OPEN ACCESS Freely available online

PLOS | COMPUTATIONAL BIOLOGY

Direct Solution of the Chemical Master Equation Using Quantized Tensor Trains

Vladimir Kazeev[1], Mustafa Khammash[2]*, Michael Nip[3], Christoph Schwab[4]

1 Seminar für Angewandte Mathematik, ETH Zürich, Zürich, Switzerland, **2** Department of Biosystems Science and Engineering, ETH Zürich, Basel, Switzerland, **3** Department of Mechanical Engineering, UC Santa Barbara, Santa Barbara, California, United States of America, **4** Seminar für Angewandte Mathematik, ETH Zürich, Zürich, Switzerland

Abstract

The Chemical Master Equation (CME) is a cornerstone of stochastic analysis and simulation of models of biochemical reaction networks. Yet direct solutions of the CME have remained elusive. Although several approaches overcome the infinite dimensional nature of the CME through projections or other means, a common feature of proposed approaches is their susceptibility to the curse of dimensionality, i.e. the exponential growth in memory and computational requirements in the number of problem dimensions. We present a novel approach that has the potential to "lift" this curse of dimensionality. The approach is based on the use of the recently proposed Quantized Tensor Train (QTT) formatted numerical linear algebra for the low parametric, numerical representation of tensors. The QTT decomposition admits both, algorithms for basic tensor arithmetics with complexity scaling linearly in the dimension (number of species) and sub-linearly in the mode size (maximum copy number), and a numerical tensor rounding procedure which is stable and quasi-optimal. We show how the CME can be represented in QTT format, then use the exponentially-converging *hp*-discontinuous Galerkin discretization in time to reduce the CME evolution problem to a set of QTT-structured linear equations to be solved at each time step using an algorithm based on Density Matrix Renormalization Group (DMRG) methods from quantum chemistry. Our method automatically adapts the "basis" of the solution at every time step guaranteeing that it is large enough to capture the dynamics of interest but no larger than necessary, as this would increase the computational complexity. Our approach is demonstrated by applying it to three different examples from systems biology: independent birth-death process, an example of enzymatic futile cycle, and a stochastic switch model. The numerical results on these examples demonstrate that the proposed QTT method achieves dramatic speedups and several orders of magnitude storage savings over direct approaches.

Citation: Kazeev V, Khammash M, Nip M, Schwab C (2014) Direct Solution of the Chemical Master Equation Using Quantized Tensor Trains. PLoS Comput Biol 10(3): e1003359. doi:10.1371/journal.pcbi.1003359

Editor: Daniel A. Beard, University of Michigan, United States of America

Received February 5, 2013; **Accepted** October 9, 2013; **Published** March 13, 2014

Funding: The research was partially supported by the European Research Council (ERC) FP7 programme project AdG247277 (erc.europa.eu), and the US National Science Foundation Grant ECCS-0835847 (www.nsf.gov) and the Human Frontier Science Program Grant RGP0061/2011 (www.hfsp.org). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

* E-mail: mustafa.khammash@bsse.ethz.ch

This is a *PLOS Computational Biology* Methods article.

Introduction

In spite of the success of continuous-variable deterministic models in describing many biological phenomena, discrete stochastic models are often necessary to describe biological phenomena inside living cells where random motion of reacting species introduces randomness in both the order and timing of biochemical reactions. Such random effects become more pronounced when one factors in the discrete nature of reactants and the fact that they are often found in low copy numbers inside the cell. Manifestations of randomness vary from copy-number fluctuations among genetically identical cells [1] to dramatically different cell fate decisions [2] leading to phenotypic differentiation within a clonal population. Characterizing and quantifying the effect of stochasticity and its role in the function of cells is a central problem in molecular systems biology.

In order to effectively capture this experimentally observed stochasticity, the evolution of the chemical species of interest are commonly modeled using jump Markov processes. Here, each state of the process corresponds to the copy number of one of the constituent species [3]. Within this framework, the evolution of the probability density over the possible configurations of the reaction network is described by a Forward Kolmogorov Equation, frequently referred to as the Chemical Master Equation (CME) within the chemical literature. While analytical solutions can be obtained under specific assumptions about the structure of the chemical network [4], these assumptions prove so restrictive as to exclude the vast majority of biologically relevant systems. In most cases, the CME cannot be solved explicitly and various numerical simulation techniques have been proposed to approximately solve the time-evolution problem.

A first class of methods seeks to compute approximations of the CME solution instead by solving a truncated version of the original Markov process. These methods are advantageous in that they provide explicit error guarantees after simulation. This class includes the finite state projection [5] and sliding window

Author Summary

Stochastic models of chemical networks are necessary to quantitatively describe random fluctuations and other probabilistic phenomena within living cells. The Chemical Master Equation (CME) describes the time evolution of molecular abundance probabilities in these models, and is a basis for many stochastic simulation and analysis methods. Yet the CME is difficult to solve directly except for very simple structures. Indeed current approaches are susceptible to the curse of dimensionality, that is, the exponential growth of memory and computational requirements in the number of problem dimensions. In this paper, we propose a novel approach that has the potential to overcome the curse of dimensionality. It is based on the use of the recently proposed Quantized Tensor Train (QTT) formatted numerical linear algebra for numerical representation of tensors, using algorithms for basic tensor arithmetics with complexity scaling linearly in the number of reacting species considered, and sub-linearly in the maximum allowed copy number per species. We present this approach and demonstrate its effectiveness by applying it to three problems from systems biology. Numerical experiments are reported which show that several orders of magnitude memory savings is typically afforded by the new approach presented here.

abstraction [6]. In these methods, the truncation is chosen so that both the number of states retained is small enough that it may be computed efficiently but large enough that it retains the majority of the probability mass over the time evolution. Clearly, these two objectives are not complementary. In order to guarantee that the approximation has low error, most biologically relevant reaction networks require truncations with so many states that they are completely intractable on available hardware. The finite buffer method [7,8] suggests a more sophisticated truncation to the states reachable from a given initial state assuming that only a prespecified finite number of molecules may be spontaneously created. However, its use is limited to explicit time-stepping schemes, in addition to requiring that the finite buffers be large to compute accurate solutions.

A second broad class of methods are the kinetic Monte Carlo approaches which instead seek to produce either exact or approximate realizations of the underlying Markov process [3,9,10]. By generating sufficiently many realizations, these methods obtain statistics for events that are biologically important. Unfortunately, in many systems, these important events occur rarely, so that producing enough realizations to estimate these statistics is prohibitive.

A third class of methods use asymptotic approximations to trade accuracy for computational or analytical tractability. This class includes the Moment Closure methods [11,12], the Linear Noise Approximation (LNA) [13], and Chemical Langevin Equation (CLE) treatments [14,15]. Each of these methods replaces the discrete description of the population counts with a continuous one and can therefore perform poorly in situations where the discrete dynamics are difficult to capture with continuum models, e.g. when even one of the reacting species exhibits low population count or is constrained to have low population count, for instance, in the presence of conservation laws.

Some of the classes of methods described so far perform well in complementary regimes and recently there has been substantial effort to combine these methods resulting in the so-called hybrid methods. Several methods require a time-scale separation of the dynamics to split the system into fast and slow species and impose a quasi-stationary assumption for the fast reactions. An approximate method which can converge quickly to an accurate approximation of a stationary distribution such as τ-leaping [16] or the Chemical Langevin Equation [17,18] is used for the fast species, while the slower but more accurate Gillespie algorithm is used for the slow species. Rather than partitioning the species by time-scales of the associated reactions, other methods separate by average molecule count. The low count species are tracked by kinetic Monte Carlo while an ODE approximation is made for the dynamics of the high count species [19,20]. While these methods allow faster simulations, speedups come at the cost of accuracy, as modeling errors are introduced by the partial replacement of the CME with cruder descriptions.

In order to provide methods that are both accurate and computationally efficient, several numerical techniques for compressing the dynamics and the solution have been explored in the recent literature. Attempts were made to expand the probability distribution as a linear combination of a small set of so-called "principal", orthogonal basis functions [21–25]. Then, either a Galerkin projection was used to map the dynamics onto the lower dimensional subspace spanned by the basis functions (Method of Lines) or first a time discretization was used and then the basis at each time step was adapted by either adding or subtracting basis elements (Rothe's Method). These methods differ primarily in their choice of orthogonal basis. A common feature of these approaches is that they begin with a basis for probability distributions of a single variable and then use the corresponding tensor product basis for multivariate distributions. This means that they are susceptible to the so-called *curse of dimensionality* [26], that is, the memory requirements and computational complexity of basic arithmetics grow exponentially in the number of dimensions. In the context of the CME, this means that all of these approaches can exhibit an exponential scaling of the complexity with the number of chemical species in the model.

Recent papers have attempted to address the curse of dimensionality by using a low-parametric representation of tensors known as *canonical polyadic* decomposition or *CANDECOMP/PARAFAC*, both notions being subsumed under the acronym *CP* [27,28]. CP is a methodology for generalizing the singular value decomposition (SVD) for matrices to tensors of dimension greater than two by representing the solution as sums of rank-one tensors (equivalently, linear combinations of distributions in which species counts are independent at each fixed time point). As long as the tensor rank of the solution to be approximated remains low, these approaches can be very computationally efficient as basic arithmetics for tensors in the CP format scales linearly in the number of tensor dimensions.

A key challenge in applying the CP decomposition to construct approximate CME solvers is to control the tensor rank of the computed solution. Basic algebraic tensor operations such as addition and matrix-vector multiplication generally increase rank and hence computational cost. In [29] it is suggested to recompute a lower rank CP decomposition after *every* arithmetic operation. This approach turned out to be problematic in practice. One reason is that the problem of tensor approximation (in the Frobenius norm) with a tensor of fixed rank is, in general, ill-posed [30]. Thus, the numerical algorithms for computing an approximate representation may easily fail. Another reason is that the problem is NP-hard [31,32] and there is no robust algorithm having any affordable complexity.

Another approach [33], related to the present work, attempts to avoid the problem of approximation in the CP format entirely by projecting the dynamics onto a manifold composed of all tensors with a CP decomposition of some predetermined maximal tensor

rank. This procedure results in a set of coupled nonlinear differential equations which are then solved using available ODE solvers. While this effectively controls the tensor rank of the approximate solution, still, to the authors' knowledge, there is no way to estimate either theoretically (*a priori*) or numerically (*a posteriori*) the CP rank of the full CME solution as a function of given data.

In this paper we propose a new, deterministic computational methodology for the direct numerical solution of the CME, without modelling or asymptotic simplifications. The approach has complexity that scales favorably in terms of the number of different species considered and the maximum allowable copy number of each of these species. It is based on the recently proposed *Quantized Tensor Train* (QTT) formatted, numerical tensor algebra [34–37] which operates on low-parametric, numerical representation of tensors, rather than on their CP representations. This decomposition admits both algorithms for basic tensor arithmetics that scale linearly in the dimension (the species number) and a robust adaptive numerical procedure for the tensor truncation, which is quasi-optimal in the Frobenius norm.

We show in the present paper how the CME can be represented in QTT format, then use *hp*-discontinuous Galerkin discretization in time to exploit the time-analyticity of the CME evolution and to reduce the CME evolution problem to a set of QTT structured linear equations that are solved at each time step [38]. We then exploit an algorithm available for solving linear systems in this format that is based on Density Matrix Renormalization Group (DMRG) methods from quantum chemistry.

The numerical experiments reported below (see, in particular, Table 1) show several orders of magnitude memory savings, which is typically afforded by the new approach presented here.

Results/Discussion

We start our development by formulating the Chemical Master Equation (CME), arising from stochastically reacting chemical species. Then we will devote the remainder of the article for its proposed solution. A "well-stirred" solution of d chemically reacting molecules in thermal equilibrium can be described by a jump Markov process, where for each fixed time $t \geq 0$, $X(t) \in \mathbb{Z}^d_{\geq 0}$ is a random vector of nonnegative integers with each component representing the number of molecules of one chemical species present in the system. In [29] and the references therein, it is shown that, given an initial condition $X(0) \in \mathbb{Z}^d_{\geq 0}$, the corresponding probability density function (PDF) $\mathbb{Z}^d_{\geq 0} \times [0,\infty) \ni (\underline{x},t) \mapsto \mathbf{p}_{\underline{x}}(t)$ of the process solves the Chemical Master Equation (CME):

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{p}_{\underline{x}}(t) = -\mathbf{p}_{\underline{x}}(t)\sum_{s=1}^{R}\omega^s(\underline{x}) + \sum_{s=1}^{R}\mathbf{p}_{\underline{x}-\underline{\eta}^s}(t)\omega^s\left(\underline{x}-\underline{\eta}^s\right) \quad (1)$$

where R is the number of reactions in the system, $\underline{\eta}^s \in \mathbb{Z}^d$ and ω^s are the stoichiometric vector and propensity function of the sth reaction, respectively. The CME is a system of coupled linear ordinary differential equations with one equation per state $X(t) = \underline{x} \in \mathbb{Z}^d_{\geq 0}$.

Our Approach to Solving the CME

We briefly outline our proposed methodology for the numerical solution of the CME. Since the state space of solutions is countably infinite, the main challenge to be overcome is the curse of dimensionality. As the state space of the CME is typically countably infinite, there is a countably infinite number of different possible states that could be reached by the chemical system. Our approach consists of employing efficient methods for tensor-structured, rank-adaptive numerical solution of very large but "finite state projection" truncations of the CME. In a nutshell, we are proposing to solve large, coupled systems of linear ODEs with a special, tensor structure inherited from the CME. We now give a

Table 1. Overview of the QTT compression of the storage needed for solutions (maximum throughout the time stepping) and CME operators.

	Direct Approach		Proposed Approach					
run	solution	operator	solution		truncated solution		operator	
	Mem	Mem	Mem	ratio	Mem	ratio	Mem	ratio
d independent birth-death processes								
$d=1$	4.10_{3}	1.68_{7}	736	1.80_{-1}	264	6.45_{-2}	992	5.91_{-5}
$d=2$	1.68_{7}	2.82_{14}	3858	2.30_{-4}	528	3.15_{-5}	2852	1.01_{-11}
$d=3$	6.87_{10}	4.72_{21}	7742	1.13_{-7}	898	1.31_{-8}	4800	1.02_{-18}
$d=4$	2.81_{14}	7.90_{28}	12176	4.33_{-11}	1432	5.09_{-12}	6748	8.52_{-26}
$d=5$	1.15_{18}	1.32_{36}	16262	1.41_{-14}	1946	1.69_{-15}	11032	8.30_{-33}
genetic toggle switch								
only	3.36_{7}	1.12_{15}	65264	1.95_{-3}	–	–	10988	9.76_{-12}
enzymatic futile cycle								
(A)	4.19_{6}	1.76_{13}	18396	4.39_{-3}	8472	2.02_{-3}	25848	1.47_{-9}
(D)	4.19_{6}	1.76_{13}	360332	8.59_{-2}	290144	6.92_{-2}	5584	3.17_{-10}

For details on "truncated solution" see **Numerical experiments. Common details**. Solution Mem in the Direct Approach is the number of states taken into account in the FSP, which is equal to the number of entries, N, in the solution vector. For the CME operator, Mem is N^2, the number of entries in the matrix. In the Proposed QTT Approach, *ratio* indicates the memory storage compression ratio, i.e. the ratio of Mem in the Proposed QTT Approach to that in the Direct Approach. In the sparse representation of the CME operator the number of nonzero entries would be $\mathcal{O}(N)$ rather than N^2. The exponents are given in boldface for the base 10.

doi:10.1371/journal.pcbi.1003359.t001

general outline of our approach, followed by detailed descriptions of each of these steps.

1. Truncate the CME to obtain a linear ODE with a finite state space. The CME describes the dynamics of probabilities of finding the chemical system in different states. In general the number of these different states is countably infinite, as it is not unknown *a priori* the maximum number of copies that each species can take. While this gives rise to an infinite number of state variables, each indicating the probability of a given chemical state, the vast majority of these probabilities are vanishingly small. This has motivated approaches for truncating the infinite number of state variables in the CME in a way that results in a finite number of state variables corresponding to chemical states that are likely to have high probability mass. The truncated CME consists of a system of linear ODEs with finite state space, that can *in principle* be solved. One such truncation approach which we will follow here is the Finite State Projection method. This truncation approach has the advantage of yielding bounds on the error between the solution of the truncated finite system and the original infinite set of ODEs (the CME). The Finite State Projection has been reported elsewhere [5], but we give a brief description of the approach in this article for completeness.

