Applied Mathematics and Computation 460 (2024) 128290

Contents lists available at ScienceDirect

Applied Mathematics and Computation

journal homepage: www.elsevier.com/locate/amc

Tensor product approach to modelling epidemics on networks ☆

Sergey Dolgov [a,1], Dmitry Savostyanov [b,*,2]

[a] University of Bath, Claverton Down, Bath, BA2 7AY, United Kingdom
[b] University of Essex, Wivenhoe Park, Colchester, CO4 3SQ, United Kingdom

ARTICLE INFO

MSC:
15A69
34A30
37N25
60J28
65F55
90B15
95C42

Keywords:
Epidemiological modelling
Networks
Chemical master equation
Tensor train
Stochastic simulation algorithm
Monte Carlo simulation
Rare events
High precision

ABSTRACT

To improve mathematical models of epidemics it is essential to move beyond the traditional assumption of homogeneous well–mixed population and involve more precise information on the network of contacts and transport links by which a stochastic process of the epidemics spreads. In general, the number of states of the network grows exponentially with its size, and a master equation description suffers from the curse of dimensionality. Almost all methods widely used in practice are versions of the stochastic simulation algorithm (SSA), which is notoriously known for its slow convergence. In this paper we numerically solve the chemical master equation for an SIR model on a general network using recently proposed tensor product algorithms. In numerical experiments we show that tensor product algorithms converge much faster than SSA and deliver more accurate results, which becomes particularly important for uncovering the probabilities of rare events, e.g. for number of infected people to exceed a (high) threshold.

1. Introduction

Modelling of epidemics is crucial to inform policies and support decision making for disease prevention and control. The recent outbreak of COVID-19 pandemic raised a significant scientific and public debate regarding the quality of the mathematical models used to predict the effect of the pandemics and to choose an appropriate response strategy. One of the first epidemiological models, proposed by Kermack and McKendrick in 1927 [1], assumes that each member of the population can be either susceptible to a disease, infected, or recovered. Its second important assumption is that the population is *well–mixed*, i.e. all members are in contact with each other and have the same chance of getting and passing a disease. Under this assumption, the system dynamics is governed only by the sizes of the compartments for susceptible, infected, and recovered part of the population, and can be described by three ordinary differential equations, one for each compartment. Despite its simplicity, the Kermack–McKendrick SIR model can describe important stages of the epidemics, such as exponential growth of the number of infected people at the beginning of epidemic, and

☆ Equal contributions. The order of authors is alphabetical.

* Corresponding author.
E-mail addresses: S.Dolgov@bath.ac.uk (S. Dolgov), D.Savostyanov@essex.ac.uk, dmitry.savostyanov@gmail.com (D. Savostyanov).

[1] Supported by the Engineering and Physical Sciences Research Council New Investigator Award EP/T031255/1.
[2] Supported by the Leverhulme Trust Research Fellowship RF-2021-258.

https://doi.org/10.1016/j.amc.2023.128290
Received 21 September 2022; Received in revised form 21 April 2023; Accepted 11 August 2023

Available online 23 August 2023

the exponential decay after the epidemics passed its peak. For this reason, this and other compartmental models are often included in academic curriculum and used to popularise epidemiological modelling among general public and present it to policy makers.

When it comes to policy making, however, we need models that can provide accurate quantitative results. The main assumption behind the compartmental models — that the population is well–mixed — does not hold very well for human population. People are not in constant contact with each other, and the probability of two people to meet each other depends significantly on where they live, where they work and what social contacts they maintain. The speed of the epidemics depends not just on the total number of infected people, but on where the infected people are located in relation to the susceptible part of the population. For example, if all infected people are located in an isolated region, the disease will spread much slower, than if the same number of infected people were spread uniformly among the susceptible part of the population. Moreover, the contacts are inherently stochastic. Due to this *internal* noise, deterministic models can give wrong results for extreme scenarios of e.g. small number of rarely contacting infected individuals.

To deliver more accurate predictions, *stochastic epidemiological models on networks* have been proposed [2,3]. Unfortunately, these models are also much more complex. They are formulated as a Markov process with the state space formed of all possible arrangements of susceptible, infected and recovered individuals on the network; while the transition probabilities depend on the structure of the network's connections and model parameters, such as infection and recovery rates. The dynamics of the probability distribution is governed by the master equation (also known as Chapman or forward Kolmogorov equation), which takes the form of an autonomous ODE system. Since each network configuration has to be treated separately, the total number of equations that we need to solve grows exponentially with respect to the population size. This obstacle, known as the *curse of dimensionality* makes numerical solution of these models prohibitively difficult for networks of moderate and large size. Therefore, virtually all widely used methods of solving stochastic population models are variants of the Monte Carlo stochastic simulation algorithm [4]. Despite its simplicity and popularity, this algorithm is known for its slow convergence following from the central limit theorem. Alternative approaches include mean–field approximations [5,6], effective degree models [7–9], and edge–based compartmental models [10], but these models are approximate and rely on truncation of the state space, effects of which on accuracy are difficult to estimate and/or keep below a desired tolerance for a general network.

Recently, a family of *tensor product* methods was proposed for breaking the curse of dimensionality and making high–dimensional problems possible to solve. In these methods, the solution (in our case, the joint probability distribution function of individual states) is approximated by a compressed format, which often converges much faster (e.g. exponentially) compared to the central limit theorem rate in Monte Carlo methods [11]. Starting with basic algorithms for approximating a given array in tensor train (TT) [12] or Hierarchical Tucker (HT) [13] format, new methods were proposed for solving linear systems [14,15] and eigenproblems [16–18], and recently for solving time–dependent problems [19]. Initially motivated by quantum physics [20–22], tensor product algorithms recently extended their domain to a variety of applications, see [23–26]. In this paper we apply tensor product algorithms to compute, approximately but with controlled accuracy, the joint probability distribution function of network states.

Another difficulty of a general master equation is the exponential number of transitions, in addition to the exponential number of states. Stochastic population models can often be written as a system of a polynomial number of stochastic chemical reactions, and solved using the chemical master equation (CME) [27]. In addition to the direct Monte Carlo simulations of the realisations of the model [4,28–30], a direct solution of the CME (or outputs thereof) was proposed using an adaptive finite state projection [31–33], sparse grids [34], radial basis functions [35], neural networks [36,37], and tensor product approximations, in particular, in the Tucker decomposition [38], CP decomposition [39,40], and TT decomposition [41,42,19,43–46]. Most of these papers consider the CME formulations of gene regulatory networks, where an accurate description of stochasticity is important due to small copy numbers. However, there seem to be a little coverage of population models. Somewhat related is a lattice model of unimolecular adsorption/desorption which was explored in [46].

In this paper we apply tensor product algorithms to solve the exponentially large systems of ODEs that mathematically capture the evolution of epidemics on networks, without any uncontrollable approximations caused by the simplification of the model. The fast convergence of the tensor product approximation allows us to solve the CME to extremely high accuracies, up to 6 decimal digits. This enables accurate estimation of probabilities of *rare events*, such as simultaneous infection of a large number of people in a network. Rare event simulation is a infamously formidable task, since the number of samples in a direct Monte Carlo method needs to be inversely proportional to the (small) event probability [47–49]. We demonstrate that we can accurately estimate events of probability as small as 10^{-6} in a small world network of 50 individuals.

2. Background

The original Kermack–McKendrick model [1] separated people in three groups — susceptible, infected and recovered — and described the state of the epidemics by the size of each group or *compartment*. For this information to be sufficient for describing the dynamics of epidemic an implied assumption has to be made that the system is *well–mixed* or homogeneous, i.e. each member is in contact with everyone and the disease can spread from each infected person to each susceptible person with the same probability. This assumption is not very realistic — the network of contacts between people normally has a complex structure, with some people having (much) more contacts than others. In this case it is not possible to describe the situation with just specifying sized of all compartments, as the location of infected people in the network plays a key role in the dynamics of epidemic. The simplest illustration can be a situation when a single infected person is completely isolated from the rest of the network (and infection can not spread), compared to this infected person been connected to all people in the network (and infection can spread rapidly).

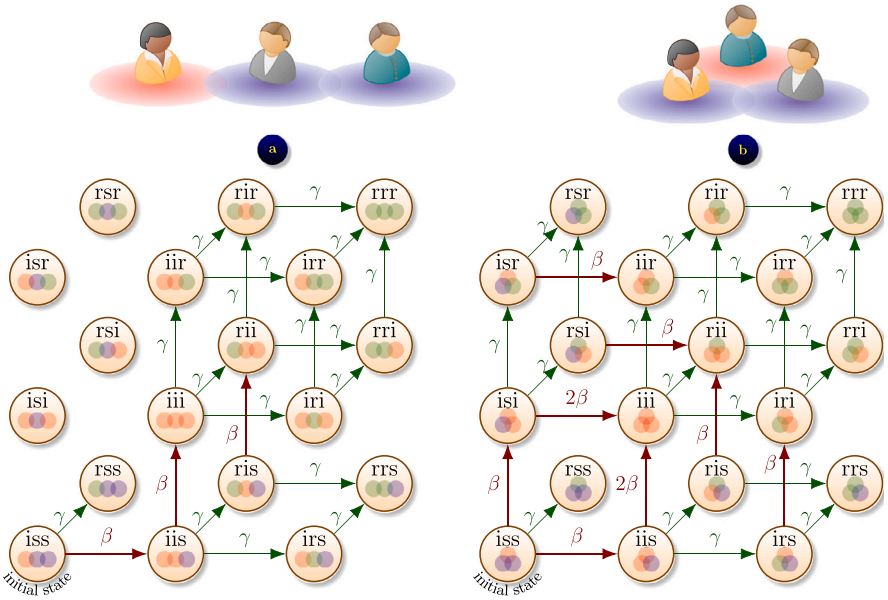

Fig. 1. Markov chain transitions between network states: (a) SIR epidemic on a chain of $N = 3$ people; (b) SIR epidemic in a fully mixed network of $N = 3$ people.

Since the compartmental description can not accurately describe epidemic on a general network, we will need to apply stochastic description, by considering for all states the network can reach and describing the evolution of probabilities of these states. In this section we recall the corresponding mathematical model of SIR epidemics on networks, explain the computational challenges arising due to the exponentially large size of this model, and briefly describe the stochastic simulation algorithm, which avoids the problem and is widely used in practice because of that.

2.1. Epidemics on networks

A network (or unweighted simple directed graph) is a set of nodes (vertices, sites)

$$\mathcal{V} = \{1, \dots, N\},$$

representing individual people, and a set of links (edges, connections)

$$\mathcal{E} = \{(m, n) : m \in \mathcal{V}, n \in \mathcal{V}, m \neq n\},$$

representing contacts between them. If $(m, n) \in \mathcal{E}$, which we will also denote using an adjacency relation $m \sim n$, a person m (if infected) can pass a disease on a person n (if susceptible). If all contacts can potentially pass disease in both directions (e.g. there are no personal protection measures in place), the network is undirected, i.e. $(m, n) \in \mathcal{E} \Leftrightarrow (n, m) \in \mathcal{E}$, in which case $m \sim n$ is a symmetric relation.

For a SIR epidemic, each person can be in exactly one of three states,

$$x_n \in \mathbb{X} = \{\mathrm{s}, \mathrm{i}, \mathrm{r}\} = \{\text{susceptible}, \text{infected}, \text{recovered}\} = \{1, 2, 3\}, \qquad \text{for } n \in \mathcal{V}. \tag{1}$$

Hence, the state of the whole system can be written as

$$\mathrm{x} = \begin{pmatrix} x_1 & x_2 & \dots & x_N \end{pmatrix}^T \in \Omega = \mathbb{X}^N.$$

We denote the probability to find the system in state x at time t as $p(\mathrm{x}, t) = p(x_1, x_2, \dots, x_N, t)$. The transitions between the states are described using the two *reactions*, i.e. infection and recovery, respectively:

$$\mathsf{P}\left(\mathrm{x}(t + \delta t) = (x_1, \dots, x_n = \mathrm{i}, \dots, x_N) \,\middle|\, \mathrm{x}(t) = (x_1, \dots, x_n = \mathrm{s}, \dots, x_N)\right) = I_n(\mathrm{x})\beta\delta t,$$

$$\mathsf{P}\left(\mathrm{x}(t + \delta t) = (x_1, \dots, x_n = \mathrm{r}, \dots, x_N) \,\middle|\, \mathrm{x}(t) = (x_1, \dots, x_n = \mathrm{i}, \dots, x_N)\right) = \gamma\delta t,$$

where β is the (per contact) infection rate, γ is the (per capita) recovery rate, and $I_n(\mathrm{x}) = |\{m \in \mathcal{V} : m \sim n, x_m = \mathrm{i}\}|$ counts the number of infected neighbours of person n in the state x. These reactions connect the states of the system (as shown in Fig. 1), thus forming a Markov chain network (weighted directed graph with loops), where nodes are network states $\mathrm{x} \in \mathbb{X}^N$, links are transitions $\{(\mathrm{x}, \mathrm{y}) : \mathsf{P}(\mathrm{y} \text{ at time } t + \delta t \mid \mathrm{x} \text{ at time } t) \neq 0\}$, and weights are reaction rates:

$$p_{\mathrm{x}\to\mathrm{y}} = \begin{cases} p^{(\mathrm{inf})}_{\mathrm{x}\to\mathrm{y}} = I_n(\mathrm{x})\beta, & \text{if } \exists n \in \mathcal{V} : x_n = \mathrm{s},\ y_n = \mathrm{i}, \text{ and } x_m = y_m \text{ for } m \neq n; \\ p^{(\mathrm{rec})}_{\mathrm{x}\to\mathrm{y}} = \gamma, & \text{if } \exists n \in \mathcal{V} : x_n = \mathrm{i},\ y_n = \mathrm{r}, \text{ and } y_m = x_m \text{ for } m \neq n; \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Since the probability of the transition depends only on the current state (and not on the history of previous events), this process is a continuous–time Markov chain. The transition rates describe the dynamics of probabilities of network states as follows:

$$p'(\mathrm{x},t) = \sum_{\mathrm{y}\in\Omega} \left(p_{\mathrm{y}\to\mathrm{x}} \cdot p(\mathrm{y},t) - p_{\mathrm{x}\to\mathrm{y}} \cdot p(\mathrm{x},t) \right). \tag{3}$$

