MULTISCALE MODEL. SIMUL.
Vol. 14, No. 2, pp. 923–963

ACCURATE CHEMICAL MASTER EQUATION SOLUTION USING MULTI-FINITE BUFFERS*

YOUFANG CAO†, ANNA TEREBUS‡, AND JIE LIANG§

Abstract. The discrete chemical master equation (dCME) provides a fundamental framework for studying stochasticity in mesoscopic networks. Because of the multiscale nature of many networks where reaction rates have a large disparity, directly solving dCMEs is intractable due to the exploding size of the state space. It is important to truncate the state space effectively with quantified errors, so accurate solutions can be computed. It is also important to know if all major probabilistic peaks have been computed. Here we introduce the accurate CME (ACME) algorithm for obtaining direct solutions to dCMEs. With multifinite buffers for reducing the state space by $O(n!)$, exact steady-state and time-evolving network probability landscapes can be computed. We further describe a theoretical framework of aggregating microstates into a smaller number of macrostates by decomposing a network into independent aggregated birth and death processes and give an a priori method for rapidly determining steady-state truncation errors. The maximal sizes of the finite buffers for a given error tolerance can also be precomputed without costly trial solutions of dCMEs. We show exactly computed probability landscapes of three multiscale networks, namely, a 6-node toggle switch, 11-node phage-lambda epigenetic circuit, and 16-node MAPK cascade network, the latter two with no known solutions. We also show how probabilities of rare events can be computed from first-passage times, another class of unsolved problems challenging for simulation-based techniques due to large separations in time scales. Overall, the ACME method enables accurate and efficient solutions of the dCME for a large class of networks.

Key words. chemical master equation, stochastic biological networks, state space truncation, steady state probability landscape, time-evolving probability landscapes, first passage time distribution

AMS subject classifications. 92BXX, 60-08

DOI. 10.1137/15M1034180

1. Introduction.

Biochemical reaction networks are intrinsically stochastic [3, 59, 71] and often multiscale when there exists a large disparity in reaction rates. When genes, transcription factors, signaling molecules, and regulatory proteins are in small quantities ($10 \sim 100$ nM), stochasticity plays important roles [5, 11, 19, 72]. Deterministic models based on chemical mass action kinetics cannot capture the stochastic nature of these networks [11, 52, 81]. Instead, the discrete chemical master equations (dCME) that describe the probabilistic jumps between discrete states due to the firing of reactions can fully describe these mesoscopic stochastic processes in a well mixed system [8, 23, 24, 27, 76].

However, studying the stochastic behavior of a multiscale network is challenging. The rate constants of different reactions often have large separations in time scale by a

*Received by the editors August 6, 2015; accepted for publication (in revised form) April 4, 2016; published electronically June 29, 2016. This work was supported by NIH grant GM079804, NSF grant MCB1415589, and the Chicago Biomedical Consortium with support from the Searle Funds at The Chicago Community Trust.

http://www.siam.org/journals/mms/14-2/M103418.html

†Department of Bioengineering, University of Illinois at Chicago, Chicago, IL 60607. Current address: Center for Nonlinear Studies (CNLS) and Theoretical Biology and Biophysics (T-6), Los Alamos National Laboratory, Los Alamos, NM 87545 (ycao@lanl.gov). This author gratefully acknowledges the support of the U.S. Department of Energy through the LANL/LDRD Program for this work.

‡Department of Bioengineering, University of Illinois at Chicago, Chicago, IL 60607 (anna.terebus@gmail.com).

§Corresponding author. Department of Bioengineering, University of Illinois at Chicago, Chicago, IL 60607 (jliang@uic.edu).

few orders of magnitude. Copy numbers of molecular species can also span a number of orders of magnitude, further exacerbating the problem of time separations between slow and fast reactions. Even with a correctly constructed model of a stochastic network, it is generally unknown if an accurate solution has been found. One does not know if a computed probabilistic landscape is overall erroneous and how such errors can be quantified. For example, it is difficult to know if all major probabilistic peaks have been identified or important peaks in the usually high dimensional space with significant probability mass are undetected. Furthermore, the best possible accuracy one can hope to achieve with given finite computing resources is generally unknown. In addition, one does not know what is required so accurate solutions with errors smaller than a predefined tolerance can be obtained.

While the time-evolving probability landscape over discrete states governed by the dCME provides detailed information of the underlying dynamic stochastic processes, the dCME cannot be solved analytically, except for a few very simple cases [16, 46, 53, 78]. Approximations to the dCME such as the chemical Fokker–Planck equation (FPE) and the chemical Langevin equation (CLE) are widely used to study stochastic reactions [4, 21, 25, 26, 32, 66, 77]. However, these approximations assume relatively large copy numbers of molecules, so the states can be regarded as continuous, and higher order terms of the Kramers–Moyal expansion of the dCME can be truncated [76]. These approximations do not provide a full account of the stochasticity of the system and are not valid when copy numbers of molecular species are small [25]. Although errors of these approximations have been assessed for simple reactions [29, 74] and a recent study showed that CLE failed to converge to the correct steady state probability landscape (see the appendix of [11]), the consequences of such approximations for realistic problems involving many molecular species and with complex reactions across multiple temporal scales are largely unknown.

The stochastic simulation algorithm (SSA) is widely used to study stochasticity in biological networks. It generates reaction trajectories dictated by the underlying dCME of the network [23]. The stochastic properties of the network can then be inferred through analysis of a large number of simulation trajectories. However, as the SSA follows high-probability reaction paths, it is therefore inefficient for sampling biologically critical rare events that often occur in stiff multiscale reaction networks, in which slow and fast reactions are well-separated in time scale [1, 10, 15, 38, 45, 79]. In addition, assessment of its convergence of simulation trajectories is also difficult. Recent development in biased sampling aims to address this problem [1, 10, 15, 45].

An attractive approach to study stochastic networks is to directly solve the dCME numerically. By computing the exact probability landscape of a stochastic network, its properties, including those involving rare events, can be studied accurately in detail. The finite state projection (FSP) method and the sliding window method are among several methods that have been developed to solve the dCME directly [9, 11, 37, 55, 82].

The FSP method is based on a truncated projection of the state space and uses numerical techniques to compute a direct solution to the dCME [55, 67]. Although the error due to state space truncation can be captured by the absorption state, to which all truncated states are projected [55], there is no systematic guidance as to which states and how many of them should be incorporated so the error can be minimized to remain within an acceptable tolerance [55, 56]. Furthermore, the introduction of the absorption state leads to an accumulation of errors as time proceeds, as this state would eventually absorb all probability mass. Designed to study transient behavior of stochastic networks, the FSP method therefore is challenged to compute the steady

state probability landscape and the first passage time distribution of rare events in a multiscale network.

The sliding window method for solving the dCME is also based on truncation of the state space. In this case, the state space is adaptively restricted to those that are likely relevant within a small time-window, with the assumption that most of the probability mass is contained within a set of preselected states [82]. However, to ensure that the truncation error is small, a large number of states need to be included, as the size of the state space takes the form of a d-dimensional hypercube, with the upper and lower bounds of copy numbers of each of the d molecular species predetermined by a Poisson model [82].

The main difficulty of all these methods is to have an adequate and accurate account of the discrete state space. As the copy number of each of the d molecular species takes an integer value, conventional hypercube-based methods incorporate all vertices in a d-dimensional hypercubic integer lattice, which has an overall size of $O(\prod_{i=1}^{d} m_i)$, where m_i is the maximally allowed copy number of molecular species i. State enumeration rapidly becomes intractable, both in storage and in computing time. For example, assuming a system has 16 molecular species, each with maximally 9 copies of molecules, a state space of size $(9+1)^{16} = 10^{16}$ would be required. This makes the direct solution of the dCME impossible for many realistic problems.

To address the issue of prohibitive size of the discrete state space, the finite buffer discrete CME (fb-dCME) method was developed for efficient state enumeration [9]. This algorithm is provably optimal in both memory usage and in the time required for enumeration when a single buffer queue is used. Instead of including every states in a hypercube, it examines only states that can be reached from a given initial state. It can be used to compute the exact probability landscape of a closed network or an open network when the net gain in newly synthesized molecules does not exceed a predefined finite capacity. However, as the available memory is limited, state truncation will eventually occur for open systems when synthesis reactions outpace degradation reactions and for closed system whose full enumeration requires memory that exceeds available capacity. In these cases, it is unclear whether the error associated with a truncated state space is within a tolerance threshold. Furthermore, similar to other methods aimed to solve the dCME directly, it is unclear how to minimize the error of a truncated state space, thus limiting the scope of applications of this method.

In this study, we introduce the accurate chemical master equation (ACME) method for solving the dCME. Our method is based on the decomposition of the multiscale stochastic reaction network into multiple independent components; each is governed by its own birth-death process, and each has a unique pattern of generation and degradation of molecules. In the ACME method, each independent component is equipped with its own finite state subspace controlled by a separate buffer queue. Similar to the original fb-dCME method, it is optimal in space and in time required for state enumeration but has the advantage of more effective usage of the overall finite state space and allows detailed analysis. This approach improves computing efficiency significantly and can generate state spaces of much larger effective sizes.

We also provide a method for rapid estimation of the errors in the computed steady state probability landscape upon truncation of the state space when using a buffer bank with a finite capacity. An estimation of the required buffer sizes can also be computed so the truncation error is within a predefined tolerance. These estimations are derived conservatively, so that the actual errors will not be larger than the estimated errors. A strategy for optimized buffer allocation is also given. Furthermore, the error bounds and required buffer sizes for each individual independent

component can all be rapidly computed a priori without costly computation of trial solutions to the dCME. These are based on results of theoretical analysis of the upper bound of the truncation error of the probability landscape at the steady state, which will be discussed in detail. The ACME algorithm along with the error estimation are implemented in the ACME package. Overall, the ACME method allows accurate solutions to the dCME with small and controlled errors for a much larger class of biological problems than previously feasible.

Our paper is organized as follows. We first review basic concepts of the dCME and issues associated with the finite discrete state space. We then describe the concept of reaction graph, its decomposition, and how independent birth-death components can be identified. We further introduce the ACME algorithm in which multifinite buffers are used for state enumeration. This is followed by a discussion of results of theoretical analysis of errors in the steady state probability landscape due to state truncation and how probability of boundary states can be used to construct upper bounds of the truncation errors. We then give detailed examples of three biological networks, namely, the toggle switch, the epigenetic circuit of lysis-lysogeny decision of phage lambda, and a model of MAPK cascade. We discuss the computed time-evolving and the steady state probability landscapes, along with the significant state space reduction achieved for these networks. Results on the challenging problem of estimating rare event probability through the computation of the first-passage times of these networks are also reported. We conclude with summaries and discussions.

2. Methods and theory.

2.1. Background.

2.1.1. Reaction network, state space, and probability landscape. In a well-mixed biochemical system with constant volume and temperature, we assume there are n molecular species, denoted as $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$, and m reactions, denoted as $\mathcal{R} = \{R_1, R_2, \ldots, R_m\}$. Each reaction R_k has an intrinsic reaction rate constant r_k. The microstate of the system at time t is given by the nonnegative integer column vector $\boldsymbol{x}(t) \in \mathbb{Z}^n_{\geq 0}$ of copy numbers of each molecular species: $\boldsymbol{x}(t) = (x_1(t), x_2(t), \ldots, x_n(t))^T$, where $x_i(t)$ is the copy number of molecular species X_i at time t. An arbitrary reaction R_k with intrinsic rate r_k takes the general form of

$$c_{1k}X_1 + c_{2k}X_2 + \cdots + c_{nk}X_n \xrightarrow{r_k} c'_{1k}X_1 + c'_{2k}X_2 + \cdots + c'_{nk}X_n,$$

which brings the system from a microstate $\boldsymbol{x}_j$ to $\boldsymbol{x}_i$. The difference between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is the stoichiometry vector $\boldsymbol{s}_k$ of reaction R_k: $\boldsymbol{s}_k = \boldsymbol{x}_i - \boldsymbol{x}_j = (s_{1k}, s_{2k}, \ldots, s_{nk})^T = (c'_{1k} - c_{1k},\, c'_{2k} - c_{2k}, \ldots, c'_{nk} - c_{nk})^T \in \mathbb{Z}^n$. The stoichiometry matrix $\boldsymbol{S}$ of the network is defined as $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_m) \in \mathbb{Z}^{n\times m}$, where each column correspond to one reaction. The rate $A_k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ of reaction R_k that brings the microstate from $\boldsymbol{x}_j$ to $\boldsymbol{x}_i$ is determined by r_k and the combination number of relevant reactants in the current microstate $\boldsymbol{x}_j$,

$$A_k(\boldsymbol{x}_i, \boldsymbol{x}_j) = A_k(\boldsymbol{x}_j) = r_k \prod_{l=1}^{n} \binom{x_l}{c_{lk}},$$

assuming the convention $\binom{0}{0} = 1$.

All possible microstates that a system can visit from a given initial condition form the state space $\Omega = \{\boldsymbol{x}(t) | \boldsymbol{x}(0),\, t \in (0, \infty)\}$. We denote the probability of each microstate at time t as $p(\boldsymbol{x}(t))$, and the probability distribution at time t over the full

state space as $\boldsymbol{p}(t) = \{(p(\boldsymbol{x}(t))|\boldsymbol{x}(t) \in \Omega)\}$. We also call $\boldsymbol{p}(t)$ the probability landscape of the network [11].

2.1.2. Discrete chemical master equation. The dCME can be written as a set of linear ordinary differential equations describing the change in probability of each discrete state over time:

$$\frac{dp(\boldsymbol{x},t)}{dt} = \sum_{\boldsymbol{x}',\, \boldsymbol{x}' \neq \boldsymbol{x}} [A(\boldsymbol{x},\boldsymbol{x}')p(\boldsymbol{x}',t) - A(\boldsymbol{x}',\boldsymbol{x})p(\boldsymbol{x},t)]. \tag{2.1}$$

Note that $p(\boldsymbol{x},t)$ is continuous in time but is over states that are discrete. In matrix form, the dCME can be written as

$$\frac{d\boldsymbol{p}(t)}{dt} = \boldsymbol{A}\boldsymbol{p}(t), \tag{2.2}$$

where $\boldsymbol{A} \in \mathbb{R}^{|\Omega|\times|\Omega|}$ is the transition rate matrix formed by the collection of all $A(\boldsymbol{x}_i,\boldsymbol{x}_j)$:

$$A(\boldsymbol{x}_i,\boldsymbol{x}_j) = \begin{cases} -\sum_{\boldsymbol{x}' \in \Omega,\ \boldsymbol{x}' \neq \boldsymbol{x}_j} A_k(\boldsymbol{x}',\boldsymbol{x}_j) & \text{if } \boldsymbol{x}_i = \boldsymbol{x}_j, \\ A_k(\boldsymbol{x}_i,\boldsymbol{x}_j) & \text{if } \boldsymbol{x}_i \neq \boldsymbol{x}_j \text{ and } \boldsymbol{x}_j \xrightarrow{R_k} \boldsymbol{x}_i, \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

2.2. Finite buffer for state space enumeration. Enumeration of the state space is a prerequisite for directly solving the dCME. The method of fb-dCME provides an efficient algorithm for state enumeration [9, 11]. By treating states as nodes and reactions as edges, the problem of state enumeration is transformed into that of a graph traversal problem [14]. The fb-dCME algorithm uses the depth-first search (DFS) to enumerate states that can be reached from an initial state [9]. For closed networks with no synthesis reactions, the finite state space can be fully enumerated, assuming the capacity of available computer memory is adequate.

For open networks with synthesis and degradation reactions found in a biological system, the size of the state space is also finite, as the total mass of molecules in a reaction system is conserved and the duration of reactions is bounded by the lifetime of a cell. Therefore, the net number of synthesized molecules that need to be modeled is finite. However, errors due to state space truncation will occur when the compute capacity is insufficient to fully account for the finite state space, as synthesis reaction can no longer proceed after memory exhaustion. Similarly, truncation error will occur when the size of the full state space of a closed network cannot be contained in the available memory.

The fb-dCME algorithm uses a buffer of a predefined capacity as a counter to keep track of the total number of molecules in the reaction system. Once the buffer capacity is determined, the maximum number of molecules in the system is given, which is the number of molecules that can be synthesized in the model. The buffer capacity is dictated by the available computer memory. When a synthesis reaction occurs, one buffer token is spent. When a degradation reaction occurs, one buffer token is deposited back. Multiple buffer tokens are taken or deposited when synthesis and degradation involve higher order reactions such as homo- or hetero-oligomers, with

the number of tokens equivalent to that of the monomers. The fb-dCME algorithm has been successfully applied in studying the stability and efficiency problem of phage lambda lysogeny-lysis epigenetic switch [11], as well as in direct computation of probabilities of critical rare events in the birth and death process, the Schlögl model, and the enzymatic futile cycle [10].

2.3. Multifinite buffers for state space enumeration. Reaction rates in a network can vary greatly: many steps of fast reactions can occur within a given time period, while only a few steps of slow reactions can occur in the same time period. The efficiency of state enumeration can be greatly improved if memory allocation is optimized based on different behavior of these reactions.

Independent birth-death (iBD) processes. It is useful to examine the reaction network in terms of birth and death processes, as birth (synthesis) and death (degradation) are the only reactions that can change the total mass of an open network by adding or removing molecules. These processes correspond to spending or depositing buffer tokens, respectively. Below we first introduce the concept of *reaction graph* and its partition into disjoint components. We then examine those components equipped with their own birth-death processes.

Reaction graph and independent reaction components. We first construct an undirected graph $\boldsymbol{G}_R$, with reactions form the set of vertices $\boldsymbol{V}$. A pair of reactions R_i and R_j are then connected by an edge e_{ij} if they share either reactant(s) or product(s). To correctly discover related reactions through the stoichiometry matrix, all molecular species in the network are represented using the combination of their most elementary form. For example, if a molecular species C is a complex formed by A bounded with B, we use the original form $A + B$ to represent C. Collectively, these reaction pairs sharing reactants or products form the edge set of the graph: $\boldsymbol{E} = \{e_{ij}\}$. The reaction graph $\boldsymbol{G}_R$ can be decomposed into u number of disjoint *independent reaction components* $\{\boldsymbol{H}_i\}$: $\boldsymbol{G}_R = \bigcup_{i=1}^{u} \boldsymbol{H}_i$ with $\boldsymbol{E}(\boldsymbol{H}_i) \cap \boldsymbol{E}(\boldsymbol{H}_j) = \emptyset$ for $i \neq j$.

We are interested in those independent reaction components $\boldsymbol{H}_j$s that contain at least one synthesis reaction. These are called iBD components $\{\boldsymbol{H}_j^{iBD}\}$. The number w of iBD components necessarily does not exceed the number u of connected components in $\boldsymbol{G}_R$: $w \leq u$.

A number of methods can be used to decompose $\boldsymbol{G}_R$ into independent reaction components. For example, the standard disjoint-set data structure and the Union-Find algorithm can be used for this purpose [14]. Another method is to represent $\boldsymbol{G}_R$ by an $m \times m$ adjacency matrix $\boldsymbol{C}$ or a Laplacian matrix $\boldsymbol{L}$. According to spectral graph theory, the connectedness of $\boldsymbol{G}_R$ is encoded in the eigenvalue spectrum of its Laplacian $\boldsymbol{L}$ [13]: the number of connected components of $\boldsymbol{G}_R$ is the multiplicity u of the 0 eigenvalue of $\boldsymbol{L}$, and the corresponding u orthogonal eigenvectors $(\boldsymbol{v}_1, \ldots, \boldsymbol{v}_u)$ gives memberships for reaction to be in each connected independent component. Specifically, the nonzero elements of the vector $\boldsymbol{v}_i$ correspond to the member reactions of an independent reaction component $\boldsymbol{H}_i$ of $\boldsymbol{G}_R$. Algorithm 1 can be used to decompose $\boldsymbol{G}_R$. Additional information on calculating $\boldsymbol{G}_R$ can be found in the appendix.

Relationship between states and iBDs. The iBDs are components of partitioned reactions according to how they share reactants/products, or equivalently, how they contribute to the change of the total mass of the network. The iBDs can be viewed as aggregated reactions and are dictated only by the topology of the network that connects reactions through shared reactants/products. Once the stoichiometry matrix of a reaction network is defined, its iBDs are also determined.

Algorithm 1 Determination of Independent Birth-Death Processes $(\mathcal{X}, \mathcal{R})$.

Network model: $\boldsymbol{O} \leftarrow \{\mathcal{X}, \mathcal{R}\}$;
Initialization of number of iBDs $w = 0$;
Obtain the stoichiometry matrix $\boldsymbol{S}$ of network $\boldsymbol{O}$;
Construct adjacency matrix $\boldsymbol{C}$ of reaction-centered graph $\boldsymbol{G}_R$ following equation (A.1) in the appendix;
Construct degree matrix $\boldsymbol{D}$ of $\boldsymbol{G}_R$ following (A.2) in the appendix;
Construct the Laplacian matrix $\boldsymbol{L}$ following equation (A.3) in the appendix;
Calculate the eigenvalue spectrum of $\boldsymbol{L}$ and obtain the multiplicity u of eigenvalue 0;
Calculate all u orthogonal eigenvectors $\boldsymbol{v}_i$, $i = 1, \ldots, u$ of the eigenvalue 0;
for $i = 1$ to u **do**
  Construct connected reaction sets $\boldsymbol{H}_i = \{R_j | \text{ if } v_{i,j} \neq 0\}$;
end for
for $i = 1$ to u **do**
  if there exists a synthesis or degradation reaction in $\boldsymbol{H}_i$ **then**
    $w \leftarrow w + 1$
    $\boldsymbol{H}_w^{iBD} = \boldsymbol{H}_i$
  end if
end for
Output the number of iBDs and buffers w and iBDs: $\boldsymbol{H}_i^{iBD}$, $i = 1, \ldots, w$.

In contrast, a state is a physical realization of the network at a particular time instance. It describes the number of molecules in the system, regardless of which iBD(s) each may participate. For a mesoscopic system, the state of the system changes with time. It is possible a state can participate in transitions in multiple iBDs. There are many ways states can be aggregated; the aggregations we study in later sections are by the total net number of synthesized molecules in an individual iBD.

