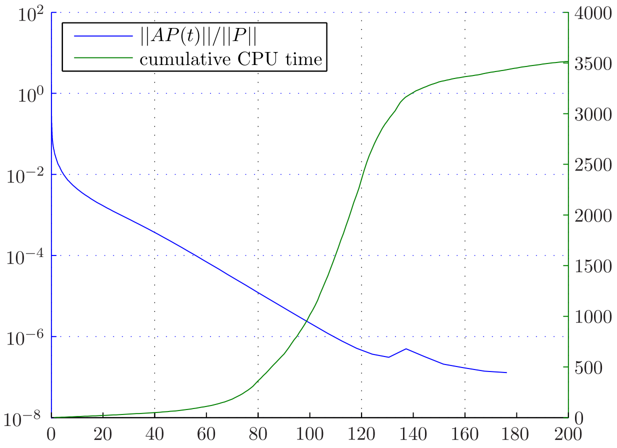


Figure 17. Closeness to the kernel $\frac{||AP||}{||P||}(t)$, and the cumulative CPU time (sec.)

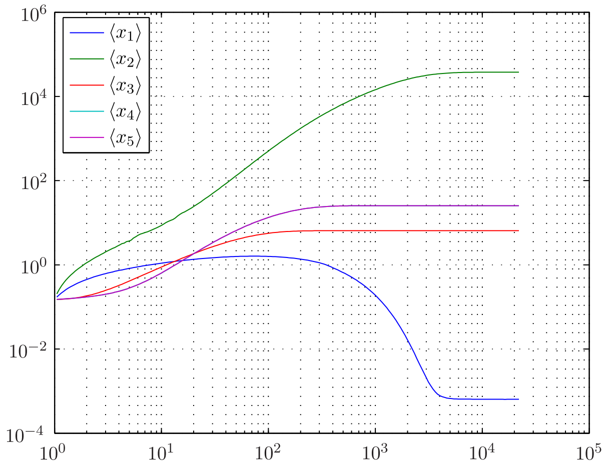


Figure 18. Average concentrations $\langle x_i \rangle$ vs. t

It shows the advantages of using the newer tensor methods even in such nontrivial problems.

7. Conclusions

We have investigated the tensor product structure approach to the Chemical Master Equation. The main contributions include:

- analysis of the operators (Hamiltonians) in the TT and QTT format,
- construction of the QTT- and QTT-Tucker-based computational algorithm for the multidimensional CME, and its complexity analysis,
- demonstration of the computational efficiency of the tensor-structured solution method to the block space-time discretized system.

The techniques have been applied to commonly used model biological systems governed by the CME, such as cascade gene networks, chemical switches, as well as more realistic ones, such as the λ-phage. The algorithms employed for the block space-time systems (16), (17), in particular, the AMR linear solver, manifest a significant reduction of the computational time and error with respect to the previous methods, such as the time stepping iterations on tensor product manifolds, and classical Monte-Carlo-like methods (SSA). Use of the virtual tensorisation (QTT format) allows to treat efficiently even the cases of very large space and time grids, which appear in high-concentration systems (Section 6.3). In addition, the simulation in presence of parametrically and uncertainly defined coefficients was considered, and the tensor methods were found to be a very promising tool.

References

[1] *Steuer R.* Effects of stochasticity in models of the cell cycle: from quantized cycle times to noise-induced oscillations // *Journal of theoretical biology*. 2004. V. 228, № 3. P. 293–301.

[2] *Arkin A., Ross J., McAdams H.* Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected Escherichia coli cells // *Genetics*. 1998. V. 149, № 4. P. 1633–1648.

[3] *van Kampen N. G.* Stochastic processes in physics and chemistry. — North Holland, Amsterdam, 1981.

[4] *Gillespie D.* A general method for numerically simulating the stochastic time evolution of coupled chemical reactions // *Journal of computational physics*. 1976. V. 22, № 4. P. 403–434.

[5] *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.

[6] *Hemberg M., Barahona M.* Perfect sampling of the master equation for gene regulatory networks // *Biophysical journal*. 2007. V. 93, № 2. P. 401–410.

[7] *Gillespie D.* Approximate accelerated stochastic simulation of chemically reacting systems // *The Journal of Chemical Physics*. 2001. V. 115. P. 1716.

[8] *Goutsias J.* Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems // *The Journal of chemical physics*. 2005. V. 122. P. 184102.

[9] *Hellander A., Lötstedt P.* Hybrid method for the chemical master equation // *Journal of Computational Physics*. 2007. V. 227, № 1. P. 100–122.

[10] *Gillespie D.* The chemical Langevin and Fokker-Planck equations for the reversible isomerization reaction // *The Journal of Physical Chemistry A*. 2002. V. 106, № 20. P. 5063–5071.

[11] *Munsky B., Khammash M.* The finite state projection algorithm for the solution of the chemical master equation // *The Journal of chemical physics*. 2006. V. 124. P. 044104.

[12] *Bellman R. E.* Dynamic programming. — Princeton University Press, 1957.

[13] *Hegland M., Burden C., Santoso L. et al.* A solver for the stochastic master equation applied to gene regulatory networks // *Journal of Computational and Applied Mathematics*. 2007. V. 205, № 2. P. 708 - 724.

[14] *Ammar A., Cueto E., Chinesta F.* Reduction of the chemical master equation for gene regulatory networks using proper generalized decompositions // *Int. J. Numer. Meth. Biomed. Engng*. 2011. V. 00. P. 1–15.

