The Journal of Chemical Physics

RESEARCH ARTICLE | JULY 15 2021

# Tensor-train approximation of the chemical master equation and its application for parameter inference

Ion Gabriel Ion ; Christian Wildner ; Dimitrios Loukrezis ; Heinz Koeppl ; Herbert De Gersem

Check for updates

*J. Chem. Phys.* 155, 034102 (2021)
https://doi.org/10.1063/5.0045521

View Online

Export Citation

CrossMark

AIP Publishing

# Tensor-train approximation of the chemical master equation and its application for parameter inference

Cite as: J. Chem. Phys. **155**, 034102 (2021); doi: 10.1063/5.0045521
Submitted: 27 January 2021 • Accepted: 28 June 2021 •
Published Online: 15 July 2021

Ion Gabriel Ion,[1,2,a)] Christian Wildner,[2] Dimitrios Loukrezis,[1,2] Heinz Koeppl,[1,2,3] and Herbert De Gersem[1,2]

AFFILIATIONS

[1] Centre for Computational Engineering, Technische Universität Darmstadt, Darmstadt, Germany
[2] Department of Electrical Engineering and Information Technology, Technische Universität Darmstadt, Darmstadt, Germany
[3] Centre for Synthetic Biology, Technische Universität Darmstadt, Darmstadt, Germany

[a)] **Author to whom correspondence should be addressed:** ion@temf.tu-darmstadt.de

## ABSTRACT

In this work, we perform Bayesian inference tasks for the chemical master equation in the tensor-train format. The tensor-train approximation has been proven to be very efficient in representing high-dimensional data arising from the explicit representation of the chemical master equation solution. An additional advantage of representing the probability mass function in the tensor-train format is that parametric dependency can be easily incorporated by introducing a tensor product basis expansion in the parameter space. Time is treated as an additional dimension of the tensor and a linear system is derived to solve the chemical master equation in time. We exemplify the tensor-train method by performing inference tasks such as smoothing and parameter inference using the tensor-train framework. A very high compression ratio is observed for storing the probability mass function of the solution. Since all linear algebra operations are performed in the tensor-train format, a significant reduction in the computational time is observed as well.

*© 2021 Author(s). All article content, except where otherwise noted, is licensed under a Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).* https://doi.org/10.1063/5.0045521

## I. INTRODUCTION

Traditional chemical kinetic models use ordinary differential equations (ODEs) to predict the concentrations of the involved molecule types. The evolution of the corresponding probability distribution is given by the chemical master equation (CME),[1] which, in principle, can be solved by numerical integration. In practice, the state space even of simple models is too large for a naive integration of the CME. Therefore, a number of approximation techniques have been developed over the years, e.g., stochastic simulation methods[2–4] and suitable time and space discretizations.[5,6] Many CME approximations are based on the observation that the probability mass is often concentrated on a small fraction of the state space. For example, the finite state projection method solves the CME on a rectangular subspace with appropriate boundary conditions.[7,8] A problem here is that the region where the significant part of the probability mass function (PMF) is located may change over time. This is tackled in the sliding window method, where the region of interest is adapted based on the solution at the previous time step.[9] Unfortunately, the computational cost of these methods grows exponentially with the number of species due to the fact that the system states must be labeled explicitly to cast the CME into an ODE.

A different line of research has explored approximations of the CME based on low-rank tensor formats.[10–14] The idea is to project the probability distribution onto a subspace of the tensor product space induced by the reaction system. The solution is then propagated by a small time step and projected back onto the chosen space. Alternatively, considering time as an additional dimension, a joint approximation of the space–time system in a low-rank tensor format can be obtained.[10,11] The low-rank tensor representation not only preserves the structure of the CME but is also much more memory-efficient compared to the matrix representation approach.[15] In addition, the low-rank tensor representation allows for accurate, dynamic approximations via rank rounding.

Low-rank tensor decompositions have also been considered in the context of parameter-dependent CMEs, however, with limited applications so far.[10] For example, considering systems and synthetic biology, stochastic chemical kinetics are used to construct quantitatively predictive models of biomolecular circuits. This requires the solution of an inverse problem, where it is often necessary to estimate the rate parameters of a structurally known candidate system from time course data. For population snapshot data, e.g., obtained from flow cytometry measurements, calibration via moment-based inference is a well-established method.[16–18] In the last decade, advances in fluorescence microscopy have led to an increasing availability of single-cell time course data. When the number of observed cells is small, standard moment-based methods break down because they rely on the central limit theorem to compute a likelihood function for sample moments.[19–21] In such a scenario, the inference based on the path likelihood of individual trajectories provides a principled way to extract more information from the data. Unfortunately, these approaches are computationally challenging since they require integrating the CME multiple times for different parameter configurations. More effective approaches can be obtained by approximating the likelihood, e.g., by Gaussian moment closure[22] or the linear noise approximation.[23,24] The underlying approximations are not applicable to systems with a low copy number of species, such as single genes, while reducing the computational demand significantly. Alternatively, approximate likelihoods can be computed via particle filtering.[25] For Bayesian parameter inference, the approximate likelihood expression is typically used within a Markov chain Monte Carlo (MCMC) scheme. Alternatively, the parameters can be included into an augmented state space allowing for direct estimation via sequential Monte Carlo.[26]

In this work, we suggest a framework for performing Bayesian inference tasks for the parameter-dependent CME by exploiting the so-called tensor-train (TT) decomposition[27] to approximate the joint distribution over the CME states and parameters. For that purpose, we construct an explicit representation of the evolution operator in the TT format and show that it can be constructed without ever assembling the corresponding matrix. The TT format has the advantage that the storage requirement scales linearly with respect to the number of dimensions while being a numerically robust tensor decomposition.[27,28] We also develop a time-domain solver based on the time-dependent alternating minimal energy (tAMEn) algorithm,[29] which additionally incorporates parameter dependence. To that end, we combine the state space and the parameter space into a higher-dimensional tensor product space. The parameter dependence is expressed by means of a B-spline basis expansion, and a Galerkin formulation is employed to derive the multilinear system with respect to the full tensor. Since typically every reaction is governed by an individual rate constant, the parameters can be seamlessly included in the tensor representation, thus allowing for efficiently solving the joint system. In practice, however, the system parameters are often unknown. Therefore, we develop a framework for filtering, smoothing, and parameter inference based on the efficient TT representation of the joint system. The proposed framework allows us to perform Bayesian inference for the model parameters with a single forward–backward pass, as demonstrated on several synthetic examples.

The remaining of this paper is organized as follows: In Sec. II, we recall the CME and explain how it can be expressed in the tensor format. Next, in Sec. III, we present the TT decomposition, apply it to the tensor-formatted CME, and present a TT-based solution method. In Sec. IV, we consider the setting of a CME with parameter dependencies, which we include in the TT-based CME format. Subsequently, we exploit the TT format of the parameter-dependent CME for inference tasks, such as filtering, smoothing, and parameter identification. Numerical results are presented in Sec. V, where we validate the TT-based CME solver and showcase the benefits of performing inference in the TT format. This paper closes with our conclusions, presented in Sec. VI.

## II. CHEMICAL MASTER EQUATION IN TENSOR FORMAT

We consider a well-mixed reaction system with $d$ chemical species denoted with $\{S_1, \ldots, S_d\}$ involved in $M$ reactions. We consider reactions of type

$$q_{m,1}S_1 + \cdots + q_{m,d}S_d \to s_{m,1}S_1 + \cdots + s_{m,d}S_d, \tag{1}$$

with $m = 1, \ldots, M$ and $q_{m,k}, s_{m,k} \in \mathbb{N}_0$, $k = 1, \ldots, d$. The state of the system at time $t$ is described by the vector $\boldsymbol{x} \in \mathbb{N}_0^d$, which contains the number of elements per species at that time instant. To describe the change in the state vector after reaction $m$ occurs, we introduce the stoichiometric change vector $\boldsymbol{v}^{(m)} \in \mathbb{Z}^d$, the $k$th element of which is given as $v_k^{(m)} = s_{m,k} - q_{m,k}$. The change in the state vector due to the $m$th reaction is then given as $\boldsymbol{x} \to \boldsymbol{x} + \boldsymbol{v}^{(m)}$.

Assuming a stochastic model of the system, the state vector $\boldsymbol{x}$ is a realization of a continuous-time jump process $\{\boldsymbol{X}(t)\}_{t\geq 0}$. Then, the evolution of the time-dependent PMF

$$\begin{aligned} p(t, \boldsymbol{x}) = p_t(\boldsymbol{x}) &= \Pr(\boldsymbol{X}(t) = \boldsymbol{x}) \\ &= \Pr(X_1(t) = x_1, \ldots, X_d(t) = x_d) \end{aligned} \tag{2}$$

is described by the so-called chemical master equation (CME)[1] such that

$$\frac{\mathrm{d}p_t(\boldsymbol{x})}{\mathrm{d}t} = \sum_{m=1}^{M} \left\{ \alpha_m(\boldsymbol{x} - \boldsymbol{v}^{(m)}) p_t(\boldsymbol{x} - \boldsymbol{v}^{(m)}) - \alpha_m(\boldsymbol{x}) p_t(\boldsymbol{x}) \right\}, \tag{3a}$$

$$p_0(\boldsymbol{x}) = P^{(0)}(\boldsymbol{x}), \tag{3b}$$

where $P^{(0)}$ is the initial probability and $\alpha_m$ is called propensity function. For a well-mixed system at thermal equilibrium, the mass-action propensity reads

$$\alpha_m(\boldsymbol{x}) = c_m \prod_{k=1}^{d} \frac{x_k!}{q_{m,k}!(x_k - q_{m,k})!}, \tag{4}$$

where $c_m$ is the so-called specific probability rate, which is a measure for the probability that reaction $m$ occurs.

Until now, the CME has been defined for an infinite state space $\mathbb{N}_0^d$, which is computationally intractable, thus inappropriate for a

© Author(s) 2021

numerical solution. To that end, a truncation of the state space is necessary such that $x_k < n_k$, $k = 1, \dots, d$. We denote the truncated state space as $\mathcal{X} = \left\{\boldsymbol{x} \in \mathbb{N}_0^d \,|\, x_k < n_k, k = 1, \dots, \mathrm{d}\right\}$. The choice of a box domain truncation is not strictly necessary; however, it is beneficial for the TT format used in the following. All states in $\mathcal{X}$ can be uniquely indexed as $\boldsymbol{x}(\boldsymbol{i})$, where $\boldsymbol{i} = (i_1, \dots, i_d) \in \mathbb{N}^d$ and $x_k(i_k) = i_k - 1$. Accordingly, $i_k = 1, \dots, n_k$.

Using the truncated state space defined above and for a given time instance $t$, the PMF $p(t, \boldsymbol{x})$ can be represented as a multidimensional array $\mathbf{p}(t) \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, the elements of which are given as

$$\mathbf{p}_{\boldsymbol{i}}(t) = p(t, \boldsymbol{x}(\boldsymbol{i})). \tag{5}$$

In the following, we shall refer to such multidimensional arrays as *tensors*.[28] The evolution equation (3a) can then be written in the tensor format such that

$$\frac{\mathrm{d}\mathbf{p}(t)}{\mathrm{d}t} = \mathbf{A}\mathbf{p}(t), \tag{6}$$

where $\mathbf{A} \in \mathbb{R}^{(n_1 \times \cdots \times n_d) \times (n_1 \times \cdots \times n_d)}$ is a tensor-operator, also called a tensor-matrix, that acts on the tensor $\mathbf{p}(t)$. Tensor-operators can be seen as generalizations of the commonly employed matrix-based operators to more than two dimensions. The elements of the CME tensor-operator are given as

$$\mathbf{A}_{\boldsymbol{i},\boldsymbol{j}} = \sum_{m=1}^{M} \alpha_m(\boldsymbol{x}(\boldsymbol{i}) - \boldsymbol{v}^{(m)}) \delta^{\boldsymbol{x}(\boldsymbol{j})}_{\boldsymbol{x}(\boldsymbol{i}) - \boldsymbol{v}^{(m)}} - \alpha_m(\boldsymbol{x}(\boldsymbol{i})) \delta^{\boldsymbol{x}(\boldsymbol{j})}_{\boldsymbol{x}(\boldsymbol{i})}, \tag{7}$$

where $\delta^{\boldsymbol{j}}_{\boldsymbol{i}} = \delta^{j_1}_{i_1} \cdots \delta^{j_d}_{i_d}$, with $\delta^{j_k}_{i_k}$ denoting the Kronecker delta. Accordingly, the product between a tensor-operator and a tensor, the result of which is elementwise given as

$$(\mathbf{A}\mathbf{p}(t))_{\boldsymbol{i}} = \sum_{\boldsymbol{j}} \mathbf{A}_{\boldsymbol{i},\boldsymbol{j}} \mathbf{p}_{\boldsymbol{j}}(t), \tag{8}$$

can be seen as a generalization of the standard matrix-vector product.