These ODEs need to be solved subject to initial conditions $p(\mathrm{x}_0, 0) = 1$ for the initial state $\mathrm{x} = \mathrm{x}_0$ and $p(\mathrm{x}, 0) = 0$ for other states $\mathrm{x} \neq \mathrm{x}_0$. Collecting all ODEs in a system, we obtain a *Markovian master equation* [27,2], also known as Chapman or *forward Kolmogorov equations*:

$$\mathbf{p}'(t) = \mathbf{A}\mathbf{p}(t), \qquad \mathbf{p}'(0) = \mathbf{p}_0, \tag{4}$$

where $\mathbf{p}(t) = \left[p(\mathrm{x},t) \right]_{\mathrm{x}\in\Omega}$ is the unknown *probability distribution function* (p.d.f.), $\mathbf{p}_0$ is a unit vector with 1 in position of the initial state x_0, and $\mathbf{A} = [A(\mathrm{x},\mathrm{y})]_{\mathrm{x},\mathrm{y}\in\Omega}$ is matrix with elements

$$\begin{cases} A(\mathrm{x},\mathrm{x}) = -\sum_{\mathrm{y}\in\Omega} p_{\mathrm{x}\to\mathrm{y}}, & \text{on the diagonal} \\ A(\mathrm{x},\mathrm{y}) = p_{\mathrm{y}\to\mathrm{x}}, & \text{off-diagonal, i.e. for } \mathrm{x} \neq \mathrm{y}. \end{cases}$$

By solving (4), we obtain probabilities $p(\mathrm{x},t)$ for all states $\mathrm{x} \in \Omega$, and can calculate statistical moments,

$$\mathsf{E}[I(t)] = \sum_{\mathrm{x}\in\Omega} I(\mathrm{x}) p(\mathrm{x},t), \qquad \mathsf{V}[I(t)] = \sum_{\mathrm{x}\in\Omega} (I(\mathrm{x}) - \mathsf{E}[I(t)])^2 p(\mathrm{x},t), \tag{5}$$

where the function $I(\mathrm{x}) = I(x_1, \dots, x_n) = |\{n \in \mathcal{V} : x_n = \mathrm{i}\}|$ counts the total number of infected individuals for a given state x.

Solving (4) is however not easy due to its large size. The state space $\Omega = \mathbb{X}^N$ contains $|\Omega| = 3^N$ states, meaning that $\mathbf{p}(t)$ is a vector of size 3^N and $\mathbf{A}$ is a $3^N \times 3^N$ sparse matrix. As N increases, the storage and computational costs grow as $\mathcal{O}(3^N)$ and become prohibitively expensive even for modest values $N \gtrsim 20$. This problem, known as the *curse of dimensionality*, is the major obstacle in solving high–dimensional problems that appear in a variety of applications, e.g. complex systems, quantum computations, machine learning, and epidemics on networks, which we consider in this paper.

2.2. Simplified models of epidemics on networks

Since master equations (4) suffer from the curse of dimensionality, a number of alternative approaches to modelling epidemics on networks were developed.

An approach known as *lumping* [50,51] attempts to group the network states in classes, in order to obtain a coarse description of the system behaviour by observing transitions between the groups, rather than individual states. Consider, for example, a network of $N = 3$ people fully connected to each other. A full stochastic description of this network involves 3^N network states as shown in Fig. 1(right). However, we can collect states $(x_1 x_2 x_3)$ in groups described by the total number of susceptible and recovered people (S, R), lumping the network model into a stochastic model for SIR epidemics in well–mixed groups.

$$\begin{aligned} &(0,0) = \{(\mathrm{iii})\}, && (1,0) = \{(\mathrm{sii}),(\mathrm{isi}),(\mathrm{iis})\}, \\ &(2,0) = \{(\mathrm{ssi}),(\mathrm{sis}),(\mathrm{iss})\}, && (3,0) = \{(\mathrm{sss})\}, \\ &(0,1) = \{(\mathrm{rii}),(\mathrm{iri}),(\mathrm{iir})\}, && (1,1) = \{(\mathrm{sir}),(\mathrm{sri}),(\mathrm{irs}),(\mathrm{isr}),(\mathrm{rsi}),(\mathrm{ris})\}, \\ &(2,1) = \{(\mathrm{ssr}),(\mathrm{srs}),(\mathrm{ssr})\}, && (0,2) = \{(\mathrm{irr}),(\mathrm{rir}),(\mathrm{irr})\}, \\ &(1,2) = \{(\mathrm{srr}),(\mathrm{rsr}),(\mathrm{srr})\}, && (0,3) = \{(\mathrm{rrr})\}. \end{aligned}$$

For fully connected network of N people, lumping combines 3^N network states in $\frac{1}{2}(N+1)(N+2) = \mathcal{O}(N^2)$ groups, massively reducing the storage and computational complexity, while maintaining all the information necessary for describing the evolution. This is achieved by sacrificing information about the exact positions of susceptible, infected and recovered people in the network, which can be considered insignificant or unnecessary in this case. The coarser description obtained by lumping remains exact and perfectly matches the results obtained by the full network description. However, the effectiveness of lumping depends heavily on the number of symmetries available in the network of connections between people. For a fully connected network, permutations of people do not change the structure of connections, hence there are exponentially many symmetries, which explains why lumping is so efficient. However, even minor modifications of the network such as stepping away from the homogeneous structure of connections, destroy the existing symmetries and make lumping impossible to apply.

Alternative approaches include mean–field approximations [5,6], effective degree models [7–9], and edge–based compartmental models [10]. These models are approximate and rely on truncation of the state space, effects of which on accuracy are difficult to estimate and/or keep below a desired tolerance for a general network.

In this paper we apply a tensor product solver [19] to integrate the high–dimensional system of ODEs (4) and hence compute, approximately but with controlled accuracy, the joint probability distribution $\mathbf{p}$. The accuracy of approximations introduced by the proposed method is controlled by a single threshold parameter ϵ which can be set accordingly to the desired precision. The method automatically adjusts parameters controlling the complexity of approximation (so-called *ranks*) to match the desired accuracy. It does not explicitly rely on network to have symmetries, and thus can be applied to a general network, although the ranks and associated computational costs are network–dependent and may grow uncontrollably for large and densely connected networks.

2.3. Stochastic simulation algorithm

In contrast to previous methods, the classical Gillespie's stochastic simulation algorithm (SSA) [52] does not attempt to solve the ODEs (4). Instead, it simulates a course of epidemic by sampling random walks $\mathrm{x}(t)$ through the state space Ω until some desired time T. The sampled trajectories are assumed piecewise–constant, i.e. $x(t) = x_k$, for $t \in [t_k, t_{k+1})$. Starting with $k = 0, t_0 = 0$, and the initial state x_0, SSA uses Monte Carlo sampling to simulate when a next reaction will occur, and which particular reaction will occur.

1. Calculate all nonzero transition probabilities (*propensities*) $p_{\mathrm{x}_k \to \mathrm{y}}$ for $\mathrm{y} \in \Omega$.
2. Calculate the total propensity $p_\Omega = \sum_{\mathrm{y} \in \Omega} p_{\mathrm{x}_k \to \mathrm{y}}$.
3. Sample a time step τ from exponential distribution with rate parameter p_Ω.
4. Sample a new state $\mathrm{y}_\star \in \Omega$ from the discrete distribution with probabilities $p_{\mathrm{x}_k \to \mathrm{y}} / p_\Omega$.
5. Implement the next step of the walk by setting $\mathrm{x}_{k+1} = \mathrm{y}_\star$, and $t_{k+1} = t_k + \tau$.
6. If $t_{k+1} < T$, set $k := k + 1$ and repeat from step 1, otherwise stop.

Running this algorithm N_{SSA} times, we obtain N_{SSA} sample paths, which can be used to estimate any expectations over the probability distribution defined by the master equation (4). Suppose that a quantity of interest $Q(\mathrm{x}, t)$ is a function depending on the state and/or time. The expectation of Q can be approximated as follows,

$$\mathsf{E}[Q(t)] \approx \frac{1}{N_{\mathrm{SSA}}} \sum_{s=1}^{N_{\mathrm{SSA}}} Q(\mathrm{x}_k^{(s)}, t_k^{(s)}), \qquad t \in [t_k^{(s)}, t_{k+1}^{(s)}),$$

where $(t_k^{(s)}, \mathrm{x}_k^{(s)})$ represent the s–th randomly sampled trajectory. In other words, the SSA performs a piecewise constant interpolation of the state in time, followed by the Monte Carlo estimator over the sample paths.

Naturally, this estimate contains a statistical error. Following the central limit theorem, we can conclude that if the variance of the quantity of interest, $\mathsf{V}[Q]$, is finite, the variance of the estimator of $\mathsf{E}[Q]$ is $\mathsf{V}[Q]/N_{\mathrm{SSA}}$. The relative error in the estimate is thus proportional to $(\sqrt{\mathsf{V}[Q]}/|\mathsf{E}[Q]|) \cdot (1/\sqrt{N_{\mathrm{SSA}}})$, which can be very large if $\sqrt{\mathsf{V}[Q]} \gg |\mathsf{E}[Q]|$. To compensate for this, a very large N_{SSA} is needed, which leads to enormous computational costs. This happens for example in estimation of probabilities of rare events. In this case $Q(\mathrm{x}, t)$ is an indicator function of the event of interest, with $\mathsf{E}[Q] \ll 1$.

Alternative algorithms include for example Tau-Leaping [4] and multi-level simulations [29,30]. The Tau-Leaping method fixes a time step τ, and samples (possibly several) reactions within this time step from a Poisson distribution. Clearly, the pre-selected time step τ can be larger than the time steps sampled by SSA, which requires fewer steps in total. However, Tau-Leaping samples biased trajectories, with the bias increasing with τ [53]. Multi-level algorithms allow one to compensate for more time steps resulting from a small τ by sampling less trajectories, and vice versa. This alleviates the problem of sampling fast reactions with small time steps. However, these methods may still struggle with high variance of the quantity of interest, $\sqrt{\mathsf{V}[Q]} \gg |\mathsf{E}[Q]|$.

3. Methods

In this section we introduce tensor product approach to solving CME for epidemics on network.

3.1. Chemical master equation for the network SIR model

The matrix $\mathbf{A}$ in the master equation (4) has exponentially large size $3^N \times 3^N$, which makes classical algorithms struggle from the curse of dimensionality. Fortunately, it has a hidden tensor product structure, which we will reveal and exploit to solve the problem using tensor product algorithms.

Firstly, note that the right–hand side in (3) contains sums over $|\Omega| = 3^N$ states y. However, for a given state $\mathrm{x} \in \Omega$ most of the transitions $\mathrm{x} \to \mathrm{y}$ and $\mathrm{y} \to \mathrm{x}$ are impossible, i.e. $p_{\mathrm{x} \to \mathrm{y}} = 0$ and $p_{\mathrm{y} \to \mathrm{x}} = 0$. We will rewrite sums in a more explicit form by keeping only possible transitions.

