

# The Finite State Projection Approach for the Analysis of Stochastic Noise in Gene Networks

Brian Munsky and Mustafa Khammash

**Abstract**—In order to capture important subcellular dynamics, researchers in computational biology have begun to turn to mesoscopic models in which molecular interactions at the gene level behave as discrete stochastic events. While the trajectories of such models cannot be described with deterministic expressions, the probability distributions of these trajectories can be described by the set of linear ordinary differential equations known as the chemical master equation (CME). Until recently, it has been believed that the CME could only be solved analytically in the most trivial of problems, and the CME has been analyzed almost exclusively with Kinetic Monte Carlo (KMC) algorithms. However, concepts from linear systems theory have enabled the Finite State Projection (FSP) approach and have significantly enhanced our ability to solve the CME without resorting to KMC simulations. In this paper we review the FSP approach and introduce a variety of systems theory based modifications and enhancements to the FSP algorithm. Notions such as observability, controllability and minimal realizations enable large reductions and increase efficiency with little to no loss in accuracy. Model reduction techniques based upon linear perturbation theory allow for the systematic projection of multiple time scale dynamics onto a slowly varying manifold of much smaller dimension. We also present a powerful new reduction approach, in which we perform computations on a small subset of configuration grid points and then interpolate to find the distribution on the full set. The power of the FSP and its various reduction approaches is illustrated on few important models of genetic regulatory networks.

## I. INTRODUCTION

The main aim of synthetic biology is to intelligently build or alter biological systems for useful purposes. This, of course, is no small task, and it requires enormous knowledge into the form and function of biological networks. While new experimental tools are successfully revealing the form of many gene regulatory networks, experiments alone may not be enough to gain a sufficient understanding of the function of these monstrously complex systems. For this reason, researchers are also beginning to demand clear quantitative models with which they hope to understand their measurements, guide future experiments, and provide another level of understanding for the medical and scientific community.

It would be very nice if all important intercellular dynamics could be described with deterministic models, just as it would be nice if every biological experiment yielded the

same results every time. If these were so, one could treat all important proteins and RNA molecules by continuous-valued concentrations that evolve over time according to non-linear ordinary differential equations (ODEs). Unfortunately, often this is not the case. Many important biochemical processes, especially those involving gene regulatory networks, occur on a very small scale, where important chemical species, such as DNA, RNA and key regulatory proteins, are restricted to only a few copies per cell. In this regime, a concentration description is meaningless, mass action kinetics are not valid, and discrete models are necessary. Also, when only a few copies exist of certain important chemical species, the system becomes dominated by intrinsic noise [1], [2], [3], and an individual molecular interaction, such as an inhibitor binding to a gene, may change the dynamics of the entire cell or organism.

For example, Gardner’s genetic toggle switch [4], shown in Fig. 1 and discussed in more detail below, is one such system where small populations and noise have great importance. The system is comprised of two promoters, each of whose product inhibits the other; if species  $s_1$  gains a slight edge it will shut off  $s_2$ , and vice versa. A deterministic model would suggest that the system would reach a steady state that depends only upon the initial condition. With noise, however, the system can not only reach different steady states, but even continue to oscillate between different states. The dynamics of such systems can no longer be usefully described by a single trajectory of the system through the state space; that trajectory may be only one of many wildly different possibilities. Instead, the system must be described by the probabilities that the system will have certain states at certain times. For discrete population chemically reacting systems, the evolution of this probability distribution is well understood to evolve according to the chemical master equation (CME) [5], [6].

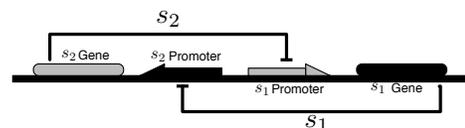


Fig. 1. Schematic representation of the Gardner’s Genetic Toggle Model [4]. The species  $s_1$  represses the gene that transcribes  $s_2$  and vice versa. For this model, deterministic and stochastic approaches yield highly contrasting results. In the deterministic model, the system reaches one of two steady state solutions ( $s_1 \gg s_2$  or  $s_2 \gg s_1$ ) the choice of which depends only upon the initial condition. In the stochastic model, the system oscillates between these two regions.

Brian Munsky is with the Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106-5070 [brianem@engr.ucsb.edu](mailto:brianem@engr.ucsb.edu)

Mustafa Khammash is with the Department of Mechanical Engineering, University of California, Santa Barbara, CA 93106-5070 [khammash@engr.ucsb.edu](mailto:khammash@engr.ucsb.edu)

For many systems the CME is an infinite set of linear ODEs that cannot be solved exactly. Most analyses of the CME have utilized kinetic Monte-Carlo algorithms such as the Stochastic Simulation Algorithm [7], [8] or one of its approximations [9], [10], [11], [12]. As these Monte Carlo methods provide only a single trajectory for each run, they must be reiterated many times to provide a statistical description of the system. We recently showed that a properly chosen finite state projection (FSP) approach can reduce the order of the CME to a finite number of ODEs. The result is an analytical approximation of the probability distribution at any instant in time [13]. By recasting the CME as a finite set of linear ODEs, the FSP paves the way for linear systems theory based analyses of stochastic biological networks, as we will show in this paper.

In the next section, we briefly review the mesoscopic description of chemical kinetics and the chemical master equation. In Section III we review the Finite State Projection (FSP) method, which yields an analytical solution to the CME. In Section IV, we review a few model reduction techniques which can greatly expand the usefulness of the FSP algorithm. These include several ideas from systems theory including concepts of controllability, observability, and time-scale separation. We also introduce an entirely new reduction scheme that is based upon assuming that the probability distribution can be approximated by a function that is piecewise linear in the configuration space and which evolves according to simple linear dynamics. In Section V, we use a couple important biochemical reactions in order to compare and contrast the FSP and its various reductions schemes. Finally, in Section VI we summarize our findings and make some concluding remarks to outline future directions for projection based approaches to solving the CME.

## II. THE MESOSCOPIC MODEL FOR CHEMICAL KINETICS

For a macroscopic model, chemicals are treated as concentrations that evolve according to deterministic ODEs. This is useless for some systems which contain a very small number of molecules. In a microscopic model, each individual molecule is tracked as it bounces around inside the cell. This can be computationally expensive even for a small number of non-reacting particles. The mesoscopic model of chemical kinetics lies somewhere in between—the system’s configuration is described by discrete populations,  $\mathbf{x}_i := [\xi_1 \ \xi_2 \ \dots \ \xi_N]_i^T \in \mathbb{N}^N$ , but the individual molecules are assumed to be well mixed, and the system is treated as a continuous time, discrete space Markov process. The probabilities of each of the configurations  $\{\mathbf{x}_1, \mathbf{x}_2, \dots\}$  make up the probability distribution state vector  $\mathbf{P}(t) := [p_1(t), p_2(t), \dots]^T$ , which evolves according to the possibly infinite dimensional set of linear ODEs known as the Chemical Master Equation (CME) [14], [6]:  $\dot{\mathbf{P}}(t) = \mathbf{A}(t)\mathbf{P}(t)$ . Until recently, the CME had not been directly solved except for the most trivial problems, and analyses have often conducted using kinetic Monte Carlo (KMC) algorithms.

The most popular KMC algorithm for mesoscopic chemical kinetics is Gillespie’s Stochastic Simulation Algorithm (SSA) [7], [8]. At each time,  $t$ , the process has a specific configuration,  $\mathbf{x}_i$ , and can transition to some other configuration  $\mathbf{x}_j = \mathbf{x}_i + \nu_\mu$ , where  $\nu_\mu$  is the stoichiometric vector of the  $\mu^{th}$  reaction. Each  $\mu^{th}$  reaction has an infinitesimal probability of occurring in the interval  $(t, t + dt]$ , and this probability is given by  $a_\mu(\mathbf{x}_i, t)dt$ , where  $a_\mu(\mathbf{x}_i, t)$  is known as the *propensity function*. If, in addition to well-mixedness, one assumes that the system has constant temperature and constant volume, then  $a_\mu$  does not depend upon time, and the time until the next reaction,  $\tau$ , is an exponentially distributed random variable with a mean equal to  $a_0^{-1}(\mathbf{x}_i) = \left(\sum_{\mu=1}^M a_\mu(\mathbf{x}_i)\right)^{-1}$ . At each KMC step  $\tau$  is generated from this exponential distribution, and one can choose which of the  $M$  reactions occurs according to the distribution defined by  $\left\{\frac{a_1(\mathbf{x}_i)}{a_0(\mathbf{x}_i)}, \dots, \frac{a_M(\mathbf{x}_i)}{a_0(\mathbf{x}_i)}\right\}$ . One then updates the time to  $t + \tau$  and the configuration to  $\mathbf{x}_i + \nu_\mu$  where  $\mu$  is the index of the chosen reaction. The process is continued until the final time of interest. This algorithm produces a very detailed trajectory for the mesoscopic model, and can be very computationally expensive for some systems. In these cases, one may give up some of the accuracy of the SSA for faster approximate MC schemes such as time-leaping methods [9], [10] and system-partitioning methods [11], [12].

These KMC algorithms do an excellent job of providing sample trajectories for Markov processes, and as such are indispensable in the study of gene regulatory networks. However, in terms of actually solving the CME to find probability distributions, these algorithms are very slow to converge and provide little in the way of accuracy guarantees.

## III. THE FINITE STATE PROJECTION (FSP) ALGORITHM

Recently, we proposed an analytical approach to solving the CME: the Finite State Projection (FSP) algorithm [13]. With this approach, one may systematically choose a projection of the CME, which satisfies any prespecified accuracy requirement. Once the finite state projection has been made, the problem of solving the infinite dimensional CME becomes one of solving a finite dimensional linear ODE. Because it is based upon linear systems theory, the FSP is ripe for the further application of modern control theory as we will review in this paper.

In the chemical master equation,  $\dot{\mathbf{P}}(t) = \mathbf{A}(t)\mathbf{P}(t)$ , the infinitesimal generator  $\mathbf{A}(t)$  is comprised of the propensity functions for transitions from one configuration to another and is defined by the reactions and the enumeration of the configuration space. Like all generator matrices,  $\mathbf{A}(t)$  has no negative off-diagonal elements, and all of  $\mathbf{A}$ ’s columns sum to zero.

For a system of finite possible configurations, the solution of the CME is easily solved using matrix exponentials or ODE solvers. However, the CME is not so easily solved when  $\mathbf{P}(t)$  has an extremely large or infinite dimension. In this case, a projection may be made to achieve an arbitrarily accurate approximation. For this we need some

convenient notation. First, we use index sets of the form  $J = \{j_1, j_2, j_3, \dots\}$  as follows: we let  $\mathbf{v}_J$  denote the subvector of  $\mathbf{v}$  chosen according to  $J$ , and we let  $\mathbf{A}_{IJ}$  denote the submatrix of  $\mathbf{A}$  whose rows and columns have been chosen according to  $I$  and  $J$ , respectively. For example, if  $I = \{3, 1\}$  and  $J = \{3, 2\}$ , then:

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

For convenience, we will use the notation  $\mathbf{A}_J := \mathbf{A}_{JJ}$ . We will also use an embedding operator,  $\mathcal{D}_J\{\cdot\}$  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  $J$ . Finally, we will use the notation  $|\cdot|$  to denote a vector norm, and  $\|\cdot\|$  to denote an induced matrix norm. With this notation, we can restate the theorems from [13] as follows, where we have extended Theorem 2.2 to include the case of time-varying  $\mathbf{A}(t)$ .