The complexity for storing for storing the tensor $\mathbf{p}(t)$, which contains all state probabilities at time instance $t$, is $\mathcal{O}(n^d)$, where $n = \max_k\{n_k\}$. Therefore, even if a truncated state space is employed, the storage needs can become intractable even for a relatively small number of species. The exponential dependence of storage needs to the number of species is one manifestation of the so-called curse of dimensionality.[30] As a remedy to this problem, low-rank tensor formats[28] can be employed, such as the TT decomposition[27] discussed next.

In the following, a commonly employed operation between tensors, tensor-operators, or matrices is the Kronecker product. The Kronecker product between two tensors $\mathbf{x} \in \mathbb{R}^{n_1 \times \cdots \times n_p}$ and $\mathbf{y} \in \mathbb{R}^{m_1 \times \cdots \times m_q}$ is defined elementwise as

$$(\mathbf{x} \otimes \mathbf{y})_{\boldsymbol{ij}} = \mathbf{x}_{\boldsymbol{i}} \mathbf{y}_{\boldsymbol{j}}. \tag{9}$$

The definition holds also for matrices and vectors, as they can be interpreted as two-dimensional and one-dimensional tensors, respectively.

## III. SOLVING THE CHEMICAL MASTER EQUATION IN THE TENSOR-TRAIN FORMAT

### A. Tensor-train decomposition

As discussed in Sec. II, the size of a tensor scales exponentially with the number of dimensions, equivalently, number of species in this work. To mitigate the curse of dimensionality, we employ tensor decompositions, resulting in tensor formats whose sizes scale linearly with the number of dimensions, instead of exponentially. Several tensor decompositions have been developed over the last decades, resulting in better-scaling tensor formats.[28] In this work, we focus on the so-called tensor-train (TT) decomposition, which combines linear complexity scaling with respect to the dimensions and computational stability.[27]

A tensor $\mathbf{x} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is said to be in the TT format if it can be elementwise written as

$$\mathbf{x}_{\boldsymbol{i}} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_{d-1}=1}^{R_{d-1}} \mathbf{g}^{(1)}_{1 i_1 r_1} \mathbf{g}^{(2)}_{r_1 i_2 r_2} \cdots \mathbf{g}^{(d)}_{r_{d-1} i_d 1}, \tag{10}$$

where the three-dimensional tensors $\mathbf{g}^{(k)} \in \mathbb{R}^{R_{k-1} \times n_k \times R_k}$ are called the TT-cores and $\boldsymbol{R} = (1, R_1, \dots, R_{d-1}, 1)$ are called the TT-ranks. The storage complexity of a tensor in the TT format is reduced to $\mathcal{O}(NR^2 d)$, i.e., it is linear with respect to the number of dimensions $d$. Moreover, all basic multilinear algebraic operations scale also linearly with the dimensions and polynomially with the TT-ranks.[27] It should be noted that the TT-ranks grow after a multilinear algebraic operation is performed in the TT format. Therefore, a rank-reduction procedure is performed after the operation, called rounding.[27] Rounding decreases the TT-rank of the tensor while maintaining a prescribed accuracy $\epsilon$, and its complexity is $\mathcal{O}(R^3 Nd)$.

An exact TT decomposition of a full tensor typically leads to high ranks $\boldsymbol{R}$, hence, to high storage needs and computational costs as well. However, in many cases, using a low-rank TT approximation $\widetilde{\mathbf{A}} \approx \mathbf{A}$ is sufficient. If the full tensor is available, a low-rank TT approximation can be efficiently computed with $d$ sequential singular value decompositions (SVDs) of auxiliary matrices $\boldsymbol{A}^{(k)} \in \mathbb{R}^{n_k \times (n_1 n_2 \cdots n_{k-1} n_{k+1} \cdots n_d)}$.[27] For tensors defined implicitly by multidimensional functions, e.g., similar to (5), efficient interpolation-based TT approximation methods are available,[31,32] in which case the assembly of the full tensor is not necessary.

A tensor-operator $\mathbf{A} \in \mathbb{R}^{(n_1 \times \cdots \times n_d) \times (m_1 \times \cdots \times m_d)}$ can be similarly decomposed using the TT format, in which case it is elementwise written as

$$\mathbf{A}_{\boldsymbol{i},\boldsymbol{j}} = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_{d-1}=1}^{R_{d-1}} \mathbf{g}^{(1)}_{1 i_1 j_1 r_1} \mathbf{g}^{(2)}_{r_1 i_2 j_2 r_2} \cdots \mathbf{g}^{(d)}_{r_{d-1} i_d j_d 1}, \tag{11}$$

where the TT-cores $\mathbf{g}^{(k)} \in \mathbb{R}^{R_{k-1} \times n_k \times m_k \times R_k}$ are now four-dimensional. The product between a tensor-operator and a tensor, e.g., the Kronecker product defined in (8), can be efficiently computed directly in the TT format.[27]

### B. Linear system solutions in the TT format

Of particular interest in the context of this work is to solve efficiently a multilinear system $\mathbf{A}\mathbf{x} = \mathbf{b}$, where

© Author(s) 2021

$\mathbf{A} \in \mathbb{R}^{(n_1\times\cdots\times n_d)\times(n_1\times\cdots\times n_d)}$ and $\mathbf{x}, \mathbf{b} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ are given in the TT format. Iterative Krylov subspace solvers, e.g., based on the conjugate gradient (CG) or generalized minimal residual (GMRES) methods, can be generalized to multilinear systems given in the TT format.[33] However, without preconditioning, the number of iterations is large and leads to a large number of rounding operations in order to keep the TT-rank small, thus resulting in an undesirable computational cost.[33,34]

An alternative approach is to minimize the norm of the system's residual with respect to the cores of the solution's TT decomposition. The corresponding minimization problem reads

$$\min_{\mathbf{x}\in\mathbb{R}^{n_1\times\cdots\times n_d}} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_{\mathrm{F}}^2, \tag{12}$$

where $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm. The feasible set $\mathbb{R}^{n_1\times\cdots\times n_d}$ is restricted to a subset of $\mathbb{R}^{n_1\times\cdots\times n_d}$, which contains all $d$-dimensional tensors that can be represented in the TT format with TT-ranks $R_k \le R_{\max}$, $k = 1,\ldots,d-1$. In this case, the minimization problem is nonlinear with respect to the core tensors $\mathbf{g}^{(k)}$ of the TT representation of $\mathbf{x}$ given in (10).

The nonlinear minimization problem (12) can be solved using the alternating least squares (ALS) method.[35,36] The method fixes all cores but one, thus resulting in a quadratic optimization problem for the minimizing core. The process is then repeated iteratively, minimizing one core at a time until the objective function decreases to a sufficiently small value.[34] The main drawback of the ALS method is that the TT-ranks must be given *a priori*. This can be avoided by minimizing over two consecutive cores instead of just one, an idea that was first introduced by the Density Matrix Renormalization Group (DMRG) algorithm,[37] which was later used in the Alternating Minimal Energy (AMEn) method for solving high-dimensional multilinear systems.[29,38] In order to bring the minimization problem to a quadratic form, a so-called supercore $\widetilde{\mathbf{g}}^{(k,k+1)} \in \mathbb{R}^{R_{k-1}\times n_k\times n_{k+1}\times R_{k+1}}$ is first defined as

$$\widetilde{\mathbf{g}}^{(k,k+1)} = \mathbf{g}^{(k)}\mathbf{g}^{(k+1)}, \tag{13}$$

thus removing all information regarding the rank $R_k$. Then, the TT representation of $\mathbf{x}$ takes the following form (in an elementwise notation):

$$\mathbf{x}_{\boldsymbol{i}} = \sum_{r_i, i\neq k} \mathbf{g}^{(1)}_{1 i_1 r_1} \cdots \widetilde{\mathbf{g}}^{(k,k+1)}_{r_{k-1} i_k i_{k+1} r_{k+1}} \cdots \mathbf{g}^{(d)}_{r_{d-1} i_d 1}. \tag{14}$$

The minimization then proceeds similar to the ALS method, where now one supercore is minimized in each iteration. After the optimization procedure is completed, the supercore is divided into two separate TT-cores, e.g., by means of SVDs of auxiliary matrices.[31] Then, the rank $R_k$ is not *a priori* given, but identified as part of the supercore's separation.

### C. Low-rank TT representation of the CME operator

The CME operator in (7) can be represented in the TT format using a sum of rank-**1** tensors, without ever assembling the full tensor-operator.[10,39] For a reaction of type (1) with the propensity function (4), one can use the separation

$$\alpha_m(\boldsymbol{x}) = c_m f_1^{(m)}(x_1)\cdots f_d^{(m)}(x_d), \tag{15a}$$

$$f_k^{(m)}(x_k) = \frac{x_k!}{q_{m,k}!(x_k - q_{m,k})!}. \tag{15b}$$

Then, the CME tensor-operator defined in (7) can be written as the difference between two tensor-matrices $\mathbf{B}, \mathbf{C} \in \mathbb{R}^{(n_1\times\cdots\times n_d)\times(n_1\times\cdots\times n_d)}$, i.e.,

$$\mathbf{A}_{\boldsymbol{i},\boldsymbol{j}} = \mathbf{B}_{\boldsymbol{i},\boldsymbol{j}} - \mathbf{C}_{\boldsymbol{i},\boldsymbol{j}}, \tag{16}$$

both of which admit exact rank-**1** TT decompositions, with the corresponding four-dimensional TT-cores $\mathbf{g}^{(k)}$ (for $\mathbf{B}$) and $\mathbf{h}^{(k)}$ (for $\mathbf{C}$) given by

$$\mathbf{g}^{(k)}_{1 i_k j_k 1} = f_k^{(m)}(x_k^{(i_k)})\delta_{i_k}^{j_k}\mathbb{I}_{n_k}(x_k(j_k) + v_k^{(m)}), \tag{17a}$$

$$\mathbf{h}^{(k)}_{1 i_k j_k 1} = f_k^{(m)}(x_k^{(j_k)})\delta_{i_k - v_k}^{j_k}\mathbb{I}_{n_k}(x_k(j_k) + v_k^{(m)}), \tag{17b}$$

where the indicator functions $\mathbb{I}_{n_k}$ are defined as $\mathbb{I}_{n_k}(x) = 1$ if $x \le n_k$ and 0 otherwise. Then, for the $m$th reaction, the maximum TT-rank of the tensor-operator $\mathbf{A}$ is at most equal to 2. If the operators coming from $M$ different reactions are added, the maximum TT-rank is at most $2M$. However, in practical examples, rank rounding with a very high accuracy, e.g., $\|\widetilde{\mathbf{A}} - \mathbf{A}\|_{\mathrm{F}} \approx 10^{-12}\|\mathbf{A}\|_{\mathrm{F}}$, yields a TT approximation $\widetilde{\mathbf{A}} \approx \mathbf{A}$ with much lower TT-ranks.

### D. Solving the CME in the TT format

The CME can be solved numerically in the TT format using finite differences.[40] This limits the choice of the time discretization to classical implicit and explicit schemes.[29] An alternative method[10,29] is to employ a basis representation of the time-dependent solution over an interval $[0, \Delta T]$ such that

$$p(t, \boldsymbol{x}(\boldsymbol{i})) = \sum_{j=1}^{T} \mathbf{p}_{\boldsymbol{i}j} b_j(t), \tag{18}$$

where $b_j(t)$ are basis functions for the interpolation in time. In this work, we employ a Chebyshev basis; however, other options are also possible, e.g., hat functions or Lagrange polynomials.

By including time as an additional dimension next to the states, a $(d+1)$-dimensional tensor is obtained. We then perform a Galerkin projection to recover the degrees of freedom for the entire subinterval by solving the system

$$\mathbf{M}\mathbf{p} = \mathbf{f}, \tag{19a}$$

$$\mathbf{M} = \mathbf{I}_{\boldsymbol{n}} \otimes (S + V) - (\mathbf{I}_{\boldsymbol{n}} \otimes P)(\mathbf{A} \otimes \boldsymbol{I}_T), \tag{19b}$$

$$\mathbf{f} = \mathbf{p}^{(0)} \otimes \boldsymbol{v}, \tag{19c}$$

where $\boldsymbol{S}$ is the stiffness matrix, $\boldsymbol{P}$ is the mass matrix,[29,41] $\boldsymbol{I}_N \in \mathbb{R}^{N\times N}$ denotes an identity matrix, and $\mathbf{I}_{\boldsymbol{n}} = \boldsymbol{I}_{n_1} \otimes \cdots \otimes \boldsymbol{I}_{n_d}$. System (19a) can then be solved, as described in Sec. III B.

