Transient Analysis of Stochastic Switches and Trajectories with Applications to Gene Regulatory Networks

Brian Munsky and Mustafa Khammash

Center for Control, Dynamical Systems and Computation;

University of California, Santa Barbara, CA 93106-5070

May 21, 2008

Abstract

Many gene regulatory networks are modeled at the mesoscopic scale, where chemical populations change according to a discrete state (jump) Markov process. The chemical master equation (CME) for such a process is typically infinite dimensional and unlikely to be computationally tractable without reduction. The recently proposed Finite State Projection (FSP) technique allows for a bulk reduction of the CME while explicitly keeping track of its own approximation error. In previous work, this error has been reduced in order to obtain more accurate CME solutions for many biological examples. In this paper, we show that this "error" has far more significance than simply the distance between the approximate and exact solutions of the CME. In particular, we show that this error term serves as an exact measure of the rate of first transition from one system region to another. We demonstrate how this term may be used to (i) directly determine the statistical distributions for stochastic switch rates, escape times, trajectory periods, and trajectory bifurcations, and (ii) evaluate how likely it is that a system will express certain behaviors during certain intervals of time. We also present two systems-theory based FSP model reduction approaches that are particularly useful in such studies. We illustrate the benefits of this approach to analyze the switching behavior of a stochastic model of Gardner's genetic toggle switch.

1 Introduction

Many biochemical systems contain so many molecules that chemicals can be described by continuous valued ordinary differential equations. However, many gene regulatory networks contain cellular components, such as genes, RNA molecules, and proteins, that must be quantified by discrete integer amounts. As a result, a slightly noisy environment will introduce significant randomness and result in phenomena such as stochastic switching [1], stochastic focussing [2], stochastic resonance, and other effects that cannot be captured with deterministic models and require a separate set of analytical tools.

A well-mixed chemical system at constant temperature and volume behaves as a discrete state Markov process [3]. Each state of this process corresponds to a specific population vector, and each reaction takes the system from one state to another. The system's probability distribution evolves according to a set of linear ODEs known as the chemical master equation (CME). The CME is relatively easy to define (see below), but it can be very difficult or impossible to solve. Most research on the CME has concentrated on simulating trajectories of the CME using methods such as the Stochastic Simulation Algorithm (SSA) [4] or a related approximation (see for example: [5,6]).

While these methods reliably provide samples of the process defined by the CME, they require a huge collection of simulations to obtain an accurate statistical solution. This becomes particularly troublesome when one wishes to compute the transition probabilities of very rare events or to compare distributions arising from slightly different parameter sets.

Recently, we developed a new approach to approximate the solution of the CME: the Finite State Projection (FSP) algorithm [7–11]. This approach systematically collapses the infinite state Markov process into a combination of a truncated finite state process and a single absorbing "error sink". The resulting system is finite dimensional and solvable. The probabilities of the truncated process give a lower bound approximation to the true CME solution. The probability measure of the error sink gives an exact computation of the error in this approximation. This error can then be decreased to reach any non-zero error tolerance through a systematic expansion of projections known as the FSP algorithm [7]. However, as we will illustrate in this paper, the "error" guarantee of the FSP provides more than a simple distance between the FSP solution and the true solution to the CME. Instead, this important term in the projection provides a wealth of *exact* information about the original Markov process. From it one can determine the statistical distributions of switch rates and escape probabilities and also analyze stochastic pathway bifurcation decisions.

Many recent studies have examined switch rates in the context of stochastic processes operating at their equilibrium or nonequilibrium steady state distributions. As a few representative examples, these methods include Transition Path Sampling [12], Transition Interface Sampling [13], and various approaches of transition path sampling with multiple interfaces [14–17]. By concentrating on trajectories that eventually result in switches and interrupting the the vast majority trajectories that do not, these approaches are far more efficient than a standard brute force Monte Carlo approach like the SSA. However, as trajectory based analyses, they are limited by the slow convergence of Monte Carlo approaches and cannot provide strict accuracy guarantees. In contrast to these methods, the current study focusses on the *transient evolution of probability distributions* and not on the sampled trajectories of a steady state process. The results sought in this paper are not histograms of waiting times between switches from one large potential well (or metastable state) to another, but are instead a set of precise upper and lower bounds on the distribution of transition times between specific states and/or arbitrarily chosen state space regions.

The paper is a significant extension of the work in [18] in which we explore the added information contained in the FSP "error" sink and present some of the types of analyses for which this information provides. The next section provides a precise summary of the original FSP results from [7] but with an emphasis on understanding the underlying intuition of the error sink. In Section 3 we show how multiple absorbing sinks can be used to effectively analyze pathway bifurcation decisions in stochastic systems. Then, in Section 4, we show how these sinks can be used to determine some statistical quantities for stochastic switches, such as switch waiting and return times, and we introduce two model reductions to the FSP that can help in the analysis of complex trajectories. In Section 5, we illustrate how these new approaches can be applied to a stochastic model of the genetic toggle switch [19]. Finally, in Section 6, we finish with some concluding remarks and future directions.

2 Background on the Finite State Projection approach

We consider the *mesoscopic* description of chemical kinetics, where the state of an N reactant process is defined by the integer population vector $\mathbf{x} \in \mathbb{Z}^N$. Reactions are transitions from one state to another $\mathbf{x} \rightarrow \mathbf{x} + \nu_\mu$, where ν_μ is known as the stoichiometry (or direction of transition) of the μ^{th} reaction. For a system containing M distinct reaction types, there are at most M reaction events that will take the system from $\mathbf{x}_i$ to some other state $\mathbf{x}_j \neq \mathbf{x}_i$ and at most M reaction events that will bring the system from $\mathbf{x}_k \neq \mathbf{x}_i$ to $\mathbf{x}_i$. Each reaction has an infinitesimal probability of occurring in the next infinitesimal time step of length dt; this state dependent quantity is known as the propensity function: $w_\mu(\mathbf{x})dt$.

If $P_i(t)$ and $P_i^\mu(t)$ are used to denote the probabilities that the system will be in $\mathbf{x}_i$ and $\mathbf{x}_i^\mu = \mathbf{x}_i - \nu_\mu$, respectively, at time t, then:

$$\frac{P_i(t+dt) - P_i(t)}{dt} = -\sum_{\mu=1}^{M} w_\mu(\mathbf{x}_i)P_i(t) - w_\mu(\mathbf{x}_i^\mu)P_i^\mu(t).$$

Taking the limit $dt \to 0$ easily yields the chemical master equation [20], which can be written in vector form as: $\dot{\mathbf{P}}(t) = \mathbf{AP}(t)$. The ordering of the infinitesimal generator, $\mathbf{A}$, is determined by the enumeration of the configuration set $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots\}$. Each i^{th} diagonal element of $\mathbf{A}$ is negative with a magnitude equal to the sum of the propensity functions of reactions that *leave* the i^{th} configuration. Each off-diagonal element, A_{ij}, is positive with magnitude $w_\mu(\mathbf{x}_j)$ if there is a reaction $\mu \in \{1, \ldots, M\}$ such that $\mathbf{x}_i = \mathbf{x}_j + \nu_\mu$ and zero otherwise. In other words:

$$\mathbf{A}_{ij} = \left\{ \begin{array}{ll} -\sum_{\mu=1}^{M} w_\mu(\mathbf{x}_i) & \text{for } (i = j) \\ w_\mu(\mathbf{x}_j) & \text{for all } j \text{ such that } (\mathbf{x}_i = \mathbf{x}_j + \nu_\mu) \\ 0 & \text{Otherwise} \end{array} \right\}. \tag{1}$$

When the cardinality of the set $\mathbf{X}$ is infinite or extremely large, the solution to the CME is unclear or vastly difficult to compute, but one can get a good approximation of that solution using Finite State Projection (FSP) techniques [7–10]. To review the FSP, we will first introduce some convenient notation. Let $J = \{j_1, j_2, j_3, \ldots\}$ denote an index set, and let J' denote the complement of the set J. If $\mathbf{X}$ is an enumerated set $\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots\}$, then $\mathbf{X}_J$ denotes the subset $\{\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, \mathbf{x}_{j_3}, \ldots\}$. Furthermore, let $\mathbf{v}_J$ denote the subvector of $\mathbf{v}$ whose elements are chosen according to J, and let $\mathbf{A}_{IJ}$ denote the submatrix of $\mathbf{A}$ such that the rows have been chosen according to I and the columns have been chosen according to J. For example, if I and J are defined as $\{3, 1, 2\}$ and $\{1, 3\}$, respectively, then:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & k \end{bmatrix}_{IJ} = \begin{bmatrix} g & k \\ a & c \\ d & f \end{bmatrix}.$$

For convenience, let $\mathbf{A}_J := \mathbf{A}_{JJ}$.

