A Multiple Time-Step Finite State Projection Algorithm for the Solution to the Chemical Master Equation

Brian Munsky and Mustafa Khammash

Department of Mechanical and Environmental Engineering
University of California, Santa Barbara

Abstract

At the mesoscopic scale, chemical processes have probability distributions that evolve according to an infinite set of linear ordinary differential equations known as the chemical master equation (CME). It is commonly believed that the CME cannot be solved except for the most trivial of cases, but recent work has raised questions regarding validity of this belief. For many cases, Finite State Projection (FSP) techniques [1] can reduce the order of the CME to a solvable system while retaining any prespecified error tolerance. Even when accuracy demands require a projection that is too large to be solve efficiently, the FSP retains the linearity of the CME, and is open to a host of additional model reductions and computational techniques. In this paper, we develop a new algorithm based upon the linearity property of super-positioning, and we illustrate the benefits of this algorithm on a simplified model of the heat shock mechanism in *E. coli*. The new algorithm retains the full accuracy of the original FSP algorithm, but with significantly increased efficiency and a greater range of applicability.

Key words: Chemical Master Equation, Stochastic Gene Regulatory Networks, Markov Processes

Email addresses: brianem@engr.ucsb.edu (Brian Munsky), khammash@engr.ucsb.edu (Mustafa Khammash).

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2006		**00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
A Multiple Time-Step Finite State Projection Algorithm for the Solution to the Chemical Master Equation	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Department of Electrical and Computer Engineering,University of California,Santa Barbara,CA,93106**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**		**20**	

1 Introduction

When studying processes at the mesoscopic level, researchers often assume that they evolve according to a continuous time, jump Markov process. In this regime, the system is best described not in terms of individual trajectories but in terms of how the system's probability distribution evolves. For chemical reactions, this distribution has been shown to evolve according to the infinite dimensional linear ordinary differential equation (ODE) known as the chemical master equation (CME) [2].

It is commonly believed that master equations, such as the CME, cannot be solved except in the simplest of circumstances. This belief has driven the computational research community to devise clever kinetic Monte Carlo (KMC) algorithms to simulate system dynamics. Gilespie's stochastic simulation algorithm (SSA) is one such algorithm developed specifically for the CME [3]. In the SSA each reaction is simulated using two pseudo-random numbers–the first tells when the next reaction will occur, and the second determines which reaction it will be. The SSA is considered exact in the sense that if one were to conduct the simulation an infinite number of times with an ideal random number generator, the collected statistical data would converge to the exact solution to the CME. Unfortunately, the convergence rate for any KMC routine is very slow; cutting the error in half requires four times the number of simulations. Since a single simulation may contain huge numbers of individual reactions, the SSA may take far too long to be of any use. Researchers have greatly improved efficiency of the SSA through various approximation schemes. Some of these approximations are made by separating the fast dynamics from the slow [4–8]. During short periods of time, the fast dynamics dominate, and the slow dynamics may be ignored. For long periods of time, one averages out the fast dynamics equilibrate in order to emphasize the slow dynamics. Other approximations are made by discretizing the time interval into τ leaps and approximating the dynamics over those subintervals [9–14]. Both approximation types, system partitioning and τ leaping, have been very successful in increasing the scope of problems to which KMC schemes such as the SSA may be reasonably applied.

We recently developed the Finite State Projection algorithm as an additional tool for the analysis of jump Markov processes [1]. Unlike KMC algorithms, the FSP algorithm solves the CME to any prespecified degree of accuracy without random number generation. The FSP approach is based upon linear systems theory and works by projecting the intractable infinite dimensional master equation onto a solvable finite dimensional space. Previous implementations of the FSP method have been very successful for some biologically inspired systems [1,15], but for many problems the projection remains too large to solve efficiently, and further model reductions are needed. While some

of these reductions come from fields such as modern controls theory [16], others are inspired from similar reductions applied to the KMC algorithms. For example, as we show in [17,18], we can reduce the FSP significantly using time-scale separation based system partitioning approaches. In the current work, we further advance the FSP by exploring how some tools of time discretization can also be used for the dramatic improvement of the FSP.

In the next section we provide some background on the basic FSP method. In section 3 we illustrate how linearity of the FSP allows us to efficiently deal with initial probability density vectors that contain many non-zero elements. In section 4 we provide a multiple time step version of the FSP and demonstrate its use with an example from the field of systems biology. Finally, in section 5, we conclude with remarks on the advantages of these approaches over the original FSP and outline a few directions for future work on this topic.

2 The Finite State Projection Method

Although the finite state projection (FSP) method presented here is valid for any continuous time, discrete space Markov process, we present it in the context of the chemical master equation.

We consider a system of N chemically reacting species. The non-negative populations of the N molecular species jointly define a unique configuration of the system, $\mathbf{x} := [\,\xi_1\ \xi_2\ \ldots\ \xi_N\,]^T$. By fixing a sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots$ of elements in $\mathbb{N}^N$ we can define the *ordered configuration set* as $\mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2, \ldots\ \ ]^T$. Let $p_i(t)$ denote the probability that the system will have the configuration $\mathbf{x}_i$ at time t. Under the assumptions that the system is continually well-mixed and kept at constant volume and temperature, one can show that the system evolves according to a discrete state, continuous time jump Markov process, whose distribution, $\mathbf{P} = [p_1, p_2, \ldots]^T$, evolves according to the linear ordinary differential equation known as the chemical master equation (CME) [19]:

$$\dot{\mathbf{P}}(t) = \mathbf{A}\mathbf{P}(t), \tag{1}$$

The infinitesimal generator matrix, $\mathbf{A}$, is defined by the reaction stoichiometries and propensities and the choice of the enumeration of $\mathbf{X}$. Like any generator matrix, $\mathbf{A}$ has the properties that all diagonal elements of are non-positive; all off-diagonal elements are non-negative; and all columns sum to zero. When the set $\mathbf{X}$ is finite dimensional, one can easily compute the solution $\mathbf{P}(t_f) = \exp(\mathbf{A}t_f)\mathbf{P}(0)$, but when the set $\mathbf{X}$ is infinite or extremely large, the corresponding solution is unclear or vastly difficult to compute. For these cases, we devised the Finite State Projection (FSP) method [1].

To best review the FSP method and present our current work, we must first introduce some convenient notation. Let $J = \{j_1, j_2, j_3, \ldots\}$ denote an index set to which the usual operations ($\bigcup$, $\bigcap$, etc.) and relations ($\subset$, $\subseteq$, etc.) apply. 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 $\mathbf{A}_J := \mathbf{A}_{JJ}$. We will also use 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. Finally, we will use the vector $\mathbf{e}^i$ to denote a vector with a one in the i^{th} location and zeros elsewhere. With this notation we can restate the following two theorems from [1]:

Theorem 1 *If $\boldsymbol{A} \in \mathbb{R}^{n \times n}$ has no negative off-diagonal elements, then for any index set, J,*

$$[exp\boldsymbol{A}]_J \geq exp(\boldsymbol{A}_J) \geq \mathbf{0}. \tag{2}$$

Theorem 2 *Consider any Markov process in which the distribution evolves according to the linear, time-invariant ODE:*

$$\dot{\mathbf{P}}(t) = \mathbf{A}\mathbf{P}(t).$$

If for some finite index set J, $\varepsilon > 0$, and $t_f \geq 0$,

$$||exp(\mathbf{A}_J t_f)\mathbf{P}_J(0)||_1 \geq 1 - \varepsilon, \tag{3}$$

then

$$\mathcal{D}_J\{exp(\mathbf{A}_J t_f)\mathbf{P}_J(0)\} \leq \mathbf{P}(t_f), \tag{4}$$

and

$$||\mathbf{P}(t_f) - \mathcal{D}_J\{exp(\mathbf{A}_J t_f)\mathbf{P}_J(0)\}||_1 \leq \varepsilon. \tag{5}$$

Using the current notation, these theorems have been strengthened slightly from their original form, but their proofs are easily found with slight modifi-

cation to those presented in [1]. Theorem 1 guarantees that as we add points to the finite configuration subset, the approximate solution monotonically increases, and Theorem 2 provides a certificate of how close the approximation is to the true solution. Together the two theorems suggest the FSP algorithm presented in Ref [1]:

The Original Finite State Projection Algorithm

Inputs Propensity functions and stoichiometry for all reactions.
Initial probability density vector, $\mathbf{P}(0)$.
Final time of interest, t_f.
Total amount of acceptable error, $\varepsilon > 0$.

Step 0 Choose an initial finite set of states, $\mathbf{X}_{J_o}$, for the FSP.
Initialize a counter, $i = 0$.

Step 1 Use propensity functions and stoichiometry to form $\mathbf{A}_{J_i}$.
Compute $\Gamma_{J_i} = ||\exp(\mathbf{A}_{J_i} t_f)\mathbf{P}_{J_i}(0)||_1$.

Step 2 If $\Gamma_{J_i} \geq 1 - \varepsilon$, **Stop**.
$\mathcal{D}_{J_i}\{\exp(\mathbf{A}_{J_i} t_f)\mathbf{P}_{J_i}(0)\}$ approximates $\mathbf{P}(t_f)$ to within a total error of ε.

Step 3 Add more states to find $\mathbf{X}_{J_{i+1}}$.
Increment i and return to **Step 1**.

For the basic FSP algorithm, if we wish to find a solution that is accurate to within ε at a time t_f, we must find a finite set of configurations such that the probability of ever leaving that set during the time interval $[0, t_f]$ is less than ε. For many problems, including the examples shown in [1,15], this set of configurations may be small enough that we can easily compute a single matrix exponential to approximate the solution to the CME. However, in other situations the configuration space required for a one matrix solution may be exorbitantly large. In this work we utilize the linearity and time invariance of the chemical master equation to address two such situations. First, in section 3 we extend the FSP to handle problems in which the initial probability distribution is supported over a large portion of the configuration space. Second, in section 4 we address the problem that occurs when non-equilibrium distributions tend to drift over time and cover large portions of the configuration space.

3 The FSP for Non-Sparse Initial Distributions

Although the original FSP method is valid for any initial probability distribution, all examples in previous work [1,15–18] begin with a specific known initial configuration; if the system begins in state k, the initial probability distribution for the CME was written, $p_i(0) = \delta_{ik}$, where δ_{ik} is the Kronecker delta. Suppose now that the initial distribution is given not by the Kronecker delta but by a vector with many non-zero elements. For example, suppose that the initial distribution is specified by the solution at the end of a previous time step. From Theorem 2, in order for the original FSP algorithm to converge, we must be able to find a set of states, $\mathbf{X}_J$, that satisfies the stopping criterion:

$$||\exp(\mathbf{A}_J t_f)\mathbf{P}_J(0)||_1 \geq (1-\varepsilon). \tag{6}$$

Since the sum of the FSP solution at t_f cannot exceed the sum of the truncated initial pdv, $\mathbf{P}_J(0)$, we must always include at least as many states in the FSP solution as is required such that $||\mathbf{P}_J(0)||_1 \geq 1-\varepsilon$. For a sparse pdv, such as that generated by δ_{ik}, this restriction on the size of the FSP solution is trivial: J need only include k. However, when the initial pdv has broad support, the size of the FSP solution may be much larger and therefore require the inefficient calculation of very high-dimensional matrix exponentials. Fortunately, one can use the property of superpositioning guaranteed by the linearity of the FSP to mitigate this concern and recover some computational efficiency as we can show in the following proposition.

Proposition 3 *Superposition of FSP Solutions*

Consider any Markov process in which the distribution evolves according to the linear ODE:

$$\dot{\boldsymbol{P}}(t) = \boldsymbol{A}\boldsymbol{P}(t).$$

Let $\gamma < 1$, $\delta < 1$ and $t_f \geq 0$. If there is an index set I such that:

$$||\boldsymbol{P}_I(0)||_1 \geq \gamma, \tag{7}$$

and if for every $i \in I$, there is a corresponding index set J_i containing i such that

$$\left|\left|\exp(\boldsymbol{A}_{J_i} t_f)\boldsymbol{e}^i_{J_i}\right|\right|_1 \geq \delta, \tag{8}$$

then,

$$\sum_{i\in I} p_i \mathcal{D}_{J_i}\left\{\exp(\boldsymbol{A}_{J_i} t_f)\boldsymbol{e}^i_{J_i}\right\} \leq \mathbf{P}(t_f), \tag{9}$$

and

$$\left|\left|\boldsymbol{P}(t_f) - \sum_{i\in I} p_i \mathcal{D}_{J_i}\left\{\exp(\boldsymbol{A}_{J_i} t_f)\boldsymbol{e}^i_{J_i}\right\}\right|\right|_1 \leq 1-\gamma\delta. \tag{10}$$

Proof. We begin by proving (9). If we define the index set $I_f = \bigcup_{i \in I} J_i$, then we have the relation,

$$\mathcal{D}_{I_f}\left\{\exp(\mathbf{A}_{I_f} t_f)\mathbf{P}_{I_f}(0)\right\} = \sum_{i \in I_f} p_i(0)\mathcal{D}_{I_f}\left\{\exp(\mathbf{A}_{I_f} t_f)\mathbf{e}^i_{I_f}\right\}, \tag{11}$$

Since $I \subseteq I_f$, we are guaranteed that

$$\mathcal{D}_{I_f}\left\{\exp(\mathbf{A}_{I_f} t_f)\mathbf{P}_{I_f}(0)\right\} \geq \sum_{i \in I} p_i(0)\mathcal{D}_{I_f}\left\{\exp(\mathbf{A}_{I_f} t_f)\mathbf{e}^i_{I_f}\right\}. \tag{12}$$

Furthermore, since for every i, $J_i \subseteq I_f$ and $p_i(0) \geq 0$, Theorem 1 guarantees that,

$$\mathcal{D}_{I_f}\left\{\exp(\mathbf{A}_{I_f} t_f)\mathbf{P}_{I_f}(0)\right\} \geq \sum_{i \in I} p_i(0)\mathcal{D}_{J_i}\left\{\exp(\mathbf{A}_{J_i} t_f)\mathbf{e}^i_{J_i}\right\}. \tag{13}$$

Furthermore, using the result from Theorem 1 that $\exp(\mathbf{A}_J t_f)$ is non-negative for any index set J, and applying conditions (7) and (8) yields

$$\begin{aligned}\left|\left|\mathcal{D}_{I_f}\left\{\exp(\mathbf{A}_{I_f} t_f)\mathbf{P}_{I_f}(0)\right\}\right|\right|_1 &\geq \left|\left|\sum_{i \in I} p_i(0)\mathcal{D}_{J_i}\left\{\exp(\mathbf{A}_{J_i} t_f)\mathbf{e}^i_{J_i}\right\}\right|\right|_1 \\ &\geq \delta \left|\left|\mathbf{P}_I(0)\right|\right|_1 \\ &\geq \delta\gamma. \end{aligned} \tag{14}$$

Theorem 2 tells us that

$$\mathcal{D}_{I_f}\left\{\exp(\mathbf{A}_{I_f} t_f)\mathbf{P}_{I_f}(0)\right\} \leq \mathbf{P}(t_f),$$

and then from Eqn (13) we show that

$$\sum_{i \in I_0} p_i(0)\mathcal{D}_{J_i}\left\{\exp(\mathbf{A}_{J_i} t_f)\mathbf{e}^i_{J_i}\right\} \leq \mathbf{P}(t_f), \tag{15}$$

which is Eqn. (9).

Combining the fact that $\left|\left|\mathbf{P}(t_f)\right|\right|_1 = 1$ and inequality (14) gives:

$$\left|\left|\sum_{i \in I} p_i(0)\mathcal{D}_{J_i}\left\{\exp(\mathbf{A}_{J_i} t_f)\mathbf{e}^i_{J_i}\right\}\right|\right|_1 \geq \left(\left|\left|\mathbf{P}(t_f)\right|\right|_1 - 1\right) + \delta\gamma. \tag{16}$$

Rearanging this result and applying 15 yields inequality (10)

$$\left|\left|\mathbf{P}(t_f) - \sum_{i \in I} p_i(0)\mathcal{D}_{J_i}\left\{\exp(\mathbf{A}_{J_i} t_f)\mathbf{e}^i_{J_i}\right\}\right|\right|_1 \leq 1 - \delta\gamma \tag{17}$$

and completes the proof.

The result of Proposition 3 now enables us to modify the above FSP algorithm to better handle situations in which the initial probability distribution is non-sparse. Before stating this new algorithm, however, we would first like to make a few notes to explain our choice of notation. First, although this algorithm can be useful on its own, we will see below that it is most effective as part of a multiple time step solution scheme. For this reason we will refer to the initial time as t_k and the final time as $t_{k+1} = t_k + \tau$. Second, the total error of the current approach is separated into two components, $\varepsilon = 1 - \delta\gamma$, where both γ and δ are numbers slightly less than 1 and will be considered as independent inputs to the algorithm. Here γ refers to the required sum of the truncated probability distribution at t_k, and δ refers to the relative accuracy requirement for the solution at t_{k+1} compared to the accuracy at t_k. Third, for added convenience we will use the notation $\mathbf{E}_i = \mathcal{D}_{J_i}\left\{\exp(\mathbf{A}_{J_i}\tau)\mathbf{e}^i_{J_i}\right\}$ to denote the J_i indexed FSP approximation of the distribution at t_{k+1} conditioned upon the i^{th} configuration at t_k. Any matrix exponential, $\exp(\mathbf{A}_{J_i}\tau)$ provides not only $\mathbf{E}_i$ but also approximations to $\mathbf{E}_j$ for every $j \in J_i$. Once we compute these matrix exponentials, we will store every $\mathbf{E}_j = \mathcal{D}_{J_i}\left\{\exp(\mathbf{A}_{J_i}\tau)\mathbf{e}^j_{J_i}\right\}$ and its corresponding index set $J_j = J_i$ that meets our accuracy requirement $||\mathbf{E}_j||_1 \geq \delta$. These are then reused to reduce the total number of matrix computations. With this notation, we can now state the following algorithm:

The FSP Algorithm for Non-Sparse Initial PDV's

Inputs Propensity functions and stoichiometry for all reactions.
Error Parameters, $0 \leq \gamma < 1$ and $0 \leq \delta < 1$.
Initial probability distribution, $\mathbf{P}(t_k)$, where $1 \geq ||\mathbf{P}(t_k)||_1 \geq \gamma$
Length of time interval, τ.

Step 0 Choose a finite set of states, $\mathbf{X}_{I_k}$ such that $||\mathbf{P}_{I_k}(0)||_1 \geq \gamma$.
Initialize a counter, i as the first element in I_k.
Initialize the FSP solution index set: $I_f = \{i\}$.
Initialize the FSP solution summation to zero: $\mathbf{P}^{FSP}_{I_f}(t_f) = 0$.

Step 1 If $\mathbf{E}_i$ has *not* already been calculated:
Use original FSP algorithm to find J_i and $\exp(\mathbf{A}_{J_i}\tau)$ such that $\left|\left|\exp(\mathbf{A}_{J_i}\tau)\mathbf{e}^i_{J_i}\right|\right|_1 \geq \delta$.
For every $j \in J_i$, if $\left|\left|\exp(\mathbf{A}_{J_i}t_f)\mathbf{e}^j_{J_i}\right|\right|_1 \geq \delta$, then record $\mathbf{E}_j = \mathcal{D}_{J_i}\left\{\exp(\mathbf{A}_{J_i}t_f)\mathbf{e}^j_{J_i}\right\}$ and $J_j = J_i$.

Step 2 Update the FSP solution index set: $I_f = I_f \bigcup J_i$.
Update the FSP solution summation: $\mathbf{P}^{FSP}_{I_f} = \mathbf{P}^{FSP}_{I_f} + p_i\mathbf{E}_i$.

Step 3 If i is the last element in I_0, **Stop**.
$\mathcal{D}_{I_f}\left\{\mathbf{P}_{I_f}^{FSP}(t_f)\right\}$ approximates $\mathbf{P}(t_f)$ to within $\varepsilon = 1 - \gamma\delta$.

Step 4 Increment i to the next element in I_0 and return to **Step 1**.

These alterations in the FSP algorithm enable one to handle problems in which the initial probability density vector is not sparse. On its own, this may be convenient when we wish to study systems that begin somewhere within a range of possible initial configurations. However, as we will illustrate in the following section, the non-sparse FSP algorithm has its greatest use when it is integrated into a multiple time stepping FSP algorithm.

4 The Multiple Time Step FSP Method

Suppose that we require that the FSP solution is precise to a 1-norm error of ε for the entire time interval $(0, t_f)$. This requires that the system remains with probability (1-ε) within the finite set $\mathbf{X}_J$ *for all times* $t \in (0, t_f)$. One can envision many simple cases where such a restriction can require an exorbitantly large space $\mathbf{X}_J$. Suppose our system begins with an initial condition at $t = 0$ far from the support of the distribution at the later time t_6 as illustrated in Figure 1a. In this case the probability distribution is likely to evolve along some path connecting the initial condition to the final solution. To achieve acceptable accuracy at all times, the projection region must contain not only the initial condition and the final solution, but also every point likely to be reached during the intervening time. In such a circumstance, it can help to break the time interval into pieces and require only that the FSP criteria are satisfied only during each sub-interval. In effect, we seek a changing projection space that follows the support of the distribution as it evolves. To do this, we utilize the linearity and time invariance properties of the chemical master equation.

Suppose we start with a known initial probability distribution, $\mathbf{P}(0)$, and we wish to approximate the solution to the CME in k time steps of equal length τ. Using the algorithm in the previous section, we can specify a positive $\delta < 1$ and require that transition vectors $\{\mathbf{E}_i\}$ satisfy $||\mathbf{E}_i||_1 \geq \delta$ for all i. For the first time step, suppose that we simply specify $\gamma_1 = \delta$ and we use the non-sparse FSP algorithm to find an approximation of the distribution at $t_1 = \tau$ such that

$$\mathbf{0} \leq \mathcal{D}_{I_1}\left\{\mathbf{P}_{I_1}^{FSP}(t_1)\right\} \leq \mathbf{P}(t_1) \text{ and } \left|\left|\mathbf{P}_{I_1}^{FSP}(t_1)\right|\right|_1 \geq \gamma_1\delta = \delta^2.$$

For the second time step, we use $\mathbf{P}_{I_1}^{FSP}(t_1)$ as the initial distribution. If we use the same δ, we can save some effort by reusing some of the $\mathbf{E}_i$'s already

computed. However, since our solution at the end of the previous step has a guaranteed sum of only δ^2, we must choose a different γ_2. A very reasonable choice is simply to use the guarantee from the previous step: $\gamma_2 = \delta^2$. With this choice, we can again apply the non-sparse FSP algorithm to find an FSP solution at the end of the second time step such that

$$\mathbf{0} \leq \mathcal{D}_{I_2}\left\{\mathbf{P}_{I_2}^{FSP}(t_2)\right\} \leq \mathbf{P}(t_2) \text{ and } \left|\left|\mathbf{P}_{I_2}^{FSP}(t_2)\right|\right|_1 \geq \delta^3.$$

Following this example, at each k^{th} step, if we use $\gamma_k = \delta^k$, then we will recover a solution such that

$$\mathbf{0} \leq \mathcal{D}_{I_k}\left\{\mathbf{P}_{I_k}^{FSP}(t_k)\right\} \leq \mathbf{P}(t_k) \text{ and } \left|\left|\mathbf{P}_{I_k}^{FSP}(\tau)\right|\right|_1 \geq \delta^{k+1}.$$

If we apply the fact that $||\mathbf{P}(t_k)||_1 = 1$, we have

$$\left|\left|\mathbf{P}_{I_k}^{FSP}(\tau)\right|\right|_1 \geq (||\mathbf{P}(t_k)||_1 - 1) + \delta^{k+1},$$

which after some rearranging yields

$$\left|\left|\mathbf{P}(t_k) - \mathcal{D}_{I_k}\left\{\mathbf{P}_{I_k}^{FSP}(\tau)\right\}\right|\right|_1 \leq 1 - \delta^{k+1}.$$

Suppose that we wish to find a solution that is within ε of the exact solution of the CME at $t_f = K\tau$. Following the ideas above, we would choose δ according to the relation $\varepsilon = 1 - \delta^{k+1}$, or $\delta = (1-\varepsilon)^{\frac{1}{K+1}}$. This procedure is stated more formally in the following algorithm.

The Multiple Time Step FSP Algorithm

Inputs Propensity functions and stoichiometry for all reactions.
Initial probability distribution, $\mathbf{P}(t_0)$.
Final time of interest, t_f.
Total error, $\varepsilon > 0$.

Step 0 Choose the number of time steps, K, and calculate $\tau = t_f/K$.
Compute the required sum for each $\mathbf{E}_i$, $\delta = (1-\varepsilon)^{\frac{1}{K+1}}$.
Initialize time step counter: k=0.
Choose initial time index set, I_0, such that $||\mathbf{P}_{I_0}(t_0)||_1 \geq \delta$.
Initialize the FSP approximate solution at t_0, $\mathbf{P}_{I_0}^{FSP}(t_0) = \mathbf{P}_{I_0}(t_0)$.

Step 1 Run the Non-Sparse FSP algorithm with the initial condition $\mathbf{P}_{I_k}^{FSP}(t_k)$, and error parameters δ and $\gamma_k = \delta^{k+1}$ and get $\mathbf{P}_{I_{k+1}}^{FSP}(t_{k+1})$.

Step 2 If $k+1 = K$, then **Stop**.
$\mathcal{D}_{I_K}\left\{\mathbf{P}_{I_K}^{FSP}(t_K)\right\}$ approximates $\mathbf{P}_{I_k}(t_f)$ to within ε.

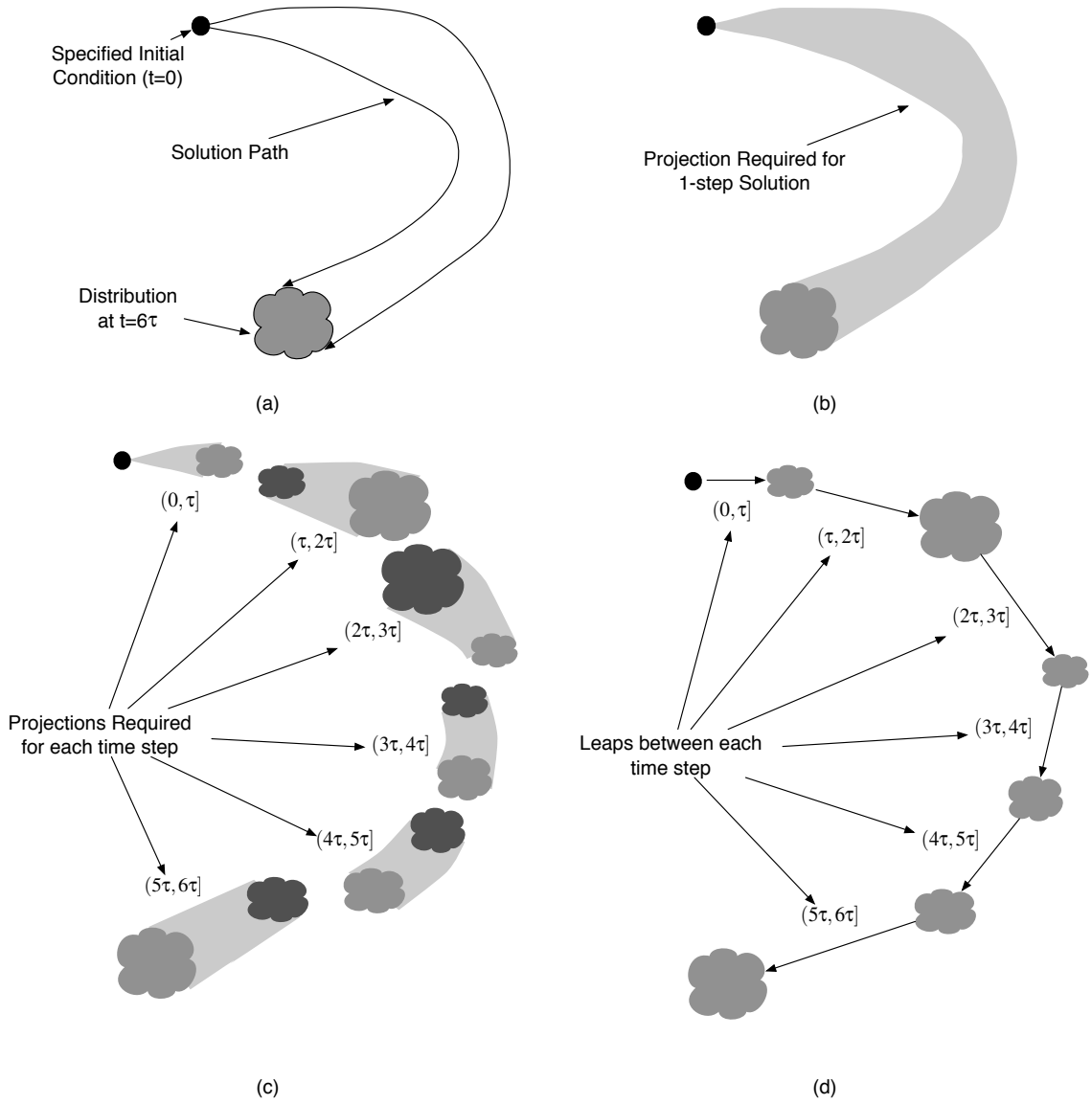

Fig. 1. Schematic of the Multiple Time Step FSP method. (a) We are given a Markov process that begins at a known initial point in the configuration space. As the probability distribution evolves, it follows a long path in the configuration space such that at time t_6 the distribution is supported in a region far from the initial condition. (b) In order to find a sufficiently accurate FSP solution for all times in the interval $[0, 6\tau]$, the FSP must include not only the initial condition and the final distribution, but also all points along the path. (c) To save computational effort, one can discretize the time interval into smaller intervals and find overlapping projections that need only satisfy the accuracy requirements during those shorter periods of time. Here the final distribution of each time step (shown in grey) becomes the initial distribution for the next time step (shown in black). (d) The end result is a discrete map taking the distribution from one instant in time to the next.

Step 3 Increment k and return to **Step 1**.

To see how one may benefit from this modification to the FSP algorithm, we refer to Figure 1. Suppose that we are interested in finding the distribution at time $t = 6\tau$ of a Markov process that begins in the known initial configuration represented by the black dot. Even though the distribution at each of the times $\{0, \tau, 2\tau, \ldots, 6\tau\}$ are supported on only a small portion of the configuration space, the one shot FSP solution must include the whole region of the configuration space that is swept by the distribution between 0 and 6τ (see Fig 1b). Therefore, the one step FSP algorithm requires a large matrix exponential computation. By splitting the full interval into six subintervals as shown in Fig. 1c, we will require more exponential computations, but since each of these computations will be much smaller, the total computational effort may be much less. The following subsection illustrates the use of this algorithm through a simplified model of the heat shock response in E-coli.

4.1 Toy Heat Shock Example

When a cell's environment changes, that cell must either adapt or perish. Continually faced with this choice, life has evolved to contain many complex gene regulatory networks that allow for quick and precise adaptation. The heat shock response in *E. coli* is excellent example of one such mechanism [20]. A simplified version of this system consists of three biochemical species that interact according to a set of three reactions,

$$s_1 \rightleftharpoons s_2 \rightarrow s_3, \tag{18}$$

where s_1, s_2 and s_3 correspond to the σ_{32}-DnaK complex, the σ_{32} heat shock regulator and the σ_{32}-RNAP complex, respectively. This model of the heat shock subsystem has been analyzed before using various computational methods including Monte Carlo implementations [7,21] as well as the FSP with the multiple time-scale model reduction [17]. Here we use this model in order to illustrate our current computational algorithm.

We will take the reaction rates and initial configuration for the system to be:

$$c_1 = 10, \;\; c_2 = 4 \times 10^4, \;\; c_3 = 2, \;\; s_1(0) = 2000, \;\; s_2(0) = s_3(0) = 0. \tag{19}$$

For these parameters there are 2,001,000 reachable configurations, and the full chemical master equation is far too large to be solved exactly. Applying the Finite State Projection method allows us to significantly reduce the order of the problem and achieve a manageable solution at least for small time intervals ($t \leq 300s$). In order to retain accuracy in the solution to an error of $\varepsilon = 10^{-3}$ at a time $t_f = 100s$, we need include the 1430 configurations where $s_3 \leq 129$

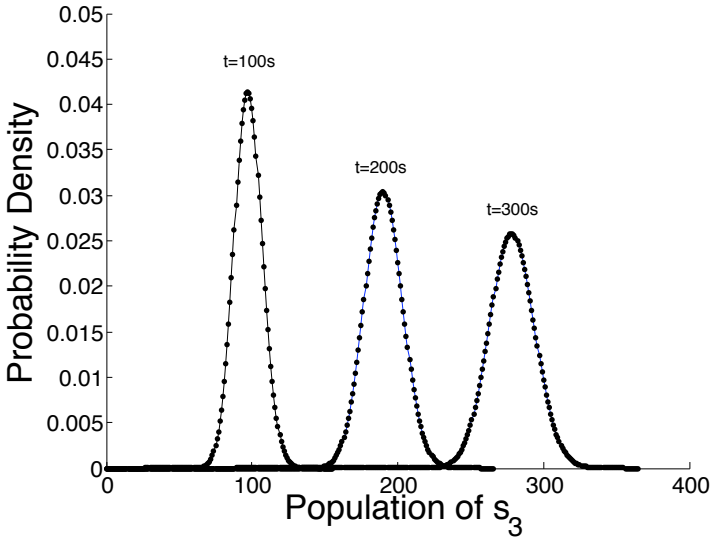

Fig. 2. Probability distribution of the population of the σ_{32}-RNAP complex at 100, 200 and 300 seconds as computed with the basic FSP algorithm (solid line) and with the Multiple Time Step FSP algorithm (dotted line).

and $s_2 \leq 10$. For larger time intervals, we need to include more points in the FSP solution: for $t_f = 200s$, we need include 2585 configurations where $s_3 \leq 234$ and $s_2 \leq 10$; and for $t_f = 300s$, we need include 3641 configurations where $s_3 \leq 330$ and $s_2 \leq 10$. The solid lines in Figure 2 present the probability distributions for the number of s_3 molecules at 100, 200 and 300s, and the top section of Table 1 summarizes the computational requirements necessary to achieve these results with the basic FSP algorithm. We have also applied the multiple time step FSP algorithm to the toy heat shock model in order to compute the probability distribution at these same times. As a first attempt, we have used a step size of five seconds. Figure 2 shows that there is no discernible difference between the results of the basic FSP algorithm and those of the multiple time step FSP algorithm. For a comparison of computational efforts, the bottom half of Table 1 provides the maximum matrix size, number of matrices, and computational time required for the multiple time step FSP algorithm. As the total time increases from 100 to 300 seconds, so does the computational benefit of the discretized algorithm; for a final time of 100, the current algorithm reduces computational time by a factor of about 2.5, for a final time of 300, the reduction is a factor of about 11.8.

Of course, while the accuracy of the multiple time step FSP is guaranteed, the efficiency of the algorithm will undoubtedly depend upon our chosen step size. Figure 3 illustrates some of the subtleties of this tradeoff by plotting the size of the largest exponentiated matrix, the number of matrix exponentials, and the computational time all as functions of the number of time steps (bottom axis) and the step length (top axis). As we use more time steps, the probability distribution has less time to disperse between one step and the next, and

the required matrix exponentials are smaller as shown in Fig. 3a. However, because the matrix dimension is a discrete integer quantity, this decrease is stepwise rather than smooth, and a large range of step lengths may require the same matrix size. If a step length is at the low end of that range, the matrix exponentials required to get each $\mathbf{E}_i$ are often slightly more precise than is absolutely necessary, and are therefore more likely to provide other $\mathbf{E}_j$'s as well–fewer exponential computations are necessary. Conversely, if a step length is at the high end of the range for a given matrix size, fewer $\mathbf{E}_j$'s will come from each exponential computation–more exponential computations are necessary. This trend is clear when one compares Fig 3a to 3b.

In order to show how these concerns affect the computation, we have broken the total computational cost in Fig 3*c*; into two components. The first cost is that of computing the matrix exponentials. The second cost is that of updating the solution from one step to the next. For $t_f = 300s$, this tradeoff is optimized for 130 time steps corresponding to a interval length of $\tau \approx 2.3s$. To obtain the solution with this time step, the algorithm needed to compute 50 matrix exponentials of size 198×198 or smaller, and the computation takes about 28s.

Extrapolating from Table 1(top) suggests that a regular FSP solution at $t = 1000s$ would require inclusion of more than 11000 configurations. Unfortunately, the memory requirement to exponentiate a 11000×11000 matrix exceeds the specifications of our machine, and the unmodified FSP algorithm cannot be used. Alternatively, by discretizing the full time interval, we can significantly reduce the computational complexity and bring the model back into the realm of a solvable problem. Once again, there is a definite tradeoff between too many and too few time steps, and Figure 4 plots the the size of the largest exponentiated matrix, the number of matrix exponentials, and the computational time as a function of the number of time steps. For $t_f = 1000s$, the computational tradeoff is optimized for 290 time steps corresponding to a interval length $\tau \approx 3.4$. To obtain the solution with this time step, the algorithm needed to compute 107 matrix exponentials of size 252×252 or smaller, and the computation takes about 166s.

5 Conclusions

Although the original finite state projection method can significantly reduce the order of the chemical master equation for many problems, this initial reduction is not sufficient for all systems. Fortunately, the FSP is amenable to numerous modifications, which can considerably improve upon the method's range and potency. In this paper we have concentrated on one computational difficulty that arises when system trajectories slowly drift over large regions of

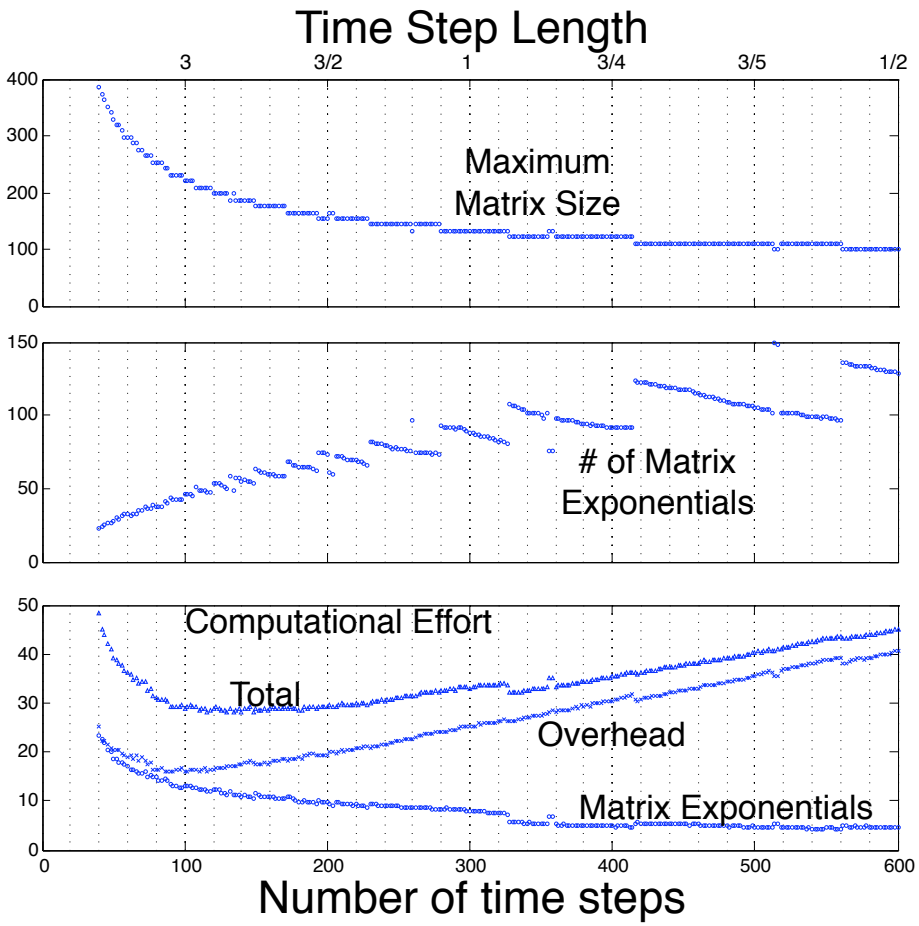

Fig. 3. Trade off between more and fewer time steps in the Multiple Time Step FSP (MTS-FSP) algorithm solution for the toy heat shock model at a final time of $t_f = 300s$. The following are plotted as function of the number of steps: (top) the size of the largest required matrix exponential computation, (middle) the number of matrix exponential computations performed, (bottom) the computational time required for the MTS-FSP algorithm split into two components: the exponential computation costs, and the overhead costs. All computations have been performed in Matlab 7.2 on a Dual 2 Ghz PowerPC G5.

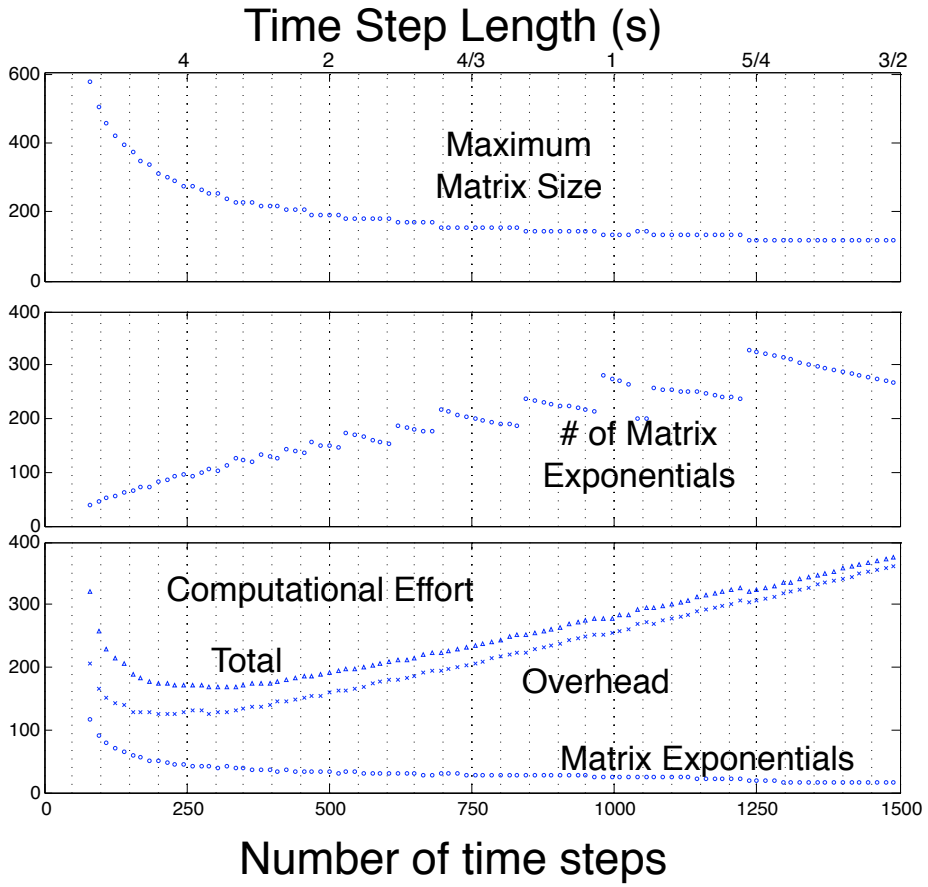

Fig. 4. Trade off between more and fewer time steps in the Multiple Time Step FSP (MTS-FSP) algorithm solution for the toy heat shock model at a final time of $t_f = 1000s$. The following are plotted as function of the number of steps: (top) the size of the largest required matrix exponential computation, (middle) the number of matrix exponential computations performed, (bottom) the computational time required for the MTS-FSP algorithm split into two components: the exponential computation costs, and the overhead costs. All computations have been performed in Matlab 7.2 on a Dual 2 Ghz PowerPC G5.

Table 1
Comparison of the computational requirements of the basic FSP algorithm and the Multiple Time Step FSP (MTS-FSP) algorithm for the toy heat shock example for three different final times: $t_f = 100$, 200, and 300 seconds. For the MTS-FSP algorithm results in this table, we have used a non-optimal step size of 5 seconds, different step sizes will provide the same accuracy at a lower computational cost.

Original FSP Algorithm			
Final Time (s)	Matrix Size	Comp. Time [a] (s)	Error $\|.\|_1$
100	1430×1430	27	9.61×10^{-4}
200	2585×2585	165	9.04×10^{-4}
300	3641×3641	437	9.67×10^{-4}
1000	$\approx 11000 \times 11000$		

Multiple Time Step FSP Algorithm				
Final Time (s)	# Exponentials	Matrix Size	Comp. Time (s)	Error $\|.\|_1$
100	14	275×275	11	2.84×10^{-4}
200	27	275×275	23	4.25×10^{-4}
300	39	275×275	37	5.08×10^{-4}
1000	104	252×252	180	1.8×10^{-4}

[a] Computations have been performed in Matlab 7.2 on a Dual 2 GHz PowerePC G5

the configuration space during long time intervals. In order to use the original FSP method for these cases, one must include vast portions of the configuration space in the projected solution. As the size of the included configuration space increases, so do the computational requirements of the FSP. However, in some cases this difficulty can be ameliorated simply by solving the FSP for a series of smaller time intervals. Here we have presented the Multiple Time Step FSP (MTS-FSP) algorithm, which is essentially an incremental approach to solving the original FSP.

The MTS-FSP algorithm is built upon three important aspects that the FSP inherits from the chemical master equation: linearity, time-invariance, and positivity. The linearity of the FSP allows us to apply the principle of superposition with regards to initial conditions–if we know the probability distribution at time 0 and we know the conditional probabilities at time τ conditioned on each configuration at time 0, then we can easily compute the probability distribution at time τ. The time invariance of the FSP assures us that if we know the probabilities at time τ conditioned on time 0, then we also know the probabilities at time $t+\tau$ conditioned on *any* time t. The positivity of the FSP guarantees us that we never over-predict the solution to the CME. Whether

we neglect some portion of the initial probability distribution or lose some of that distribution to configurations excluded from our various projections, the resulting error is known at the end of each time step. By directly controlling the error at each time-step, we can control the final error.

We have demonstrated the MTS-FSP algorithm on the toy heat shock problem. For time intervals of one, two and three hundred seconds the FSP and the current algorithm produce nearly identical results, but with the new method, we can compute those results much faster. In addition, the new algorithm extends the range of problems to which the FSP approach may be applied. To solve the toy heat shock problem over a time interval of one thousand seconds, original FSP algorithm requires a configuration space of over 11000 points, which is too large to manage. However, we can now easily solves the problem using matrices less than one thirtieth of the size.

This time stepping approach is just one of many mutually beneficial improvements that are quickly expanding the ability of the FSP to directly solve the chemical master equation. This current approach retains the full accuracy and properties of the original FSP and can easily be combined with other model reduction techniques such as those based linear systems and modern control theory. While we may never be able to directly solve every master equation, it remains to be seen just how far these FSP based approaches can push back the boundary between solvable and unsolvable.

6 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] 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.

[2] D. McQuarrie, "Stochastic approach to chemical kenetics," *J. Applied Probability*, vol. 4, pp. 413–478, 1967.

[3] D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *J. Phys. Chem.*, vol. 81, pp. 2340–2360, May 1977.

[4] E. Haseltine and J. Rawlings, "Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics," *J. Chem. Phys.*, vol. 117, pp. 6959–6969, Jul. 2002.

[5] C. V. Rao and A. P. Arkin, "Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the gillespie algorithm," *J. Chem. Phys.*, vol. 118, pp. 4999–5010, Mar. 2003.

[6] Y. Cao, D. T. Gillespie, and L. R. Petzold, "Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems," *J. Comp. Phys.*, vol. 206, pp. 395–411, July 2005.

[7] Y. Cao, D. Gillespie, and L. Petzold, "The slow-scale stochastic simulation algorithm," *J. Chem. Phys.*, vol. 122, Jan. 2005.

[8] H. Salis and Y. Kaznessis, "Accurate hybrid stochastic simulation of a system of coupled chemical or biological reactions," *J. Chem. Phys.*, vol. 112, no. 054103, 2005.

[9] D. T. Gillespie, "Approximate accelerated stochastic simulation of chemically reacting systems," *J. Chem. Phys.*, vol. 115, pp. 1716–1733, Jul. 2001.

[10] D. T. Gillespie and L. R. Petzold, "Improved leap-size selection for accelerated stochastic simulation," *J. Chem. Phys.*, vol. 119, pp. 8229–8234, Oct. 2003.

[11] M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie, "Stiffness in stochastic chemically reacting systems: The implicit tau-leaping method," *J. Chem. Phys.*, vol. 119, pp. 12784–12794, Dec. 2003.

[12] T. Tian and K. Burrage, "Binomial leap methods for simulating stochastic chemical kinetics," *J. Chem. Phys.*, vol. 121, pp. 10356–10364, Dec. 2004.

[13] Y. Cao, D. T. Gillespie, and L. R. Petzold, "Avoiding negative populations in explicit poisson tau-leaping," *J. Chem. Phys.*, vol. 123, no. 054104, 2005.

[14] Y. Cao and L. R. Petzold, "Accuracy limitations and the measurement of errors in the stochastic simulation of chemically reacting systems," *J. Comp. Phys.*, vol. 212, pp. 6–24, Feb. 2006.

[15] B. Munsky, A. Hernday, D. Low, and M. Khammash, "Stochastic modeling of the pap-pili epigenetic switch," *Proc. FOSBE*, pp. 145–148, August 2005.

[16] B. Munsky and M. Khammash, "A reduced model solution for the chemical master equation arising in stochastic analyses of biological networks," *Proc. 45th IEEE CDC*, Dec. 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.*, vol. 125, Nov. 2006.

[18] B. Munsky, S. Peles, and M. Khammash, "Stochastic analysis of gene regulatory networks using finite state projections and singular perturbation," *Submitted to ACC*, Jul. 2007.

[19] van Kampen, *Stochastic Processes in Physics and Chemistry.* Epsevier, 3 ed., 2001.

[20] H. El Samad, H. Kurata, J. Doyle, C. Gross, and K. M., "Surviving heat shock: Control strategies for robustness and performance," *PNAS*, vol. 102, no. 8, p. 27362741, 2005.

[21] H. El Samad, M. Khammash, L. Petzold, and D. Gillespie, "Stochastic modeling of gene regulatory networks," *Int. J. Robust Nonlin.*, vol. 15, pp. 691–711, 2005.