From now on we will denote the states of individual nodes (1) using numbers, $x_n \in \{1, 2, 3\}$. Note that an infection $\mathrm{x} \to \mathrm{y}$ which makes a susceptible person $x_n = 1$ infected $y_n = 2$ can be written as $\mathrm{y} = \mathrm{x} + \mathrm{e}_n$ where $\mathrm{e}_n \in \mathbb{R}^N$ is the n-th unit vector. A recovery $\mathrm{x} \to \mathrm{y}$ which makes an infected person $x_n = 2$ recovered $y_n = 3$ also can be written as $\mathrm{y} = \mathrm{x} + \mathrm{e}_n$. Hence, both the infection and recovery reactions are described by the *stoichiometry* 1. This means that $p_{\mathrm{y} \to \mathrm{x}} = 0$ unless $\exists n \in \mathcal{V} : \mathrm{x} = \mathrm{y} + \mathrm{e}_n$, and $p_{\mathrm{x} \to \mathrm{y}} = 0$ unless $\exists n \in \mathcal{V} : \mathrm{y} = \mathrm{x} + \mathrm{e}_n$. Hence we can rewrite (3) as follows

$$
\begin{aligned}
p'(\mathrm{x},t) &= \sum_{n=1}^{N} \left[p_{\mathrm{y}\to\mathrm{x}} \cdot p(\mathrm{y},t)\right]_{\mathrm{x}=\mathrm{y}+\mathrm{e}_n} - \sum_{n=1}^{N} \left[p_{\mathrm{x}\to\mathrm{y}} \cdot p(\mathrm{x},t)\right]_{\mathrm{y}=\mathrm{x}+\mathrm{e}_n} \\
&= \sum_{n=1}^{N} \underbrace{p_{(\mathrm{x}-\mathrm{e}_n)\to\mathrm{x}}}_{a_n(\mathrm{x}-\mathrm{e}_n)} \cdot p(\mathrm{x}-\mathrm{e}_n,t) - \sum_{n=1}^{N} \underbrace{p_{\mathrm{x}\to(\mathrm{x}+\mathrm{e}_n)}}_{a_n(\mathrm{x})} \cdot p(\mathrm{x},t),
\end{aligned} \tag{6}
$$

keeping only N terms in each sum and introducing notation $a_n(\mathrm{x}) = p_{\mathrm{x}\to(\mathrm{x}+\mathrm{e}_n)}$ for reaction rates of stoichiometry 1. This form of the master equation is often called *chemical master equation* (CME) following [4]. As shown in (2), the specific formula for the reaction rate $a_n(\mathrm{x})$ depends on whether this reaction is infection or recovery, which in turn depends on the value of x_n. Using an *indicator* function

$$
\mathbf{1}_{condition} = \begin{cases} 1, & \text{if } condition \text{ is true} \\ 0, & \text{if } condition \text{ is false,} \end{cases}
$$

we rewrite (2) as follows:

$$
\begin{aligned}
a_n(\mathrm{x}) = p_{\mathrm{x}\to(\mathrm{x}+\mathrm{e}_n)} &= \mathbf{1}_{x_n=1} \cdot p^{(\mathrm{inf})}_{\mathrm{x}\to(\mathrm{x}+\mathrm{e}_n)} + \mathbf{1}_{x_n=2} \cdot p^{(\mathrm{rec})}_{\mathrm{x}\to(\mathrm{x}+\mathrm{e}_n)} \\
&= \mathbf{1}_{x_n=1} \cdot I_n(x)\beta + \mathbf{1}_{x_n=2} \cdot \gamma \\
&= \sum_{m\sim n} \underbrace{\beta \mathbf{1}_{x_n=1}\mathbf{1}_{x_m=2}}_{a^{(\mathrm{inf})}_{m\to n}(\mathrm{x})} + \underbrace{\gamma \mathbf{1}_{x_n=2}}_{a^{(\mathrm{rec})}_{n}(\mathrm{x})},
\end{aligned} \tag{7}
$$

where $I_n(\mathrm{x}) = \sum_{m\sim n} \mathbf{1}_{x_m=2}$ counts infected neighbours of person n.

All ODEs (6) taken together form the master equation with the vector of unknowns $\mathbf{p}(t) = [p(\mathrm{x},t)]_{\mathrm{x}\in\Omega}$, where the probability $p(\mathrm{x},t)$ of the state $\mathrm{x} = (x_1, x_2, \dots, x_N)$ appears as element with index

$$
\overline{x_1 x_2 \dots x_N} = 3^{N-1}(x_1 - 1) + 3^{N-2}(x_2 - 1) + \cdots + 3^0 x_N.
$$

With this *big-endian* ordering, the vector $\mathbf{a}^{(\mathrm{rec})}_n = [a^{(\mathrm{rec})}_n(\mathrm{x})]_{\mathrm{x}\in\Omega}$ from (7) can be written as

$$
\mathbf{a}^{(\mathrm{rec})}_n = \gamma \vec{e} \otimes \cdots \otimes \vec{e} \otimes \vec{\imath} \otimes \vec{e} \otimes \cdots \otimes \vec{e}, \tag{8}
$$

where $\vec{\imath} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^T$ appears in position n, $\vec{e} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T$ appear in all positions $1, \dots, N$ except n, and $\otimes$ denotes Kronecker (tensor) product. Recall that the Kronecker product $C = A \otimes B$ for $A \in \mathbb{R}^{p\times q}$ and $B \in \mathbb{R}^{m\times n}$ is a $pm \times qn$ matrix with elements $C(i + (j-1)m, k + (\ell - 1)n) = A(j,\ell)B(i,k)$. The Kronecker product is distributive and associative.

Similarly, vector $\mathbf{a}^{(\mathrm{inf})}_{m\to n} = [a^{(\mathrm{inf})}_{m\to n}(\mathrm{x})]_{\mathrm{x}\in\Omega}$ from (7) can be written as

$$
\mathbf{a}^{(\mathrm{inf})}_{m\to n} = \beta \vec{e} \otimes \cdots \otimes \vec{e} \otimes \vec{s} \otimes \vec{e} \otimes \cdots \otimes \vec{e} \otimes \vec{\imath} \otimes \vec{e} \otimes \cdots \otimes \vec{e}, \tag{9}
$$

where $\vec{s} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T$ appears in position n, $\vec{\imath} = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^T$ appear in positions $m \sim n$, $\vec{e} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T$ appear elsewhere. Note that appearance of basis vectors $\vec{s}$ and $\vec{\imath}$ realises conditions of indicator functions $\mathbf{1}_{x_n=1}$ and $\mathbf{1}_{x_m=2}$ in (7), and $\vec{e}$ appears in positions of nodes not affected by any conditions. Now the vector $[a_n(\mathrm{x})p(\mathrm{x},t)]_{\mathrm{x}\in\Omega}$ in the second term of (6) can be written as

$$
\begin{aligned}
&\mathrm{diag}(\mathbf{a}_n)\mathbf{p} = \left(\sum_{m\sim n} \mathrm{diag}(\mathbf{a}^{(\mathrm{inf})}_{m\to n}) + \mathrm{diag}(\mathbf{a}^{(\mathrm{rec})}_n) \right) \mathbf{p}, \\
&\mathrm{diag}(\mathbf{a}^{(\mathrm{inf})}_{m\to n}) = \beta \cdot \mathrm{Id} \otimes \cdots \otimes \mathrm{Id} \otimes \mathrm{diag}(\vec{s}) \otimes \mathrm{Id} \otimes \cdots \otimes \mathrm{Id} \otimes \mathrm{diag}(\vec{\imath}) \otimes \mathrm{Id} \otimes \cdots \otimes \mathrm{Id}, \\
&\mathrm{diag}(\mathbf{a}^{(\mathrm{rec})}_n) = \gamma \cdot \mathrm{Id} \otimes \cdots \otimes \mathrm{Id} \otimes \mathrm{diag}(\vec{\imath}) \otimes \mathrm{Id} \otimes \cdots \otimes \mathrm{Id},
\end{aligned} \tag{10}
$$

where $\mathrm{Id} = \mathrm{diag}(\vec{e})$ is a 3×3 identity matrix.

The first term in the right–hand side of (6) contains the probability of the shifted state $p(\mathrm{x} - \mathrm{e}_n, t)$ which we can express in terms of probabilities $p(\mathrm{y},t)$ as follows

$$
\begin{aligned}
q(\mathrm{x},t) &= p(\mathrm{x}-\mathrm{e}_n,t) \\
&= \sum_{\mathrm{y}\in\Omega} \mathbf{1}_{x_1=y_1} \cdots \mathbf{1}_{x_{n-1}=y_{n-1}} \cdot \mathbf{1}_{x_n-1=y_n} \cdot \mathbf{1}_{x_{n+1}=y_{n+1}} \cdots \mathbf{1}_{x_N=y_N} \cdot p(\mathrm{y},t), \\
\mathbf{q}(t) &= \underbrace{\left(\mathrm{Id} \otimes \cdots \otimes \mathrm{Id} \otimes \boldsymbol{J}^T \otimes \mathrm{Id} \otimes \cdots \otimes \mathrm{Id}\right)}_{\mathbf{J}^T_n} \mathbf{p}(t),
\end{aligned} \tag{11}
$$

where the *shift matrix* $J^T = \begin{pmatrix} 0&0&0 \\ 1&0&0 \\ 0&1&0 \end{pmatrix}$ appears in position n, and the identity matrix $\mathrm{Id} = \begin{pmatrix} 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{pmatrix}$ appears elsewhere. Because of this special structure, we say that the $3^N \times 3^N$ matrix $\mathbf{J}^T_n$ acts on n–th site of the system only.

The same process can be applied to the vector $[a_n(\mathrm{x}-\mathrm{e}_n)p(\mathrm{x}-\mathrm{e}_n,t)]_{\mathrm{x}\in\Omega}$ in the first term of (6), which gives

$$\mathbf{p}' = \sum_{n=1}^{N} \mathbf{J}_n^T \operatorname{diag}(\mathbf{a}_n)\mathbf{p} - \sum_{n=1}^{N} \operatorname{diag}(\mathbf{a}_n)\mathbf{p}.$$

Plugging in the tensor product expansion (10), we obtain the matrix of the master equation (4) in *tensor product* form

$$\begin{aligned}\mathbf{A} = &\sum_{n=1}^{N}\sum_{m\sim n} \beta\cdot \mathrm{Id}\otimes\cdots\otimes\mathrm{Id}\otimes(J^T-\mathrm{Id})\operatorname{diag}(\vec{s})\otimes\mathrm{Id}\otimes\cdots\otimes\mathrm{Id}\otimes\operatorname{diag}(\vec{\imath})\otimes\mathrm{Id}\otimes\cdots\otimes\mathrm{Id}\\ &+\sum_{n=1}^{N}\gamma\cdot\mathrm{Id}\otimes\cdots\otimes\mathrm{Id}\otimes(J^T-\mathrm{Id})\operatorname{diag}(\vec{\imath})\otimes\mathrm{Id}\otimes\cdots\otimes\mathrm{Id}.\end{aligned} \tag{12}$$

The tensor product representation of matrices $\mathbf{J}_n^T$ in (11) and $\mathbf{A}$ in (12) allows us to define large $3^N\times 3^N$ matrices as a tensor product of small 3×3 matrices acting on individual sites of the system. By working with matrices and vectors is this form we can avoid computing them explicitly, hence reducing storage requirements significantly. For example, a $3^N\times 3^N$ matrix $\mathbf{J}_n^T$ has $2\cdot 3^{N-1}$ nonzero elements, hence storing it in full (but sparse) form requires $\mathcal{O}(3^N)$ memory. However the factors of tensor product in (11) have $3N-1$ nonzero elements in total, hence we can keep the factorised matrix $\mathbf{J}_n^T$ using $\mathcal{O}(3N)$ storage. Similarly, matrix $\mathbf{A}$ in (12) is represented by $(|\mathcal{E}|+N)$ tensor product terms, reducing total storage from $\mathcal{O}(3^N)$ to $\mathcal{O}(3|\mathcal{E}|N+3N^2)=\mathcal{O}(3(\langle k\rangle+1)N^2)$, where $\langle k\rangle=|\mathcal{E}|/|\mathcal{V}|$ is the average degree of the network.

3.2. Tensor product factorisations

Maintaining the factorised tensor product form for the matrix $\mathbf{A}$ of the chemical master equation (12), we remove the *curse of dimensionality* for storage of $\mathbf{A}$. To similarly reduce the storage and computational costs for the unknown probability distribution function $\mathbf{p}(t)$ and make the numerical solution possible, we need to assume a tensor product representation for $\mathbf{p}(t)$ to hold exactly or approximately with sufficiently good accuracy. For the sake of simplicity, let's first drop the dependency on t and consider a p.d.f. $\mathbf{p}=[p(\mathrm{x})]_{\mathrm{x}\in\Omega}$. Since $\Omega=\mathbb{X}^N$, we can consider $\mathbf{p}$ as a vector of size $|\Omega|=3^N$, or, equivalently, as a *tensor* $\mathbf{p}=[p(x_1,x_2,\dots,x_N)]$ of size $3\times 3\times\cdots\times 3$. Similar to (12), we want to factorise $\mathbf{p}$ into a product of factors, each representing a single node of the network (*aka* site of the system).

The simplest approach would be to factorise $\mathbf{p}$ as a sum of tensor products,

$$\mathbf{p}\approx\tilde{\mathbf{p}}=\sum_{\alpha=1}^{R}\mathbf{p}_\alpha^{[1]}\otimes\cdots\otimes\mathbf{p}_\alpha^{[N]}. \tag{13}$$

This decomposition, known as the *canonical polyadic* (CP) format [54], factorises a given tensor into $3\times R$ matrices $\mathbf{p}^{[n]}=\left[p_\alpha^{[n]}(x_n)\right]$, known as CP factors. If all high–dimensional tensors are kept in tensor product format, all computations can be performed with one–site factors instead of full vectors and matrices, lifting the curse of dimensionality. Unfortunately, the CP format (13) can be unstable and the best approximation does not always exist [55], which makes it less attractive.

A different tensor product format that admits stable computations is the *tensor train* (TT) decomposition [12]. We say that a $3\times 3\times\cdots\times 3$ tensor $\mathbf{p}$, and equivalently its vectorisation $\mathrm{vec}(\mathbf{p})$ of size 3^N, are approximated in TT format, if

$$\mathbf{p}\approx\tilde{\mathbf{p}}=\sum_{\alpha_0,\dots,\alpha_N=1}^{r_0,\dots,r_N}\mathbf{p}^{(1)}_{\alpha_0,\alpha_1}\otimes\cdots\otimes\mathbf{p}^{(n)}_{\alpha_{n-1},\alpha_n}\otimes\cdots\otimes\mathbf{p}^{(N)}_{\alpha_{N-1},\alpha_N}. \tag{14}$$

Here, the $r_{n-1}\times 3\times r_n$ factors $\mathbf{p}^{(n)}=\left[\mathbf{p}^{(n)}_{\alpha_{n-1},\alpha_n}(x_n)\right]$, $n=1,\dots,N$, are called TT cores, and the ranges of the summation indices $r_0,\dots,r_N$ are called *TT ranks*. Each core $\mathbf{p}^{(n)}$ contains information related to person n in the network, and the summation indices α_{n-1},α_n of core $\mathbf{p}^{(n)}$ link it to cores $\mathbf{p}^{(n-1)}$ and $\mathbf{p}^{(n+1)}$. This linear arrangement of tensor train cores explains the name of the format.