ACME multibuffer algorithm for state enumeration. To enumerate the state space more effectively, we introduce the multibuffer state enumeration algorithm for solving the discrete chemical master equation (mb-dCME). We assign a separate buffer queue B_i of size $b_i \in \mathbb{Z}_{\geq 0}$ to each of the ith iBD component. Collectively, they form a buffer bank $\mathcal{B} = (B_1, \ldots, B_w)$. The current sizes of the buffer queues, or the numbers of the remaining buffer tokens, form a vector $\boldsymbol{b} = (b_1, b_2, \ldots, b_w) \in \mathbb{Z}_{\geq 0}^w$. The ith synthesis reaction cannot proceed if the ith buffer queue is exhausted, i.e., $b_i = 0$, resulting in state truncation.

When all iBDs have infinite buffer capacities, we have the infinite buffer bank $\mathcal{I} = (\infty, \infty, \ldots, \infty)$. The infinite state space $\Omega^{(\mathcal{I})}$ associated with buffer bank $\mathcal{I}$ gives the full state space, which will give the exact solution of the dCME: $\Omega^{(\mathcal{I})} \equiv \Omega = \{\boldsymbol{x}(t) | \boldsymbol{x}(0), t \in (0, \infty)\}$. We further use $\mathcal{I}_j = (\infty, \ldots, \infty, B_j, \infty, \ldots, \infty)$ to denote a buffer bank when only the jth iBD is finite with capacity B_j. We can define a partial order $\mathcal{B}' \leq \mathcal{B}''$ for buffer banks if $B_j' \leq B_j''$ for all $j = 1, \ldots, w$. We then have $\mathcal{B} \leq \mathcal{I}_j \leq \mathcal{I}$. We also have $\Omega^{(\mathcal{B})} \subseteq \Omega^{(\mathcal{I}_j)} \subseteq \Omega^{(\mathcal{I})}$.

With the total amount of available computer memory fixed, each enumerated state $\boldsymbol{x} \in \mathbb{Z}_{\geq 0}^n$ is associated with a vector of buffer sizes $\boldsymbol{b}(\boldsymbol{x}) = (b_1(\boldsymbol{x}), b_2(\boldsymbol{x}), \ldots, b_w(\boldsymbol{x}))$, which records the remaining number of unspent tokens in each buffer queue. We can augment the state vector $\boldsymbol{x}$ by concatenating $\boldsymbol{b}(\boldsymbol{x})$ after $\boldsymbol{x}$ to obtain the expanded state vector $\hat{\boldsymbol{x}} = (\boldsymbol{x}, \boldsymbol{b}) \in \mathbb{Z}_{\geq 0}^{n+w}$. With the buffer queues in $\mathcal{B}$ defined, we list the

mb-dCME algorithm in Algorithm 2. The associated transition rate matrix $\boldsymbol{A}$ can also be calculated using Algorithm 2.

Algorithm 2 Multifinite Buffer Optimal State Space Enumeration and Transition Rate Matrix Generation ($\mathcal{X}, \mathcal{R}$, $\{\boldsymbol{H}_i^{iBD}\}$, buffer capacities: $\boldsymbol{b} = (b_1, b_2, \ldots, b_w)$).

Network model: $\boldsymbol{O} \leftarrow \{\mathcal{X}, \mathcal{R}\}$;
Initialization of w iBD processes: $\boldsymbol{H}_1^{iBD}, \boldsymbol{H}_2^{iBD}, \ldots, \boldsymbol{H}_w^{iBD}$;
Buffer capacities: $\boldsymbol{b} = (b_1, b_2, \ldots, b_w)$;
Initial state: $\boldsymbol{x}^{t=0} \leftarrow \{x_1^0, x_2^0, \ldots, x_n^0\}$;
Initialize the state space and the set of transitions: $\Omega \leftarrow \emptyset$; $\boldsymbol{T} \leftarrow \emptyset$;
$\Omega \leftarrow \Omega \cup (\boldsymbol{x}^{t=0}, \boldsymbol{b})$; Stack $ST \leftarrow \emptyset$; Push(ST, $\boldsymbol{x}^{t=0}$);
while $ST \neq \emptyset$ **do**
  $StateGenerated \leftarrow$FALSE; $\boldsymbol{x}_i \leftarrow$ Pop (ST);
  for $k = 1$ to m **do** ▷ There are m reactions.
    for $j = 1$ to w **do** ▷ Look up which iBD reaction R_k belongs to.
      if $R_k \in \boldsymbol{H}_j^{iBD}$ **then**
        Break;
      end if
    end for
    if Reaction R_k can occur in state $\boldsymbol{x}_i$ **then**
      if R_k is a synthesis reaction generating g_k new copies of X_i **then**
        if $b_j \geq g_k$ **then** ▷ Check if buffer tokens are sufficient for synthesis ▷ reaction.
          Generate state $\boldsymbol{x}_j$ that is reached via reaction R_k from $\boldsymbol{x}_i$;
          $b_j \leftarrow b_j - g_k$; $StateGenerated \leftarrow$TRUE;
        end if
      else
        if R_k is a degradation and breaks down d_k copies of X_i **then**
          $b_j \leftarrow b_j + d_k$;
        end if
        Generate state $\boldsymbol{x}_j$ that is reached via reaction R_k from $\boldsymbol{x}_i$;
        $StateGenerated \leftarrow$TRUE;
      end if
      if ($StateGenerated =$ TRUE) **then**
        Combined state $\hat{\boldsymbol{x}}_j = (\boldsymbol{x}_j, \boldsymbol{b})$;
        if ($\hat{\boldsymbol{x}}_j \notin \Omega$) **then**
          $\Omega \leftarrow \Omega \cup \hat{\boldsymbol{x}}_j$; Push($ST$, $\boldsymbol{x}_j$); $\boldsymbol{T} \leftarrow \boldsymbol{T} \cup t_{\boldsymbol{x}_i, \boldsymbol{x}_j}$; ▷ $t_{\boldsymbol{x}_i, \boldsymbol{x}_j}$ records this transition.
          $A(\boldsymbol{x}_i, \boldsymbol{x}_j) \leftarrow$ ReactionRate($\boldsymbol{x}_i$, $\boldsymbol{x}_j$, R_k)
        end if
      end if
    end if
  end for
end while
Output Ω, $\boldsymbol{T}$ and $\boldsymbol{A} = \{A(\boldsymbol{x}_i, \boldsymbol{x}_j)\}$.

Instead of truncating the state space by specifying a maximum allowed copy number B for each individual molecular species as in the conventional hypercube approach, the multibuffer method specifies a maximum allowed copy number B for

each buffer. Assume the jth buffer contains n_j distinct molecular species, and the number of all possible states for the jth buffer is then that of the number of integer lattice nodes in an n_j-dimensional orthogonal corner simplex, with equal length B for all edges starting from the origin. The total number of integer lattice nodes in this n_j-dimensional simplex gives the precise number of states of the jth buffer, which is the multiset number $\binom{B+n_j}{n_j}$. The size of the state space is therefore much smaller than the size of the state space B^{n_j} that would be generated by the hypercube method, with a dramatic reduction factor of roughly $n_j!$ factorial. Note that under the constraint of mass conservation, each molecular species in this buffer can still have a maximum of B copies of molecules. With a conservative assumption that different buffers are independent, the size of the overall truncated state space is then $O(\prod_j \binom{B+n_j}{n_j})$. This is much smaller than the n-dimensional hypercube, which has an overall size of $O(\prod_j B^{n_j}) = O(B^n)$ with n total number of molecular species in the network. Overall, the state spaces generated using the multibuffer algorithm are dramatically smaller than those generated using the conventional hypercube method without loss of resolutions.

2.4. Controlling truncation errors. When one or more buffer queues are exhausted, no new states can be enumerated and synthesis reaction(s) cannot proceed, resulting in errors due to state truncation. Below we describe a theoretical framework for analyzing effects of truncating state space. We give an error estimate such that the truncation error is bounded from above, namely, the actual error will be smaller than the estimated error bound. Furthermore, we give an estimate on the minimal size of buffer required so the truncation error is within a specified tolerance. It is important to note that this error estimate is obtained a priori without computing costly trial solutions. Detailed proofs for all statements of facts can be found in [12].

2.4.1. Overall description. We briefly outline our approach to construct error bounds. We first define truncation error $\mathrm{Err}^{(\mathcal{B})}$ when a finite state space $\Omega^{(\mathcal{B})}$ instead of a full infinite state space $\Omega^{(\mathcal{I})}$ is used to solve the dCME. We then introduce the concept of boundary states $\partial\Omega^{(\mathcal{B})}$ of the state space $\Omega^{(\mathcal{B})}$ and boundary states $\partial\Omega^{(B_j)}$ of the individual jth iBD, as well as the corresponding steady state probabilities $\pi^{(\mathcal{B})}_{\partial,\,\mathcal{B}}$ and $\pi^{(\mathcal{B})}_{\partial,\,B_j}$. We show that the steady state probability $\pi^{(\mathcal{B})}_{\partial,\,\mathcal{B}}$ provides an upper-bound for the truncation error $\mathrm{Err}^{(\mathcal{B})}$. This is established by first examining the truncation error $\mathrm{Err}^{(\mathcal{B}_j)}$ when only one iBD is truncated. The techniques used include (1) permuting the transition rate matrix $\boldsymbol{A}$ and lumping microstates into groups with the same number of net synthesized molecules or buffer usage of the iBD; and (2) constructing a quotient matrix $\boldsymbol{B}$ on the lumped groups from the permuted matrix $\boldsymbol{A}$ and its associated steady state probability distribution. We then show that the truncation error $\mathrm{Err}^{(\mathcal{B}_j)}$ can be asymptotically bounded by $\pi^{(\mathcal{B})}_{\partial,\,B_j}$ computed from the quotient matrix $\boldsymbol{B}$. We further analyze the asymptotic behavior of the boundary probability $\pi^{(\mathcal{B})}_{\partial,\,\mathcal{B}_j}$ and show that this probability increases when additional iBDs are truncated. The upper and lower bounds for truncation error are then obtained based on known facts of stochastic ordering. We then generalize our results on error bounds to truncation errors when two, three, and all buffer queues are of finite capacity.

It is useful to also examine an intuitive picture of the probability landscape governed by a dCME. Starting from an initial condition, the probability mass flows following a diffusion process dictated by the dynamics of the reaction network. At any given time t, the front of the probability flow traces out a boundary ∂_t, which expands to a new boundary $\partial_{t+\Delta t}$ at a subsequent time. Given long enough time, the

probability distribution will reach a steady state. Since the probability flows across the boundaries, we can compare the difference in the probability mass between the boundary surfaces of ∂_t and $\partial_{t+\Delta t}$ to infer how much total probability mass has fluxed out of the finite volume of the state space through its boundary. Our asymptotic analysis is aided by decomposing the overall probability flow into several different fluxes, each governed by a different iBD component.

2.4.2. Truncation error decreases with increasing buffer capacity. Denote the true probability landscape governed by a dCME over $\Omega^{(\mathcal{I})}$ without truncation as $\boldsymbol{p}^{(\mathcal{I})}(t)$. When the state space is truncated to $\Omega^{(\mathcal{B})} \subset \Omega^{(\mathcal{I})}$ using a buffer bank $\mathcal{B}$, the deviation of the summed probability mass of $\boldsymbol{p}^{(\mathcal{I})}(t)$ over $\Omega^{(\mathcal{B})}$ from 1 gives the truncation error:

$$\mathrm{Err}^{(\mathcal{B})}(t) = 1 - \sum_{\boldsymbol{x}\in\Omega^{(\mathcal{B})}} p^{(\mathcal{I})}(\boldsymbol{x}, t) = \sum_{\boldsymbol{x}\in\Omega^{(\mathcal{I})},\, \boldsymbol{x}\notin\Omega^{(\mathcal{B})}} p^{(\mathcal{I})}(\boldsymbol{x}, t). \tag{2.4}$$

As the overall buffer size of $\mathcal{B}$ increases, $\mathrm{Err}^{(\mathcal{B})}(t)$ decreases. Using $\mathrm{Err}^{(\mathcal{B})}$ to denote the steady state error, we have

$$\mathrm{Err}^{(\mathcal{B})} \equiv \mathrm{Err}^{(\mathcal{B})}(t=\infty) = 1 - \sum_{\boldsymbol{x}\in\Omega^{(\mathcal{B})}} \pi^{(\mathcal{I})}(\boldsymbol{x}, t).$$

In addition, we consider error resulting from truncating only the jth buffer queue to the state space $\Omega^{(\mathcal{I}_j)} \subset \Omega^{(\mathcal{I})}$ using buffer bank $\mathcal{I}_j = (\infty, \ldots, \infty, B_j, \infty, \ldots, \infty)$. Similarly, we have

$$\mathrm{Err}^{(\mathcal{I}_j)} = 1 - \sum_{\boldsymbol{x}\in\Omega^{(\mathcal{I}_j)}} \pi^{(\mathcal{I})}(\boldsymbol{x}, t).$$

FACT 1. *For any two truncated state spaces $\Omega^{(\mathcal{B}')}$ and $\Omega^{(\mathcal{B}'')}$, we have $\mathrm{Err}^{(\mathcal{B}')}(t) \geq \mathrm{Err}^{(\mathcal{B}'')}(t)$ if $\mathcal{B}' \leq \mathcal{B}''$ componentwise.*

Note that if $\mathcal{B}' \leq \mathcal{B}'' \leq \mathcal{I}$, then $\mathrm{Err}^{(\mathcal{B}')} \geq \mathrm{Err}^{(\mathcal{B}'')} \geq \mathrm{Err}^{(\mathcal{I})} \equiv 0$.

2.4.3. Probabilities of boundary states of finite state space and increments of truncation error. It is difficult to compute the exact truncation error $\mathrm{Err}^{(\mathcal{B})}(t)$, as it requires $\boldsymbol{p}^{(\mathcal{I})}(t)$ to be known. However, only the computed probability landscape $\boldsymbol{p}^{(\mathcal{B})}(t)$ using a finite state space $\Omega^{(\mathcal{B})}$ is known.

We now consider the steady state probabilities $\boldsymbol{\pi}^{(\mathcal{I})} \equiv \boldsymbol{p}^{(\mathcal{I})}(\infty)$, $\boldsymbol{\pi}^{(\mathcal{I}_j)} \equiv \boldsymbol{p}^{(\mathcal{I}_j)}(\infty)$, and $\boldsymbol{\pi}^{(\mathcal{B})} \equiv \boldsymbol{p}^{(\mathcal{B})}(\infty)$. We further consider the boundary states $\partial\Omega^{(\mathcal{B})}$ of $\Omega^{(\mathcal{B})}$ and show that $\pi^{(\mathcal{B})}(\partial\Omega^{(\mathcal{B})})$ can be used as a surrogate for estimating the steady state error $\mathrm{Err}^{(\mathcal{B})}$ and for assessing the convergence behavior of $\mathrm{Err}^{(\mathcal{B})}$.

Boundary of state space $\Omega^{(\mathcal{B})}$ and boundary states of the jth iBD. The boundary states $\partial\Omega^{(\mathcal{B})}$ of $\Omega^{(\mathcal{B})}$ are those states with at least one depleted buffer queue:

$$\partial\Omega^{(\mathcal{B})} = \{\boldsymbol{x} | b_i \text{ of } \boldsymbol{b}(\boldsymbol{x}) = 0,\, i \in (1, \ldots, w)\}, \tag{2.5}$$

i.e., there are exactly B_i net synthesized molecules for at least one of the iBDs. The time-evolving and steady state probability mass of $p^{(\mathcal{I})}(t)$ over $\partial\Omega^{(\mathcal{B})}$ is denoted as $p^{(\mathcal{I})}_{\partial,\mathcal{B}}(t)$ and $\pi^{(\mathcal{I})}_{\partial,\mathcal{B}}$, respectively.

In addition to boundary states of the full buffer bank, we also consider boundary states of individual buffer queues. We consider a subset of the boundary states $\partial\Omega^{(\mathcal{B})}_{(B_j)} \in \partial\Omega^{(\mathcal{B})}$ that are associated with the jth iBD component:

$$\partial\Omega^{(\mathcal{B})}_{(B_j)} \equiv \{\boldsymbol{x} | b_j \text{ of } \boldsymbol{b}(\boldsymbol{x}) = 0\}. \tag{2.6}$$

Probabilities of boundary states of $\partial\Omega^{(\mathcal{B})}$ and $\partial\Omega^{(\mathcal{B})}_{(B_j)}$. We name the summation of the true steady state probability $\pi^{(\mathcal{I})}(\boldsymbol{x})$ over all boundary states $\partial\Omega^{(\mathcal{B})}$ the *true total boundary probability* $\pi^{(\mathcal{I})}_{\partial,\,\mathcal{B}}$:

$$\pi^{(\mathcal{I})}_{\partial,\,\mathcal{B}} \equiv \sum_{\boldsymbol{x}\in\partial\Omega^{(\mathcal{B})}} \pi^{(\mathcal{I})}(\boldsymbol{x}).$$

The summation of the computed probability $\pi^{(\mathcal{B})}(\boldsymbol{x})$ using the truncated state space $\Omega^{(\mathcal{B})}$ over the same boundary states $\partial\Omega^{(\mathcal{B})}$ is the *computed total boundary probability* $\pi^{(\mathcal{B})}_{\partial,\,\mathcal{B}}$:

$$\pi^{(\mathcal{B})}_{\partial,\,\mathcal{B}} \equiv \sum_{\boldsymbol{x}\in\partial\Omega^{(\mathcal{B})}} \pi^{(\mathcal{B})}(\boldsymbol{x}).$$

Similarly, we call the summation of the true probability $\pi^{(\mathcal{I})}(\boldsymbol{x})$ associated with the boundary states of the jth iBD the *true boundary probability of jth iBD* $\pi^{(\mathcal{I})}_{\partial,\,B_j}$:

$$\pi^{(\mathcal{I})}_{\partial,\,B_j} \equiv \sum_{\boldsymbol{x}\in\partial\Omega^{(\mathcal{B})}_{(B_j)}} \pi^{(\mathcal{I})}(\boldsymbol{x}).$$

The summation of the computed probability $\pi^{(\mathcal{B})}(\boldsymbol{x})$ using the truncated state space $\Omega^{(\mathcal{B})}$ over the same boundary states associated with the jth iBD in $\partial\Omega^{(\mathcal{B})}_{(B_j)}$ is the *computed boundary probability of the jth iBD* $\pi^{(\mathcal{B})}_{\partial,\,B_j}$:

$$\pi^{(\mathcal{B})}_{\partial,\,B_j} \equiv \sum_{\boldsymbol{x}\in\partial\Omega^{(\mathcal{B})}_{(B_j)}} \pi^{(\mathcal{B})}(\boldsymbol{x}).$$

It is also useful to examine the total boundary probability $\pi^{(\mathcal{I}_j)}_{\partial,\,B_j}$ of the jth iBD on the state space $\Omega^{(\mathcal{I}_j)}$:

$$\pi^{(\mathcal{I}_j)}_{\partial,B_j} \equiv \sum_{\boldsymbol{x}\in\partial\Omega^{(\mathcal{I}_j)}_{B_j}} \pi^{(\mathcal{I}_j)}(\boldsymbol{x}).$$

Note that when B_j goes to infinity, the probability $\pi^{(\mathcal{I}_j)}_{\partial,B_j}$ approaches $\pi^{(\mathcal{I})}_{\partial,B_j}$.

Incremental truncation errors. The state space $\Omega^{(\mathcal{B})}$ is obtained from enumeration by adding 1 to the capacity of every buffer queue used to obtain the state space $\Omega^{(\mathcal{B}-\mathbb{1})}$. Let $\mathbb{1} = (1, 1, \ldots, 1) \in \mathbb{Z}^w$. The boundary of $\Omega^{(\mathcal{B})}$ can then be written as $\partial\Omega^{(\mathcal{B})} = \Omega^{(\mathcal{B})} - \Omega^{(\mathcal{B}-\mathbb{1})}$. It is obvious that the true total boundary probability $\pi^{(\mathcal{I})}_{\partial,\,\mathcal{B}}$ is the increment of the truncation error between $\Omega^{(\mathcal{B}-\mathbb{1})}$ and $\Omega^{(\mathcal{B})}$:

$$\pi^{(\mathcal{I})}_{\partial,\,\mathcal{B}} = \Delta\,\mathrm{Err}^{(\mathcal{B})} = \mathrm{Err}^{(\mathcal{B}-\mathbb{1})} - \mathrm{Err}^{(\mathcal{B})}\,. \tag{2.7}$$

Figure 1 gives an illustration.