Due to the increased complexity of the solution over long simulation times, one can divide the time domain and apply the presented

© Author(s) 2021

method to the individual subdomains by taking the end state as the initial condition for the next subinterval. For a constant subinterval length and if the solution is smooth, the convergence is exponential in $T$.[29,41] An error indicator is represented by computing the norm of the residual of (19a) for an enriched basis,[29]

$$\varepsilon(T,\Delta t) = \|\mathbf{M}'\mathbf{Q}\mathbf{p} - \mathbf{f}'\|_F, \tag{20}$$

where $\mathbf{M}'$ and $\mathbf{f}'$ are the tensors from (19b) and (19c) constructed for $T' = 2T$. The operator $\mathbf{Q}$ interpolates the solution on the finer time-domain basis with $T' = 2T$. This can be used to adapt the subinterval length if the indicator $\epsilon(T,\Delta t)$ is larger than a prescribed tolerance (tol),[29]

$$\Delta t' = \left(\frac{\text{tol}}{\varepsilon(T,\Delta t)}\right)^{\frac{1}{T}} \Delta t, \tag{21}$$

where $\Delta t'$ is the modified subinterval length. One bottleneck is the prescribed accuracy of the TT-solver, which, as shown in the numerical results, acts like a lower limit for the error.

In the same framework, one can also include classical implicit time-stepping schemes, such as the Crank–Nicolson and the implicit Euler, by appropriately choosing the stiffness and mass matrices.[29] The motivation behind this is that the time dynamics is captured in the low rank structure of the tensor, reducing the storage complexity. Moreover, as observed from numerical experiments, the runtime is also reduced.

### E. Quantized TT CME solver

One way to optimize the aforementioned TT-based CME solver is to employ the so-called quantized tensor-train (QTT) decomposition.[42] The QTT format has proven to further increase the storage and computational efficiency of the TT representation by reshaping the tensors into higher-dimensional ones while reducing the mode sizes. Let $\mathbf{x} \in \mathbb{R}^{n_1 \times n_2 \times \cdots \times n_d}$ be a tensor with $\log_2 n_k \in \mathbb{N}, k = 1,\ldots,d$, i.e., with mode sizes that are powers of 2. If the tensor $\mathbf{x}$ is represented in the TT format, then reshaping it into a $(\sum_k \log_2 n_k)$-dimensional tensor can be easily achieved by performing the TT decomposition on the individually reshaped cores.

Let $\mathbf{x} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ be a tensor with $\log_2 n_k \in \mathbb{N}$. The tensor admits a rank $\boldsymbol{R}$ TT decomposition with the cores $\mathbf{g}^{(k)}$. The quantization process implies reshaping the individual cores to tensors of shape $r_{k-1} \times 2 \times \cdots \times 2 \times r_k$ and then the TT decomposition of the reshaped cores is computed. The resulting cores correspond to the QTT decomposition of the tensor.[32] In the case of the tensor-matrices, the procedure is similar. Compared to the computational complexity of the solver, the complexity of the transformation between TT and QTT can be neglected. This procedure has been proven to be effective for reducing the storage requirements and the computation time for solving the CME in the TT format.[10,11] If the ranks of the QTT decomposition remain bounded, the storage complexity is $\mathcal{O}(d \log_2 N)$.[43] Moreover, the solver benefits from the linearity with respect to the number of dimensions.

## IV. BAYESIAN INFERENCE FOR THE CHEMICAL MASTER EQUATION WITH PARAMETER DEPENDENCIES

### A. Parameter-dependent CME

We now consider a parameter-dependent CME,[10] described by

$$\frac{\mathrm{d}\mathbf{p}(\boldsymbol{\theta})}{\mathrm{d}t} = \mathbf{A}(\boldsymbol{\theta})\mathbf{p}(\boldsymbol{\theta}), \tag{22}$$

where $\boldsymbol{\theta} \in \mathbb{R}^{n_p}$ denotes the parameter vector. The parameter vector $\boldsymbol{\theta}$ is assumed to take values in the tensor product space $\mathcal{P} = [\theta_1^{\min}, \theta_1^{\max}] \times \cdots \times [\theta_{n_p}^{\min}, \theta_{n_p}^{\max}]$.

Solving the CME for one parameter realization $\boldsymbol{\theta}^{(\boldsymbol{l})} \in \Theta$, $\boldsymbol{l} = (l_1,\ldots,l_{n_p})$, yields the conditional PMF $p_t\left(\boldsymbol{x}|\boldsymbol{\theta}^{(\boldsymbol{l})}\right)$. If instead of a fixed initial PMF, one starts with the joint PMF $p_{t_0}(\boldsymbol{x},\boldsymbol{\theta})$, the solution will be the joint PMF over the discretized time interval. Since our goal is to compute the conditional/joint PMF over the entire parameter space, we use a basis expansion[44] to describe the parameter dependence such that

$$p_t(\boldsymbol{x}(\boldsymbol{i}),\boldsymbol{\theta}) \approx \sum_j \sum_{\boldsymbol{l}} \mathbf{p}_{\boldsymbol{i}\boldsymbol{l}j}(t) L_{\boldsymbol{l}}(\boldsymbol{\theta}), \tag{23}$$

where $\mathbf{p} \in \mathbb{R}^{n_1 \times \cdots \times n_d \times \ell_1 \times \cdots \times \ell_{n_p}}$ and $\{L_{\boldsymbol{l}}\}_{\boldsymbol{l}=\mathbf{1}}^{\boldsymbol{\ell}}$ is a tensor product basis $L_{\boldsymbol{l}}(\boldsymbol{\theta}) = L^{(1)}(\theta_1)\cdots L^{(n_p)}(\theta_{n_p})$.

In this work, B-spline basis functions are chosen for the parameter dependence.[45]

In order to retrieve the degrees of freedom, a Galerkin formulation is used to derive a multilinear system with respect to the full tensor. We choose the test functions $\mathbf{q}(\boldsymbol{\theta})$ from the same space, which yield the formulation

$$\left\langle \frac{\mathrm{d}\mathbf{p}(\boldsymbol{\theta})}{\mathrm{d}t}, \mathbf{q} \right\rangle = \langle \mathbf{A}\mathbf{p}, \mathbf{q} \rangle, \tag{24}$$

where $\langle\cdot,\cdot\rangle$ is an inner product with respect to $\boldsymbol{\theta}$. One can then derive the multilinear system

$$\mathbf{M}\frac{\mathrm{d}\mathbf{p}(\theta)}{\mathrm{d}t} = \mathbf{K}\mathbf{p}, \tag{25}$$

with the mass tensor-matrix

$$\mathbf{M}_{\boldsymbol{mn},\boldsymbol{il}} = \delta_{\boldsymbol{m}}^{\boldsymbol{i}} \int_{\mathcal{P}} L_{\boldsymbol{n}}(\boldsymbol{\theta}) L_{\boldsymbol{l}}(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta} \tag{26}$$

and the stiffness tensor-matrix

$$\mathbf{K}_{\boldsymbol{mn},\boldsymbol{il}} = \int_{\mathcal{P}} \mathbf{A}_{\boldsymbol{m},\boldsymbol{i}}(\boldsymbol{\theta}) L_{\boldsymbol{n}}(\boldsymbol{\theta}) L_{\boldsymbol{l}}(\boldsymbol{\theta}) \mathrm{d}\boldsymbol{\theta}. \tag{27}$$

The mass matrix can be easily constructed as a rank-$\mathbf{1}$ TT-operator using the individual mass matrices of the univariate bases. On the contrary, the stiffness matrix requires the evaluation of the parameter-dependent CME operator over a tensor product quadrature grid $\Theta = \left\{\theta_1^{(r_1)}\right\}_{r_1=1}^{\ell_1} \times \left\{\theta_2^{(r_2)}\right\}_{r_2=1}^{\ell_2} \times \cdots \times \left\{\theta_{n_p}^{(r_{n_p})}\right\}_{r_{n_p}=1}^{\ell_{n_p}}$ such that

© Author(s) 2021

$$\mathbf{K}_{mn,il} \approx \sum_r \mathbf{w}_r \bar{\mathbf{A}}_{mr,ir} L_n(\boldsymbol{\theta}^{(r)}) L_l(\boldsymbol{\theta}^{(r)}), \tag{28}$$

where $\mathbf{w}$ is the weight tensor and $\bar{\mathbf{A}}_{ik,jr} = \mathbf{A}\left(\boldsymbol{\theta}^{(r)}\right)\delta_r^k$. The tensor-matrix $\mathbf{K}$ admits a TT representation or approximation, assuming that the evaluation of the CME operator can also be represented or approximated in the TT format.

A direct construction of the extended operator $\bar{\mathbf{A}}$ can be easily accomplished if each parameter affects only one reaction. For example, this situation occurs if the parameters are the reaction rates, i.e., $\boldsymbol{\theta} = (c_1, c_2, \ldots)$. Then, the operator is extended using the Kronecker product such that

$$\begin{aligned}\bar{\mathbf{A}} = \mathbf{A}^{(1)} &\otimes \left(\operatorname{diag}\left(\theta_1^{(1)}, \ldots, \theta_1^{(\ell_1)}\right) \otimes \mathbf{I}_{\ell_2} \otimes \mathbf{I}_{\ell_3} \otimes \cdots\right) \\ &+ \mathbf{A}^{(2)} \otimes \left(\mathbf{I}_{\ell_1} \otimes \operatorname{diag}\left(\theta_2^{(1)}, \ldots, \theta_2^{(\ell_2)}\right) \otimes \mathbf{I}_{\ell_3} \otimes \cdots\right) + \cdots,\end{aligned} \tag{29}$$

where $\mathbf{A}^{(m)}$ is the CME operator corresponding to reaction $m$ for a unity reaction rate.

We note that the parameter dependence is not necessarily restricted to the reaction rates. Equation (29) can be extended to accommodate other types of parameter dependencies. If the propensity functions have a representation as in (15b) and every $\alpha_m$ depends on at most one parameter, then the individual CME operator for every reaction evaluated on the grid $\Theta$ can be expressed as a sum of rank-$\mathbf{1}$ TT tensors and then rounded to eliminate overshooting ranks. Moreover, even the structure of the reaction network can be incorporated as a parameter; however, this is out of scope for the present work.

## B. Filtering and smoothing in the TT format

One relevant inference task in computational biology applications is the filtering and smoothing of observations, for example, in order to estimate the dynamics of genes that are measured indirectly via a fluorescent reporter protein. We consider $N_\text{o}$ state observations $\left\{\boldsymbol{y}^{(j)}\right\}_{j=1}^{N_\text{o}}$, which are sampled at discrete time steps $\{t_j\}_{j=1}^{N_\text{o}}$. The observations are considered to be realizations of the random variables $\left\{\boldsymbol{Y}^{(j)}\right\}_{j=1}^{N_\text{o}}$, which are assumed to be conditionally independent, given the latent states $\{\boldsymbol{X}(t_j)\}_{j=1}^{N_\text{o}}$. Thus, the observation model is assumed to be dependent only on the current state. Its probability density function (PDF) is denoted with $p_{Y|X}(\boldsymbol{y}|\boldsymbol{x})$. In practice, $p_{Y|X}$ often corresponds to additive Gaussian or multiplicative log-normal noise. However, the presented framework is not limited to these particular cases.