Let $\mathcal{M}$ denote a Markov chain on the configuration set $\mathbf{X}$, such as that shown in Fig. 1a, whose master equation is $\dot{\mathbf{P}}(t) = \mathbf{A}\mathbf{P}(t)$, with initial distribution $\mathbf{P}(0)$. Let $\mathcal{M}_J$ denote a reduced Markov chain, such as that in Fig. 1b, comprised of the configurations indexed by J plus a single absorbing state. The master equation of $\mathcal{M}_J$ is given by

$$\begin{bmatrix} \dot{\mathbf{P}}_J^{FSP}(t) \\ \dot{G}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_J & \mathbf{0} \\ -\mathbf{1}^T\mathbf{A}_J & 0 \end{bmatrix} \begin{bmatrix} \mathbf{P}_J^{FSP}(t) \\ G(t) \end{bmatrix}, \tag{2}$$

with initial distribution,

$$\begin{bmatrix} \mathbf{P}_J^{FSP}(0) \\ G(0) \end{bmatrix} = \begin{bmatrix} \mathbf{P}_J(0) \\ 1 - \sum \mathbf{P}_J(0) \end{bmatrix}.$$

At this point it is crucial to have a very clear understanding of how the process $\mathcal{M}_J$ relates to $\mathcal{M}$ and in particular the definitions of the terms $\mathbf{P}_J^{FSP}(t)$ and $G(t)$. First, the scalar $G(0)$ is the *exact* probability that the system begins in the set $\mathbf{X}_{J'}$ at time $t = 0$, and $G(t)$ is the *exact probability* that the system has been in the set $\mathbf{X}_{J'}$ at *any* time $\tau \in [0,t]$. Second, the vector $\mathbf{P}_J^{FSP}(0)$ contains the *exact probabilities* that the system begins in the set $\mathbf{X}_J$ at time $t=0$, and $\mathbf{P}_J^{FSP}(t)$ are the *exact joint probabilities* that the system (*i*) is in the corresponding states $\mathbf{X}_J$ at time t, and (*ii*) the system has remained in the set $\mathbf{X}_J$ *for all* $\tau \in [0,t]$. Note that $\mathbf{P}_J^{FSP}(t)$ also provides a finite dimensional approximation of the solution to the CME, as is clearly seen in the following reformulation of the original FSP theorems [7]:

Theorem 1. For any index set J and any initial distribution $\mathbf{P}(0)$, it is guaranteed that

$$\mathbf{P}_J(t) \geq \mathbf{P}_J^{FSP}(t) \geq \mathbf{0}.$$

Proof. $\mathbf{P}_J^{FSP}(t)$ is a more restrictive joint distribution than $\mathbf{P}_J(t)$.

Theorem 2. Consider any Markov chain $\mathcal{M}$ and its reduced Markov chain $\mathcal{M}_J$. If $G(t_f)=\varepsilon$, then

$$\left\| \begin{bmatrix} \mathbf{P}_J(t_f) \\ \mathbf{P}_{J'}(t_f) \end{bmatrix} - \begin{bmatrix} \mathbf{P}_J^{FSP}(t_f) \\ \mathbf{0} \end{bmatrix} \right\|_1 = \varepsilon. \tag{3}$$

Proof. The left side of (3) can be expanded to:

$$LHS = \left|\mathbf{P}_J(t_f) - \mathbf{P}_J^{FSP}(t_f)\right|_1 + \left|\mathbf{P}_{J'}(t_f)\right|_1.$$

Applying the Theorem 1 yields

$$LHS = \left|\mathbf{P}_J(t_f)\right|_1 - \left|\mathbf{P}_J^{FSP}(t_f)\right|_1 + \left|\mathbf{P}_{J'}(t_f)\right|_1.$$

Since $\mathbf{P}(t_f)$ is a probability distribution $\left|\mathbf{P}_J(t_f)\right|_1 + \left|\mathbf{P}_{J'}(t_f)\right|_1 = \left|\mathbf{P}(t_f)\right|_1 = 1$ and the LHS can be rewritten:

$$LHS = 1 - \left|\mathbf{P}_J^{FSP}(t_f)\right|_1.$$

Because the pair $\{G(t_f), \mathbf{P}_J^{FSP}(t_f)\}$ are a probability distribution for $\mathcal{M}_J$, one can see that the right hand side is precisely equal to $|G(t_f)|_1$ and the proof is complete.

Theorem 1 guarantees that as we add points to the subset $\mathbf{X}_J$, then $\mathbf{P}_J^{FSP}(t_f)$ monotonically increases toward the true solution of the CME, and Theorem 2 provides a certificate the approximation's accuracy.

In previous work, the probability lost to the absorbing state, $G(t)$, has been used only as in Theorem 2 as a means to evaluate the FSP projection in terms of its accuracy compared to the true CME solution. As a probability of first transition, however, this term is an exact exit rate, and therefore has an importance of its own, as we will see in the remainder of this paper.

3 Pathway Bifurcation analysis with the FSP

There are numerous examples in which biological systems decide between expressing two or more vastly different responses. These decisions occur in developmental pathways in multicellular organisms as heterogeneous cells divide and differentiate, in single cell organisms that radically adapt to survive or compete in changing environments, and even in viruses that must decide to lay dormant or make copies of themselves and ultimately destroy their host [1]. Many of these decisions are stochastic in nature, and models and methods are needed to determine the nature and probability of these decisions. Next, we show how the FSP approach can be adapted to answer some of these questions.

In the original FSP approach, a single absorbing state has been used, whose probability coincides with the probability that the system has exited the region $\mathbf{X}_J$. Suppose one wishes to know a little more about *how* the system has exited this region. For example in the process in Fig. 1a, one may ask:

Problem 1: What is the probability that the *first* time the system exits $\mathbf{X}_J$ it does so via reaction 1 (rightward horizontal arrow) or via reaction 3 (leftward diagonal arrow)?

Problem 2: What is the probability distribution for the population of species s_2 when the population of s_1 *first* exceeds a specific threshold, s_1^{max}?

These questions can be answered by creating a new Markov process with multiple absorbing states as shown in Fig. 1(c,d). Let $\mathcal{M}_J^*$ refer to such a chain where we have included K different absorbing states. The CME for the two problems above can be written as:

$$\begin{bmatrix} \dot{\mathbf{P}}_J^{FSP}(t) \\ \dot{\mathbf{G}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_J & \mathbf{0} \\ \mathbf{Q} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{P}_J^{FSP}(t) \\ \mathbf{G}(t) \end{bmatrix}, \tag{4}$$

where $\mathbf{G} = [G_0, \ldots, G_K]^T$ and the matrix $\mathbf{Q}$ is given in Problem 1 by:

$$\mathbf{Q}_{\mu i} = \left\{ \begin{array}{cc} a_\mu(\mathbf{x}_{j_i}) & \text{if } (\mathbf{x}_{j_i} + \nu_\mu) \notin \mathbf{X}_J \\ 0 & \text{Otherwise} \end{array} \right\},$$

and in Problem 2 by:

$$\mathbf{Q}_{ki} = \left\{ \begin{array}{cc} \sum a_\mu(\mathbf{x}_{j_i}) & \text{For all } j_i \text{ s.t. } (\mathbf{x}_{j_i})_2 = k \\ & \text{and } \mu \text{ s.t. } (\mathbf{x}_{j_i} + \nu_\mu)_1 > s_1^{max} \\ 0 & \text{Otherwise} \end{array} \right\}.$$

Note the underlying requirement that each j_i is an element of the index set J. Also recall that $\mathbf{x}_j$ is a population vector–the integer $(\mathbf{x}_j)_n$ is the n^{th} element of that population vector.

For either problem, the solution of (4) at a time t_f is found by taking the exponential of the matrix in (4) and has the form

$$\left[\begin{array}{c} \mathbf{P}_J^{FSP}(t) \\ \mathbf{G}(t) \end{array} \right] = \left[\begin{array}{cc} \exp(\mathbf{A}_J t_f) & \mathbf{0} \\ \int_0^{t_f} \mathbf{Q} \exp(\mathbf{A}_J \tau) d\tau & \mathbf{I} \end{array} \right] \left[\begin{array}{c} \mathbf{P}_J^{FSP}(0) \\ \mathbf{G}(0) \end{array} \right]. \tag{5}$$

This solution yields all of the same information as previous projections with regards to the accuracy of $\mathbf{P}_J^{FSP}(t)$, but it now provides additional useful knowledge. Specifically, each $G_k(t)$ gives the cumulative probability distribution at time t that the system will have exited from $\mathbf{X}_J$ at least once and that that exit transition will have occurred in the specific manner that was used to define the k^{th} absorbing state.