2. Express the truncated CME using tensors; Employ numerical rank reduction and compression to save storage and to speed up algebraic operations. In conventional approaches to solving the CME, the state-space is enumerated by a "long" index and the corresponding probabilities are stacked into a vector $\hat{\mathbf{p}}(t)$ that is then multiplied by the CME operator to form the right-hand side of the ODE: $\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathbf{p}}(t)=\mathbf{A}\hat{\mathbf{p}}(t)$. At all times t the solution is an array indexed by a multi-index, e.g., $\underline{x}$, which is a d-tuple of indices $x_k\in\{0,1,2,\ldots,n_k-1\}$, where k ranges from 1 to d. We shall also refer to $\mathbf{p}_{\underline{x}}(t)$ as a d-dimensional $n_1\times\ldots\times n_d$-vector. Our approach is based on exploiting the high-dimensional structure of the vectors and matrices involved, related to the separation of the indices, instead of stacking all indices into a single "long" index.

Linear operators acting on these d-dimensional $n_1\times\ldots\times n_d$-vectors can themselves be expressed using tensor notation. In our case, the action of the CME operator A is one such operator. A key aspect of our approach is that both the tensor vector $\mathbf{p}_{\underline{x}}$ and the tensor operator $\mathbf{A}$ arising from CME problems admit a dramatic level of compression. This tensor compression is achieved through the so called tensor train representation (TT). Tensor train compression goes beyond exploiting the sparsity structure, and actually exploits the rank structure of the tensor. This reduced rank compression is at the heart of our approach to the CME solution. The compression itself is analogous to the compression of the low-rank representation of usual matrices. Indeed, an $n\times n$ matrix of rank $r\ll n$ can be stored using $2rn$ variables, while the approximation can be based, e.g., on the Singular Value Decomposition (SVD). In a similar fashion, the TT format is a generalization of this compression to multidimensional tensors. This is both true for tensors and linear operators acting on these. Further *adaptive data reduction and compression* is afforded by the so-called quantized tensor train (QTT) format. Both the TT and QTT formats will be discussed later in this article, along with simple examples demonstrating the compression that can be achieved by using these formats.

3. Employ *hp*-discontinuous Galerkin discretization in time to solve the truncated ODE. Once the problem has been represented in QTT format, we use *hp*-discontinous Galerkin (*hp*-DG) discretization in time as the time-stepping scheme [38] to solve the truncated ODE. Given a time mesh, the *hp*-DG method finds an approximate solution to the initial value problem that is a polynomial when restricted to each subinterval of the time mesh and possibly discontinuous at each mesh point. This method allows adaptation of the size of each time step (*h*-adaptation), allowing good resolution of fast transients, as well as the order of the approximating polynomial on each time step (*p*-adaptation), or both simultaneously (*hp*-adaptation). For linear ODE initial value problems like the projected CME, the *hp*-DG approach achieves *exponential rates of convergence* to the classical solution with respect to the number of temporal degrees of freedom. Practically, *hp*-DG discretization in time reduces the projected CME evolution problem to a sequence of systems of QTT structured linear equations that must be solved at each time step. Our computational method then exploits an algorithm available for solving linear systems in this format that is based on Density Matrix Renormalization Group (DMRG) methods from quantum chemistry.

CME Truncation: Separability and Finite State Projection of the CME Operator

Munsky and Khammash [5] rewrote the right-hand side of the CME (1) as the action of a linear operator $\mathbf{A}$ on the probability density at the current time:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{p}(t)=\mathbf{A}\mathbf{p}(t) \tag{2}$$

Throughout this paper, we refer to $\mathbf{A}$ as the *CME operator*.

Hegland and Garcke introduced an explicit representation of the CME operator as sums and compositions of a few elementary linear operators [29]: let $\mathbf{S}_{\underline{\eta}}$ be the spatial shift of a probability density by a vector $\underline{\eta}\in\mathbb{Z}^d$ and let $\mathbf{M}_\omega$ be multiplication by a real-valued function ω:

$$\left(\mathbf{S}_{\underline{\eta}}\,\mathbf{p}\right)_{\underline{x}}=\mathbf{p}_{\underline{x}-\underline{\eta}},\quad \left(\mathbf{M}_\omega\,\mathbf{p}\right)_{\underline{x}}=\omega(\underline{x})\cdot\mathbf{p}_{\underline{x}}.$$

Then the CME operator can be written as follows, with $\mathbb{I}$ denoting the identity operator:

$$\mathbf{A}=\sum_{s=1}^{R}\left(\mathbf{S}_{\underline{\eta}^s}-\mathbb{I}\right)\circ\mathbf{M}_{\omega^s}. \tag{3}$$

To simplify the exposition, we assume that all propensity functions are *rank-one separable*, i.e. they are of the form

$$\omega^s(\underline{x})=\prod_{k=1}^{d}\omega_k^s(x_k),\quad \underline{x}\in\mathbb{Z}^d_{\geq 0}, \tag{4}$$

for $1\leq s\leq R$, where each $\omega_k^s(x_k)$ is a nonnegative function in the single variable x_k. Considering rank-one separable propensity functions is sufficient for all elementary reactions which occur as building blocks in more complicated reaction kinetics.

The CME (2) is posed on the (countably) infinite space $\mathbb{Z}^d_{\geq 0}$ of states. In this form, the CME (1) is an infinite-dimensional coupled evolution problem which necessitates truncation prior to numerical discretization. In the case of a particular class of monomolecular reactions, Jahnke and Huisinga were able to construct an explicit solution in terms of convolutions of products of Poisson and multinomial distributions [4].

In order to be able to address more complex systems computationally, Munsky and Khammash proposed the Finite State Projection Algorithm (FSP) [5] which seeks to truncate the

countably infinite dimensional space $\mathbb{Z}^d_{\geq 0}$ of states of the process to its finite subset

$$\Omega^{\underline{n}} = \left\{ \underline{x} \in \mathbb{Z}^d_{\geq 0} : 0 \leq x_k < n_k \quad \text{for} \quad 1 \leq k \leq d \right\} \subset \mathbb{Z}^d_{\geq 0}, \quad (5)$$

associated with a multi-index $\underline{n} = (n_1, n_2, ..., n_d) \in \mathbb{N}^d$, so that the dynamics over $\Omega^{\underline{n}}$ are close to those of the original system; see Theorem 1. In practice, the truncation satisfying a given error tolerance may still require a very large number of states: the dimension of the FSP vector $\hat{\mathbf{p}}$ equals $\text{card}(\Omega^{\underline{n}}) = \Pi_{k=1}^d n_k$ rendering a direct numerical solution of even the projected equation (S1.1) infeasible in many cases. The remainder of the paper presents a novel approach for the numerical solution of such FSP truncated systems that retain large numbers of states. For notational convenience, we drop the superscripts $\underline{n}$ and the hat from $\hat{\mathbf{p}}$ indicating the FSP since we will only consider systems which have already been truncated. Similarly, we now use the shift and multiplication operators in (3) restricted to the truncated state space without change of notation.

Assuming that a FSP has been performed, we henceforth treat $\mathbf{p}_{\underline{x}}(t)$ as a d-dimensional $n_1 \times \ldots \times n_d$-vector, i.e. as an array indexed by $\Omega^{\underline{n}}$ which we identify with *ordered* d-tuples of indices $i_k \in \{0,1,2,\ldots,n_k-1\}$, where k ranges from 1 to d. Each dimension k (alternatively referred to as a *mode* or *level*) has a corresponding *mode size* n_k, that is, the number of values which the index for that dimension can take. For our chemically reacting system, $n_k - 1$ corresponds to the maximum number of copies of the kth species that is considered. For a more detailed introduction to basic tensor operations and terminology see, for example, [39,40].

For the same ordering of $\underline{x}$, consider the corresponding d-dimensional $n_1 \times \ldots \times n_d$-vectors ω^s, $1 \leq s \leq R$, containing the values of the propensities on $\Omega^{\underline{n}}$ to which we shall refer as *propensity vectors*:

$$\omega^s_{\underline{x}} = \omega^s(\underline{x}) \quad \text{for all} \quad \underline{x} \in \Omega^{\underline{n}}. \quad (6)$$

Within the projected CME (S1.1), the operators corresponding to weighting by the propensity functions, involved in (3), are finite matrices: $\mathbf{M}_{\omega^s} = \text{diag}\omega^s$. Then, under the rank one separability assumption (4), with $\left(\omega^s_k\right)_{x_k} = \omega^s_k(x_k)$ for $0 \leq x_k \leq n_k$, $1 \leq k \leq d$ there holds

$$\omega^s = \omega^s_1 \otimes \ldots \otimes \omega^s_d, \qquad 1 \leq s \leq R. \quad (7)$$

Tensor Representation of the CME: TT and QTT Formats

Tensor Train representation of vectors and matrices. Our approach to the direct numerical solution of the CME (2) is based on the structured, low-parametric representation of all vectors and matrices involved in the solution in the *Tensor Train* (TT) format [34,41] developed by Oseledets and Tyrtyshnikov. To present it, let us consider a d-dimensional $n_1 \times \ldots \times n_d$-vector $\mathbf{p}$ and assume that two- and three-dimensional arrays $U_1, U_2, \ldots, U_d$ satisfy

$$\mathbf{p}_{j_1,\ldots,j_d} = \sum_{\alpha_1=1}^{r_1} \cdots \sum_{\alpha_{d-1}=1}^{r_{d-1}} U_1(j_1,\alpha_1) \cdot U_2(\alpha_1,j_2,\alpha_2) \cdot \ldots \cdot U_{d-1}(\alpha_{d-2},j_{d-1},\alpha_{d-1}) \cdot U_d(\alpha_{d-1},j_d) \quad (8)$$

for $0 \leq j_k \leq n_k - 1$, where $1 \leq k \leq d$. Then $\mathbf{p}$ is said to be represented in the TT decomposition in terms of the *core tensors* $U_1, U_2, \ldots, U_d$. The summation indices $\alpha_1, \ldots, \alpha_{d-1}$ and limits $r_1, \ldots, r_{d-1}$ on the right-hand side of (8) are called, respectively, *rank indices* and *ranks* of the decomposition. See Figure 1 for a schematic drawing.

The TT decomposition can potentially result in large compression of the tensor by exploiting the rank structure of the tensor. This is demonstrated in a simple example

Example 1 (TT Compression) *Consider a vector* $\mathbf{p}$ *of size* $n \times n \times n$ *given elementwise by*

$$\mathbf{p}_{j_1 j_2 j_3} = \sin(\alpha_1 j_1 + \alpha_2 j_2 + \alpha_3 j_3), \quad 0 \leq j_1, j_2, j_3 < n,$$

where $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}$. *By applying trigonometric identities, one obtains for all* $0 \leq j_1, j_2, j_3 < n$ *the following row-matrix-column factorization:*

$$\mathbf{p}_{j_1 j_2 j_3} = \begin{pmatrix} \sin \alpha_1 j_1 & \cos \alpha_1 j_1 \end{pmatrix} \begin{pmatrix} \cos \alpha_2 j_2 & -\sin \alpha_2 j_2 \\ \sin \alpha_2 j_2 & \cos \alpha_2 j_2 \end{pmatrix} \begin{pmatrix} \cos \alpha_3 j_3 \\ \sin \alpha_3 j_3 \end{pmatrix},$$

in which the indices are separated so that every factor depends on the corresponding index only. This factorization implies a TT representation of the form (8) with ranks $r_1 = r_2 = 2$ *and the cores given for* $0 \leq j_1, j_2, j_3 < n$ *by*

$$U_1(j_1,\cdot) = \begin{pmatrix} \sin \alpha_1 j_1 & \cos \alpha_1 j_1 \end{pmatrix},$$
$$U_2(\cdot,j_2,\cdot) = \begin{pmatrix} \cos \alpha_2 j_2 & -\sin \alpha_2 j_2 \\ \sin \alpha_2 j_2 & \cos \alpha_2 j_2 \end{pmatrix}, \quad U_3(\cdot,j_3) = \begin{pmatrix} \cos \alpha_3 j_3 \\ \sin \alpha_3 j_3 \end{pmatrix}$$

This TT decomposition involves $nr_1 + r_1 n r_2 + r_2 n = 8n$ *parameters instead of* n^3 *required for the elementwise representation. The case of* $d > 3$ *dimensions is considered in* [*42, Theorem 4*], *the number of parameters being under* $4dn$ *compared to* n^d.

Unlike CP, the TT format allows the construction of a decomposition, exact or *approximate*, through the low-rank representation of a sequence of single matrices; for example, by SVD. In particular, note that for every $k = 1, \ldots, d-1$ the decomposition (8) implies a rank-r_k representation of an *unfolding matrix* $\mathbf{U}^{(k)}$ which consists of the entries

$$\mathbf{U}^{(k)}_{\overline{j_1,\ldots,j_k};\,\overline{j_{k+1},\ldots,j_d}} = \mathbf{p}_{j_1,\ldots,j_k,j_{k+1},\ldots,j_d}.$$

Here, the overscore denotes the vectorization of multi-indices: $\overline{j_1, \ldots, j_k} = \sum_{m=1}^k j_m \prod_{\ell=m+1}^k n_\ell$ for $1 \leq k \leq d$. Conversely, if the vector $\mathbf{p}$ is such that the unfolding matrices $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(d-1)}$ are of ranks $r_1, \ldots, r_{d-1}$ respectively, then the cores $U_1, U_2, \ldots, U_d$, such that (8) holds, do exist; see Theorem 2.1 in [41]. The ranks of the unfolding matrices are the lowest possible ranks of a TT decomposition of the vector and, therefore, are called *TT ranks of the vector*.

What is more important is that the low-rank matrix structure of the unfolding matrices translates into the low-rank TT structure of the vector. Once the former can be approximated in the Frobenius norm with ranks $r_1, \ldots, r_{d-1}$ and accuracies $\varepsilon_1, \ldots, \varepsilon_{d-1}$, the latter can be approximated in the same norm in the TT format with ranks $r_1, \ldots, r_{d-1}$ and accuracy $\sqrt{\sum_{k=1}^{d-1} \varepsilon_k^2}$. The proof relies on the TT approximation algorithm. For details, we refer to Theorem 2.2 with the corollaries and to Algorithms 1 and 2 in [41]. This low rank approximation of the unfolding matrices can be considered

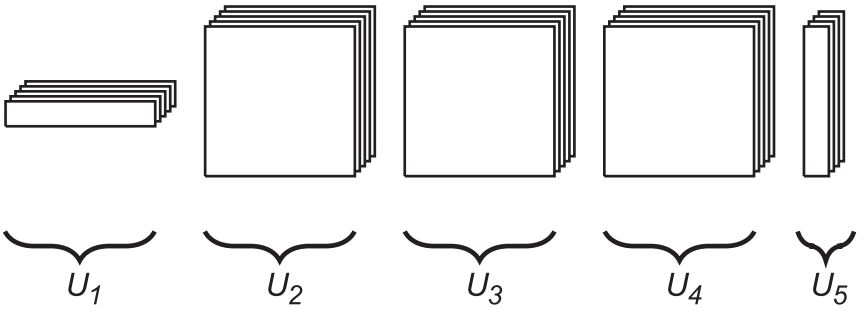

Figure 1. Schematic drawing of a TT decomposition of a five-dimensional array. Each TT core can be visualized as a stack of matrices with the size of the stack equal to the corresponding mode size. The number of TT cores is equal to the number of dimensions of the array. Element $\mathbf{u}(j_1,\ldots,j_5)$ of the full array is given by the (matrix) product of matrix j_1 selected from core U_1, matrix j_2 from core U_2, etc. Note that the size of each matrix within a core must be the same, but may differ between distinct cores. Note also that the number of matrices in each core depends on the corresponding mode size of the full tensor and generally differs between cores. Such an interpretation in the sense of a product of parametric matrices is widely used for the *Matrix Product States*, see [46–48].
doi:10.1371/journal.pcbi.1003359.g001

and is implemented as adaptive and compressive data representation at each stage of computation.