The conditional probability $\Pr(\boldsymbol{X}(t) = \boldsymbol{x}|\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(j)})$ for $j = \max\{k \in \mathbb{N}|t_k < t\}$ satisfies the unconditional master equation[46]

$$\frac{\mathrm{d}p_t(\boldsymbol{x})}{\mathrm{d}t} = \sum_{m=1}^{M}\left\{\alpha_m(\boldsymbol{x} - \boldsymbol{\nu}^{(m)})p_t(\boldsymbol{x} - \boldsymbol{\nu}^{(m)}) - \alpha_m(\boldsymbol{x})p_t(\boldsymbol{x})\right\}, \tag{30a}$$

with the reset conditions

$$p_{t_j}(\boldsymbol{x}) = \frac{1}{Z_j} p_{t_j^-}(\boldsymbol{x}) p_{Y|X}\left(\boldsymbol{y}^{(j)}|\boldsymbol{x}\right), \tag{30b}$$

where $p_{t_j^-}(\boldsymbol{x})$ represents the left approaching limit $t \to t_j, t < t_j$, and $Z_i = \sum_{\boldsymbol{x}} p_{t_j^-}(\boldsymbol{x}) p_{Y|X}\left(\boldsymbol{y}^{(j)}|\boldsymbol{x}\right)$. If all observations are taken into consideration, we deal with the smoothing case. The PMF $\tilde{p}_t(\boldsymbol{x})$ of $\Pr(\boldsymbol{X}(t) = \boldsymbol{x}|\boldsymbol{y}^{(1)}, \ldots, \boldsymbol{y}^{(N_\text{o})})$ can be factorized as

$$\tilde{p}_t(\boldsymbol{x}) = p_t(\boldsymbol{x})\beta_t(\boldsymbol{x}), \tag{31}$$

where the PMF satisfies the backward master equation

$$\frac{\mathrm{d}\beta_t(\boldsymbol{x})}{\mathrm{d}t} = \sum_{m=1}^{M}\left\{\alpha_m(\boldsymbol{x})\beta_t(\boldsymbol{x}) - \alpha_m(\boldsymbol{x})\beta_t(\boldsymbol{x} + \boldsymbol{\nu}^{(m)})\right\}, \tag{32a}$$

$$\beta_{t_j^-}(\boldsymbol{x}) = \frac{1}{Z_j} p_{t_j}(\boldsymbol{x}) p_{Y|X}(\boldsymbol{y}^{(j)}|\boldsymbol{x}), \tag{32b}$$

where $\beta(\boldsymbol{x}, t_{N_\text{o}}) = 1$ is the terminal condition and

$$\beta_t(\boldsymbol{x}) \propto p(\boldsymbol{y}^{(j)}, \ldots, \boldsymbol{y}^{(N_\text{o})}|\boldsymbol{X}(t) = \boldsymbol{x}), \tag{33}$$

where $j = \min\{k \in \mathbb{N}|t_k > t\}$. The PMF $\tilde{p}_t(\boldsymbol{x})$ satisfies the evolution equation

$$\frac{\mathrm{d}\tilde{p}_t(\boldsymbol{x})}{\mathrm{d}t} = \sum_{m=1}^{M}\left\{\tilde{\alpha}_m(\boldsymbol{x} - \boldsymbol{\nu}^{(m)}, t)\tilde{p}_t(\boldsymbol{x} - \boldsymbol{\nu}^{(m)}) - \tilde{\alpha}(\boldsymbol{x}, t)\tilde{p}_t(\boldsymbol{x})\right\}, \tag{34}$$

with the time-varying smoothing propensity functions

$$\tilde{\alpha}_m(\boldsymbol{x}, t) = \alpha_m(\boldsymbol{x})\frac{\beta(\boldsymbol{x} + \boldsymbol{\nu}^{(m)}, t)}{\beta(\boldsymbol{x}, t)}. \tag{35}$$

The method is also known in the literature as the forward–backward algorithm[47] and can be interpreted as a message-passing algorithm in a Hidden Markov Model (HMM). Moreover, it can be efficiently performed in the TT format, as shown in Algorithm 1. The PMF $p(\boldsymbol{x}^{(j)}|\boldsymbol{y}^{(0)}, \ldots, \boldsymbol{y}^{(j-1)})$ is the forward message and is denoted by the tensor $\mathbf{a}^{(j)} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$. The prediction step is performed by solving the CME over the interval $[t_{j-1}, t_j]$ with the initial condition $\mathbf{a}^{(j-1)}$. The observation is used to construct a tensor $\mathbf{p}^\text{obs} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ with $\mathbf{p}_{\boldsymbol{i}}^\text{obs} = p_{Y|X}\left(\boldsymbol{y}^{(j)}|\boldsymbol{x}(\boldsymbol{i})\right)$. If every species is observed independently, i.e., the observation model can be factorized, the tensor $\mathbf{p}^\text{obs}$ is rank-$\mathbf{1}$ and can be expressed as a Kronecker product. For the backward pass, the message is $p\left(\boldsymbol{y}^{(j+1)}, \ldots, \boldsymbol{y}^{(N_\text{o})}|\boldsymbol{x}^{(j)}\right)$ and is represented with the tensor $\mathbf{b}^{(j)}$. The CME is now solved using the transposed operator $\mathbf{A}^\top$ and with the initial condition $\mathbf{b}^{(j+1)} * \mathbf{p}^\text{obs}$, where $*$ denotes the elementwise multiplication operation. The last step is to multiply and normalize the forward and the backward messages in order to get the conditional $p\left(\boldsymbol{x}^{(j)}|\boldsymbol{y}^{(0)}, \ldots, \boldsymbol{y}^{(N_\text{o})}\right)$. In the presented framework, we a get smoother distribution only at the observation points. In order to have information in between the observations, prediction steps must be added.

© Author(s) 2021

ALGORITHM 1. Forward–backward algorithm in the TT format.

```
Input: Sample $\{y^{(j)}\}_{j=0}^{N_o}$, initial PMF $\mathbf{p}^{(0)}$
$\mathbf{a}^{(0)} \leftarrow \mathbf{p}^{(0)}$
for $j = 1, \ldots, N_o$ do
    Solve the CME with $\mathbf{a}^{(j-1)}$ as initial condition.
    Compute $\mathbf{p}^{\text{obs}}$ for $y^{(j)}$ in the TT format.
    $\mathbf{a}^{(j)} \leftarrow \mathbf{p}_i^{\text{obs}} * \mathbf{a}^{(j-1)}$
end for
$\mathbf{b}^{(N_o)} \leftarrow 1$
for $j = N_o - 1, \ldots, 0$ do
    Compute $\mathbf{p}^{\text{obs}}$ for $y^{(j+1)}$ in the TT format.
    Solve the CME with operator $\mathbf{A}^\top$ and initial condition $Z^{-1}\mathbf{b}^{(j+1)} * \mathbf{p}^{\text{obs}}$.
    $\mathbf{b}^{(j)} \leftarrow \mathbf{p}^{\text{obs}} * \mathbf{b}^{(j+1)}$
end for
for $j = 0, \ldots, N_o$ do
    $\mathbf{p}^{(j)} \leftarrow Z^{-1}\mathbf{a}^{(j)} * \mathbf{b}^{(j)}$
end for
Output: $\mathbf{p}^{(j)}$ for $j = 0, \ldots, N_o$
```

### C. Bayesian parameter inference in the TT format

Consider an observation sample $\left\{y^{(j)}\right\}_{j=1}^{N_o}$ satisfying the assumptions detailed in Sec. IV B, but now connected to a realization of the random process $X(t, \hat{\theta})$, where $\hat{\theta}$ is the parameter vector governing the system. Given a prior distribution $p(\theta)$, we are interested in computing the Bayesian parameter posterior $p(\theta | y^{(0)}, \ldots, y^{(N_o)})$. By viewing the parameters as part of an augmented process $\{\mathbf{X}(t), \theta(t)\}_{t \geq 0}$, the distribution of the parameters over the parameter space $\mathcal{P}$ can be obtained by performing filtering over the joint space of states and parameters. The prediction step is given by

$$p\left(x^{(j)}, \theta^{(j)} | y^{(0)}, \ldots, y^{(j-1)}\right) = \sum_{x^{(j-1)}} \int \left\{p_{j|j-1}\left(x^{(k)}, \theta^{(j)} | x^{(j-1)}, \theta^{(j-1)}\right) \times p\left(x^{(j-1)}, \theta^{(j-1)} | y^{(0)}, \ldots, y^{(j-1)}\right)\right\} d\theta^{(j-1)}, \tag{36}$$

where $p_{j|j-1}$ is the transition PDF and implies solving the parameter-dependent CME from $t_{j-1}$ to $t_j$. Next, the update step reads

$$p\left(x^{(j)}, \theta^{(j)} | y^{(0)}, \ldots, y^{(j)}\right) = \frac{1}{Z} p_{Y|X}\left(y^{(j)} | x\right) p\left(x^{(j)}, \theta^{(j)} | y^{(0)}, \ldots, y^{(j-1)}\right). \tag{37}$$

In the TT format, this parameter inference procedure can be implemented as follows: The posterior $p\left(x^{(j)}, \theta^{(j)} | y^{(0)}, \ldots, y^{(j)}\right)$ is represented by the tensor $\mathbf{p} \in \mathbb{R}^{n_1 \times \cdots \times n_d \times \ell_1 \times \cdots \times \ell_{n_p}}$ such that

$$p\left(x^{(j)}, \theta^{(j)} | y^{(0)}, \ldots, y^{(j)}\right) = \sum_{l} \mathbf{p}_{xl}^{(j)} L_l(\theta). \tag{38}$$

The prediction step involves solving the parameter-dependent CME with $p\left(x^{(j)}, \theta^{(j)} | y^{(0)}, \ldots, y^{(j)}\right)$ as the initial condition, returning

ALGORITHM 2. Parameter identification for the parameter-dependent CME in the TT format.

```
Input: Sample $\{y^{(j)}\}_{j=0}^{N_o}$, initial PMF $\mathbf{p}^{(0)}$, prior over the parameter space $\mathbf{p}^{\text{prior}}$
$\mathbf{p}^{(0)} \leftarrow \mathbf{p}^{(0)} * \mathbf{p}^{\text{prior}}$
for $j = 1, \ldots, N_o$ do
    Solve the CME with $\mathbf{p}^{(j-1)}$ as initial condition to obtain the solution $\mathbf{p}^{(j\to j+1)}$.
    Compute $\mathbf{p}^{\text{obs}}$ for $y^{(j)}$ in TT.
    $\mathbf{p}_{il}^{(j+1)} \leftarrow \mathbf{p}_i^{\text{obs}} \mathbf{p}_{il}^{(j\to j+1)}$
    $\mathbf{p}^{(j+1)} \leftarrow Z^{-1}\mathbf{p}^{(j+1)}$ for $Z = \sum_{il} \mathbf{p}_{il}^{(j+1)} \mathbf{w}_l$
end for
$\mathbf{p}_l^{\text{post}} \leftarrow Z^{-1} \sum_i \mathbf{p}_{il}^{(N_o)}$
Output: $\mathbf{p}^{\text{post}}$
```

© Author(s) 2021

the predicted PMF $\mathbf{p}^{(\mathrm{pred})}$ as a result. The resulting tensor is multiplied with the observation tensor at step $j+1$, and normalization is performed to get the new joint distribution

$$\mathbf{p}_{\boldsymbol{xl}}^{(j+1)} = Z^{-1}\mathbf{p}_{\boldsymbol{x}}^{\mathrm{obs}}\mathbf{p}_{\boldsymbol{xl}}^{(j\to j+1)}, \tag{39}$$

where $Z = \sum_{\boldsymbol{il}} \mathbf{p}_{\boldsymbol{il}}^{(j+1)}\mathbf{w}_{\boldsymbol{l}}$ is the normalization constant and comes from numerically integrating over the parameter space with the integration weight tensor $\mathbf{w}$. A prior distribution $p\left(\boldsymbol{\theta}^{(0)}\right)$ and an exact knowledge of the state for $j=0$ is used for the first step, where $p\left(\boldsymbol{x}^{(0)},\boldsymbol{\theta}^{(0)}|\boldsymbol{y}^{(0)}\right) = p\left(\boldsymbol{x}^{(0)}\right)p\left(\boldsymbol{\theta}^{(0)}\right)$. Once all observations have been used, the state is marginalized to obtain the posterior over the parameter space as

$$p\left(\boldsymbol{\theta}^{(j)}|\boldsymbol{y}^{(0)},\dots,\boldsymbol{y}^{(j)}\right) = \sum_{\boldsymbol{x}^{(j)}} p\left(\boldsymbol{x}^{(j)},\boldsymbol{\theta}^{(j)}|\boldsymbol{y}^{(0)},\dots,\boldsymbol{y}^{(j)}\right), \tag{40}$$

which is computationally efficient if performed in the TT format. The procedure is summarized in Algorithm 2.

## V. NUMERICAL RESULTS

The following numerical experiments aim to showcase the advantages of the proposed framework in terms of accuracy and computational efficiency. With respect to the latter, storage requirements and computation times are reported for every individual test case. All tests were run on a workstation with a 10-core Intel Xeon processor with 64 GB of RAM. For the TT operations, the *ttpy* Python package was used in combination with the Intel MKL library.

### A. Validation of the TT-based CME solver

#### 1. Two-dimensional simple gene expression model

We first validate the developed TT ODE solver and perform a convergence study based on the two-dimensional simple gene expression model.[48] The four reactions are presented in Table I. The initial state is $\boldsymbol{x}^{(0)} = (2,4)^\top$ with probability 1. The CME is solved in the time interval $[0,1024]$ with a subinterval size of 128, where arbitrary time units are used.

The first validation test concerns the maximum relative error of the solution to the CME, computed with the method presented in Sec. III D, in dependence to the time dimension $t$ of the basis representation from (18) inside one subinterval. The maximum relative error is given as $\max\left|p_{t_{\mathrm{end}}}^{(\mathrm{ref})}(\boldsymbol{x}) - p_{t_{\mathrm{end}}}(\boldsymbol{x})\right| / \max|p_{t_{\mathrm{end}}}(\boldsymbol{x})|$, where $t_{\mathrm{end}} = 1024$ and the reference solution $p_{t_{\mathrm{end}}}^{(\mathrm{ref})}(\boldsymbol{x})$ is computed by numerically solving the CME without the TT decomposition for a very fine time grid. We note that no truncation of the TT-rank was performed during this validation test, and the relative residual that signifies the convergence of the TT-solver was set to $10^{-13}$.