In contrast to the CP format, the TT ranks r_n are ranks of $3^n\times 3^{N-n}$ *unfolding* matrices, i.e. reshapes of the original tensor, where the first n indices merge to form a multi-index for column, and the remaining $N-n$ indices merge to form a multi-index for row,

$$r_n=\operatorname{rank}P^{[n]},\quad P^{[n]}=\left[\tilde{p}(\overline{x_1\dots x_n},\overline{x_{n+1}\dots x_N})\right],\quad \tilde{p}(\overline{x_1\dots x_n},\overline{x_{n+1}\dots x_N})=\tilde{p}(x_1,\dots,x_N).$$

This ensures the existence of best approximation in the TT format with the TT ranks $r_n\leqslant\min(3^n,3^{N-n})$, for $n=0,\dots,N$, and, in particular, $r_0=r_N=1$.

Since TT ranks are matrix ranks, the approximation in (14) is usually considered with respect to Frobenius norm. Recall that for matrices (i.e. $N=2$) the best low-rank approximation in Frobenius norm can be computed in a stable and robust way using the (truncated) singular value decomposition (SVD) algorithm [56]. For small N, when the tensor $\mathbf{p}$ can be available in full, the TT approximation $\tilde{\mathbf{p}}_{\mathrm{TT-SVD}}$ can be computed by TT-SVD algorithm [12] with quasi–optimal accuracy

$$\|\tilde{\mathbf{p}}_{\mathrm{TT-SVD}}-\mathbf{p}\|_F\leqslant\sqrt{N-1}\|\tilde{\mathbf{p}}_{\mathrm{best}}-\mathbf{p}\|_F,$$

where $\tilde{\mathbf{p}}_{\text{best}}$ is the best possible TT approximation of the same TT ranks, and the quasi–optimality factor $\sqrt{N-1}$ depends only on the dimensionality of the tensor. For larger dimensions, the TT approximations can be computed by alternating least square optimisation over the TT cores as shown in [57] and also in earlier work on matrix product states (MPS) [58,21] in context of quantum physics.

A general tensor $\mathbf{p}$ can have the maximal TT ranks $r_n = \min(3^n, 3^{N-n})$ which are exponential in N. On the other hand, if $x_1, \dots, x_N$ were independent random variables, the p.d.f. $\mathbf{p}$ would be a direct product of univariate distributions, $p(x_1, \dots, x_N) = p^{(1)}(x_1) \cdots p^{(N)}(x_N)$, which is a TT format (14) with TT ranks equal to one, $r_0 = r_1 = \dots = r_N = 1$. Our approach is aimed at intermediate scenarios with *weak correlations* between *far* variables x_n and x_m for $|n-m| \gg 1$, in which case $\mathbf{p}$ may lend itself to a TT approximation with moderate TT ranks. For example, the multivariate normal distribution with low-rank off-diagonal blocks of the precision matrix was shown to admit a TT approximation with TT ranks growing poly-logarithmically with error [59]. Existence of TT approximations with $r = \mathcal{O}(N)$ was proven for stationary distributions describing mass-action and Michaelis–Menten kinetics [60]. When TT ranks are bounded by $r_n \leqslant r \ll \min(3^n, 3^{N-n})$, we can represent the p.d.f. using only $\mathcal{O}(Nr^2) \ll 3^N$ elements of the TT approximation (14), lifting the curse of dimensionality.

3.3. Discretisation in time

We discretise the dynamical system (4) on $t \in [0, T]$, where T is the desired time horizon. We introduce a set of *reference* time points $\{t_k\}_{k=1}^K$, such that $0 = t_0 < t_1 < \dots < t_K = T$. These can be the points where the solution is ultimately sought, or they can be determined adaptively to control the discretisation error [61]. On each subinterval $(t_{k-1}, t_k]$ we introduce a basis of Lagrange polynomials $\{\varphi_\ell^{(k)}(t)\}_{\ell=1}^L$ centred at Chebyshëv nodes

$$t_\ell^{(k)} = \tfrac{1}{2}(t_{k-1} + t_k) + \tfrac{1}{2}(t_k - t_{k-1}) \cos\left(\pi \tfrac{\ell - 1}{L}\right), \qquad \ell = 1, \dots, L. \tag{15}$$

The p.d.f. can now be approximated as

$$\mathbf{p}(t) \approx \sum_{\ell=1}^{L} \mathbf{p}(t_\ell^{(k)}) \cdot \varphi_\ell^{(k)}(t), \qquad t \in (t_{k-1}, t_k],$$

which is known as *spectral approximation*. Since $\mathbf{p}(t)$ is a solution of linear autonomous system of ODEs (4), it is analytic, hence the best spectral approximation converges exponentially in L, see [62, Ch. 5]. On each subinterval we want to compute all values $p(\mathrm{x}, t_\ell^{(k)})$, forming a vector $\bar{\mathbf{p}}^{(k)} = [p(x_1, x_2, \dots, x_N, t_\ell^{(k)})] \in \mathbb{R}^{3^N L}$, which can be also seen as a $3 \times 3 \times \dots \times 3 \times L$ tensor with $(N+1)$ indices. The TT decomposition (14) is expanded accordingly:

$$\bar{\mathbf{p}}^{(k)} \approx \tilde{\mathbf{p}}^{(k)} = \sum_{\alpha_0, \dots, \alpha_{N+1}=1}^{r_0, \dots, r_{N+1}} \mathbf{p}_{\alpha_0, \alpha_1}^{(k,1)} \otimes \dots \otimes \mathbf{p}_{\alpha_{n-1}, \alpha_n}^{(k,n)} \otimes \dots \otimes \mathbf{p}_{\alpha_{N-1}, \alpha_N}^{(k,N)} \otimes \mathbf{p}_{\alpha_N, \alpha_{N+1}}^{(k,N+1)}, \tag{16}$$

where now $r_{N+1} = 1$, $r_N \geqslant 1$ (in general), and the new TT core $\mathbf{p}_{\alpha_N, \alpha_{N+1}}^{(k,N+1)} \in \mathbb{R}^L$ encodes the dependence on time. This p.d.f. can now be interpolated at any $t \in (t_{k-1}, t_k]$ as

$$\mathbf{p}(t) \approx \tilde{\mathbf{p}}(t) = \sum_{\alpha_0, \dots, \alpha_{N+1}} \mathbf{p}_{\alpha_0, \alpha_1}^{(k,1)} \otimes \dots \otimes \mathbf{p}_{\alpha_{n-1}, \alpha_n}^{(k,n)} \otimes \dots \otimes \mathbf{p}_{\alpha_{N-1}, \alpha_N}^{(k,N)} \cdot \left(\sum_{\ell=1}^{L} \mathbf{p}_{\alpha_N, \alpha_{N+1}}^{(k,N+1)}(\ell) \cdot \varphi_\ell^{(k)}(t) \right). \tag{17}$$

The time derivative is replaced by a *differentiation matrix* $D^{(k)} = [(\varphi_{\ell'}^{(k)})'(t_\ell^{(k)})]_{\ell, \ell'=1}^L$. This allows us to propagate the master equation (4) through the interval $(t_{k-1}, t_k]$ by solving a linear equation

$$\underbrace{\left(\mathrm{Id}_{3^N} \otimes D^{(k)} - \mathbf{A} \otimes \mathrm{Id}_L\right)}_{\mathbf{A}^{(k)}} \tilde{\mathbf{p}}^{(k)} = \underbrace{\mathbf{p}(t_{k-1}) \otimes (D^{(k)} \mathrm{e}_L)}_{\mathbf{f}^{(k)}}, \tag{18}$$

where Id_n is $n \times n$ identity matrix, and e_n is the vector of all ones of size n. The initial state $\mathbf{p}(t_{k-1})$ is taken as the initial condition $\mathbf{p}(0)$ if $k = 1$ or interpolated from the previous subinterval using (17). Plugging in (12) and noticing that $\mathrm{Id}_{3^N} = \mathrm{Id} \otimes \dots \otimes \mathrm{Id}$, the matrix $\mathbf{A}^{(k)}$ can be also written in a tensor product form similar to (12) but with one extra term. In the same way, if $\mathbf{p}(t_{k-1})$ is replaced by the TT decomposition (17), the right hand side $\mathbf{f}^{(k)}$ can be written as a TT decomposition as well.

3.4. Tensor product algorithms for solving linear systems

To solve the linear system (18) we can use the Alternating Linear Scheme (ALS) algorithm [57], or the more robust Alternating Minimal Energy (AMEn) algorithm [15]. The basic ALS algorithm solves (18) by iterating over $n = 1, \dots, N+1$, fixing in each step all TT cores in (16) but $\mathbf{p}^{(k,n)}$, and solving the resulting over-determined system for the elements of $\mathbf{p}^{(k,n)}$. This can be seen by vectorising the TT core $\mathbf{p}^{(k,n)}$, i.e. reshaping it into a long vector

$$\mathrm{p}^{(k,n)} = \mathrm{vec}(\mathbf{p}^{(k,n)}) = [\mathbf{p}_{\alpha_{n-1}, \alpha_n}^{(k,n)}(x_n)].$$

We then introduce a *frame* matrix

$$\mathbf{P}^{(k)}_{\neq n} = \left(\sum_{\alpha_0 \dots \alpha_{n-2}} \mathbf{p}^{(k,1)}_{\alpha_0,\alpha_1} \otimes \dots \otimes \mathbf{p}^{(k,n-1)}_{\alpha_{n-2},:} \right) \otimes \mathrm{Id} \otimes \left(\sum_{\alpha_{n+1} \dots \alpha_{N+1}} \mathbf{p}^{(k,n+1)}_{:,\alpha_{n+1}} \otimes \dots \otimes \mathbf{p}^{(k,N+1)}_{\alpha_N,\alpha_{N+1}} \right),$$

which is of size $3^N L \times 3 r_{n-1} r_n$ for $n \leqslant N$, and of size $3^N L \times r_N L$ for $n = N+1$. Now the TT decomposition (16) can be written as a linear map $\tilde{\mathbf{p}}^{(k)} = \mathbf{P}^{(k)}_{\neq n} \mathrm{p}^{(k,n)}$. The ALS method performs the Galerkin projection to solve a reduced linear system

$$((\mathbf{P}^{(k)}_{\neq n})^T \mathbf{A}^{(k)} \mathbf{P}^{(k)}_{\neq n}) \mathrm{p}^{(k,n)} = (\mathbf{P}^{(k)}_{\neq n})^T \mathbf{f}^{(k)} \tag{19}$$

subsequently for $n = 1, \dots, N+1$. This system can be assembled and solved efficiently [57] because $\mathbf{P}^{(k)}_{\neq n}$, $\mathbf{A}^{(k)}$ and $\mathbf{f}^{(k)}$ are available in TT format. The total complexity is $\mathcal{O}(N r^3)$. However, this simple ALS method has two drawbacks: TT ranks are fixed from the beginning (and may not match the ranks required for the unknown solution), and the sequential optimisation process may stuck in a local minimum corresponding to an inaccurate solution.

The AMEn method [15] circumvents these issues by computing TT approximations of both the solution $\tilde{\mathbf{p}}^{(k)}$ and the residual $\tilde{\mathbf{z}}^{(k)} \approx \mathbf{z}^{(k)} = \mathbf{f}^{(k)} - \mathbf{A}^{(k)} \tilde{\mathbf{p}}^{(k)}$. This can be done using the TT-SVD truncation algorithm [12], but in practice the approximation can be obtained more effectively using standard ALS algorithm [23,57] that minimises the Frobenius norm of the error $\|\tilde{\mathbf{z}}^{(k)} - \mathbf{z}^{(k)}\|_F$ by sequential optimisations over the TT cores of $\tilde{\mathbf{z}}^{(k)}$. The obtained information about the residual is used in AMEn to expand the search space by replacing the frame matrices in (19) with

$$\mathbf{P}^{(k)}_{\neq n} = \left(\sum_{\alpha_0 \dots \alpha_{n-2}} \mathbf{p}^{(k,1)}_{\alpha_0,\alpha_1} \otimes \dots \otimes \begin{bmatrix} \mathbf{p}^{(k,n-1)}_{\alpha_{n-2},:} & \mathbf{z}^{(k,n-1)}_{\alpha_{n-2},:} \end{bmatrix} \right) \otimes \mathrm{Id} \otimes \left(\sum_{\alpha_{n+1} \dots \alpha_{N+1}} \mathbf{p}^{(k,n+1)}_{:,\alpha_{n+1}} \otimes \dots \otimes \mathbf{p}^{(k,N+1)}_{\alpha_N,\alpha_{N+1}} \right).$$

Adding extra vectors to the frame matrix allows one to increase the TT rank r_{n-1} if it was underestimated. On the other hand, truncating the singular values of $\mathbf{p}^{(k,n-1)}$ below the desired error threshold [12], one can reduce TT ranks if they are overestimated. Injection of the residual direction in the optimisation process ensures global convergence of the AMEn algorithm [15] to the solution of (18) and enhances its convergence rate in practical computations [17]. Remarkably, the approximation $\tilde{\mathbf{z}}^{(k)} \approx \mathbf{z}^{(k)}$ does not have to be very accurate — even a rough approximation $\|\tilde{\mathbf{z}}^{(k)} - \mathbf{z}^{(k)}\|_F \leqslant \frac{1}{2} \|\mathbf{z}^{(k)}\|_F$ would suffice to guarantee global convergence of AMEn [15].