Example 2 (Unfolding of a tensor) *Consider a tensor* $\mathbf{p}$ *of size* $3\times2\times2$. *It has two unfolding matrices* $\mathbf{U}^{(1)}$ *and* $\mathbf{U}^{(2)}$ *given by*

$$\mathbf{U}^{(1)}=\begin{pmatrix}\mathbf{p}_{111} & \mathbf{p}_{121} & \mathbf{p}_{112} & \mathbf{p}_{122}\\ \mathbf{p}_{211} & \mathbf{p}_{221} & \mathbf{p}_{212} & \mathbf{p}_{222}\\ \mathbf{p}_{311} & \mathbf{p}_{321} & \mathbf{p}_{312} & \mathbf{p}_{322}\end{pmatrix}\quad\text{and}\quad\mathbf{U}^{(2)}=\begin{pmatrix}\mathbf{p}_{111} & \mathbf{p}_{112}\\ \mathbf{p}_{211} & \mathbf{p}_{212}\\ \mathbf{p}_{311} & \mathbf{p}_{312}\\ \mathbf{p}_{121} & \mathbf{p}_{122}\\ \mathbf{p}_{221} & \mathbf{p}_{222}\\ \mathbf{p}_{321} & \mathbf{p}_{322}\end{pmatrix}.$$

While $\mathbf{p}$, $\mathbf{U}^{(1)}$, *and* $\mathbf{U}^{(2)}$ *are structured differently, all have the same entries and represent the same data. The two TT ranks of* $\mathbf{p}$ *are exactly the (matrix) ranks of* $\mathbf{U}^{(1)}$ *and* $\mathbf{U}^{(2)}$.

Note also that, unlike CP, the TT representation relies on a certain ordering of the dimensions so that *reordering dimensions may affect the numerical values of TT ranks significantly*. We discuss this issue in more detail in the transposed QTT section.

The TT representation may be applied to multidimensional matrices in a similar way as to vectors. Consider a d-dimensional $(m_1\times\ldots\times m_d)\times(n_1\times\ldots\times n_d)$-matrix $\mathbf{A}$. Let us vectorize it and merge the corresponding row and column indices to obtain a d-dimensional $m_1\cdot n_1\times\ldots\times m_d\cdot n_d$-vector $\mathbf{a}$. Then the TT representation of the vector $\mathbf{a}$, given by the elementwise equality

$$\mathbf{A}_{i_1,\ldots,i_d;\,j_1,\ldots,j_d}=\mathbf{a}_{\overline{i_1j_1},\ldots,\overline{i_dj_d}}=\sum_{\alpha_1=1}^{r_1}\cdots\sum_{\alpha_{d-1}=1}^{r_{d-1}}V_1(i_1,j_1,\alpha_1)\cdot V_2(\alpha_1,i_2,j_2,\alpha_2)\cdot\ldots\cdot V_{d-1}(\alpha_{d-2},i_{d-1},j_{d-1},\alpha_{d-1})\cdot V_d(\alpha_{d-1},i_d,j_d),\quad(9)$$

is called a TT representation of the matrix $\mathbf{A}$, the cores $V_1,\ldots,V_d$ are now three- and four-dimensional arrays. Our discussion of the efficiency and robustness of the TT decomposition of vectors also applies to the matrix case.

Note that the *Hierarchical Tensor Representation* [43,44] itself and coupled with the *tensorization* [45], an extensive overview of which is available in [40], are closely related counterparts of the TT and QTT formats respectively. Also, the structure called now TT decomposition has been known in theoretical chemistry as *Matrix Product States (MPS)*. It has been exploited by physicists to describe quantum spin systems theoretically and numerically for at least two decades now, see [46,47], cf. [48].

Basic operations of the numerical calculus with vectors and matrices in the TT format, such as addition, Hadamard and dot products, multi-dimensional contraction, matrix-vector multiplication, etc. are considered in detail in [41]. Since the main aim of using tensor-structured approximations is to reduce the complexity of computations and avoid the curse of dimensionality, we emphasize that the storage cost and complexity of basic operations of the TT arithmetics, applied to the representation (8), can be bounded by dnr^α with $\alpha\in\{2,3\}$, where $n\geq n_1,\ldots,n_d$ and $r\geq r_1,\ldots,r_{d-1}$. This estimate is formally linear in d; however, the TT ranks $r_1,\ldots,r_{d-1}$ in (8) may depend on d and n. Showing that the TT ranks are moderate, e. g. constant or growing linearly with respect to d and constant or growing logarithmically with respect to n, is a crucial issue in the context of TT-structured methods and has been addressed so far mostly experimentally, see, e. g. [49–53].

The TT approximation of a vector with d indices treated separately results in a decomposition with $d-1$ TT ranks which may take different values. To characterize them, an aggregate characteristic such as the *effective rank* of the TT decomposition is used. Consider an $n_1\times\ldots\times n_d$-tensor is given in a TT decomposition with ranks $r_1,\ldots,r_{d-1}$. We call the positive root $r_{\mathrm{eff}}=r$ of the quadratic equation

$$n_1r_1+\sum_{k=2}^{d-1}r_{k-1}n_kr_k+r_{d-1}n_d=n_1r+\sum_{k=2}^{d-1}rn_kr+rn_d\quad(10)$$

the *effective rank of the decomposition*. Note that, for integer values of r, the definition (10) equates the memory needed to store two TT representations. The one corresponding to the left-hand side, is the given decomposition. The one corresponding to the right-hand side is of a vector with the same mode sizes, but with equal $d-1$ ranks $r,\ldots,r$. This renders r_{eff} "effective" with respect to memory. On the other hand, the complexity of some TT-structured operations, such as the matrix-vector multiplication and Hadamard product, can also be estimated with the use of r_{eff}.

Quantized Tensor Train representation. So far, we have discussed the use of the TT format for extracting low-rank structure with respect to the "physical" dimensions, which are naturally distinguished in the data due to their meaning in the context of a particular problem. For the subject of the present paper, such dimensions represent the reacting species. However,

every such a dimension can be unfolded, or *quantized*, into a few virtual dimensions representing its levels, or scales. Then the data can be represented in the TT format applied to all the virtual dimensions introduced. The use of quantization in the context of tensor decompositions dates back to [54]. For the TT format, it results in the *Quantized Tensor Train* (QTT) format [35–37]. For the convenience of the reader, we provide a brief review and refer to [35–37] for details.

Consider a d-dimensional vector of size $n_1 \times \ldots \times n_d$. Assume that the kth mode size n_k can be factorized as $n_k = n_{k,1} \cdot n_{k,2} \cdot \ldots \cdot n_{k,l_k}$ in terms of integral factors $n_{k,1}, \ldots, n_{k,l_k} \geq 2$. Then the kth mode index j_k can be represented in a one-to-one fashion through a tuple $(j_{k,1}, \ldots, j_{k,l_k})$ of l_k *virtual* indices. Here, j_{k,m_k} runs from 0 to $n_{k,m_k} - 1$ for $1 \leq m_k \leq l_k$. The index transformation rule can be defined in many ways.

In order to associate the virtual indices with the scales in the vector, the transformation $j_k \leftrightarrow \sum_{m_k=1}^{l_k} i_{k,m} \prod_{\ell=m_k+1}^{l_k} n_{k,\ell}$ can be chosen. This index (bijective) transformation is analogous to the positional notation for encoding numbers. In this work we indicate this by the overscore notation $\overline{j_{k,1}, \ldots, j_{k,l_k}} = \sum_{m_k=1}^{l_k} j_{k,m_k} \prod_{\ell=m_k+1}^{l_k} n_{k,\ell}$. In the most general case, the "virtual" mode factors $n_{k,1}, \ldots, n_{k,l_k}$, which are analogous to the bases in the positional notation, may differ for different positions $1, \ldots, l_k$.

In terms of the vector, such an index transformation is often called *quantization*. It is equivalent to folding, or reshaping, the kth mode of size n_k into l_k modes of sizes $n_{k1}, \ldots, n_{k,l_k}$. When applied to all dimensions, this procedure transforms a d-dimensional $n_1 \times \ldots \times n_d$-vector indexed by $j_1 = \overline{j_{1,1}, \ldots, j_{1,l_1}}, \ldots j_d = \overline{j_{d,1}, \ldots, j_{d,l_d}}$ into an $l_1 + \ldots + l_d$-dimensional $n_{1,1} \times \ldots \times n_{1,l_1} \times \ldots\ldots \times n_{d,1} \times \ldots \times n_{d,l_d}$-vector indexed by $j_{1,1}, \ldots, j_{1,l_1}, \ldots\ldots, j_{d,1}, \ldots j_{d,l_d}$. A TT decomposition of the quantized vector is referred to as *QTT decomposition* of the original vector; the ranks of this TT decomposition are called *ranks of the QTT decomposition* of the original vector. For details, we refer to the papers [35–37] cited above.

Example 3 (QTT Compression) *Consider a vector* $\mathbf{q}$ *of size* $N > 1$ *given elementwise by*

$$\mathbf{q}_j = \sin(\alpha j), \quad 0 \leq j < N,$$

where $\alpha \in \mathbb{R}$. *Assume that* $N = n^l$, *where* $l = 3$ *and* $n \in \mathbb{N}$. *Then the index* j *running from 0 to* $N-1$ *can be represented as* $j = \overline{j_1 j_2 j_3} = n^2 j_1 + n j_2 + j_3$ *through* $l = 3$ *"virtual" indices* j_1, j_2, j_3 *running from* 0 *to* $n-1$ *each. The corresponding quantization* $\mathbf{p}$ *of* $\mathbf{q}$ *of size* $n \times n \times n$ *is given by*

$$\mathbf{p}_{j_1 j_2 j_3} = \mathbf{q}_{\overline{j_1 j_2 j_3}} = \sin\left(\alpha n^2 j_1 + \alpha n j_1 + \alpha j_3\right), \quad 0 \leq j_1, j_2, j_3 < n.$$

By applying the discussion of Example 1 to $\mathbf{p}$, *we see that the QTT format represents* $\mathbf{q}$ *with the cores and ranks given in Example 1 through* $8n$ *parameters intead of* N *required for the elementwise representation. The case of* $l > 3$ *virtual levels is considered in* [*37, Lemma 2.5*] *and* [*42, Theorem 7*], *the number of parameters being under* $4ln = n \log_n N$ *instead of* N. *In these paper, we use the binary quantization with* $n = 2$.

If the natural ordering

$$\underbrace{j_{1,1}, \ldots, j_{1,l_1}}_{\text{1st dimension}}, \underbrace{j_{2,1}, \ldots, j_{2,l_2}}_{\text{2nd dimension}}, \ldots\ldots, \underbrace{j_{d,1}, \ldots, j_{d,l_d}}_{d\text{th dimension}} \tag{11}$$

of the "virtual" indices is used for representing the quantized vector in the TT format, then the ranks of the QTT decomposition can be enumerated as follows:

$$\underbrace{r_{1,1}, \ldots, r_{1,l_1-1}}_{\text{1st dimension}}, \hat{r}_1, \underbrace{r_{2,1}, \ldots, r_{2,l_2-1}}_{\text{2nd dimension}}, \hat{r}_2, \ldots\ldots, \hat{r}_{d-1}, \underbrace{r_{d,1}, \ldots, r_{d,l_d-1}}_{d\text{th dimension}},$$

where $\hat{r}_1, \ldots, \hat{r}_{d-1}$ are the TT ranks of the original tensor, i.e. the ranks of the separation of "physical" dimensions. That is, the TT ranks of a tensor are some of the QTT ranks of the same tensor, provided that the natrual ordering (11) is used.

Note that equations (8) and (9) can also be understood as QTT representations of a "one-dimensional" vector (i.e. a vector with a single "physical" index) $\mathbf{q}$ and of a "one-dimensional" matrix (i.e. a matrix transforming such vectors) $\mathbf{B}$ with entries $\mathbf{q}_{\overline{j_1, \ldots, j_d}} = \mathbf{p}_{j_1, \ldots, j_d}$ and $\mathbf{B}_{\overline{i_1, \ldots, i_d}; \overline{j_1, \ldots, j_d}} = \mathbf{A}_{i_1, \ldots, i_d; j_1, \ldots, j_d}$ respectively. In this case, d denotes the number of virtual dimensions corresponding to the single "physical" dimension.

As a QTT decomposition is a TT decomposition of an appropriately quantized (and possibly, as we discuss in a later section, transposed) tensor, the TT arithmetics referred to in the previous section, when applied to QTT decompositions, naturally provides the same basic operations in the QTT format.

Quantization is crucial for reducing the computational complexity further, as it allows the TT decomposition to resolve and represent more structure in the data by splitting the "virtual" dimensions introduced by the quantization, as well as the "physical" ones. In practice it appears the most efficient to use as fine a quantization (i.e. with small n_{k,m_k}) as possible and to generate as many virtual modes as possible. For example, when $n_k = 2^{l_k}$ for $1 \leq k \leq d$, one may consider the "*ultimate quantization*" with $n_{k,m_k} = 2$ for all m_k and k. In this case, $j_k = \overline{j_{k,1}, \ldots, j_{k,l_k}} = \sum_{m_k=1}^{l_k} 2^{l_k - m_k} j_{k,m_k}$, where the indices $j_{k,1}, \ldots, j_{k,l_k}$ take the values 0 and 1.

The storage cost and complexity of basic QTT-structured operations are estimated from above through dlr^α with $\alpha \in \{2,3\}$, where $l \geq l_1, \ldots, l_d$ and r is an upper bound on all the QTT ranks of the decomposition in question. Note that this estimate may be, depending on r, logarithmic in n (also in $n^d = 2^{dl}$, which is an upper bound on the number of entries). The notion of the effective rank defined by (10) for TT decompositions applies verbatim to vectors and matrices represented in the QTT format.

Structure of the CME operator in the QTT format. In the following we consider the Finite State Projection of the CME, as described previously, with $n_k = 2^{l_k}$ for $1 \leq k \leq d$ and assume that the PDF $\mathbf{p}$ of the truncated model and of the CME operator $\mathbf{A}$ from (3) are represented in the QTT format outlined in the previous section. We use the ultimate quantization, so that $n_{km} = 2$ for $1 \leq m \leq l_k$ and $1 \leq k \leq d$. In this section we mathematically establish rigorous upper bounds on the QTT ranks of $\mathbf{A}$ under certain assumptions on the propensity vectors ω^s, $1 \leq s \leq R$, defined by (6).

Theorem 4 *Consider the projected CME operator* $\mathbf{A}$ *defined by (3). Assume that for every* $s = 1, \ldots, R$ *and* $k = 1, \ldots, d$ *the one-dimensional vector* ω_k^s *from* (6)–(7) *is given in a QTT decomposition of ranks bounded by* r_k^s; *and that* $\eta_k^s = 0$ *implies* $r_k^s = 1$. *Then the CME operator* $\mathbf{A}$ *admits a QTT decomposition of ranks*

$$q_1, \ldots, q_1, \hat{q}_1, q_2, \ldots, q_2, \hat{q}_2, \ldots, \ldots, \hat{q}_{d-1}, q_d, \ldots, q_d$$

with $\hat{q}_k = 2R$ *for* $1 \le k \le d-1$ *and*

$$q_k = \sum_{\substack{s=1,\dots,R:\\ \eta_k^s = 0}} 2 \quad + \sum_{\substack{s=1,\dots,R:\\ \eta_k^s \neq 0}} 3r_k^s$$

for $1 \le k \le d$.

The proof is provided at the end of Text S1.

A crude upper bound on the QTT ranks of the CME operator, following from Theorem 4 in terms of $r = \max_{s,k} r_k^s$, equals $3 \cdot R \cdot r$ and is still favorable, since it ensures the estimate $\mathcal{O}(dlR^2r^2)$ for the number of parameters, i.e. the storage cost, where $l_1, \dots, l_d \le l$. Note that if the kth factor ω_k^s of the s-th propensity function is a polynomial of degree p_k^s, then ω_k^s (7) can be represented in the QTT format with ranks bounded by $r_k^s = p_k^s + 1$ uniformly in l_k, see [45, Corollary 13] and [42, Theorem 6]. In particular, this is the case when the reaction network is composed entirely of elementary reactions. Our numerical experiments show that the QTT ranks of propensity vectors corresponding to rational propensity functions are low as well, which results in low QTT ranks of the CME operator (in particular, see the toggle switch example).