TABLE I. Reactions of the simple gene expression model.

| Reaction | $\alpha_m(\boldsymbol{x})$ | Rates $c_i$ | Description |
|---|---|---|---|
| mRNA → Ø | $c_1x_1$ | 0.002 | mRNA degradation |
| mRNA → mRNA + protein | $c_2x_1$ | 0.015 | Translation |
| Ø → mRNA | $c_3$ | 0.1 | Transcription |
| Protein → Ø | $c_4x_2$ | 0.01 | Protein degradation |

The results of this first validation set are presented in Fig. 1, where the employed time-interpolation on the Chebyshev polynomial basis, as shown in (18), is compared against classical time-stepping methods such as implicit Euler and Crank–Nicolson finite-difference schemes. As would be expected, irrespective of the time-stepping method, the TT-based CME solver yields increasingly more accurate results for finer discretizations of the time interval. Moreover, as expected from theory, the convergence of the explicit Euler scheme is $\mathcal{O}(\Delta t)$, accordingly $\mathcal{O}(\Delta t^2)$ for Crank–Nicolson.[49] In the case of the Chebyshev polynomials, an exponential convergence is observed and the error stagnates for a basis of only $T=8$ polynomials.

The combined impact of the time step and the maximum residual of the TT-solver is investigated next for the Chebyshev basis representation, with the results presented in Fig. 2. As can be observed, for a fixed $T$, the accuracy of the TT-solver's solution stagnates after a certain value of the maximum residual. The stagnation point is actually dependent on the value of $T$, i.e., smaller $T$ values allow for smaller maximum residuals. Hence, the step size and maximum residual must be chosen according to the desired accuracy of the TT-solution, also taking into consideration the related computational cost.

#### 2. Four-dimensional SEIR model

For the previously considered two-dimensional model, the reference solution could easily be obtained using an ODE solver. We now consider a four-dimensional virus spreading model, namely, the SEIR model,[50] in which case standard ODE solution methods result in high computational demands in terms of computation time and storage needs.

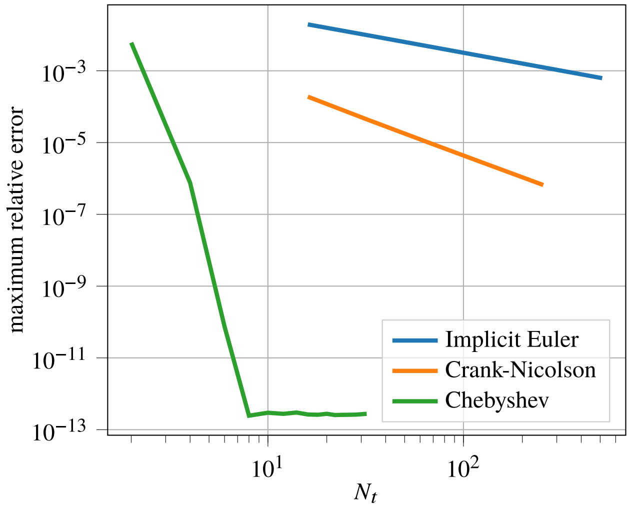


FIG. 1. Convergence of the TT-solver with respect to the dimension of the time basis.

© Author(s) 2021

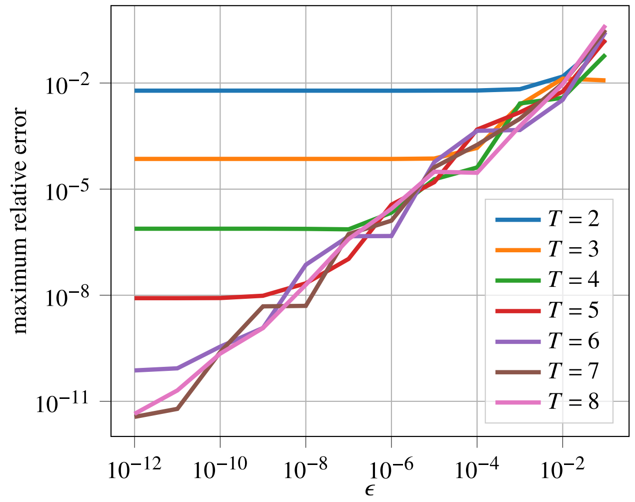


**FIG. 2.** Error vs solver accuracy for different sizes of the basis.

The individuals of the virus spreading model are separated into four distinct categories: (1) susceptible ($S$), i.e., individuals who may become infected; (2) exposed ($E$), i.e., infected individuals who are not yet contagious; (3) infected ($I$), i.e., infected individuals who are contagious; and (4) recovered ($R$), i.e., individuals with immunity to the virus. The interactions between the individuals are described by the reactions presented in Table II. The initial condition is $\boldsymbol{x}(0) = (50, 4, 0, 0)^\top$, and the state space is truncated to $\boldsymbol{n} = (n_1, n_2, n_3, n_4) = (128, 128, 64, 64)$. The simulation was performed over the interval $[0, 8]$. For the TT-solver, the subinterval length is equal to 0.5 and the subinterval basis dimension is $T = 8$. The reference solution is computed by numerically solving the CME without the TT decomposition for a very fine time grid.

Even if a sparse format is employed, ≈ 1.3 GB RAM is needed to store the CME operator for the reference solution. In comparison, using the TT format, the CME TT-operator has the TT-ranks $\boldsymbol{R} = (1, 5, 6, 3, 1)$, resulting in storage needs of only ≈ 2.32 MB RAM, i.e., 0.17% of the storage space needed by the standard solver. If the operator is reshaped in the QTT format, the storage requirements decreases to ≈ 42 KB. Moreover, using the solver with the QTT format, the solution is obtained in ≈ 180 s, which is a considerably smaller computation time than the one needed for the reference solution, i.e., ≈ 12 600 s. Without the use of the QTT format, the number of solver iterations and the computation time increase by one order of magnitude.

**TABLE II.** Reactions of the SEIR model.

| Reaction | $\alpha_m(\boldsymbol{x})$ | Rate $c_i$ | Description |
|---|---|---|---|
| $S + I \rightarrow E + I$ | $c_1 x_1 x_3$ | 0.1 | Susceptible becomes exposed |
| $E \rightarrow I$ | $c_2 x_2$ | 0.5 | Exposed becomes infected |
| $I \rightarrow S$ | $c_3 x_3$ | 1.0 | Infected recovers without immunity |
| $S \rightarrow \emptyset$ | $c_4 x_1$ | 0.01 | Susceptible dies |
| $E \rightarrow \emptyset$ | $c_5 x_2$ | 0.01 | Exposed dies |
| $I \rightarrow R$ | $c_6 x_3$ | 0.01 | Infected recovers with immunity |
| $\emptyset \rightarrow S$ | $c_7$ | 0.4 | New susceptible individuals arrive |

Figure 3 shows the time evolution of the marginal *EI* distribution, as well as the pointwise error at the end of the simulation compared to the reference marginals. Finally, at $t_{\mathrm{end}}$, we obtain the relative errors

$$\epsilon_{\max} = \frac{\max_{\boldsymbol{x}} |p^{(\mathrm{ref})}_{t_{\mathrm{end}}}(\boldsymbol{x}) - p_{t_{\mathrm{end}}}(\boldsymbol{x})|}{\max_{\boldsymbol{x}} |p^{(\mathrm{ref})}_{t_{\mathrm{end}}}(\boldsymbol{x})|} = 2.9 \cdot 10^{-5},$$

$$\epsilon_{\mathrm{mean}} = \frac{\frac{1}{N^4} \sum_{\boldsymbol{x}} |p^{(\mathrm{ref})}_{t_{\mathrm{end}}}(\boldsymbol{x}) - p_{t_{\mathrm{end}}}(\boldsymbol{x})|}{\max_{\boldsymbol{x}} |p^{(\mathrm{ref})}_{t_{\mathrm{end}}}(\boldsymbol{x})|} = 2.539 \cdot 10^{-9}.$$

Hence, the TT-solver yields accurate solutions at significantly reduced execution times and with tremendous storage savings compared to the standard solver. Indicatively, the solution at $t = 8$ requires only 2.5 MB storage, which is ~0.4% the storage requirement of the reference solution.

One issue is the ordering of the species. If species that are highly correlated are apart from each other in the train, the ranks in between must carry the information and therefore the overall rank structure increases. This can also be observed in the representation of the CME operator. In this example, the S, E, I, R ordering is chosen so that most of the reactions involve species that are neighbors in the chain.

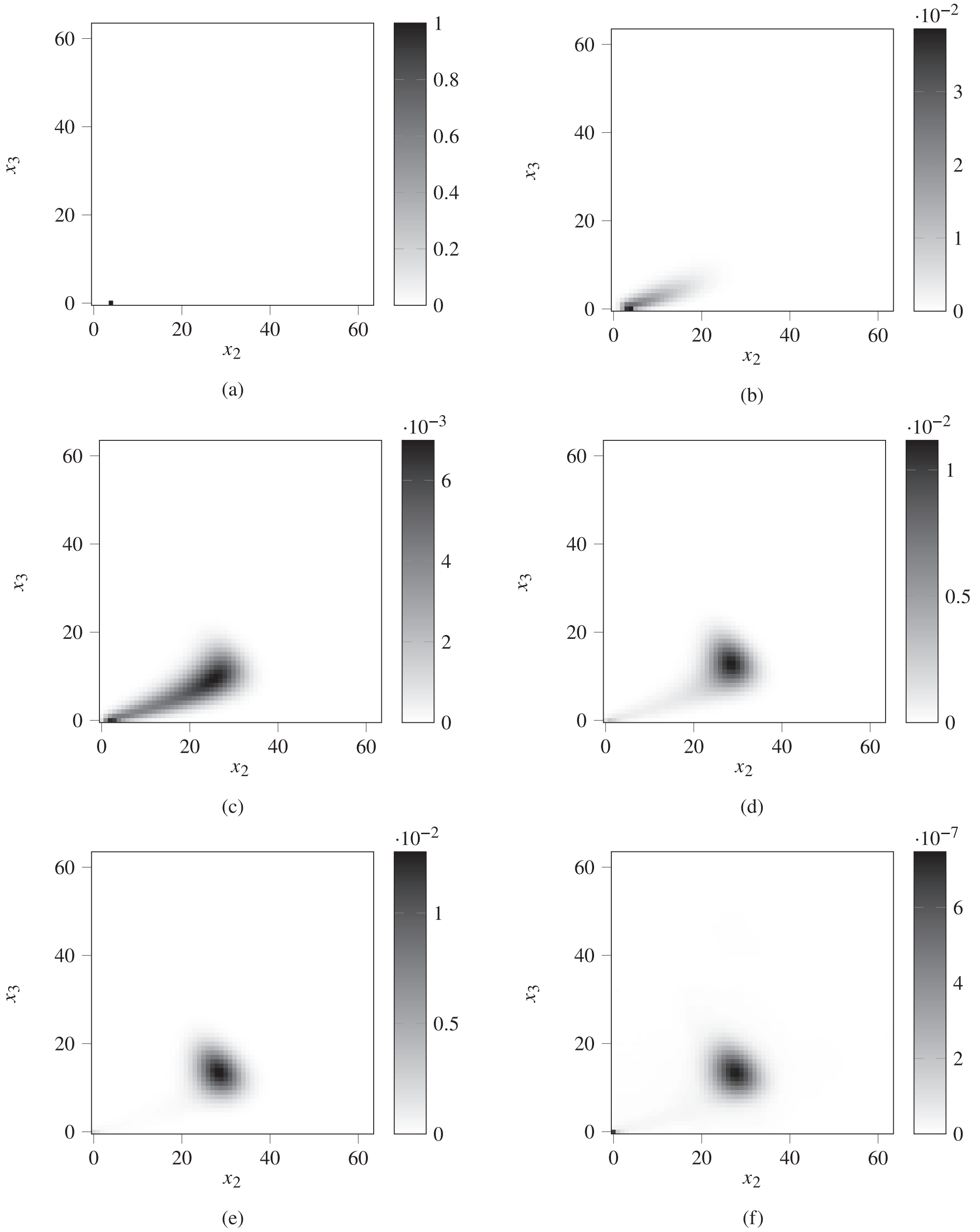
## B. Filtering and smoothing

As discussed in Sec. IV B, state filtering and smoothing can be performed in the TT-framework. We consider here the SEIR model presented in Sec. V A 2. We assume $N_o = 33$ equidistant observations with $\Delta t = 0.3125$. The time interval is now chosen as $[0, 10]$. The realizations are obtained using the Stochastic Simulation Algorithm (SSA)[4] (blue continuous lines in Fig. 4). The noise is assumed to be lognormal with variance 0.1 for $S$, $E$, and $I$ and 0.05 for $R$ (black × symbol in Fig. 4).