3.5. Time step adaptation for local error control

Finally, the pseudospectral time discretisation (18), combined with the TT decomposition (16), allows one to estimate the time discretisation error in a computationally efficient way, and hence to adapt the reference time points t_k using standard local error control methods. We check the accuracy of the computed solution $\tilde{\mathbf{p}}^{(k)}$, on a finer Chebyshëv grid $\hat{t}^{(k)}_\ell \in (t_{k-1}, t_k]$, given by (15) with $2L$ nodes.

We construct the linear system (18) on the finer grid $\{\hat{t}^{(k)}_\ell\}_{\ell=1}^{2L}$, plug in the solution $\tilde{\mathbf{p}}(t)$, interpolated from the current grid to the finer grid using (17), and evaluate the residual, which can be done efficiently due to the TT and Kronecker product structures. If the residual norm exceeds the desired error threshold, the current solution is rejected, the time step is reduced, and the solution is recomputed on a smaller interval $(t_{k-1}, t_k]$. If the residual norm is well below the desired efficiency threshold, the size of the next time interval is increased. This error control mechanism is implemented in the tAMEn (time–dependent AMEn) algorithm [19], which we use in the numerical experiments.

3.6. Evaluation of observables

When the p.d.f. $\mathbf{p}(t)$ is computed in the TT format, we can evaluate observables, such as $\mathsf{E}[I(t)]$ and $\mathsf{V}[I(t)]$ in (5). However, we need to avoid taking sums over 3^N network states $\mathrm{x} \in \Omega$ and use more efficient strategy exploiting the properties of the TT format (14).

As an introductory example, suppose that we solved CME (4) and obtained $\mathbf{p}(t)$ in TT format (16). The representation $\mathbf{p}(t)$ is essentially a sequence of arrays (16) spanning intervals $(t_{k-1}, t_k]$. It is therefore sufficient to discuss how to calculate observables at a particular time t, and then repeat the procedure for all intervals, thus covering the entire integration region $(0, T]$. Suppose we obtained $\mathbf{p}(t) \approx \tilde{\mathbf{p}}$ in the TT format (14). We may be interested in calculating total probability $\sum_{\mathrm{x} \in \Omega} \tilde{p}(\mathrm{x})$, which equals 1 in theory, but may slightly deviate due to discretisation errors during numerical integration of (4) and approximation of the solution in TT format. Note that

$$\sum_{\mathrm{x} \in \Omega} \tilde{p}(\mathrm{x}) = \sum_{\alpha_1, \dots, \alpha_{N-1}} \left[\left(\sum_{x_1 \in \mathbb{X}} \mathbf{p}^{(1)}_{\alpha_1}(x_1) \right) \cdots \left(\sum_{x_{N-1} \in \mathbb{X}} \mathbf{p}^{(N-1)}_{\alpha_{N-2},\alpha_{N-1}}(x_{N-1}) \right) \left(\sum_{x_N \in \mathbb{X}} \mathbf{p}^{(N)}_{\alpha_{N-1}}(x_N) \right) \right]. \tag{20}$$

Computing univariate sums inside round brackets costs $\mathcal{O}(N r^2)$ operations, where r denotes the largest TT rank of $\tilde{\mathbf{p}}$. Summation over $\alpha = (\alpha_1, \dots, \alpha_{N-1})$, can be implemented as a sequence of matrix products [12] in another $\mathcal{O}(N r^2)$ operations. Hence, the total complexity is no longer exponential, but linear in number of people in the network. Normalising the p.d.f. $\mathbf{p} = \tilde{\mathbf{p}} / (\sum_{\mathrm{x} \in \Omega} \tilde{p}(\mathrm{x}))$, we make the total probability 1, and proceed to computing observables.

First, let's consider computing the mean

$$\mathsf{E}[I] = \sum_{\mathrm{x} \in \Omega} I(\mathrm{x}) p(\mathrm{x}) = \langle \mathbf{I}, \mathbf{p} \rangle,$$

where $\mathbf{I} = [I(\mathrm{x})]_{\mathrm{x}\in\Omega}$ represents the observable $I(\mathrm{x})$ for all states, and $\langle\cdot,\cdot\rangle$ denotes scalar product. Similarly to (8) and (9), we obtain

$$\begin{aligned} I(\mathrm{x}) &= I(x_1, x_2, \dots, x_N) = \mathbf{1}_{x_1=\mathrm{i}} + \mathbf{1}_{x_2=\mathrm{i}} + \cdots + \mathbf{1}_{x_N=\mathrm{i}}, \\ \mathbf{I} &= \vec{\imath}\otimes\vec{e}\otimes\cdots\otimes\vec{e} + \vec{e}\otimes\vec{\imath}\otimes\cdots\otimes\vec{e} + \cdots + \vec{e}\otimes\vec{e}\otimes\cdots\otimes\vec{\imath}. \end{aligned} \tag{21}$$

We note that the tensor $\mathbf{I}$ admits CP decomposition (13) with tensor rank $R = N$. However, due to a special structure of the rank-one terms the corresponding TT decomposition has TT ranks all equal to two:

$$\mathbf{I} = \sum_{\alpha_1,\dots,\alpha_{N-1}=1}^{2} \begin{bmatrix}\vec{e} & \vec{\imath}\end{bmatrix}_{\alpha_1} \otimes \begin{bmatrix}\vec{e} & \vec{\imath}\\ 0 & \vec{e}\end{bmatrix}_{\alpha_1,\alpha_2} \otimes\cdots\otimes \begin{bmatrix}\vec{e} & \vec{\imath}\\ 0 & \vec{e}\end{bmatrix}_{\alpha_{N-2},\alpha_{N-1}} \otimes \begin{bmatrix}\vec{\imath}\\ \vec{e}\end{bmatrix}_{\alpha_{N-1}}. \tag{22}$$

A similar explicit TT representation appears for the high–dimensional Laplace [63], and diffusion [64] operators for high–dimensional PDEs. Note also a related work on explicit tensor product representation of Fourier transform operator [65,66].

To compute the variance in (5), we use the formula $\mathsf{V}[I] = \mathsf{E}[I^2] - (\mathsf{E}[I])^2$, for which we need

$$\mathsf{E}[I^2] = \sum_{\mathrm{x}\in\Omega} I^2(\mathrm{x}) p(\mathrm{x}) = \langle \mathbf{I}\odot\mathbf{I}, \mathbf{p}\rangle,$$

where $\odot$ denotes the Hadamard (pointwise) product of vectors. As noted in [67], linear operations between vectors and matrices in tensor product formats can be computed efficiently in the same format. In particular, the Hadamard product of vectors $\mathbf{I}$ and $\mathbf{p}$ can be computed efficiently in TT format with TT ranks of the product being the product of TT ranks of the terms [12]. Since multiplication by $\mathbf{I}$ only doubles the TT ranks, the complexity of evaluating $\langle \mathbf{I}\odot\mathbf{I}, \mathbf{p}\rangle$ is of the same order, $\mathcal{O}(Nr^2)$, as the cost of evaluating $\sum_{\mathrm{x}\in\Omega} p(\mathrm{x})$. However, we can suggest a more elegant explicit formula for the TT factorisation of tensor $\left[I(x)^2\right]_{\mathrm{x}\in\Omega} = \mathbf{I}\odot\mathbf{I}$ with all TT ranks equal to three. From (21) we obtain

$$I(\mathrm{x})^2 = \left(\sum_{n=1}^{N} \mathbf{1}_{x_n=\mathrm{i}}\right)^2 = \underbrace{\sum_{1\leqslant n\leqslant N} \left(\mathbf{1}_{x_n=\mathrm{i}}\right)^2}_{I(\mathrm{x})} + 2\underbrace{\sum_{1\leqslant m<n\leqslant N} \mathbf{1}_{x_m=\mathrm{i}}\mathbf{1}_{x_n=\mathrm{i}}}_{B(\mathrm{x})}. \tag{23}$$

Since $\mathbf{1}^2 = \mathbf{1}$, the first term equals $I(\mathrm{x})$ and the TT decomposition is given by (22). The second term is a sum of $\frac{1}{2}N(N-1)$ rank-one terms $\vec{e}\otimes\cdots\otimes\vec{e}\otimes\vec{\imath}\otimes\vec{e}\otimes\cdots\otimes\vec{e}\otimes\vec{\imath}\otimes\vec{e}\otimes\cdots\otimes\vec{e}$, where $\vec{\imath}$'s appear in positions m and n, $n > m$. Collecting linearly independent terms in each variable similarly to [63], we arrive at a TT representation of ranks three:

$$[B(\mathrm{x})]_{\mathrm{x}\in\Omega} = \sum_{\alpha_1,\dots,\alpha_{N-1}=1}^{3} \begin{bmatrix}\vec{e} & \vec{\imath} & 0\end{bmatrix}_{\alpha_1} \otimes \begin{bmatrix}\vec{e} & \vec{\imath} & 0\\ 0 & \vec{e} & \vec{\imath}\\ 0 & 0 & \vec{e}\end{bmatrix}_{\alpha_1,\alpha_2} \otimes\cdots\otimes \begin{bmatrix}\vec{e} & \vec{\imath} & 0\\ 0 & \vec{e} & \vec{\imath}\\ 0 & 0 & \vec{e}\end{bmatrix}_{\alpha_{N-2},\alpha_{N-1}} \otimes \begin{bmatrix}0\\ \vec{\imath}\\ \vec{e}\end{bmatrix}_{\alpha_{N-1}}.$$

Extending the TT representation (22) to TT rank three by zero-padding the first and the last TT core allows us to represent $I(\mathrm{x})^2 = I(\mathrm{x}) + 2B(\mathrm{x})$ as a TT decomposition of TT ranks three:

$$\mathbf{I}\odot\mathbf{I} = \sum_{\alpha_1,\dots,\alpha_{N-1}=1}^{3} \begin{bmatrix}\vec{e} & \vec{\imath} & 0\end{bmatrix}_{\alpha_1} \otimes \begin{bmatrix}\vec{e} & \vec{\imath} & 0\\ 0 & \vec{e} & \vec{\imath}\\ 0 & 0 & \vec{e}\end{bmatrix}_{\alpha_1,\alpha_2} \otimes\cdots\otimes \begin{bmatrix}\vec{e} & \vec{\imath} & 0\\ 0 & \vec{e} & \vec{\imath}\\ 0 & 0 & \vec{e}\end{bmatrix}_{\alpha_{N-2},\alpha_{N-1}} \otimes \begin{bmatrix}\vec{\imath}\\ 2\vec{\imath}+\vec{e}\\ 2\vec{e}\end{bmatrix}_{\alpha_{N-1}}. \tag{24}$$

In addition to the above, we may want to calculate so-called *exceedance probabilities*

$$\mathsf{P}(I(t)\geqslant I_\star) = \sum_{\mathrm{x}\in\Omega} p(\mathrm{x},t)\mathbf{1}_{I(\mathrm{x})\geqslant I_\star} = \left\langle \mathbf{p}(t), [\mathbf{1}_{I(\mathrm{x})\geqslant I_\star}]_{\mathrm{x}\in\Omega}\right\rangle, \tag{25}$$

with some critical threshold $I_\star$, e.g. related to a hospital capacity. Similar to previous examples, we can evaluate this sum efficiently if we construct a TT representation for $\left[\mathbf{1}_{I(\mathrm{x})\geqslant I_\star}\right]_{\mathrm{x}\in\Omega}$. Since $\mathbf{1}_{I(\mathrm{x})\geqslant I_\star} = \sum_{I=I_\star}^{N} \mathbf{1}_{I(\mathrm{x})=I}$, we start by constructing TT representations for $\left[\mathbf{1}_{I(\mathrm{x})=I}\right]_{\mathrm{x}\in\Omega}$. The states $\mathrm{x} = (x_1, x_2, \dots, x_N)^T$ with $I(\mathrm{x}) = I$ are such that exactly I nodes are infected, and other $N-I$ nodes are not. Extending the technique [63] used to derive (22) and (24), the indicators can be shown to have the following TT representations

$$\begin{aligned} \left[\mathbf{1}_{I(\mathrm{x})=I}\right]_{\mathrm{x}\in\Omega} &= \sum_{\alpha_1,\dots,\alpha_{N-1}=1}^{I+1} \mathbf{u}^{(1)}_{\alpha_1} \otimes\cdots\otimes \mathbf{u}^{(n)}_{\alpha_{n-1},\alpha_n} \otimes\cdots\otimes \mathbf{u}^{(N)}_{\alpha_{N-1}}, \\ \mathbf{u}^{(1)} = \begin{bmatrix}\vec{e}-\vec{\imath} & \vec{\imath} & 0 & \cdots & 0\end{bmatrix}, \quad \mathbf{u}^{(n)} &= \begin{bmatrix} \vec{e}-\vec{\imath} & \vec{\imath} & 0 & \cdots & 0\\ 0 & \vec{e}-\vec{\imath} & \vec{\imath} & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & 0\\ \vdots & & \ddots & \vec{e}-\vec{\imath} & \vec{\imath}\\ 0 & \cdots & \cdots & 0 & \vec{e}-\vec{\imath}\end{bmatrix}, \quad \mathbf{u}^{(N)} = \begin{bmatrix}0\\ \vdots\\ 0\\ \vec{\imath}\\ \vec{e}-\vec{\imath}\end{bmatrix}, \end{aligned} \tag{26}$$