**Theorem 2.1**<sup>1</sup> If every off-diagonal element of  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is non-negative, then for any index set  $J$ ,

$$[\exp(\mathbf{A})]_J \geq \exp([\mathbf{A}]_J) \geq \mathbf{0}. \quad (1)$$

**Theorem 2.2** Consider any distribution which evolves according to the linear ODE:  $\dot{\mathbf{P}}(t) = \mathbf{A}(t)\mathbf{P}(t)$ . Let  $\mathcal{F}_J(t_2, t_1)$  be the state transition matrix from time  $t_1$  to time  $t_2$  of the  $J$ -indexed finite state projection system  $\dot{\mathbf{P}}_{FSP} = \mathbf{A}_J\mathbf{P}_{FSP}$ .

If for  $\varepsilon > 0$ , and  $t_f \geq 0$ ,

$$\mathbf{1}^T \mathcal{F}_J(t_f, 0) \mathbf{P}_J(0) \geq 1 - \varepsilon, \quad (2)$$

then

$$\mathcal{F}_J(t_f, 0) \mathbf{P}_J(0) \leq \mathbf{P}_J(t_f), \text{ and} \quad (3)$$

$$|\mathcal{F}_J(t_f, 0) \mathbf{P}_J(0) - \mathbf{P}_J(t_f)|_1 \leq \varepsilon \quad (4)$$

*Proof:* We begin by proving (3). Let  $J'$  denote the complement of  $J$ . The evolution of the probability distribution on the set  $J$  is governed by:

$$\dot{\mathbf{P}}_J(t) = \mathbf{A}_J(t)\mathbf{P}_J(t) + \mathbf{A}_{JJ'}(t)\mathbf{P}_{J'}(t), \quad (5)$$

where the submatrix  $\mathbf{A}_{JJ'}(t)$  is nonnegative for any generator  $\mathbf{A}(t)$ . The solution for (5) is

$$\mathbf{P}_J(t_f) = \mathcal{F}_J(t_f, 0) \mathbf{P}_J(0) + \int_0^{t_f} \mathcal{F}_J(t_f, \tau) \mathbf{A}_{JJ'}(\tau) \mathbf{P}_{J'}(\tau) d\tau$$

Since  $\mathbf{A}_{JJ'}(t)$ ,  $\mathbf{P}_{J'}(t)$ , and  $\mathcal{F}_J(t, \tau)$  are all nonnegative for  $t \geq \tau \geq 0$ , we obtain the inequality in (3).

Since all probability distributions are non-negative and sum to one we are assured that  $|\mathbf{P}_J(t_f)|_1 \leq 1$  and the condition (2) becomes:

$$|\mathcal{F}_J(t_f, 0) \mathbf{P}_J(0)|_1 \geq |\mathbf{P}_J(t_f)|_1 - \varepsilon, \quad (6)$$

Finally, applying (3) and rearranging terms yields (4) and completes the proof. ■

<sup>1</sup>For proof and additional details, see [13].

In the original form of [13], we assumed that  $\mathbf{A}$  did not vary in time, and the result is the same except that  $\mathcal{F}(t_2, t_1) = \exp(\mathbf{A}(t_2 - t_1))$ . Using this FSP theorem, we can use an algorithmic approach to add and remove states in the finite projection until we obtain an error,  $\varepsilon$ , that is less than a prespecified bound.

#### IV. SPEEDING UP THE FSP

In its most basic form, the FSP method is not feasible for every problem. For example, when there are many reacting species or when large excursions occur frequently in small periods of time, the number of ODEs required for the finite projection to meet a given accuracy requirement may be far too large. However, there are many additional tools available from systems theory that can help us to meet this challenge. As discussed in the introduction, kinetic Monte Carlo algorithms for the solution of the CME benefit from two approximation schemes: time leaping algorithms [9], [10] and system partitioning methods [11], [12]. In order for these approximations to apply to the KMC solutions to the CME, it is reasonable to expect that the same or similar approximations will also apply to the CME itself and therefore the FSP. In Subsection IV-A we look at reductions to the CME which rely upon separating the full time interval into a series of shorter steps and then using the FSP to compute how the distribution evolves from one step to the next. In Subsection IV-B, we show various possible partitioning schemes in which we approximate the FSP solution by projecting it onto a lower dimensional space. Each of these reduction techniques are summarized below in the context of time-invariant systems, but many can also be extended to time-varying systems.

##### A. Multiple Time Interval FSP Approaches

The CME is a linear system describing the evolution of a nonnegative probability distribution. As such, the finite state projection solution of the CME must obey the principle of superposition. The concept of super-positioning provides the advantage that rather than compute the matrix exponential of a very large matrix for a given initial probability distribution vector (pdv), one can break the pdv into separate parts and compute the exponentials for a series of smaller, less expensive matrices. For some systems, through clever ordering, very large regions of the configuration space can be considered with relatively few exponential computations.

A second advantage is that superposition supports a time stepping algorithm for the solution of the infinite dimensional ODE [15]. Beginning at a certain point in the configuration space at time  $t_0$ , one can approximate the probability distribution vector at  $t_1$ . One can then remove unimportant portions of the approximate pdv, partition it into smaller subvectors and use these subvectors as initial conditions for the next time step. For time invariant systems, one can achieve significant computational savings by caching and then reusing previous matrix exponential computations from one time step to the next. This approach can further benefit by running a few SSA runs before the beginning the FSP

analysis—in addition to helping determine the set  $J_0$  for the first time step, a few SSA runs can also provide an estimate of how large each time step should be in the FSP formulation. By carefully choosing the projections, the error accrued at each step can be automatically controlled to guarantee a final error bound in the full pdv. Recently, Burrage and coauthors presented a new version of the FSP algorithm [16] that takes advantage of multiple time intervals but does not use the properties of superposition to reduce the computational costs.

### B. Partitioning and Projecting the CME

Several readily available tools from linear systems theory facilitate lower order approximations of larger systems and promise significant reductions in computational cost. For each of these reductions, we begin with the CME in the form:  $\dot{\mathbf{P}}(t) = \mathbf{A}\mathbf{P}(t)$ , and we seek to approximate the vector  $\mathbf{P}(t) \in \mathbb{R}^n$  (here,  $n$  may or may not be finite) as some linear transformation of a lower dimensional vector,  $\mathbf{q}(t) \in \mathbb{R}^{m \leq n}$ .

It is important to note that the original FSP itself is one such projection in which the elements of  $\mathbf{q}(t)$  correspond to  $\mathbf{P}_J(t)$ . There are many other possible choices for our projection, as we will discuss in the following subsections. In Subsection IV-B.1, we use concepts of controllability and observability taken from modern control theory to obtain a minimal basis set for the space in which the solution to the FSP evolves. Alternatively, as previously presented in [17], [18], Subsection IV-B.2 shows how one may project the system onto a space spanned by an appropriately chosen set of eigenvectors. If the time is relatively long, we can use the eigenvectors corresponding to the slow eigenvalues. Conversely, if the time is relatively short, we can choose the eigenvectors that correspond to the fast eigenvalues. In Subsection IV-B.3 we introduce another Finite Element like approach in which we perform computations on a small set of “discretization” points and interpolate the remaining points of the distribution.

#### 1) Observability, controllability, and minimal realizations:

Because the FSP approach formulates the CME as a finite dimensional problem, it opens the analysis to linear systems theory based model reductions as we first explored in [19] and now expand upon here. We can pose the initial value CME problem as an equivalent impulse response problem:  $\dot{\mathbf{P}}(t) = \mathbf{A}\mathbf{P}(t) + \mathbf{b}\delta(t)$ , where  $\mathbf{b} = \mathbf{P}(0)$ .

Suppose that we wish only to compute the output  $\mathbf{y}(t) = \mathbf{C}\mathbf{P}(t)$ . For example,  $\mathbf{y}$  may include statistical information such as means or variances, or could correspond to the probability of certain important portions of the configuration set. The resulting problem now takes on a familiar form:

$$\begin{aligned} \dot{\mathbf{P}}(t) &= \mathbf{A}\mathbf{P}(t) + \mathbf{b}\delta(t); \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{P}(t). \end{aligned} \quad (7)$$

For systems constrained to a finite configuration set, or for systems that have been projected onto a finite configuration set using the FSP, this standard representation is open to a host of computational tools. We define a configuration  $\mathbf{x}_i$  as *reachable* from another configuration  $\mathbf{x}_j$  if there is

a possible trajectory that will take the system from  $\mathbf{x}_i$  to  $\mathbf{x}_j$ ; otherwise  $\mathbf{x}_i$  is *unreachable* from  $\mathbf{x}_j$ . A configuration  $\mathbf{x}_i$  is defined to be *observable* by  $\mathbf{x}_j$  if  $\mathbf{x}_j$  is reachable from  $\mathbf{x}_i$ ; otherwise  $\mathbf{x}_i$  is *unobservable* by  $\mathbf{x}_j$ . We assume *a priori* knowledge of which configurations are reachable from the initial condition and which are observable from the output. We use the index  $X$  to denote the full configuration set, and we will use the index sets  $R$ ,  $R'$ ,  $O$ , and  $O'$  to define the reachable, unreachable, observable and unobservable configuration sets, respectively. We then chose an index set  $J \in RO$  and its complement on the reachable observable set,  $J' = (RO - J) = \{j'_1, j'_2, \dots\}$ . With these index sets, we can define the the projection system,

$$\mathbf{q} = \begin{bmatrix} \mathbf{q}_1(t) \\ q_2(t) \end{bmatrix} \approx \begin{bmatrix} \mathbf{P}_J(t) \\ \mathbf{1}_{RO'}^T \mathbf{P}_{RO'}(t) \end{bmatrix}$$

Applying this projection to the CME yields the ODE:

$$\dot{\mathbf{q}}(t) = \begin{bmatrix} \mathbf{A}_{Jt_f} & 0 \\ \mathbf{1}^T \mathbf{A}_{RO'Jt_f} & 0 \end{bmatrix} \mathbf{q}(t). \quad (8)$$

This projection is the basis for the Observability Aggregated FSP (OAFSP) algorithm as presented in [19] in which we showed that if the solution to (8) satisfies

$$\mathbf{1}^T \begin{bmatrix} \mathbf{q}_1(t) \\ q_2(t) \end{bmatrix} \geq 1 - \varepsilon,$$

then

$$\left\| \begin{bmatrix} \mathbf{P}_J(t) \\ \mathbf{1}^T \mathbf{P}_{RO'}(t) \\ \mathbf{P}_{J'}(t) \end{bmatrix} - \begin{bmatrix} \mathbf{q}_1(t) \\ q_2(t) \\ \mathbf{0} \end{bmatrix} \right\|_1 \leq \varepsilon. \quad (9)$$

As an example, Fig. 2(top) illustrates a two dimensional state lattice for a two chemical ( $a$  and  $b$ ) reacting system. The system begins with an initial configuration  $\mathbf{x}_u$  at time  $t = 0$ , and we are interested in calculating the probability that the system has configuration,  $\mathbf{x}_y$ , at the time  $t = t_f \geq 0$ . The configuration set can be separated into three disjoint subsets: the unreachable region,  $\mathbf{X}_{R'}$ ; the unobservable region,  $\mathbf{X}_{O'}$ ; and the reachable/observable region  $\mathbf{X}_{RO}$ . Each of these three subsets contains a countable number of points. Using the OAFSP, we first remove the unreachable configuration subset from the system and aggregate the unobservable configuration subset to a single point, as shown in Fig. 2(bottom left). We then project  $\mathbf{X}_{RO}$  onto a finite configuration subset  $\mathbf{X}_J$ ; in Fig. 2 this finite subset is enclosed by the black polygon. The projected system is shown in Fig. 2(bottom right), where the subsets  $\mathbf{X}_{J'}$  and  $\mathbf{X}_{RO'}$  have each been aggregated to a single point. Since the probability distribution on the projected system evolves on a finite configuration subset, its solution at any time can be computed using the matrix exponential function or by using a linear ODE solver.