The TT-based forward–backward Algorithm 1 presented in Sec. IV B is used to perform state filtering and smoothing. The state is truncated to $(128, 128, 64, 32)$, and the Chebyshev basis is used for the time dependency. The runtime of the SEIR experiment is 12 min for a solver accuracy of $10^{-6}$ in terms of relative TT-solver residual. As the estimated state, we compute the expected value of the distribution given by $p\left(\boldsymbol{x}^{(k)} | \boldsymbol{y}^{(0)}, \ldots, \boldsymbol{y}^{(N_o)}\right)$ (red discontinuous line in Fig. 4) and the corresponding standard deviation (gray envelope in Fig. 4). In this case, the incorporation of the observation model in the TT-framework is beneficial to the reduction in the error since it acts like a reset condition. This can be observed in the decrease of the TT-rank after the multiplication with the tensor corresponding to the observation operator. The numerical experiment shows a decrease in the rank of up to 3 times, as can be observed in Fig. 5.

One more significant advantage of the TT representation is that the storage of the messages decreases dramatically compared to the full tensor representation, as the total storage needed is

© Author(s) 2021



(a)

(b)

(c)

(d)

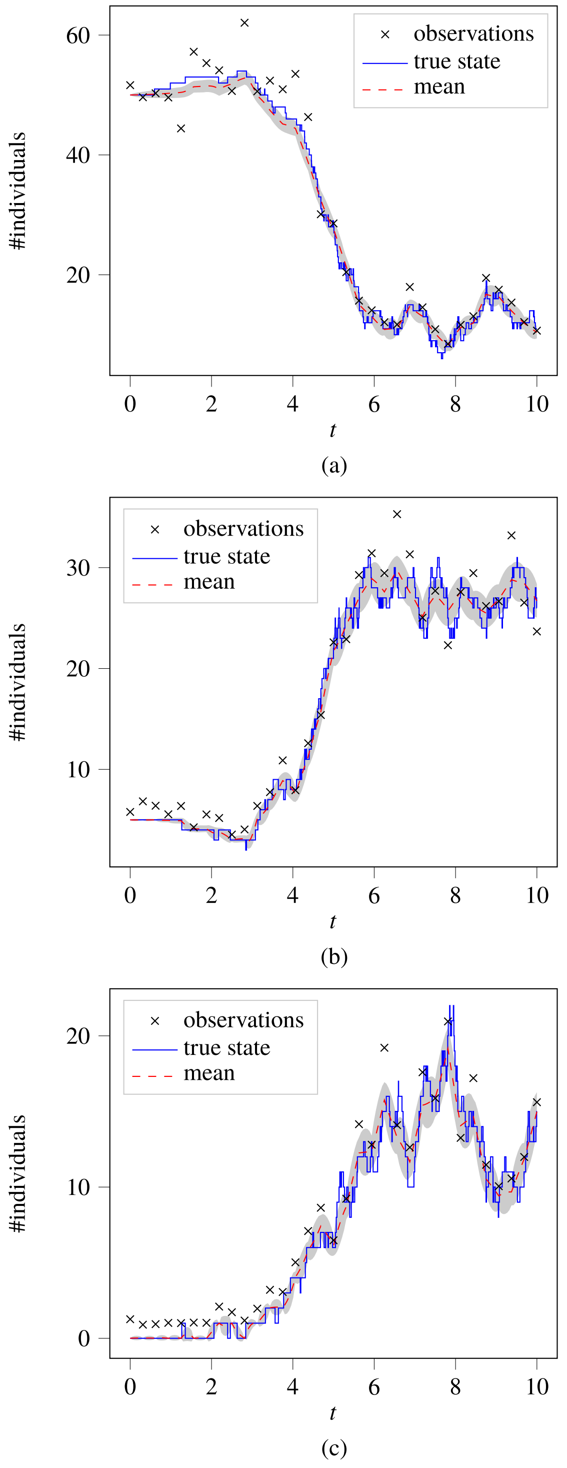
(e)

(f)

**FIG. 3.** Time evolution of exposed–infected (*EI*) marginal distribution at $t \in \{0, 2, 4, 6, 8\}$ and pointwise absolute error at $t_{\text{end}} = 8$. The solution is computed in the TT format, and the reference is obtained by integrating the CME over a fine time grid.

≈ 150 MB for the forward propagating messages and ≈ 190 MB for the backward propagating messages. In addition to that, computing the moments of the smooth distribution consists of multiplication with rank-**1** TT tensors. Moreover, the lognormal observation model is also translated to a rank-**1** tensor. The presented results are performed in the QTT format; however, the same test was performed without quantization. For the given state truncation, using the QTT format results in an acceleration by more than an order of magnitude in computation time, compared to the standard TT format.

© Author(s) 2021



(a)

(b)

(c)

**FIG. 4.** Smoothing for the SEIR model with initial population $\boldsymbol{x} = (50, 2, 1, 0)^{\top}$. The sample path is given by the blue line, the observations are marked with "×," and for the smoothed distribution, the mean (red dashed line) and the standard deviation (gray envelope) are plotted.

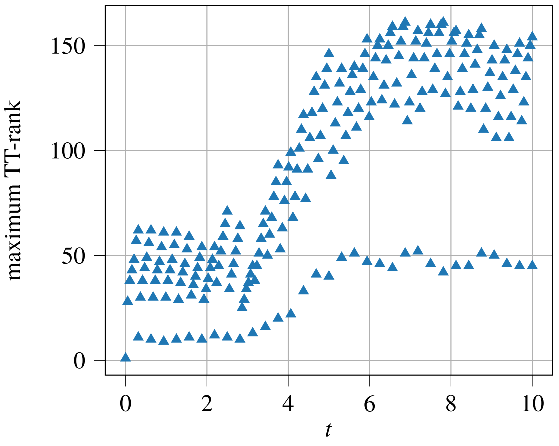


**FIG. 5.** Ranks for the forward pass over simulation time (triangle markers).

## C. Parameter inference

### 1. Simple gene expression

We now use Algorithm 2 to identify all four parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \theta_3, \theta_4) = (c_1, c_4, c_3, c_4)$ of the simple gene expression model from a noisy sample with $N_o = 64$ observations. In this case, the solver uses the QTT format, the observations are taken equidistantly every four time units, and the parameter priors are independent, truncated Gamma distributions, chosen such that they do not match the actual parameter. The parameter domain $\mathbb{R}_+^4$ is restricted to $\theta_i \in [0, 6c_i]$. As a reference, a sample of size $5 \cdot 10^5$ is drawn from the posterior using the Metropolis–Hastings algorithm. The CME in this case is solved in the full format with the built-in Python ODE solver.

With respect to the parameter dependence approximation, the basis of choice in this case is quadratic B-splines with equidistant knots scaled to the parameter range. The dimension of the individual univariate bases is 64. For the time integrator, a Chebyshev basis of dimension $T = 8$ is used with a subinterval size of 0.5 time units. The runtime is in this case ≈ 21 min with a maximum storage requirement of ≈ 9.2 MB for the joint over state and parameters (represented by a 6D tensor with mode size 64). The storage requirement for the extended CME operator in the QTT format is only 128 KB. Storing the tensor in the full format is intractable on standard machines even for this 2D example. For the given sample size, the Metropolis–Hastings algorithm is run for ≈ 1.5 days.

Since the posterior over the parameter space is four-dimensional, hence, not easy to visualize, the marginals for the individual parameters are computed and compared to the 1D histograms of the posterior sample for the purpose of validation. The results are presented in Fig. 6, where it can be observed that the posterior modes offer a reasonable approximation of the true values, the latter being marked with red vertical dashed lines. As a further characterization of the posterior, we compute its expected value, variance, and the mode of the PDF,

© Author(s) 2021

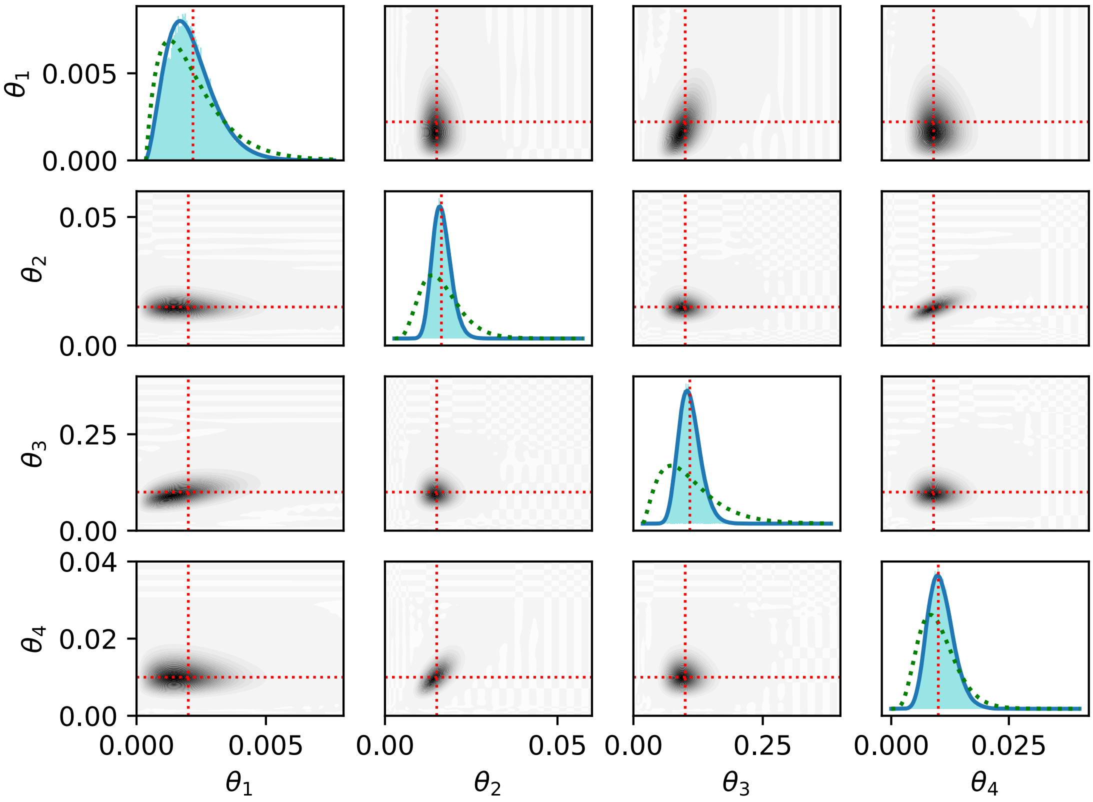


**FIG. 6.** Posterior marginal distributions for the four unknown reaction rates of the simple gene expression model. The black regions correspond to the high density of the PDF. The exact parameters are marked with the red dashed lines. For the 1D marginals, a histogram of the posterior sample is represented as a reference, as well as the prior (green dashed lines).

$$\mathbb{E}[\boldsymbol{\theta}] = (0.001\,924, 0.015\,12, 0.099\,85, 0.010\,57),$$

$$\mathbb{V}[\boldsymbol{\theta}] = (1.034 \times 10^{-6}, 8.685 \times 10^{-6}, 5.428 \times 10^{-4}, 7.624 \times 10^{-6}),$$

$$\hat{\boldsymbol{\theta}} = (0.001\,373, 0.014\,12, 0.090\,65, 0.009\,567).$$

For comparison, the mean and variance of the reference posterior sample are

$$\boldsymbol{\mu}_{\boldsymbol{\theta}} = (0.001\,922, 0.015\,07, 0.099\,92, 0.010\,52),$$

$$\boldsymbol{\sigma}^2_{\boldsymbol{\theta}} = (9.975 \times 10^{-7}, 8.498 \times 10^{-6}, 5.233 \times 10^{-4}, 7.518 \times 10^{-6}).$$

Since there is no analytical estimate for the combined error of the method, several runs with different hyperparameters are performed for this model. First, the accuracy of the solver in terms of relative residual, here denoted with $\epsilon$, is varied and the relative error of the TT-solver's solution is analyzed, first with respect to the MCMC solution and second with respect to the most accurate solution of the TT-solver, i.e., for $\epsilon = 10^{-6}$. The corresponding results are presented in Table III, where the simulation time and memory requirement for storing the joint in the TT format are also reported. As can be seen from Table III, the accuracy of the MCMC solution is reached already for a TT-solver accuracy of $\epsilon = 10^{-4}$. Looking at the memory consumption and the execution time, they both increase

**TABLE III.** Simple gene expression model error analysis with respect to solver accuracy $\epsilon$.

| $\epsilon$ | Error with respect to MCMC | Error with respect to $\epsilon = 10^{-6}$ | Time (min) | Memory (MB) |
|---|---|---|---|---|
| $10^{-3}$ | $7.35 \times 10^{-3}$ | $4.376 \times 10^{-3}$ | 2.4 | 1.8 |
| $10^{-4}$ | $2.35 \times 10^{-3}$ | $1.309 \times 10^{-3}$ | 7.4 | 4.57 |
| $10^{-5}$ | $3.59 \times 10^{-3}$ | $6.399 \times 10^{-5}$ | 21 | 9.29 |
| $10^{-6}$ | $3.64 \times 10^{-3}$ | ⋯ | 60 | 17.74 |

© Author(s) 2021

**TABLE IV.** Reactions of the three stage gene expression model.

| Reaction | $\alpha_m(\boldsymbol{x})$ | Rate $c_i$ |
|---|---|---|
| $G \rightarrow G + M$ | $c_1 x_1$ | 4.0 |
| $M \rightarrow M + P$ | $c_2 x_2$ | 10.0 |
| $M \rightarrow \emptyset$ | $c_3 x_2$ | 1.0 |
| $G + P \rightarrow G^*$ | $c_4 x_1 x_3$ | 0.2 |
| $G^* \rightarrow G + P$ | $c_5 x_4$ | 0.6 |
| $P \rightarrow \emptyset$ | $c_6 x_3$ | 1.0 |

for a higher TT-solver accuracy, which is expected since more solver iterations are needed to reach the desired residual.