[15] *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.

[16] *Le Bris C., Leliévre T., Maday Y.* Results and Questions on a Nonlinear Approximation Approach for Solving High-dimensional Partial Differential Equations // *Constr. Approx.* 2009. V. 30. P. 621-651.

[17] *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.

[18] *Binev P., Cohen A., Dahmen W. et al.* Convergence rates for greedy algorithms in reduced basis methods // *SIAM J. Math. Anal.* 2011. V. 43, № 3. P. 1457-1472.

[19] *Jahnke T., Huisinga W.* A Dynamical Low-Rank Approach to the Chemical Master Equation // *Bulletin of Mathematical Biology.* 2008. V. 70. P. 2283-2302.

[20] *Koch O., Lubich C.* Dynamical low rank approximation // *SIAM J. Matrix Anal. Appl.* 2007. V. 29, № 2. P. 434-454.

[21] *Lubich C., Rohwedder T., Schneider R., Vandreycken B.* Dynamical approximation of hierarchical Tucker and tensor-train tensors: Tech. rep.: University of Tübingen, 2012.

[22] *Oseledets I. V., Khoromskij B. N., Schneider R.* Efficient time-stepping scheme for dynamics on TT-manifolds: Preprint 24: MPI MIS, 2012. `http://www.mis.mpg.de/preprints/2012/preprint2012_24.pdf`.

[23] *Dolgov S. V., Khoromskij B. N., Oseledets I. V.* Fast solution of multi-dimensional parabolic problems in the TT/QTT–format with initial application to the Fokker-Planck equation // *SIAM J. Sci. Comput.* 2012. accepted.

[24] *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.

[25] *Dolgov S. V., Savostyanov D. V.* Alternating minimal residual methods for the solution of high-dimensional linear systems in the tensor train format: in preparation: 2013.

[26] *Kolda T. G., Bader B. W.* Tensor decompositions and applications // *SIAM Review.* 2009. V. 51, № 3. P. 455–500.

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

[28] *Holtz S., Rohwedder T., Schneider R.* The alternating linear scheme for tensor optimization in the tensor train format // *SIAM J. Sci. Comput.* 2012. V. 34, № 2. P. A683-A713.

[29] *Dolgov S. V., Oseledets I. V.* Solution of linear systems and matrix inversion in the TT-format // *SIAM J. Sci. Comput.* 2012. V. 34, № 5. P. A2718-A2739.

[30] *Iman R. L., Helton J. C.* An Investigation of Uncertainty and Sensitivity Analysis Techniques for Computer Models // *Risk Analysis.* 1988. V. 8, № 1. P. 71–90.

[31] *Schwab C., Todor R. A.* Sparse finite elements for elliptic problems with stochastic loading // *Numerische Mathematik.* 2003. V. 95, № 4. P. 707–734.

[32] *Östlund S., Rommer S.* Thermodynamic limit of Density Matrix Renormalization // *Phys. Rev. Lett.* 1995. V. 75, № 19. P. 3537–3540.

[33] *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.

[34] *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.

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

[36] *Holtz S., Rohwedder T., Schneider R.* On manifolds of tensors of fixed TT–rank // *Numer. Math.* 2012. V. 120, № 4. P. 701-731.

[37] *Dolgov S. V., Khoromskij B. N., Savostyanov D. V.* Superfast Fourier transform using QTT approximation // *J. Fourier Anal. Appl.* 2012. V. 18, № 5. P. 519-953.

[38] *Dolgov S. V., Khoromskij B. N.* Two-level Tucker-TT-QTT format for optimized tensor calculus: Preprint 19: MPI MIS, 2012. `http://www.mis.mpg.de/preprints/2012/preprint2012_19.pdf`.

[39] *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.

[40] *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`.

[41] *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.

[42] *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.

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

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

[45] *Fannes M., Nachtergaele B., Werner R. F.* Ground states of VBS models on cayley trees // *Journal of Statistical Physics.* 1992. V. 66. P. 939-973.

[46] *Oseledets I. V.* Approximation of $2^{\mathrm{d}} \times 2^{\mathrm{d}}$ matrices using tensor decomposition // *SIAM J. Matrix Anal. Appl.* 2010. V. 31, № 4. P. 2130-2145.

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

[48] *Kazeev V. A., Khoromskij B. N.* Low-rank explicit QTT representation of the Laplace operator and its inverse // *SIAM J. Matrix Anal. Appl.* 2012. V. 33, № 3. P. 742-758.

[49] *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`.

[50] *Ptashne M.* A genetic switch: λ-phage and higher organisms. — Wiley-Blackwell, 1992.

[51] *Huckle T., Waldherr K., Schulte-Herbrüggen T.* Computations in quantum tensor networks // *Linear Algebra Appl.* 2012.

[52] *Huckle T., Waldherr K., Schulte-Herbrüggen T.* Exploiting Matrix Symmetries and Physical Symmetries in Matrix Product States and Tensor Trains: Preprint: TU München, 2012.

[53] *Rohwedder T., Uschmajew A.* Local convergence of alternating schemes for optimization of convex problems in the TT format: Preprint 112: DFG-SPP1324, 2011.

[54] *Gardner T., Cantor C., Collins J.* Construction of a genetic toggle switch in Escherichia coli // *Nature*. 2000. V. 403. P. 339–342.