The OAFSP approach, is similar but not identical to standard observability based model reductions. Here the index set  $RO$  defines the smallest euclidean basis set enclosing the space that can be reached by the initial condition and observed from the output. This is an easily constructed space, but it does not necessarily provide a *minimal* realization for the system.

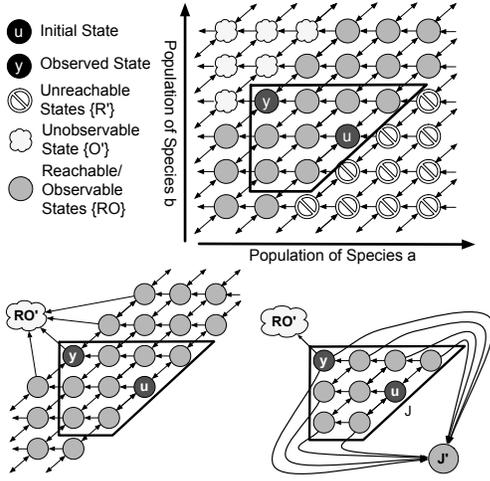


Fig. 2. Top: schematic of a two dimensional integer lattice representing the countable configuration set of a discrete valued Markov process. Each integer valued configuration point  $[a,b]$  is represented by a circle and the directionality of transitions between configurations are shown by the connecting arrows. Bottom: aggregation of the unobservable configuration subsets (left), and projection of the observable/reachable configuration subset onto a finite configuration subset:  $\mathbf{X}_J \in \mathbf{X}_{RO}$  (right).

For a larger reduction, we can use more traditional concepts of controllability and observability as follows. Since the solution to a finite  $N$ -dimensional linear ODE is given by the equation  $\mathbf{P}(t) = \exp(\mathbf{A}t)\mathbf{P}(0)$ , the Cayley-Hamilton theorem allows us to write this solution as

$$\mathbf{P}(t) = \sum_{i=0}^{N-1} \alpha_i(t) \mathbf{A}^i \mathbf{P}(0).$$

Therefore, the solution is constrained to the range of the observability matrix,

$$\{\mathbf{P}(0), \mathbf{A}\mathbf{P}(0), \mathbf{A}^2\mathbf{P}(0), \dots, \mathbf{A}^{k-1}\mathbf{P}(0)\},$$

for some  $k \leq N$ . In this case, we choose the  $k$ -dimensional vector  $\mathbf{q}$  to refer to the probability distribution in the transformed coordinates of the controllable subspace. Algorithmically, one can systematically increase  $k < N$  until the vector  $\mathbf{A}^k \mathbf{P}(0)$  can be written as a linear combination of the previous vectors  $\{\mathbf{P}(0), \mathbf{A}\mathbf{P}(0), \dots, \mathbf{A}^{k-1}\mathbf{P}(0)\}$ . This is essentially the Krylov subspace approach that has been taken by Burrage and co-authors in [16].

2) *Multiple time scale partitioning*: In many biological models, certain reactions occur much faster and more frequently than others. For KMC algorithms like the SSA, the majority of the simulated reactions correspond to those with large propensities. In the case of the CME or its projection, this separation of time scales results in numerical stiffness. There has been significant progress in developing approximate KMC algorithms to deal with these concerns [11], [12]. In each of these, the fast dynamics are essentially averaged, and the slow dynamics are simulated assuming the fast dynamics have instantaneously reached thermal equilibrium. We have previously shown that the FSP algorithm is also

amenable to time-partitioning approximation schemes that speed up computation at a small cost to the accuracy [17], [18]. In the original works, the time scale separation is carried out using perturbation theory. In the control community, perturbation methods have also had long use as described in [20]. In this paper we take a linear systems theory approach to such problems. In our analysis, we use the term “interconnected” to mean that the configurations form a non-separable Markov process. A finite dimensional interconnected system can readily be shown to have a simple eigenvalue at zero.

In the configuration space, some subsets of configuration points are often interconnected by fast reactions and separated from each other by slow reactions. One such example is the 4-configuration Markov process illustrated in Fig. 3(a). We assume that the fast reactions (solid lines) have propensities equal to one, and the slow reactions (dashed lines) have propensities equal to  $\varepsilon$ . The master equation for this particular process has the generator

$$\mathbf{A} = \begin{bmatrix} -r - \varepsilon & r & \varepsilon & 0 \\ r & -r - \varepsilon & 0 & \varepsilon \\ \varepsilon & 0 & -r - \varepsilon & r \\ 0 & \varepsilon & r & -r - \varepsilon \end{bmatrix}.$$

If one groups together the fast interconnected configurations (possibly requiring a permutation of the configuration set), one can separate the system into fast and slow parts:  $\mathbf{A} = \mathbf{H} + \varepsilon\mathbf{G}$ , where  $\mathbf{H}$  is block diagonal with each block representing a fast interconnected configuration set. For the schematic in Fig. 3 this separation gives us

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_2 \end{bmatrix} = \begin{bmatrix} -r & r & 0 & 0 \\ r & -r & 0 & 0 \\ 0 & 0 & -r & r \\ 0 & 0 & r & -r \end{bmatrix},$$

and

$$\varepsilon\mathbf{G} = \begin{bmatrix} -\varepsilon & 0 & \varepsilon & 0 \\ 0 & -\varepsilon & 0 & \varepsilon \\ \varepsilon & 0 & -\varepsilon & 0 \\ 0 & \varepsilon & 0 & -\varepsilon \end{bmatrix}.$$

It is easily seen that each  $\mathbf{H}_i$  is the generator matrix for the  $i^{\text{th}}$  fast cluster, and  $\varepsilon\mathbf{G}$  is the generator matrix of the reactions that take the system from one cluster to another.

For an  $N$  dimensional finite state projection with  $m$  fast interconnected configuration sets, the master equation can be written

$$\dot{\mathbf{P}}(t) = (\mathbf{H} + \varepsilon\mathbf{G})\mathbf{P}(t) \quad (10)$$

$\mathbf{H}$  can be written  $\mathbf{H} = \text{diag}\{\mathbf{H}_1, \mathbf{H}_2, \dots, \mathbf{H}_m\}$ . Because they are generators, each  $\mathbf{H}_i$  has a single eigenvalue equal to zero, and its corresponding left and right eigenvectors are  $\mathbf{u}_i = \mathbf{1}^T$  and  $\mathbf{v}_i$ , respectively. We define the following matrices.

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1 & 0 & \dots \\ 0 & \mathbf{u}_2 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}, \text{ and } \mathbf{V} = \begin{bmatrix} \mathbf{v}_1 & 0 & \dots \\ 0 & \mathbf{v}_2 & \dots \\ \vdots & \vdots & \ddots \end{bmatrix}.$$

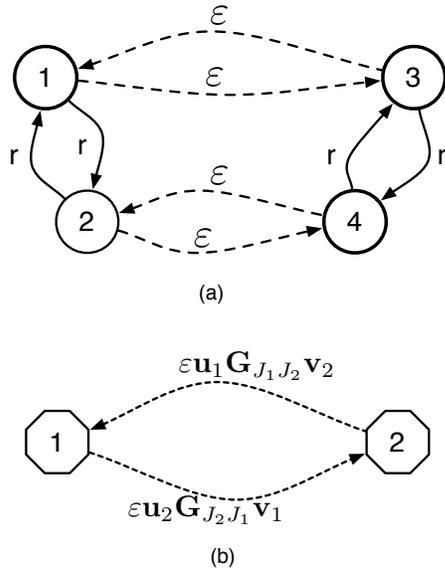


Fig. 3. (a) A four configuration Markov process that undergoes reactions on two different time scales. For  $r \gg \varepsilon$ , the fast reactions are represented by solid lines, and slow reactions are represented by dashed lines. There are two sets of strongly connected configurations:  $J_1 = \{1, 2\}$  and  $J_2 = \{3, 4\}$ . The fast reactions form two infinitesimal generators,  $\mathbf{H}_1$  and  $\mathbf{H}_2$ , and the slow reactions form a single generator for the whole system,  $\varepsilon\mathbf{G}$ . (b) In the reduced model, each fast interconnected set becomes a single configuration. The strength of the slow reactions from the  $i^{\text{th}}$  to the  $j^{\text{th}}$  set is given by  $\varepsilon\mathbf{u}_j\mathbf{G}_{J_jJ_i}\mathbf{v}_i$ , where  $\mathbf{u}_i$  and  $\mathbf{v}_i$  are the left and right zero-eigenvectors of  $\mathbf{H}_i$ , and  $\mathbf{G}_{J_jJ_i}$  is the submatrix of  $\mathbf{G}$  with columns corresponding to  $J_i$  and rows corresponding to  $J_j$ .

Let  $\mathbf{S} = [\mathbf{V} \quad \mathbf{R}]$  be a square matrix in which the columns of  $\mathbf{R}$  are the remaining  $N - m$  right eigenvectors of  $\mathbf{H}$ . The inverse of  $\mathbf{S}$  is given by  $\mathbf{S}^{-1} = [\mathbf{U}^T \quad \mathbf{L}^T]^T$  such that we have the following similarity transformation for  $\mathbf{H}$ :

$$\mathbf{S}^{-1}\mathbf{H}\mathbf{S} = \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda} \end{bmatrix}, \quad \mathbf{\Lambda} = \text{diag}(\lambda_{m+1}, \dots, \lambda_N).$$

where the first  $m$  diagonal elements correspond to the zero eigenvalues of the  $\mathbf{H}_i$  blocks. We assume that non-zero eigenvalues of  $\mathbf{H}$  are ordered so that  $0 > \text{Re}\{\lambda_{m+1}\} \geq \text{Re}\{\lambda_{m+2}\}, \dots \geq \text{Re}\{\lambda_N\}$ . Applying the coordinate transformation  $[\mathbf{y}_1^T(t) \quad \mathbf{y}_2^T(t)]^T = \mathbf{S}^{-1}\mathbf{P}(t)$ , (10) becomes:

$$\begin{bmatrix} \dot{\mathbf{y}}_1(t) \\ \dot{\mathbf{y}}_2(t) \end{bmatrix} = \begin{bmatrix} \varepsilon\mathbf{UGV} & \varepsilon\mathbf{UGR} \\ \varepsilon\mathbf{LGV} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \mathbf{y}_1(t) \\ \mathbf{y}_2(t) \end{bmatrix}, \quad (11)$$

where we have defined the matrix  $\mathbf{Q} = \mathbf{\Lambda} + \varepsilon\mathbf{LGR}$ .

There are two important observations to make regarding this transformed system. First, the matrix  $\mathbf{UGV}$  is itself a generator for a Markov process in that it satisfies the two sufficient conditions: (i) its columns sum to zero, and (ii) its off-diagonal elements are non-negative. To show that, note that  $\mathbf{1}^T\mathbf{U} = \mathbf{1}^T$  and therefore  $\mathbf{1}^T\mathbf{UG} = \mathbf{1}^T\mathbf{G} = \mathbf{0}$ . Furthermore,

$$[\mathbf{UGV}]_{ij} = \mathbf{u}_i\mathbf{G}_{J_iJ_j}\mathbf{v}_j,$$

where  $\mathbf{u}_i$  and  $\mathbf{v}_j$  are non-negative for any  $(i, j)$  and the submatrix  $\mathbf{G}_{J_iJ_j}$  is non-negative for any  $i \neq j$ . Hence the

off-diagonal elements of  $\mathbf{UGV}$  are indeed non-negative. The second observation that one can make is that for  $\varepsilon \ll |\text{Re}\{\lambda_{m+1}\}|$ , linear perturbation theory assures us that the matrix  $\mathbf{Q}$  is Hurwitz, and its eigenvalues are close to  $\{\lambda_{m+1}, \lambda_{m+2}, \dots, \lambda_N\}$ . In particular if we will let  $\tilde{\lambda}$  denote the real part of the least stable eigenvalue of  $\mathbf{Q}$ , we know that  $\tilde{\lambda} \approx \text{Re}\{\lambda_{m+1}\}$ .

With these observations in mind, we now examine the forced dynamics of  $\mathbf{y}_2(t)$ :

$$\dot{\mathbf{y}}_2(t) = \varepsilon\mathbf{LGV}\mathbf{y}_1(t) + \mathbf{Q}\mathbf{y}_2(t),$$

which has a solution comprised of a zero-state and a zero-input response:

$$\mathbf{y}_2(t) = \mathbf{y}_2^{zs}(t) + \mathbf{y}_2^{zi}(t).$$

Because  $\mathbf{Q}$  is Hurwitz, with eigenvalues all having real parts less than or equal to  $\tilde{\lambda}$ , the zero-input response,  $\mathbf{y}_2^{zi}(t)$ , is bounded by the exponentially decaying expression. Therefore, there exists a constant  $K_1$  such that

$$|\mathbf{y}_2^{zi}(t)|_1 \leq K_1 \exp(\tilde{\lambda}t), \quad \forall t \geq 0.$$

By the definition of our transformation  $|\mathbf{y}_1(t)|_1 = |\mathbf{UP}(t)|_1 = 1$ , and  $|\mathbf{LGV}\mathbf{y}_1(t)|_1$  is bounded. Since  $\mathbf{Q}$  is Hurwitz and the input is  $O(\varepsilon)$ , we are guaranteed that the zero-state solution,  $\mathbf{y}_2^{zs}(t)$  satisfies

$$|\mathbf{y}_2^{zs}(t)|_1 = O(\varepsilon), \quad \forall t \geq 0.$$

Combining the two solutions, we have the following bounds on  $\mathbf{y}_2(t)$

$$|\mathbf{y}_2(t)|_1 \leq K_1 \exp(\tilde{\lambda}t) + O(\varepsilon), \quad (12)$$

for all times  $t \geq 0$ .

The forced dynamics of  $\mathbf{y}_1(t)$  given by the

$$\dot{\mathbf{y}}_1(t) = \varepsilon\mathbf{UGV}\mathbf{y}_1(t) + \varepsilon\mathbf{UGR}\mathbf{y}_2(t),$$

has a solution at the chosen final time  $t_f$ :

$$\begin{aligned} \mathbf{y}_1(t_f) &= \exp(\varepsilon\mathbf{UGV}t_f)\mathbf{y}_1(0) \\ &+ \varepsilon \int_0^{t_f} \exp(\varepsilon\mathbf{UGV}(t_f - \tau))\mathbf{UGR}\mathbf{y}_2(\tau)d\tau. \end{aligned}$$

Note that since  $\mathbf{UGV}$  is a infinitesimal generator of a Markov process, every column of  $\exp(\mathbf{UGV}t)$  has a sum of exactly one for any  $t \geq 0$ , and  $\|\exp(\varepsilon\mathbf{UGV}(t - \tau))\|_1 = 1$  for all  $\varepsilon \geq 0$  and  $t \geq \tau$ . Therefore

$$|\mathbf{y}_1(t_f) - \exp(\varepsilon\mathbf{UGV}t_f)\mathbf{y}_1(0)|_1 \leq \varepsilon \int_0^{t_f} \|\mathbf{UGR}\mathbf{y}_2(\tau)\|_1 d\tau.$$

Combining this with (12) and defining the constant  $K_2 = K_1 \|\mathbf{UGR}\|_1$ , one obtains the following bound on the error of  $\mathbf{y}_1$  at  $t = t_f$ :

$$\begin{aligned} &|\mathbf{y}_1(t_f) - \exp(\varepsilon\mathbf{UGV}t_f)\mathbf{y}_1(0)|_1 \\ &\leq \varepsilon \int_0^{t_f} K_2 \exp(\tilde{\lambda}\tau) + O(\varepsilon) d\tau \\ &\leq \varepsilon K_2 \frac{1}{|\tilde{\lambda}|} + t_f O(\varepsilon^2). \end{aligned}$$

Therefore, for any fixed  $t_f \geq 0$ ,

$$|\mathbf{y}_1(t_f) - \exp(\varepsilon \mathbf{U} \mathbf{G} \mathbf{V} t_f) \mathbf{y}_1(0)|_1 = O(\varepsilon). \quad (13)$$

Combining (12) and (13), we have the following bounds on our approximation error:

$$\left\| \begin{bmatrix} \mathbf{y}_1(t_f) \\ \mathbf{y}_2(t_f) \end{bmatrix} - \begin{bmatrix} \exp(\varepsilon \mathbf{U} \mathbf{G} \mathbf{V} t_f) \mathbf{y}_1(0) \\ \mathbf{0} \end{bmatrix} \right\|_1 \leq K_1 \exp(\tilde{\lambda} t_f) + O(\varepsilon).$$

Substituting the initial condition,

$$\begin{bmatrix} \mathbf{y}_1(0) \\ \mathbf{y}_2(0) \end{bmatrix} = \mathbf{S}^{-1} \mathbf{P}(0) = \begin{bmatrix} \mathbf{U} \mathbf{P}(0) \\ \mathbf{L} \mathbf{P}(0) \end{bmatrix},$$

and performing the reverse similarity transformation,  $\mathbf{P}(t_f) = \mathbf{V} \mathbf{y}_1(t_f) + \mathbf{R} \mathbf{y}_2(t_f)$ , yields:

$$|\mathbf{P}(t_f) - \mathbf{V} \exp(\varepsilon \mathbf{U} \mathbf{G} \mathbf{V} t_f) \mathbf{U} \mathbf{P}(0)|_1 \leq K_1 \|\mathbf{R}\|_1 \|\mathbf{L} \mathbf{P}(0)\|_1 \exp(\tilde{\lambda} t_f) + O(\varepsilon).$$

Thus, this reduced model differs from the full system by at most an exponentially decreasing transient term plus a term of order  $\varepsilon$ .

In the toy example in Fig. 3, the blocks  $\mathbf{H}_1$  and  $\mathbf{H}_2$  were identical, with eigenvalues of zero and  $-2r$ . The left and right eigenvectors for the zero eigenvalue are  $\mathbf{u}_i = \begin{bmatrix} 1 & 1 \end{bmatrix}$  and  $\mathbf{v}_i^T = \begin{bmatrix} 1/2 & 1/2 \end{bmatrix}$ , respectively. The generator for the reduced system (as shown in Fig. 3(b)) is

$$\mathbf{U} \mathbf{G} \mathbf{V} = \begin{bmatrix} \mathbf{u}_1 \mathbf{G}_{J_1} \mathbf{v}_1 & \mathbf{u}_1 \mathbf{G}_{J_1 J_2} \mathbf{v}_2 \\ \mathbf{u}_2 \mathbf{G}_{J_2 J_1} \mathbf{v}_1 & \mathbf{u}_2 \mathbf{G}_{J_2} \mathbf{v}_2 \end{bmatrix},$$

where the the index set for the first and second blocks are  $J_1 = \{1, 2\}$  and  $J_2 = \{3, 4\}$ , respectively.

3) *Projection through Interpolation:* In the previous reductions schemes, knowledge of the system is exploited to provide smaller order models while maintaining known bounds on the error of the achieved approximation. In this subsection, we present a simpler reduction scheme, which can be very effective, but which no longer provides accuracy guarantees.

Suppose that one wishes to find a vector  $\mathbf{q}(t) \in \mathbb{R}^m$ , for some known interpolation operator  $\Phi \in \mathbb{R}^{n \times m}$  such that  $\Phi \mathbf{q}(t)$  provides an approximation of  $\mathbf{P}(t)$ . We assume that  $\mathbf{q}(t)$  has linear dynamics and can be expressed by  $\mathbf{q}(t) = \exp(\mathcal{A}t) \mathbf{q}(0)$  for some choice of  $\mathbf{q}(0)$  and  $\mathcal{A}$ , and we pose the following problem

$$\min_{\mathbf{q}(0), \mathcal{A}} |\mathbf{P}(t) - \Phi \exp(\mathcal{A}t) \mathbf{q}(0)|.$$

Performing a Taylor series expansion, the cost of the minimization becomes

$$|(\mathbf{P}(0) - \Phi \mathbf{q}(0)) + (\mathbf{A} \mathbf{P}(0) - \Phi \mathcal{A} \mathbf{q}(0)) t + O(t^2)|.$$

While this optimization problem is difficult to solve, one can easily minimize the first term in the least squares sense to yield  $\mathbf{q}(0) = \Phi^{-L} \mathbf{P}(0)$ , and minimize the second to give  $\mathcal{A} = \Phi^{-L} \mathbf{A} \Phi$ , where  $\Phi^{-L}$  is the left inverse of  $\Phi$ . Although this approximation is only guaranteed to retain accuracy for

short time intervals, in practice it often works well even for much longer intervals.

As an aside, with the correct choice of  $\Phi$ , all previous projections shown here can also be derived with this same formalism. The original FSP uses  $\Phi_{FSP} = \mathbf{I}_{XJ}$ ; the original OAFSP uses the projection  $\Phi_{OAFSP} = \begin{bmatrix} \mathbf{I}_{JX} & \mathcal{D}_{RO'} \{ \mathbf{1}_{RO'}^T \} \end{bmatrix}$ ; in the standard controllability or observability reduction, the columns of  $\Phi$  form a basis for the range of the minimal model; and in the multiple time scale reduction,  $\Phi$  is simply the matrix of right eigenvectors:  $\mathbf{V}$ . In the above minimization, problem one could also explore Krylov based methods of simultaneously choosing  $\Phi$  as well as  $\mathcal{A}$  and  $\mathbf{q}(0)$ , but these are left to be reported elsewhere.

To illustrate this interpolation based projection technique, we first consider a Markov process evolving along a one dimensional lattice such as that involving a single chemically reacting species,  $a$ . We begin with the full lattice, which we project to a finite subset as illustrated in Fig. 4(a,b). We choose a smaller subset of interpolation points as shown in Fig. 4(c). When the number of  $a$  molecules is small, we need greater precision and these points must be closer together, but when the number is larger, a coarser grid is more likely to suffice. Each two consecutive values  $q_i(t)$  and  $q_{i+1}(t)$  approximate the probability distribution at the points indexed by integers  $L_i$  and  $R_i$ , respectively. We assume that the probability distribution varies linearly between these two points; and we interpolate the distribution for any intervening point according to:

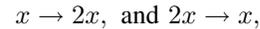
$$p_j(t) = \begin{bmatrix} \left(1 - \frac{j-L_i}{R_i-L_i}\right) & \frac{j-L_i}{R_i-L_i} \end{bmatrix} \begin{bmatrix} q_i(t) \\ q_{i+1}(t) \end{bmatrix}.$$

From this formulation, if we use  $m$  nodes to represent a distribution with  $n$  elements, we can obtain the projection operator,  $\Phi \in \mathbb{R}^{n \times m}$  as

$$[\Phi_{j_i}, \Phi_{j_{i+1}}] = \left[ \left(1 - \frac{j-L_i}{R_i-L_i}\right), \frac{j-L_i}{R_i-L_i} \right],$$

for all  $j$ .

As an example, suppose that the 1 dimension lattice in Fig. 4 represents a one species chemical reaction with the following two reactions



where the propensity of the first reaction is  $a_1(x) = 3x$ , the propensity of the second is  $a_2(x) = x(x-1)$ , and the initial condition is  $x_{t=0} = 1$ . By choosing to include only the first ten configurations of the system  $J = \{1, 2, \dots, 10\}$ , one can obtain the finite state projection  $\dot{\mathbf{P}}_J^{FSP}(t) = \mathbf{A}_J \mathbf{P}_J^{FSP}(t)$ , where the elements of  $\mathbf{A}$  are given by

$$A_{ij} = \begin{cases} -j^2 - 2j & \text{for } i = j \\ 3j & \text{for } i = j + 1 \\ j^2 - j & \text{for } i = j - 1 \end{cases}.$$

and the initial distributions is given as  $\mathbf{P}_J^{FSP}(0) = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]^T$ . One may choose to the interpolate the distribution among the points in the smaller 6 element

set  $\{1, 2, 4, 6, 8, 10\}$  which corresponds to the using the projection operator

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Applying the reduction yields

$$\mathcal{A} = \Phi^{-L} \mathbf{A} \Phi \approx \begin{bmatrix} -3.0000 & 2.0000 & 0 & 0 & 0 & 0 \\ 2.4853 & -5.2965 & 5.8865 & -2.8133 & 0.9546 & -0.3091 \\ -0.4264 & 3.4823 & -8.4323 & 14.0664 & -4.7729 & 1.5454 \\ 0.0732 & -0.5976 & 4.7073 & -16.5854 & 27.6829 & -8.9634 \\ -0.0126 & 0.1030 & -0.8116 & 5.4458 & -28.3246 & 52.2351 \\ 0.0025 & -0.0206 & 0.1623 & -1.0892 & 6.2649 & -79.4470 \end{bmatrix},$$

and  $\mathbf{q}(0) = \Phi^{-L} \mathbf{P}_J^{FSP}(0) = [1, 0, 0, 0, 0, 0]$ . Fig. 5 shows the probability distribution at  $t_f = 1s$  for the 10-state FSP solution,  $\mathbf{P}_J^{FSP}(t_f) = \exp(\mathbf{A}_J t_f) \mathbf{P}_J(0)$ , as well as the reduced 6-state solution,  $\mathbf{P}_J^I(t_f) = \Phi \exp(\mathcal{A} t_f) \mathbf{q}(0)$ . From the figure, one can see that the two solutions are in relatively good agreement.

For a lattice of two or more dimensions, the process is essentially the same, but the interpolation is slightly more involved and must be approached with more care. For the case of two species, each point  $(a_j, b_j)$  on the lattice is interpolated between the four corners of the mesh rectangle in which that point resides:  $(B_j, L_j)$ ,  $(B_j, R_j)$ ,  $(T_j, L_j)$ ,  $(T_j, R_j)$ , where  $(B_j, R_j)$  is the grid point to the bottom-right side of lattice point indexed by  $j$ ,  $(T_j, R_j)$  is the grid point lying to its top-right side, and so on. The probability at time  $t$  at each of these grid points is given by  $p_{BL(j)}(t)$ ,  $p_{BR(j)}(t)$ ,  $p_{TL(j)}(t)$ , and  $p_{TR(j)}(t)$ . In our approximation scheme, these variables will be approximated by  $q_{BL(j)}(t)$ ,  $q_{BR(j)}(t)$ ,  $q_{TL(j)}(t)$ , and  $q_{TR(j)}(t)$ , whose dynamics evolve in a lower dimensional space than the original system. To assign an approximation for  $p_j(t)$  where  $j$  is the index of lattice point surrounded by the mesh rectangle, we interpolate the four computed  $q$  variables, i.e.

$$q_j(t) = \mathcal{N}(j) \mathbf{q}_j(t) = \begin{bmatrix} (1-\alpha)(1-\beta) \\ \alpha(1-\beta) \\ (1-\alpha)\beta \\ \alpha\beta \end{bmatrix}^T \begin{bmatrix} q_{LB(j)}(t) \\ q_{RB(j)}(t) \\ q_{LT(j)}(t) \\ q_{RT(j)}(t) \end{bmatrix},$$

where

$$\alpha = \frac{a_j - L_j}{R_j - L_j},$$

$$\beta = \frac{b_j - T_j}{T_j - B_j}.$$

As in the one dimensional case, these Finite-Element-Method-like ‘‘shape functions,’’ and our chosen enumeration will directly provide the operator  $\Phi$ :

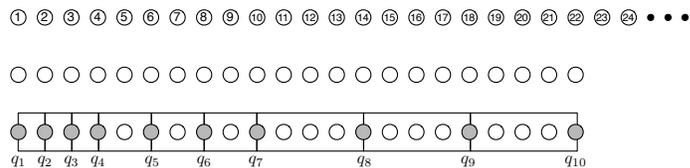


Fig. 4. One dimensional lattice Markov process. (a) The full infinite dimensional configuration space, (b) The finite state projection, (c) The finite lattice after it has been broken into 9 lattice elements with ten nodes (shaded). In this projection distributions at the unshaded points are interpolated from the distributions approximated at the nodes.

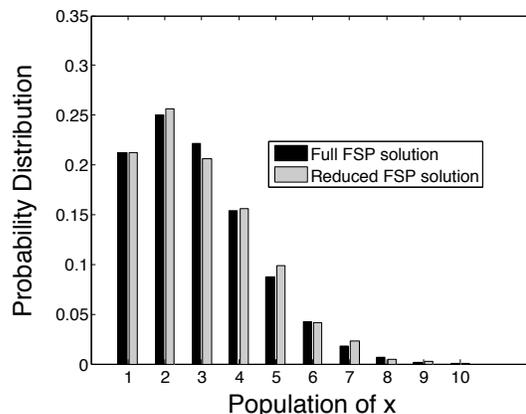


Fig. 5. The probability distribution for a simple one species chemical reaction as computed using the original FSP solution scheme as well as an interpolation-based reduction of the FSP. The distribution is computed at time  $t_f = 1s$ .

$$[\Phi_{j,LB(j)}, \Phi_{j,RB(j)}, \Phi_{j,LT(j)}, \Phi_{j,RT(j)}] = \mathcal{N}(j), \quad \forall j$$

In the next section, we will illustrate this reduction method on a few example gene regulatory networks.

## V. EXAMPLES

In this section we will illustrate some of the above analytical solutions to the chemical master equation on a few simple genetic regulatory networks. For each model, we compare and contrast the various methods and make some observations as to when each method is suitable.

### A. The Genetic Toggle Model

One of the most important genetic regulatory problems is that of stochastic switching. Two identical cells within the same environment and with the same initial conditions may express wildly different phenotypes; a few such examples include the *pap* (pili) regulatory switch in *E. coli* [21] as well as cell fate decisions in developing organisms. Alternatively, a cell may switch from one state to another as has been shown in previous stochastic models of the lysis-lysogeny decision of phage lambda [22]. Here, we consider a simple stochastic version of the genetic toggle system constructed and presented by Gardner, Cantor and Collins [4]. Fig. 1

illustrates this genetic regulatory system, which is comprised of two promoters each of whose products inhibits the other promoter. The signals of the network are the populations of the two repressors,  $s_1$  and  $s_2$ . We assume that these repressors react according to the simple production and degradation reactions:



where the degradation rates (left arrows) of  $s_1$  and  $s_2$  are  $\delta_1$  and  $\delta_2$ , respectively, and the synthesis rates (right arrows) of  $s_1$  and  $s_2$  depend upon the populations  $[s_2]$  and  $[s_1]$ , respectively, and are given by:

$$\alpha([s_2]) = \frac{\alpha_1}{1 + [s_2]^\beta}, \text{ and } \beta([s_1]) = \frac{\alpha_2}{1 + [s_1]^\gamma},$$

respectively.

First we consider the following set of parameters:

$$\delta_1 = \delta_2 = \gamma = 1, \alpha_1 = 25, \alpha_2 = 30, \beta = 1, \quad (14)$$

and we begin with an initial condition of zero for both species  $s_1$  and  $s_2$ . Three methods have been used to find the probability distribution at the time  $t_f = 10^4 s$ : (1) the finite state projection method (FSP), (2) the interpolation-reduced FSP (FSP-I), and (3) the stochastic simulation algorithm (SSA). Other methods have been considered, but an initial examination of the system presents no clear separation between time scales, so the slow manifold based reductions (either for the FSP or for KMC algorithms) have not been considered. Furthermore, with the entire distribution as the output, every configuration will eventually be attained. Therefore, the system is already minimal, and the reductions in Subsection IV-B.1 is helpful only over very short time intervals. To evaluate the accuracy of each method, we find the maximum error in the computed distribution. To evaluate the efficiency of each method, we consider two costs:  $J_{ODE}$ : the time required to compute the solution after the system reduction, and  $J_{total}$  the total time required to find and solve the reduced system. Each of these costs can be important in different situations:  $J_{total}$  is the stand-alone cost of solving this problem only once, and  $J_{ODE}$  represents the repetitive cost of solving the system as part of a larger more complex problem. With these metrics accuracy and efficiency of these different methods is discussed below and summarized in Table I.

**Method 1: FSP.** In order to use the original FSP method, one must first choose a configuration subset on which to perform the projection. Fig. 6 illustrates one such set chosen to include all configurations such that  $[s_1] \leq 64$ ,  $[s_2] \leq 88$ , and  $[s_1][s_2] \leq 220$ . For this configuration subset, the finite state projection of the CME is comprised of 1014 configurations, which took about 8 seconds to solve. Fig. 7(a) provides a contour map of the distribution for the full FSP approach for this first parameter set; Fig. 8(a,b) show the same probability distributions for the populations of  $s_1$  and  $s_2$ , respectively. In terms of accuracy, the full FSP implementation yielded a maximum error less than  $5.3 \times 10^{-5}$ .

**Method 2: FSP-I.** Using the methodology in Subsection IV-B.3, one can project the finite state system from the previous method onto a grid defined by integers distributed as follows. The first 8 are separated by one point:  $\{0, 1, \dots, 7\}$ ; the next 8 are separated by two points:  $\{8, 12, \dots, 22\}$ ; the next 8 by four points:  $\{24, 28, \dots, 52\}$ ; and the remaining points are separated by eight points up until the maximum value is reached. Each of these grid points is illustrated in Fig. 6 by a single dot. Figures 7(b) and 8(a,b) show the distribution contours as computed using the interpolation-based model reduction approach. From these figures, one can see that there is very little observable difference between the full FSP results and the interpolation-reduced FSP results. However, the interpolation-based approach required solving about a third as many ODEs and took less than a quarter of the time to complete, including the model reduction. The interpolation-based reduction had a maximum error of about  $2.9 \times 10^{-4}$ .

**Method 3: SSA.** For comparison with a typical Monte Carlo algorithm, the SSA [7] has also been run. After  $10^3$  simulations of the SSA, the total computational time was almost two hours, and the maximum error was about 30 times greater than that of the other methods. While approximations to the SSA, such as  $\tau$  leaping, may significantly speed up the computational time, they can do little to improve the accuracy of solution.

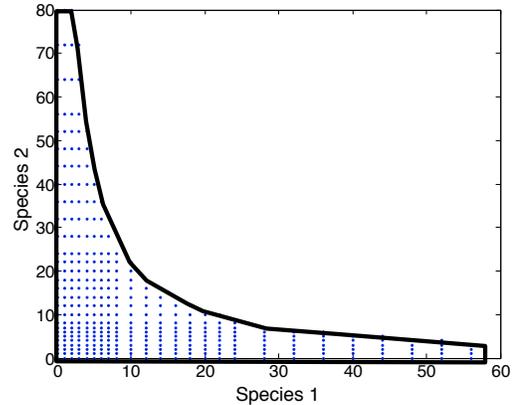


Fig. 6. Projection used for the genetic toggle model for parameter set (14). Here the finite state projection is first applied to remove the unlikely configurations and reduce the system to a set of 1014 ODEs, then the distribution of the remaining configurations is projected to a lower dimensional space for a set of 353 ODEs. The probability density is approximated assuming linear dynamics for the distribution at each point, and the full distribution is approximated via interpolation.

**Different Parameter Sets.** To extend the comparison of the three methods, two additional parameter sets have been considered:

$$\delta_1 = \delta_2 = \gamma = 1, \alpha_1 = 80, \alpha_2 = 100, \beta = 1, \quad (15)$$

and

$$\delta_1 = \delta_2 = \gamma = 1, \alpha_1 = 100, \alpha_2 = 25, \beta = 2.5, \quad (16)$$

TABLE I

COMPARISON OF THE COMPUTATIONAL EFFICIENCY AND ACCURACY OF THREE DIFFERENT SOLUTIONS OF THE CHEMICAL MASTER EQUATION FOR THE STOCHASTIC GENETIC TOGGLE MODEL. FOR OUR ANALYSIS,

WE BEGIN AT A KNOWN INITIAL CONDITION OF  $s_1 = s_2 = 0$  AND COMPUTE THE DISTRIBUTIONS AT  $t_f = 10^4 s$ . TWO COMPUTATIONAL COSTS ARE GIVEN:  $J_{ODE}$ —THE TIME REQUIRED TO SOLVE THE REDUCED SYSTEM OF ODES AND  $J_{total}$ —THE TOTAL TIME REQUIRED TO REDUCE AND SOLVE THE SYSTEM. SEE ALSO FIGS. 7-9.

For Parameters (14) and final time $t_f = 10^4 s$				
Method	Matrix Size	$J_{solve}^a$	$J_{total}$	$\infty$ -norm Error
FSP	1014	7.27s	7.41s	$\leq 5.3 \times 10^{-5}$
FSP-I	353	0.89s	1.40s	$\approx 2.9 \times 10^{-4}$
SSA ( $10^3$ )	-	-	6920s	$\approx 8.9 \times 10^{-3}$
SSA ( $10^4$ )	-	-	$7.1 \times 10^4 s$	$\approx 3.4 \times 10^{-3}$

For Parameters (15) and final time $t_f = 10^4 s$				
Method	Matrix Size	$J_{solve}$	$J_{total}$	$\infty$ -norm Error
FSP	3340	287s	288s	$< 6.4 \times 10^{-5}$
FSP-I	665	7.56s	8.75s	$\approx 8.1 \times 10^{-5}$
SSA ( $10^4$ )	-	-	$2.2 \times 10^5 s$	$\approx 2.9 \times 10^{-3}$

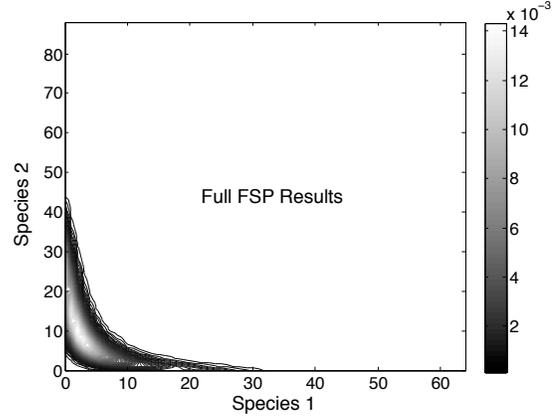
For Parameters (16) and final time $t_f = 10^4 s$				
Method	Matrix Size	$J_{solve}$	$J_{total}$	$\infty$ -norm Error
FSP	2404	93.2s	93.8s	$< 6.5 \times 10^{-6}$
FSP-I	556	4.09s	5.02s	$8.5 \times 10^{-4}$
SSA ( $10^4$ )	-	-	$1.7 \times 10^5 s$	$5.6 \times 10^{-3}$

<sup>a</sup>All computations have been performed in Matlab 7.2 on a Dual 2 GHz PowerPC G5

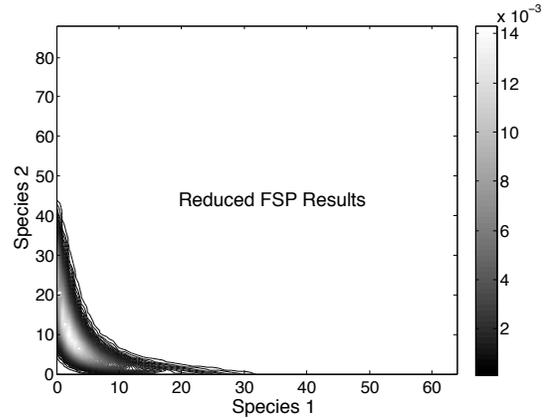
which are more computationally difficult to solve (for all methods) because more reactions occur, and the system tends to reach a larger portion of the configuration set. For parameter set (15), the chosen FSP solution includes all configurations such that  $[s_1] \leq 120$ ,  $[s_2] \leq 200$  and  $[s_1][s_2] \leq 700$ . For the original FSP algorithm this requires solving an 3340<sup>th</sup> order ODE and takes 288s to compute, but with the grid described for the previous set of parameters, the system is reduced to 665 dimensions and took only eight seconds to compute. Comparable improvements were also found for the third set of parameters for which the FSP solution includes every configuration such that  $[s_1] \leq 176$ ,  $[s_2] \leq 96$  and  $[s_1][s_2] \leq 500$ . Fig. 9 shows the probability distribution of species  $s_2$  for at the time  $10^4 s$  for both parameter sets (15) and (16) as computed with the FSP and the interpolation reduced FSP methods, and Table I summarizes the accuracy and efficiency of the same implementations. Once again there is very good agreement between the two solutions. For parameter sets (15) and (16), single runs of the SSA took 20.8 and 17.4 seconds, respectively. At these rates,  $10^4$  simulations take two days for each set yet and yield ten times worse convergence than any of the other methods.

### B. The Toy Heat Shock Problem

Biological systems have evolved many intricate mechanisms to deal with the frequent changes that occur in complex environments. One particular such system that has received a lot of recent attention is the cellular heat shock response in



(a)



(b)

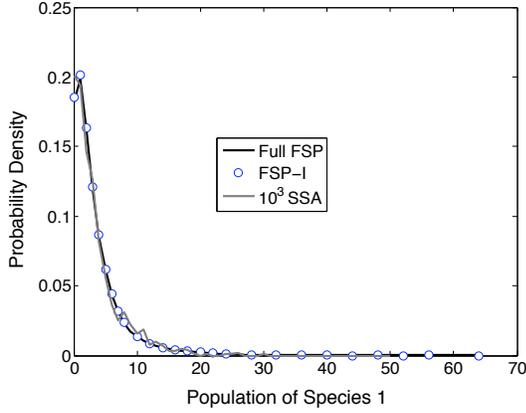
Fig. 7. Contour plots of the probability distributions of the reachable configurations of the Genetic toggle regulatory network with parameters in (14). We begin with a known initial condition of  $s_1 = s_2 = 0$  and compute the distributions at  $t_f = 10^4 s$ . Computations have been made using two analytical solution techniques: (a) The full, original FSP implementation. (b) The interpolation-based reduction of the FSP.

*E. coli*. At higher than normal temperatures, cellular proteins often fold incorrectly, and are no longer able to perform their functions. In order to survive, the cell avoids this outcome, by producing molecular chaperones and proteases, which refold denatured proteins and degrade irreversibly aggregated proteins. At the heart of the heat shock response mechanism in *E. coli* is the formation of the  $\sigma_{32}$ -RNAP complex [23], shown in Fig. 10. Here we use a simplified model for  $\sigma_{32}$ -RNAP formation to illustrate how one can combine the reduction methods in Subsections IV-B.2 and IV-B.3 to significantly increase the power of the FSP algorithm.

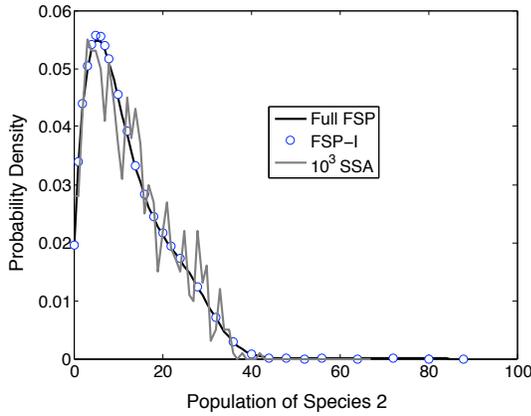
The simple Heat Shock regulatory mechanism is comprised of three reactions,



where  $s_1$ ,  $s_2$  and  $s_3$  correspond to the  $\sigma_{32}$ -DnaK complex, the  $\sigma_{32}$  heat shock regulator and the  $\sigma_{32}$ -RNAP complex, respectively. This model of the heat shock subsystem has



(a)



(b)

Fig. 8. An alternate representation of the probability distributions of the configurations reachable in the genetic toggle model with parameters in (14) (see also Fig. 7). (a) The probability density for species 1, (b) The probability density of species 2. The density as computed with the full FSP equations is represented by the smooth line, and the density as computed with the interpolation based reduction approach is represented with circles, and the density as computed with the  $10^4$  runs of the SSA is given by the jagged line.

been analyzed before using various computational methods including Monte Carlo implementations [12], [24], [17], [18], [15].

In the cell, the relative rates of the reactions are such that the reaction from  $s_2$  to  $s_1$  is by far the fastest, and  $\sigma_{32}$  molecules infrequently escape from DnaK long enough to form the  $\sigma_{32}$ -RNAP complex. The purpose of this mechanism is to strike a balance between fixing the damage produced by heat and saving the cell's resources, as a significant portion of cell energy is consumed when producing heat shock proteins. We use the following set of parameters values for the reaction rates:

$$c_1 = 10, \quad c_2 = 4 \times 10^4, \quad c_3 = 2, \\ s_1(0) = 2000, \quad s_2(0) = s_3(0) = 0.$$

With only the reactions above, the total number of  $\sigma_{32}$ -

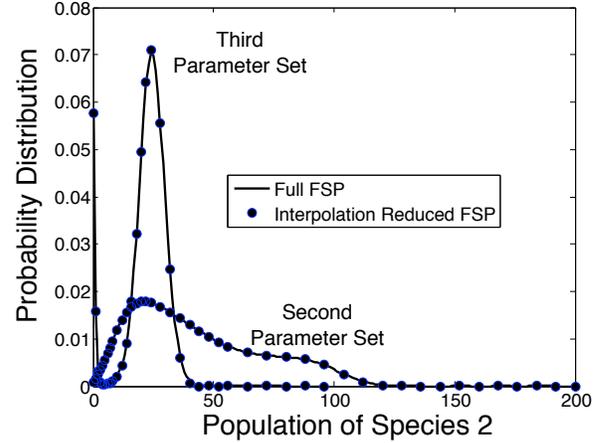


Fig. 9. The probability distribution of the population of  $s_2$  molecules in the genetic toggle model for parameter sets (15). The densities as computed with the full FSP equations are represented by the solid lines, and the densities computed with the interpolation based reduction approach are represented with circles.

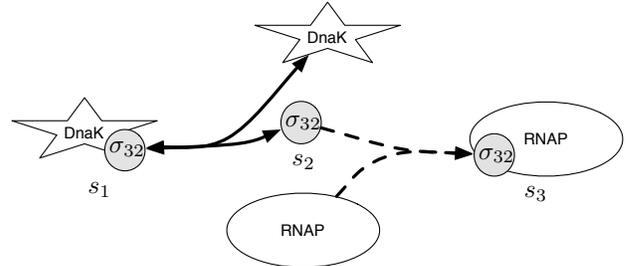


Fig. 10. Schematic representation of the Toy Heat shock model [23]. Species  $s_1$ ,  $s_2$  and  $s_3$  represent the  $\sigma_{32}$ -DnaK complex, free  $\sigma_{32}$  and the  $\sigma_{32}$ -RNAP complexes, respectively. In the model the free amounts of DnaK and RNAP are assumed constant and are lumped into the reaction rates. The solid arrows correspond to rapid binding and unbinding of DnaK and free  $\sigma_{32}$ . The dashed lines correspond to infrequent irreversible binding of RNAP and  $\sigma_{32}$ .

free or in compounds—is constant, so that  $s_1 + s_2 + s_3 = \text{const}$ . With this constraint the reachable states of this three species problem can be represented on a two dimensional lattice as shown in Fig. 11a. For our initial conditions there are 2,001,000 reachable states, and the full chemical master equation is too large to be solved exactly. Therefore we wish to find an approximate solution at ant time of interest. In particular, we wish to find the probability distribution of the population of  $s_3$  at time  $t_f = 300s$ . We have acquired this solution seven different methods: (1) the original finite state projection method (FSP), (2) the FSP with a multiple time step algorithm (FSP-MTS) [15], (3) the FSP with the slow manifold assumption (FSP-SM), (4) the FSP with the interpolation based reduction (FSP-I), (5) the FSP with first the slow manifold assumption and then an interpolation based reduction (FSP-SM/I), (6) the basic stochastic simulation algorithm (SSA), and (7) the SSA with the slow manifold approximation (SSA-SM). Fig. 12 shows

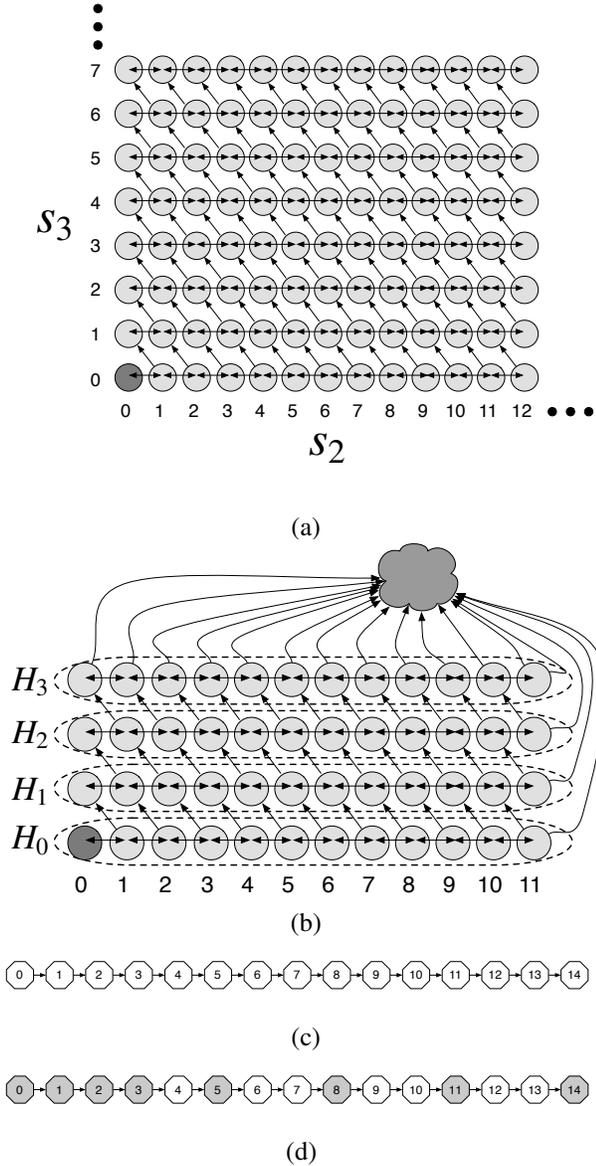


Fig. 11. (a) Two dimensional integer lattice representing possible configurations of the toy heat shock model. Here  $s_2$  and  $s_3$  are populations of free  $\sigma_{32}$  molecules and  $\sigma_{32}$ -RNAP compounds, respectively, while  $s_1$  is the population of  $\sigma_{32}$ -DnaK compounds. Reactions  $s_1 \rightleftharpoons s_2$ , are represented by bidirectional horizontal arrows and reactions  $s_2 \rightarrow s_3$  is represented with diagonal arrows. The total number of  $\sigma_{32}$  is constant, so the chemical state of the system is uniquely defined by  $s_2$  and  $s_3$  alone. (b) The same lattice after applying the finite state projection. Unlikely states have been aggregated into a single sink state. Each horizontal row of configurations is separated from the rest by the slow reaction 3 and then is used to form the fast block generator  $\mathbf{H}_i$ . (c) The slow manifold FSP that is found by projecting the dynamics of each fast interconnected set onto its equilibrium distribution. In this new system, each fast interconnected configuration subset is represented by a single node. (d) Applying the interpolation-based projection to the Slow Manifold FSP system. Here the shaded nodes are interpolation points on which the dynamics are now to be approximated.

the distribution as computed with each of these methods, and Table II summarizes the efficiency and accuracy of each method.

*Method 1: FSP.* For the original finite state projection approach, we include all configurations such that  $s_2 \leq 12$  and  $s_3 \leq 342$ . The resulting master equation that governs the evolution of the probability distribution on these configurations is a set of 4459 ODEs, which takes 750 seconds to solve and provides an error that is guaranteed to be less than  $3 \times 10^{-5}$ .

*Method 2: FSP-MTS.* By separating the time interval into 150 equal time intervals as described in Subsection IV-A and [15], one can significantly improve the efficiency of the FSP for this system. Rather than compute a 4459<sup>th</sup> order matrix exponential, one can instead acquire a solution by computing 70 different matrix exponentials each of which is 195<sup>th</sup> order or smaller. The total required time of this approach was 40.2 seconds, and the maximum error is guaranteed to be less than  $1.7 \times 10^{-4}$ .

*Method 3: FSP-SM.* In the Heat Shock model, the first two reactions,  $s_1 \rightarrow s_2$  and  $s_2 \rightarrow s_2$ , are much faster than the third. These fast reactions are used to define sets of fast interconnected configurations. If Fig. 11b, these sets are the horizontal rows of configurations. Using the slow manifold projection discussed above and in [17], [18], each fast interconnected configuration set can be collapsed to a single point to form the 1D lattice Markov chain shown in Fig. 11c. The reduced problem can now be solved as a system of only 343 ODEs, which takes only 0.94s to solve including reduction time.

*Method 4: FSP-I.* For an interpolation based solution to the FSP (see Subsection IV-B.3), we have chosen a grid where  $s_2$  is in the set  $\{0,1,2,3,4,5,6,7,8,10,12\}$  and  $s_3$  is in the set  $\{0,1,2,3,5,8,11,14,\{14+8n\}\}$ . This enables us to reduce the 4459<sup>th</sup> order FSP solution to a set of 539 ODEs. The reduced problem takes 6.1s to compute (including reduction time), and provides a solution with a maximum error of  $7.7 \times 10^{-4}$ . As in the previous example, our choice of interpolation points may not have provided the best possible reduction; better choices in terms of accuracy and efficiency may exist and are left as a topic for further research.

*Method 5: FSP-SM/I.* For a fifth solution scheme we have applied the interpolation based reduction of method 4 to the slow manifold model of method 3. From the reduced 1D lattice in Fig. 11c, we have chosen to include only the grid points where  $s_3$  is in the set  $\{0,1,2,3,5,8,11,14,\{14+8n\}\}$ . The resulting reduced system contains only 49 ODEs and takes less than 0.04 seconds to solve (after the reduction), but its results are nearly indistinguishable from the full system in that the maximum error is only  $8.2 \times 10^{-4}$ .

*Method 6: SSA.* We have also used Gillespie's stochastic simulation algorithm to generate the probability distribution. A single run of the SSA takes about 20 seconds to complete.  $10^4$  simulations would take over 50 hours and have not been computed for this paper.

*Method 7: SSA-SM.* As discussed above, the toy heat

shock model exhibits two significantly different time scales. Therefore, in addition to being an excellent candidate for the analytical FSP-SM method, the heat model is also amenable to Monte Carlo algorithms that utilize the same time scale separation reduction. We have applied one such approach where the SSA is simulated only on its slow manifold. This method, for which we use the acronym SSA-SM is very similar to the methods in [11], [12]. This SSA-SM takes only 0.1s per run and is 170 times faster than the original SSA, but it still requires many realizations before the solution to the CME will sufficiently converge. A set of  $10^3$  runs take 84 seconds to compute and yields a maximum error of about 0.012. By increasing the number of runs by a factor of one hundred, this implementation takes 100 times longer and yields an error less than ten times better as summarized in Table II.

*Longer Time Intervals.* If we were to consider longer time intervals for the toy heat shock model, the size of the projection would also need to increase. For an interval of 1200s, we need to include every configuration such that  $s_2 \leq 12$  and  $s_3 \leq 1022$ . This means that we must include over 13000 configurations with one ODE for each. While computing a system of that size is often possible using Krylov based solutions such as Roger Sidje’s *expokit* [25], it is beyond the capabilities of our chosen software (Matlab’s *expm(.)* routine), especially when there is significant numerical stiffness in the ODE’s. In this case the reduced solutions are not only beneficial, they are necessary. Fig. 12(b) shows the distribution of the number of  $s_3$  molecules as computed with the various FSP reduction schemes. In the interpolation-based FSP reduction, we now use a slightly coarser mesh in which we have chosen to include all configurations where  $s_3$  is in the set  $\{0,1,2,3,5,8,11,14,\{14+12n\}\}$ . Once again, all FSP based methods provide results that are virtually indistinguishable from the true solution, but they reach these results in far less time. In particular, the reduced model formed by projecting the system onto its slow manifold and then performing the interpolation-based projection results in a model of only 92 ODEs which takes less than one tenth of a second to solve.

## VI. SUMMARY AND CONCLUSIONS

Recently we showed that one can use the Finite State Projection (FSP) method to solve the chemical master equation (CME) and describe the dynamics of discrete stochastic chemical processes. In this paper, we have reviewed various systems-theory based approaches that can be used to further expand the applicability of the FSP. The multiple time step FSP approach, FSP-MTS, relies upon the linearity of the CME and the principle of super-position to solve the problem not as a single large system of ODEs but rather a set of smaller dimensional ODEs evolving over much shorter intervals [15]. The other three approaches explored in this work are based upon linear projections. The first of these, in Subsection IV-B.1 allows one to obtain a minimal realization by determining the configurations that are both

TABLE II  
COMPARISON OF THE COMPUTATIONAL EFFICIENCY AND ACCURACY OF VARIOUS SOLUTIONS OF THE CHEMICAL MASTER EQUATION FOR THE TOY HEAT SHOCK MODEL.

For final time $t_f = 300s$				
Method	Matrix Size	$J_{solve}$	$J_{total}$	$\infty$ -norm Error
FSP	4459	750s	750s	$< 3.0 \times 10^{-5}$
FSP-MTS	195 <sup>a</sup>	-	40.2s	$< 1.68 \times 10^{-4}$
FSP-SM	343	0.25s	0.94s	$\approx 5.1 \times 10^{-4}$
FSP-I	539	5.1s	6.1s	$\approx 7.7 \times 10^{-4}$
FSP-SM/I	49	0.04s	0.78s	$\approx 8.2 \times 10^{-4}$
$10^4$ SSA	Results would take more than 55 hours.			
$10^3$ SSA-SM	-	-	84.1s	$\approx 0.0116$
$10^4$ SSA-SM	-	-	925s	$\approx 3.4 \times 10^{-3}$
$10^5$ SSA-SM	-	-	9360s	$\approx 1.6 \times 10^{-3}$

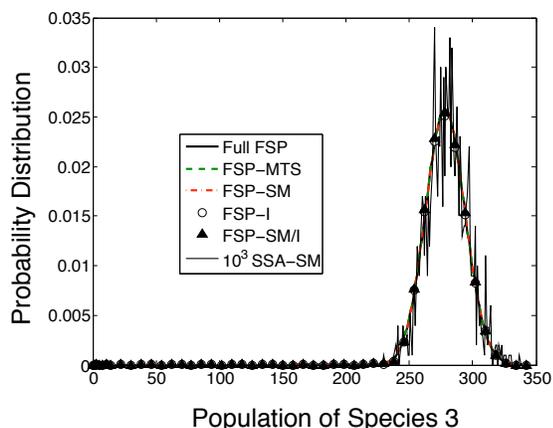
For final time $t_f = 1200s$				
Method	Matrix Size	$J_{ODE}$	$J_{total}$	$\infty$ -norm Error
FSP	13274	Exceeds machine capabilities		
FSP-MTS	325	-	253s	$< 1.2 \times 10^{-4}$
FSP-SM	1023	4.66s	10.66s	$\approx 1.2 \times 10^{-4}$
FSP-I	1012	40.5s	44.6s	$\approx 6.1 \times 10^{-4}$
FSP-SM/I	92	0.09s	6.19s	$\approx 5.7 \times 10^{-4}$
$10^4$ SSA	Results would take more than 180 hours.			
$10^3$ SSA-SM	-	-	272s	$\approx 9.9 \times 10^{-3}$
$10^4$ SSA-SM	-	-	3000s	$\approx 3.5 \times 10^{-3}$
$10^5$ SSA-SM	-	-	$2.99 \times 10^4s$	$\approx 1.2 \times 10^{-3}$

<sup>a</sup>For the multiple time-step FSP algorithm smaller matrices can be used at the cost of having to compute exponentials more than once.

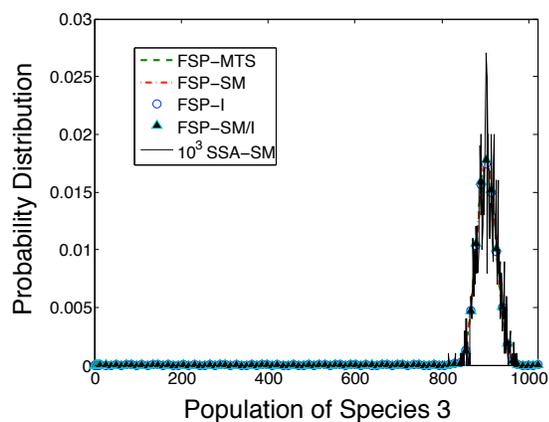
observable from the output as well as controllable from the initial condition. The second FSP-SM approach relies upon projecting the dynamics of the full FSP onto its slow manifold. Here we have presented a new systems theory based proof for this approach, which was originally presented in [17]. In the third, interpolation based FSP-I approach, one chooses a small subset of configuration points and assumes (i) that the probability distribution varies linearly between these points and (ii) that the resulting model has linear dynamics.

These and similar reduction approaches can yield great improvements over the original FSP as has been illustrated here through use of two genetic regulatory networks with various parameter choices: a stochastic version of the genetic toggle model and the toy heat shock model. In each case, reduced order models have been found that are far easier to solve and analyze yet which retain the important characteristics of the true system. Furthermore, all of the reduction approaches presented here can easily be used in conjunction with one another such that the greatest reductions can often be achieved by sequentially applying two or more methods.

In the work presented here, the interpolation based reductions have been chosen *a priori* with very limited knowledge of the system. Some choices of interpolation points are inherently better than others, but we have made no attempt to find an optimal set in this paper. Current studies suggest that large benefits can be gained from adaptive strategies for grid selection, and such approaches can even be combined with the multiple time step FSP algorithm of [15] so that different



(a)



(b)

Fig. 12. The probability distribution of the amount of the  $\sigma$ -TNAP complex formed at (a)  $t_f = 300s$  and (b)  $t_f = 1200s$  as computed using the toy heat shock model. Different analytical reductions have been applied to the chemical master equation, and each provides results that are virtually indistinguishable from the full FSP solution. See also Table II

grids could be used at different instances in time. Finally, we have limited ourselves to reduced models with linear dynamics; low order nonlinear models may be both more accurate and more efficient for many systems. While this work has shown that enormous reductions are feasible, many more may be possible with further application of systems theory on stochastic biological networks.

## VII. ACKNOWLEDGMENTS

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] M. McAdams and A. Arkin. Its a noisy business! *Tren. Gen.*, 15(2):65–69, 1999.  
 [2] M. Thattai and A. van Oudenaarden. Intrinsic noise in gene regulatory networks. *Proc. Natl. Acad. Sci.*, 98:8614–8619, 2001.

[3] M. Elowitz, A. Levine, E. Siggia, and P. Swain. Stochastic gene expression in a single cell. *Science*, 297:1183–1186, 2002.  
 [4] T. Gardner, C. Cantor, and J. Collins. Construction of a genetic toggle switch in escherichia coli. *Nature*, 403:339–242, 2000.  
 [5] van Kampen. *Stochastic Processes in Physics and Chemistry*. Elsevier, 3 edition, 2001.  
 [6] D. T. Gillespie. A rigorous derivation of the chemical master equation. *Physica A*, 188:404–425, 1992.  
 [7] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81(25):2340–2360, May 1977.  
 [8] M. A. Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem.*, 104:1876–1889, 2000.  
 [9] D. T. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 115(4):1716–1733, Jul. 2001.  
 [10] T. Tian and K. Burrage. Binomial leap methods for simulating stochastic chemical kinetics. *J. Chem. Phys.*, 121(21):10356–10364, Dec. 2004.  
 [11] C. V. Rao and A. P. Arkin. Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the gillespie algorithm. *J. Chem. Phys.*, 118(11):4999–5010, Mar. 2003.  
 [12] Y. Cao, D. Gillespie, and L. Petzold. The slow-scale stochastic simulation algorithm. *J. Chem. Phys.*, 122(014116), Jan. 2005.  
 [13] B. Munsky and M. Khammash. The finite state projection algorithm for the solution of the chemical master equation. *J. Chem. Phys.*, 124(044104), 2006.  
 [14] D. McQuarrie. Stochastic approach to chemical kinetics. *J. Applied Probability*, 4:413–478, 1967.  
 [15] B. Munsky and Khammash M. A multiple time interval finite state projection algorithm for the solution to the chemical master equation. *To appear in J. Comp. Phys.*, 2007.  
 [16] 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*, 2006.  
 [17] 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.*, 125(204104), Nov. 2006.  
 [18] B. Munsky, S. Peles, and M. Khammash. Stochastic analysis of gene regulatory networks using finite state projections and singular perturbation. *Proc. 26th American Control Conference (ACC)*, July 2007.  
 [19] B. Munsky and M. Khammash. A reduced model solution for the chemical master equation arising in stochastic analyses of biological networks. *Proc. 45th IEEE Conference on Decision and Control*, December 2006.  
 [20] P. Kokotovich, H. Khalil, and J. O’Reilly. *Singular Perturbation Methods in Control: Analysis and Design*. Academic Press, 1986.  
 [21] B. Munsky, A. Hernday, D. Low, and M. Khammash. Stochastic modeling of the pap-pili epigenetic switch. *Proc. FOSBE*, pages 145–148, August 2005.  
 [22] A. Arkin, J. Ross, and McAdams H. Stochastic kinetic analysis of developmental pathway bifurcation in phage  $\lambda$ -infected escherichia coli cells. *Genetics*, 149:1633–1648, 1998.  
 [23] H. El Samad, H. Kurata, J. Doyle, C. Gross, and Khammash M. Surviving heat shock: Control strategies for robustness and performance. *PNAS*, 102(8):27362741, 2005.  
 [24] H El Samad, M. Khammash, L. Petzold, and D. Gillespie. Stochastic modeling of gene regulatory networks. *Int. J. Robust Nonlin.*, 15:691–711, 2005.  
 [25] Roger B. Sidje. EXPOKIT: Software package for computing matrix exponentials. *ACM Transactions on Mathematical Software*, 24(1):130–156, March 1998.