Additionally, the dimension of the parameter basis has been investigated. If the tensor product basis is constructed using univariate B-spline bases of dimension 16, the prior can be well approximated. However during the inference, the decrease in the variance of the joint leads to an incapability to resolve the posterior since a finer basis is needed. If the discretization is increased to 32 for every parameter, the oscillations become negligible.

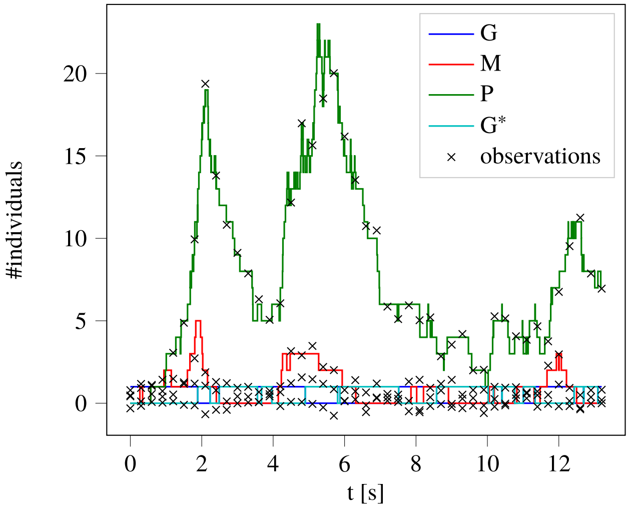


**FIG. 7.** Noisy observation sample for the three stage gene expression model (the number of observations is 45).

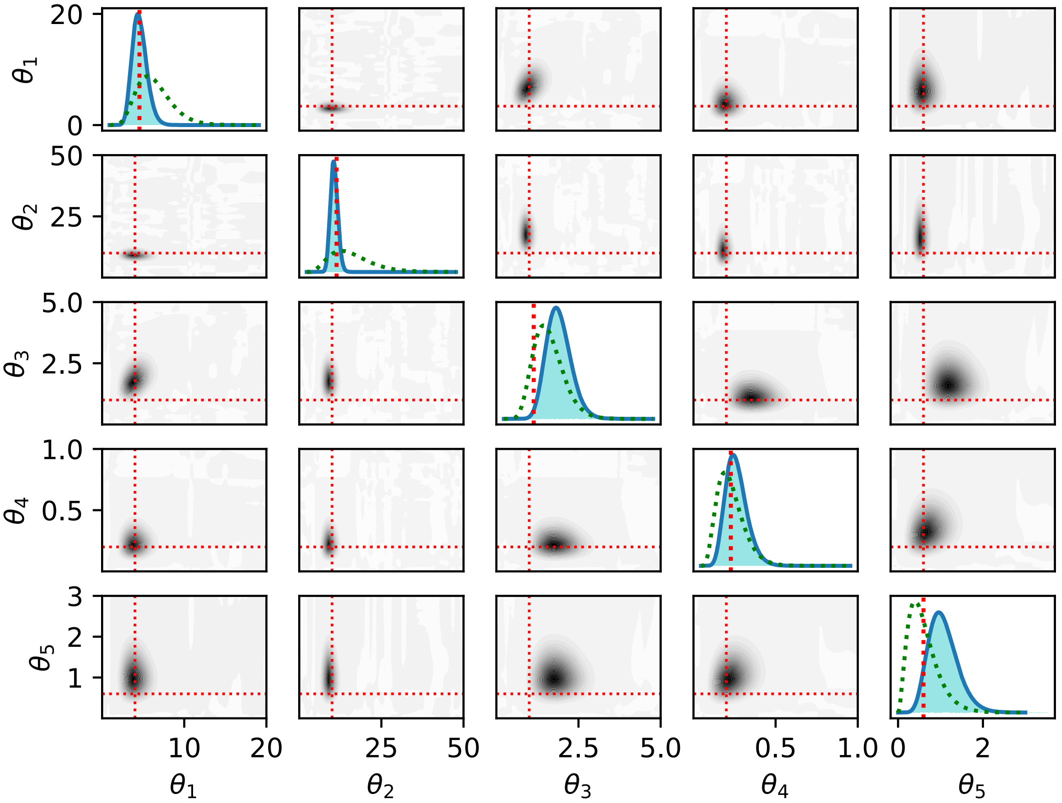


**FIG. 8.** Posterior marginal distributions for the five unknown reaction rates of the three stage gene expression model. The black regions correspond to the high density of the PDF. The exact parameters are marked with the red dashed lines. For the 1D marginals, a histogram of the posterior sample is represented as a reference, as well as the prior (green dashed lines).

© Author(s) 2021

As a conclusion, the limiting factor in the inference framework seems to be the accuracy of the TT-solver. For the purpose of inference, however, a relative residual value of $\epsilon = 10^{-5}$ seems to be sufficient for obtaining an accurate approximation and an acceptable computational time.

### 2. Gene expression model with feedback

The second model where the parameter identification is employed is the three stage gene expression model with a feedback loop.[51] The reactions as well as the reaction rates are presented in Table IV. A realization is drawn using the SSA, and equidistant sampling is performed with additive Gaussian noise (see Fig. 7).

The parameters to be identified are in this case $\boldsymbol{\theta} = (c_1, \ldots, c_5)$, and the parameter space is bounded to $[0, 5c_i]$. For the priors, we choose again independent Gamma distributions, which are truncated within the parameter space. The parameter dependence is approximated using a tensor product basis of univariate quadratic B-splines with dimension 64. The tolerance of the TT-solver is set to $10^{-5}$ in terms of relative residual.

The results are reported in Fig. 8 where we can see the visual match between the histograms and the 1D marginals on the diagonal. The expected value and variance of the posterior are

$$\mathbb{E}[\boldsymbol{\theta}] = (4.0358, 9.1720, 1.8398, 0.2378, 1.0686),$$
$$\mathbb{V}[\boldsymbol{\theta}] = (0.9649, 1.5117, 0.1669, 0.005187, 0.1172).$$

As a comparison, the mean and variance of the reference posterior sample are computed using MCMC,

$$\boldsymbol{\mu}_{\boldsymbol{\theta}} = (4.0503, 9.1995, 1.8443, 0.2379, 1.0680),$$
$$\boldsymbol{\sigma}^2_{\boldsymbol{\theta}} = (0.9874, 1.2367, 0.4123, 0.0720, 0.3467).$$

The relative error between the reference and the TT-solver-based modes is in the range of $10^{-3}$ for the expectation and $10^{-2}$ for the variance. The limiting factor is in this case the small MCMC sample size. Using the TT-solver, the execution time for this test case is 50 min. Regarding storage needs, only ≈ 12 MB of RAM is used. As a comparison, the MCMC simulation took ~2.5 days to complete for a sample size equal to $5 \cdot 10^5$.

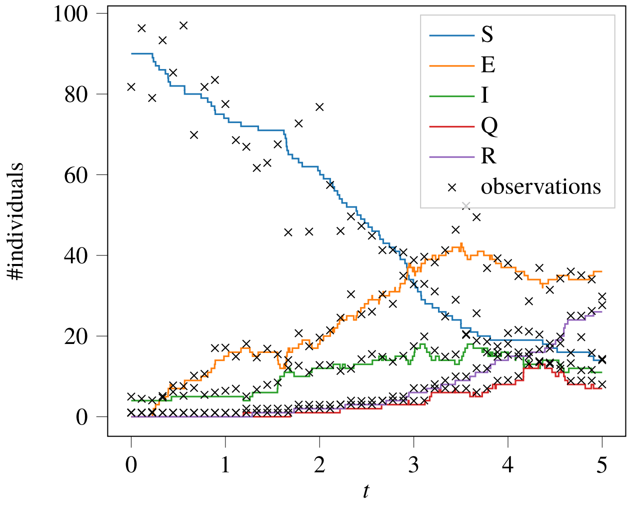


**FIG. 9.** Noisy observation sample for the SEIQR model (the number of observations is 45).

### 3. SEIQR model

The model considered now is an extension of the SEIR model presented in the filtering section and has one additional species: quarantined (Q). The modified reactions are found in Table V. We infer in this case the parameters $\boldsymbol{\theta} = (c_1, c_2, c_3, c_4)$ from 45 observations affected by lognormal noise (see Fig. 9). The species susceptible and exposed are observed with a higher degree of uncertainty, while quarantined and recovered are observed exactly. The execution time for a TT-solver accuracy of $\epsilon = 10^{-5}$ is ≈ 55 min with a maximum posterior size in the QTT format of ≈ 30 MB. As a comparison, the chosen state truncation of $(128, 64, 64, 32, 32)$ would require ≈ 4.2 GB only for storing the state for one parameter realization. The storage complexity for the parameter-dependent CME operator in the QTT format is ≈ 200 KB.

For this setup, the variance of the approximated posterior (see Fig. 10) is two orders lower than the prior for the first parameter and one order lower for the second and third parameters. This implies a higher confidence in the reconstruction of the parameters, which also indicates the need for a denser basis. In this case, choosing less

**TABLE V.** Reactions of the SEIQR model.

| Reaction | $\alpha_m(\boldsymbol{x})$ | Rate $c_i$ | Description |
|---|---|---|---|
| $S + I \to E + I$ | $c_1 x_1 x_3$ | 0.04 | Susceptible becomes exposed |
| $E \to I$ | $c_2 x_2$ | 0.4 | Exposed becomes infected |
| $I \to Q$ | $c_3 x_3$ | 0.4 | Infected is quarantined |
| $I \to \emptyset$ | $c_4 x_3$ | 0.004 | Infected individual dies |
| $I \to R$ | $c_5 x_3$ | 0.12 | Infected recovers with immunity |
| $Q \to R$ | $c_6 x_4$ | 0.8765 | Quarantined recovers with immunity |
| $I \to S$ | $c_7 x_3$ | 0.01 | Infected recovers without immunity |
| $Q \to S$ | $c_8 x_4$ | 0.01 | Quarantined recovers without immunity |
| $\emptyset \to S$ | $c_9$ | 0.01 | New susceptible individual |

© Author(s) 2021

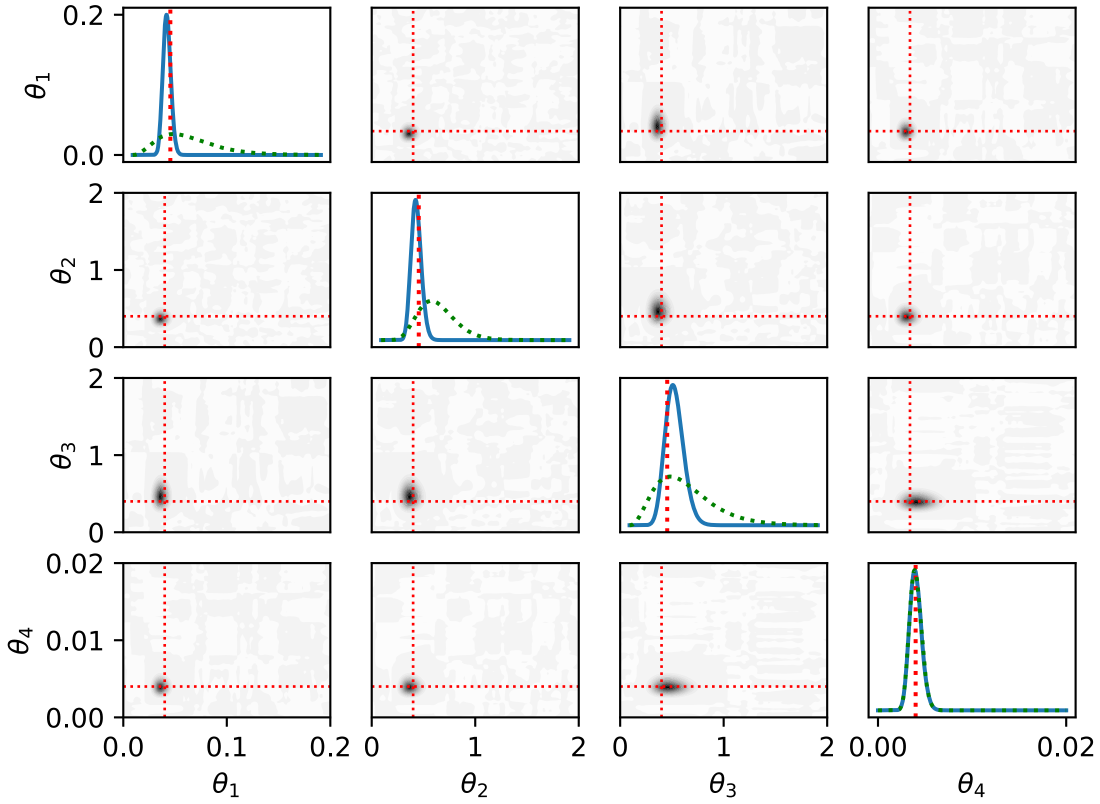


**FIG. 10.** Posterior marginal distributions for the four unknown reaction rates of the SEIQR model. The black regions correspond to the high density of the PDF. The exact parameters are marked with the red dashed lines, and the prior parameters are marked with green dashed lines.

than 64 points per parameter would lead to oscillations and inability to infer the posterior. This can be overcome by adaptively reducing the bounds of the parameter domain and re-interpolating the posterior on the new basis.