In [7] we showed a FSP algorithm that relied on increasing the set $\mathbf{X}_J$ until the solution reaches a certain pre-specified accuracy. This expansion was performed using the concept of $N-$step reachability, where each set $\{\mathbf{X}_{J_N}\}$ included all configurations that are reachable from $\mathbf{X}_{J_0}$ in N reactions or fewer. The additional knowledge gained from solving Problems 1 or 2 above is easily incorporated into this algorithm. If most of the probability measure left via one particular reaction or from one particular region of $\mathbf{X}_J$, it is reasonable to expand $\mathbf{X}_J$ accordingly. Such an approach is far more efficient than the original FSP algorithm and has been considered in [9].

4 Analyzing switch statistics with the FSP

As discussed above, the term $G(t)$ for the process $\mathcal{M}_J$ is simply the probability that the system has escaped from $\mathbf{X}_J$ at least once in the time interval $[0, t]$. With such an expression, it is almost trivial to find quantities such as median or p^{th} percentile escape times from the set $\mathbf{X}_J$. We need only find the time t such that $G(t)$ in (2) is equal to $p\%$. In other words, we find t such that

$$G(t) = 1 - \left|\exp(\mathbf{A}_J t)\mathbf{P}_J(0)\right|_1 = 0.01p. \tag{6}$$

This can be solved with a relatively simple line search as we will do in the example of Section 5. Using a multiple time interval FSP approach such as those explored in [9] or [10] could significantly speed up such a search, but this has not been utilized in this study.

Alternatively, one may wish to ask not only for escape times, but for the periods required to complete more complicated trajectories. For example, suppose the we have a Markov chain such as that in Fig. 2a. The system begins in the state represented by the shaded circle, and we wish to know the distribution for the time until the system will first visit the region in the grey box and then return to the original state. Biologically this may correspond to the probability that a system will switch from one phenotypical expression to another and then back again. To solve this problem, we can duplicate the lattice as shown in Fig. 2b. In this description, the top lattice corresponds to states where the system has never reached the grey box, and the bottom lattice corresponds to states where the system has first passed through that box. The master equation for this system is given by:

$$\begin{bmatrix} \dot{\mathbf{P}}^1_{J_1}(t) \\ \dot{\mathbf{P}}^2_{J_2}(t) \\ \dot{G}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{J_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_2\mathbf{C}_1 & \mathbf{A}_{J_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_2 & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{P}^1_{J_1}(t) \\ \mathbf{P}^2_{J_2}(t) \\ G(t) \end{bmatrix}, \tag{7}$$

where $\mathbf{X}_{J_1}$ includes every state except those in the grey box, and $\mathbf{X}_{J_2}$ includes every state except the final destination. The matrix $\mathbf{C}_1$ is the output matrix for the first sub-chain and accounts for transitions that exit the $\mathbf{X}_{J_1}$ (via a transition into the grey box):

$$[\mathbf{C}_1]_{ik} = \left\{ \begin{array}{ll} w_\mu(\mathbf{x}) & \text{for } \mathbf{x} = k^{th} \text{ state in } \mathbf{X}_{J_1}, \text{ and } \mathbf{x} + \nu_\mu = i^{th} \text{ state in the grey box} \\ 0 & \text{Otherwise} \end{array} \right\}. \tag{8}$$

The matrix $\mathbf{B}_2$ is the input matrix that maps the outputs of the first sub-chain to the correct states of the second sub-chain:

$$[\mathbf{B}_2]_{ji} = \left\{ \begin{array}{ll} 1 & \text{for } \mathbf{x} = j^{th} \text{ state in } \mathbf{X}_{J_2}, \text{ and } \mathbf{x} = i^{th} \text{ state in the grey box} \\ 0 & \text{Otherwise} \end{array} \right\}. \tag{9}$$

The probability of the absorbing point, $G(t)$, in this description is now exactly the probability that the system has completed the return trip in the time interval $[0, t]$. This solution scheme requires a higher dimensional problem than the original problem. However, with the FSP approach from [7], this dimension can be reduced while maintaining a strict measure of the method's accuracy.

4.1 Input-Output Description of Connected Markov Chains

Each part of the multiple phase trajectories described above has a common form:

$$\begin{aligned}\dot{\mathbf{P}}_i(t) &= \mathbf{A}_i\mathbf{P}_i(t) + \mathbf{B}_i\mathbf{u}_i(t)\\ \mathbf{y}_i(t) &= \mathbf{C}_i\mathbf{P}_i(t),\end{aligned} \tag{10}$$

where $\mathbf{u}_i(t)$ and $\mathbf{y}_i(t)$ are the flow of probability into and out of the the i^{th} Markov sub-chain, respectively, and $\mathbf{P}_i(t)$ is the vector of probabilities of the states within the i^{th} Markov sub-chain. In this description the *input matrix* $\mathbf{B}_i$ shows where and how the inputs enter into the i^{th} sub-chain, and the *output matrix* $\mathbf{C}_i$ maps the distribution $\mathbf{P}_i(t)$ to the output $\mathbf{y}_i(t)$. Once each input-output sub-system has been written in the form of the triplet $(\mathbf{A}_i, \mathbf{B}_i, \mathbf{C}_i)$, one may apply many standard tools to reduce their orders based upon Hankel singular values (see, for example, Chapter 4 of [21]). Many of these tools are available as part of the Robust Control Toolbox in MATLAB, and for the examples below, we will apply the MATLAB function *balancmr*. Upon application of these tools, the reduced system is then characterized by a lower order triplet $(\tilde{\mathbf{A}}_i, \tilde{\mathbf{B}}_i, \tilde{\mathbf{C}}_i)$, which can be directly substituted into (7).

4.2 Numerical Convolution to Compute Trajectory Times

So far complex trajectories were analyzed by creating a Markov sub-chain for each phase of the trajectory and then creating a new, much larger Markov chain by connecting these sub-chains in series. This can quickly result in a very high dimensional problem, that can require excessive memory and/or be very computationally intensive to solve. As an alternative, one can utilize the linearity of the system to treat each sub-chain separately and then reconnect them with a numerical convolution approach. For example, in Fig. 2b, one can first consider the top portion of the chain to find the rate of probability flow into the grey box as a response to beginning at the initial state u_0 at time $t = 0$. This flow is simply the response to the initial distribution:

$$\mathbf{y}(\tau) = \mathbf{C}_1 \exp(\mathbf{A}_{J_1}\tau)\mathbf{P}_{J_1}(0),$$

where each element of the vector $\mathbf{y}(\tau)$ corresponds to the flow into a specific point in the grey box. This probability flow is then the input to the bottom portion of the Markov chain. In practice $\mathbf{y}(\tau)$ is computed using an ODE solver and then stored at N points logarithmically distributed points between $t = 0$ and $t = t_f$. This discrete time signal is then interpolated for use as the forcing term for a second ODE system describing the bottom portion of the chain. Thus two smaller order ODEs are solved rather than a single much larger order system. One can readily extend this approach to compute the time distributions to complete more complicated trajectories such as hitting multiple way points or completing multiple circuits of the same return trip. In the next subsection, we illustrate how such a convolution based approach can be particularly useful in the computation of probabilities of complex trajectories.

4.3 Probabilities of Specific Trajectories

In addition to computing the time a system would take to complete a trajectory, one can also compute the probability that a system will exhibit specific traits at specific instances in time. Define a *partial* probability density vector $\mathbf{P}\{\mathcal{C}_n\}(t_n) = \mathbf{P}\{(\mathbf{X}_{J_0}, t_0); (\mathbf{X}_{J_1}, t_1); \ldots ; (\mathbf{X}_{J_{n-1}}, t_{n-1})\}(t_n)$, as the probability that the system satisfies the conditions $\{\mathcal{C}_n\}$:={it begins in the region $\mathbf{X}_{J_0}$ at $t = t_0$; is later is in the region $\mathbf{X}_{J_1}$ at the time $t_1 \geq t_0$; and so on until it is finally in the various states of $\mathbf{X}$ at the time $t_n \geq t_{n-1}\}$. Note that the vector $\mathbf{P}\{\mathcal{C}_n\}(t_n)$ has the same dimension as $\mathbf{X}$. The FSP approach provides a simple method to compute $\mathbf{P}\{\mathcal{C}_n\}(t_n)$, but first we need to define an embedding operator, $\mathcal{D}_J\{.\}$, as follows. Given any vector $\mathbf{v}$ and its J indexed subvector $\mathbf{v}_J$, the vector $\mathcal{D}_J\{\mathbf{v}_J\}$ has the same dimension as $\mathbf{v}$ and its only non-zero entries are the elements of $\mathbf{v}_J$ distributed according to the indexing set J. Furthermore, let $\mathbf{\Phi}(t_2 - t_1) = \exp(\mathbf{A}(t_2 - t_1))$ denote the transition operator that maps distributions at the time t_1 to the corresponding distributions at the later time t_2. Finally, let the vector $\mathbf{P}_{J_n}\{\mathcal{C}_n\}(t_n)$ denote the J_n-indexed sub-vector of $\mathbf{P}\{\mathcal{C}_n\}(t_n)$.