with TT cores for $n = 2, \dots, N-1$ being the same $(I+1)\times 3\times(I+1)$ array, all $(I+1)\times(I+1)$ slices of which are two-diagonal Toeplitz matrices. Note that all TT ranks of this decomposition are equal to $I+1$. For $I(\mathrm{x}) > N/2$ it is more convenient to 'flip' the variables and count how many people are not infected, which leads to the following decomposition

$$\begin{gathered}\left[\mathbf{1}_{I(\mathrm{x})=N-H}\right]_{\mathrm{x}\in\Omega} = \sum_{\alpha_1,\dots,\alpha_{N-1}=1}^{H+1} \mathbf{v}^{(1)}_{\alpha_1} \otimes \cdots \otimes \mathbf{v}^{(n)}_{\alpha_{n-1},\alpha_n} \otimes \cdots \otimes \mathbf{v}^{(N)}_{\alpha_{N-1}},\\ \mathbf{v}^{(1)} = \begin{bmatrix}\vec{\imath} & \vec{e}-\vec{\imath} & 0 & \cdots & 0\end{bmatrix}, \quad \mathbf{v}^{(n)} = \begin{bmatrix}\vec{\imath} & \vec{e}-\vec{\imath} & 0 & \cdots & 0\\ 0 & \vec{\imath} & \vec{e}-\vec{\imath} & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & 0\\ \vdots & & \ddots & \vec{\imath} & \vec{e}-\vec{\imath}\\ 0 & \cdots & \cdots & 0 & \vec{\imath}\end{bmatrix}, \quad \mathbf{v}^{(N)} = \begin{bmatrix}0\\ \vdots\\ 0\\ \vec{e}-\vec{\imath}\\ \vec{\imath}\end{bmatrix},\end{gathered} \tag{27}$$

with all TT ranks are equal to $H+1 = N-I+1$. Summing the above equation for $H = 0, \dots, N-I_\star$, we obtain the TT representation for the vector needed for computing the exceedance probability

$$\begin{gathered}\left[\mathbf{1}_{I(\mathrm{x})\geqslant I_\star}\right]_{\mathrm{x}\in\Omega} = \sum_{\alpha_1,\dots,\alpha_{N-1}=1}^{N-I_\star+1} \mathbf{w}^{(1)}_{\alpha_1} \otimes \cdots \otimes \mathbf{w}^{(n)}_{\alpha_{n-1},\alpha_n} \otimes \cdots \otimes \mathbf{w}^{(N)}_{\alpha_{N-1}},\\ \mathbf{w}^{(1)} = \begin{bmatrix}\vec{\imath} & \vec{e}-\vec{\imath} & 0 & \cdots & 0\end{bmatrix}, \quad \mathbf{w}^{(n)} = \begin{bmatrix}\vec{\imath} & \vec{e}-\vec{\imath} & 0 & \cdots & 0\\ 0 & \vec{\imath} & \vec{e}-\vec{\imath} & \ddots & \vdots\\ \vdots & \ddots & \ddots & \ddots & 0\\ \vdots & & \ddots & \vec{\imath} & \vec{e}-\vec{\imath}\\ 0 & \cdots & \cdots & 0 & \vec{\imath}\end{bmatrix}, \quad \mathbf{w}^{(N)} = \begin{bmatrix}\vec{e}\\ \vdots\\ \vec{e}\\ \vec{e}\\ \vec{\imath}\end{bmatrix}.\end{gathered} \tag{28}$$

Implementing this formula allows us to compute (25) in $\mathcal{O}(Nr^2(N-I_\star)^2)$ operations, where r is the largest TT rank of $\mathbf{p}$.

Finally, for a general observable that can be realised by an expectation

$$q(t) = \mathsf{E}[Q(\mathrm{x})] = \sum_{\mathrm{x}\in\Omega} Q(\mathrm{x})\mathbf{p}(\mathrm{x},t) = \langle \mathbf{Q}, \mathbf{p}(t)\rangle, \tag{29}$$

we can compute a TT approximation of the vector $[Q(\mathrm{x})]_{\mathrm{x}\in\Omega}$ by using TT cross interpolation methods [68–71]. These methods sample the function $Q(\mathrm{x})$ typically at $\mathcal{O}(Nr^2)$ adaptively chosen states $\mathrm{x}\in\Omega$, followed by $\mathcal{O}(Nr^3)$ other floating point operations in linear algebra.

Note that observables are evaluated as a post-processing step after solving the master equation (4), in contrast to SSA, for which the desired observations have to be stated in advance.

4. Results

The proposed method and necessary tensor product algorithms are implemented by authors in Matlab. The SSA algorithm is implemented by authors in Matlab. Where possible, the accuracy of results obtained by numerical methods is verified against analytic solutions, which were obtained as follows. First, for a given network of contacts, the Markov chain transition graphs (such as the one in Fig. 1) were constructed and the ODEs (3) were written using Julia language. After that, the analytic solutions for the ODEs were obtained using SageMath software package, which runs Maxima computer algebra system as backend. The numerical experiments were computed in MATLAB 2020b on an Intel Xeon E5-2640 v4 CPU with 2.4 GHz.

The codes are publicly available from

- github.com/savostyanov/ttsir.

4.1. Linear chain network

As a first experiment, we consider a linear network of $N = 9$ people. Out of $|\Omega| = 3^N = 19683$ network states only $1022 = 2(2^N-1)$ are accessible from the initial state. This relatively modest scale of the problem makes it possible to write the ODEs (3) and to solve them analytically using SageMath software, which took us about 7 days of CPU time. From the analytic expressions for the p.d.f. $\mathbf{p}_\star(t)$ we evaluated analytic expressions for observables (5) and (25) and used them as reference values. The observables $q(t)$ obtained by numerical algorithms were compared with the reference values $q_\star(t)$ and the relative accuracy (relative error) was computed as

$$\text{relative accuracy} = \frac{\|q-q_\star\|_{L_2}}{\|q_\star\|_{L_2}}, \tag{30}$$

where $\|\cdot\|_{L_2}$ denotes a function norm, $\|u(t)\|^2_{L_2} = \int_0^\infty |u(t)|^2 \mathrm{d}t$.

In Fig. 2, we show the relative accuracies (30) and CPU times of the TT and SSA methods for both the total mean number of infected individuals, and the occupancy probabilities for the linear chain. It should be noted that the TT algorithms are parameterised

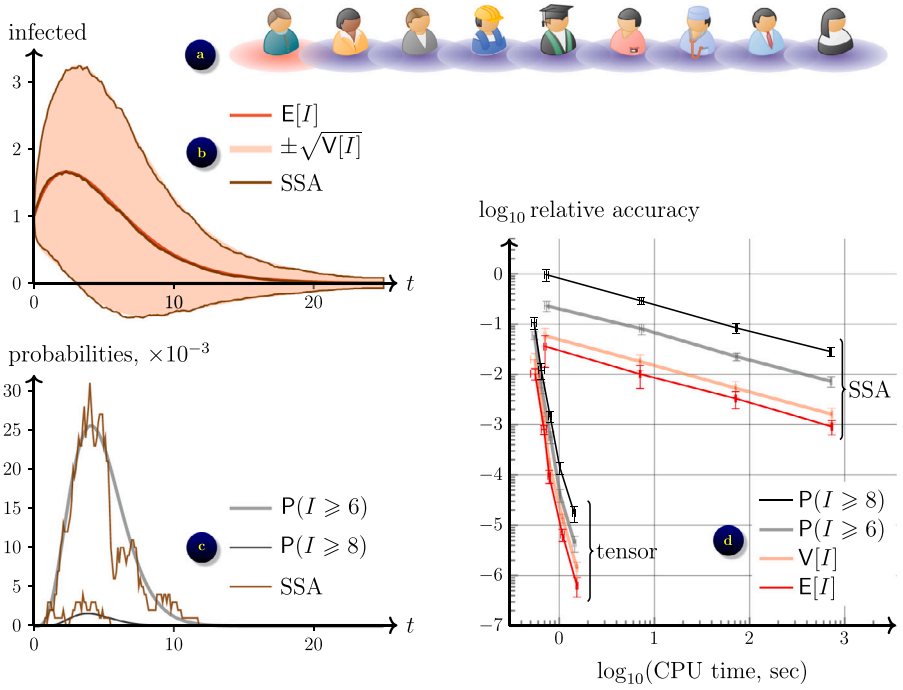

Fig. 2. SIR epidemic on a linear chain of $N = 9$ people, shown for $\beta = 1$ and $\gamma = 0.3$: (a) the network in its initial state; (b) mean and variance of the number of infected computed analytically and simulated using SSA with $N_{\text{SSA}} = 10^3$ sampled trajectories; (c) exceedance probabilities computed analytically and simulated using SSA with $N_{\text{SSA}} = 10^3$ sampled trajectories; (d) relative accuracy (30) of mean, variance, and exceedance probabilities, computed by SSA and tensor product approach.

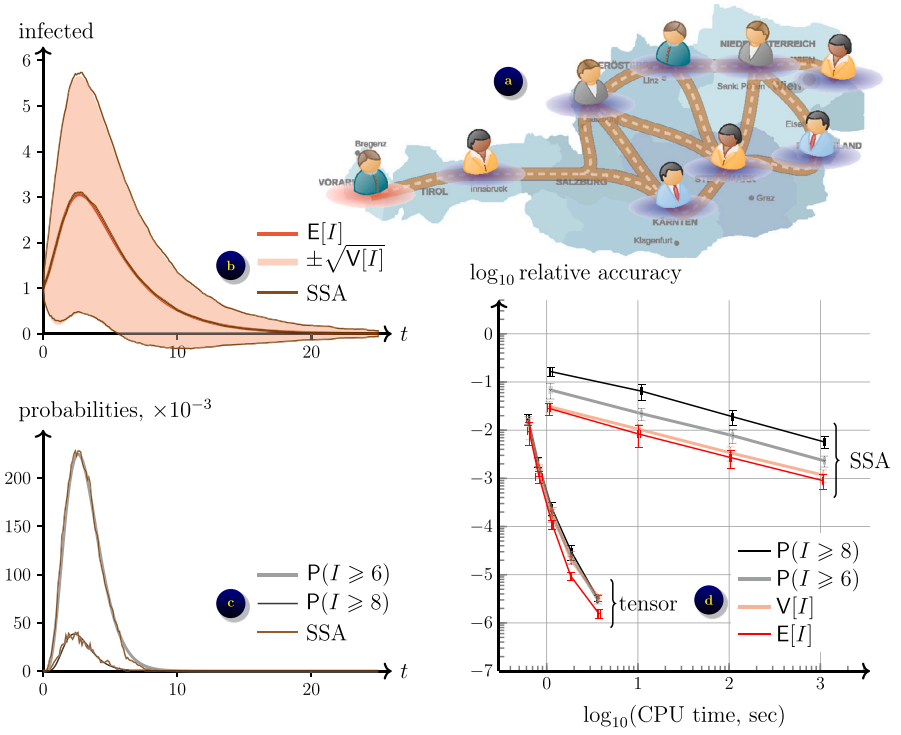

Fig. 3. SIR epidemic on a road network in Austria, shown for $\beta = 1$ and $\gamma = 0.3$: (a) the network in its initial state; (b) mean and variance of the number of infected computed analytically and simulated using SSA with $N_{\text{SSA}} = 10^3$ sampled trajectories; (c) exceedance probabilities computed analytically and simulated using SSA with $N_{\text{SSA}} = 10^3$ sampled trajectories; (d) relative accuracy (30) of mean, variance, and exceedance probabilities, computed by SSA and tensor product approach.

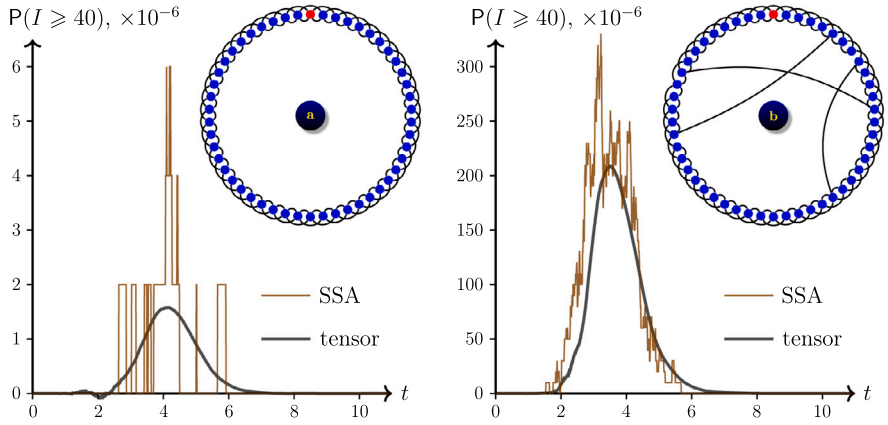

Fig. 4. SIR epidemic on small world networks with $N = 50$ people, with probability of rare event $\mathsf{P}(I \geqslant 40)$ computed using SSA and tensor product approach, shown for $\beta = 1$ and $\gamma = 0.3$: (a) the small world network with short–range connections, where each person is linked with two neighbours left of them and two neighbours right of them on the circle; (b) the same network but with 3 randomly selected nodes rewired to a random node.

by the error tolerance, which is used as a threshold for the relative accuracy in Frobenius norm for both the truncation of the TT decompositions and for stopping of the tAMEn algorithm. In contrast, the SSA method is parameterised by the number of samples N_{SSA}. Since these parameters don't match directly, we compare the CPU times of both methods. For all quantities of interest, SSA converges with a $\mathcal{O}(N_{\text{SSA}}^{-1/2})$ rate as expected. Although a seemingly modest number of samples may be sufficient to estimate mean population numbers, probabilities of rare events are much more difficult to estimate. In particular, SSA gives a rather misleading information about the high occupancy probability even with tens thousands of samples, requiring hundreds of seconds of computing for this (relatively simple) example. In contrast, the tensor product approach can compute the entire p.d.f. (and hence any derived statistics) with 4 accurate decimal digits in just a couple of seconds.