The rank estimate of Theorem 4 is based on the construction of the CME operator, in which the reactions are treated independently, and the ranks of the terms corresponding to different reactions are summed. However, the bases of the QTT representation of these terms can be related so that the resulting decomposition of the CME operator can be reduced without introducing any error; for example, in the case of polynomial propensity functions. However, the rank bound of Theorem 4 is sharp for general vectors used as propensity vectors.

Transposed QTT representation. So far we have shown that the CME operator (3) under the FSP projection admits the low-parametric representation in the standard QTT format introduced previously. However, such a compressibility of the operator does not imply that the format is suitable for the efficient numerical solution of the CME. The example presented in Section S1.2 hints at a natural modification of the QTT decomposition. We represent in the TT format the quantized vector with virtual dimensions permuted so that the "virtual" indices corresponding to the same levels of quantization of different physical dimensions are adjacent; for example, for $l_1 = \dots = l_d = l$ instead of (11) we use the ordering

$$\underbrace{j_{1,1}, \dots, j_{d,1}}_{\text{1st level}}, \underbrace{j_{1,2}, \dots, j_{d,2}}_{\text{2nd level}}, \dots\dots, \underbrace{j_{1,l}, \dots, j_{d,l}}_{d\text{th level}}. \qquad (12)$$

When $l_1, \dots, l_d$ are not equal, in order to obtain a similar to (12) transposed ordering of indices, we introduce void indices j_{k,m_k} with $n_{k,m_k} = 1$ for $l_k + 1 \le m_k \le \max_{1 \le k' \le d} l_{k'}$, reorder all the "virtual" indices according to (12) and then drop the void ones. This modification of the QTT format, which we refer here to as *quantized-and-transposed Tensor Train*; shortly, *transposed QTT* or *QT3*. It was first applied to vectors in [55].

The index ordering (12) aims at the low-rank representation of such tensors, in which the physical dimensions are coupled on the corresponding virtual levels, i.e. *scales*, much more than different scales are within each single dimension. This is the case for the extreme example (S1.5), where we end up with a rank-one decomposition if we choose to separate the scales first, and the physical dimensions, then. Despite such a difference in approximation properties, from the algorithmic point of view, QT3 is a minor modification of the standard, widely used form of the QTT format. We do not imply any particular ordering of indices by referring simply to QTT.

Structure of the CME operator in the transposed QTT format. Similarly to Theorem 4, we can bound the ranks of the CME operator in the transposed QTT format relying on the ordering (12) of "virtual" indices.

Theorem 5 *Consider the projected CME operator* $\mathbf{A}$ *defined by (3). Assume that for every* $s = 1, \dots, R$ *and* $k = 1, \dots, d$ *the one-dimensional vector* ω_k^s *from (6)–(7) is given in a QTT decomposition of ranks bounded by* r_k^s; *and that* $\eta_k^s = 0$ *implies* $r_k^s = 1$. *Then the CME operator* $\mathbf{A}$ *admits a QT3 decomposition of ranks bounded by*

$$\sum_{s=1}^{R} \left(1 + \prod_{k \in \mathcal{K}^s} 2\right) \left(\prod_{k \in \mathcal{K}^s} r_k^s\right),$$

where $\mathcal{K}^s = \{k \in \mathbb{N} : 1 \le k \le d \text{ and } \eta_k^s \neq 0\}$.

The proof is given at the end of Text S1.

We observe in the enzymatic futile cycle example below that the QT3 ranks of the CME operator may be significantly lower than the bound of Theorem 4.

Time Integration of the CME: hp-Discontinuous Galerkin Discretization

Let us consider the truncated CME (S1.1) with a state space $X = \mathbb{R}^{n_1 \times \dots \times n_d}$ on a finite interval $J = (0, T)$. The Cauchy problem with an initial value $\mathbf{p}_0 \in X$ reads as find a continuously differentiable function $\mathbf{p} : \overline{J} \to X$ such that

$$\begin{cases} \dot{\mathbf{p}}(t) &= \mathbf{A} \cdot \mathbf{p}(t) \quad \text{for} \quad t \in \overline{J}, \\ \mathbf{p}(0) &= \mathbf{p}_0. \end{cases} \qquad (13)$$

The solution to (13) is given theoretically by $\mathbf{p}(t) = \exp(t\mathbf{A}) \cdot \mathbf{p}_0$ for $t \in \overline{J}$, but the straightforward numerical evaluation of the matrix exponential involved is a very challenging task due to the "curse of dimensionality". Instead, we use the QTT-structured *hp-discontinuous Galerkin* (*hp-DG-QTT* for short) time-stepping scheme, proposed in [38], to solve (13). The *hp*-DG time stepping was proposed earlier in [56] for initial value problems for abstract, possibly non-linear, ODEs. We recapitulate the analysis results from [56] for problems of the particular form (13), which have unique, analytic in time classical solutions. To discuss the tensor structure of the *hp*-DG-QTT approach, we revisit [38].

Let us denote by $\mathcal{P}^\rho(I, X)$ the space of polynomials defined on a finite interval I, of degree ρ at most and with coefficients from X. Let $\mathcal{M} = \{J_m\}_{m=1}^M$ be a partition of the time interval J into subintervals $J_m = (t_{m-1}, t_m)$, $1 \le m \le M$, and $\underline{\rho} \in \mathbb{N}_{\ge 0}^M$. Consider the space

$$\mathcal{P}^{\underline{\rho}}(\mathcal{M}, X) = \left\{\mathbf{p} : J \to X : \mathbf{p}|_{J_m} \in \mathcal{P}^{\rho_m}(J_m, X) \quad \text{for} \quad 1 \le m \le M\right\}$$

of functions, which are polynomials of degree ρ_m at most on J_m for all m. For all $\mathbf{q} \in \mathcal{P}^{\underline{\rho}}(\mathcal{M}, X)$ let $\mathbf{q}_m^+ = \lim_{t \downarrow t_m} \mathbf{q}(t)$ and $\mathbf{q}_m^- = \lim_{t \uparrow t_m} \mathbf{q}(t)$ for all feasible m.

Definition 6. *The hp-DG formulation of (13), corresponding to the partition* $\mathcal{M}$ *of the time interval and the vector* $\underline{\rho}$ *of polynomial degrees, reads*

as follows: find $\mathbf{p}\in\mathcal{P}^{\underline{\rho}}(\mathcal{M},X)$ *such that*

$$\sum_{m=1}^{M}\int_{J_m}\langle\dot{\mathbf{p}}-\mathbf{A}\mathbf{p},\mathbf{q}\rangle\mathrm{d}t+\sum_{m=1}^{M}\langle\mathbf{p}_{m-1}^{+}-\mathbf{p}_{m-1}^{-},\mathbf{q}_{m-1}^{+}\rangle=0 \qquad (14)$$

for all $\mathbf{q}\in\mathcal{P}^{\underline{\rho}}(\mathcal{M},X)$*, where* $\mathbf{p}_0^-$ *stands for the initial value* $\mathbf{p}_0$.

Equation (14) can be understood as a time-stepping method: if for all m from 1 up to $\ell-1$ the polynomial $\mathbf{p}|_{J_m}\in\mathcal{P}^{\rho_m}(J_m,X)$ is known through ρ_m+1 coefficients from X, then $\mathbf{p}|_{J_\ell}\in\mathcal{P}^{\rho_\ell}(J_\ell,X)$ can be found as the solution to

$$\int_{J_\ell}\langle\dot{\mathbf{p}}-\mathbf{A}\mathbf{p},\mathbf{q}\rangle\mathrm{d}t+\langle\mathbf{p}_{\ell-1}^{+}-\mathbf{p}_{\ell-1}^{-},\mathbf{q}_{\ell-1}^{+}\rangle=0. \qquad (15)$$

For $1\le m\le M$ let $\{\phi_j\}_{j=0}^{\rho_m}$ be a basis in $\mathcal{P}^{\rho_m}((-1,1),X)$, then the corresponding temporal shape functions on J_m are $\phi_j\circ F_m^{-1}$, $0\le j\le\rho_m$, where the affine map $F_m:(-1,1)\to J_m$ is defined by $t=F_m(\tau)=\frac{1}{2}(t_m+t_{m-1})+\frac{1}{2}(t_m-t_{m-1})\tau$ for $\tau\in(-1,1)$. If $\mathbf{p}|_{J_m}=\sum_{j=0}^{\rho_m}(\mathbf{P}_m)_j\cdot(\phi_j\circ F_m^{-1})$, where $\mathbf{P}_m\in X^{\rho_m+1}\simeq\mathbb{R}^{(\rho_m+1)\times n_1\times\ldots\times n_d}$, then (15) yields the following linear system on the coefficients:

$$(\mathbf{C}_m\otimes\mathbb{I}-\mathbf{G}_m\otimes\mathbf{A})\cdot\mathbf{P}_m=\phi_{m-1}\otimes\mathbf{p}_{m-1}^{-}, \qquad (16)$$

where $(\mathbf{C}_m)_{i,j}=\int_{-1}^{1}\phi'_j(\tau)\phi_i(\tau)\mathrm{d}\tau+\phi_j(-1)\phi_i(-1)$ and $(\mathbf{G}_m)_{i,j}=\int_{-1}^{1}\phi_j(\tau)\phi_i(\tau)\mathrm{d}\tau$ for $0\le i,j\le\rho_m$, while $(\phi_{m-1})_i=\phi_i(-1)$ for $0\le i\le\rho_m$.

The *hp*-DG time discretization allows, on the one hand, to resolve fast transients in the evolution by the time-step and polynomial order adaptation for time-analytic solutions given through matrix exponentials of the CME operator. In particular, due to the time-analyticity of the solution, exponential rates of convergence in $\underline{\rho}$ are achieved; for example, for the "*h*-version" with $\underline{\rho}=(\rho_0,\ldots,\rho_0)$ the error bound of Proposition 3 of Text S1 can be recast as

$$\sup_{t\in\bar{J}}\|\mathbf{p}(t)-\hat{\mathbf{p}}(t)\|_2\le C\exp(-b\rho_0)$$

with constants $C,b>0$ asymptotically independent of ρ_0, see [56, Theorem 3.18]. This implies that a prescribed level of accuracy ε can be reached with $\rho_0 M=\mathcal{O}(\log\varepsilon^{-1})$ temporal degrees of freedom.

In the tensor representation of the system (16) we keep the QTT format used for $\mathbf{A}$ and attach the temporal index as a single dimension (without quantization) to the first "virtual" spatial index. In Section S1.3 we present this format in more detail.

Theorem 7. *Assume that* $\mathbf{A}$ *is represented in the QTT or QT3 format in terms of* D *cores with ranks* $r_1,\ldots,r_{D-1}$*. Then the matrix of system* (16) *can be represented in the corresponding format in terms of* $D+1$ *cores with ranks* $2,r_1+1,\ldots,r_{D-1}+1$.

The proof is given at the end of Text S1.

As an alternative to the presently considered order and stepsize adaptive time-stepping, it has been proposed in [5] to use a low-order time discretization with a uniformly small step and rely on tensor-structured compression methods also for time-adaptivity. This approach leads to one large linear system with low-rank structure. We found this approach to be more demanding to the tensor-structured solvers, since the aggregate linear system for all time steps seems to be more difficult to solve. A remedy may be to partition the time interval into subintervals with possibly different time steps being used within each such subinterval, which already shifts the approach in the direction of the presently proposed *hp*-DG method. In the presence of time inhomogeneity the aggregate systems in general lose their low-rank structure rendering the space-time tensor approach less efficient, while the *hp*-DG method would still perform well.

Algorithm 1. Assemble Projected CME Operator in QTT Matrix Format.

Require: Rank-1 separable propensity functions $\omega^s(\underline{x})$, stoichiometric vectors $\underline{\eta}^s$, rectangular FSP truncation $[0,\ldots,2^{l_1}-1]\times\ldots\times[0,\ldots,2^{l_d}-1]$, propensity QTT compression tolerance ϵ_{prop}, a QTT approximation subroutine QTT_Approx implementing [4, Algorithm 1] for quantized vectors.

Ensure: Projected CME operator $\mathbf{A}$ in QTT matrix format

Initialize $\mathbf{A}=0$;

for $s=1,\ldots,R$ **do**

$\mathbf{S}_{\underline{\eta}^s}=\mathbf{S}_{\eta_1^s}^{(l_1)}\otimes\ldots\otimes\mathbf{S}_{\eta_d^s}^{(l_d)}$;

for $k=1,\ldots,d$ **do**

$\boldsymbol{\omega}_k^s=\mathrm{QTT\_Approx}(\boldsymbol{\omega}_k^s(0,\ldots,2^{l_k}-1))$ with tolerance ϵ_{prop};

end for

$\omega^s=\omega_1^s\otimes\ldots\otimes\omega_d^s$;

$\mathbf{M}_{\omega^s}=\mathrm{diag}\,\omega^s$;

$\mathbf{A}=\mathbf{A}+\left(\mathbf{S}_{\underline{\eta}^s}-\mathbb{I}\right)\circ\mathbf{M}_{\omega^s}$;

end for

Algorithm Summary

Assuming we have a finite state projection of the CME, we summarize our approach to the CME solution by outlining the two main algorithms we propose for its subsequent efficient solution. Given a reaction network and a finite state projection Algorithm 1 (Box 1) approximates the CME operator in QTT format. Algorithm 2 (Box 2) then describes the time-stepping procedure for computing the solution. Note that the integrals in Algorithm 2 may be pre-computed depending on the choice of temporal basis functions. E.g. if one chooses the Legendre polynomials as the basis, then there are explicit solutions of the integrals involved.

Comparison to Krylov Subspace Methods

The solution at a particular time of a finite state projection of the CME is given analytically by the matrix exponential, but the numerical computation of such solutions for large $\mathbf{A}$ is often expensive. When $\mathbf{A}$ is sparse, however, the Krylov subspace method [57,58] is one approach for performing the computation for the CME as described in [59]. The method uses the Arnoldi iteration to compute the Krylov subspace up to some order of accuracy then computes the matrix exponential in that smaller space (by diagonal Padé approximation). The publicly available Expokit Toolbox by Sidje [60] provides an implementation of the algorithm.

Algorithm 2. *hp*-DG-QTT CME Solver.
Require: Projected CME operator A in QTT format, time mesh $\mathcal{M}=\{J_m\}_{m=1}^{M}$, polynomial orders $\underline{\rho}\in\mathbb{N}_{\geq 0}^{M}$, basis of temporal shape functions $\{\phi_j\}_{j=0}^{\infty}$, DMRG-solver tolerance RES
Ensure: Approximate solution $\mathbf{p}\in\mathcal{P}^{\underline{\rho}}(\mathcal{M},X)$ of the evolution $\dot{p}=Ap$
for $m=1,\ldots,M$ **do**
for $i,j=0,\ldots,\rho_m$ **do**

$$(\mathbf{C}_m)_{i;j}=\int_{-1}^{1}\phi'_j(\tau)\phi_i(\tau)\mathrm{d}\tau+\phi_j(-1)\phi_i(-1);$$

$$(\mathbf{G}_m)_{i;j}=\int_{-1}^{1}\phi_j(\tau)\phi_i(\tau)\mathrm{d}\tau;$$

end for
Solve $(\mathbf{C}_m\otimes\mathrm{II}-\mathbf{G}_m\otimes\mathbf{A})\cdot\mathbf{P}_m=\phi_{m-1}\otimes\mathbf{p}_{m-1}^{-}$, for $\mathbf{P}_m$ using DMRG-solver with tolerance RES;

$$\mathbf{p}_m=\sum_{j=1}^{\rho_m}\mathbf{P}_{m,j}\vartheta_j(1);$$

end for

It is important to note that the algorithm steps incrementally in time rather than jumping to the desired time step. In the context of the CME, this means that the faster the support of the pdf fills the set of reachable states, the more expensive this algorithm becomes to compute. When there is reason to believe the support of the pdf remains small, then the algorithm can be expected to compute efficiently over large time intervals. Generically, however,the support of the pdf quickly fills the set of reachable states which may include every state retained in the projection. This renders the Arnoldi iteration computationally expensive at each time step.