Proposition 1.[1] Using the above notation, the vector $\mathbf{P}\{\mathcal{C}_n\}(t_n)$ follows the recursive formula

$$\mathbf{P}\{\mathcal{C}_{n+1}\}(t_{n+1}) = \mathbf{\Phi}(t_{n+1} - t_n)\mathcal{D}_{J_n}\left\{\mathbf{P}_{J_n}\{\mathcal{C}_n\}(t_n)\right\}, \tag{11}$$

for all $t_0 \leq t_1 \leq \ldots \leq t_{n+1}$.

As a more general form, suppose that the conditions in $\mathcal{C}_n$ are that the system will be in sets $\{\mathbf{X}_{J_i}\}$ not at specific t_i's but at any time during the finite intervals $\{T_i = [a_i, b_i]\}$. In this case, we let the partial probability density vector $\mathbf{P}\{\mathcal{C}_n\}(t_n) = \mathbf{P}\{(\mathbf{X}_{J_0}, T_0); (\mathbf{X}_{J_1}, T_1); \ldots; (\mathbf{X}_{J_n}, T_{n-1})\}(t_n)$ denote the probability density that the system satisfies the conditions $\{\mathcal{C}_n\}$ that it begins in the region $\mathbf{X}_{J_0}$ at some $t_0 \in T_0$, is later in the region $\mathbf{X}_{J_1}$ at some $t_1 \in T_1$, and so on until it is finally in the various states of $\mathbf{X}$ at a time $t_n \geq b_n$. As above $\mathbf{P}\{\mathcal{C}_n\}(t_n)$ satisfies a recursive formula but in a more general form:

[1] The proof of Propositions 1 and 2 are to be included in supplemental materials (See Appendix in manuscript).

Proposition 2.[1] The vector $\mathbf{P}\{\mathcal{C}_n\}(t_n)$ follows the recursive formula

$$\mathbf{P}\{\mathcal{C}_{n+1}\}(t_{n+1}) = \mathbf{\Phi}(t_{n+1} - a_n)\mathcal{D}_{J_n}\left\{\mathbf{P}_{J_n}\{\mathcal{C}_n\}(a_n)\right\} + \int_{a_n}^{b_n} \mathbf{\Phi}(t_{n+1} - \tau)\mathbf{y}(\tau)d\tau, \tag{12}$$

where $\mathbf{y}(\tau)$ is given by

$$\mathbf{y}(\tau) = \mathcal{D}_{J_n}\{\mathbf{A}_{J_n,J'_n}\exp(\mathbf{A}_{J'_n}(\tau - a_n))\mathbf{P}_{J'_n}\{\mathcal{C}_n\}(a_n)\}.$$

In the following section, this approach and the methods from above are each applied in the analysis of a stochastic model for Gardner's genetic toggle switch [19].

5 Analyzing the genetic toggle switch

To illustrate the usefulness of the absorbing sink of the FSP in the analysis of stochastic gene regulatory networks, we consider a stochastic model of Gardner's gene toggle model [19]. This system, shown in Fig. 3 is comprised of two mutually inhibiting proteins, s_1 and s_2. The four reactions, R_μ, and corresponding propensity functions, w_μ, are given as:

$$\begin{array}{llllllll}
R_1 & ; & R_2 & ; & R_3 & ; & R_4 \\
\emptyset \rightarrow s_1 & ; & s_1 \rightarrow \emptyset & ; & \emptyset \rightarrow s_2 & ; & s_2 \rightarrow \emptyset \\
w_1 = \frac{16}{1+s_2} & ; & w_2 = s_1 & ; & w_3 = \frac{50}{1+s_1^{2.5}} & ; & w_4 = s_2
\end{array}$$

For these parameters, the system exhibits two distinct phenotypes, which for convenience we will label OFF and ON. By definition, we will call the cell OFF when the population of s_1 exceeds 5 molecules and s_2 is less than 15 molecules, and we will label the system as being ON when the population of s_2 exceeds 15 molecules and s_1 is less than 5 molecules. Each of these phenotypes is relatively stable–once the system reaches the ON or OFF state, it tends to stay there for some time. For this study, we assume that the system begins with a population $s_1 = s_2 = 0$, and we wish to analyze the subsequent switching behavior.

Q1. *After the process starts, the system will move within its configuration space until eventually the cell turns OFF or the cell turns ON. What percentage will choose to turn ON first (s_2 exceeds 15* before *s_1 exceeds 5)?*

To analyze this initial switch decision, we use the methodology outlined in Section 3. We choose $\mathbf{X}_J$ to include all states such that $s_1 \leq 5$ and $s_2 \leq 15$. There are only two means through which the system may exit this region: If $s_1 = 5$ and R_1 occurs (making $s_1 = 6$), then the system is absorbed into a sink state G_{OFF}. If $s_2 = 15$ and R_3 occurs, then the system is absorbed into a sink state G_{ON}. The master equation for this Markov chain has the form of that in (4) and contains 98 states including the two absorbing sinks. By solving this equation for the given initial condition, one can show that the probability of turning ON first is 78.1978%. Thus, nearly four-fifths of the cells will turn ON before they turn OFF. The asymptotes of the dashed lines in Fig. 4b correspond to the probabilities of that the system will first turn ON and OFF, respectively.

Q2. *Find the times t_{50} and t_{99} at which 50% and 99% of all cells will have made their initial decision to turn ON or OFF?*

To solve this question we can use the same Markov chain as in Q1, and search for the times, t_{50} and t_{99}, at which $G_{OFF}(t_{50}) + G_{ON}(t_{50}) = 0.5$ and $G_{OFF}(t_{99}) + G_{ON}(t_{99}) = 0.99$, respectively. This has been done using a simple line search, in which we found that $t_{50} = 0.5305s$ and $t_{99} = 5.0595s$. In Fig 4b these times correspond to the points in time where the dashed line labeled "First Switch" crosses 0.5 and 0.99, respectively.

Q3. *What is the time at which 99% of all cells will have turned ON at least once?*

Because we must include the possibility that the cell will first turn OFF and then turn ON, the solution for this question requires a different projection. Let $\mathbf{X}$ be the set of states such that $s_1 \leq 50$, $s_2 \leq 105$, and $s_1 s_2 \leq 300$. Furthermore, let the projection, $\mathbf{X}_{ON'}$ includes all states in $\mathbf{X}$ that are *not* ON ($s_1 < 15$ *or* $s_2 > 5$). As time passes, probability will leave this region in two manners: either it exits in to the aggregated ON sink (G_{ON}) or it exits out of $\mathbf{X}$ altogether in to a second absorbing sink G_{err}, which results in a loss of precision. This error comes into play as follows: If t_1 is defined as the time at which $G_{ON}(t_1)+G_{err}(t_1) = 0.99$, and t_2 is defined as the time at which $G_{ON}(t_2) = 0.99$, then the time, t_{99}, at which 99% turn ON is bounded by $t_1 \leq t_{99} \leq t_2$. For the chosen projection, this bound is very tight yielding a guarantee that $t_{99} \in [1733.3153, 1733.3157]s$. For comparison 10^4 runs of the SSA give a much less accurate estimate of $t_{99} \approx 1735.7$. Similarly, one can use a projection $\mathbf{X}_{OFF'}$, which includes all points in $\mathbf{X}$ that are not OFF, to find that it will take between 800.495 and 800.487 seconds until 99% of cells will turn OFF (compared to $t_{99} \approx 827s$ found with 10^4 SSA runs). Median times, t_{50}, have also been computed and are listed in Table 1.

Note that the times for Q3 are very large in comparison to those in Q2; this results from the fact that the ON and OFF regions are relatively stable. This trait is evident in Fig. 4, where the dashed lines correspond to the time of the first ON (or OFF) decision *provided that the system has not previously turned OFF (or ON).* Since about 78% percent turn ON before they turn OFF, this dashed ON curve asymptotes at about 0.78 (see Q1 and Q2). On the other hand, the solid lines corresponds to the times for the first ON (or OFF) decision *whether or not the system has previously turned OFF (or ON).* The kinks in these distributions, where the solid and dashed curves separate, result from the stability of the two regions region. In particular, the solid ON curve exhibits a more severe kink due to the fact that the OFF region is more stable than the ON region (compare solid lines).

The projections $\mathbf{X}_{ON'}$ and $\mathbf{X}_{OFF'}$ used here included 715 and 782 states respectively. While systems of this size are still relatively inexpensive to analyze, the computational cost will build significantly should we desire to add more complexity. Using balanced truncation, each of these systems can be reduced to 10^{th} order with very little loss in accuracy (compare solid lines and circle markers in Fig. 4, and see Table 1).