Let $\boldsymbol{e}_j = (0, \ldots, 0, 1, 0, \ldots, 0) \in \mathbb{Z}^w_{\geq 0}$ be an elementary vector with only the jth element as 1 and all others 0. The boundary states of the jth iBD is given by $\partial\Omega^{(\mathcal{B})}_{(B_j)} = \Omega^{(\mathcal{B})} - \Omega^{(\mathcal{B}-\boldsymbol{e}_j)}$. Analogous to (2.7), the boundary probability $\pi^{(\mathcal{I})}_{\partial,\,B_j}$ is therefore the increment of the truncation error between $\Omega^{(\mathcal{B}-\boldsymbol{e}_j)}$ and $\Omega^{(\mathcal{B})}$,

$$\pi^{(\mathcal{I})}_{\partial,\,B_j} = \Delta\,\mathrm{Err}^{(B_j)} = \mathrm{Err}^{(\mathcal{B}-\boldsymbol{e}_j)} - \mathrm{Err}^{(\mathcal{B})}, \tag{2.8}$$

as the only difference between $\Omega^{(\mathcal{B}-\boldsymbol{e}_j)}$ and $\Omega^{(\mathcal{B})}$ are those states containing exactly

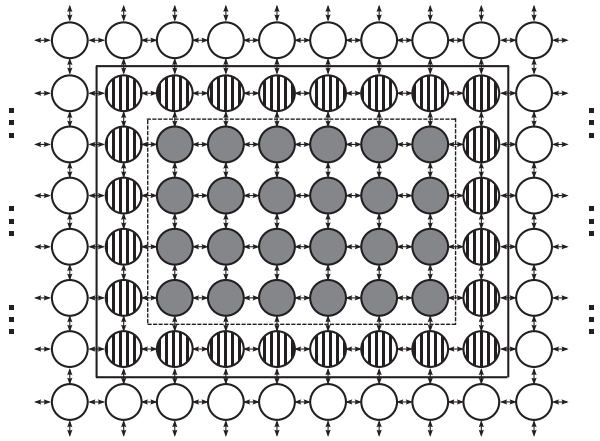

FIG. 1. *Probability of boundary states and truncation errors. The gray states in the center box of dashed lines form the state space $\Omega^{(\mathcal{B}-\mathbb{1})}$, with the rest as states truncated from $\Omega^{(\mathcal{B}-\mathbb{1})}$. The stripe-filled states are the newly added states when the buffer capacity is increased from $\mathcal{B}-\mathbb{1}$ to $\mathcal{B}$. These new states and those gray states, both enclosed in the box in solid lines, form the state space $\Omega^{(\mathcal{B})}$. The summed true probability mass over the white states outside the solid-lined box is the error $\mathrm{Err}^{(\mathcal{B})}$ of the truncated state space $\Omega^{(\mathcal{B})}$. The summed true probability mass over all states outside of the dashed line box is the error $\mathrm{Err}^{(\mathcal{B}-\mathbb{1})}$. The summed true probability over stripe-filled states $\pi^{(\mathcal{I})}_{\partial,\mathcal{B}}$ is the incremental error $\Delta\,\mathrm{Err}^{(\mathcal{B})}(t)$ when the buffer capacity of all buffer queues is increased by 1 from $\mathcal{B}-\mathbb{1}$. We have $\pi^{(\mathcal{I})}_{\partial,\mathcal{B}} = \Delta\,\mathrm{Err}^{(\mathcal{B})}(t) = |\,\mathrm{Err}^{(\mathcal{B})}(t) - \mathrm{Err}^{(\mathcal{B}-\mathbb{1})}(t)|$.*

B_j net synthesized molecules in the jth iBD, namely, the states with the jth buffer queue depleted.

Total true error is no greater than summed errors over all iBDs. Overall, we have $\partial\Omega^{(\mathcal{B})} = \bigcup_{j=1}^{w} \partial\Omega^{(\mathcal{B})}_{(B_j)}$. As some boundary states may have multiple depleted buffer queues, it is possible that $\partial\Omega^{(\mathcal{B})}_{(B_i)} \cap \partial\Omega^{(\mathcal{B})}_{(B_j)} \neq \emptyset, \ldots, \bigcap_{i=1}^{w} \partial\Omega^{(\mathcal{B})}_{(B_i)} \neq \emptyset$. Therefore, the actual total boundary probability $\pi^{(\mathcal{I})}_{\partial,\mathcal{B}}$ is smaller than or equal to the summation of individual $\pi^{(\mathcal{I})}_{\partial,B_j}$:

$$\pi^{(\mathcal{I})}_{\partial,\mathcal{B}} \leq \sum_{j=1}^{w} \pi^{(\mathcal{I})}(\partial\Omega^{(\mathcal{B})}_{(B_j)}) \equiv \sum_{j=1}^{w} \pi^{(\mathcal{I})}_{\partial,B_j}. \tag{2.9}$$

As the state space $\Omega^{(\mathcal{B})} = \bigcap_{i=1}^{w} \Omega^{(\mathcal{I}_j)}$ and the buffer capacity of the jth iBD in $\Omega^{(\mathcal{I}_j)}$ is the same as that in $\Omega^{(\mathcal{B})}$, we have that the total true error of the state space $\Omega^{(\mathcal{B})}$ is bounded by the summation of true errors from individually truncated state spaces $\Omega^{(\mathcal{I}_j)}$:

$$\mathrm{Err}^{(\mathcal{B})} \leq \sum_{j=1}^{w} \mathrm{Err}^{(\mathcal{I}_j)}. \tag{2.10}$$

An example. Figure 2 shows an example of the enumerated state space using Algorithm 2 for a simple network with reversible reactions $\emptyset \rightleftharpoons X$ and $\emptyset \rightleftharpoons Y$. The network is partitioned into two iBD components, one for $\emptyset \rightleftharpoons X$ and another for $\emptyset \rightleftharpoons Y$. A buffer bank $\mathcal{B} = (B_1, B_2)$ with two buffer queues is assigned to the network, with the size vector $(B_1, B_2) = (7, 5)$. A synthesis reaction is halted once

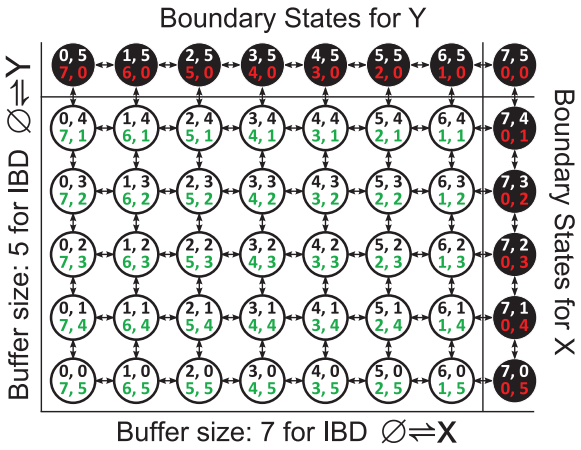

FIG. 2. *An illustration of the enumerated state space and the boundary states of a simple network with two reactions* $\emptyset \rightleftharpoons X$ *and* $\emptyset \rightleftharpoons Y$. *There are two iBDs in this network, with two buffer queues* B_1 *and* B_2 *of size* 5 *and* 7 *assigned to the first and second iBD, respectively. Each circle represents an enumerated state. Filled circles are boundary states, in which at least one of the two buffer queues is depleted. There are four integers inside each circle. The two at the top are copy numbers* x *and* y *of molecular species* X *and* Y, *namely,* $\boldsymbol{x} = (x, y)$. *The two at the bottom are the remaining numbers* b_1 *and* b_2 *of tokens in the buffer queues* B_1 *and* B_2, *namely,* $\boldsymbol{b} = (b_1, b_2)$.

its buffer queue is depleted, resulting in truncation error. Boundary states, in which at least one of the two buffer queues is depleted, are shown as filled black circles, with states of the buffer queues shown in red numbers. The union of all black filled circles in Figure 2 form the boundary $\partial\Omega^{(\mathcal{B})}$ of the state space. The boundary states associated with the buffer queue corresponding to the iBD of reaction $\emptyset \rightleftharpoons X$ are

$$\partial\Omega^{(\mathcal{B})}_{(B_1)} = \{(x = 7, y = 5), (x = 7, y = 4), (x = 7, y = 3), \\ (x = 7, y = 2), (x = 7, y = 1), (x = 7, y = 0)\}$$

in which the buffer queue B_1 is depleted. The boundary states associated with the iBD of reaction $\emptyset \rightleftharpoons Y$ are

$$\partial\Omega^{(\mathcal{B})}_{(B_2)} = \{(x = 7, y = 5), (x = 6, y = 5), (x = 5, y = 5), (x = 4, y = 5), \\ (x = 3, y = 5), (x = 2, y = 5), (x = 1, y = 5), (x = 0, y = 5)\},$$

in which the buffer queue B_2 is depleted (Figure 2). We have $\partial\Omega^{(\mathcal{B})} = \partial\Omega^{(\mathcal{B})}_{(B_1)} \cup \partial\Omega^{(\mathcal{B})}_{(B_2)}$. We also observe that $\partial\Omega^{(\mathcal{B})}_{(B_1)} \cap \partial\Omega^{(\mathcal{B})}_{(B_2)} = (x = 7, y = 5)$ is nonempty. Those states that are not on the boundary are shown as unfilled circles.

2.4.4. Bounding errors due to a truncated buffer queue. We show how to construct an error bound after truncating an individual buffer queue. We first examine the steady state boundary probability $\pi^{(\mathcal{I}_j)}_{\partial, B_j}$. For ease of discussion, we use N instead of B_j to denote the buffer capacity of the jth iBD and use $\pi^{(\mathcal{I}_j)}_N \equiv \pi^{(\mathcal{I}_j)}_{\partial,\, B_j}$ to denote the boundary probability of $\Omega^{(\mathcal{I}_j)}$. The true error $\mathrm{Err}^{(\mathcal{I}_j)}$ associated with buffer bank $\mathcal{I}_j = (\infty, \ldots, \infty, B_j = N, \infty, \ldots, \infty)$ for the steady state is unknown, as it requires knowledge of $\pi^{(\mathcal{I})}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \Omega^{(\mathcal{I})}$. Here, we show that $\mathrm{Err}^{(\mathcal{I}_j)}$ converges to the true boundary probability $\pi^{(\mathcal{I}_j)}_N$ asymptotically as the size of the buffer queue

N increases. Specifically, if the size of the buffer queue is sufficiently large, $\mathrm{Err}^{(\mathcal{I}_j)}$ is bounded by $\pi_N^{(\mathcal{I}_j)}$ up to a constant factor. As N further increases, $\mathrm{Err}^{(\mathcal{I}_j)}$ converges to $\pi_N^{(\mathcal{I}_j)}$.

Aggregating states by buffer queue usage. To show how boundary probability $\pi_N^{(\mathcal{I}_j)}$ can be used to construct a truncation error bound, we first aggregate states in the original state space $\Omega^{(\mathcal{I}_j)}$ into $N+1$ nonintersecting subsets according to the net number of tokens in use from buffer B_j: $\Omega^{(\mathcal{I}_j)} \equiv \{\mathcal{G}_0, \mathcal{G}_1, \ldots, \mathcal{G}_N\}$. Here states in each aggregated subset $\mathcal{G}_s \subseteq \Omega^{(\mathcal{I}_j)}$, $s = 1, \ldots, N$, all have the same s number of buffer tokens spent from buffer queue B_j, or equivalently, $(N-s)$ tokens unused in buffer B_j. Note that each $\mathcal{G}_s$ can be of infinite size if the capacity of any other buffer queues are infinite. Conceptually disregarding the practical issue of time complexity for now, the states in the state space Ω can be sorted according to the buffer token from buffer queue B_j in use. This can be done using any sorting algorithm, such as the bucket sort algorithm with $N+1$ buckets, with each bucket $\mathcal{G}_s$ containing only states with exactly s buffer tokens spent.

With this partition, we can construct a transition rate matrix $\tilde{\boldsymbol{A}}$ from the sorted state space $\Omega^{(\mathcal{I}_j)}$. The new transition rate matrix $\tilde{\boldsymbol{A}}$ is a permutation of the original dCME matrix $\boldsymbol{A}$ equation (2.2):

$$\tilde{\boldsymbol{A}} = \begin{pmatrix} \boldsymbol{A}_{0,0} & \boldsymbol{A}_{0,1} & \cdots & \boldsymbol{A}_{0,N} \\ \boldsymbol{A}_{1,0} & \boldsymbol{A}_{1,1} & \cdots & \boldsymbol{A}_{1,N} \\ \cdots & \cdots & \cdots & \cdots \\ \boldsymbol{A}_{N,0} & \boldsymbol{A}_{N,1} & \cdots & \boldsymbol{A}_{N,N} \end{pmatrix}, \tag{2.11}$$

where each block submatrix $\boldsymbol{A}_{i,j}$ includes all transitions from states in group $\mathcal{G}_j$ to states in group $\mathcal{G}_i$ and can be defined as $\boldsymbol{A}_{i,j} = \{\boldsymbol{a}_{m,n}\}_{||\mathcal{G}_i|| \times ||\mathcal{G}_j||}$, and each entry $\boldsymbol{a}_{m,n}$ in $\boldsymbol{A}_{i,j}$ is the transition rate from a state $\boldsymbol{x}_n \in \mathcal{G}_j$ to a state $\boldsymbol{x}_m \in \mathcal{G}_i$.

Although in principle one can obtain the sorted state space partition $\Omega^{(\mathcal{I}_j)} \equiv \{\mathcal{G}_0, \mathcal{G}_1, \ldots, \mathcal{G}_N\}$ and the permuted transition rate matrix $\tilde{\boldsymbol{A}}$, there is no need to do so in practice. The construction of $\Omega^{(\mathcal{I}_j)}$ and $\tilde{\boldsymbol{A}}$ only serves the purpose for proving lemmas and theorems. Specifically, we only need to know that conceptually the original state space can be sorted and partitioned, and a permuted transition rate matrix $\tilde{\boldsymbol{A}}$ can be constructed from the sorted state space according to the aggregation.

Assuming the partition and the steady state probability distribution over the state space $\Omega^{(\mathcal{I}_j)}$ are known, we can construct an aggregated synthesis rate $\alpha_i^{(N)}$ for the group $\mathcal{G}_i$ and an aggregated degradation rate $\beta_{i+1}^{(N)}$ for the group $\mathcal{G}_{i+1}$ at the steady state as two constants (Figure 3):

$$\alpha_i^{(N)} \equiv \left(\mathbb{1}^T \boldsymbol{A}_{i+1,i}\right) \cdot \frac{\boldsymbol{\pi}^{(\mathcal{I}_j)}(\mathcal{G}_i)}{\mathbb{1}^T \boldsymbol{\pi}^{(\mathcal{I}_j)}(\mathcal{G}_i)} \quad \text{and} \quad \beta_{i+1}^{(N)} \equiv \left(\mathbb{1}^T \boldsymbol{A}_{i,i+1}\right) \cdot \frac{\boldsymbol{\pi}^{(\mathcal{I}_j)}(\mathcal{G}_{i+1})}{\mathbb{1}^T \boldsymbol{\pi}^{(\mathcal{I}_j)}(\mathcal{G}_{i+1})}, \tag{2.12}$$

where vector $\boldsymbol{\pi}^{(\mathcal{I}_j)}(\mathcal{G}_i)$ and $\boldsymbol{\pi}^{(\mathcal{I}_j)}(\mathcal{G}_{i+1})$ are steady state probability vectors over the permuted microstates in the lumped group $\mathcal{G}_i$ and $\mathcal{G}_{i+1}$, respectively. Row vectors $\mathbb{1}^T \boldsymbol{A}_{i+1,i}$ and $\mathbb{1}^T \boldsymbol{A}_{i,i+1}$ are summed columns of block submatrices $\boldsymbol{A}_{i+1,i}$ and $\boldsymbol{A}_{i,i+1}$, respectively.

Similarly, if the buffer queue B_j has infinite capacity, we have

$$\alpha_i^{(\infty)} \equiv \left(\mathbb{1}^T \boldsymbol{A}_{i+1,i}\right) \cdot \frac{\boldsymbol{\pi}^{(\mathcal{I})}(\mathcal{G}_i)}{\mathbb{1}^T \boldsymbol{\pi}^{(\mathcal{I})}(\mathcal{G}_i)} \quad \text{and} \quad \beta_{i+1}^{(\infty)} \equiv \left(\mathbb{1}^T \boldsymbol{A}_{i,i+1}\right) \cdot \frac{\boldsymbol{\pi}^{(\mathcal{I})}(\mathcal{G}_{i+1})}{\mathbb{1}^T \boldsymbol{\pi}^{(\mathcal{I})}(\mathcal{G}_{i+1})}. \tag{2.13}$$

We can then construct an aggregated transition rate matrix $\boldsymbol{B}$ from the permuted matrix $\tilde{\boldsymbol{A}}$ based on Fact 2.

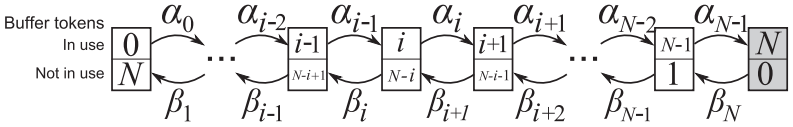

FIG. 3. *The birth-death system associated with the aggregated rate matrix* $\boldsymbol{B}$. *Each box represents an aggregated state consisting of all microstates with the same number of buffer tokens in use. The top half of each box lists the number of buffer tokens in use, and the bottom half lists the number of remaining free buffer tokens in the buffer queue. The gray box contains the boundary states. The total number of spent and free tokens sums to the buffer capacity* N. *These aggregated states are connected by aggregated birth and death reactions, with apparent synthesis rates* α_i *and degradation rates* β_{i+1} *(see Fact 2).*

FACT 2. *Consider a homogeneous continuous-time Markov process with the infinitesimal generator rate matrix* $\boldsymbol{A}$ *on the infinite state space* $\Omega^{(\mathcal{I}_j)}$ *equipped with buffer queues* $\mathcal{I}_j = (\infty, \ldots, B_j, \ldots, \infty)$ *with a finite buffer capacity* $B_j = N$ *for the* j*th iBD and infinite capacities for all other iBDs. Denote its steady state probability distribution as* $\boldsymbol{\pi}^{(\mathcal{I}_j)} \equiv \boldsymbol{\pi}(\Omega^{(\mathcal{I}_j)})$. *An aggregated continuous-time Markov process with a finite size rate matrix* $\boldsymbol{B}_{(N+1)\times(N+1)}$ *can be constructed on the partition* $\tilde{\Omega}_{B_j}^{(\mathcal{I}_j)} = \{\mathcal{G}_0, \mathcal{G}_1, \ldots, \mathcal{G}_N\}$ *with respect to the buffer queue* B_j. *Denote* $\tilde{\pi}_s^{(N)} \equiv \tilde{\pi}(\mathcal{G}_s) = \sum_{\boldsymbol{x}\in\mathcal{G}_s} \pi^{(\mathcal{I}_j)}(\boldsymbol{x})$. *The steady state probability vector* $\tilde{\boldsymbol{\pi}}(\tilde{\Omega}_{B_j}^{(\mathcal{I}_j)}) = (\tilde{\pi}_0^{(N)}, \ldots, \tilde{\pi}_N^{(N)}) = (\tilde{\pi}(\mathcal{G}_0), \ldots, \tilde{\pi}(\mathcal{G}_N))$ *of the aggregated Markov process gives the same steady state probability distribution for the partitioned groups* $\{\mathcal{G}_s\}$ *as that given by the original matrix* $\boldsymbol{A}$ *for all* $s = 0, 1, \ldots, N$. *Furthermore, the* $(N+1)\times(N+1)$ *transition rate matrix* $\boldsymbol{B}$ *can be constructed as*

$$\boldsymbol{B}^{(N)} = \left(\boldsymbol{\alpha}^{(N)}, \boldsymbol{\gamma}^{(N)}, \boldsymbol{\beta}^{(N)} \right) \tag{2.14}$$

with the lower off-diagonal vector

$$\boldsymbol{\alpha}^{(N)} = (\alpha_i^{(N)}),\ i = 0, \ldots, N-1,$$

the upper off-diagonal vector

$$\boldsymbol{\beta}^{(N)} = (\beta_i^{(N)}),\ i = 1, \ldots, N,$$

and the diagonal vector

$$\boldsymbol{\gamma}^{(N)} = (\gamma_i^{(N)}) = (-\alpha_i^{(N)} - \beta_i^{(N)}),\ i = 0, \ldots, N.$$

It is equivalent to transforming the transition rate matrix $\tilde{\boldsymbol{A}}$ *in* (2.11) *to* $\boldsymbol{B}$ *by substituting each block submatrix* $\boldsymbol{A}_{i+1,\,i}$ *of synthesis reactions with the corresponding aggregated synthesis rate* $\alpha_i^{(N)}$ *and each block* $\boldsymbol{A}_{i,\,i+1}$ *of degradation reactions with the aggregated degradation rate* $\beta_{i+1}^{(N)}$ *in* (2.12), *respectively.*

A detailed proof for Fact 2 can be found in Lemma 1 in [12].

Computing steady state boundary probabilities. Following [73, 78] on birth-death processes (Figure 3), the analytic solution for the steady state $\tilde{\pi}_i^{(N)}$ and $\tilde{\pi}_0^{(N)}$ can be written as

$$\tilde{\pi}_i^{(N)} = \prod_{k=0}^{i-1} \frac{\alpha_k^{(N)}}{\beta_{k+1}^{(N)}} \tilde{\pi}_0^{(N)} \tag{2.15}$$

and

$$\tilde{\pi}_0^{(N)} = \frac{1}{1 + \sum_{j=1}^{N} \prod_{k=0}^{j-1} \frac{\alpha_k^{(N)}}{\beta_{k+1}^{(N)}}}. \tag{2.16}$$

The boundary probability $\tilde{\pi}_N^{(N)}$ is then

$$\tilde{\pi}_N^{(N)} \equiv \pi_{\partial, B_j}^{(\mathcal{I}_j)} = \frac{\prod_{k=0}^{N-1} \frac{\alpha_k^{(N)}}{\beta_{k+1}^{(N)}}}{1 + \sum_{j=1}^{N} \prod_{k=0}^{j-1} \frac{\alpha_k^{(N)}}{\beta_{k+1}^{(N)}}}. \tag{2.17}$$

If we have infinite buffer capacity for the jth iBD, we will have the true probability mass over the same fixed set of states in $\mathcal{G}_N$ as

$$\tilde{\pi}_N^{(\infty)} \equiv \tilde{\pi}_N^{(\mathcal{I})} \equiv \tilde{\pi}_{\partial, B_j}^{(\mathcal{I})} = \frac{\prod_{k=0}^{N-1} \frac{\alpha_k^{(\infty)}}{\beta_{k+1}^{(\infty)}}}{1 + \sum_{j=1}^{\infty} \prod_{k=0}^{j-1} \frac{\alpha_k^{(\infty)}}{\beta_{k+1}^{(\infty)}}}. \tag{2.18}$$

Boundary probability as error bound of state truncation. According to Fact 1, the error $\mathrm{Err}^{(\mathcal{I}_j)}$ converges to 0 as the buffer capacity $B_j = N$ increases to infinity. For a truncated state space, the series of the true boundary probabilities $\{\tilde{\pi}_N^{(\mathcal{I})} | N = 1, 2, \ldots, \}$ (equation (2.18)) also converges to 0, as the sequence of its partial sums converges to 1. That is, the Nth member $\tilde{\pi}_N^{(\mathcal{I})}$ of this series converges to 0 while the residual sum of this series $\mathrm{Err}^{(\mathcal{I}_j)} \equiv \sum_{i=N+1}^{\infty} \tilde{\pi}_i^{(\infty)}$ also converges to 0.

We now examine the convergence behavior of the truncation error $\mathrm{Err}_{(N)}^{(\mathcal{I}_j)}$ and the true boundary probability $\tilde{\pi}_N^{(\infty)}$.