4.2. Road network in Austria

Now we test the methods on a network shown in Fig. 3(a), which illustrates the Austrian state adjacency map. This network has $N = 9$ nodes but more edges than the linear chain, resulting in better mixing and higher number of accessible network states. Assume that the initial state is deterministic with the first node in the infected state and all other nodes in the susceptible state, the Markov chain has 4982 accessible states for this network compared to 1022 for the linear chain network of the same size. Nevertheless, we were able to compute the analytic solution for this problem using SageMath and used it as a reference to benchmark the accuracy of numerical algorithms.

The accuracy of tAMEn and SSA algorithms is shown in Fig. 3(d). Similarly to the results in Fig. 2(d), we see that SSA converges according to the central limit theorem law, whereas the TT method can achieve a faster rate. Due to a more connected network, the exceedance probabilities are about ten times larger than those in the chain network, which makes it easier for the SSA algorithm to recover them. However, as the geometry of this network is somewhat elongated in one direction, and matches (although not perfectly) the linear geometry of the tensor train format, the tAMEn algorithm also performs well. Overall, we can see that if two or more accurate digits are desired in observables, the tensor product approach is more attractive for this example.

4.3. Small world networks

Lastly, we consider two small world networks, produced by the Watts–Strogatz algorithm, as shown in Fig. 4. The first contains $N = 50$ people, each connected to 2 next and 2 previous neighbours on a circular chain. The second additionally has 3 of its edges rewired to random vertices. The epidemics start with the first node in the infected state, all others in the susceptible state. We are interested in the probability that $I \geqslant 40$, i.e. 4/5 of the population is infected at once. For the first network, $\mathsf{P}(I \geqslant 40)$ reaches the level of $1.5 \cdot 10^{-6}$ at its peak, which makes it a rare event. To accurately resolve this small value, we apply the tAMEn algorithm with the approximation threshold of 10^{-9}. The TT ranks of the p.d.f. $\mathbf{p}(t)$ reach the value of 401, and the computation takes about 5 hours of CPU time. We then applied the SSA method with $5 \cdot 10^5$ trajectories, which requires approximately the same CPU time, and compared the results in Fig. 4(a). We see that only a tiny fraction of SSA trajectories has hit the event of interest, resulting in a rather inaccurate estimate of the probability. The TT method was able to produce more accurate and smooth estimate of the probability. We note that the probability recovered by the tensor product approach has a numerical artefact at $t \approx 2$ where it became negative with the magnitude of 10^{-7}, which is caused by the approximation error. This suggests that our choice of the error threshold is reasonable, as more aggressive compression may destroy the structure of the probability of interest, while a more accurate approximation would result in larger TT ranks and CPU time. To reach a similar accuracy with SSA we would have to increase the number of trajectories to 10^7 that would require more than 120 hours of computing, while the tAMEn algorithm recovers the whole p.d.f. in just 5 hours.

The rewired network has more long–range connections that facilitate the propagation of the infection, and makes it much more probable for a large number of people to be infected at once. For instance, for this network $\mathsf{P}(I \geqslant 40)$ peaks at about $2 \cdot 10^{-4}$, making

it easier for SSA to discover this event. With only 10^5 trajectories, SSA yields a reasonable estimate in 1.3 hours. For the TT method we can take the approximation threshold of 10^{-7}, observe the maximal TT rank of the p.d.f. to be 337, and recover $\mathbf{p}(t)$ using 2.5 hours of CPU time. The results are compared in Fig. 4(b), where we again can see that the estimate obtained by the TT algorithm is more accurate.

We see that the TT method is beneficial for rare event simulations requiring high overall accuracy and/or weakly correlated systems, since the TT decomposition converges rapidly when the correlations are local.

5. Discussion and conclusion

We have demonstrated numerically that the TT approximation of the probability distribution function converges rapidly for stochastic population models on networks with local connections, with only a polynomial scaling in the number of individuals, and a poly-logarithmic scaling in the error. This allows one to compute any statistics of such models more accurately than using the stochastic simulation. In particular, we have managed to compute probabilities of high infectivity events of the order of 10^{-6}. Thus, tensor methods can be recommended for models of moderate size, local connectivity, and/or if rare event statistics are of interest.

Two limitations of the proposed approach are very high dimensions and long–range connections in the network. The TT approximations are based on the SVD algorithm, which is known to produce an optimal approximation in the Frobenius norm, so $\|\mathbf{p}-\tilde{\mathbf{p}}\|_F$ is controlled. However, in practical computations we are interested in controlling the accuracy of the observables, not the p.d.f. Considering, for example, the mean $\mathsf{E}[I]=\langle \mathbf{I},\mathbf{p}\rangle$, the error of this observable can be bounded as $|\langle \mathbf{I},\mathbf{p}-\tilde{\mathbf{p}}\rangle| \leqslant \|\mathbf{I}\|_F \|\mathbf{p}-\tilde{\mathbf{p}}\|_F$. However, $\|\mathbf{I}\|_F$ can be extremely large for large N, purely because of the exponentially large length of the vector $\mathbf{I}$. This means that to ensure a reasonable accuracy of the observable, we need to approximate p.d.f. to a much higher accuracy. Alternatively, we can bound the error of the observable as $|\langle \mathbf{I},\mathbf{p}-\tilde{\mathbf{p}}\rangle| \leqslant \|\mathbf{I}\|_\infty \cdot \|\mathbf{p}-\tilde{\mathbf{p}}\|_1$, where $\|\mathbf{u}\|_1=\sum_{\mathrm{x}\in\Omega}|u(\mathrm{x})|$ is the 1–norm of vector, and $\|\mathbf{u}\|_\infty=\max_{\mathrm{x}\in\Omega}|u(\mathrm{x})|$ is the maximum–norm. This approach seems more promising, since $\max_{\mathrm{x}\in\Omega}|I(\mathrm{x})|\leqslant N$, i.e. grows at most linearly with the network size. However, in this case we need to control $\|\mathbf{p}-\tilde{\mathbf{p}}\|_1$ while solving ODE (4) in the TT format (14). Potentially, this can be done by formulating a nonlinear ODE on $q(\mathrm{x},t)=\sqrt{p(\mathrm{x},t)}$, in which case we can control $\|\mathbf{q}-\tilde{\mathbf{q}}\|_F$ during the solution and ensure

$$\|\mathbf{p}-\tilde{\mathbf{p}}\|_1=\sum_{\mathrm{x}\in\Omega}|q(\mathrm{x})^2-\tilde{q}(\mathrm{x})^2|=\langle|\mathbf{q}-\tilde{\mathbf{q}}|,|\mathbf{q}+\tilde{\mathbf{q}}|\rangle\leqslant\|\mathbf{q}-\tilde{\mathbf{q}}\|_F\|\mathbf{q}+\tilde{\mathbf{q}}\|_F\leqslant\|\mathbf{q}-\tilde{\mathbf{q}}\|_F\underbrace{(\|\mathbf{q}\|_F+\|\tilde{\mathbf{q}}\|_F)}_{2}.$$

However, the ODE for $\mathbf{q}$ features the reciprocals $1/q(\mathrm{x},t)$, complicating calculations when $q(\mathrm{x},t)=0$ for some states x, for instance when the initial state is deterministic.

Long–range connections in the network inflate the ranks of the TT decomposition and slow the method down. This drawback may be curable by using tree tensor networks with the topology adapted to the given network [72–74]. This is a subject of future work.

Finally, we should note that SSA algorithm is embarrassingly parallel, because all trajectories can be sampled independently. To be competitive with SSA, tensor product algorithms also have to perform well on distributed–memory high–performance computing platforms. Recent progress in this area includes [75–77,71].

Data availability

No data was used for the research described in the article.

Acknowledgements

Sergey Dolgov was supported by the Engineering and Physical Sciences Research Council New Investigator Award EP/T031255/1. Dmitry Savostyanov was supported by the Leverhulme Trust Research Fellowship RF-2021-258. Funders took no part in study design; in the collection, analysis and interpretation of data; in the writing of the report; and in the decision to submit the article for publication. Both authors contributed equally to this study and article; the order of authors is alphabetical.

References

[1] W.O. Kermack, A.G. McKendrick, A contribution to the mathematical theory of epidemics, Proc. R. Soc. Lond. A 115 (772) (1927) 700–721, https://doi.org/10.1098/rspa.1927.0118.
[2] W.-Y. Chen, S. Bokka, Stochastic modeling of nonlinear epidemiology, J. Theor. Biol. 234 (4) (2005) 455–470, https://doi.org/10.1016/j.jtbi.2004.11.033.
[3] M. Youssef, C. Scoglio, An individual–based approach to SIR epidemics in contact networks, J. Theor. Biol. 283 (1) (2011) 136–144, https://doi.org/10.1016/j.jtbi.2011.05.029.
[4] D.T. Gillespie, Approximate accelerated stochastic simulation of chemically reacting systems, J. Chem. Phys. 115 (4) (2001) 1716–1733, https://doi.org/10.1063/1.1378322.
[5] M.J. Keeling, The effects of local spatial structure on epidemiological invasions, Proc. Biol. Sci. 266 (1421) (1999) 859–867, https://doi.org/10.1098/rspb.1999.0716.
[6] D.A. Rand, Correlation equations and pair approximations for spatial ecologies, in: Advanced Ecological Theory: Principles and Applications, Blackwell Science, Oxford, 1999, pp. 100–142, Ch. 4.
[7] J.P. Gleeson, High-accuracy approximation of binary-state dynamics on networks, Phys. Rev. Lett. 107 (2011) 068701, https://doi.org/10.1103/PhysRevLett.107.068701.
[8] J. Lindquist, J. Ma, P. van den Driessche, F.H. Willeboordse, Effective degree network disease models, J. Math. Biol. 62 (2011) 143–164, https://doi.org/10.1007/s00285-010-0331-2.