Q4. *What is the distribution for the round trip time until a cell will first turn ON and then turn OFF?*

In order to answer this question we use the round-trip methodology from the latter half of Section 4. Intuitively, this approach is very similar to that depicted in Fig. 2b, except that the top and bottom portions of the Markov chain are not identical and the final destination is a region of the chain as opposed to a single point. Also, since the Markov process under examination is infinite dimensional, we first apply a finite state projection to reduce this system to the finite set $\mathbf{X}$ described in Q3. For the system's outbound journey into the ON region, we use the projection $\mathbf{X}_{ON'}$ from Q3. After the system turns ON, it begins the second leg of its trip to the OFF region through a different projection $\mathbf{X}_{OFF'}$. When the system reaches the OFF region on the second leg, it is absorbed into a sink $G(t)$. The full master equation for this process can be written as:

$$\begin{bmatrix} \dot{\mathbf{P}}^1_{ON'}(t) \\ \dot{\mathbf{P}}^2_{OFF'}(t) \\ \dot{G}(t) \\ \varepsilon(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{ON'} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_2\mathbf{C}_1 & \mathbf{A}_{OFF'} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_3\mathbf{C}_2 & 0 & 0 \\ \mathbf{B}_\varepsilon\mathbf{C}_1 & \mathbf{B}_\varepsilon\mathbf{C}_2 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{P}^1_{ON'}(t) \\ \mathbf{P}^2_{OFF'}(t) \\ G(t) \\ \varepsilon(t) \end{bmatrix}, \tag{13}$$

where $\mathbf{A}_{ON'}$ and $\mathbf{A}_{OFF'}$ are defined as in (1). The matrices $\mathbf{C}_1$ and $\mathbf{B}_2$ are defined as in (8) and (9) above and account for the transitions from the states in $\mathbf{X}_{ON'}$ to the corresponding states in $\mathbf{X}_{OFF'}$. The vector $\mathbf{B}_3\mathbf{C}_2$ corresponds to the transitions that exit $\mathbf{X}_{OFF'}$ and turn OFF (completing the full trip). The last two vectors $\mathbf{B}_\varepsilon\mathbf{C}_1$ and $\mathbf{B}_\varepsilon\mathbf{C}_2$ correspond to the rare transitions that leave the projected space, $\mathbf{X}$, and therefore contribute to a computable error, $\varepsilon(t)$ in our analysis.

The solution of this system for the scalar $G(t)$ then gives us the joint probability that (*i*) the system remains in the set $\mathbf{X}_{ON'}$ until it enters the ON region at some time $\tau_1 \in [0, t)$, and (*ii*) it then remains in the set $\mathbf{X}_{OFF'}$ until it enters the OFF region at some time $\tau_2 \in (\tau_1, t]$. This distribution is plotted with the dotted lines in Fig. 4. Once again we can see the effect that the asymmetry of the switch plays on the times of these trajectories; the ON region is reached first more often and the ON region is less stable, thus the ON then OFF trajectory will occur significantly faster than the OFF then ON trajectory (compare dotted lines in Fig. 4, and see Table 1).

In Fig. 4, the distributions have been computed in two different manners, which yield nearly indistinguishable results (Compare lines and circles in Fig. 4). First, the lines correspond to solutions where (13) has been solved as a single large system of 1496 ODEs. In the second approach, the system has been analyzed as two separate sub-systems defined by the triplets $SYS_1 = (\mathbf{A}_{ON'}, \mathbf{P}_{ON'}, \mathbf{C}_1)$ and $SYS_2 = (\mathbf{A}_{OFF'}, \mathbf{B}_2, \mathbf{B}_3\mathbf{C}_2)$. Each of these systems has been reduced to 10^{th} order using balanced truncation. Once reduced, the systems were again reconnected resulting in a 22^{nd} order approximation, consisting of the two 10^{th} order reduced systems plus $G(t)$ and $\varepsilon(t)$. Table 1 gives the predicted median time t_{50} and the associated computational costs for these methods as well as for 10^4 runs of the stochastic simulation algorithm (SSA). Both FSP methods are far faster and more accurate than the corresponding SSA approach. Comparing the full and reduced FSP approaches, note that the reduced systems retain a high degree of the full systems' accuracy, but the reduction itself is very expensive. In these numerical experiments, we have used MATLAB's balanced truncation code *balancmr*, which does not take advantage of the extreme sparsity of the FSP formulation. With parallel algorithms for the balanced truncation of sparse systems, such as those in [22], much of this computational cost may be recovered.

Q5. *What is the probability that the system will be* (i) *ON at some point* $t_1 \in [a_1, b_1] = [100s, 110s]$*, then* (ii) *OFF at some point* $t_2 \in [a_2, b_2] = [200s, 210s]$ *and finally* (iii) *ON at* $t_3 = 300s$*?*

To answer this question we again use the projections, $\mathbf{X}$, $\mathbf{X}_{ON'}$ and $\mathbf{X}_{OFF'}$ from above. In terms of the notation used in Section 4.3, we are seeking to compute $\mathbf{P}_{ON}\{\mathcal{C}_3\}(t_3)$, where $\{\mathcal{C}_3\} =$

$\{(\mathbf{x}_0, 0); (\mathbf{X}_{ON}, [100, 110]); (\mathbf{X}_{OFF}, [200, 210])\}$. This computation is done recursively as follows:

$$\begin{aligned}
\mathbf{P}\{\mathcal{C}_1\}(a_1) &= \exp(\mathbf{A}a_1)\mathbf{P}(0) \\
\mathbf{P}\{\mathcal{C}_2\}(a_2) &= \exp(\mathbf{A}(a_2 - a_1))\mathcal{D}_{ON}\{\mathbf{P}_{ON}\{\mathcal{C}_1\}(a_1)\} + \int_{a_1}^{b_1} \exp(\mathbf{A}(a_2 - \tau))\mathbf{y}_1(\tau)d\tau \\
\mathbf{y}_1(\tau) &= \mathcal{D}_{ON}\{\mathbf{A}_{ON,ON'} \exp(\mathbf{A}_{ON'}(\tau - a_1))\mathbf{P}_{ON'}\{\mathcal{C}_1\}(a_1)\} \\
\mathbf{P}\{\mathcal{C}_3\}(a_3) &= \exp(\mathbf{A}(t_3 - a_2))\mathcal{D}_{OFF}\{\mathbf{P}_{OFF}\{\mathcal{C}_2\}(a_2)\} + \int_{a_2}^{b_2} \exp(\mathbf{A}(t_3 - \tau))\mathbf{y}_2(\tau)d\tau \\
\mathbf{y}_2(\tau) &= \mathcal{D}_{OFF}\{\mathbf{A}_{OFF,OFF'} \exp(\mathbf{A}_{OFF'}(\tau - a_2))\mathbf{P}_{OFF'}\{\mathcal{C}_2\}(a_2)\}
\end{aligned}$$

Using this approach, one can compute the probability of the first measurement $|\mathbf{P}_{ON}\{\mathcal{C}_1\}(100)|_1 = 0.543$, $|\mathbf{P}_{OFF}\{\mathcal{C}_2\}(200)|_1 = 0.174$, and $|\mathbf{P}_{ON}\{\mathcal{C}_3\}(300)|_1 = 0.0266$. Also, by keeping track of the amount of the probability measure that exits $\mathbf{X}$ through each stage, one can obtain a guarantee that these computations are accurate to within relative errors of 9.1×10^{-6}, 4.9×10^{-5}, and 3.3×10^{-4} percent, respectively. The total computational effort is 63.2s. For comparison 10^4 SSA runs take 2020s to complete this same study, and provide an estimate for $|\mathbf{P}_{ON}\{\mathcal{C}_3\}(300)|_1$ of 0.0270, which is a relative error of 1.63%.

6 Conclusion

In order to account for the importance of intrinsic noise, researchers model many gene regulatory networks as being jump Markov processes. In this description, each state corresponds to a specific integer population vector, and transitions correspond to individual reactive events. These processes have probability distributions that evolve according to a possibly infinite dimensional chemical master equation (CME) [3]. In previous work, we showed that the Finite State Projection (FSP) method [7] can provide a very accurate solution to the CME for some stochastic gene regulatory networks. The FSP method works by choosing a small finite set of possible states and then keeping track of how much of the probability measure exits that set as time passes. In the original FSP, the amount of the probability measure that exits the chosen region yields bounds on the FSP approximation error. In this paper we have shown that this exit probability has intrinsic value and can allow for the precise computation of the statistics of switching times, escape times and completion times for more complicated trajectories. Unlike previous analyses of stochastic switch rates that utilize Monte Carlo type approaches [12–17], the current method directly approximates the transient solution to the master equation and provides otherwise unachievable precision guarantees on the switch time distribution. At present this precision comes at a cost of adverse complexity scaling. For system with large numbers of molecular species, the approach suffers from an exponential explosion of states. But with model reduction and advanced solution techniques such as the balanced truncation approach above and those described in [9–11], it is envisioned that one may overcome much of this curse of dimensionality.

Although the methods presented here have applicability well beyond systems biology, we have specifically illustrated these techniques on a stochastic model of Gardner's genetic toggle switch [19]. We have used the FSP to find the distribution for the times at which the system first turns ON or OFF as well as the time until the system will complete a trajectory in which it first switches one way and then the other. In each of these computations, the FSP results come with extremely precise guarantees as to their own accuracy.