FACT 3. *For a truncated state space associated with a buffer bank $\mathcal{I}_j$, if the buffer capacity N for queue B_j increases to infinity, the truncation error of B_j obeys the following inequality:*

$$\mathrm{Err}_{(N)}^{(\mathcal{I}_j)} \leq \frac{\alpha_N^{(\infty)} / \beta_{N+1}^{(\infty)}}{1 - \alpha_N^{(\infty)} / \beta_{N+1}^{(\infty)}} \cdot \tilde{\pi}_{\partial, B_j}^{(\mathcal{I})}. \tag{2.19}$$

A detailed proof for Fact 3 can be found in Theorem 1 in [12].

That is, the true error $\mathrm{Err}_{(N)}^{(\mathcal{I}_j)}$ is bounded by a simple function of $\alpha_N^{(\infty)}$ and $\beta_{N+1}^{(\infty)}$ multiplied by the boundary probability $\tilde{\pi}_{\partial, B_j}^{(\mathcal{I})}$. We can use this inequality to construct an upper-bound for $\mathrm{Err}_{(N)}^{(\mathcal{I}_j)}$. We take advantage of the following fact.

FACT 4. *For any biological system in which the total amount of mass is finite, e.g., cells with finite mass and growth, the aggregated synthesis rate $\alpha_N^{(\infty)}$ becomes smaller than the aggregated degradation rate $\beta_{N+1}^{(\infty)}$ when the buffer capacity N is sufficiently large:*

$$\lim_{N \to \infty} \frac{\alpha_N^{(\infty)}}{\beta_{N+1}^{(\infty)}} < 1.$$

A detailed proof for Fact 4 can be found in Lemma 2 in [12].

Let $C \equiv \frac{\alpha_N^{(\infty)}/\beta_{N+1}^{(\infty)}}{1-\alpha_N^{(\infty)}/\beta_{N+1}^{(\infty)}}$. If $\alpha_N^{(\infty)}/\beta_{N+1}^{(\infty)} < 0.5$, we have $C < 1$, and the true error $\mathrm{Err}_{(N)}^{(\mathcal{I}_j)}$ is always less than the true boundary probability $\tilde{\pi}_{\partial,B_j}^{(\mathcal{I})}$. If $\alpha_N^{(\infty)}/\beta_{N+1}^{(\infty)} = 0.5$, then $C = 1$, and the true error converges asymptotically to the true boundary probability $\tilde{\pi}_{\partial,B_j}^{(\mathcal{I})}$. If $0.5 < \frac{\alpha_N^{(\infty)}}{\beta_{N+1}^{(\infty)}} < 1.0$, then $C > 1$, and the error is larger than $\tilde{\pi}_{\partial,B_j}^{(\mathcal{I})}$ but is bounded by $\tilde{\pi}_{\partial,B_j}^{(\mathcal{I})}$ up to the constant factor $C \equiv \frac{\alpha_N^{(\infty)}/\beta_{N+1}^{(\infty)}}{1-\alpha_N^{(\infty)}/\beta_{N+1}^{(\infty)}}$. Therefore, we can conclude that the true boundary probability $\tilde{\pi}_{\partial,B_j}^{(\mathcal{I})}$ provides an error bound to the state space truncation.

Note that in real biological reaction networks, the inequality $\alpha_N^{(\infty)}/\beta_{N+1}^{(\infty)} < 0.5$ usually holds when buffer capacity N is sufficiently large. This is because synthesis reactions usually have constant rates, while rates of degradation reactions depend on the copy number of net molecules in the network. As a result, the ratio between aggregated synthesis and degradation rates decreases monotonically when the total number of molecules in the system increases.

2.4.5. True boundary probability and computed boundary probability on truncated space. However, it is not possible to calculate the true boundary probability $\tilde{\pi}_{\partial,B_j}^{(\mathcal{I})}$ on the infinite state space. We have the following fact.

FACT 5. *The total probability $\tilde{\pi}_N^{(\mathcal{I}_j)}$ of the boundary states $\partial\Omega_{B_j}^{(\mathcal{I}_j)}$ of the jth iBD with buffer capacity $B_j \equiv N$ obtained from the truncated state space $\Omega^{(\mathcal{I}_j)}$ is greater than or equal to the true probability $\tilde{\pi}_N^{(\mathcal{I})}$ over the same boundary states, i.e., $\tilde{\pi}_N^{(\mathcal{I})} \leq \tilde{\pi}_N^{(\mathcal{I}_j)}$.*

A detailed proof for Fact 5 can be found in Theorem 2 in [12].

We can therefore conclude that the true boundary probability is no greater than the truncated boundary probability given in (2.17) in the general case when $\alpha_i^{(N)} \neq 0$ and $\beta_{i+1}^{(N)} \neq 0$. We further consider two additional cases. When reactions associated with the jth iBD have zero synthesis and nonzero degradation constants, namely, $\alpha_i^{(N)} = 0$ and $\beta_{i+1}^{(N)} \neq 0$, the aggregated system with respect to jth iBD is a death process and there is no synthesis reactions. The associated iBD is closed and a finite buffer works once all states of the closed iBD are enumerated. When reactions associated with the jth iBD have nonzero synthesis but zero degradation constants, we have $\alpha_i^{(N)} \neq 0$ but $\beta_{i+1}^{(N)} = 0$. The aggregated system with respect to the jth iBD is a birth process without degradation reactions. In this case, the error for the time-evolving probability can be estimated using a Poisson distribution with parameter $\alpha_i^{(N)} \cdot t$, where $\alpha_i^{(N)}$ is the maximum aggregated rate, and t is the elapsed time used for computing the time evolution of the probability landscape [20, 82]. We dispense with details here.

2.4.6. Bounding errors when truncating multiple buffer queues. We now consider truncating one additional buffer queue at the ith iBD. We denote the buffer bank as $\mathcal{I}_{i,j} = (\infty, \ldots, B_i, \ldots, B_j, \ldots, \infty)$ with B_i and B_j as the buffer capacities of the ith and jth iBDs, respectively. The rest of the buffer queues all have infinite capacities. We denote the corresponding state space as $\Omega^{\mathcal{I}_{i,j}}$, the transition rate matrix as $\boldsymbol{A}^{\mathcal{I}_{i,j}}$, and the steady state probability distribution as $\boldsymbol{\pi}^{\mathcal{I}_{i,j}}$. We have the fact that the probability of each state in the state space $\Omega^{\mathcal{I}_{i,j}}$ is no less than the corresponding probability on $\Omega^{\mathcal{I}_j}$, i.e., $\pi^{\mathcal{I}_{i,j}}(\boldsymbol{x}) \geq \pi^{\mathcal{I}_j}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \Omega^{\mathcal{I}_{i,j}}$.

FACT 6. *At steady state,* $\boldsymbol{\pi}^{\mathcal{I}_{i,j}} \geq \boldsymbol{\pi}^{\mathcal{I}_j}$ *and* $\boldsymbol{\pi}^{\mathcal{I}_{i,j}} \to \boldsymbol{\pi}^{\mathcal{I}_j}$ *componentwise over state space* $\Omega^{\mathcal{I}_{i,j}}$ *when buffer capacity* $B_i \to \infty$.

A detailed proof for Fact 6 can be found in Theorem 3 in [12].

That is, the computed boundary probability of the jth iBD after introducing an additional truncation at the ith iBD will be no smaller than when the buffer capacity is sufficiently large. Therefore, the boundary probability from double truncated state space $\Omega^{\mathcal{I}_{i,j}}$ can be conservatively and safely used to bound the truncation error. We can further show by induction that boundary probability computed from state space truncated at multiple iBDs $\Omega^{(\mathcal{B})}$ will not be smaller and therefore can be used to bound the true boundary probabilities.

Error bound inequality. According to (2.10) and Facts 1–6, we have the following inequality to bound the true error of state space truncation using the finite buffer bank $\mathcal{B} = (B_1, \ldots, B_w)$:

$$\text{Err}^{(\mathcal{B})} \leq \sum_{j=1}^{w} \text{Err}^{(\mathcal{I}_j)} \leq \sum_{j=1}^{w} C_j \, \tilde{\pi}^{(\mathcal{I})}_{\partial, B_j} \leq \sum_{j=1}^{w} C_j \, \tilde{\pi}^{(\mathcal{I}_j)}_{\partial, B_j} \leq \sum_{j=1}^{w} C_j \, \tilde{\pi}^{(\mathcal{B})}_{\partial, B_j}, \tag{2.20}$$

where $C_j \equiv \frac{\alpha^{(\infty)}_{B_j-1}/\beta^{(\infty)}_{B_j}}{1-\alpha^{(\infty)}_{B_j-1}/\beta^{(\infty)}_{B_j}}$, $j = 1, \ldots, w$, are finite constants for each individual buffer queue.

2.4.7. Upper and lower bounds for steady state boundary probability. However, the boundary probability $\tilde{\pi}^{(\mathcal{B})}_{\partial, B_j}$ cannot be calculated a priori without solving the dCME. To efficiently estimate if the size of the truncated state space is adequate to compute the steady state probability landscape with errors smaller than a predefined tolerance, we now introduce an easy-to-compute method to obtain the upper- and lower-bounds of the boundary probabilities $\tilde{\pi}^{(N)}_N$ a priori without solving the dCME.

Denote the maximum and minimum aggregated synthesis rates from the block submatrix $\boldsymbol{A}_{i+1,\,i}$ as $\overline{\alpha}^{(N)}_i$ and $\underline{\alpha}^{(N)}_i$, respectively. They can be computed as the maximum and minimum element of the row vector obtained from the column sums:

$$\overline{\alpha}^{(N)}_i = \max\{\mathbb{1}^T \boldsymbol{A}_{i+1,i}\} \quad \text{and} \quad \underline{\alpha}^{(N)}_i = \min\{\mathbb{1}^T \boldsymbol{A}_{i+1,i}\}, \tag{2.21}$$

respectively. The maximum and minimum aggregated degradation rates can be computed similarly from the block submatrix $\boldsymbol{A}_{i,\,i+1}$ as

$$\overline{\beta}^{(N)}_{i+1} = \max\{\mathbb{1}^T \boldsymbol{A}_{i,i+1}\} \quad \text{and} \quad \underline{\beta}^{(N)}_{i+1} = \min\{\mathbb{1}^T \boldsymbol{A}_{i,i+1}\}, \tag{2.22}$$

respectively. Note that $\overline{\alpha}^{(N)}_i$, $\underline{\alpha}^{(N)}_i$, $\overline{\beta}^{(N)}_{i+1}$, and $\underline{\beta}^{(N)}_{i+1}$ can be easily calculated a priori without the need for explicit state enumeration and generation of the partitioned transition rate matrix $\tilde{\boldsymbol{A}}$. The block submatrix $\boldsymbol{A}_{i+1,i}$ only contains synthesis reactions, and $\boldsymbol{A}_{i,i+1}$ only contains degradation reactions. The maximum total copy numbers of reactants are fixed at each aggregated state group when the maximum buffer capacity is specified, and therefore $\overline{\alpha}^{(N)}_i$, $\underline{\alpha}^{(N)}_i$, $\overline{\beta}^{(N)}_{i+1}$, and $\underline{\beta}^{(N)}_{i+1}$ can be easily calculated by examining the maximum and minimum synthesis and degradation reaction rates. As the original $\alpha^{(N)}_i$ and $\beta^{(N)}_{i+1}$ given in (2.12) are weighted sums of vector $\mathbb{1}^T \boldsymbol{A}_{i+1,i}$ and $\mathbb{1}^T \boldsymbol{A}_{i,i+1}$ with regard to the steady state probability distribution $\tilde{\boldsymbol{\pi}}^{(N)}(\mathcal{G}_i)$, respectively, we have

$$\underline{\alpha}^{(N)}_i \leq \alpha^{(N)}_i \leq \overline{\alpha}^{(N)}_i \quad \text{and} \quad \underline{\beta}^{(N)}_{i+1} \leq \beta^{(N)}_{i+1} \leq \overline{\beta}^{(N)}_{i+1}.$$

We use results from the theory of stochastic ordering for comparing Markov processes to bound $\tilde{\pi}_N^{(N)}$. Stochastic ordering "$\leq_{st}$" between two infinitesimal generator matrices $\boldsymbol{P}_{n\times n}$ and $\boldsymbol{Q}_{n\times n}$ of Markov processes is defined as [36, 75]

$$\boldsymbol{P} \leq_{st} \boldsymbol{Q} \quad \text{if and only if } \sum_{k=j}^{n} P_{i,k} \leq \sum_{k=j}^{n} Q_{i,k} \text{ for all } i, j.$$

Stochastic ordering between two vectors are similarly defined as

$$\boldsymbol{p} \leq_{st} \boldsymbol{q} \quad \text{if and only if } \sum_{k=j}^{n} p_k \leq \sum_{k=j}^{n} q_k \text{ for all } j.$$

To derive an upper bound for $\tilde{\pi}_N^{(N)}$ in (2.17), we construct a new matrix $\overline{\boldsymbol{B}}$ by replacing $\alpha_k^{(N)}$ with the corresponding $\overline{\alpha}_k^{(N)}$ and $\beta_{k+1}^{(N)}$ with the corresponding $\underline{\beta}_{k+1}^{(N)}$ in the matrix $\boldsymbol{B}$. Similarly, to derive an lower bound for $\tilde{\pi}_N^{(N)}$, we construct the matrix $\underline{\boldsymbol{B}}$ by replacing $\alpha_k^{(N)}$ with the corresponding $\underline{\alpha}_k^{(N)}$ and replace $\beta_{k+1}^{(N)}$ with $\overline{\beta}_{k+1}^{(N)}$ in $\boldsymbol{B}$. We then have the following stochastic ordering:

$$\underline{\boldsymbol{B}} \leq_{st} \boldsymbol{B} \leq_{st} \overline{\boldsymbol{B}}.$$

All three matrices $\underline{\boldsymbol{B}}$, $\boldsymbol{B}$, and $\overline{\boldsymbol{B}}$ are "$\leq_{st}$ $-$monotone" according to the definitions in Truffet [75]. The steady state probability distributions of matrices $\underline{\boldsymbol{B}}$, $\boldsymbol{B}$, and $\overline{\boldsymbol{B}}$ are denoted as $\boldsymbol{\pi}_{\underline{\boldsymbol{B}}}$, $\boldsymbol{\pi}_{\boldsymbol{B}}$, and $\boldsymbol{\pi}_{\overline{\boldsymbol{B}}}$, respectively. They maintain the same stochastic ordering (Theorem 4.1 of Truffet [75]):

$$\boldsymbol{\pi}_{\underline{\boldsymbol{B}}} \leq_{st} \boldsymbol{\pi}_{\boldsymbol{B}} \leq_{st} \boldsymbol{\pi}_{\overline{\boldsymbol{B}}}.$$

Therefore, we have the inequality for the jth buffer queue with capacity N:

$$\underline{\tilde{\pi}}_N^{(N)} \leq \tilde{\pi}_N^{(N)} \leq \overline{\tilde{\pi}}_N^{(N)}.$$

Here the upper bound $\overline{\tilde{\pi}}_N^{(N)}$ is the boundary probability computed from $\tilde{\boldsymbol{\pi}}_{\overline{\boldsymbol{B}}}$, the lower bound $\underline{\tilde{\pi}}_N^{(N)}$ is the boundary probability computed from $\tilde{\boldsymbol{\pi}}_{\underline{\boldsymbol{B}}}$, and $\tilde{\pi}_N^{(N)}$ is the boundary probability from $\tilde{\boldsymbol{\pi}}_{\boldsymbol{B}}$. From (2.17), the upper bound $\overline{\tilde{\pi}}_N^{(N)}$ can be calculated a priori from reaction rates:

$$\overline{\tilde{\pi}}_N^{(N)} = \frac{\prod_{k=0}^{N-1} \frac{\overline{\alpha}_k^{(N)}}{\underline{\beta}_{k+1}^{(N)}}}{1 + \sum_{j=1}^{N} \prod_{k=0}^{j-1} \frac{\overline{\alpha}_k^{(N)}}{\underline{\beta}_{k+1}^{(N)}}}, \tag{2.23}$$

and the lower bound $\underline{\tilde{\pi}}_N^{(N)}$ can be calculated as

$$\underline{\tilde{\pi}}_N^{(N)} = \frac{\prod_{k=0}^{N-1} \frac{\underline{\alpha}_k^{(N)}}{\overline{\beta}_{k+1}^{(N)}}}{1 + \sum_{j=1}^{N} \prod_{k=0}^{j-1} \frac{\underline{\alpha}_k^{(N)}}{\overline{\beta}_{k+1}^{(N)}}}. \tag{2.24}$$

These are general upper and lower bounds of truncation error valid for any iBD in a reaction network. The upper and lower bounds for the total error of a reaction network with multiple iBDs can be obtained straightforwardly by taking summations of bounds for each individual iBDs:

$$\sum_{i=1}^{w} \underline{\tilde{\pi}}_{B_i}^{(\mathcal{B})} \le \sum_{i=1}^{w} \tilde{\pi}_{B_i}^{(\mathcal{B})} \le \sum_{i=1}^{w} \overline{\tilde{\pi}}_{B_i}^{(\mathcal{B})}. \tag{2.25}$$

In summary, we have shown from (2.10), Facts 1–6, (2.20), and (2.25) that the truncation error of the steady state probability landscape from each individual iBD $\mathrm{Err}^{(B_j)}$ using finite buffer bank $\mathcal{B} = (B_1, \ldots, B_w)$ can be bounded using the inequality

$$\begin{aligned} \mathrm{Err}^{(\mathcal{I}_j)} &\le C_j\, \tilde{\pi}_{\partial, B_j}^{(\mathcal{I})} \le C_j\, \tilde{\pi}_{\partial, B_j}^{(\mathcal{I}_j)} \le C_j\, \tilde{\pi}_{\partial, B_j}^{(\mathcal{B})} \le C_j\, \overline{\tilde{\pi}}_{B_j}^{(\mathcal{B})} \le \overline{C}_j\, \overline{\tilde{\pi}}_{B_j}^{(\mathcal{B})} \\ &= \frac{\frac{\overline{\alpha}_{B_j-1}^{(B_j)}}{\underline{\beta}_{B_j}^{(B_j)}}}{1 - \frac{\overline{\alpha}_{B_j-1}^{(B_j)}}{\underline{\beta}_{B_j}^{(B_j)}}} \cdot \frac{\prod_{k=0}^{B_j-1} \frac{\overline{\alpha}_k^{(B_j)}}{\underline{\beta}_{k+1}^{(B_j)}}}{1 + \sum_{j=1}^{B_j} \prod_{k=0}^{j-1} \frac{\overline{\alpha}_k^{(B_j)}}{\underline{\beta}_{k+1}^{(B_j)}}}, \end{aligned} \tag{2.26}$$

and the overall truncation error $\mathrm{Err}^{(\mathcal{B})}$ using the finite buffer bank $\mathcal{B} = (B_1, \ldots, B_w)$ can therefore be bounded by the following inequality:

$$\mathrm{Err}^{(\mathcal{B})} \le \sum_{j=1}^{w} \mathrm{Err}^{(\mathcal{I}_j)} \le \sum_{j=1}^{w} \overline{C}_j\, \overline{\tilde{\pi}}_{B_j}^{(\mathcal{B})} = \sum_{j=1}^{w} \frac{\frac{\overline{\alpha}_{B_j-1}^{(B_j)}}{\underline{\beta}_{B_j}^{(B_j)}}}{1 - \frac{\overline{\alpha}_{B_j-1}^{(B_j)}}{\underline{\beta}_{B_j}^{(B_j)}}} \cdot \frac{\prod_{k=0}^{B_j-1} \frac{\overline{\alpha}_k^{(B_j)}}{\underline{\beta}_{k+1}^{(B_j)}}}{1 + \sum_{j=1}^{B_j} \prod_{k=0}^{j-1} \frac{\overline{\alpha}_k^{(B_j)}}{\underline{\beta}_{k+1}^{(B_j)}}}, \tag{2.27}$$

where $C_j \equiv \frac{\alpha_{B_j-1}^{(\infty)}/\beta_{B_j}^{(\infty)}}{1-\alpha_{B_j-1}^{(\infty)}/\beta_{B_j}^{(\infty)}}$ and $\overline{C}_j \equiv \frac{\overline{\alpha}_{B_j-1}^{(B_j)}/\underline{\beta}_{B_j}^{(B_j)}}{1-\overline{\alpha}_{B_j-1}^{(B_j)}/\underline{\beta}_{B_j}^{(B_j)}}$, $j = 1, \ldots, w$, are finite constants for each individual buffer queue, and we have $C_j \le \overline{C}_j$ as $\frac{\alpha_{B_j-1}^{(\infty)}}{\beta_{B_j}^{(\infty)}} \le \frac{\overline{\alpha}_{B_j-1}^{(B_j)}}{\underline{\beta}_{B_j}^{(B_j)}}$.

2.5. Optimizing buffer allocation.

2.5.1. Determining minimal buffer sizes satisfying predefined error tolerance. To determine the minimal buffer sizes for the w iBDs so a predefined error tolerance ϵ is satisfied, we first calculate a priori the upper bound from the boundary probability $\overline{\tilde{\pi}}_N^{(N)}$ of each iBD using (2.23) for different buffer sizes. The minimal N for each iBD with $\overline{\tilde{\pi}}_N^{(N)} < \epsilon/w$ is then chosen as the size of that buffer queue. Another weighted scheme is also possible. We then proceed to enumerate the state space using a buffer bank whose sizes have been thus determined a priori to numerically solve the dCME .

It is possible that this a priori upper bound is overly conservative, and buffer sizes can be further decreased based on numerical results. Specifically, if the boundary probability computed from numerical solution for an iBD with an a priori determined buffer size is much smaller than the predefined error tolerance ϵ, it is possible to further decrease the buffer size of that iBD to gain in memory space and improve computing efficiency.