## VI. CONCLUSION

We presented a method based on the TT decomposition to solve the CME, either in its standard form or including parameter dependencies, and approximate the joint distribution over the state–parameter space, including the time dependency as well. Using the considered TT-framework, inference tasks such as state smoothing and parameter identification can be performed accurately and efficiently. The proposed TT-framework is also applied to solve the inverse problem of identifying the values of the parameters governing the system under investigation from noisy state observations. The resulting numerical approximation of the posterior PDF over the truncated parameter space can be efficiently stored and manipulated in the TT format.

A series of numerical experiments with a simple gene expression model and with the SEIR virus model show clearly that the state-time TT approximation reduces the storage needs of the CME to a mere fraction of what a standard CME solution method requires. Moreover, by performing multilinear algebraic operations in the TT format, the execution time is significantly reduced as well. Similar benefits are observable in the context of inference tasks, where the proposed TT-based filtering, smoothing, and parameter identification approaches yield accurate results for a significantly reduced computational cost.

While standard inference procedures for the single trajectory setting such as MCMC require repeated solutions of the CME for different parameter configurations, the joint approach presented here requires only one forward pass on the augmented state space. One drawback of the parameter space discretization is that it can cause problems when the posterior is much more concentrated than the prior. As demonstrated on the SEIQR model, this can be overcome by dynamically adapting the basis.

In this work, we have focused on inferring the rate constants of structurally known models from a single trajectory. An important direction for future research is to extend the approach to multiple trajectories with shared parameters. Another interesting direction is to consider different types of uncertainties, e.g., the involved species or the types of reactions. Since the presented method is fully Bayesian, this could be realized by scoring different candidate structures via Bayesian model comparison.

© Author(s) 2021

## ACKNOWLEDGMENTS

The work of I. G. Ion was supported by the Graduate School of Computational Engineering within the Centre for Computational Engineering at Technische Universität Darmstadt. D. Loukrezis was supported by the German Federal Ministry for Education and Research (BMBF) via Research Contract No. 05K19RDB. C. Wildner and H. Koeppl were supported by the European Research Council (ERC) within the CONSYN project, Grant Agreement No. 773196.

## DATA AVAILABILITY

The data that support the findings of this study are freely available at https://github.com/ion-g-ion/paper-cme-tt.

## REFERENCES

[1] D. T. Gillespie, "A rigorous derivation of the chemical master equation," Physica A **188**, 404–425 (1992).

[2] M. A. Gibson and J. Bruck, "Efficient exact stochastic simulation of chemical systems with many species and many channels," J. Phys. Chem. A **104**, 1876–1889 (2000).

[3] M. Hemberg and M. Barahona, "Perfect sampling of the master equation for gene regulatory networks," Biophys. J. **93**, 401–410 (2007).

[4] D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," J. Comput. Phys. **22**, 403–434 (1976).

[5] S. Engblom, "Spectral approximation of solutions to the chemical master equation," J. Comput. Appl. Math. **229**, 208–221 (2009).

[6] P. Deuflhard, W. Huisinga, T. Jahnke, and M. Wulkow, "Adaptive discrete Galerkin methods applied to the chemical master equation," SIAM J. Sci. Comput. **30**, 2990–3011 (2008).

[7] B. Munsky and M. Khammash, "The finite state projection algorithm for the solution of the chemical master equation," J. Chem. Phys. **124**, 044104 (2006).

[8] K. Burrage, M. Hegland, F. Macnamara, and R. Sidje, "A Krylov-based finite state projection algorithm for solving the chemical master equation arising in the discrete modelling of biological systems," in Proceedings of the Markov 150th Anniversary Conference, 2006.

[9] V. Wolf, R. Goel, M. Mateescu, and T. A. Henzinger, "Solving the chemical master equation using sliding windows," BMC Syst. Biol. **4**, 42 (2010).

[10] S. Dolgov and B. Khoromskij, "Simultaneous state-time approximation of the chemical master equation using tensor product formats," Numer. Linear Algebra Appl. **22**, 197 (2015).

[11] V. Kazeev, M. Khammash, M. Nip, and C. Schwab, "Direct solution of the chemical master equation using quantized tensor trains," PLoS Comput. Biol. **10**, e1003359 (2014).

[12] T. Dinh and R. B. Sidje, "An adaptive solution to the chemical master equation using quantized tensor trains with sliding windows," Phys. Biol. **17**, 065014 (2020).

[13] H. D. Vo and R. B. Sidje, "An adaptive solution to the chemical master equation using tensors," J. Chem. Phys. **147**, 044102 (2017).

[14] T. Jahnke and W. Huisinga, "A dynamical low-rank approach to the chemical master equation," Bull. Math. Biol. **70**, 2283 (2008).

[15] A. Gupta, J. Mikelson, and M. Khammash, "A finite state projection algorithm for the stationary solution of the chemical master equation," J. Chem. Phys. **147**, 154101 (2017).

[16] P. Kügler, "Moment fitting for parameter inference in repeatedly and partially observed stochastic biological models," PLoS One **7**, e43001 (2012).

[17] C. Zechner, J. Ruess, P. Krenn, S. Pelet, M. Peter, J. Lygeros, and H. Koeppl, "Moment-based inference predicts bimodality in transient gene expression," Proc. Natl. Acad. Sci. U. S. A. **109**, 8340–8345 (2012).

[18] F. Fröhlich, P. Thomas, A. Kazeroonian, F. J. Theis, R. Grima, and J. Hasenauer, "Inference for stochastic chemical kinetics using moment equations and system size expansion," PLoS Comput. Biol. **12**, e1005030 (2016).

[19] B. Munsky, G. Li, Z. R. Fox, D. P. Shepherd, and G. Neuert, "Distribution shapes govern the discovery of predictive models for gene regulation," Proc. Natl. Acad. Sci. U. S. A. **115**, 7533–7538 (2018).

[20] Z. Cao and R. Grima, "Accuracy of parameter estimation for auto-regulatory transcriptional feedback loops from noisy data," J. R. Soc., Interface **16**, 20180967 (2019).

[21] L. Bronstein and H. Koeppl, "A variational approach to moment-closure approximations for the kinetics of biomolecular reaction networks," J. Chem. Phys. **148**(1), 014105 (2018).

[22] P. Milner, C. S. Gillespie, and D. J. Wilkinson, "Moment closure based parameter inference of stochastic kinetic models," Stat. Comput. **23**, 287–295 (2013).

[23] V. Stathopoulos and M. A. Girolami, "Markov chain Monte Carlo inference for Markov jump processes via the linear noise approximation," Philos. Trans. R. Soc., A **371**, 20110541 (2013).

[24] P. Fearnhead, V. Giagos, and C. Sherlock, "Inference for reaction networks using the linear noise approximation," Biometrics **70**, 457–466 (2014).

[25] A. Golightly and D. J. Wilkinson, "Bayesian parameter inference for stochastic biochemical network models using particle Markov chain Monte Carlo," Interface Focus **1**, 807–820 (2011).

[26] C. Zechner, M. Unger, S. Pelet, M. Peter, and H. Koeppl, "Scalable inference of heterogeneous reaction kinetics from pooled single-cell recordings," Nat. Methods **11**, 197–202 (2014).

[27] I. V. Oseledets, "Tensor-train decomposition," SIAM J. Sci. Comput. **33**, 2295–2317 (2011).

[28] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," SIAM Rev. **51**, 455–500 (2009).

[29] S. V. Dolgov, "A tensor decomposition algorithm for large ODEs with conservation laws," Comput. Methods Appl. Math. **19**, 23 (2018).

[30] I. V. Oseledets and E. E. Tyrtyshnikov, "Breaking the curse of dimensionality, or how to use SVD in many dimensions," SIAM J. Sci. Comput. **31**, 3744–3759 (2009).

[31] D. Savostyanov and I. Oseledets, "Fast adaptive interpolation of multi-dimensional arrays in tensor train format," in *The 2011 International Workshop on Multidimensional (nD) Systems* (IEEE, 2011), pp. 1–8.

[32] I. Oseledets and E. Tyrtyshnikov, "TT-cross approximation for multidimensional arrays," Linear Algebra Appl. **432**, 70–88 (2010).

[33] S. V. Dolgov, "TT-GMRES: Solution to a linear system in the structured tensor format," Russ. J. Numer. Anal. Math. Modell. **28**, 149–172 (2013).

[34] I. V. Oseledets and S. V. Dolgov, "Solution of linear systems and matrix inversion in the TT-format," SIAM J. Sci. Comput. **34**, A2718 (2012).

[35] S. Holtz, T. Rohwedder, and R. Schneider, "The alternating linear scheme for tensor optimization in the tensor train format," SIAM J. Sci. Comput. **34**, A683–A713 (2012).

[36] L. Grasedyck, M. Kluge, and S. Krämer, "Variants of alternating least squares tensor completion in the tensor train format," SIAM J. Sci. Comput. **37**, A2424–A2450 (2015).

[37] S. R. White, "Density-matrix algorithms for quantum renormalization groups," Phys. Rev. B **48**, 10345 (1993).

[38] S. V. Dolgov and D. V. Savostyanov, "Alternating minimal energy methods for linear systems in higher dimensions," SIAM J. Sci. Comput. **36**, A2248 (2014).

[39] M. Hegland and J. Garcke, "On the numerical solution of the chemical master equation with sums of rank one tensors," ANZIAM J. **52**, 628 (2011).

[40] P. Gelß, "The tensor-train format and its applications: Modeling and analysis of chemical reaction networks, catalytic processes, fluid flows, and Brownian dynamics," Ph.D. thesis, Freie Universität Berlin, 2017.

[41] L. Trefethen, *Spectral Methods in MATLAB, Software, Environments, and Tools* (Society for Industrial and Applied Mathematics SIAM, Philadelphia, PA, 2000).

[42] B. N. Khoromskij, "$O(d\log N)$-quantics approximation of $N$-$d$ tensors in high-dimensional numerical modeling," Constr. Approximation **34**, 257–280 (2011).

[43] S. V. Dolgov, B. N. Khoromskij, and I. V. Oseledets, "Fast solution of parabolic problems in the tensor train/quantized tensor train format with initial application to the Fokker–Planck equation," SIAM J. Sci. Comput. **34**, A3016–A3038 (2012).

[44] D. Bigoni, A. P. Engsig-Karup, and Y. M. Marzouk, "Spectral tensor-train decomposition," SIAM J. Sci. Comput. **38**, A2405 (2016).

[45] C. de Boor, *A Practical Guide to Spline* (Springer-Verlag, 1978), Vol. 27.

[46] L. Huang, L. Pauleve, C. Zechner, M. Unger, A. S. Hansen, and H. Koeppl, "Reconstructing dynamic molecular states from single-cell time series," J. R. Soc., Interface **13**, 20160533 (2016).

[47] "Forward-backward algorithm," in *Encyclopedia of Biometrics*, edited by S. Z. Li and A. Jain (Springer US, Boston, MA, 2009), p. 580.

[48] B. Alberts, A. Johnson, J. Lewis, P. Walter, M. Raff, and K. Roberts, *Molecular Biology of the Cell*, 4th ed.; International Student ed. (Routledge, 2002).

[49] J. C. Butcher, "Numerical differential equation methods," in *Numerical Methods for Ordinary Differential Equations* (John Wiley & Sons, Ltd., 2016), https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119121534.ch2.

[50] H. W. Hethcote, "The mathematics of infectious diseases," SIAM Rev. **42**, 599–653 (2000).

[51] K. Öcal, R. Grima, and G. Sanguinetti, "Parameter estimation for biochemical reaction networks using Wasserstein distances," J. Phys. A: Math. Theor. **53**, 034002 (2019).

© Author(s) 2021