Acknowledgment

The authors would like to acknowledge Eric Klavins, with whom we have many interesting discussions on related topics. This material is based upon work supported by the National Science Foundation under Grant NSF-ITR CCF-0326576 and the Institute for Collaborative Biotechnologies through Grant DAAD19-03-D-0004 from the U.S. Army Research Office.

References

[1] A. Arkin, J. Ross, and M. H., “Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected escherichia coli cells,” *Genetics*, vol. 149, pp. 1633–1648, 1998.

[2] J. Paulsson, O. Berg, and M. Ehrenberg, “Stochastic focusing: Fluctuation-enhanced sensitivity of intracellular regulation,” *PNAS*, vol. 97, no. 13, pp. 7148–7153, 2000.

[3] D. T. Gillespie, “A rigorous derivation of the chemical master equation,” *Physica A*, vol. 188, pp. 404–425, 1992.

[4] ——, “Exact stochastic simulation of coupled chemical reactions,” *J. Phys. Chem.*, vol. 81, no. 25, pp. 2340–2360, May 1977.

[5] ——, “Approximate accelerated stochastic simulation of chemically reacting systems,” *J. Chem. Phys.*, vol. 115, no. 4, pp. 1716–1733, Jul. 2001.

[6] Y. Cao, D. Gillespie, and L. Petzold, “The slow-scale stochastic simulation algorithm,” *J. Chem. Phys.*, vol. 122, no. 014116, Jan. 2005.

[7] B. Munsky and M. Khammash, “The finite state projection algorithm for the solution of the chemical master equation,” *J. Chem. Phys.*, vol. 124, no. 044104, 2006.

[8] S. Peles, B. Munsky, and M. Khammash, “Reduction and solution of the chemical master equation using time-scale separation and finite state projection,” *J. Chem. Phys.*, vol. 125, no. 204104, Nov. 2006.

[9] B. Munsky and M. Khammash, “A multiple time interval finite state projection algorithm for the solution to the chemical master equation,” *J. Comp. Phys.*, vol. 226, no. 1, pp. 818–835, 2007.

[10] K. Burrage, M. Hegland, S. Macnamara, and R. Sidje, “A krylov-based finite state projection algorithm for solving the chemical master equation arising in the discrete modelling of biological systems,” *Proc. of The A.A.Markov 150th Anniversary Meeting*, pp. 21–37, 2006.

[11] B. Munsky and M. Khammash, “The finite state projection approach for the analysis of stochastic noise in gene networks,” *IEEE Trans. Automat. Contr./IEEE Trans. Circuits and Systems: Part 1*, vol. 52, no. 1, pp. 201–214, Jan. 2008.

[12] P. Bolhuis, D. Chandler, C. Dellago, and P. Geissler, “Transition path sampling: Throwing ropes over rough mountain passes, in the dark,” *Annu. Rev. Phys. Chem.*, vol. 53, pp. 291–318, 2002.

[13] T. van Erp. and P. Bolhuis, “Elaborating transition interface sampling methods,” *J. Comp. Phys.*, vol. 205, pp. 157–181, 2005.

[14] A. Faradjian and R. Elber, “Computing time scales from reaction coordinates by milestoning,” *J. Chem. Phys.*, vol. 120, no. 23, pp. 10 880–10 889, 2004.

[15] D. Moroni, P. Bolhuis, and T. van Erp., "Rate constants for diffusive processes by partial path sampling," *J. Chem. Phys.*, vol. 120, no. 9, pp. 4055–4065, 2004.

[16] R. Allen, P. Warren, and P. Rein ten Wolde, "Sampling rare switching events in biochemical networks," *Phys. Rev. Lett.*, vol. 94, no. 018104, Jan. 2005.

[17] R. Allen and P. Frenkel, D. Rein ten Wolde, "Simulating rare events in equilibrium or nonequilibrium stochastic systems," *J. Chem. Phys.*, vol. 124, no. 024102, Jan. 2006.

[18] B. Munsky and M. Khammash, "Computation of switch time distributions in stochastic gene regulatory networks," *Submitted to the 26th American Control Conference (ACC)*, July 2008.

[19] T. Gardner, C. Cantor, and J. Collins, "Construction of a genetic toggle switch in escherichia coli," *Nature*, vol. 403, pp. 339–242, 2000.

[20] N. van Kampen, *Stochastic Processes in Physics and Chemistry*, 3rd ed. Elsevier, 2001.

[21] G. Dullerud and F. Paganini, *A Course in Robust Control Theory: a Convex Approach*, 1st ed. Springer, 2000.

[22] J. Badfa, P. Benner, R. Mayo, and E. Quintana-Orti, "Parallel algorithms for balanced truncation model reduction of sparse systems," *Applied Parallel Computing*, vol. 3732, pp. 267–275, 2006.

Supporting Material for "Precise Transient Analysis of Switches and Trajectories in Stochastic Gene Regulatory Networks."

Here we give the proofs of Propositions 1 and 2 from Section 4.3. These are to be supplied online as supporting materials.

Proposition 1. Using the above notation, the vector $\mathbf{P}\{\mathcal{C}_n\}(t_n)$ follows the recursive formula

$$\mathbf{P}\{\mathcal{C}_{n+1}\}(t_{n+1}) = \mathbf{\Phi}(t_{n+1} - t_n)\mathcal{D}_{J_n}\left\{\mathbf{P}_{J_n}\{\mathcal{C}_n\}(t_n)\right\}, \tag{14}$$

for all $t_0 \leq t_1 \leq \ldots \leq t_{n+1}$.

Proof. Let $\mathbf{P}(t_n)$ be the full probability distribution at t_n, which can be separated into two parts:

$$\mathbf{P}(t_n) = \mathbf{P}\{\mathcal{C}_n\}(t_n) + \mathbf{P}\{\mathcal{C}_n'\}(t_n),$$

where $\mathbf{P}\{\mathcal{C}_n\}(t_n)$ and $\mathbf{P}\{\mathcal{C}_n'\}(t_n)$ are the partial distributions at t_n that do and do not satisfy the conditions in $\{\mathcal{C}_n\}$, respectively. The full distribution distribution at t_{n+1} is

$$\begin{aligned}
\mathbf{P}(t_{n+1}) &= \mathbf{\Phi}(t_{n+1} - t_n)\mathbf{P}(t_n) \\
&= \mathbf{\Phi}(t_{n+1} - t_n)\left(\mathbf{P}\{\mathcal{C}_n\}(t_n) + \mathbf{P}\{\mathcal{C}_n'\}(t_n)\right) \\
&= \mathbf{\Phi}(t_{n+1} - t_n)\big(\mathcal{D}_{J_n}\mathbf{P}_{J_n}\{\mathcal{C}_n\}(t_n) + \mathcal{D}_{J_n'}\mathbf{P}_{J_n'}\{\mathcal{C}_n\}(t_n) + \mathbf{P}\{\mathcal{C}_n'\}(t_n)\big)
\end{aligned}$$

where J_n' denotes the complement of J_n. By definition the partial distribution $\mathbf{\Phi}(t_{n+1}-t_n)\mathcal{D}_{J_n}\mathbf{P}_{J_n}\{\mathcal{C}_n\}(t_n)$ satisfies the conditions $\{\mathcal{C}_{n+1}\}$, while the second and third partial distribution terms $\mathbf{\Phi}(t_{n+1} - t_n)\mathcal{D}_{J_n'}\mathbf{P}_{J_n'}\{\mathcal{C}_n\}(t_n)$ and $\mathbf{\Phi}(t_{n+1} - t_n)\mathbf{P}\{\mathcal{C}_n'\}(t_n)$ do not, and we are left with the final result in (14).