2.5.2. Optimized memory allocation based on error bounds. Our method can also be used to optimize the allocation of memory space to improve the accuracy or computing efficiency of the solution to the dCME. When the total size of the state space is fixed, we can allocate buffer capacities for buffer queues differently, so that the total error of the dCME solution is minimized. A simple strategy is to distribute the errors equally to all buffer queues or according to some weight scheme, for example, based on the error bounds of individual iBDs, the complexity of computing the rates of individual iBDs, or the effects on numerical efficiency. We then determine the buffer size of each iBD. The relative ratio of buffer sizes of different iBDs can be used to allocate memory. When the state space to be enumerated is too large to fit into the computer memory, we can further decrease buffer capacities for all iBDs simultaneously according to the allocation ratio. Such optimization can be done a priori without trial computations.

2.6. Numerical solutions of dCME.

Time-evolving probability landscape. The time-evolving probability landscape derived from a dCME of (2.2) can be expressed in the form of a matrix exponential: $\boldsymbol{p}(t) = e^{\boldsymbol{A}t}\boldsymbol{p}(0)$, where $\boldsymbol{p}(0)$ is the initial probability landscape and $\boldsymbol{A}$ is the transition rate matrix over the enumerated state space $\Omega^{(\mathcal{B})}$. Once $\boldsymbol{p}(0)$ is given, $\boldsymbol{p}(t)$ can be calculated using numerical methods such as the Krylov subspace projection method, e.g., as implemented in the EXPOKIT package of Sidjie [67]. Other numerical techniques can also be applied [40]. All results of the time-evolving probability landscape in this study are computed using the EXPOKIT package.

Steady state probability landscape. The steady state probability landscape $\boldsymbol{\pi}$ is of great general interest. It is governed by the equation $\boldsymbol{A}\boldsymbol{\pi} = 0$ and corresponds to the right eigenvector of the 0 eigenvalue. With the states enumerated by the mb-dCME method, $\boldsymbol{\pi}$ can be computed using numerical techniques such as iterative solvers [9, 47, 62, 70]. In this study, we use the Gauss–Seidel solver to compute all of the steady state probability landscape. To our knowledge, the ACME method and its predecessor are the only known methods for computing the steady state probability landscape for an arbitrary biological reaction network.

First passage time distribution. The first passage time from a specific initial state to a given end state is of great importance in studying rare events. The probability that a network transits from the starting state $\boldsymbol{x}_s$ to the end state $\boldsymbol{x}_e$ within time t is the *first passage time probability* $p(t, \boldsymbol{x}_e|\boldsymbol{x}_s)$.

The distribution of $p(t, \boldsymbol{x}_e|\boldsymbol{x}_s)$ at all possible time intervals and the corresponding cumulative probability distribution $F(t, \boldsymbol{x}_e|\boldsymbol{x}_s)$, namely, the probability distribution that the system transits from $\boldsymbol{x}_s$ to $\boldsymbol{x}_e$ within time t, can be computed using the ACME method. To obtain $F(t, \boldsymbol{x}_e|\boldsymbol{x}_s)$, we use the absorbing matrix $\boldsymbol{A}_{abs}$ instead of the original rate matrix $\boldsymbol{A}$ by simply replacing the end state $\boldsymbol{x}_e$ with an absorbing state [30]. In addition, we assign the initial state $\boldsymbol{x}_i$ with a probability of 1. The time-evolving probability landscape of this absorbing system then can be computed as described earlier. The cumulative first passage probabilities $F(t, \boldsymbol{x}_e|\boldsymbol{x}_s)$ at time t is the marginal probability of the end state $\boldsymbol{x}_e$ at time t [30]. The probability of a rare event can be easily found from the cumulative distribution of first passage time between the appropriate states.

3. Biological examples. Below we describe applications using the ACME method in computing the time-evolving and the steady state probability landscapes of several biological reaction networks. We study the genetic toggle switch, the phage lambda lysogenic-lytic epigenetic switch, and the MAPK cascade reaction network.

TABLE 1
Detailed reactions and rate constants of genetic toggle switch.

$R_1 : GeneX \xrightarrow{k_1} GeneX + X, \quad k_1 = 50\, s^{-1}$	$R_3 : X \xrightarrow{k_3} \emptyset, \quad k_3 = 1\, s^{-1}$
$R_2 : GeneY \xrightarrow{k_2} GeneY + Y, \quad k_2 = 100\, s^{-1}$	$R_4 : Y \xrightarrow{k_4} \emptyset, \quad k_4 = 1\, s^{-1}$
$R_5 : 2X + GeneY \xrightarrow{k_5} BGeneY, \quad k_5 = 1 \times 10^{-5}\, nM^{-2} \cdot s^{-1}$	$R_7 : BGeneY \xrightarrow{k_7} 2X + GeneY, \quad k_7 = 0.1\, s^{-1}$
$R_6 : 2Y + GeneX \xrightarrow{k_6} BGeneX, \quad k_6 = 1 \times 10^{-5}\, nM^{-2} \cdot s^{-1}$	$R_8 : BGeneX \xrightarrow{k_8} 2Y + GeneX, \quad k_8 = 0.1\, s^{-1}$

We first show how minimal buffer capacities required for specific error tolerance can be determined a priori. The time-evolving and the steady state probability landscapes of these networks are then computed. We further generate the distributions of first passage times to study the probabilities of rare transition events. Although these three networks are well known, results reported here are significant, as the full stochasticity and the time-evolving probability landscapes have not been computed by solving the underlying dCME for the latter two networks. Furthermore, estimating rare event probabilities such as short first passage time of transition between different states has been a very challenging problem, even for the relatively simple one-dimensional Schlögl model [18, 28].

3.1. Genetic toggle switch and its six-dimensional probability landscapes. The genetic toggle switch consists of two genes repressing each other through binding of their protein dimeric products on the promoter sites of the other genes. This genetic network has been studied extensively [22, 41, 43, 64]. We follow [9, 64] and study a detailed model of the genetic toggle switch with a more realistic control mechanism of gene regulations. Different from simpler toggle switch models [17, 40, 57, 68], in which gene binding and unbinding reactions are approximated by Hill functions, here detailed negative feedback regulation of gene expressions are modeled explicitly through gene binding and unbinding reactions. Although Hill functions are useful to curve-fit gene regulation models with experimental observations [63], it may be inaccurate to model stochastic networks [42, 63]. It is also difficult to obtain the cooperativity parameters in Hill functions and relate them to the detailed rate constants [42, 63]. Furthermore, a Hill function-based model may not capture important multistability characteristics of the reaction network. The genetic toggle switch studied in this example and other previous studies [9, 64] using detailed reaction network with explicit gene binding and unbinding exhibits four distinct stable states (on/off, off/on, on/on, and off/off) for the $GeneX$ and $GeneY$. However, a similar genetic toggle switch modeled using Hill function exhibits only two stable states (on/off and off/on) [40, 43].

The molecular species, reactions, and their rate constants for the genetic toggle switch are listed in Table 1. Specifically, two genes $GeneX$ and $GeneY$ express protein products X and Y, respectively. Two X/Y protein monomers can bind on the promoter site of $GeneY/GeneX$ to form protein-DNA complexes $BGeneY/BGeneX$ and turn off the expression of $GeneY/GeneX$, respectively.

Number of buffer queues and comparison of state space sizes. According to Algorithm 1, there are two iBDs in this network, namely, iBD_1 with reactions $R1$, $R3$, $R5$ and $R7$, and iBD_2 with $R2$, $R4$, $R6$, and $R8$. Each is assigned a separate buffer queue. Detailed steps of iBD partition for the genetic toggle switch network using Algorithm 1 are illustrated in Figure 4. In this network, reaction $R1$ generates a new molecule X and does not alter the copy number of all other species. Therefore, row X of column $R1$ is 1 (Figure 4(a)), and 0 for all other rows of column $R1$. Reaction

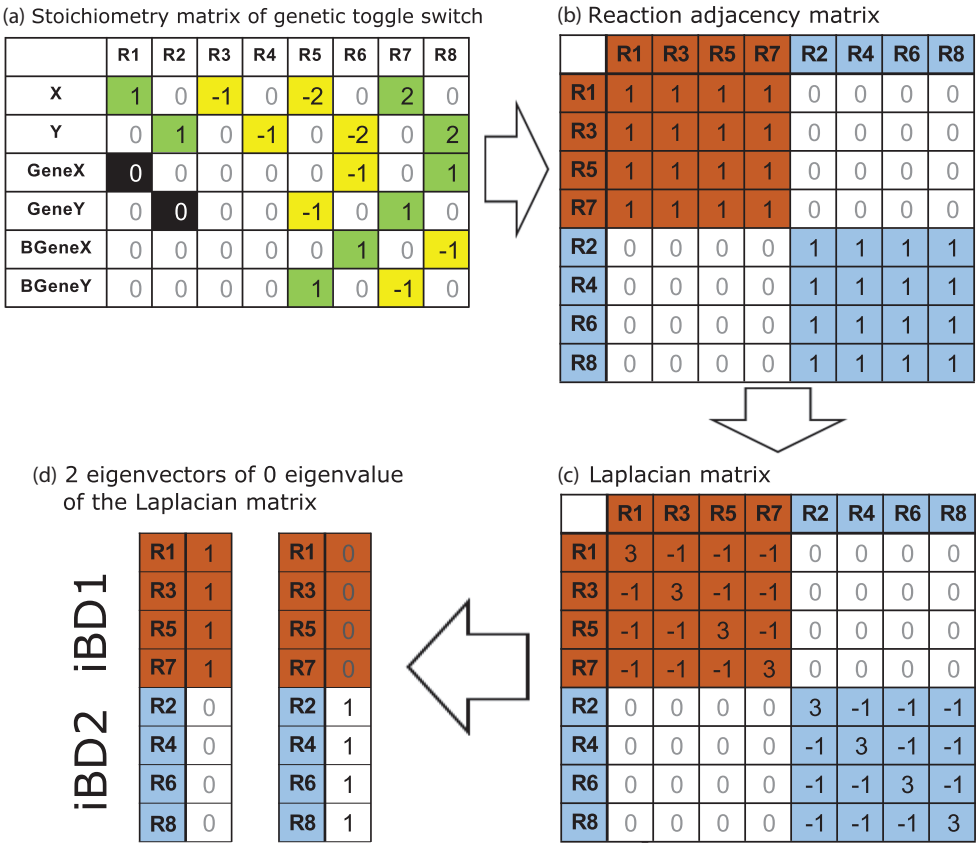

FIG. 4. *Partitioning the bistable genetic toggle switch network into multiple iBD components using Algorithm* 1. (a) *Stoichiometry matrix of the genetic toggle switch constructed from the reaction network in* (1) *in the appendix.* (b) *The reaction adjacency matrix constructed from the stoichiometry matrix according to* (A.1). (c) *The Laplacian matrix of the reaction network constructed using* (A.3). *There are two* 0 *eigenvalues for the Laplacian matrix in* (c). (d) *The two eigenvectors corresponding to the two* 0 *eigenvalues give the partition of the reaction network.*

$R8$ converts one copy of bound gene X ($BGeneX$) into an unbound gene X ($GeneX$) and generates two copies of Y molecules. Therefore, row $BGeneX$ of column $R8$ in the stoichiometry matrix is -1, row $GeneX$ is 1, and row Y is 2. All other rows of column $R8$ are 0s (Figure 4(a)). The remaining column vectors of the stoichiometry matrix for other reactions can be obtained similarly. Each row in the resulting stoichiometry matrix records the stoichiometry of a molecular species participating in all of the reactions. The reaction graph can then be constructed by examining which pairs of reactions share reactant(s) or product(s). Molecular species X changes copy numbers in both reaction $R1$ and $R3$, and therefore we have the edge $e_{R1,R3} = 1$ in the reaction graph $\boldsymbol{G}_R$. We use an adjacency matrix to encode the graph, and the entry for row $R1$ and column $R3$ is therefore 1 (Figure 4(b)). Similarly, $R1$ and $R5$ both involve copy number changes in X, hence $e_{R1,R5} = 1$. As $R1$ and $R7$ also involve copy number changes in X, we have $e_{R1,R7} = 1$. In contrast, as X is the only species that changes copy number in reaction $R1$, and X does not participate either as a reactant or a product with altered copy number in $R2$, $R4$, $R6$, and $R8$, the corresponding entries in the adjacency matrix of $\boldsymbol{G}_R$ therefore have 0s as entries. More

TABLE 2

Size comparison of enumerated state spaces for the genetic toggle switch, the epigenetic switch network of phage lambda, and the MAPK network. Column 1 *lists sizes of buffer queues used in the mb-dCME method, and columns* 2 *and* 3 *sizes of the space enumerated by the dCME and the traditional hypercube methods, respectively. Column* 4 *lists the reduction factors using the mb-dCME method over the hypercube method.*

Sizes of buffer queues	mb-dCME	Hypercube method	Reduction factor
Bistable genetic toggle switch			
10, 10	400	1,936	4.84
20, 20	1,600	7,056	4.41
30, 30	3,600	15,376	4.27
40, 40	6,400	26,896	4.20
Phage lambda epigenetic switch network			
10, 10	2,151	61,952	28.80
20, 20	9,711	225,792	23.25
30, 30	22,671	492,032	21.70
40, 40	41,031	860,672	20.98
MAPK signaling network			
3, 3	2,176	4.3×10^9	2.0×10^6
6, 6	209,304	3.3×10^{13}	1.6×10^8
9, 9	6,210,644	1.0×10^{16}	1.6×10^9
14, 6	2,706,935	1.1×10^{11}	4.1×10^4

generally, if the dot product of the stoichiometry vectors of two reactions R_i and R_j is nonzero, $e_{Ri,\,Rj} = 1$, otherwise the entry is zero. Once the full adjacency matrix for the reaction graph is complete (Figure 4(b)), the Laplacian matrix (Figure 4(c)) can be obtained following (A.3) in the appendix. The number of the eigenvectors of the Laplacian matrix corresponding to the eigenvalue of 0 gives the number of iBDs in the reaction network, and the nonzero entries of each eigenvector gives the membership of the corresponding iBD (Figure 4(d)). In this example of genetic toggle switch, 0 is an eigenvalue of multiplicity of 2 of the Laplacian matrix. The two eigenvectors associated with the eigenvalue of 0 give the two iBDs (Figure 4(d)). Specifically, the reactions with nonzero entries in each eigenvector form the corresponding iBD: iBD_1 consists of reactions $R1$, $R3$, $R5$, and $R7$, and iBD_2 consists of $R2$, $R4$, $R6$, and $R8$ (Figure 4(d)).

The genetic toggle switch is sufficiently complex to exhibit reduced sizes of the enumerated state spaces using the multifinite buffer algorithm, when compared with the traditional hypercube method. Table 2 lists the sizes of the state spaces using these two methods. The size of enumerated state space for the hypercube method is the product of the maximum number of possible states of each individual species. For example, when both buffer queues have a buffer capacity of 40, the state space size is $41^2 \times 2^4 = 26{,}896$, in which $40 + 1 = 41$ is the total number of all possible different copy numbers of protein X and protein Y, and 2^4 is the total different binding and unbinding configurations for each of $GeneX$, $GeneY$, $BGeneX$, and $BGeneY$. The traditional approach generates a state space that is about four times larger than that generated by the mb-dCME method in this case.

Errors and buffer size determinations. The sizes combination of buffer queues $\boldsymbol{B} = (200, 400)$ is found to be sufficient to obtain the exact steady state probability landscape (estimated error $< 10^{-30}$) according to calculations using (2.23). With the exact steady state probability landscape known, true errors calculated using (2.4) for different sizes of the two buffer queues are shown in Figure 5(a) and Figure 5(b) (red dotted lines and circles), both of which decrease monotonically with increasing buffer sizes.

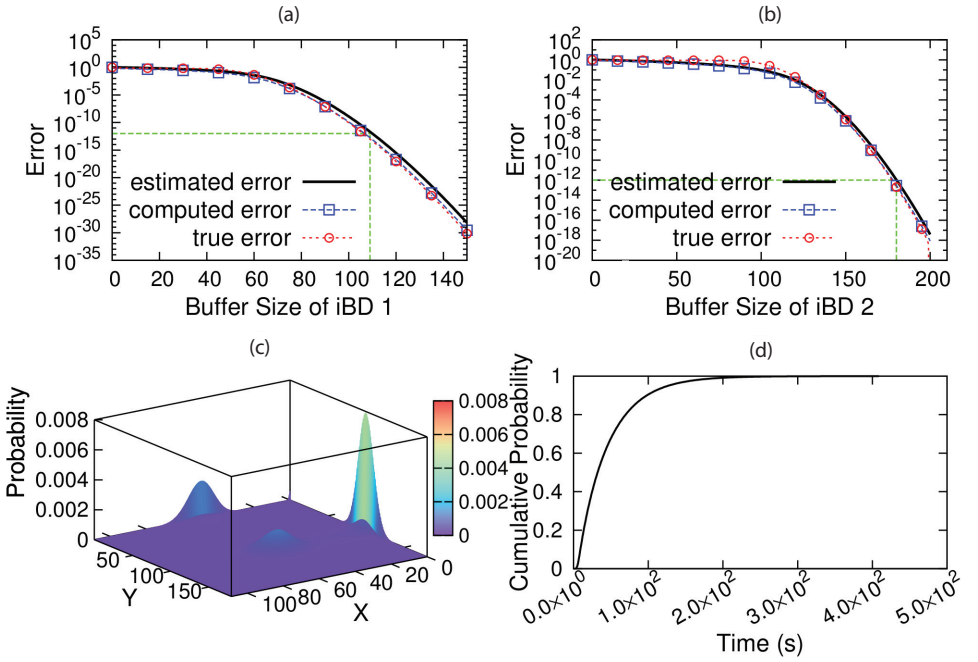

FIG. 5. *Error estimation and computing the steady state probability landscape and the first passage time of the genetic toggle switch network.* (a) *and* (b): *The a priori estimated error (black solid curve), the computed error (blue dashed line and squares), and the true error (red dotted line and circles) of the steady state probability landscape for* iBD_1 *and* iBD_2*, respectively. The a priori estimated error is always larger than the computed error. The green dashed lines indicate the estimated minimal buffer size required so the error is within the predefined tolerance of* 1×10^{-12}. (c): *Steady state probability landscape.* (d): *The cumulative distribution of the first passage time from the initial state* ($\boldsymbol{x}_s = \{X = 49, Y = 0, GeneX = 1, GeneY = 1, BGeneX = 0, BGeneY = 0\}$) *to the end state* ($\boldsymbol{x}_e = \{X = 0, Y = 99\}$).

The computed error estimates by solving the boundary probability from the underlying dCME (Figure 5(a) and (b), blue dashed lines and squares) also decrease monotonically with increasing buffer size. The computed error estimates for the first and second iBD are larger than the true error when the buffer size is larger than 89 and 163, respectively, as would be expected from Fact 3.

To estimate a priori the required minimum buffer sizes for both buffer queues for a predefined error tolerance of $\epsilon = 1.0 \times 10^{-12}$ so that the total error does not exceed 2.0×10^{-12}, we use (2.23) to estimate errors at different buffer sizes (black solid lines in Figure 5(a) and (b)). We follow (2.21) and (2.22) to compute $\overline{\alpha}_i = k_1$ and $\underline{\beta}_{(i+1)} = [(i+1)-2] \cdot k_3$ for the first iBD, where the subscript $(i+1)$ is the total copy number of species X in the system, and the subtraction of 2 is necessary because up to two copies of X can be protected from degradation by binding to $GeneY$. This corresponds to the extreme case when $GeneX$ is constantly turned on and $GeneY$ is constantly turned off. The a priori error estimates at different buffer size are shown in Figure 5(a) (black solid lines). Similarly, we have $\overline{\alpha}_i = k_2$ and $\underline{\beta}_{i+1} = [(i+1)-2] \cdot k_4$ following (2.21) and (2.22) for the second iBD. This corresponds to the other extreme case when the $GeneY$ is constantly turned on, and $GeneX$ is constantly turned off (Figure 5(b), black solid lines). As discussed earlier, the a priori estimated error bounds can be easily computed by examining the maximum and minimum reaction

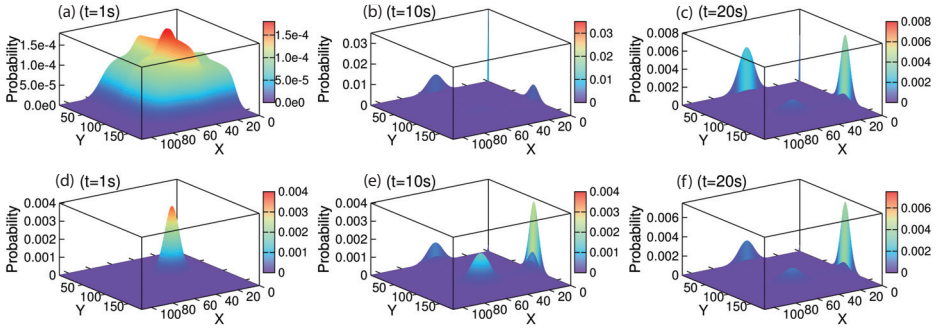

FIG. 6. *The time-evolving probability landscapes of the genetic toggle switch network.* (a), (b), *and* (c): *Probability landscapes at* $t = 1s$, $t = 10s$, *and* $t = 20s$, *starting from the uniform distribution, respectively.* (d), (e), *and* (f): *Probability landscapes at* $t = 1s$, $t = 10s$, *and* $t = 20s$ *starting from the initial distribution with* $p(X = 0, Y = 0, GeneX = 1, GeneY = 1, BGeneX = 0, BGeneY = 0; t = 0) = 1$, *respectively.*

rates. There is no need for the transition rate matrix. For both buffer queues, the a priori estimated errors are conservative and are larger than computed errors at all buffer sizes. They are also larger than the true errors when the buffer sizes are sufficiently large. We can therefore determine that the minimal buffer size to satisfy the predefined error tolerance of $\epsilon = 2.0 \times 10^{-12}$ is 109 for the first iBD (green dashed lines in Figure 5(a)) and 180 for the second iBD (green dashed lines in Figure 5(b)). This combination of buffer sizes $\boldsymbol{B} = (109, 180)$ is used for all subsequent calculations. The enumerated state space has a total of 78,480 states. The $78{,}480 \times 78{,}480$ transition rate matrix is sparse and contains a total of 468,564 nonzero elements.