The QTT method effectively circumvents this problem by storing the computed solution at each time step in the QTT format and exploiting the fast algorithms for basic tensor arithmetic available in this format. While it is unknown whether a given reaction network and initial probability distribution will produce an evolution that can be represented well by a QTT formatted tensor with low QTT ranks, our numerical experiments find this often is the case and that the savings over using traditional sparse representations of vectors and matrices may be quite substantial.

Below we compare our method to the Krylov subspace approach in the toggle switch example which does not exhibit any pronounced structure favoring either one of the methods (rank-one separability and sparse structure respectively).

Numerical experiments

Common details. At the mth time step, after having obtained $\mathbf{P}_m$ as an approximate solution of the corresponding linear system (16), we evaluate $\mathbf{p}_m^-$ and reapproximate it in the TT format with relative ℓ_2-accuracy EPS in order to drop excessive QTT components. The values of EPS and the complete set of parameters of the time discretization and of the DMRG solver are reported for each experiment in Section S1.7.

We compare the evaluated solution or its marginal to a reference data. By Δ_{ℓ_p} we denote the ℓ_p-norm of the discrepancy. Generally we start with the ℓ_2-norm, which can be easily computed even when the comparison is made only in the (Q)TT format and cannot be made in the full format (which is the case in the d-independent birth-death processes experiments for $d\geq 3$). In some cases we compute also the discrepancy for $p=1$ and the probability deficiency $\mathrm{ERR}_{\Sigma}\left[\mathbf{p}_m^-\right]=\left|1-\sum\mathbf{p}_m^-\right|$. The reference data is also obtained with a certain accuracy which cannot be reduced arbitrarily. Moreover, in some cases our solution appears to be more accurate, which accounts for using the term "discrepancy" instead of "error".

In the first and third examples we reapproximate the solution once more, with relative ℓ_2-accuracy $\alpha\cdot\frac{\Delta_{\ell_2}}{\left\|\mathbf{p}_m^-\right\|}$, where α is 0.05 and 0.01 respectively. Below we refer to this procedure as *truncation*, and the approximated vector, as *truncated solution*. The procedure ensures that the relative discrepancy in the ℓ_2-norm grows by the factor of $1+\alpha$ at most and shows what QTT ranks allow for our numerical solution, obtained without using any reference data, to ensure *almost* the same discrepancy from the reference data (which is related to the accuracy of both the solution and reference data) as before truncation.

***d* independent birth-death processes.** As a first example we consider a system composed of d chemical species with $\{X_1,\ldots,X_d\}$ a vector of random variables representing the species count of each. The dynamics of the random vector are governed by independent birth-death processes. For the k-th species, the corresponding reactions are given by

$$\emptyset\underset{d_k}{\overset{b_k}{\rightleftharpoons}}\mathbf{X}_k$$

where b_k is the spontaneous creation rate and d_k is the destruction rate for species X_k. This problem is perfectly separable in the sense that the dynamics of any one chemical species of this system is independent of the dynamics of all others. Given the initial condition $X_k(0)=\xi_k$ for each k, the marginal distribution for any one species X_k at time t is given by:

$$p_k(x_k;t)=\mathcal{P}(x_k,\lambda_k(t))\star_{x_k}\mathcal{M}(x_k,\xi_k,p^{(k)}(t)),\quad x_k\in\mathbb{Z}_{\geq 0}$$

where $\mathcal{P}(\cdot,\lambda_k(t))$ is the Poisson distribution with parameter $\lambda_k(t)$, $\star_{x_k}$ indicates the discrete convolution in variable x_k, $\mathcal{M}\left(x_k,\xi_k,p^{(k)}(t)\right)$ the multinomial distribution with parameter $p^{(k)}(t)$, and the parameters $p^{(k)}$ and λ_k evolve according to the reaction rate equations

$$\frac{\mathrm{d}}{\mathrm{d}t}p^{(k)}(t)=-d_kp^{(k)}(t),\quad \frac{\mathrm{d}}{\mathrm{d}t}\lambda_k(t)=b_k-d_k\lambda_k(t),$$
$$p^{(k)}(0)=1,\qquad \lambda_k(t)=0.$$

See [4, Theorem 1] for details. Since $X_1,\ldots,X_k$ are mutually independent, the joint PDF at time t, $\mathbf{p}(t)$, is the product of the marginals:

$$\mathbf{p}(t)=\prod_{k=1}^{d}p_k(t)$$

that is, this system has an explicit formula for the solution regardless of the number of chemical species involved. We can, therefore, evaluate the accuracy and observe the complexity scaling of the *hp*-DG-QTT solver as the number of chemical species increases.

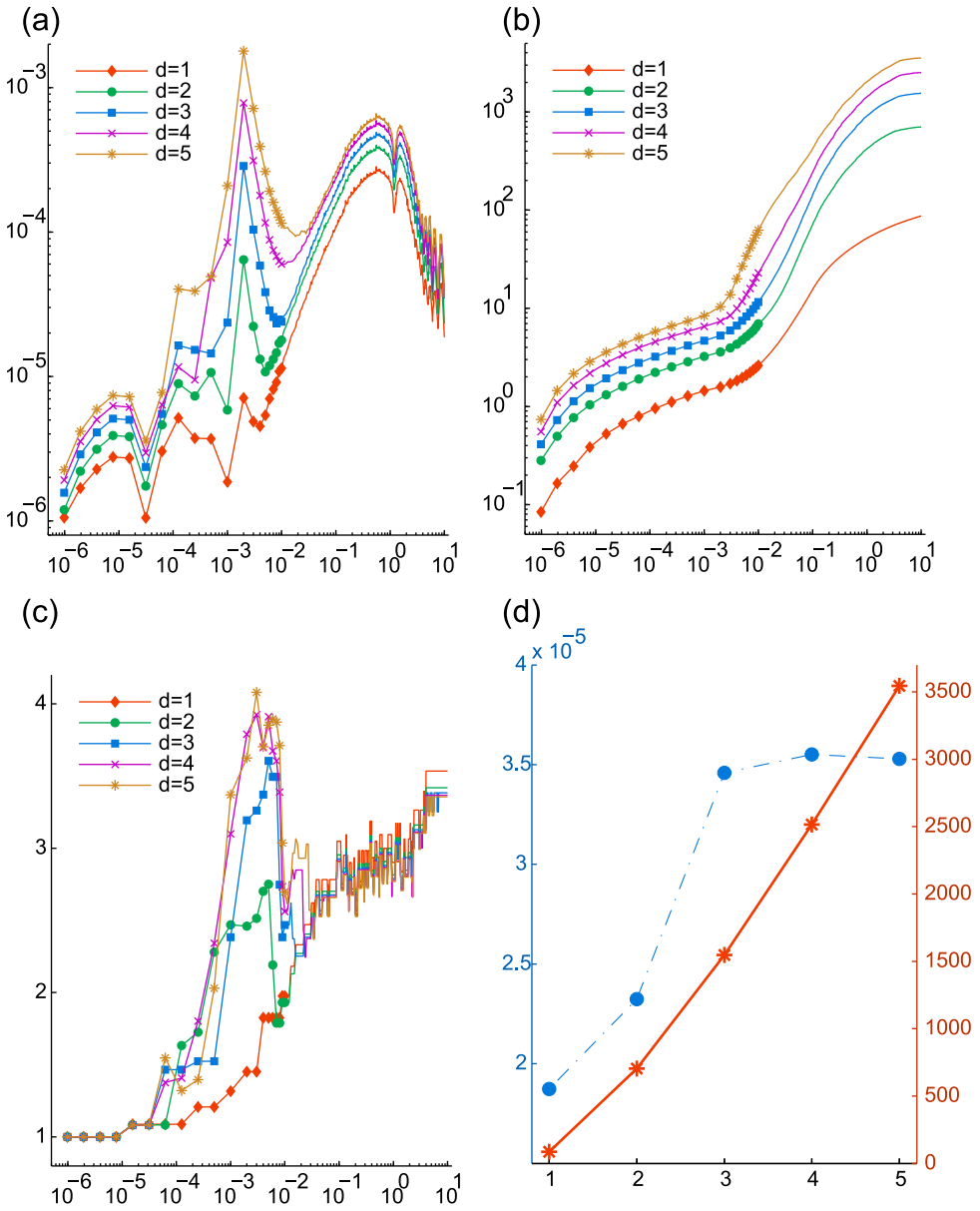

Figure 2. d independent birth-death processes. The maximum QTT ranks of the solutions, $r_{\max}\left[\mathbf{p}_M^-\right]=6$ for each d. Markers are omitted for $t_m>10^{-2}$ in (a)–(c). (a) Relative discrepancy $\Delta_{\ell_2}\left[\mathbf{p}_m^-\right]/\left\|\mathbf{p}_M^-\right\|$ (after truncation) vs. t_m. (b) Cumulative computation time (in seconds) vs. t_m. (c) Effective QTT rank $r_{\text{eff}}\left[\mathbf{p}_m^-\right]$ (after truncation) vs. t_m. (d) Relative discrepancy $\Delta_{\ell_2}\left[\mathbf{p}_M^-\right]/\left\|\mathbf{p}_M^-\right\|$ (blue) and total computation time (red) vs. d.
doi:10.1371/journal.pcbi.1003359.g002

For numerical simulations we assume $b_k=1000$ and $d_k=1$ for $1\le k\le d$ and consider the FSP with $l_k=12$. We solve the corresponding projected CME for $d=1,2,3,4,5$ to check that in all these cases the *hp*-DG-QTT method using the ordering (11) without transposition is capable of revealing the same low-rank QTT structure of the solution. For the CME operator we have $r_{\max}[\mathbf{A}]\le 8$ up to accuracy $5\cdot 10^{-15}$. We compute the evolution of the PDF of the system for the zero initial value through $M=569$ time steps till $T=10$.

The results, which are presented in Figure 2 and Table 2, show that the same low-rank structure of the solution is adaptively reconstructed by the algorithm for all d considered. The transient phase causes the growth of QTT ranks, because at certain steps of every sweep the DMRG solver merges virtual dimensions corresponding to different species and attempts to reduce the numerical rank by re-separating these dimensions. As a consequence, during the transient phase numerical QTT ranks are overestimated, which does not affect the QTT structure of the numerical solution at larger times.

Toggle switch. The next example models a synthetic gene-regulatory circuit designed to produce bistability over a wide range of parameter values [61]. The network consists of two promoters constructed in a mutually inhibitory configuration that implement a double negative feedback loop, causing the network to exhibit robust bistable behavior (see Figure 3). If the concentration of one repressor is high, this lowers the production rate of the other repressor, keeping its

Table 2. d independent birth-death processes: $r_{\text{eff}} = r_{\text{eff}}\left[\mathbf{p}_M^-\right]$, $\Delta_{\ell_2} = \Delta_{\ell_2}\left[\mathbf{p}_M^-\right]$, computational TIME in seconds; $r_{\max}\left[\mathbf{p}_M^-\right] = 6$ for all d.

d	N	$\frac{\|\mathbf{A}\mathbf{p}_0\|_2}{\|\mathbf{p}_0\|_2}$	$\frac{\|\mathbf{A}\mathbf{p}_M^-\|_2}{\|\mathbf{p}_M^-\|_2}$	r_{eff}	Δ_{ℓ_2}	TIME
1	2^{12}	1.4_{+3}	1.0_{-3}	3.53	1.9_{-5}	87
2	2^{24}	2.4_{+3}	1.4_{-3}	3.42	2.3_{-5}	704
3	2^{36}	3.5_{+3}	1.8_{-3}	3.38	3.5_{-5}	1548
4	2^{48}	4.5_{+3}	2.0_{-3}	3.37	3.6_{-5}	2516
5	2^{60}	5.5_{+3}	2.3_{-3}	3.36	3.5_{-5}	3544

N is the number of states taken into account in the FSP. The exponents are given in boldface for the base 10.

doi:10.1371/journal.pcbi.1003359.t002

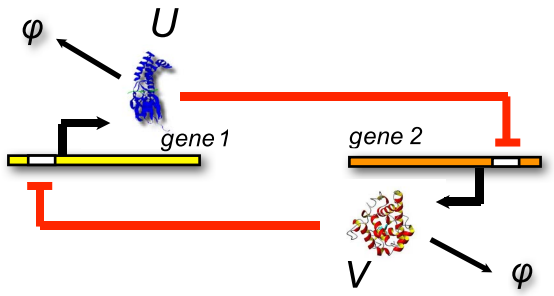

Figure 3. Toggle Switch consisting of double negative feedback loop. Species U represses the production of species V and vice versa.

doi:10.1371/journal.pcbi.1003359.g003

concentration low. This allows a high rate of production of the original repressor, thereby stabilizing its high concentration.

A stochastic model of the toggle switch was considered in [62] and consists of the following four reactions:

$$\emptyset \xrightarrow{\frac{\alpha_1}{1+V^\beta}} \mathrm{U}, \qquad \emptyset \xrightarrow{\frac{\alpha_2}{1+U^\gamma}} \mathrm{V}$$

$$\mathrm{U} \xrightarrow{\delta_1} \emptyset, \qquad \mathrm{V} \xrightarrow{\delta_2} \emptyset$$

where U and V represent the two repressors. Denote the species counts of each by U and V, respectively. The stochastic model admits a bimodal stationary distribution over a wide range of values of the rate constants. We consider the set of parameters from [62] which were selected to test the efficiency of using available numerical algorithms to calculate matrix exponentials to solve low dimensional FSP approximations of the CME. We then scaled the parameters so that a larger set of states would be required to guarantee an FSP truncation with low approximation error. While a different set of parameters were considered in [23,63], which required a larger FSP truncation, this choice of values renders the system symmetric under interchange of the roles of U and V. This situation is less biologically relevant than what we consider here.

For this numerical example we assume $\alpha_1 = 5000$, $\alpha_2 = 1600$, $\beta = 2.5$, $\gamma = 1.5$, $\delta_1 = \delta_2 = 1$. We consider the FSP with $l_U = 13$, $l_V = 12$, which allows to take into account 2^{25} states. The initial value is zero. We use the ordering (11) without transposition. For the CME operator we have $r_{\max}[\mathbf{A}] = 14$ and $r_{\text{eff}}[\mathbf{A}] = 10.89$ up to accuracy 10^{-14}. We compute the evolution of the PDF up to time $T = 100$ with $M = 1111$ time steps.

The results are presented in Figure 4. At the terminal time T we have $\mathrm{ERR}_\Sigma\left[\mathbf{p}_M^-\right] = 3.17 \cdot 10^{-5}$. The overall computation time is 14728 seconds. The validation with the PDF based on 816 million Monte Carlo simulations (every 1000 draws taking on average over 360 seconds, adding up to the overall CPU time over $3 \cdot 10^8$ seconds), indicates $\Delta_{\ell_1}\left[\mathbf{p}_M^-\right] = 8.34 \cdot 10^{-4}$, and for the 2- and Chebyshev norms we have $\Delta_{\ell_2}\left[\mathbf{p}_M^-\right] / \left\|\mathbf{p}_M^-\right\|_2 = 6.62 \cdot 10^{-4}$ and $\Delta_{\ell_\infty}\left[\mathbf{p}_M^-\right] = 5.50 \cdot 10^{-6}$. As for the ranks, $r_{\text{eff}}\left[\mathbf{p}_M^-\right] = 8.74$ and $r_{\max}\left[\mathbf{p}_M^-\right] = 13$. Figure 4 (c) shows that after $t \approx 20$ the norm of the time derivative stagnates at approximately 10^{-5} determined by the accuracy parameters chosen, and the following time steps require negligible computational effort. At the same time, as we see in Figure 4 (b), all QTT ranks stabilize under 15, but the transient phase preceding that moment involves far higher ranks. Figure 5 (a) presents a snapshot of the distribution.