[9] M. Taylor, T.J. Taylor, I.Z. Kiss, Epidemic threshold and control in a dynamic network, Phys. Rev. E 85 (2012) 016103, https://doi.org/10.1103/PhysRevE.85.016103.
[10] J.C. Miller, A.C. Slim, E.M. Volz, Edge-based compartmental modelling for infectious disease spread, J. R. Soc. Interface 9 (2012) 890–906, https://doi.org/10.1098/rsif.2011.0403.
[11] M. Griebel, H. Harbrecht, Analysis of tensor approximation schemes for continuous functions, Found. Comput. Math. 23 (2023) 219–240, https://doi.org/10.1007/s10208-021-09544-6.
[12] I.V. Oseledets, Tensor-train decomposition, SIAM J. Sci. Comput. 33 (5) (2011) 2295–2317, https://doi.org/10.1137/090752286.
[13] W. Hackbusch, S. Kühn, A new scheme for the tensor representation, J. Fourier Anal. Appl. 15 (5) (2009) 706–722, https://doi.org/10.1007/s00041-009-9094-9.
[14] S.V. Dolgov, TT-GMRES: solution to a linear system in the structured tensor format, Russ. J. Numer. Anal. Math. Model. 28 (2) (2013) 149–172, https://doi.org/10.1515/rnam-2013-0009.
[15] S.V. Dolgov, D.V. Savostyanov, Alternating minimal energy methods for linear systems in higher dimensions, SIAM J. Sci. Comput. 36 (5) (2014) A2248–A2271, https://doi.org/10.1137/140953289.
[16] S.V. Dolgov, B.N. Khoromskij, I.V. Oseledets, D.V. Savostyanov, Computation of extreme eigenvalues in higher dimensions using block tensor train format, Comput. Phys. Commun. 185 (4) (2014) 1207–1216, https://doi.org/10.1016/j.cpc.2013.12.017.
[17] S.V. Dolgov, D.V. Savostyanov, Corrected one-site density matrix renormalization group and alternating minimal energy algorithm, in: Numerical Mathematics and Advanced Applications — ENUMATH 2013, vol. 103, 2015, pp. 335–343.
[18] M. Rakhuba, A. Novikov, I. Oseledets, Low-rank Riemannian eigensolver for high-dimensional Hamiltonians, J. Comput. Phys. 396 (1) (2019) 718–737, https://doi.org/10.1016/j.jcp.2019.07.003.
[19] S.V. Dolgov, A tensor decomposition algorithm for large ODEs with conservation laws, Comput. Methods Appl. Math. 19 (2019) 23–38, https://doi.org/10.1515/cmam-2018-0023.
[20] S.R. White, Density-matrix algorithms for quantum renormalization groups, Phys. Rev. B 48 (14) (1993) 10345–10356, https://doi.org/10.1103/PhysRevB.48.10345.
[21] A. Klümper, A. Schadschneider, J. Zittartz, Matrix product ground states for one-dimensional spin-1 quantum antiferromagnets, Europhys. Lett. 24 (4) (1993) 293–297, https://doi.org/10.1209/0295-5075/24/4/010.
[22] U. Schollwöck, The density-matrix renormalization group in the age of matrix product states, Ann. Phys. 326 (1) (2011) 96–192, https://doi.org/10.1016/j.aop.2010.09.012.
[23] T.G. Kolda, B.W. Bader, Tensor decompositions and applications, SIAM Rev. 51 (3) (2009) 455–500, https://doi.org/10.1137/07070111X.
[24] W. Hackbusch, Tensor Spaces and Numerical Tensor Calculus, Springer–Verlag, Berlin, 2012.
[25] B.N. Khoromskij, Tensor numerical methods for multidimensional PDEs: theoretical analysis and initial applications, ESAIM Proc. 48 (2015) 1–28, https://doi.org/10.1051/proc/201448001.
[26] J. Ballani, L. Grasedyck, M. Kluge, A review on adaptive low-rank approximation techniques in the hierarchical tensor format, in: Extraction of Quantifiable Information from Complex Systems, in: Lecture Notes in Computational Science and Engineering, vol. 102, Springer, 2014, pp. 195–210.
[27] N.G. van Kampen, Stochastic Processes in Physics and Chemistry, North Holland, Amsterdam, 1981.
[28] M. Hemberg, M. Barahona, Perfect sampling of the master equation for gene regulatory networks, Biophys. J. 93 (2) (2007) 401–410, https://doi.org/10.1529/biophysj.106.099390.
[29] D.F. Anderson, D.J. Higham, Multilevel Monte Carlo for continuous time Markov chains, with applications in biochemical kinetics, Multiscale Model. Simul. 10 (1) (2012) 146–179, https://doi.org/10.1137/110840546.
[30] C. Lester, R.E. Baker, M.B. Giles, C.A. Yates, Extending the multi-level method for the simulation of stochastic biological systems, Bull. Math. Biol. 78 (8) (2016) 1640–1677, https://doi.org/10.1007/s11538-016-0178-9.
[31] B. Munsky, M. Khammash, The finite state projection algorithm for the solution of the chemical master equation, J. Chem. Phys. 124 (2006) 044104, https://doi.org/10.1063/1.2145882.
[32] T. Jahnke, An adaptive wavelet method for the chemical master equation, SIAM J. Sci. Comput. 31 (6) (2010) 4373, https://doi.org/10.1137/080742324.
[33] Y. Cao, A. Terebus, J. Liang, State space truncation with quantified errors for accurate solutions to discrete chemical master equation, Bull. Math. Biol. 78 (4) (2016) 617–661, https://doi.org/10.1007/s11538-016-0149-1.
[34] M. Hegland, C. Burden, L. Santoso, S. MacNamara, H. Booth, A solver for the stochastic master equation applied to gene regulatory networks, J. Comput. Appl. Math. 205 (2) (2007) 708–724, https://doi.org/10.1016/j.cam.2006.02.053.
[35] I. Kryven, S. Röblitz, C. Schütte, Solution of the chemical master equation by radial basis functions approximation with interface tracking, BMC Syst. Biol. 9 (1) (2015) 67, https://doi.org/10.1186/s12918-015-0210-y.
[36] A. Gupta, C. Schwab, M. Khammash, DeepCME: a deep learning framework for computing solution statistics of the chemical master equation, PLoS Comput. Biol. 17 (12) (2021) 1–23, https://doi.org/10.1371/journal.pcbi.1009623.
[37] A. Sukys, K. Öcal, R. Grima, Approximating solutions of the chemical master equation using neural networks, bioRxiv, https://doi.org/10.1101/2022.04.26.489548, 2022.
[38] T. Jahnke, W. Huisinga, A dynamical low-rank approach to the chemical master equation, Bull. Math. Biol. 70 (2008) 2283–2302, https://doi.org/10.1007/s11538-008-9346-x.
[39] A. Ammar, E. Cueto, F. Chinesta, Reduction of the chemical master equation for gene regulatory networks using proper generalized decompositions, Int. J. Numer. Methods Biomed. Eng. 28 (9) (2012) 960–973, https://doi.org/10.1002/cnm.2476.
[40] M. Hegland, J. Garcke, On the numerical solution of the chemical master equation with sums of rank one tensors, ANZIAM J. 52 (2011) C628–C643, https://doi.org/10.21914/anziamj.v52i0.3895.
[41] V. Kazeev, M. Khammash, M. Nip, C. Schwab, Direct solution of the chemical master equation using quantized tensor trains, PLoS Comput. Biol. 10 (3) (2014) e100359, https://doi.org/10.1371/journal.pcbi.1003359.
[42] S. Dolgov, B. Khoromskij, Simultaneous state-time approximation of the chemical master equation using tensor product formats, Numer. Linear Algebra Appl. 22 (2) (2015) 197–219, https://doi.org/10.1002/nla.1942.
[43] H.D. Vo, R.B. Sidje, An adaptive solution to the chemical master equation using tensors, J. Chem. Phys. 147 (4) (2017) 044102, https://doi.org/10.1063/1.4994917.
[44] T. Dinh, R.B. Sidje, An adaptive solution to the chemical master equation using quantized tensor trains with sliding windows, Phys. Biol. 17 (6) (2020) 065014, https://doi.org/10.1088/1478-3975/aba1d2.
[45] I.G. Ion, C. Wildner, D. Loukrezis, H. Koeppl, H. De Gersem, Tensor-train approximation of the chemical master equation and its application for parameter inference, J. Chem. Phys. 155 (3) (2021) 034102, https://doi.org/10.1063/5.0045521.
[46] P. Gelß, S. Matera, C. Schütte, Solving the master equation without kinetic Monte Carlo: tensor train approximations for a CO oxidation model, J. Comput. Phys. 314 (2016) 489–502, https://doi.org/10.1016/j.jcp.2016.03.025.
[47] Z.I. Botev, D.P. Kroese, An efficient algorithm for rare-event probability estimation, combinatorial optimization, and counting, Methodol. Comput. Appl. Probab. 10 (4) (2008) 471–505, https://doi.org/10.1007/s11009-008-9073-7.
[48] B. Peherstorfer, B. Kramer, K. Willcox, Multifidelity preconditioning of the cross-entropy method for rare event simulation and failure probability estimation, SIAM/ASA J. Uncertain. Quantificat. 6 (2) (2018) 737–761, https://doi.org/10.1137/17M1122992.

[49] F. Wagner, J. Latz, I. Papaioannou, E. Ullmann, Multilevel sequential importance sampling for rare event estimation, SIAM J. Sci. Comput. 42 (4) (2020) A2062–A2087, https://doi.org/10.1137/19M1289601.
[50] J.G. Kemeny, J.L. Snell, Finite Markov Chains, Springer, 1976.
[51] L.C.G. Rogers, J.W. Pitman, Markov functions, Ann. Probab. 9 (4) (1981) 573–582, https://doi.org/10.1214/aop/1176994363.
[52] D.T. Gillespie, Exact stochastic simulation of coupled chemical reactions, J. Phys. Chem. 81 (25) (1977) 2340–2361, https://doi.org/10.1021/j100540a008.
[53] D.F. Anderson, A. Ganguly, T.G. Kurtz, Error analysis of Tau-Leap simulation methods, Ann. Appl. Probab. 21 (6) (2011) 2226–2262, https://doi.org/10.1214/10-AAP756.
[54] F.L. Hitchcock, The expression of a tensor or a polyadic as a sum of products, J. Math. Phys. 6 (1) (1927) 164–189.
[55] V. de Silva, L.-H. Lim, Tensor rank and the ill-posedness of the best low-rank approximation problem, SIAM J. Matrix Anal. Appl. 30 (3) (2008) 1084–1127, https://doi.org/10.1137/06066518x.
[56] G. Golub, W. Kahan, Calculating the singular values and pseudo-inverse of a matrix, SIAM J. Numer. Anal. 2 (2) (1965) 205–224.
[57] S. Holtz, T. Rohwedder, R. Schneider, The alternating linear scheme for tensor optimization in the tensor train format, SIAM J. Sci. Comput. 34 (2) (2012) A683–A713, https://doi.org/10.1137/100818893.
[58] M. Fannes, B. Nachtergaele, R. Werner, Finitely correlated states on quantum spin chains, Commun. Math. Phys. 144 (3) (1992) 443–490, https://doi.org/10.1007/BF02099178.
[59] P.B. Rohrbach, S. Dolgov, L. Grasedyck, R. Scheichl, Rank bounds for approximating Gaussian densities in the Tensor-Train format, SIAM/ASA J. Uncertain. Quantificat. 10 (3) (2022) 1191–1224, https://doi.org/10.1137/20M1314653.
[60] V. Kazeev, C. Schwab, Tensor approximation of stationary distributions of chemical reaction networks, SIAM J. Matrix Anal. Appl. 36 (3) (2015) 1221–1247, https://doi.org/10.1137/130927218.
[61] G.D. Byrne, A.C. Hindmarsh, A polyalgorithm for the numerical solution of ordinary differential equations, ACM Trans. Math. Softw. 1 (1) (1975) 71–96, https://doi.org/10.1145/355626.355636.
[62] L.N. Trefethen, Spectral Methods in MATLAB, SIAM, Philadelphia, 2000.
[63] V.A. Kazeev, B.N. Khoromskij, Low-rank explicit QTT representation of the Laplace operator and its inverse, SIAM J. Matrix Anal. Appl. 33 (3) (2012) 742–758, https://doi.org/10.1137/100820479.
[64] V. Kazeev, O. Reichmann, C. Schwab, Low-rank tensor structure of linear diffusion operators in the TT and QTT formats, Linear Algebra Appl. 438 (11) (2013) 4204–4221, https://doi.org/10.1016/j.laa.2013.01.009.
[65] S.V. Dolgov, B.N. Khoromskij, D.V. Savostyanov, Superfast Fourier transform using QTT approximation, J. Fourier Anal. Appl. 18 (5) (2012) 915–953, https://doi.org/10.1007/s00041-012-9227-4.
[66] D.V. Savostyanov, QTT-rank-one vectors with QTT-rank-one and full-rank Fourier images, Linear Algebra Appl. 436 (9) (2012) 3215–3224, https://doi.org/10.1016/j.laa.2011.11.008.
[67] I.V. Oseledets, D.V. Savostyanov, E.E. Tyrtyshnikov, Linear algebra for tensor problems, Computing 85 (3) (2009) 169–188, https://doi.org/10.1007/s00607-009-0047-6.
[68] I.V. Oseledets, E.E. Tyrtyshnikov, TT-cross approximation for multidimensional arrays, Linear Algebra Appl. 432 (1) (2010) 70–88, https://doi.org/10.1016/j.laa.2009.07.024.
[69] D.V. Savostyanov, I.V. Oseledets, Fast adaptive interpolation of multi-dimensional arrays in tensor train format, in: Proceedings of 7th International Workshop on Multidimensional Systems (nDS), IEEE, 2011.
[70] D.V. Savostyanov, Quasioptimality of maximum–volume cross interpolation of tensors, Linear Algebra Appl. 458 (2014) 217–244, https://doi.org/10.1016/j.laa.2014.06.006.
[71] S. Dolgov, D. Savostyanov, Parallel cross interpolation for high–precision calculation of high–dimensional integrals, Comput. Phys. Commun. 246 (2020) 106869, https://doi.org/10.1016/j.cpc.2019.106869.
[72] G. Barcza, O. Legeza, K.H. Marti, M. Reiher, Quantum-information analysis of electronic states of different molecular structures, Phys. Rev. A 83 (2011) 012508, https://doi.org/10.1103/PhysRevA.83.012508.
[73] J. Ballani, L. Grasedyck, Tree adaptive approximation in the hierarchical tensor format, SIAM J. Sci. Comput. 36 (4) (2014) A1415–A1431, https://doi.org/10.1137/130926328.
[74] M. Bebendorf, C. Kuske, Separation of variables for function generated high-order tensors, J. Sci. Comput. 61 (1) (2014) 145–165, https://doi.org/10.1007/s10915-014-9822-4.
[75] E. Solomonik, D. Matthews, J.R. Hammond, J.F. Stanton, J. Demmel, A massively parallel tensor contraction framework for coupled-cluster computations, J. Parallel Distrib. Comput. 74 (12) (2014) 3176–3190, https://doi.org/10.1016/j.jpdc.2014.06.002.
[76] L. Grasedyck, R. Kriemann, C. Löbbert, A. Nägel, G. Wittum, K. Xylouris, Parallel tensor sampling in the hierarchical Tucker format, Comput. Vis. Sci. 17 (2) (2015) 67–78, https://doi.org/10.1007/s00791-015-0247-x.
[77] P. Secular, N. Gourianov, M. Lubasch, S. Dolgov, S.R. Clark, D. Jaksch, Parallel time-dependent variational principle algorithm for matrix product states, Phys. Rev. B 101 (2020) 235123, https://doi.org/10.1103/PhysRevB.101.235123.