Steady state and time-evolving probability landscapes. The time-evolving probability landscapes from two different initial conditions are shown in Figure 6. We use a time step $\Delta t = 0.5s$ and a total simulation time of $t = 50s$. The probability landscape in Figure 6(a)–(c) starts from the uniform initial distribution, in which each state takes the same initial probability of 1/78,480. The probability landscape in Figure 6(d)–(f) starts from an initial probability distribution, in which the state ($X = 0$, $Y = 0$, $GeneX = 1$, $GeneY = 1$, $BGeneX = 0$, $BGeneY = 0$) has probability 1 and all other states have probability 0.

The time-evolving probability landscapes for both initial conditions converge to the same steady state (Figure 5(c)) at time $t = 40s$, with the computed error for buffer queues 1 and 2 being 1.741×10^{-13} and 2.881×10^{-13} for results in Figure 6(a)–(c), and 1.716×10^{-13} and 2.898×10^{-13} for results in Figure 6(d)–(f), respectively. Note that the Z-scale is different for the time-evolving probability landscapes. The calculation is completed within two minutes using one single core of a 1GHz Quad-Core AMD CPU.

The steady state probability landscape is also computed separately (Figure 5(c)) for species X and Y). It has four peaks that are centered at $(X = 0, Y = 99)$ with a probability of 7.910×10^{-3}, at $(X = 49, Y = 0)$ with a probability of 2.473×10^{-3}, at $(X = 49, Y = 99)$ with a probability of 1.269×10^{-3}, and at $(X = 0, Y = 0)$ with a probability of 5.909×10^{-4}, respectively. The computed error estimates of 1.715×10^{-13} for the first iBD and 2.899×10^{-13} for the second iBD are both smaller than the predefined error tolerance of $\epsilon = 1.0 \times 10^{-12}$. The computing time is within one minute.

First passage time distribution and rare event probabilities. We study the problem of the first passage time when the system travels from the initial starting state $\boldsymbol{x}_s =$

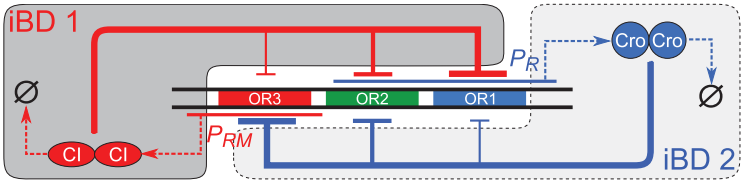

FIG. 7. *The network model of the lysogeny-lysis decision circuit of phage lambda. CI and Cro proteins can repress the expression of each other by differentially binding to three operator sites (*OR1, OR2*, and* OR3*). The network can be partitioned into two iBDs using Algorithm* 1*, as shown in two shaded areas of different color. There are a total of* 11 *molecular species and* 50 *reactions in this network (see Table* 3 *in the appendix).*

$\{X = 49, Y = 0, GeneX = 1, GeneY = 1, BGeneX = 0, BGeneY = 0\}$ to the end state $\boldsymbol{x}_e = \{X = 0, Y = 99\}$. We modified the transition rate matrix by making the end state an absorbing state [10, 30]. The time evolving probability landscape using the absorbing transition rate matrix $\boldsymbol{A}_{abs}$ is then calculated using a time step $\Delta t = 0.5$ for a total of $500s$ simulation time.

When the duration is short, the transition from the initial starting state to the end state is of very low probability. When the first passage time is set to $t \leq 3s$, the probability is calculated to be 1.993×10^{-5}, with a computation time of about 10 seconds. Our method enables accurate and rapid calculations of probabilities of such rare events. As the sampling space of the toggle switch is two-dimensional (X, Y), the rare event probability estimations in this network are far more challenging than the Schlögl model, which was already beyond the original SSA algorithm [23] and a number of biased stochastic simulation algorithms [15, 28, 45, 61]. To our knowledge, no other methods have succeeded in calculating accurately the rare event probabilities in this model of genetic switch.

The computed full cumulative probability distribution of the first passage time is plotted in Figure 5(d). It increases monotonically with time and approaching probability 1. The full calculation is completed within 10 minutes.

3.2. Phage lambda epigenetic switch and its 11-dimensional probability landscapes. The epigenetic switch for lysogenic maintenance and lytic induction in phage lambda is a classic problem in systems biology [58]. The efficiency and stability of the decision circuit of the lysogeny-lysis switch have been studied extensively [5, 6, 7, 83, 84]. Here we use a more realistic model of the reaction network adapted from reference [11]. It consists of 11 molecular species and 50 reactions. The network diagram is shown in Figure 7, and detailed reaction schemes and rate constants are based on previous studies [5, 11, 33, 34, 44, 48, 65] and are listed in Table 3 in the appendix. Molecular species enclosed in parenthesis are required for the specific reactions to occur, but with no changes in stoichiometry. Here $COR(i)$ denotes operator sites OR_i bounded by Cro_2 dimer, $ROR(i)$ for OR_i bounded by CI_2 dimer, $i = 1, 2, 3$.

Number of buffer queues and comparison of state space sizes. There are two iBDs in this network according to Algorithm 1. The first iBD contains all reactions involving CI (the dark gray shaded area in Figure 7), and the second iBD contains all reactions involving Cro (the light gray shaded area in Figure 7). Each iBD is therefore assigned a separate buffer queue.

Table 2 lists the sizes of the state spaces using the mb-dCME method and the traditional hypercube method. As before, the latter is the product of the maximum

TABLE 3
Detailed reactions and rate constants of phage lambda epigenetic switch.

$R_1 : \emptyset + (OR3 + OR2) \overset{s_{CI}^0}{\rightarrow} CI + (OR3 + OR2), s_{CI}^0 = 0.0069/s,$
$R_2 : \emptyset + (OR3 + COR2) \overset{s_{CI}^0}{\rightarrow} CI + (OR3 + COR2), s_{CI}^0 = 0.0069/s,$
$R_3 : \emptyset + (OR3 + ROR2) \overset{s_{CI}^1}{\rightarrow} CI + (OR3 + ROR2), s_{CI}^1 = 0.066/s,$
$R_4 : \emptyset + (OR1 + OR2) \overset{s_{Cro}}{\rightarrow} Cro + (OR1 + OR2), s_{Cro} = 0.0929/s,$
$R_5 : CI \overset{d_{CI}}{\rightarrow} \emptyset, d_{CI} = 0.0027/s,$
$R_6 : Cro \overset{d_{Cro}}{\rightarrow} \emptyset, d_{Cro} = 0.0025/s,$
$R_7 : 2CI + OR1 \overset{b_{CI}}{\rightarrow} ROR1, b_{CI} = 0.0021/nM^2 \cdot s,$
$R_8 : 2CI + OR2 \overset{b_{CI}}{\rightarrow} ROR2, b_{CI} = 0.0021/nM^2 \cdot s,$
$R_9 : 2CI + OR3 \overset{b_{CI}}{\rightarrow} ROR3, b_{CI} = 0.0021/nM^2 \cdot s,$
$R_{10} : 2Cro + OR1 \overset{b_{Cro}}{\rightarrow} COR1, b_{Cro} = 0.01289/nM^2 \cdot s,$
$R_{11} : 2Cro + OR2 \overset{b_{Cro}}{\rightarrow} COR2, b_{Cro} = 0.01289/nM^2 \cdot s,$
$R_{12} : 2Cro + OR3 \overset{b_{Cro}}{\rightarrow} COR3, b_{Cro} = 0.01289/nM^2 \cdot s,$
$R_{13} : ROR1 + (OR2) \overset{u_{ROR1}}{\rightarrow} 2CI + OR1 + (OR2), u_{ROR1} = 0.03998/s,$
$R_{14} : ROR1 + (ROR2 + OR3) \overset{u_{ROR1}^{12}}{\rightarrow} 2CI + OR1 + (ROR2 + OR3), u_{ROR1}^{12} = 0.0005/s,$
$R_{15} : ROR1 + (ROR2 + ROR3) \overset{u_{ROR1}^{123}}{\rightarrow} 2CI + OR1 + (ROR2 + ROR3), u_{ROR1}^{123} = 0.05531/s,$
$R_{16} : ROR1 + (ROR2 + COR3) \overset{u_{ROR1}^{12}}{\rightarrow} 2CI + OR1 + (ROR2 + COR3), u_{ROR1}^{12} = 0.0005/s,$
$R_{17} : ROR1 + (COR2) \overset{u_{ROR1}}{\rightarrow} 2CI + OR1 + (COR2), u_{ROR1} = 0.03998/s,$
$R_{18} : ROR2 + (OR1 + OR3) \overset{u_{ROR2}}{\rightarrow} 2CI + OR2 + (OR1 + OR3), u_{ROR2} = 1.026/s,$
$R_{19} : ROR2 + (ROR1 + OR3) \overset{u_{ROR2}^{12}}{\rightarrow} 2CI + OR2 + (ROR1 + OR3), u_{ROR2}^{12} = 0.01284/s,$
$R_{20} : ROR2 + (OR1 + ROR3) \overset{u_{ROR2}^{23}}{\rightarrow} 2CI + OR2 + (OR1 + ROR3), u_{ROR2}^{23} = 0.00928/s,$
$R_{21} : ROR2 + (ROR1 + ROR3) \overset{u_{ROR2}^{123}}{\rightarrow} 2CI + OR2 + (ROR1 + ROR3), u_{ROR2}^{123} = 0.01284/s,$
$R_{22} : ROR2 + (COR1 + OR3) \overset{u_{ROR2}}{\rightarrow} 2CI + OR2 + (COR1 + OR3), u_{ROR2} = 1.026/s,$
$R_{23} : ROR2 + (OR1 + COR3) \overset{u_{ROR2}}{\rightarrow} 2CI + OR2 + (OR1 + COR3), u_{ROR2} = 1.026/s,$
$R_{24} : ROR2 + (COR1 + COR3) \overset{u_{ROR2}}{\rightarrow} 2CI + OR2 + (COR1 + COR3), u_{ROR2} = 1.026/s,$
$R_{25} : ROR2 + (ROR1 + COR3) \overset{u_{ROR2}^{12}}{\rightarrow} 2CI + OR2 + (ROR1 + COR3), u_{ROR2}^{12} = 0.01284/s,$
$R_{26} : ROR2 + (COR1 + ROR3) \overset{u_{ROR2}^{23}}{\rightarrow} 2CI + OR2 + (COR1 + ROR3), u_{ROR2}^{23} = 0.00928/s,$
$R_{27} : ROR3 + (OR2) \overset{u_{ROR3}}{\rightarrow} 2CI + OR3 + (OR2), u_{ROR3} = 5.19753/s,$
$R_{28} : ROR3 + (ROR2 + OR1) \overset{u_{ROR3}^{23}}{\rightarrow} 2CI + OR3 + (ROR2 + OR1), u_{ROR3}^{23} = 0.04702/s,$
$R_{29} : ROR3 + (ROR2 + ROR1) \overset{u_{ROR3}^{123}}{\rightarrow} 2CI + OR3 + (ROR2 + ROR1), u_{ROR3}^{123} = 5.19753/s,$
$R_{30} : ROR3 + (ROR2 + COR1) \overset{u_{ROR3}^{23}}{\rightarrow} 2CI + OR3 + (ROR2 + COR1), u_{ROR3}^{23} = 0.04702/s,$
$R_{31} : ROR3 + (COR2) \overset{u_{ROR3}}{\rightarrow} 2CI + OR3 + (COR2), u_{ROR3} = 5.19753/s,$
$R_{32} : COR1 + (OR2) \overset{u_{COR1}}{\rightarrow} 2Cro + OR1 + (OR2), u_{COR1} = 0.08999/s,$
$R_{33} : COR1 + (ROR2) \overset{u_{COR1}}{\rightarrow} 2Cro + OR1 + (ROR2), u_{COR1} = 0.08999/s,$
$R_{34} : COR1 + (COR2 + OR3) \overset{u_{COR1}^{12}}{\rightarrow} 2Cro + OR1 + (COR2 + OR3), u_{COR1}^{12} = 0.01776/s,$
$R_{35} : COR1 + (COR2 + ROR3) \overset{u_{COR1}^{12}}{\rightarrow} 2Cro + OR1 + (COR2 + ROR3), u_{COR1}^{12} = 0.01776/s,$
$R_{36} : COR1 + (COR2 + COR3) \overset{u_{COR1}^{123}}{\rightarrow} 2Cro + OR1 + (COR2 + COR3), u_{COR1}^{123} = 0.05531/s,$
$R_{37} : COR2 + (OR1 + OR3) \overset{u_{COR2}}{\rightarrow} 2Cro + OR2 + (OR1 + OR3), u_{COR2} = 0.6306/s,$
$R_{38} : COR2 + (ROR1 + OR3) \overset{u_{COR2}}{\rightarrow} 2Cro + OR2 + (ROR1 + OR3), u_{COR2} = 0.6306/s,$
$R_{39} : COR2 + (OR1 + ROR3) \overset{u_{COR2}}{\rightarrow} 2Cro + OR2 + (OR1 + ROR3), u_{COR2} = 0.6306/s,$
$R_{40} : COR2 + (ROR1 + ROR3) \overset{u_{COR2}}{\rightarrow} 2Cro + OR2 + (ROR1 + ROR3), u_{COR2} = 0.6306/s,$
$R_{41} : COR2 + (COR1 + OR3) \overset{u_{COR2}^{12}}{\rightarrow} 2Cro + OR2 + (COR1 + OR3), u_{COR2}^{12} = 0.12448/s,$
$R_{42} : COR2 + (OR1 + COR3) \overset{u_{COR2}^{23}}{\rightarrow} 2Cro + OR2 + (OR1 + COR3), u_{COR2}^{23} = 0.23822/s,$
$R_{43} : COR2 + (COR1 + COR3) \overset{u_{COR2}^{123}}{\rightarrow} 2Cro + OR2 + (COR1 + COR3), u_{COR2}^{123} = 0.14641/s,$
$R_{44} : COR2 + (ROR1 + COR3) \overset{u_{COR2}^{23}}{\rightarrow} 2Cro + OR2 + (ROR1 + COR3), u_{COR2}^{23} = 0.23822/s,$

TABLE 3
(cont.)

$R_{45}: COR2+(COR1+ROR3) \overset{u^{12}_{COR2}}{\rightarrow} 2Cro+OR2+(COR1+ROR3), u^{12}_{COR2}=0.12448/s,$
$R_{46}: COR3+(OR2) \overset{u_{COR3}}{\rightarrow} 2Cro+OR3+(OR2), u_{COR3}=0.00928/s,$
$R_{47}: COR3+(ROR2) \overset{u_{COR3}}{\rightarrow} 2Cro+OR3+(ROR2), u_{COR3}=0.00928/s,$
$R_{48}: COR3+(COR2+OR1) \overset{u^{23}_{COR3}}{\rightarrow} 2Cro+OR3+(COR2+OR1), u^{23}_{COR3}=0.00351/s,$
$R_{49}: COR3+(COR2+ROR1) \overset{u^{23}_{COR3}}{\rightarrow} 2Cro+OR3+(COR2+ROR1), u^{23}_{COR3}=0.00351/s,$
$R_{50}: COR3+(COR2+COR1) \overset{u^{123}_{COR3}}{\rightarrow} 2Cro+OR3+(COR2+COR1), u^{123}_{COR3}=0.01092/s$

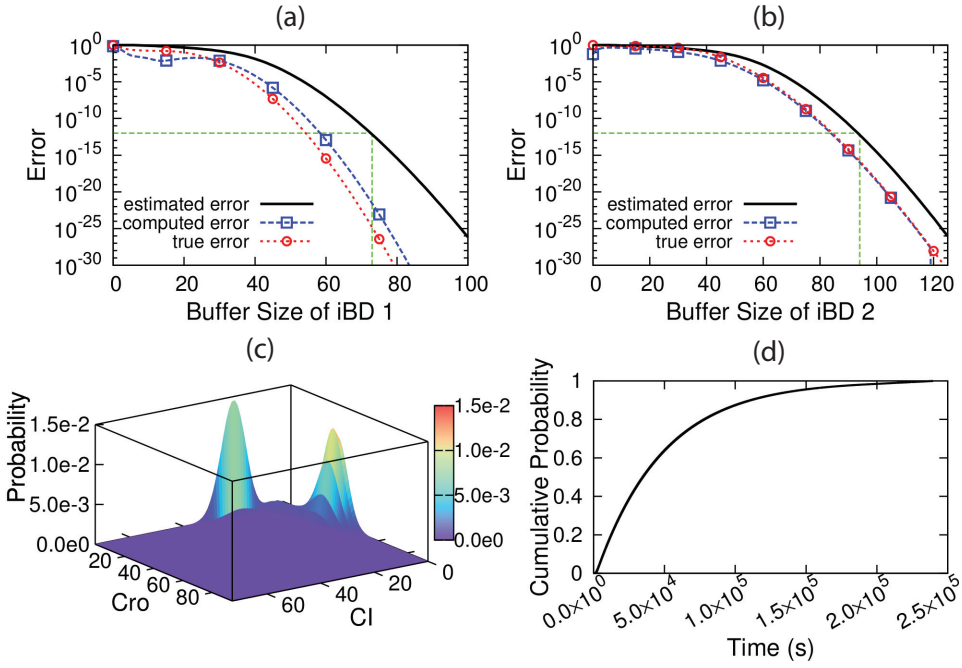

FIG. 8. *Computing the* 11*-dimension steady state probability landscape and the first passage time of the network of epigenetic switch of phage lambda.* (a) *and* (b): *The a priori estimated error (black solid curve), the computed error (blue dashed line and squares), and the true error (red dotted line and circles) of the steady state probability landscape for* iBD_1 *and* iBD_2*, respectively. The computed errors and the true errors are always smaller than the a priori estimated errors. The green dashed lines indicate the estimated minimal buffer sizes required so the error is within the predefined tolerance of* 10^{-12}*.* (c): *The* 11*-dimensional steady state probability landscape projected onto the CI–Pro plane.* (d): *The cumulative distribution of first passage time from the initial state* ($\boldsymbol{x}_s = \{CI = 21, Cro = 0, OR1 = OR2 = OR3 = 0, ROR1 = ROR2 = ROR3 = 0, COR1 = COR2 = COR3 = 0\}$) *to the end state* ($\boldsymbol{x}_e = \{CI = 2, Cro = 33\}$).

number of possible states of each individual species. The size of the state space by the traditional approach is about 21–29 times larger than that by the mb-dCME method.

Errors and buffer size determinations. The size combination of buffer queues of $\boldsymbol{B} = (150, 150)$ is sufficient to obtain the exact steady state probability landscape according to calculations using (2.23) (estimated error $< 10^{-30}$). The true errors calculated using (2.4) for different sizes of two buffer queues are shown in Figure 8(a) and (b) (red dotted lines and circles), both of which decrease monotonically with increasing buffer size.

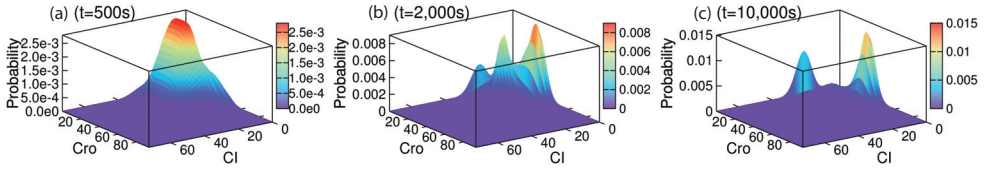

FIG. 9. *Projection of the* 11*-dimensional time-evolving probability landscape of the epigenetic switch of phage lambda projected to the CI–Cro plane starting from the uniform distribution, with the probability landscape* (a) *at* $t = 500s$*;* (b) *at* $t = 2{,}000s$*; and* (c) *at* $t = 10{,}000s$.

The computed error estimates by solving the boundary probability from the underlying dCME (Figure 8(a) and (b), blue dashed lines and squares) also decrease monotonically with increasing buffer size, when buffer sizes are larger than 23 and 6 for the first and second iBD, respectively. The computed error estimates for the first and second iBD are larger than the true error when the buffer size is larger than 28 and 69, respectively, as would be expected from Fact 3.

To estimate a priori the required minimum buffer sizes for a predefined error tolerance of $\epsilon = 1.0 \times 10^{-12}$, we use (2.23) to estimate a priori errors at different buffer sizes (black solid lines in Figure 8(a) and (b)). We follow (2.21) and (2.22) to compute $\overline{\alpha}_i = s^1_{CI}$ and $\underline{\beta}_{(i+1)} = [(i+1) - 6] \cdot d_{CI}$ for the first iBD, where subscript $(i+1)$ is the total copy number of species CI in the system, and the subtraction of 6 is because there can be maximally 6 copies of CI molecules protected from degradation by binding on the three operator sites $OR1$, $OR2$, and $OR3$. This corresponds to the extreme case when CI is constantly synthesized at the maximum rate and degraded at the minimum rate. Similarly, we assign values of $\overline{\alpha}_i = s_{Cro}$ and $\underline{\beta}_{i+1} = [(i+1) - 6] \cdot d_{Cro}$ in (2.21) and (2.22) to calculate the estimated error for the second iBD, which corresponds to the other extreme case when the Cro is constantly synthesized at its maximum rate and degraded at the minimum rate. In both cases, a priori estimated errors are larger than computed errors at all buffer sizes. We can therefore determine conservatively a priori that the minimal buffer size necessary to satisfy the predefined error tolerance of 1.0×10^{-12} is 73 for the first iBD (green straight dashed lines in Figure 8(a)) and 94 for the second iBD (green straight dashed lines in Figure 8(b)). This combination of buffer sizes $\boldsymbol{B} = (73, 94)$ is used for all subsequent calculations. The enumerated state space has a total of 180,756 states. The $180{,}756 \times 180{,}756$ transition rate matrix is sparse and contains a total of 1,330,838 nonzero elements.

Steady state and time-evolving probability landscapes. A projection of the time-evolving 11-dimension probability landscape starting from the uniform initial distribution is shown in Figure 9, in which each state takes the same initial probability of 1/180,756. We use a time step $\Delta t = 5s$ and a total simulation time of $t = 300{,}000s$. The time-evolving probability landscape converges to the steady state (shown separately on Figure 8(c)) at around $t = 250{,}000s$ with the computed error of 1.496×10^{-21} for buffer queue 1 and 2.722×10^{-16} for buffer queue 2. The calculation took 18 hours using one single core of a 1GHz Quad-Core AMD CPU.

The steady state probability landscape is also computed separately. Its projection to the CI–Cro plane is plotted in Figure 8(c), which has two peaks centered at $(X = 21Y = 0)$ with a probability of 1.447×10^{-2} and at $(X = 2, Y = 33)$ with a probability of 1.211×10^{-2}, respectively. The computed error of 1.503×10^{-21} for the first iBD and 2.711×10^{-16} for the second iBD are both significantly smaller than the

predefined error tolerance of $\epsilon = 1.0 \times 10^{-12}$. The computation of the steady state probability landscape is completed within 50 minutes.

First passage time distribution and rare event probabilities. We study the problem of the first passage time when the system travels from the initial state $\boldsymbol{x}_s = \{CI = 21, Cro = 0, OR1 = OR2 = OR3 = 0, ROR1 = ROR2 = ROR3 = 0, COR1 = COR2 = COR3 = 0\}$ in the peak of CI on the $CI-Cro$ plane, to the end state of $\boldsymbol{x}_e = \{CI = 2, Cro = 33\}$, which contains 27 different microstates at the peak of Cro. We modified the transition rate matrix by making these end microstates absorbing [10, 30]. The time evolving probability landscape using the absorbing transition rate matrix $\boldsymbol{A}_{abs}$ is then calculated using a time step $\Delta t = 5$ for a total of 250,000s simulation time.

When the duration is short, the transition from the initial starting state to the end state is of very low probability. When the first passage time is set to $t \leq 500s$, the probability is calculated to be 7.184×10^{-9}, with a computation time of 9 minutes. Similar results would require billions of trajectories when using the alternative method of the stochastic simulation algorithm. Similar to the toggle switch example, this rare event problem is two-dimensional (CI and Cro), and no current methods we are aware of can accurately calculate such rare event probabilities.

The computed full cumulative probability distribution of the first passage time is plotted in Figure 8(d). It increases monotonically with time, and approaching probability 1. That is, given enough time, the system will reach the end state $\boldsymbol{x}_e = \{X = 2, Y = 33\}$ with certainty 1. The full calculation is completed within 25 hours.

3.3. Bistable MAPK signaling cascade and its 16-dimensional probability landscapes. The mitogen-activated protein kinase (MAPK) cascades play critical roles in controlling cell responses to external signals and in regulating cell behavior, including proliferation, migration, differentiation, and polarization [39]. There are multiple levels of signal transduction in a MAPK cascade, where activated kinase at each level phosphorylates the kinase at the next level. The MAP kinase is activated by dual phosphorylations at two conserved threonine (T) and tyrosine (Y) residues. Phosphorylated MAPKs can also be dephosphorylated by specific MAP kinase phosphatases (MKPs). Numerous mathematical models have been developed to study the complex behavior of the MAPK cascade in signal transduction [35, 51, 60, 69, 80].

We examine in detail both the time-evolving and the steady state probability landscapes of a MAPK cascade model consisting of two levels of kinases, namely, the extracellular signal-regulated kinase (ERK) and its kinase MEK. This network model of 16 molecular species is an open network, in which the phosphorylation processes for ERK (reactions R_5 to R_{21} in Table 4) [51], as well as the synthesis and degradation of both ERK and MEK (reactions R_1 to R_4 in Table 4) are modeled in detail. A feedback loop in the network enhances the synthesis of MEK by activating ERKs (Figure 10), leading to bistability [69]. The full network is shown in Figure 11. It includes a total of 16 molecular species and 35 individual reactions. Details of the molecular species are listed in Table 5, and reaction schemes and rate constants are specified in Table 4 in the appendix. We set the copy number of MKP3 to 1 and assume that phosphorylations do not protect the ERK from degradation. To our knowledge, this is the largest network where full stochastic probability landscapes are computed by solving the underlying dCME.

Number of buffer queues and comparison of state space sizes. According to Algorithm 1, there are two iBDs in the network. The first iBD contains all reactions related to the ERK, labeled as K, (reactions 3–21 in Table 4 and species in the lightly

TABLE 4
Detailed reactions and rate constants in MAPK signaling network.

$R_1 : \emptyset \underset{d_2}{\overset{s_2}{\rightleftharpoons}} \text{MEK}, s_2 = 0.001/s, d_2 = 0.15/s,$
$R_2 : \emptyset + (\text{Kpp}) \overset{s_3}{\rightarrow} \text{MEK} + (\text{Kpp}), s_3 = 0.005/s,$
$R_3 : \emptyset \underset{d_1}{\overset{s_1}{\rightleftharpoons}} \text{K}, s_1 = 0.00024/s, d_1 = 0.0001/s,$
$R_4 : \text{KpY} \overset{d_1}{\rightarrow} \emptyset, \text{KpT} \overset{d_1}{\rightarrow} \emptyset, \text{Kpp} \overset{d_1}{\rightarrow} \emptyset, d_1 = 0.0001/s,$
$R_5 : \text{K} + \text{MEK} \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} \text{K\_MEK\_Y}, k_1 = 0.375/nM \cdot s, k_{-1} = 1.0/s,$
$R_6 : \text{K\_MEK\_Y} \overset{k_2}{\rightarrow} \text{KpY} + \text{MEK}, k_2 = 0.06/s,$
$R_7 : \text{KpY} + \text{MEK} \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} \text{KpY\_MEK}, k_3 = 0.375/nM \cdot s, k_{-3} = 1.0/s,$
$R_8 : \text{KpY\_MEK} \overset{k_4}{\rightarrow} \text{Kpp} + \text{MEK}, k_4 = 4.5/s,$
$R_9 : \text{K} + \text{MEK} \underset{k_{-5}}{\overset{k_5}{\rightleftharpoons}} \text{K\_MEK\_T}, k_5 = 0.375/nM \cdot s, k_{-5} = 1.0/s,$
$R_{10} : \text{K\_MEK\_T} \overset{k_6}{\rightarrow} \text{KpT} + \text{MEK}, k_6 = 0.06/s,$
$R_{11} : \text{KpT} + \text{MEK} \underset{k_{-7}}{\overset{k_7}{\rightleftharpoons}} \text{KpT\_MEK}, k_7 = 0.375/nM \cdot s, k_{-7} = 1.0/s,$
$R_{12} : \text{KpT\_MEK} \overset{k_8}{\rightarrow} \text{Kpp} + \text{MEK}, k_8 = 4.5/s,$
$R_{13} : \text{Kpp} + \text{MKP3} \underset{h_{-1}}{\overset{h_1}{\rightleftharpoons}} \text{Kpp\_MKP3}, h_1 = 0.015/nM \cdot s, h_{-1} = 1.0/s,$
$R_{14} : \text{Kpp\_MKP3} \overset{h_2}{\rightarrow} \text{KpT\_MKP3\_Y}, h_2 = 0.032/s,$
$R_{15} : \text{KpT\_MKP3\_Y} \underset{h_{-3}}{\overset{h_3}{\rightleftharpoons}} \text{KpT} + \text{MKP3}, h_3 = 0.31/s, h_{-3} = 0.01/nM \cdot s,$
$R_{16} : \text{KpT} + \text{MKP3} \underset{h_{-4}}{\overset{h_4}{\rightleftharpoons}} \text{KpT\_MKP3\_T}, h_4 = 0.01/nM \cdot s, h_{-4} = 1.0/s,$
$R_{17} : \text{KpT\_MKP3\_T} \overset{h_5}{\rightarrow} \text{K\_MKP3\_T}, h_5 = 0.5/s,$
$R_{18} : \text{K\_MKP3\_T} \underset{h_{-6}}{\overset{h_6}{\rightleftharpoons}} \text{K} + \text{MKP3}, h_6 = 0.086/s, h_{-6} = 0.0011/nM \cdot s,$
$R_{19} : \text{KpY} + \text{MKP3} \underset{h_{-7}}{\overset{h_7}{\rightleftharpoons}} \text{KpY\_MKP3}, h_7 = 0.01/nM \cdot s, h_{-7} = 1.0/s,$
$R_{20} : \text{KpY\_MKP3} \overset{h_8}{\rightarrow} \text{K\_MKP3\_Y}, h_8 = 0.47/s,$
$R_{21} : \text{K\_MKP3\_Y} \underset{h_{-9}}{\overset{h_9}{\rightleftharpoons}} \text{K} + \text{MKP3}, h_9 = 0.14/s, h_{-9} = 0.0018/nM \cdot s.$

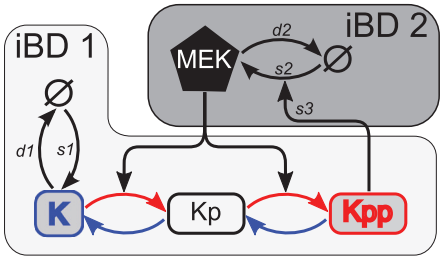

FIG. 10. *A simplified conceptual model of the MAPK network. The MEK and ERK (K) form a positive feedback loop.*

shaded area in Figure 11). The second iBD contains reactions of synthesis and degradation of MEK (reactions 1–2 in Table 4 and species in the darkly shaded box in Figure 11). Each iBD is assigned a separate buffer queue.

To demonstrate the advantage of the mb-dCME state space enumeration method over the traditional hypercube method, Table 2 lists the sizes of the state space with

TABLE 5
Molecular species in the network of bistable MAPK signaling cascade.

Molecular species	Descriptions
MEK	ERK kinase
MKP3	ERK phosphatase
K	ERK, extracellular signal-regulated kinase
KpY	Single phosphorylated ERK on Y residue
KpT	Single phosphorylated ERK on T residue
Kpp	Dual phosphorylated ERK on both Y and T residue
K_MEK_Y	K bound by MEK at residue Y
K_MEK_T	K bound by MEK at residue T
KpY_MEK	KpY bound by MEK
KpT_MEK	KpT bound by MEK
Kpp_MKP3	Kpp associated with MKP3
KpY_MKP3	KpY associated with MKP3
KpT_MKP3_Y	KpT associated with MKP3 at residue Y
KpT_MKP3_T	KpT associated with MKP3 at residue T
K_MKP3_T	K associated with MKP3 at residue T
K_MKP3_Y	K associated with MKP3 at residue Y

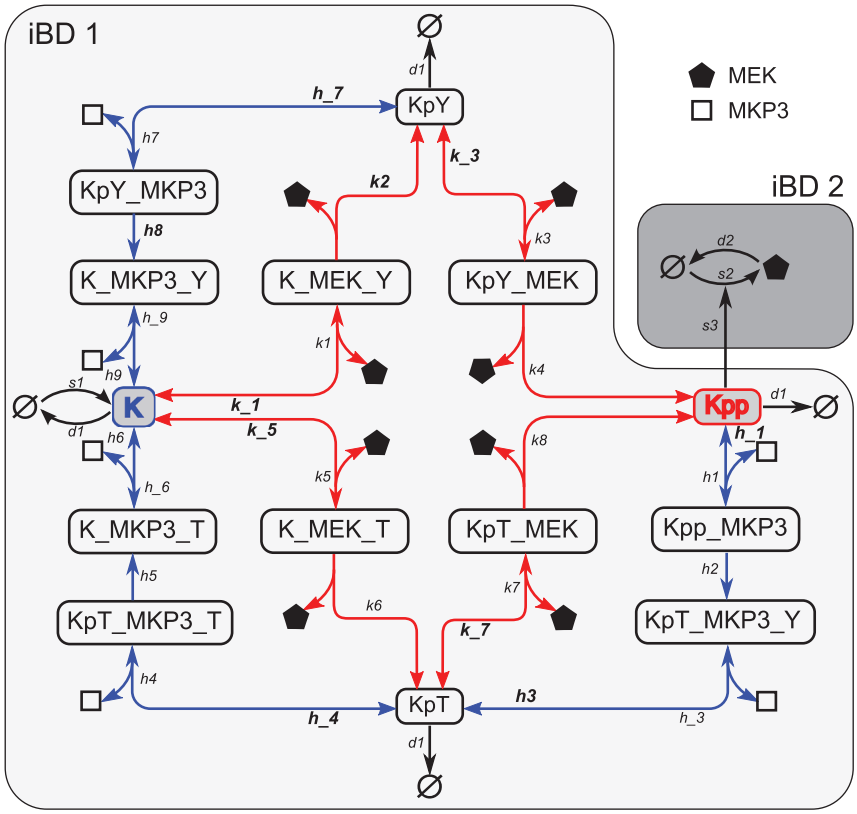

FIG. 11. *A detailed network model of the MAPK cascade. The ERK(K) phosphorylation is catalyzed by the kinase MEK, whereas MEK synthesis is up-regulated by dual phosphorylated ERK(Kpp). Detailed reactions during the dual phosphorylation process of the ERK(K), the synthesis and degradation of MEK, are explicitly modeled. Red and blue arrows represent phosphorylation and dephosphorylation reactions, respectively. Bidirectional arrows represent reversible reactions. The network can be partitioned into two iBDs using Algorithm* 1*, as shown in two shaded areas of different color. There are a total of* 16 *molecular species and* 35 *individual reactions in the network (see Tables* 5 *(in the appendix) and* 4 *for more detail).*

three different choices of the buffer queues. The state spaces generated using the traditional hypercube approach is about 10^4 to 10^9 times larger than that generated by the mb-dCME method. For example, when both buffer queues have a capacity of 9, the size of the enumerated state space using the traditional hypercube method is $(9+1)^{16}$, in which 16 is the number of molecular species. Compared to the size of 6,210,644 using the mb-dCME method, the reduction factor is approximately 1.6×10^9. Without this dramatic reduction, it would not be feasible to compute the exact probability landscape of this model of MAPK cascade network.

Errors and buffer size determinations. The size combination of buffer queues $\boldsymbol{B} = (16, 7)$ is used to approximate the exact solution to the steady state probability landscape (estimated error $\epsilon < 10^{-4}$) according to calculations using (2.23). Although this estimated ϵ is larger than what is used in other models, it is still quite small, as it is the summation of differences in probabilities of the whole state space. This is due to the complexity of this MAPK model and the limitation of the 3GB CUDA memory of the GPU processor we used. Access to a more capable computing facility would allow a different choice of sizes of buffer queues such that a smaller a priori ϵ can be used. Note that the computed errors for the steady state are considerably smaller ($10^{-8} - 10^{-11}$) as described below. With the landscape computed using $\boldsymbol{B} = (16, 7)$ regarded as approximately the true steady state probability landscape, the approximated true errors calculated using (2.4) for different sizes of two buffer queues are shown in Figure 12(a) and 12(b) (red dotted lines and circles), both of which decrease monotonically with increasing buffer sizes.

To estimate a priori the required minimum buffer sizes for both buffer queues for a predefined error tolerance of $\epsilon = 10^{-3}$, we use (2.23) to estimate errors a priori at different buffer size (black solid lines in Figure 12(a) and 12(b)). We follow (2.21) and (2.22) to compute $\overline{\alpha}_i = s_1$ and $\underline{\beta}_{(i+1)} = [(i+1) - 5] \cdot d_1$ for the first iBD. Here the subscript $(i+1)$ is the total copy number of ERK. As an ERK molecule can be protected from degradation by forming as many as five copies of ERK-MKP3 and ERK-MEK complexes in our model (one copy for each of the four species involving "$\_MEK\_$" and one copy for all species involving "$\_MKP3$," Table 2), the actual minimum degradation rates are conservatively calculated to be $[(i+1)-5] \cdot d_1$, where $d_1 = 0.0001$ is the degradation rate of ERK (Table 4). This corresponds to the extreme case when the ERK is constantly synthesized at its maximum rate and degraded at the minimum rate. Similarly, we have $\overline{\alpha}_i = s_3$ and $\underline{\beta}_{i+1} = [(i+1)-4] \cdot d_2$ for (2.21) and (2.22) for the second iBD. As MEK can be protected from degradation by forming as many as four copies of complexes with ERK, the actual minimum degradation rates $\underline{\beta}_{i+1}$ are then conservatively calculated as $[(i+1)-4] \cdot d_2$, where $d_2 = 0.15$ is the degradation rate of MEK (Table 4). This corresponds to the other extreme case when the MEK is constantly synthesized at its maximum rate and degraded at the minimum rate. For both buffer queues, estimated errors are larger than computed errors and true errors at all buffer sizes. We can therefore determine from a priori estimated errors that the minimal buffer size to satisfy the predefined error tolerance 10^{-3} is 14 for the first iBD (green straight dashed lines in Figure 12(a)), and 6 for the second iBD (green straight dashed lines in Figure 12(b)). This combination of buffer sizes $\boldsymbol{B} = (14, 6)$ is used for all subsequent calculations. The enumerated state space has a total of 2,706,935 states. The $2{,}706{,}935 \times 2{,}706{,}935$ transition rate matrix is sparse and contains a total of 36,869,845 nonzero elements.

Steady state and time-evolving probability landscapes. The 16-dimension time-evolving probability landscapes starting from the initial probability distribution with

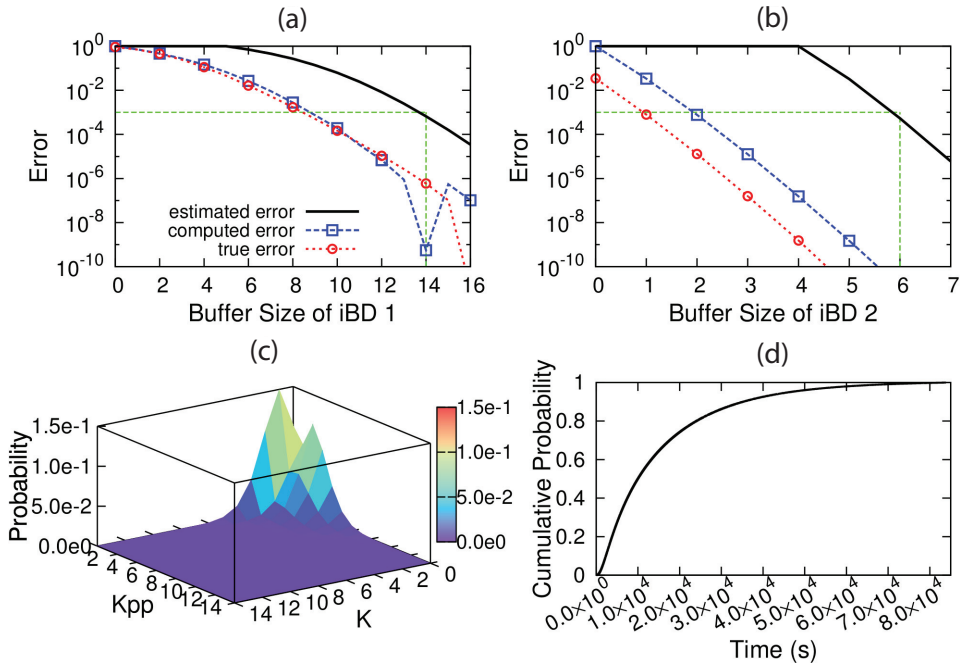

FIG. 12. *Computing the 16-dimension steady state probability landscape and the first passage time of the MAPK cascade network model.* (a) *and* (b): *The a priori estimated error (black solid curve), the computed error (blue dash line and squares), and the true error (red dotted line and circles) of the steady state probability landscape for* iBD_1 *and* iBD_2, *respectively. The computed error is significantly smaller than the a priori estimated error. The green straight dashed lines indicate the estimated minimal buffer size required so the error is within the predefined tolerance of* 10^{-3}. (c): *The steady state probability landscape projected to the* K–K_{pp} *plane.* (d): *The cumulative distribution of first passage time from the initial state* $(K = 3, MKP3 = 1)$ *to the end state* $(Kpp = 2, MKP3 = 1)$.

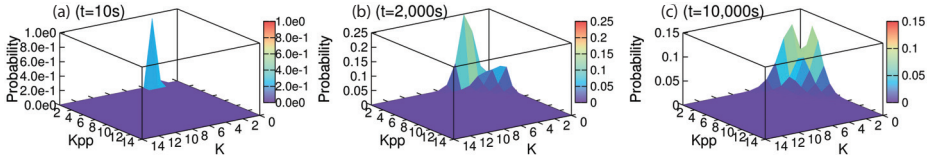

FIG. 13. *The projected time evolving 16-dimension probability landscape of the MAPK cascade reaction network starting from the initial probability distribution with* $p(K = 3, MKP3 = 1, AllOther = 0) = 1$. (a): *The probability landscape projected to the* K–K_{pp} *plane at* $t = 10s$. (b): *Projected probability landscape at* $t = 2{,}000s$. (c): *Projected probability landscape at* $t = 10{,}000s$.

$p(K = 3, MKP3 = 1, \text{others} = 0) = 1$ are shown in Figure 13. We use a time step $\Delta t = 10s$ and a total simulation time of $t = 30{,}000s$. The time-evolving probability landscape converges to the steady state (Figure 12(c)) at about $t = 80{,}000s$. The calculation took 160 minutes using a GPU workstation with an nVidia GeForce GTX 580 card (3GB CUDA memory) [50].

The steady state probability landscape is also solved separately (Figure 12(c), projected onto the K-Kpp plane). It has two peaks centered at $(K = 1, Kpp = 0)$ with the probability of 0.1495, and $(K = 0, Kpp = 2)$ with probability 0.1133, respectively. The computed errors of 3.447×10^{-8} for the first iBD and 1.335×10^{-11} for the second

iBD are both significantly smaller than the predefined error tolerance of $\epsilon = 10^{-3}$. The computation is completed within 50 minute using the same GPU workstation.

First passage time distribution and rare event probabilities. We study the problem of first passage time when the system travels from an initial start state of $\boldsymbol{x}_s = \{K = 3, MPK3 = 1\}$, with all other species 0 copies, to an end state of $\boldsymbol{x}_e = \{Kpp = 2, MPK3 = 1\}$, with all other species 0 copies. We modified the transition rate matrix by making the end state an absorbing state [10, 30]. The time evolving probability landscape using the absorbing transition rate matrix $\boldsymbol{A}_{abs}$ is then calculated using a time step $\Delta t = 1s$ for a total of 85,000s simulation time.

When the duration is short, the transition from the initial starting state to the end state is of very low probability. When the first passage time is set to $t \leq 10s$, the probability is calculated to be 6.047×10^{-9}, with a computation time of about 22 seconds. Similar results would require billions of trajectories when using the alternative method of the stochastic simulation algorithm. Like the toggle switch model, this rare event problem is two-dimensional (K, Kpp) and no current methods we are aware of can accurately calculate such rare event probabilities.

The computed full cumulative probability distribution of the first passage time is plotted in Figure 12(d). It increases monotonically with time and approaching probability 1. That is, given enough time, the system will reach the end state $\boldsymbol{x}_e = \{Kpp = 2, MPK3 = 1\}$ with certainty 1. The full calculation is completed within 41 hours.

4. Discussions and conclusions. Direct solution to the dCME is of fundamental importance. Because the dCME plays the role in system biology analogous to that of the Schrödinger equation in quantum mechanics [8], developing methods for solving the dCME has important implications, just as developing techniques for solving the Schrödinger equation for systems with many atoms does.

Without the truncation of higher order expansions of the discrete jump operator and without assumptions of lower order noise as in the chemical Langevin and the Fokker–Planck equations, accurate direct computation of the time-evolving as well as the steady state probability landscapes allows the stochastic properties of a biological network to be fully characterized. The overall stochastic behavior of a network, including the presence or absence of multistabilities, the often small probabilities of transitions between states, as well as the overall dynamic behavior of the network can all be fully assessed.

A key challenge to obtain direct solution to the dCME is the obstacle of the enormous discrete state space. The conventional hypercube method for state enumeration is easy to implement but rapidly becomes intractable when the network architecture is nontrivial. In this study, we develop the ACME algorithm using multibuffers for directly solving the dCME. By decomposing the reaction network into independent components of birth-death processes, multiple buffer queues for these components are employed for more effective state enumeration. With orders of magnitude reduction in the size of the enumerated state space, our algorithm enables an accurate solution of the dCME for a large class of problems, whose solutions were previously unobtainable. As the network inside each birth-death component becomes more complex, significant reduction can be achieved. For example, computational studies of the MAPK network shows that a reduction factor of 6–9 orders (e.g., from 1.0×10^{16} to 6.2×10^{6}) can be achieved, allowing a stochastic problem otherwise unsolvable to be computed on a desktop computer.

As truncation of the state space will eventually occur for systems of a given fixed finite buffer capacity with fast synthesis reactions, it is essential to quantify the trun-

cation error and to establish a conservative upper bound of the error, so one can assess whether the computed results are within a predefined error tolerance and are therefore trustworthy. This critically important task is made possible through theoretical analysis of the boundary states and their associated steady state probability, via the construction of an aggregated continuous-time Markov process based on factoring of the state space by the buffer queue usage. With explicit formulae for calculating conservative error bounds for the steady state, one can easily calculate error bounds a priori for a finite state space associated with a given buffer capacity. One can also determine the minimal buffer capacity required if a predefined error tolerance is to be satisfied. This eliminates the need for multiple iterations of costly trial computations to solve the dCME for determining the appropriate buffer capacity necessary to ensure small truncation errors. Furthermore, for a given fixed memory, we can also strategically allocate the memory to different buffer queues so the overall error is minimized, or computing efficiency is optimized.

The analysis of the truncation error also enables accurate computation of the steady state probability landscape of a stochastic network. This differs significantly from the finite state projection (FSP) method, which was developed to compute the transient time evolving probability landscape [55, 56]. The FSP method treats all boundary states effectively as one absorbing state, which will eventually trap all probability mass, resulting in a truncation error that can increase to 1 as time proceeds. The error certificate in the FSP method is used for bounding this leaked probability mass and requires trial solutions to the dCME, which can be costly. This error certificate therefore may be unsuitable for studying the long-time behavior or the steady state of the probability landscape; as time proceeds it approaches to 1.0 and becomes uninformative [55, 56]. In contrast, no absorbing states are introduced in the mb-dCME method; the error bound is based on analysis of the probability mass on the boundary states. To our knowledge, the ACME method is among the first general methods that can directly compute the steady state probabilistic landscape of stochastic networks.

We have also provided computational results of three well-known stochastic networks, namely, the toggle switch, the phage lambda epigenetic circuit, and the MAPK cascade. They are bi- or multi-stable networks. Both the time-evolving and the steady state probability landscapes are computed, all with error less than a predefined threshold. Many biologically critical but rare events, such as the spontaneous induction of latent lysogeny of phage lambda provirus into lysis [6, 11, 49], or the cancerogenesis of a normal cell [31], can in principle be formulated as a problem of estimating the distribution of the first-passage time. The ACME method can be used to directly compute the exact probability of rare events in a stochastic model occurring in an arbitrary time interval. This has been demonstrated in all three examples. Our method can provide solutions to this challenging problem that various forms of specifically designed stochastic simulation algorithms have difficulties to resolve [2, 10, 15, 61].

In this study, we use the EXPOKIT, a Krylov subspace projection method [67], to compute all time-evolving probability landscapes. Exploiting the special structure of the state space, recent development in methods of tensor train decomposition offers another attractive approach to compute the probability landscape by decomposing the dCME transition rate matrix into multiplication of smaller tensors [40]. It would be interesting to explore how this technique can be applied to a state space enumerated by the mb-dCME method. Although the ACME method dramatically reduces the state space and can quantify the truncation error asymptotically, it can still fail when the biological network in question is so large that a reduction factor of $O(n!)$ is insufficient.

In addition, the a priori error estimate may not be tight for some complex networks. Further improvements and developments will be the focus of future studies.

Since we have a quantitative estimation of the truncation error, we can be sure that all major probability peaks are contained in the computed solution of probability landscapes when the estimated errors are sufficiently small, as are the cases for the three examples given here. Overall, the goal of this study is to provide a methodology for high precision solutions to the dCME that can be applied to a large class of problems. Important unknown features such as basins, attractors, and transitions for many biological networks can be uncovered and analyzed and their biological significance assessed. It is now possible to analyze details of the topological, topographical, as well as dynamic properties of the probability landscape for a large number of biologically important stochastic networks that are previously not amenable to computational investigations.

Appendix.

Graph of reaction network, its adjacency and Laplacian matrices. $\boldsymbol{G}_R$ can be represented by an $m \times m$ adjacency matrix $\boldsymbol{C}$, where

$$\boldsymbol{C}^{m\times m} = ||\boldsymbol{C}_{i,\,j}|| = \begin{cases} 1 & \text{if } e_{ij} \text{ exists,} \\ 0 & \text{otherwise.} \end{cases} \tag{A.1}$$

The diagonal degree matrix $\boldsymbol{D}$ of the graph $\boldsymbol{G}_R$ is

$$\boldsymbol{D}^{m\times m} = ||\boldsymbol{D}_{i,\,j}|| = \begin{cases} \sum_{k=1}^{m} \boldsymbol{C}_{i,k} & \text{if } i = j, \\ 0 & \text{if } i \neq j, \end{cases} \tag{A.2}$$

where each diagonal element $\boldsymbol{D}_{i,i}$ is the vertex degree of the corresponding reaction R_i. The Laplacian matrix $\boldsymbol{L}$ of the graph $\boldsymbol{G}_R$ can be then written as [54]

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{C}. \tag{A.3}$$

Acknowledgments. We thank Dr. Ao Ma for helpful comments and Alan Perez-Rathke for helpful discussions. YC thanks Dr. Alan Perelson for helpful comments.

REFERENCES

[1] R.J. Allen, P.B. Warren, and P.R. Ten Wolde, *Sampling rare switching events in biochemical networks*, Phys. Rev. Lett., 94 (2005), 18104.
[2] R.J. Allen, C. Valeriani, and P.R. ten Wolde, *Forward flux sampling for rare event simulations*, J. Phys., 21 (2009), 463102.
[3] P. Ao, *Laws in Darwinian evolutionary theory*, Phys. Life Rev., 2 (2005), pp. 117–156.
[4] P. Ao, D. Galas, L. Hood, L. Yin, and X.M. Zhu, *Towards predictive stochastic dynamical modeling of cancer genesis and progression*, Interdiscip. Sci. Comput. Life Sci., 2 (2010), pp. 140–144.
[5] A. Arkin, J. Ross, and H.H. McAdams, *Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected Escherichia coli cells*, Genetics, 149 (1998), pp. 1633–1648.
[6] E. Aurell, S. Brown, J. Johanson, and K. Sneppen, *Stability puzzles in phage λ*, Phys. Rev. E, 65 (2002), 051914.
[7] E. Aurell and K. Sneppen, *Epigenetics as a first exit problem*, Phys. Rev. Lett., 88 (2002), 048101.
[8] D.A. Beard and H. Qian, *Chemical Biophysics: Quantitative Analysis of Cellular Systems*, Cambridge University Press, Cambridge, 2008.
[9] Y. Cao and J. Liang, *Optimal enumeration of state space of finitely buffered stochastic molecular networks and exact computation of steady state landscape probability*, BMC Syst. Biol., 2 (2008), 30.

[10] Y. Cao and J. Liang, *Adaptively biased sequential importance sampling for rare events in reaction networks with comparison to exact solutions from finite buffer dCME method*, J. Chem. Phys., 139 (2013), 025101.
[11] Y. Cao, H.-M. Lu, and J. Liang, *Probability landscape of heritable and robust epigenetic state of lysogeny in phage lambda*, Proc. Natl. Acad. Sci. USA, 107 (2010), pp. 18445–18450.
[12] Y. Cao, A. Terebus, and J. Liang, *State space truncation with quantified errors for accurate solutions to discrete chemical master equation*, Bull. Math. Biol., 78 (2016), pp. 617–661.
[13] F.R.K. Chung, *Spectral Graph Theory*, Vol. 92, AMS, Providence, RI, 1997.
[14] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, et al., *Introduction to Algorithms*, Vol. 2, MIT Press, Cambridge, MA, 2001.
[15] B.J. Daigle, M.K. Roh, D.T. Gillespie, and L.R. Petzold, *Automated estimation of rare event probabilities in biochemical systems*, J. Chem. Phys., 134 (2011), 044110.
[16] I.G. Darvey, B.W. Ninham, and P.J. Staff, *Stochastic models for second order chemical reaction kinetics. The equilibrium state*, J. Chem. Phys., 45 (1966), pp. 2145–2155.
[17] P. Deuflhard, W. Huisinga, T. Jahnke, and M. Wulkow, *Adaptive discrete Galerkin methods applied to the chemical master equation*, SIAM J. Sci. Comput., 30 (2008), pp. 2990–3011.
[18] R.M. Donovan, A.J. Sedgewick, J.R. Faeder, and D.M. Zuckerman, *Efficient stochastic simulation of chemical kinetics networks using a weighted ensemble of trajectories*, J. Chem. Phys., 139 (2013), 115105.
[19] M.B. Elowitz, A.J. Levine, E.D. Siggia, and P.S. Swain, *Stochastic gene expression in a single cell*, Science, 297 (2002), pp. 1183–1186.
[20] B.L. Fox and P.W. Glynn, *Computing Poisson probabilities*, Commun. ACM, 31 (1988), pp. 440–445.
[21] C.W. Gardiner, *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences*, Springer, New York, 2004.
[22] T.S. Gardner, C.R. Cantor, and J.J. Collins, *Construction of a genetic toggle switch in Escherichia coli*, Nature, 403 (2000), pp. 339–342.
[23] D.T. Gillespie, *Exact stochastic simulation of coupled chemical reactions*, J. Phys. Chem., 81 (1977), pp. 2340–2361.
[24] D.T. Gillespie, *A rigorous derivation of the chemical master equation*, Phys. A, 188 (1992), pp. 404–425.
[25] D.T. Gillespie, *The chemical Langevin equation*, J. Chem. Phys., 113 (2000), pp. 297–306.
[26] D.T. Gillespie, *The chemical Langevin and Fokker–Planck equations for the reversible isomerization reaction*, J. Phys. Chem. A, 106 (2002), pp. 5063–5071.
[27] D.T. Gillespie, *A diffusional bimolecular propensity function*, J. Chem. Phys., 131 (2009), 164109.
[28] D.T. Gillespie, M. Roh, and L.R. Petzold, *Refining the weighted stochastic simulation algorithm*, J. Chem. Phys., 130 (2009), 174103.
[29] R. Grima, P. Thomas, and A.V. Straube, *How accurate are the nonlinear chemical Fokker–Planck and chemical Langevin equations?*, J. Chem. Phys., 135 (2011), 084103.
[30] D. Gross and D.R. Miller, *The randomization technique as a modeling tool and solution procedure for transient Markov processes*, Oper. Res., 32 (1984), pp. 343–361.
[31] D. Hanahan and R.A. Weinberg, *The hallmarks of cancer*, Cell, 100 (2000), pp. 57–70.
[32] E.L. Haseltine and J.B. Rawlings, *Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics*, J. Chem. Phys., 117 (2002), pp. 6959–6969.
[33] D. Hawley and W. McClure, *In vitro comparison of initiation properties of bacteriophage lambda wild-type PR and x3 mutant promoters*, Proc. Natl. Acad. Sci. USA, 77 (1980), pp. 6381–6385.
[34] D. Hawley and W. McClure, *Mechanism of activation of transcription initiation from the lambda PRM promoter*, J. Molecular Biol., 157 (1982), pp. 493–525.
[35] C.-Y. Huang and J.E. Ferrell, *Ultrasensitivity in the mitogen-activated protein kinase cascade*, Proc. Natl. Acad. Sci. USA, 93 (1996), pp. 10078–10083.
[36] A. Irle, *Stochastic ordering for continuous-time processes*, J. Appl. Probab., (2003), pp. 361–375.
[37] T. Jahnke, *On reduced models for the chemical master equation*, Multiscale Model. Simul., 9 (2011), pp. 1646–1676.
[38] S. Jiao, Y. Wang, B. Yuan, and P. Ao, *Kinetics of Muller's ratchet from adaptive landscape viewpoint*, in Proceedings of the IEEE International Conference on Systems Biology (ISB), 2011, IEEE, 2011, pp. 27–32.
[39] G.L. Johnson and R. Lapadat, *Mitogen-activated protein kinase pathways mediated by erk, jnk, and p38 protein kinases*, Science, 298 (2002), pp. 1911–1912.

[40] V. Kazeev, M. Khammash, M. Nip, and C. Schwab, *Direct solution of the chemical master equation using quantized tensor trains*, PLoS Comput. Biol., 10 (2014), e1003359.
[41] T.B. Kepler and T.C. Elston, *Stochasticity in transcriptional regulation: Origins, consequences, and mathematical representations*, Biophys. J., 81 (2001), pp. 3116–3136.
[42] H. Kim and E. Gelenbe, *Stochastic gene expression modeling with Hill function for switch-like gene responses*, IEEE/ACM Trans. Comput. Biol. Bioinform., 9 (2012), pp. 973–979.
[43] K.-Y. Kim and J. Wang, *Potential energy landscape and robustness of a gene regulatory network: Toggle switch*, PLoS Comput. Biol., 3 (2007), e60.
[44] C. Kuttler and J. Niehren, *Gene regulation in the pi calculus: Simulating cooperativity at the lambda switch*, Trans. Comput. Syst. Biol. VII, 4230 (2006), pp. 24–55.
[45] H. Kuwahara and I. Mura, *An efficient and exact stochastic simulation method to analyze rare events in biochemical systems*, J. Chem. Phys., 129 (2008), 165101.
[46] I.J. Laurenzi, *An analytical solution of the stochastic master equation for reversible bimolecular reaction kinetics*, J. Chem. Phys., 113 (2000), pp. 3315–3322.
[47] R. Lehoucq, D. Sorensen, and C. Yang, *Arpack Users' Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Philadelphia, 1998.
[48] M. Li, W. McClure, and M. Susskind, *Changing the mechanism of transcriptional activation by phage lambda repressor*, Proc. Natl. Acad. Sci. USA, 94 (1997), pp. 3691–3696.
[49] J.W. Little, D.P. Shepley, and D.W. Wert, *Robustness of a gene regulatory circuit*, EMBO Journal, 18 (1999), pp. 4299–4307.
[50] M. Maggioni, T. Berger-Wolf, and J. Liang, *GPU-based steady-state solution of the chemical master equation*, in Proceedings of the 27th International, Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), IEEE, 2013, pp. 579–588.
[51] N.I. Markevich, J.B. Hoek, and B.N. Kholodenko, *Signaling switches and bistability arising from multisite phosphorylation in protein kinase cascades*, J. Cell Biol., 164 (2004), pp. 353–359.
[52] H.H. McAdams and A. Arkin, *It's a noisy business! Genetic regulation at the nanomolar scale*, Trends Genet., 15 (1999), pp. 65–69.
[53] D.A. McQuarrie, *Stochastic approach to chemical kinetics*, J. Appl. Probab., 4 (1967), pp. 413–478.
[54] R. Merris, *Laplacian matrices of graphs: A survey*, Linear Algebra Appl., 197 (1994), pp. 143–176.
[55] B. Munsky and M. Khammash, *The finite state projection algorithm for the solution of the chemical master equation*, J. Chem. Phys., 124 (2006), 044104.
[56] B. Munsky and M. Khammash, *A multiple time interval finite state projection algorithm for the solution to the chemical master equation*, J. Comput. Phys., 226 (2007), pp. 818–835.
[57] B. Munsky and M. Khammash, *The finite state projection approach for the analysis of stochastic noise in gene networks*, IEEE Trans. Automat. Control, 53 (2008), pp. 201–214.
[58] M. Ptashne, *A Genetic Switch: Phage Lambda Revisited*, 3rd ed., Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY, 2004.
[59] H. Qian, *Cooperativity in cellular biochemical processes: Noise-enhanced sensitivity, fluctuating enzyme, bistability with nonlinear feedback, and other mechanisms for sigmoidal responses*, Ann. Rev. Biophys., 41 (2012), pp. 179–204.
[60] L. Qiao, R.B. Nachbar, I.G. Kevrekidis, and S.Y. Shvartsman, *Bistability and oscillations in the Huang–Ferrell model of MAPK signaling*, PLoS Comput. Biol., 3 (2007), e184.
[61] M.K. Roh, B.J. Daigle, D.T. Gillespie, and L.R. Petzold, *State-dependent doubly weighted stochastic simulation algorithm for automatic characterization of stochastic biochemical rare events*, J. Chem. Phys., 135 (2011), 234108.
[62] Y. Saad and M.H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
[63] M. Santillán, *On the use of the Hill functions in mathematical models of gene regulatory networks*, Math. Model. Nat. Phenom., 3 (2008), pp. 85–97.
[64] D. Schultz, J.N. Onuchic, and P.G. Wolynes, *Understanding stochastic simulations of the smallest genetic networks*, J. Chem. Phys., 126 (2007), 245102.
[65] M.A. Shea and G.K. Ackers, *The OR control system of bacteriophage lambda a physical-chemical model for gene regulation*, J. Mol. Biol., 181 (1985), pp. 211–230.
[66] J. Shi, T. Chen, R. Yuan, B. Yuan, and P. Ao, *Relation of a new interpretation of stochastic differential equations to Ito process*, J. Stat. Phys., 148 (2012), pp. 579–590.
[67] R.B. Sidje, *Expokit: A software package for computing matrix exponentials*, ACM Trans. Math. Software, 24 (1998), pp. 130–156.
[68] P. Sjöberg, P. Lötstedt, and J. Elf, *Fokker–Planck approximation of the master equation in molecular biology*, Comput. Vis. Sci., 12 (2009), pp. 37–50.

[69] P. Smolen, D.A. Baxter, and J.H. Byrne, *Bistable MAP kinase activity: A plausible mechanism contributing to maintenance of late long-term potentiation*, American J. Phys. Cell Phys., 294 (2008), pp. C503–C515.
[70] W.J. Stewart, *Introduction to the Numerical Solution of Markov Chains*, Princeton University Press, Princeton, NJ, 1994.
[71] J. Stewart-Ornstein and H. El-Samad, *Stochastic modeling of cellular networks*, Comput. Methods Cell Biol., 110 (2012), pp. 111–137.
[72] P.S. Swain, M.B. Elowitz, and E.D. Siggia, *Intrinsic and extrinsic contributions to stochasticity in gene expression*, Proc. Natl. Acad. Sci. USA, 99 (2002), pp. 12795–12800.
[73] H.M. Taylor and S. Karlin, *An Introduction to Stochastic Modeling*, 3rd ed., Academic Press, New York, 1998.
[74] P. Thomas, H. Matuschek, and R. Grima, *How reliable is the linear noise approximation of gene regulatory networks?*, BMC Genomics, 14 (2013), S5.
[75] L. Truffet, *Near complete decomposability: Bounding the error by a stochastic comparison method*, Adv. Appl. Probab., 29 (1997), pp. 830–855.
[76] N.G. Van Kampen, *Stochastic processes in physics and chemistry*, 3rd ed., Elsevier Science and Technology Books, New York, 2007.
[77] N.G. Van Kampen, *A power series expansion of the master equation*, Canad. J. Phys., 39 (1961), pp. 551–567.
[78] M. Vellela and H. Qian, *A quasistationary analysis of a stochastic chemical reaction: Keizer's paradox*, Bull. Math. Biol., 69 (2007), pp. 1727–1746.
[79] G. Wang, X. Zhu, J. Gu, and P. Ao, *Quantitative implementation of the endogenous molecular–cellular network hypothesis in hepatocellular carcinoma*, Interface Focus, 4 (2014), 20130064.
[80] X. Wang, N. Hao, H.G. Dohlman, and T.C. Elston, *Bistability, stochasticity, and oscillations in the mitogen-activated protein kinase cascade*, Biophys. J., 90 (2006), pp. 1961–1978.
[81] D.J. Wilkinson, *Stochastic modelling for quantitative description of heterogeneous biological systems*, Nat. Rev. Genet., 10 (2009), pp. 122–133.
[82] V. Wolf, R. Goel, M. Mateescu, and T. Henzinger, *Solving the chemical master equation using sliding windows*, BMC Syst. Biol., 4 (2010), 42.
[83] X.-M. Zhu, L. Yin, L. Hood, and P. Ao, *Calculating biological behaviors of epigenetic states in the phage λ life cycle*, Funct. Int. Genomics, 4 (2004), pp. 188–195.
[84] X.-M. Zhu, L. Yin, L. Hood, and P. Ao, *Robustness, stability and efficiency of phage lambda genetic switch: Dynamical structure analysis*, J. Bioinform. Comput. Biol., 2 (2004), pp. 785–817.