Comparison to the Krylov subspace approach. We compared the performance of our proposed method to that of the Krylov subspace approach implemented in Sidje's Expokit [60]. In order to make the comparison as fair as possible we further restricted the FSP truncation used by the Krylov approach to a *hyperbolic cross*, that is, we only kept states with indices (j_U, j_V) satisfying the condition $(j_U + 1) \cdot (j_V + 1) < 9216000$. Effectively, this reduces the states in the truncation from 2^{25} to 21120695, a reduction of about a third. A similar truncation was used for this model in [62].

We emphasize that formulating this hyperbolic cross truncation requires special insight into the problem on the part of the modeler. In constrast, our proposed method is completely naive in this respect, instead relying on the adaptivity of the QTT compression.

For the FSP with 2^{25} states considered we reach $t \approx 1$ with the first 43 time steps of our method in 4385 seconds; with the Krylov subspace method restricted to the hyperbolic cross, in 10333 seconds. For the discrepancy between the two solutions obtained we have $\Delta_{\ell_1} = 4.04 \cdot 10^{-5}$ and $\Delta_{\ell_\infty} = 9.64 \cdot 10^{-8}$.

At approximately $t = t_{43} \approx 1$, the decay of the relative norm of the solution becomes exponential; see Figure 4 (c). That is exploited by our method in two ways. On the one hand, we adjust the time mesh manually, which reduces the overall number of time steps needed to reach $t_{1111} = T$ from t_{43}: we take 1068 steps intead of approximately 3307 we would need if we had used a uniform time mesh for the long-term dynamics. On the other hand, what is more significant, the adaptive QTT representation used at each step yields a substantial speedup of the solution of linear systems, which is possible due to the rapid convergence of the solution to a stationary distribution. The Krylov subspace solver adapts the time mesh on its own, but employs no self-adaptivity for efficient storage of numerically computed states. As a result, the performance (in terms of the computational time vs. physical time of the system) decays much slower for the Krylov subspace solver, and our method excels even more in modelling the long-term dynamics. For example, our method achieves $t \approx 30$, when $\|\dot{\mathbf{p}}\|_2 / \|\mathbf{p}\|_2$ reaches $1.1 \cdot 10^{-5}$, with the overall computation time 14541 seconds compared to 126530 seconds of the Krylov subspace solver, i.e. approximately 8.7 times faster. For larger

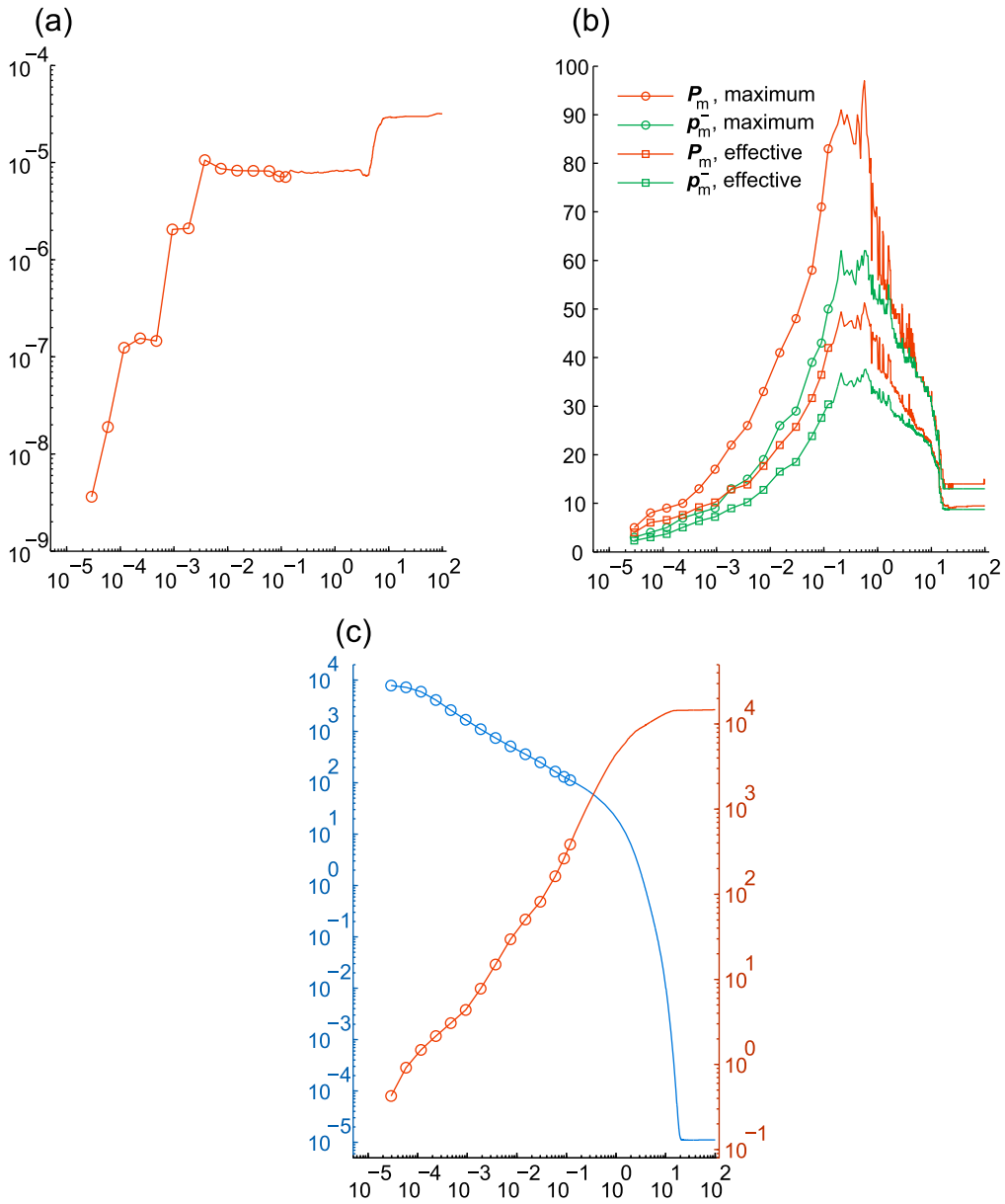

Figure 4. Genetic toggle switch. The values are given vs. t_m. Markers are omitted for $t_m > 10^{-1}$. (a) Probability deficiency $\mathrm{ERR}_\Sigma\left[\mathbf{p}_m^-\right]$. (b) Maximum and effective QTT ranks of the computed solution. (c) Relative norm $\frac{\left\|\mathbf{A}\mathbf{p}_m^-\right\|_2}{\left\|\mathbf{p}_m^-\right\|_2}$ of the derivative (blue) and cumulative computation time (red, sec.)
doi:10.1371/journal.pcbi.1003359.g004

terminal times the advantage of our method becomes even more pronounced.

Enzymatic futile cycle. Futile cycles are composed of two metabolic or signaling pathways that work in opposite directions so that the products of one pathway are the precursors of the other and vice versa, see Figure 6. This biochemical network structure results in no net production of molecules and often results only in the dissipation of energy as heat [64]. Nevertheless, there is an abundance of known pathways that use this motif and it is thought to provide a highly tunable control mechanism with potentially high sensitivity [64,65].

[65] introduced a stochastic version of the model with just the essential network components required to model the dynamics. The stochastic model consists of six chemical species and six reactions:

$$X + E_+^{\mathrm{f}} \underset{k_{+2}}{\overset{k_{+1}}{\rightleftharpoons}} E_+^b, \quad \mathrm{X}^* + \mathrm{E}_-^{\mathrm{f}} \underset{k_{-2}}{\overset{k_{-1}}{\rightleftharpoons}} \mathrm{E}_-^{\mathrm{b}},$$

$$\mathrm{E}_+^{\mathrm{b}} \xrightarrow{k_{+3}} \mathrm{E}_+^{\mathrm{f}} + \mathrm{X}^*, \quad \mathrm{E}_-^{\mathrm{b}} \xrightarrow{k_{-3}} \mathrm{E}_-^{\mathrm{f}} + \mathrm{X},$$

$\{\mathrm{X},\mathrm{X}^*\}$ represent the forward substrate and product, $\{\mathrm{E}_+,\mathrm{E}_-\}$

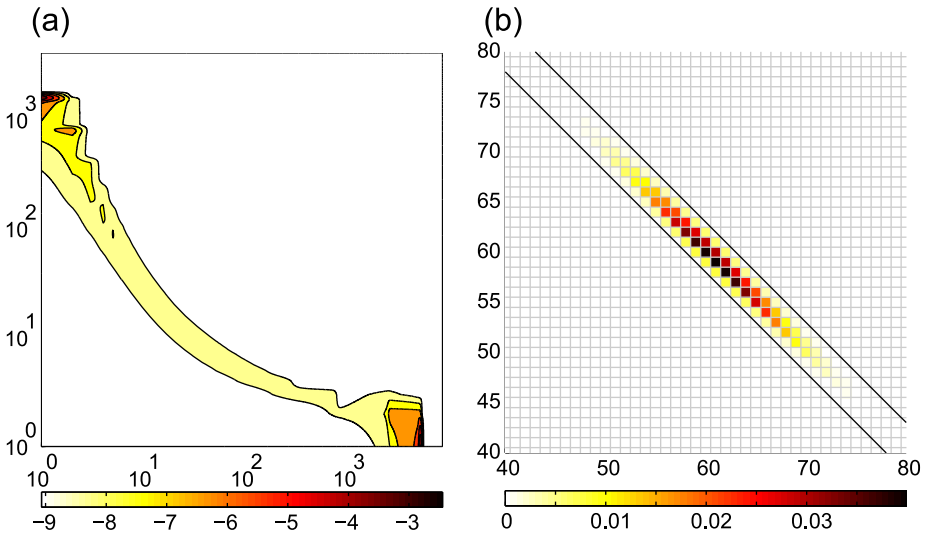

Figure 5. Snapshots of solutions. (a) Genetic toggle switch. The PDF for $m=350$, $t_m \approx 10.18$, U (hor.) vs. V (vert.). As the process evolves, the probability mass becomes concentrated in two distinct regions. Contour coloring is logarithmically scaled with base 10. (b) Enzymatic futile cycle. The marginal PDF for $m=20$, $t_m = 5 \cdot 10^{-3}$, X (vert.) vs. X^* (hor.). Black diagonal lines delimit the states reachable from the initial condition. The transposed QTT format automatically exploits this sparsity pattern *of the full PDF* for compression without special input from the user.
doi:10.1371/journal.pcbi.1003359.g005

denote the forward and reverse enzymes, respectively. Note that this system is closed meaning that particles are neither created nor destroyed. We denote the random variables representing the molecule count of each species with italics.

For the particular set of initial conditions considered in [65] the number of states that are reachable is large enough to render a direct numerical solution of the CME impractical. The authors instead used the Gillespie Direct SSA to generate a large number of sample paths to estimate the distribution. The authors also applied a diffusion approximation to their model which resulted in a SDE which produced qualitatively similar dynamics. To the authors' knowledge, no attempt has been made so far towards the direct numerical solution of the CME for this system.

At time t, let $X^{\mathrm{T}}(t)$ denote the total amount of both free and bound substrate, and $E_+^{\mathrm{T}}(t)$ and $E_-^{\mathrm{T}}(t)$ the total forward and reverse enzymes, respectively. We observe the following conservation relations:

$$E_+^{\mathrm{f}}(t)+E_+^{\mathrm{b}}(t)=E_+^{\mathrm{T}}(t)=E_+^{\mathrm{T}}(0)$$

$$E_-^{\mathrm{f}}(t)+E_-^{\mathrm{b}}(t)=E_-^{\mathrm{T}}(t)=E_-^{\mathrm{T}}(0)$$

$$X(t)+X^*(t)+E_+^{\mathrm{b}}(t)+E_-^{\mathrm{b}}(t)=X^{\mathrm{T}}(t)=X^{\mathrm{T}}(0)$$

Using the above, one can establish an upper and lower bound relating the species count of $\mathrm{X}(t)$ to $\mathrm{X}^*(t)$ that depends only on the total initial amount of substrate and the total initial amount of enzymes in the system

$$X^{\mathrm{T}}(0)-X^*(t)\geq X(t)\geq X^{\mathrm{T}}(0)-X^*(t)-\left(E_+^{\mathrm{T}}(0)+E_-^{\mathrm{T}}(0)\right).$$

Assuming that the initial quantity of enzymes $E_+^{\mathrm{T}}(0)+E_-^{\mathrm{T}}(0)$ is small, for a given copy number of $X^*(t)$, $X(t)$ may take at most

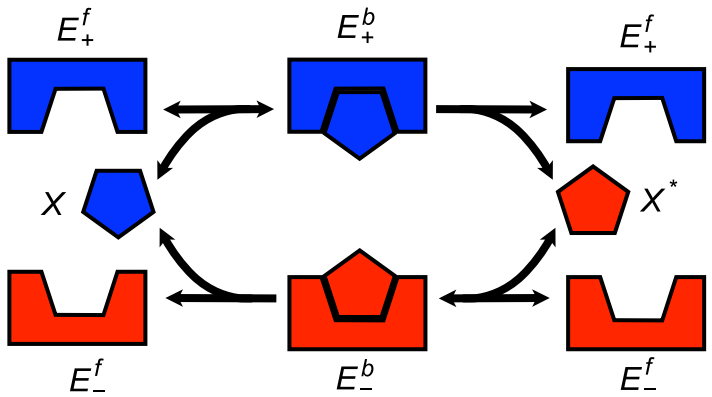

Figure 6. Enzymatic futile cycle. X is transformed into X^* and vice versa by enzymes E_+ and E_-, respectively.
doi:10.1371/journal.pcbi.1003359.g006

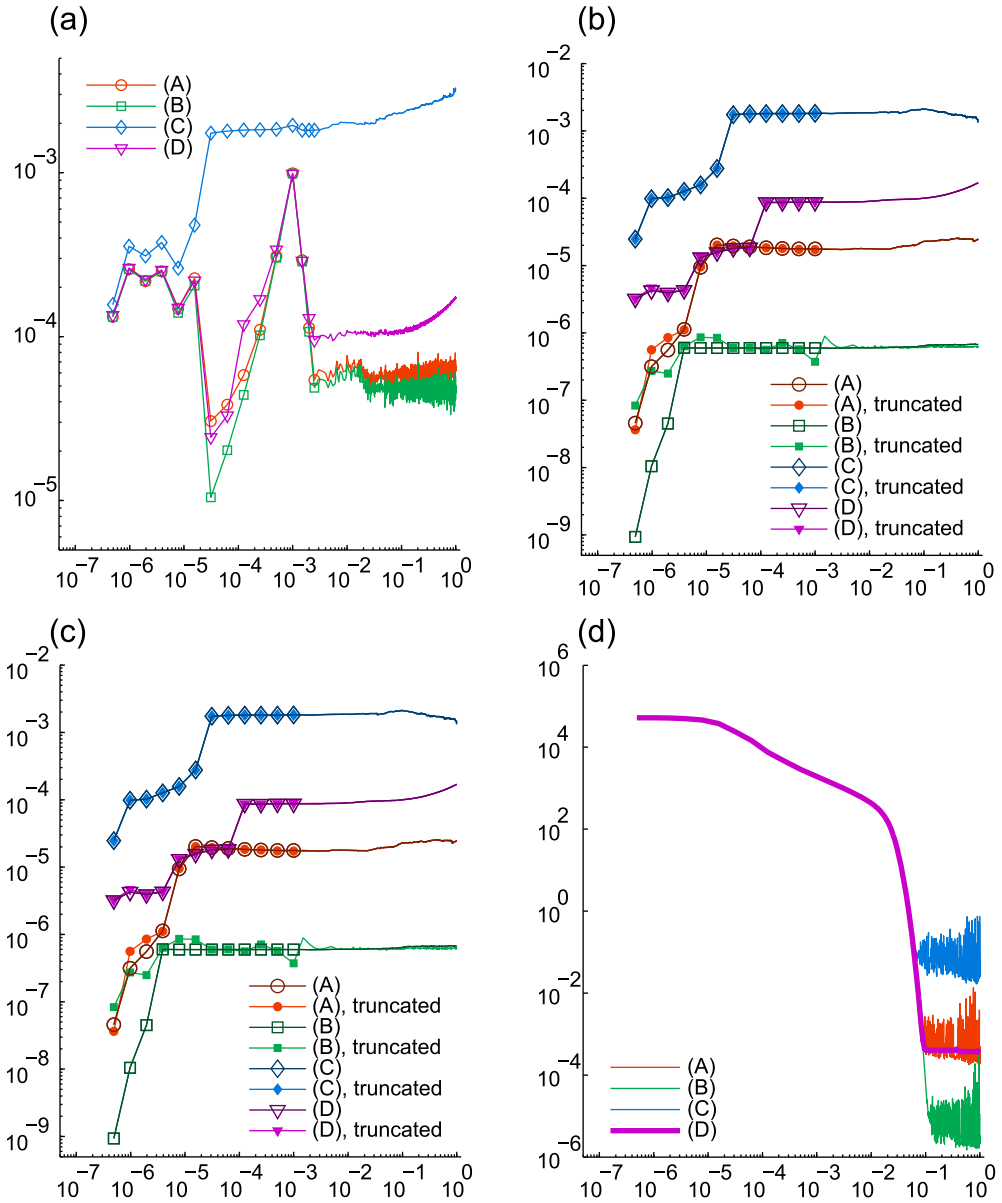

Figure 7. Enzymatic futile cycle. The values are given vs. t_m. Markers are omitted for $t_m \geq 2 \cdot 10^{-3}$ in (a)–(c). (a) Discrepancy Δ_{ℓ_1} (before truncation) from the marginal PDF based on Monte Carlo simulations. (b) Probability deficiency $\mathrm{ERR}_{\Sigma}[\mathbf{p}_m^-]$. (c) Cumulative computation time (sec.) (d) Relative norm $\frac{\|\mathbf{A}\mathbf{p}_m^-\|_2}{\|\mathbf{p}_m^-\|_2}$ of the derivative.

doi:10.1371/journal.pcbi.1003359.g007

$E_+^{\mathrm{T}}(0)+E_-^{\mathrm{T}}(0)$ different values. Since $X^{\mathrm{T}}(t)$ is a conserved quantity, this means that $X(t)$ and $X^*(t)$ will be strongly anti-correlated with the set of reachable states having an affine structure. Under these circumstances, we find in our numerical experiments that the transposed QTT format is better suited than the standard QTT to efficiently represent the corresponding PDF, since the transposed format better utilizes the sparsity pattern of the full PDF for compression.

Following [65], we consider $k_{+1}=40$, $k_{+2}=10^4$, $k_{+3}=10^4$, $k_{-1}=200$, $k_{-2}=100$, $k_{-3}=5000$. For initial value we take $E_{\pm}^f=2, E_{\pm}^b=0, X=30, X^*=90$. We consider the FSP projection with $l_{E_{\pm}^{b,f}}=2$ and $l_X=l_{X^*}=7$, i.e. with 2^{22} states. We present 4 runs: (A), (B) and (C) use the transposed QTT format, and (D), the standard QTT. Theorems 5 and 4 bound the exact QTT ranks of the CME operator by 216 and 21 respectively, and numerically for accuracy 10^{-14} we have $r_{\max}[\mathbf{A}]=38$, $r_{\mathrm{eff}}[\mathbf{A}]=17.93$ in (A)–(C) and $r_{\max}[\mathbf{A}]=11$, $r_{\mathrm{eff}}[\mathbf{A}]=8.30$ in (D). We compute the evolution of the PDF up to time $T=1$ with $M=1332$ time steps.

For the runs (A) and (D), which differ in the format, we keep the same accuracy parameters. The runs (B) and (C) use the same format as (A), but different accuracy parameters, so that they yield, respectively, a more accurate and a cruder solution as compared to (A).

This experiment shows, in particular, that lower ranks of the operator do not necessarily lead to lower ranks of the solution, and that in this example the transposed QTT format actually ensures smaller ranks of the solution than the QTT format without

Table 3. Enzymatic futile cycle: $r_{\mathrm{eff}} = r_{\mathrm{eff}}\left[\mathbf{p}_m^-\right]$, $r_{\max} = r_{\max}\left[\mathbf{p}_m^-\right]$, $\Delta_{\ell_1} = \Delta_{\ell_1}\left[\sum_{E_\pm^{\mathrm{b,f}}} \mathbf{p}_m^-\right]$, $\mathrm{ERR}_\Sigma = \mathrm{ERR}_\Sigma\left[\mathbf{p}_m^-\right]$ are given for the truncated solution $\mathbf{p}_m^-$; computational TIME is given in seconds; $\frac{\|\mathbf{A}\mathbf{p}_0\|_2}{\|\mathbf{p}_0\|_2} = 5.2\cdot 10^4$.

run	$\frac{\|\mathbf{A}\mathbf{p}_m^-\|_2}{\|\mathbf{p}_m^-\|_2}$	r_{eff}	$r_{\max}$	Δ_{ℓ_1}	ERR_Σ	TIME
$m = 210$, $t_m = 0.1$						
(A)	3.5_{-4}	13.17	27	5.7_{-4}	2.3_{-5}	1.07_3
(B)	6.5_{-5}	12.14	25	4.6_{-5}	6.1_{-7}	1.60_3
(C)	1.3_{-1}	12.16	24	2.3_{-3}	2.1_{-3}	9.87_2
(D)	4.1_{-4}	60.06	109	1.1_{-4}	1.0_{-4}	9.23_3
$m = M = 1322$, $t_m = T = 1$						
(A)	1.8_{-4}	13.66	27	7.2_{-5}	2.5_{-5}	3.70_3
(B)	1.1_{-5}	12.06	25	5.7_{-5}	6.2_{-7}	4.21_3
(C)	2.5_{-2}	12.85	24	3.3_{-3}	1.3_{-3}	4.03_3
(D)	3.7_{-4}	58.97	107	1.7_{-4}	1.7_{-4}	1.52_4

The exponents are given in boldface for the base 10.

doi:10.1371/journal.pcbi.1003359.t003

transposition does and than Theorem 5 suggests. As for the solution, we observe that $\max_{0\le t_m\le 0.1} r_{\max}[\mathbf{P}_m]$ reaches 51 for (A) and 359 for (D).

For every m, we validate our solution $\mathbf{p}_m^-$ by comparing its marginal distribution $\sum_{E_\pm^{\mathrm{b,f}}} \mathbf{p}_m^-$ to that based on $18.6\cdot 10^9$ Monte Carlo simulations (every 10000 draws taking at least 110 seconds, amounting to the overall CPU time over $2\cdot 10^8$ seconds). The discrepancy $\Delta_{\ell_p} = \Delta_{\ell_p}\left[\sum_{E_\pm^{\mathrm{b,f}}} \mathbf{p}_m^-\right]$ in the marginal distribution with respect to X and X^* is reported for $p = 1$ in Figure 7 (a) and Table 3. With $p = 2$ we use it for the discrepancy-based truncation, which, as Figure 7 (b) shows, does not affect the probability deficiency significantly.

Figure 7 (a) shows that the refined run (B) yields the smallest discrepancy, which suggests that the reference distribution is sufficiently accurate to allow for the discrepancy to represent the actual error in the results of (A), (B) and (C). As we can see from Figure 7 (d), in all 4 runs the time derivative stagnates after $t \approx 0.1$, at lower levels for more accurate runs. Let us note that at that stage in (A)–(C) it exhibits relatively strong oscillations compared to (D), which happens due to different effect of the addition of random components in the DMRG solver in the presence and absence of the transposition. On the other hand, compared to (A), the run (D) yields a less accurate solution and reaches $t = 0.1$ almost 9 times later, the accuracy settings being the same in these two runs. In all, the transposition appears to make the QTT format far more efficient in this experiment, and we expect it to be even more so in larger systems of such type.

The results are given in Figures 7 and 8 and in Table 3. Figure 5 (b) presents a snapshot of the marginal distribution.

Conclusion

We presented a novel, "ab-initio" computational methodology for the direct numerical solution of the CME. The methodology exploits the time-analytic nature of solutions to the CME and the low-rank, tensor structure of the CME operator by combining an *hp*-timestepping method that is order and step size adaptive, unconditionally stable and exponentially convergent with respect to the number of time discretization parameters, with novel, tensor-formatted linear algebra techniques for the numerical realization of the method. In particular, after an initial projection on a (sufficiently rich) finite state, the QTT representation allows for the dynamic adaptation of the effective state-space size, as well as of the principal components, or basis elements of the numerical representation of solution vectors in the numerical simulation of the time evolution of the CME solution. We emphasize that, while the performance of our approach is better when the solution can be approximated in the QTT format with a high degree of separability of the "physical" and "virtual" variables (i.e. with low TT ranks), the approach does not require a particular degree of separability, but instead reveals possibly present low TT rank in the solution at runtime. In the course of rank adaptation, the singular vectors, in the span of which the solution is approximated, are also adapted. Hence, the presently proposed approach is superior to fixed basis approaches (even when used with adaptivity), such as those reported in [19,22,23,66]. The precise class of chemical reaction networks that lead to low TT rank in the solution tensor is currently unknown. To the extent that this rank increase during runtime, the effectiveness of the compression will be decreased, which could prove limiting for some problems. However, in this case other methods will be equally challenged. Identifying the architecture of the chemical reaction networks that lead to very low ranks is currently a research problem under investigation.

While the discussion following Theorem 4 relates to the case when the factors of the propensity functions are monomial, the approach presented herein applies equally well to models with propensity functions that are merely smooth enough. For example [67], gives bounds on the QTT ranks of the propensity functions and CME operator in the case of the stochastic mass-action and Michaelis–Menten kinetics with separable propensity functions. Also, the same work proves the bounds on the QTT ranks of product-form stationary distributions [68] of *weakly-reversible* reaction networks of *zero deficiency in the sense of Feinberg* [69]. Those bounds explain some of the experimental observations made in the present paper. Furthermore, the approach proposed is suitable for non-separable propensity functions. However, in that case the characterization of the rank structure of the CME operator needs to rely on some extra assumptions ensuring moderate QTT ranks, even though more general than separability, and Algorithm 1 needs to be altered accordingly.

The performance of the approach proposed essentially relies on the efficiency of the numerical solution of TT-structured linear systems of equations. In particular, a globally (or "less strictly locally") convergent iterative solver would allow us to take larger time steps and to exploit the exponential convergence of the *hp*-DG time discretization. We believe that while the presently reported numerical results which were obtained with the DMRG solver are quite encouraging, ongoing research on TT-structured linear system solvers holds the promise for a substantial efficiency increase of the present methodology. We only mention a family of alternating minimal energy methods which was announced very recently in [70].

We also mention that, of course, the choice of the tensor format and, possibly, index ordering, has an essential impact on the performance of the approach. The computational experiments reported in the present paper show that even a straightforward permutation of "virtual" indices produced by quantization may allow to exploit additional structure in the data and the QTT formatted CME solution and, therefore, may improve the performance of the QTT-structured approach dramatically. We point out that the TT format can be considered as a special case of

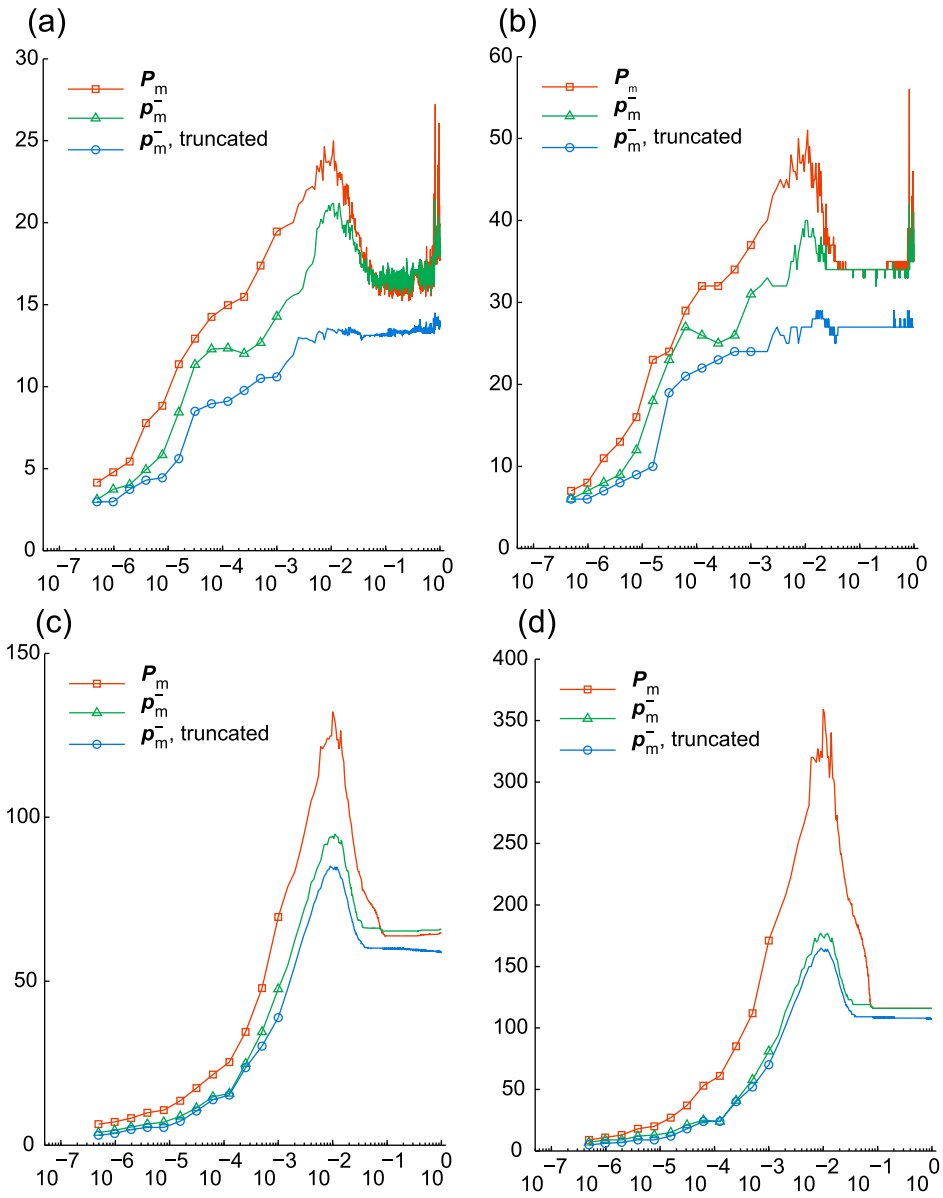

Figure 8. Enzymatic futile cycle. QTT ranks of the solution. The values are given vs. t_m. Markers are omitted for $t_m \geq 2 \cdot 10^{-3}$. (a) Effective QTT ranks r_{eff} for parameter set (A). (b) Maximum QTT ranks $r_{\max}$ for parameter set (A). (c) Effective QTT ranks r_{eff} for parameter set (D). (d) Maximum QTT ranks $r_{\max}$ for parameter set (D).
doi:10.1371/journal.pcbi.1003359.g008

tensor network states: TT formatted tensors belong to the class of simple, rooted tree-type tensor networks. Relating the architecture of the chemical reaction networks and appropriate tensor networks representing its states efficiently, i.e. with low ranks, is currently a research problem under development. The results of [67] mentioned above can be considered as the first step in this direction.

A general discussion of tensor networks and their use in numerical simulations for quantum spin systems can be found in [71,72]. As for the numerical solution of the CME, particular real-life problems might require more sophisticated tensor networks to be used to efficiently approximate reachable states of the systems in question. The mathematical investigation of the relative merits and drawbacks of tensor formats for particular applications is currently undergoing rather active development; we mention only the recent monograph [40] and the references there.

We finally mention that recently, and independently, TT formatted linear algebra methods for the CME were proposed in [73]; a low order time stepping, and no transposition of tensor trains was used in that work. The CME examples presented in [73] also included a toggle switch, but the authors mostly rely on the intrinsic convergence of their method without analyzing actual accuracies. The latter are reported only for moderate sized examples which are computationally tractable with the direct approach in the full format. However, no attempt is made to analyze the accuracy in comparison to other simulation methods, which are typically applied to larger problems featuring essential difficulties for the direct approach. In the present paper we give comparisons with a state-of-the-art, massively parallel stochastic simulation package. This allows us, on the one hand, to validate the accuracy of the QTT-based solutions obtained here and, on the other hand, to provide evidence of the dramatic increase in

efficiency afforded by the new deterministic approach: Monte Carlo simulations on 1500 cores of a high-performance cluster were matched in accuracy and outperformed in the wall-clock time by a MATLAB implementation running on a notebook.