Proposition 2. The vector $\mathbf{P}\{\mathcal{C}_n\}(t_n)$ follows the recursive formula

$$\mathbf{P}\{\mathcal{C}_{n+1}\}(t_{n+1}) = \mathbf{\Phi}(t_{n+1} - a_n)\mathcal{D}_{J_n}\left\{\mathbf{P}_{J_n}\{\mathcal{C}_n\}(a_n)\right\} + \int_{a_n}^{b_n} \mathbf{\Phi}(t_{n+1} - \tau)\mathbf{y}(\tau)d\tau, \tag{15}$$

where $\mathbf{y}(\tau)$ is given by

$$\mathbf{y}(\tau) = \mathcal{D}_{J_n}\{\mathbf{A}_{J_n,J_n'}\exp(\mathbf{A}_{J_n'}(\tau - a_n))\mathbf{P}_{J_n'}\{\mathcal{C}_n\}(a_n)\}.$$

Proof. Let $\mathbf{P}(t)$ be the full probability distribution at $t \geq a_n$, which can be separated into two parts:

$$\mathbf{P}(t_{a_n}) = \mathbf{P}\{\mathcal{C}_n\}(t_{a_n}) + \mathbf{P}\{\mathcal{C}_n'\}(t_{a_n}),$$

where $\mathbf{P}\{\mathcal{C}_n\}(t_{a_n})$ and $\mathbf{P}\{\mathcal{C}_n'\}(t)$ are the portions of the distribution that do and do not satisfy $\{\mathcal{C}_n\}$, respectively. Furthermore, we can separate $\mathbf{P}\{\mathcal{C}_n\}(t_{a_n})$ into two components

$$\mathbf{P}\{\mathcal{C}_n\}(t_{a_n}) = \mathbf{w}(t) + \mathcal{D}_{J_n'}\{\mathbf{z}_{J_n'}(t)\},$$

where $\mathbf{w}(t)$ is the partial probability distribution that satisfies $\{\mathcal{C}_n\}$ and the additional condition that the system is in $\mathbf{X}_{J_n}$ at *any* time $\tau \in [a_n, t]$, and $\mathbf{z}_{J_n}(t)$ is the partial distribution where the system satisfies $\{\mathcal{C}_n\}$ and the additional condition that the system remains in the set $\mathbf{X}_{J'_n}$ for *all* times $\tau \in [a_n, t]$. Note that elements of $\mathbf{w}(t)$ refer to each of the states in $\mathbf{X}$ while $\mathbf{z}_{J'_n}(t)$ refers only to states in $\mathbf{X}_{J'_n}$. During the interval $[a_n, b_n]$ the partial distributions $\mathbf{w}(t)$ and $\mathbf{z}_{J'_n}(t)$ evolve according to the linear system:

$$\begin{bmatrix} \dot{\mathbf{w}}_{J_n}(t) \\ \dot{\mathbf{w}}_{J'_n}(t) \\ \dot{\mathbf{z}}_{J'_n}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{J_n} & \mathbf{A}_{J_n,J'_n} & \mathbf{A}_{J_n,J'_n} \\ \mathbf{A}_{J'_n,J_n} & \mathbf{A}_{J'_n} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_{J'_n} \end{bmatrix} \begin{bmatrix} \mathbf{w}_{J_n}(t) \\ \mathbf{w}_{J'_n}(t) \\ \mathbf{z}_{J'_n}(t) \end{bmatrix},$$

with initial conditions

$$\begin{bmatrix} \mathbf{w}_{J_n}(a_n) \\ \mathbf{w}_{J'_n}(a_n) \\ \mathbf{z}_{J'_n}(a_n) \end{bmatrix} = \begin{bmatrix} \mathcal{D}_{J_n}\{\mathbf{P}_{J_n}\{\mathcal{C}_n\}(a_n)\} \\ \mathbf{0} \\ \mathcal{D}_{J'_n}\{\mathbf{P}_{J'_n}\{\mathcal{C}_n\}(a_n)\} \end{bmatrix}.$$

Solving this system at $t = b_n$ yields:

$$\begin{aligned} \mathbf{w}(b_n) &= \mathbf{\Phi}(b_n - a_n)\mathcal{D}_{J_n}\{\mathbf{P}_{J_n}\{\mathcal{C}_n\}(a_n)\} + \int_{a_n}^{b_n} \mathbf{\Phi}(b_n - \tau)\mathcal{D}\{\mathbf{A}_{J_n,J'_n}\exp(\mathbf{A}_{J'_n}(\tau - a_n))\mathbf{P}_{J'_n}\{\mathcal{C}_n\}(a_n)\} \\ &= \mathbf{\Phi}(b_n - a_n)\mathcal{D}_{J_n}\{\mathbf{P}_{J_n}\{\mathcal{C}_n\}(a_n)\} + \int_{a_n}^{b_n} \mathbf{\Phi}(b_n - \tau)\mathbf{y}(\tau), \text{ and} \\ \mathbf{z}_{J'_n}(b_n) &= \exp(\mathbf{A}_{J'_n}(b_n - a_n))\mathbf{P}_{J'_n}\{\mathcal{C}_n\}(a_n). \end{aligned}$$

The total distribution at time t_{n+1} can be written as:

$$\begin{aligned}\mathbf{P}(t_{n+1}) &= \mathbf{\Phi}(t_{n+1}-b_n)\mathbf{P}\{\mathcal{C}_n(t_{a_n})\} + \mathbf{\Phi}(t_{n+1}-a_n)\mathbf{P}\{\mathcal{C}'_n\}(t_{a_n}) \\ &= \mathbf{\Phi}(t_{n+1}-b_n)\left(\mathbf{w}(b_n) + \mathcal{D}_{J'_n}\{\mathbf{z}_{J'_n}(b_n)\}\right) + \mathbf{\Phi}(t_{n+1}-a_n)\mathbf{P}\{\mathcal{C}'_n\}(t_{a_n})\end{aligned}$$

By the definitions of $\mathbf{w}(t)$, $\mathbf{z}_{J'_n}(b_n)$ and $\mathbf{P}\{\mathcal{C}'_n\}(t_{a_n})$, only $\mathbf{\Phi}(t_{n+1}-b_n)\mathbf{w}(b_n)$ satisfies the conditions of $\{\mathcal{C}_{n+1}\}$, and

$$\begin{aligned}\mathbf{P}\{\mathcal{C}_{n+1}\}(t_{n+1}) &= \mathbf{\Phi}(t_{n+1}-b_n)\mathbf{w}(b_n) \\ &= \mathbf{\Phi}(t_{n+1}-b_n)\left(\mathbf{\Phi}(b_n-a_n)\mathcal{D}_{J_n}\{\mathbf{P}_{J_n}\{\mathcal{C}_n\}(a_n)\} + \int_{a_n}^{b_n}\mathbf{\Phi}(b_n-\tau)\mathbf{y}_n(\tau)\right) \\ &= \mathbf{\Phi}(t_{n+1}-a_n)\mathcal{D}_{J_n}\{\mathbf{P}_{J_n}\{\mathcal{C}_n\}(a_n)\} + \int_{a_n}^{b_n}\mathbf{\Phi}(t_{n+1}-\tau)\mathbf{y}_n(\tau),\end{aligned}$$

thus completing the proof.

Figure 1: (a): A Markov chain for a two species chemically reacting system, $\mathcal{M}$. The process begins in the configuration shaded in grey and undergoes three reactions: The first reaction $\emptyset \to s_1$ results in a net gain of one s_1 molecule and is represented by right arrows. The second reaction $s_1 \to \emptyset$ results in a net loss of one s_1 molecule and is represented by a left arrow. The third reaction $s_1 \to s_2$ results in a loss of one s_1 molecule and a gain of one s_2 molecule. The dimension of the Master equation is equal to the total number of configurations in $\mathcal{M}$, and is too large to solve exactly. (b) In the FSP algorithm a configuration subset $\mathbf{X}_J$ is chosen, and all remaining configurations are projected to a single absorbing point G. This results in a small dimensional Markov process, $\mathcal{M}_J$. (c,d) Instead of considering only a single absorbing point, transitions out of the finite projection can be sorted as to how they leave the projection space. (c) G_1 and G_3 absorb the probability that has leaked out through reactions 1 or 3, respectively. This information can then be used to analyze the probabilities of certain decisions or to expand the configuration set in later iterations of the FSP algorithm. (d) Each G_i absorbs the probability that s_1 first exceeds a certain threshold, s_1^{max} when $s_2 = i$.

Figure 2: Schematic representation for the computation of round trip times for discrete state Markov processes. (a) A Markov chain $\mathcal{M}$ where the system begins in the shaded circle, and we wish to find the distribution for the time at which the system first enters then shaded region and then returns to the initial state. (b) A corresponding Markov process where the top points correspond to states on the journey from the dark circle to the shaded box, and the bottom circles correspond to states along the return trip. In this description, the absorbing point $G(t)$ corresponds to the probability that the system has gone from the initial condition to the grey box and then back again.

Figure 3: Schematic of the toggle model comprised of two inhibitors: s_1 inhibits the production of s_2 and vice-versa.

Figure 4: Probability densities (a) and cumulative distributions (b) of the times of switch decisions for a stochastic model of Gardner's gene toggle switch [19]. The dashed lines correspond to the probabilities that the first switch decision will be to enter the ON or OFF region. Note that the system will turn ON first for about 78% of trajectories (Q1); the rest will turn OFF first–see asymptotes of dashed lines in (b). A third dashed line in (b) corresponds to the cumulative distribution until the time of the first switch decision (Q2). The solid lines correspond to the probabilities for the first time the system will reach the ON (or OFF) region (Q3). The dotted lines correspond to the times until the system completes a trajectory in which it begins at $s_1 = s_2 = 0$, it turns ON (or OFF), and finally turns OFF (or ON) (Q4). Two methods have been used in these analyses: the lines correspond to the original FSP solution, and circle markers denote the reduced order model solutions (See also Table 1).