Methods

To solve the initial value problem for (2), we exploit the *hp*-DG-QTT algorithm proposed in [38] and adapted to the CME as described above, implemented in MATLAB. It uses an implicit, exponentially convergent spectral time discretization of discontinuous Galerkin type. The resulting, time-discrete CME in "species space" is solved in the QTT format. Our implementation relies on the public domain *TT Toolbox* which provides basic TT-structured operations and solvers for linear systems in the QTT format. The TT toolbox is publicly available at http://spring.inm.ras.ru/osel and http://github.com/oseledets/TT-Toolbox; to be consistent, we use the GitHub version of July 12, 2012 in all examples below. We run the *hp*-DG-QTT solver in MATLAB 7.12.0.635 (R2011a) on a laptop with a 2.7 GHz dual-core processor and 4 GB RAM, and report the computational time in seconds.

For the solution of the large, linear systems in the QTT and QT3 formats in each time step, we use the optimization solver, based on the DMRG approach [46–48] and elaborated on in the context of the TT format in [74] and available as the function dmrg_solve3 of the TT Toolbox. While the "DMRG" solver still lacks a rigorous theoretical foundation, it proves to be highly efficient in many applications, including our experiments. In [75] a closely related *Alternating Least Squares (ALS)* approach was mathematically analyzed and shown to converge locally. More on the mathematical ideas behind the ALS and DMRG optimization in the TT format can be found in [76].

The "DMRG" solver, under certain restrictions on the time step, manages to find a parsimonious QTT formatted solution of the linear system (up to a specified tolerance). Moreover, the solver in effect automatically adapts both the QTT rank as well as the QTT "basis" of the solution at every time step guaranteeing that it is sufficiently rich in order to capture the principal dynamics of interest.

In the first numerical example the solution is symmetric and exactly rank-one separable, which allows us to use the standard MATLAB solver ode15 s in the sparse format to obtain the univariate factor of a reference solution. In other examples we used SPSens beta 3.4, a massively parallel package for the stochastic simulation of chemical networks (http://sourceforge.net/projects/spsens/) [77], to construct reference PDFs. The stochastic simulations were carried out on up to 1500 cores of Brutus, a high-performance cluster of ETH Zürich (*https://www1.ethz.ch/id/services/list/comp_zentral/cluster/index_EN*).

Supporting Information

Text S1 Supplementary Material for direct solution of the Chemical Master Equations using Quantized Tensor Trains.
(PDF)

Author Contributions

Conceived and designed the experiments: VK MN. Performed the experiments: VK MN. Analyzed the data: VK MN. Wrote the paper: VK MK MN CS. Conceived the approach: VK MK MN CS. Implemented hp-DG-QTT approach and the transposed QTT format: VK.

References

1. Elowitz M, Levine A, Siggia E, Swain P (2002) Stochastic gene expression in a single cell. Nature 297: 1183–1186.
2. McAdams HH, Arkin A (1997) Stochastic mechanisms in gene expression. Proceedings of the National Academy of Sciences of the United States of America 94: 814–819.
3. Gillespie DT (1976) A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. Journal of Computational Physics 22: 403–434.
4. Jahnke T, Huisinga W (2007) Solving the chemical master equation for monomolecular reaction systems analytically. Journal of Mathematical Biology 54: 1–26.
5. Munsky B, Khammash M (2006) The finite state projection algorithm for the solution of the chemical master equation. The Journal of Chemical Physics 124: 044104.
6. Henzinger T, Mateescu M, Wolf V (2009) Sliding window abstraction for infinite markov chains. In: Bouajjani A, Maler O, editors, Computer Aided Verification, Springer Berlin/Heidelberg, volume 5643 of *Lecture Notes in Computer Science*. pp. 337–352. URL *http://dx.doi.org/*10.1007/978-3-642-02658-42$_2$ 7. 10.1007/978-3-642-02658-4-27.
7. Cao Y, Liang J (2008) Optimal enumeration of state space of finitely buffered stochastic molecular networks and exact computation of steady state landscape probability. BMC Systems Biology 2: 30.
8. Cao Y, Lu HM, Liang J (2010) Probability landscape of heritable and robust epigenetic state of lysogeny in phage lambda. Proceedings of the National Academy of Sciences of the United States of America 107: 18445–18450.
9. Gibson MA, Bruck J (2000) Efficient exact stochastic simulation of chemical systems with many species and many channels. The Journal of Physical Chemistry A 104: 1876–1889.
10. Gillespie DT (2001) Approximate accelerated stochastic simulation of chemically reacting systems. The Journal of Chemical Physics 115: 1716–1733.
11. Hespanha JP, Singh A (2005) Stochastic models for chemically reacting systems using polynomial stochastic hybrid systems. Int J on Robust Control, Special Issue on Control at Small Scales: Issue 1 15: 669–689.
12. Gomez-Uribe CA, Verghese GC (2007) Mass fluctuation kinetics: Capturing stochastic effects in systems of chemical reactions through coupled mean-variance computations. The Journal of Chemical Physics 126: 024109.
13. van Kampen NG (1992) Stochastic Processes in Physics and Chemistry. Amsterdam and New York: North-Holland.
14. Gillespie DT (2000) The chemical langevin equation. The Journal of Chemical Physics 113: 297–306.
15. Ethier SN, Kurtz TG (2005) Markov Processes: Characterization and Convergence. New York: Wiley-Interscience.
16. Puchalka J, Kierzek AM (2004) Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. Biophysical Journal 86: 1357–1372.
17. Haseltine EL, Rawlings JB (2002) Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. The Journal of Chemical Physics 117: 6959–6969.
18. Salis H, Kaznessis Y (2005) Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions. The Journal of Chemical Physics 122: 054103.
19. Hellander A, Lotstedt P (2007) Hybrid method for the chemical master equation. Journal of Computational Physics 227: 100–122.
20. Jahnke T (2011) On reduced models for the chemical master equation. Multiscale Modeling and Simulation 9: 1646.
21. Nip M, Hespanha J, Khammash M (2012) A spectral methods-based solution of the chemical master equation for gene regulatory networks. In: Decision and Control (CDC), 2012 IEEE 51st Annual Conference on. pp. 5354–5360. doi:10.1109/CDC.2012.6425804.
22. Engblom S (2009) Spectral approximation of solutions to the chemical master equation. Journal of Computational and Applied Mathematics 229: 208–221.
23. Deuflhard P, Huisinga W, Jahnke T, Wulkow M (2008) Adaptive discrete Galerkin methods applied to the chemical master equation. SIAM Journal on Scientific Computing 30: 2990–3011.
24. Hegland M, Burden C, Santoso L, MacNamara S, Booth H (2007) A solver for the stochastic master equation applied to gene regulatory networks. Journal of Computational and Applied Mathematics 205: 708–724.
25. Jahnke T, Udrescu T (2010) Solving chemical master equations by adaptive wavelet compression. Journal of Computational Physics 229: 5724–5741.
26. Bellman R (1961) Adaptive Control Processes: A Guided Tour. Princeton, NJ: Princeton University Press.
27. Hitchcock FL (1926) The expression of a tensor or a polyadic as a sum of products. Journal of Mathematics and Physics 6: 164–189.
28. Caroll JD, Chang JJ (1970) Analysis of individual differences in multidimensional scaling via an n-way generalization of "Eckart-Young" decomposition. Psychometrika 35: 283–319.

29. Hegland M, Garcke J (2011) On the numerical solution of the chemical master equation with sums of rank one tensors. ANZIAM Journal 52: C628–C643.
30. de Silva V, Lim LH (2008) Tensor rank and the ill-posedness of the best low-rank approximation problem. SIAM Journal on Matrix Analysis and Applications 30: 1084–1127.
31. Håstad J (1990) Tensor rank is NP-complete. Journal of Algorithms 11: 644–654.
32. Hillar C, Lim LH (2009) Most tensor problems are NP hard. arXiv abs/0911.1393.
33. Jahnke T, Huisinga W (2008) A dynamical low-rank approach to the chemical master equation. Bulletin of Mathematical Biology 70: 2283–2302.
34. Oseledets IV, Tyrtyshnikov EE (2009) Breaking the curse of dimensionality, or how to use SVD in many dimensions. SIAM Journal on Scientific Computing 31: 3744–3759.
35. Oseledets I (2009) Approximation of matrices with logarithmic number of parameters. Doklady Mathematics 80: 653–654.
36. Oseledets IV (2010) Approximation of $2^d \times 2^d$ matrices using tensor decomposition. SIAM Journal on Matrix Analysis and Applications 31: 2130–2145.
37. Khoromskij BN (2011) $\mathcal{O}(d \log n)$-quantics approximation of *n-d* tensors in high-dimensional numerical modeling. Constructive Approximation 34: 257–280.
38. Kazeev V, Reichmann O, Schwab C (2012) *hp*-DG-QTT solution of high-dimensional degenerate diffusion equations. Research Report 11, Seminar for Applied Mathematics, ETH Zürich. URL *http://www.sam.math.ethz.ch/reports/2012/11*.
39. Kolda TG, Bader BW (2009) Tensor decompositions and applications. SIAM Review 51: 455–500.
40. Hackbusch W (2012) Tensor Spaces and Numerical Tensor Calculus, volume 42 of *Springer Series in Computational Mathematics*. *Springer*. *doi:10.1007/978-3-642-28027-6*. *URL http*://www.springerlink.com/content/l62t86.
41. Oseledets IV (2011) Tensor Train decomposition. SIAM Journal on Scientific Computing 33: 2295–2317.
42. Oseledets I (2013) Constructive representation of functions in low-rank tensor formats. Constructive Approximation 37: 1–18.
43. Hackbusch W, Kühn S (2009) A new scheme for the tensor representation. Journal of Fourier Analysis and Applications 15: 706–722.
44. Grasedyck L (2010) Hierarchical Singular Value Decomposition of tensors. SIAM Journal on Matrix Analysis and Applications 31: 2029–2054.
45. Grasedyck L (2010) Polynomial approximation in Hierarchical Tucker Format by vectortensorization. Preprint 308, Institut für Geometrie und Praktische Mathematik, RWTH Aachen. URL *http://www.igpm.rwth-aachen.de/Download/reports/pdf/IGPM308k.pdf*.
46. White SR (1993) Density-matrix algorithms for quantum renormalization groups. Phys Rev B 48: 10345–10356.
47. Verstraete F, Porras D, Cirac JI (2004) Density matrix renormalization group and periodic boundary conditions: A quantum information perspective. Phys Rev Lett 93: 227205.
48. Vidal G (2003) Efficient classical simulation of slightly entangled quantum computations. Phys Rev Lett 91: 147902.
49. Ballani J, Grasedyck L (2013) A projection method to solve linear systems in tensor format. Numerical Linear Algebra with Applications 20: 27–43.
50. Kressner D, Tobler C (2011) Preconditioned low-rank methods for high-dimensional elliptic PDE eigenvalue problems. Computational Methods in Applied Mathematics 11: 363–381.
51. Dolgov S, Khoromskij B, Oseledets I (2012) Fast solution of parabolic problems in the tensor train/quantized tensor train format with initial application to the fokker–planck equation. SIAM Journal on Scientific Computing 34: A3016–A3038.
52. Kressner D, Tobler C (2010) Low-rank tensor Krylov subspace methods for parametrized lin- ear systems. Research Report 16, Seminar for Applied Mathematics, ETH Zürich. URL *http://www.sam.math.ethz.ch/reports/2010/16*.
53. Khoromskij BN, Schwab C (2011) Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs. SIAM Journal on Scientific Computing 33: 364–385.
54. Tyrtyshnikov EE (2003) Tensor approximations of matrices generated by asymptotically smooth functions. Sbornik: Mathematics 194: 941–954.
55. Oseledets IV (2010) QTT decomposition of the characteristic function of a simplex. Personal communication.
56. Schotzau D, Schwab C (2000) An hp a priori error analysis of the DG time-stepping method for initial value problems. Calcolo 37: 207–232.
57. Saad Y, Schultz M (1986) Gmres: A generalized minimal residual algorithm for solving nonsym- metric linear systems. SIAM Journal on Scientific and Statistical Computing 7: 856–869.
58. Saad Y (1992) Analysis of some krylov subspace approximations to the matrix exponential operator. SIAM Journal on Numerical Analysis 29: 209–228.
59. MacNamara S, Burrage K, Sidje R (2008) Multiscale modeling of chemical kinetics via the master equation. Multiscale Modeling & Simulation 6: 1146–1168.
60. Sidje RB (1998) Expokit: a software package for computing matrix exponentials. ACM Trans Math Softw 24: 130–156.
61. Gardner TS, Cantor CR, Collins JJ (2000) Construction of a genetic toggle switch in Escherichia coli. Nature 403: 339–342.
62. Munsky B, Khammash M (2008) The finite state projection approach for the analysis of stochastic noise in gene networks. Automatic Control, IEEE Transactions on 53: 201–214.
63. Sjöberg P, Lötstedt P, Elf J (2009) Fokker–Planck approximation of the master equation in molecular biology. Computing and Visualization in Science 12: 37–50.
64. Schwender J, Ohlrogge J, Shachar-Hill Y (2004) Understanding flux in plant metabolic networks. Current Opinion in Plant Biology 7: 309–317.
65. Samoilov M, Plyasunov S, Arkin AP (2005) Stochastic amplification and signaling in enzymatic futile cycles through noise-induced bistability with oscillations. Proceedings of the National Academy of Sciences of the United States of America 102: 2310–2315.
66. Jahnke T (2010) An adaptive wavelet method for the chemical master equation. SIAM Journal on Scientific Computing 31: 4373.
67. Kazeev V, Schwab C (2013) Tensor approximation of stationary distributions of chemical reaction networks. Research Report 18, Seminar for Applied Mathematics, ETH Zürich. URL *http://www.sam.math.ethz.ch/reports/2013/18*.
68. Anderson DF, Craciun G, Kurtz TG (2010) Product-form stationary distributions for deficiency zero chemical reaction networks. Bulletin of Mathematical Biology 72: 1947–1970.
69. Feinberg M (1979). Lectures on chemical reaction networks.URL *http://www.chbmeng.ohio-state.edu/feinberg/LecturesOnReactionNetworks*.
70. Dolgov SV, Savostyanov DV (2013) Alternating minimal energy methods for linear sys- tems in higher dimensions. Part I: SPD systems. arXiv preprint 1301.6068. URL *http://arxiv.org/abs/1301.6068*.
71. Verstraete F, Cirac JI, Murg V (2009) Matrix Product States, Projected Entangled Pair States, and variational renormalization group methods for quantum spin systems. arXiv preprint 0907.2796. URL *http://arxiv.org/abs/0907.2796*.
72. Cirac JI, Verstraete F (2009) Renormalization and tensor product states in spin chains and lattices. Journal of Physics A: Mathematical and Theoretical 42: 504004.
73. Dolgov SV, Khoromskij BN (2012) Tensor-product approach to global time-space-parametric discretization of chemical master equation. Preprint 68, Max-Planck-Institut für Mathematik in den Naturwissenschaften. URL *http://www.mis.mpg.de/publications/preprints/2012/prepr2012 – 68.html*.
74. Oseledets I, Dolgov S (2012) Solution of linear systems and matrix inversion in the tt-format. SIAM Journal on Scientific Computing 34: A2718–A2739.
75. Rohwedder T, Uschmajew A (2012) Local convergence of alternating schemes for optimization of convex problems in the TT format. Preprint 112, DFG-Schwerpunktprogramm 1324. URL *http://www.dfg-spp1324.de/download/preprints/preprint112.pdf*.
76. Holtz S, Rohwedder T, Schneider R (2012) The alternating linear scheme for tensor optimization in the Tensor Train format. SIAM Journal on Scientific Computing 34: A683–A713.
77. Sheppard PW, Rathinam M, Khammash M (2013) Spsens: a software package for stochastic parameter sensitivity analysis of biochemical reaction networks. Bioinformatics 29: 140–142.