Table 1: Comparison of the computational efficiency of computing switch rates of a Stochastic Gene Toggle Switch using three techniques to solve the chemical master equation: the original Finite State Projection approach (FSP), the FSP approach with balance truncation reduction (FSP-RED), and 10^4 runs of the SSA.

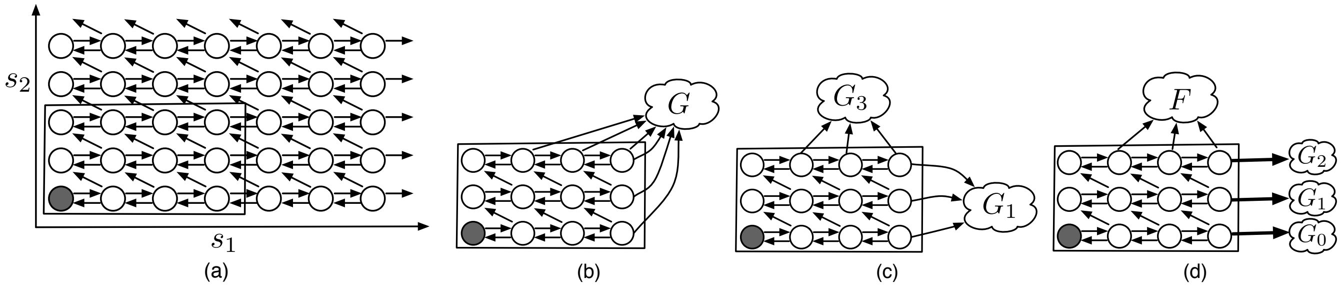

Figure 1:

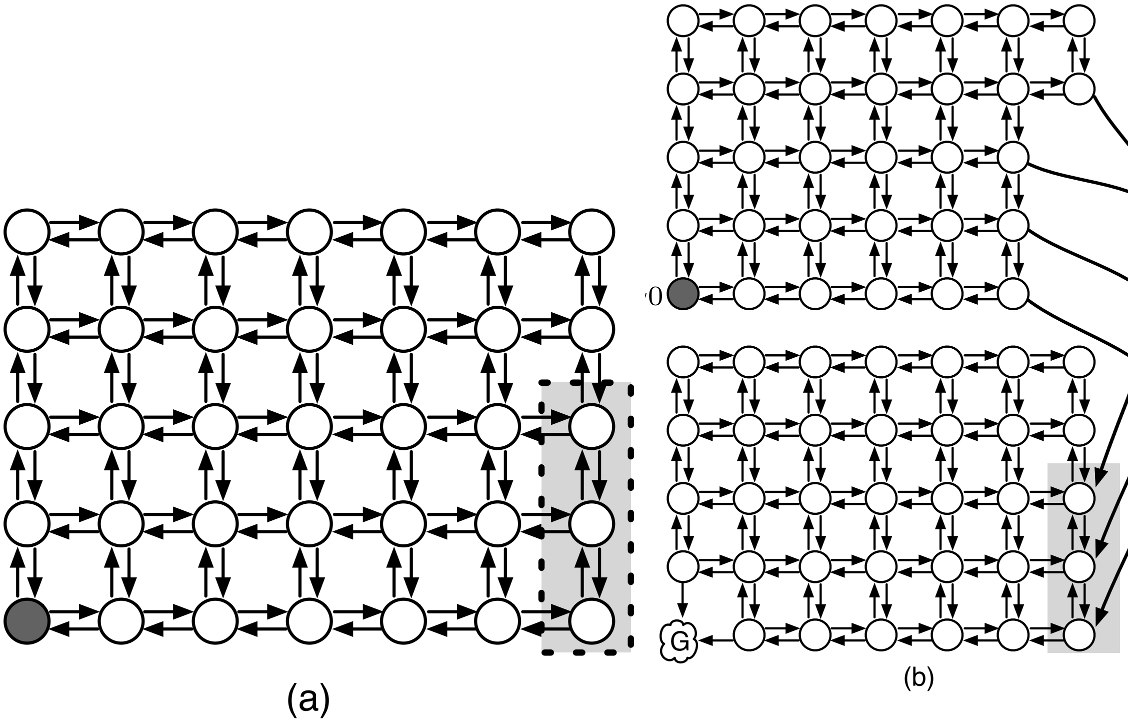

Figure 2:

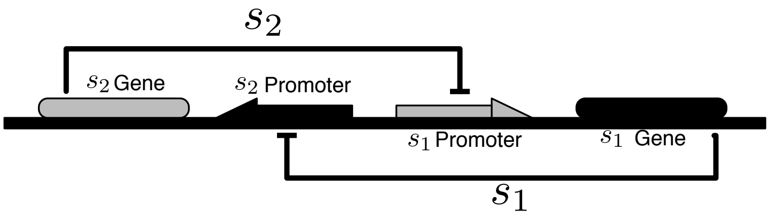

Figure 3:

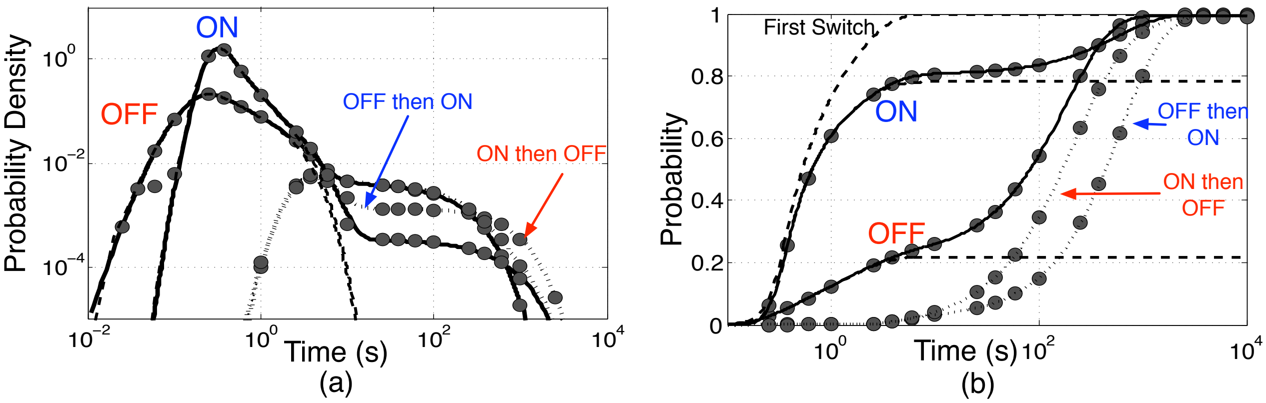
Probability Density
ON
OFF
OFF then ON
ON then OFF
10^0
10^{-2}
10^{-4}
10^{-2}
10^0
10^2
10^4
Time (s)
(a)
1
0.8
0.6
0.4
0.2
0
Probability
First Switch
ON
OFF
OFF then ON
ON then OFF
10^0
10^2
10^4
Time (s)
(b)

Figure 4:

Single Stage Trajectories

First Switch to OFF					
Method	J_{red}	J_{solve}	J_{total}[a]	t_{50}	% Error
FSP	-	31.0s	31.0s	81.952s	$< 2 \times 10^{-5}$
FSP-RED	111.8	1.85s	113.7s	81.952s	$< 4 \times 10^{-5}$
10^4 SSA	-	2068s	2068s	78.375s	≈ 4.3

First Switch to ON					
	J_{red}	J_{solve}	J_{total}	t_{50}	% Error
FSP	-	25.7s	25.7s	0.65655s	$< 1 \times 10^{-7}$
FSP-RED	133.5s	1.85s	135.3s	0.65656s	$< 8 \times 10^{-4}$
10^4 SSA	-	404.4s	404.4	0.65802s	≈ 0.22

Two Stage Trajectories

First Completion of OFF then ON trajectory					
	J_{red}	J_{solve}	J_{total}	t_{50}	% Error
FSP	-	46.9s	46.9s	434.969s	$< 3.5 \times 10^{-5}$
FSP-RED	222.0s	1.95s	224.0s	434.968s	$< 4.5 \times 10^{-3}$
10^4 SSA	-	3728s	3728s	441.394	≈ 1.5

First Completion of ON then OFF trajectory					
	J_{red}	J_{solve}	J_{total}	t_{50}	% Error
FSP	-	51.0s	51.0s	167.530s	$< 6 \times 10^{-7}$
FSP-RED	241.4s	1.98s	243.4s	167.939	≈ 0.24
10^4 SSA	-	3073s	3073	166.860	≈ 0.40

[a] All simulations have been performed in MATLAB version R2007a on a MacBook Pro with a 2.16 GHz Intel Core Duo processor and 2 GB of memory. All ODEs have been solved with MATLAB's stiff ODE solver *ode15s* with relative tolerance 10^{-8} and absolute tolerance of 10^{-20}.

Table 1: