Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 226 (2007) 818–835

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A multiple time interval finite state projection algorithm for the solution to the chemical master equation

Brian Munsky, Mustafa Khammash *

Department of Mechanical Engineering, University of California Santa Barbara, Santa Barbara, CA 93106, United States

Received 29 December 2006; received in revised form 19 April 2007; accepted 4 May 2007
Available online 31 May 2007

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). Although only a few classes of CME problems are known to have exact and computationally tractable analytical solutions, the recently proposed finite state projection (FSP) technique provides a systematic reduction of the CME with guaranteed accuracy bounds. For many non-trivial systems, the original FSP technique has been shown to yield accurate approximations to the CME solution. Other systems may require a projection that is still too large to be solved efficiently; for these, the linearity of the FSP allows for many model reductions and computational techniques, which can increase the efficiency of the FSP method with little or no loss in accuracy. In this paper, we present a new approach for choosing and expanding the projection for the original FSP algorithm. Based upon this approach, we develop a new algorithm that exploits the linearity property of super-position. The new algorithm retains the full accuracy guarantees of the original FSP approach, but with significantly increased efficiency for some problems and a greater range of applicability. We illustrate the benefits of this algorithm on a simplified model of the heat shock mechanism in *Escherichia coli*.

Keywords: Chemical master equation; Stochastic gene regulatory networks; Markov processes

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, individual trajectories may not adequately describe the dynamics, and one may prefer to explore the evolution of the system's probability distribution. For chemical reactions at this scale, this probability distribution, $\mathbf{P}(t)$, has been shown to evolve according to the infinite dimensional linear ordinary differential equation,

* Corresponding author.
E-mail addresses: brianem@engr.ucsb.edu (B. Munsky), khammash@engr.ucsb.edu (M. Khammash).

0021-9991/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.05.016

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

which is generally known as the chemical master equation (CME) [1].

Although master equations, such as the CME, have been explicitly solved for some simple examples [2,3], computationally tractable, closed-form solutions are not known for most non-trivial systems. This lack of analytical solutions has helped drive computational research in kinetic Monte–Carlo (KMC) algorithms to simulate system dynamics. Gillespie's stochastic simulation algorithm (SSA) is one such algorithm developed specifically for processes described by the CME [4]. 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 to be exact in that given an ideal random number generator, its realizations will provide an unbiased sampling of the probability distribution that evolves according to the CME. In this sense, each SSA run provides a detailed example of how the system might evolve over time. In some cases, a few runs of the SSA provide a sufficient description of the system's dynamics, and it is not necessary to solve the CME. In other cases, when one is interested in accurately computing the probabilities of certain rare events, another approach may be desired. In terms of solving the CME, the SSA is still exact but now only in the sense that, if one were to conduct the SSA an infinite number of times, the collected statistical data would converge to the exact solution to the CME. However, in this regard the SSA suffers in that the convergence rates for KMC routines are 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 unmodified SSA may take a long time. 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 [5–9]. 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 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 [10–15]. 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 applied.

We recently developed the finite state projection algorithm as an additional tool for the analysis of jump Markov processes [16]. The FSP method approximates the solution to the CME without random number generation. This approach is based upon linear systems theory and works by projecting the intractable infinite dimensional master equation onto a solvable finite dimensional space. The FSP result comes in the form $\mathbf{P}^{\mathrm{FSP}}(t_f) = \mathbf{\Phi}(t_f, t_0)\mathbf{P}(t_0)$, where $\mathbf{\Phi}(t_f, t_0)$ is an operator that acts on distributions at the time t_0 and yields approximate distributions at the time t_f. One advantage of the FSP solution is that $\mathbf{\Phi}(t_f, t_0)$ is not linked to a specific initial probability distribution and can be applied to any initial distribution $\mathbf{P}(t_0)$. Another, perhaps more important, strength of the FSP is that it provides strict lower and upper bounds on the accuracy of $\mathbf{\Phi}(t_f, t_0)\mathbf{P}(t_0)$ for any $\mathbf{P}(t_0)$ [16].

Previous implementations of the FSP method have been very successful for some biologically inspired systems [16,17], 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 performing minimal realizations [18] or using Krylov subspace methods [19], others are inspired from similar reductions applied to the KMC algorithms. For example, in [20,21], we use perturbation theory to significantly reduce the dimension of the FSP equations without noticeable detriment to the method's accuracy. This reduction, which is based upon partitioning the system dynamics according to time-scale separations, closely parallels similar reductions to KMC algorithms in [6–8].

Fig. 1 illustrates the basic concept behind the current approach to improve the FSP method. Over a given time interval, a system is likely to exhibit to a number of different configurations or states. In order to retain accuracy, an FSP solution for such a system must include one ODE for each of these configurations. Longer time intervals mean that more configurations and more equations will be necessary. By subdividing the time interval into shorter intervals, one may save significant computational effort by solving a few smaller dimensional ODEs rather than a single large dimensional ODE. Recently, Burrage et. al have utilized this benefit of solving the FSP problem over a set of small time intervals in their Krylov-based FSP algorithm [19]. Their approach is more efficient than the original FSP in that they no longer solve for the full operator $\mathbf{\Phi}(t_f, t_0)$,

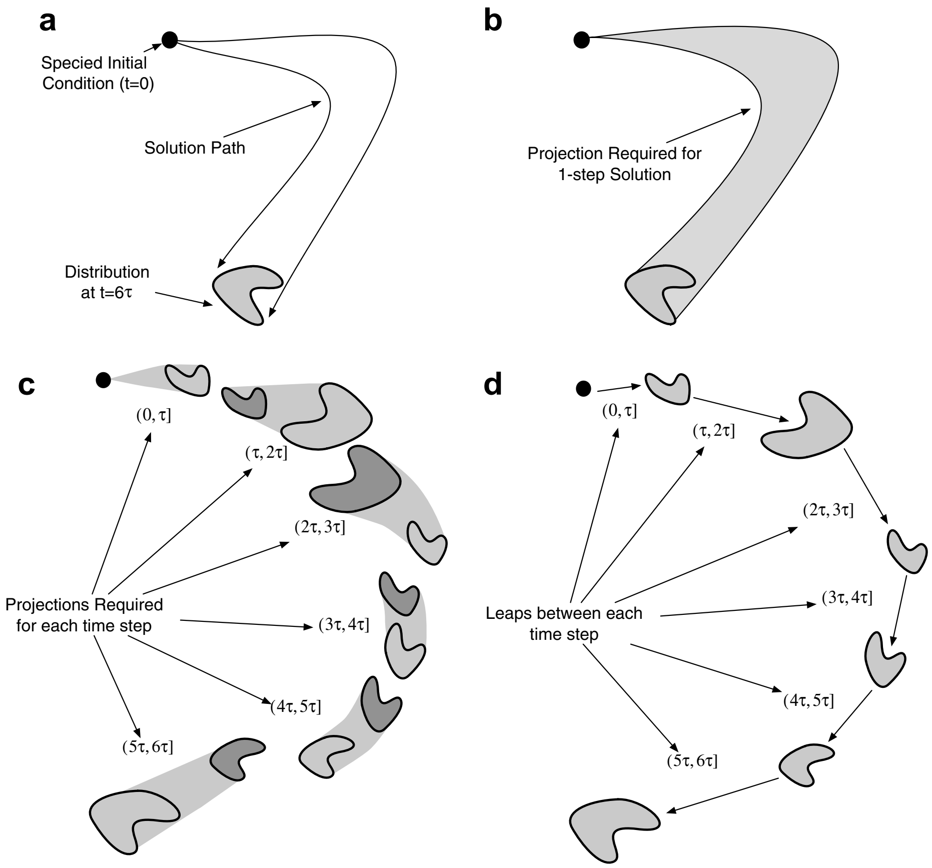

Fig. 1. Schematic of the multiple time interval FSP method. (a) We are given a Markov process that begins at a known initial point in the configuration space. As the probability distribution evolves, it follows a long path in the configuration space such that at time t_6 the distribution is supported 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, 6τ], 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 interval (shown in grey) becomes the initial distribution for the next time interval (shown in black). (d) The end result is a discrete map taking the distribution from one instant in time to the next.

but instead restrict their efforts to directly computing $\mathbf{P}^{\mathrm{FSP}}(t_f) = \mathbf{\Phi}(t_f, t_0)\mathbf{P}(t_0)$. In this paper, we take a very different approach. Rather than sacrifice the original FSP's ability to handle different initial distributions, as must be done in order to use the Krylov reductions in [19], we will instead exploit this flexibility. By restricting all time intervals to the same length, τ, time invariance of the CME guarantees that much of $\mathbf{\Phi}(\tau) = \mathbf{\Phi}(t + \tau, t)$ can be reused from one time interval to the next.

The outline of the remainder of this article is as follows: Section 2 provides greater background details on the original FSP approach, and introduces a new approach to automatically choosing and expanding the projection necessary to achieve a desired accuracy. In Section 3 we illustrate how the linearity of the FSP allows us to efficiently deal with initial probability density vectors that contain many non-zero elements; this approach essentially allows one to efficiently compute the operator $\mathbf{\Phi}(\tau)$ a few columns at a time rather than all at once. In Section 4 we introduce a multiple time interval version of the FSP and prove that it retains the same accuracy guarantees of the original FSP method. Section 5 then demonstrates the use of this algorithm with an example from the field of systems biology. Finally, in Section 6, 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, jump 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, $\mathbf{x} := [\xi_1, \xi_2, \ldots, \xi_N]^{\mathrm{T}}$. 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 $\mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2, \ldots]^{\mathrm{T}}$. Suppose that there are M possible reaction types, where each reaction is a transition from one configuration to another: $\mathbf{x}_i \to \mathbf{x}_j = \mathbf{x}_i + \nu_\mu$. In this notation, ν_μ is the directional transition on the configuration set (or stoichiometry) of the μth reaction type. Furthermore, let $a_\mu(\mathbf{x}_i)\mathrm{d}t$ be the infinitesimal probability that a system beginning in configuration $\mathbf{x}_i$ at time t will transition to the configuration $\mathbf{x}_i + \nu_\mu$ in the time interval $(t, t + \mathrm{d}t)$. Typically, $a_\mu(\mathbf{x}_i)$ is referred to as the propensity function of the μth reaction [1].

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, $\mathbf{P} = [p_1, p_2, \ldots]^{\mathrm{T}}$, evolves according to the linear ordinary differential equation known as the chemical master equation (CME) [1]:

$$\dot{\mathbf{P}}(t) = \mathbf{A}\mathbf{P}(t). \tag{2}$$

In the CME, the infinitesimal generator matrix, $\mathbf{A}$, contains information regarding every transition that the system can undergo from one configuration to another. The ordering of $\mathbf{A}$ is determined by the enumeration of the configuration set $\mathbf{X}$. Each ith diagonal element is negative with a magnitude equal to the sum of the propensity functions of reactions that *leave* the ith configuration. Each off-diagonal element, A_{ij}, is positive with magnitude $a_\mu(\mathbf{x}_j)$ if there is a reaction $\mu \in \{1, \ldots, M\}$ such that $\mathbf{x}_i = \mathbf{x}_j + \boldsymbol{\nu}_\mu$ and zero otherwise. In other words:

$$\mathbf{A}_{ij} = \left\{ \begin{array}{cc} -\sum_{\mu=1}^{M}(a_\mu(\mathbf{x}_i)) & \text{for } (i = j) \\ a_\mu(\mathbf{x}_j) & \text{for all } j \text{ such that}(\mathbf{x}_i = \mathbf{x}_j + \nu_\mu) \\ 0 & \text{Otherwise} \end{array} \right\}. \tag{3}$$

Like any generator matrix, the definition of $\mathbf{A}$ guarantees that all diagonal elements are non-positive; all off-diagonal elements are non-negative; and all columns sum to zero. When the set $\mathbf{X}$ has a finite number of members, one can in principle compute the solution $\mathbf{P}(t_f) = \mathbf{\Phi}(t_f)\mathbf{P}(0)$, where the operator $\mathbf{\Phi}(t_f)$ is given by $\mathbf{\Phi}(t_f) = \exp(\mathbf{A}t_f)$. However, 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 [16].

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 ($\cup, \cap$, 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, $\mathcal{D}_J\{\cdot\}$ 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 vector $\mathbf{e}^i$ to denote a vector with a one in the ith location and zeros elsewhere. With this notation we can restate the following two theorems from [16]:

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

$$[\exp \mathbf{A}]_J \geqslant \exp(\mathbf{A}_J) \geqslant \mathbf{0}. \tag{4}$$

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* $t_f \geqslant 0$,

$$\| \exp(\mathbf{A}_J t_f)\mathbf{P}_J(0)\|_1 \geqslant 1 - \varepsilon, \tag{5}$$

then

$$\mathcal{D}_J\{\exp(\mathbf{A}_J t_f)\mathbf{P}_J(0)\} \leqslant \mathbf{P}(t_f), \tag{6}$$

and

$$\|\mathbf{P}(t_f) - \mathcal{D}_J\{\exp(\mathbf{A}_J t_f)\mathbf{P}_J(0)\}\|_1 \leqslant \varepsilon. \tag{7}$$

Using the current notation, these theorems have been strengthened slightly from their original form, but their proofs are easily found with slight modification to those presented in [16]. To explain the intuition behind these two theorems, Fig. 2(top) illustrates a two-dimensional lattice corresponding to a two-species chemical reaction. The original FSP [16] includes a small subset of the configuration points, within the rectangle, and projects the remainder to a single absorbing point as shown in Fig. 2(bottom-left). The master equation for this new finite dimensional Jump Markov process is given by

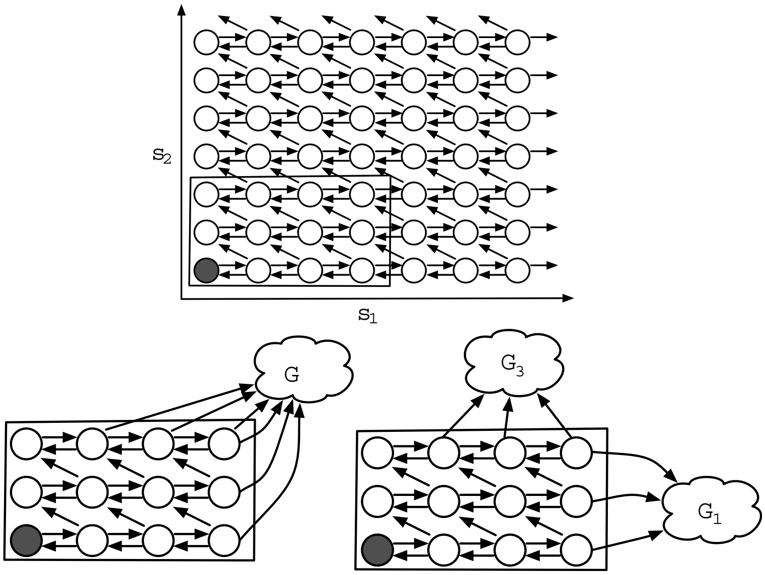

Fig. 2. (Top) Two-dimensional lattice of configurations for a chemically reacting system with two species. The system begins in the configuration shaded in grey and undergoes three reactions: The first reaction $\emptyset \to s_1$ results in a net gain of one s_1 molecule and is represented by right arrows. The second reaction $s_1 \to \emptyset$ results in a net loss of one s_1 molecule and is represented by a left arrow. The third reaction $s_1 \to s_2$ results in a loss of one s_1 molecule and a gain of one s_2 molecule. The dimension of the Master equation is equal to the total number of configurations, and is too large to solve exactly. Bottom Left: In the original FSP algorithm a configuration subset is chosen and all remaining configurations are projected to a single absorbing point, G. This results in a small dimensional solvable master equation, where the total error is given by the probability that has leaked into G. (Bottom Right) Instead of considering only a single absorbing point, transitions out of the finite projection can be sorted as to how they leave the projection space. In this case, G_1 and G_2 absorb the probability that has leaked out through reactions 1 and 2, respectively. this information can then be used to expand the configuration set in later iterations of the FSP algorithm.

$$\begin{bmatrix}\dot{\mathbf{P}}_J^{\mathrm{FSP}}(t)\\ \dot{G}(t)\end{bmatrix} = \begin{bmatrix}\mathbf{A}_J & \mathbf{0}\\ -\mathbf{1}^{\mathrm{T}}\mathbf{A}_J & 0\end{bmatrix}\begin{bmatrix}\mathbf{P}_J^{\mathrm{FSP}}(t)\\ G(t)\end{bmatrix}, \tag{8}$$

where the initial conditions are given as

$$\mathbf{P}_J^{\mathrm{FSP}}(t_0) = \mathbf{P}_J(t_0) \quad \text{and} \quad G(t_0) = \mathbf{1}^{\mathrm{T}}\mathbf{P}_{J'}(t_0) = 1 - \mathbf{1}^{\mathrm{T}}\mathbf{P}_J(t_0).$$

The solution to this new master equation is simply

$$\begin{bmatrix}\mathbf{P}_J^{\mathrm{FSP}}(t_f)\\ G(t_f)\end{bmatrix} = \begin{bmatrix}\exp(\mathbf{A}_J t_f) & \mathbf{0}\\ \mathbf{1}^{\mathrm{T}}\left(\mathbf{I} - \exp(\mathbf{A}_J t_f)\right) & 1\end{bmatrix}\begin{bmatrix}\mathbf{P}_J^{\mathrm{FSP}}(0)\\ G(0)\end{bmatrix}. \tag{9}$$

In terms of the original system, this expression is exact in the sense that $\mathbf{P}_J^{\mathrm{FSP}}(t_f)$ is a vector describing the probabilities that (1) the system is in the corresponding configurations $\mathbf{X}_J$ at $t = t_f$ *and* (2) that the process has been within the set $\mathbf{X}_J$ for the entire interval $(0, t_f)$. Furthermore, $G(t_f)$ is the probability that the system has left the set $\mathbf{X}_J$ at least once before the time t_f. Theorem 1 guarantees that as we add points to the finite configuration subset, $\mathbf{P}_J^{\mathrm{FSP}}(t_f)$, monotonically increases, and Theorem 2 provides a certificate of how close the approximation is to the true solution. In particular, since $\|\mathbf{P}_J^{\mathrm{FSP}}(t_f)\|_1 = 1 - G(t_f)$, then $\mathbf{P}_J^{\mathrm{FSP}}(t_f)$ is within a one norm error of $G(t_f)$ from the true distribution, i.e. $\|\mathbf{P}_J(t) - \mathbf{P}_J^{\mathrm{FSP}}(t)\|_1 \leqslant G(t)$. Furthermore, since the ∞-norm of any vector is never more than the 1-norm, the bound applies not only to the error in the whole distribution, but also to each individual point in the configuration space.

Together, these two theorems suggest the FSP algorithm [16]. Basically, this procedure works by examining a sequence of finite projections of the CME. For each projection set, one can obtain an accuracy guarantee using Theorem 2. If this accuracy is insufficient, more configurations can be added to the projection set, thereby monotonically improving the accuracy as guaranteed by Theorem 1. The full algorithm can be stated as follows:

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, $\mathbf{X}_{J_0}$, 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 $\Gamma_{J_i} \geqslant 1 - \varepsilon$, 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 $\varepsilon$.
Step 3  Add more states to find $\mathbf{X}_{J_{i+1}}$.
        Increment $i$ and return to Step 1.
```

In this FSP algorithm and those to be developed below, Steps 0 and 3 are left intentionally open as to how the projection sets are chosen. In practice, there may be many good ways to choose and expand the projection set, and it has not yet been adequately addressed in the literature as to which method, or methods, may be the best for which circumstances. Nevertheless, for the original expansion routine presented in [16] and for that to be introduced below, if there exists a sufficiently accurate FSP solution, then the FSP algorithm is guaranteed to converge in a finite number of steps.

2.1. Choosing and expanding the finite state projection

In previous work [16], the initial projection set $\mathbf{X}_{J_0}$ was an arbitrarily chosen set of configurations reachable from the initial condition. The most obvious choice is simply to choose $\mathbf{X}_{J_0}$ to contain only the initial configuration: $\mathbf{X}_{J_0} = \{\mathbf{x}(0)\}$. Here, we present a better approach to initializing the set $\mathbf{X}_{J_0}$ in Step 0 of the above algorithm. Instead of choosing $\mathbf{X}_{J_0}$ offline, we run the SSA [4] a few times and record every configuration reached in those simulations; we will use that set as the initial projection space, $\mathbf{X}_{J_0}$. If we use more SSA runs, $\mathbf{X}_{J_0}$ will likely be larger and therefore retain a larger measure of the probability distribution in the specified time interval. Therefore, as one uses more SSA runs in the initialization portion of Step 0, fewer iterations of the FSP algorithm should be necessary.

In [16] the FSP expansion was performed using the concept of N-step reachability, where each set $\mathbf{X}_{J_N}$ included all configurations that are reachable from $\mathbf{X}_{J_0}$ in N reactions or fewer. Proposition 3.1 in [16] guarantees that if there exists a sufficiently accurate FSP solution, then the FSP algorithm with such an expansion routine is guaranteed to converge in a finite number of steps. Here, our expansions will follow a new improved version of the N-step reachability routine. In the original FSP approach all configurations outside the set $\mathbf{X}_J$ have been projected to a single point yielding (8). Many alternative projections are possible. In particular, one can choose M absorbing points $\{G_1, \ldots, G_M\}$ where each $G_\mu(t)$ corresponds to the probability that the system has left the set $\mathbf{X}_J = \{\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, \ldots\}$ via a μth reaction. Fig. 2(bottom-right) illustrates such a projection choice. For this choice, we arrive at a new master equation:

$$\begin{bmatrix} \dot{\mathbf{P}}_J^{\mathrm{FSP}}(t) \\ \dot{\mathbf{G}}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_J & \mathbf{0} \\ \mathbf{Q} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{P}_J^{\mathrm{FSP}}(t) \\ \mathbf{G}(t) \end{bmatrix}, \tag{10}$$

where $\mathbf{G} = [G_1, \ldots, G_M]^{\mathrm{T}}$ and the matrix $\mathbf{Q}$ is given by

$$\mathbf{Q}_{\mu k} = \left\{ \begin{matrix} a_\mu(\mathbf{x}_{j_k}) & \text{if } (\mathbf{x}_{j_k} + \nu_\mu) \notin \mathbf{X}_J \\ 0 & \text{Otherwise} \end{matrix} \right\}.$$

The solution of (10) at a time t_f has the form

$$\begin{bmatrix} \mathbf{P}_J^{\mathrm{FSP}}(t) \\ \mathbf{G}(t) \end{bmatrix} = \begin{bmatrix} \exp(\mathbf{A}_J t_f) & \mathbf{0} \\ \int_0^{t_f} \mathbf{Q} \exp(\mathbf{A}_J \tau) \mathrm{d}\tau & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{P}_J^{\mathrm{FSP}}(0) \\ \mathbf{G}(0) \end{bmatrix}, \tag{11}$$

and yields all of the same information as (9), but it now provides additional useful knowledge. Specifically, each column of the operator in (11) corresponds to a specific $\mathbf{x}_i \in \mathbf{X}_J$. Each of the last M elements of the column corresponding to $\mathbf{x}_i$ gives the exact probability that a trajectory beginning in $\mathbf{x}_i$ at time $t = 0$ will have exited the set $\mathbf{X}_J$ via a specific reaction channel before the time $t = t_f$. This knowledge is easily incorporated into Step 3 of the above algorithm. If most of the probability measure left via one particular reaction, it is reasonable to expand $\mathbf{X}_J$ in the corresponding direction. If very little of the probability measure leaks out via a given reaction, it would be useless to expand the projection in that direction.

For the basic FSP algorithm with this or any other expansion routine, 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 [16,17], 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 [16–21] begin with a specific known initial configuration; if the system begins in configuration $\mathbf{x}_k$, 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 interval. 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:

$$\|\exp(\mathbf{A}_J t_f)\mathbf{P}_J(0)\|_1 \geqslant (1-\varepsilon). \tag{12}$$

Since the sum of the FSP solution at t_f cannot exceed the sum of the truncated initial pdv, $P_J(0)$, we must always include at least as many states in the FSP solution as is required such that $\|\mathbf{P}_J(0)\|_1 \geqslant 1-\varepsilon$. 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 super-positioning guaranteed by the linearity of the FSP. The underlying concept of this approach is illustrated in Fig. 3 and formalized in the following proposition.

Proposition 3. Super-position of FSP Solutions: *Consider any Markov process in which the distribution evolves according to the linear ODE*:

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

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

$$\|\mathbf{P}_I(0)\|_1 \geqslant \gamma, \tag{13}$$

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

$$\|\exp(\mathbf{A}_{J_i} t_f)\mathbf{e}^i_{J_i}\|_1 \geqslant \eta, \tag{14}$$

then,

$$\sum_{i\in I} p_i \mathcal{D}_{J_i}\{\exp(\mathbf{A}_{J_i} t_f)\mathbf{e}^i_{J_i}\} \leqslant \mathbf{P}(t_f), \tag{15}$$

and

$$\|\mathbf{P}(t_f) - \sum_{i\in I} p_i \mathcal{D}_{J_i}\{\exp(\mathbf{A}_{J_i} t_f)\mathbf{e}^i_{J_i}\}\|_1 \leqslant 1-\gamma\eta. \tag{16}$$

Proof. We begin by proving (15). If we define the index set $I_f = \cup_{i\in I} J_i$, then we have the relation,

$$\mathcal{D}_{I_f}\{\exp(\mathbf{A}_{I_f} t_f)\mathbf{P}_{I_f}(0)\} = \sum_{i\in I_f} p_i(0)\mathcal{D}_{I_f}\left\{\exp(\mathbf{A}_{I_f} t_f)\mathbf{e}^i_{I_f}\right\}. \tag{17}$$

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

$$\mathcal{D}_{I_f}\{\exp(\mathbf{A}_{I_f} t_f)\mathbf{P}_{I_f}(0)\} \geqslant \sum_{i\in I} p_i(0)\mathcal{D}_{I_f}\left\{\exp(\mathbf{A}_{I_f} t_f)\mathbf{e}^i_{I_f}\right\}. \tag{18}$$

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

$$\mathcal{D}_{I_f}\{\exp(\mathbf{A}_{I_f} t_f)\mathbf{P}_{I_f}(0)\} \geqslant \sum_{i\in I} p_i(0)\mathcal{D}_{J_i}\{\exp(\mathbf{A}_{J_i} t_f)\mathbf{e}^i_{J_i}\}. \tag{19}$$

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 (13) and (14) yields

$$\|\mathcal{D}_{I_f}\{\exp(\mathbf{A}_{I_f} t_f)\mathbf{P}_{I_f}(0)\}\|_1 \geqslant \left\|\sum_{i\in I} p_i(0)\mathcal{D}_{J_i}\{\exp(\mathbf{A}_{J_i} t_f)\mathbf{e}^i_{J_i}\}\right\|_1 \geqslant \eta\|\mathbf{P}_I(0)\|_1 \geqslant \eta\gamma. \tag{20}$$

Theorem 2 tells us that

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

and then from Eq. (19) we show that

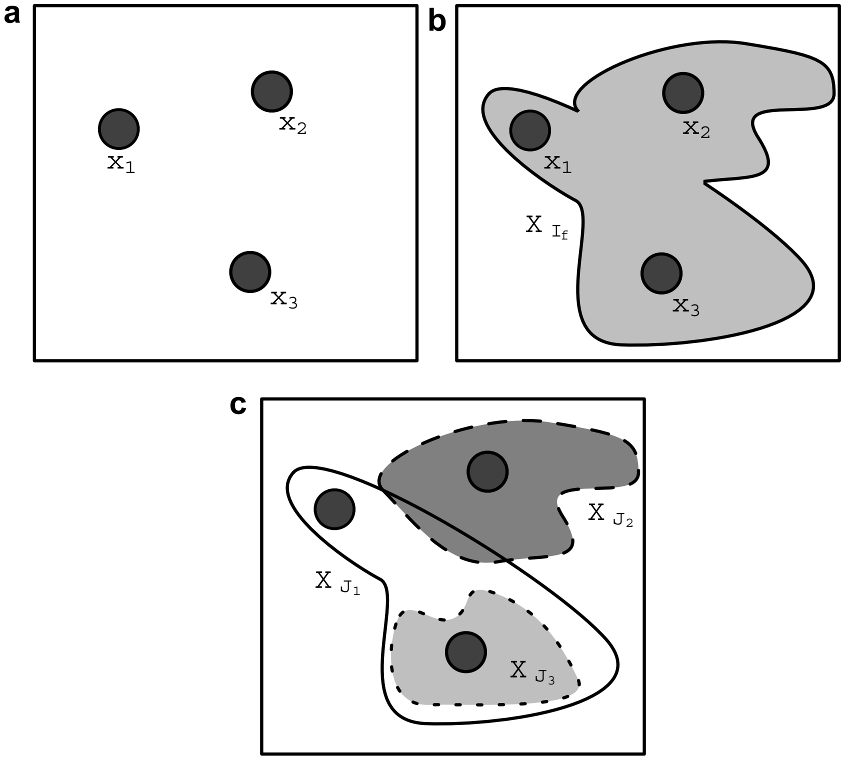

Fig. 3. Schematic of the FSP method using the property of super-position. (a) We are given a Markov process that begins with some distribution among a few possible initial configurations: $\mathbf{X}_{I_0} = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$. (b) In order to find a sufficiently accurate FSP solution for all times in the interval [0, t_f], the original FSP must include a large configuration region $\mathbf{X}_{I_f} \supseteq \mathbf{X}_{I_0}$. (c) To save computational effort, one can instead find projections that would satisfy each initial configuration separately; in this example sets $\mathbf{X}_{J_1}$, $\mathbf{X}_{J_2}$ and $\mathbf{X}_{J_3}$ are sufficient projections for system starting in $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$, respectively. Furthermore since $\mathbf{X}_{J_1} \supset \mathbf{X}_{J_3}$, the set $\mathbf{X}_{J_1}$ also suffices as the projection for the configuration $\mathbf{x}_3$. Therefore, instead of solving this problem as one big system as in (b), the system can instead be solved by considering two much smaller, linearly independent systems on the sets $\mathbf{X}_{J_1}$ and $\mathbf{X}_{J_2}$.

$$\sum_{i \in I_0} p_i(0) \mathcal{D}_{J_i} \left\{ \exp(\mathbf{A}_{J_i} t_f) \mathbf{e}^i_{J_i} \right\} \leqslant \mathbf{P}(t_f), \tag{21}$$

which is Eq. (15).

Combining the fact that $\|\mathbf{P}(t_f)\|_1 = 1$ and inequality (20) gives

$$\left\| \sum_{i \in I} p_i(0) \mathcal{D}_{J_i} \left\{ \exp(\mathbf{A}_{J_i} t_f) \mathbf{e}^i_{J_i} \right\} \right\|_1 \geqslant \left(\|\mathbf{P}(t_f)\|_1 - 1 \right) + \eta\gamma. \tag{22}$$

Rearranging this result and applying (21) yields inequality (16)

$$\left\| \mathbf{P}(t_f) - \sum_{i \in I} p_i(0) \mathcal{D}_{J_i} \left\{ \exp(\mathbf{A}_{J_i} t_f) \mathbf{e}^i_{J_i} \right\} \right\|_1 \leqslant 1 - \eta\gamma \tag{23}$$

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 interval 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, $\varepsilon = 1 - \eta\gamma$, where both γ and η 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 η 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 $\mathbf{E}_i = \mathcal{D}_{J_i}\{\exp(\mathbf{A}_{J_i}\tau)\mathbf{e}^i_{J_i}\}$ to denote the J_i indexed FSP approximation of the distribution at t_{k+1} conditioned upon the ith configuration at t_k. Each 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 $\mathbf{E}_j = \mathcal{D}_{J_i}\{\exp(\mathbf{A}_{J_i}\tau)\mathbf{e}^j_{J_i}\}$ and its corresponding index set $J_j = J_i$ that meets our accuracy requirement $\|\mathbf{E}_j\|_1 \geqslant \eta$. Note that each vector $\mathbf{E}_i$ is an approximation of the ith column of the operator $\mathbf{\Phi}(\tau)$, and the one norm error in this approximation is exactly $(1 - \mathbf{1}^{\mathrm{T}}\mathbf{E}_i)$. This means we are effectively storing a few columns of $\mathbf{\Phi}(\tau)$ at a time. These can later be reused to reduce the total number of matrix computations for a given initial probability distribution $\mathbf{P}(0)$. In addition, we can reuse $\mathbf{\Phi}(\tau)$ for any initial distribution that is supported on the set for which these columns of $\mathbf{\Phi}(\tau)$ have already been computed. 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, $0 \leqslant \gamma < 1$ and $0 \leqslant \eta < 1$.
        Initial probability distribution, $\mathbf{P}(t_k)$, where $1 \geqslant \|\mathbf{P}(t_k)\|_1 \geqslant \gamma$.
        Length of time interval, $\tau$.
Step 0  Choose a finite set of states, $\mathbf{X}_{I_k}$ such that $\|\mathbf{P}_{I_k}(0)\|_1 \geqslant \gamma$.
        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: $\mathbf{P}^{\mathrm{FSP}}_{I_f}(t_f) = 0$.
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 $\|\exp(\mathbf{A}_{J_i}\tau)\mathbf{e}^i_{J_i}\|_1 \geqslant \eta$.
        For every $j \in J_i$, if $\|\exp(\mathbf{A}_{J_i}t_f)\mathbf{e}^j_{J_i}\|_1 \geqslant \eta$, then record
        $\mathbf{E}_j = \mathcal{D}_{J_i}\{\exp(\mathbf{A}_{J_i}t_f)\mathbf{e}^j_{J_i}\}$ and $J_j = J_i$.
Step 2  Update the FSP solution index set: $I_f = I_f \cup J_i$.
        Update the FSP solution summation: $\mathbf{P}^{\mathrm{FSP}}_{I_f} = \mathbf{P}^{\mathrm{FSP}}_{I_f} + p_i\mathbf{E}_i$.
Step 3  If $i$ is the last element in $I_0$, **Stop**.
        $\mathcal{D}_{I_f}\{\mathbf{P}^{\mathrm{FSP}}_{I_f}(t_f)\}$ approximates $\mathbf{P}(t_f)$ to within $\varepsilon = 1 - \gamma\eta$.
Step 4  Increment $i$ to the next element in $I_0$ and return to **Step 1**.
```

As discussed above, there may be many choices for initializing and expanding the projection during the call to the FSP algorithm in Step 1. Here, the initial projection is chosen using a few SSA runs only on the first time that Step 1 is executed, but the initial projections for subsequent executions of Step 1 are found a little differently. In the previous step, we already computed a set $\mathbf{X}_{J_i}$ that is sufficient for an initial configuration $\mathbf{x}_i$, and we now wish to find a projection that is sufficient for a different initial configuration $\mathbf{x}_j$. As a first guess for $\mathbf{X}_{J_j}$, we take the set $\mathbf{X}_{J_i}$ and translate it by the amount $\mathbf{x}_j - \mathbf{x}_i$. In some cases, this may lead to unrealistic choices for the initial set, such as negative populations, but these are unreachable configurations that are automatically removed from the configuration set. Once this initial projection has been chosen, the expansion routine is the same as above in Section 2.1.

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 interval FSP algorithm.

4. The multiple time interval FSP method

Suppose that we require that the FSP solution be precise to a 1-norm error of ε for the entire time interval $(0, t_f)$. This requires that the system remains with probability $(1 - \varepsilon)$ within a 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 Fig. 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 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 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 intervals of equal length τ. Using the algorithm in the previous section, we can specify a positive $\eta < 1$ and require that transition vectors $\{\mathbf{E}_i\}$ satisfy $\|\mathbf{E}_i\|_1 \geqslant \eta$ for all i. For the first time interval, suppose that we simply specify $\gamma_1 = \eta$ and we use the non-sparse FSP algorithm to find an approximation of the distribution at $t_1 = \tau$ such that

$$\mathbf{0} \leqslant \mathcal{D}_{I_1}\left\{\mathbf{P}_{I_1}^{\text{FSP}}(t_1)\right\} \leqslant \mathbf{P}(t_1) \quad \text{and} \quad \|\mathbf{P}_{I_1}^{\text{FSP}}(t_1)\|_1 \geqslant \gamma_1\eta = \eta^2.$$

For the second time interval, we use $\mathbf{P}_{I_1}^{\text{FSP}}(t_1)$ as the initial distribution. If we use the same η, 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 interval has a guaranteed sum of only η^2, we must choose a different γ_2. A very reasonable choice is simply to use the guarantee from the previous interval: $\gamma_2 = \eta^2$. With this choice, we can again apply the non-sparse FSP algorithm to find an FSP solution at the end of the second time interval such that

$$\mathbf{0} \leqslant \mathcal{D}_{I_2}\left\{\mathbf{P}_{I_2}^{\text{FSP}}(t_2)\right\} \leqslant \mathbf{P}(t_2) \quad \text{and} \quad \|\mathbf{P}_{I_2}^{\text{FSP}}(t_2)\|_1 \geqslant \eta^3.$$

Following this example, at each kth step, if we use $\gamma_k = \eta^k$, then we will recover a solution such that:

$$\mathbf{0} \leqslant \mathcal{D}_{I_k}\left\{\mathbf{P}_{I_k}^{\text{FSP}}(t_k)\right\} \leqslant \mathbf{P}(t_k) \quad \text{and} \quad \|\mathbf{P}_{I_k}^{\text{FSP}}(\tau)\|_1 \geqslant \eta^{k+1}.$$

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

$$\|\mathbf{P}_{I_k}^{\text{FSP}}(\tau)\|_1 \geqslant (\|\mathbf{P}(t_k)\|_1 - 1) + \eta^{k+1},$$

which after some rearranging yields

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

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 η according to the relation $\varepsilon = 1 - \eta^{K+1}$, or $\eta = (1-\varepsilon)^{\frac{1}{K+1}}$. This procedure is stated more formally in the following algorithm.

The multiple time interval 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 intervals, $K$, and calculate $\tau = t_f/K$.
        Compute the required sum for each $\mathbf{E}_i$, $\eta = (1-\varepsilon)^{\frac{1}{K+1}}$.
        Initialize time step counter: $k = 0$.
        Choose initial time index set, $I_0$, such that $\|\mathbf{P}_{I_0}(t_0)\|_1 \geqslant \eta$.
        Initialize the FSP approximate solution at $t_0$, $\mathbf{P}_{I_0}^{\text{FSP}}(t_0) = \mathbf{P}_{I_0}(t_0)$.
Step 1  Run the non-sparse FSP algorithm with the initial condition $\mathbf{P}_{I_k}^{\text{FSP}}(t_k)$, and error parameters $\eta$ and
        $\gamma_k = \eta^{k+1}$ and get $\mathbf{P}_{I_{k+1}}^{\text{FSP}}(t_{k+1})$.
Step 2  If $k + 1 = K$, then Stop.
        $\mathcal{D}_{I_K}\{\mathbf{P}_{I_K}^{\text{FSP}}(t_K)\}$ approximates $\mathbf{P}_{I_k}(t_f)$ to within $\varepsilon$.
Step 3  Increment $k$ and return to Step 1.
```

To see how one may benefit from this modification to the FSP algorithm, we refer again to Fig. 1 above. 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 distributions 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.

In order to estimate the computational complexity of the new algorithm we will make a few assumptions. First, we will assume that in every call to the original FSP algorithm, the initially chosen projection $\mathbf{X}_{J_0}$ is sufficient to meet the desired accuracy tolerance. This allows us to analyze the complexity separate from the choice of FSP initialization and expansion routines. Let n denote the number of configurations necessary to solve the FSP in a single time interval. The cost of this solution is dominated by the matrix exponential computation on the order of $O(n^3)$. Suppose that the current multiple time interval version of the FSP can solve the same problem with K time intervals while using z matrix exponential computations of equal size s. Furthermore, assume that the sets $\{\mathbf{X}_{I_k}\}$ needed to support the probability distribution at the beginning of each time interval all have exactly w elements. In this case the cost of computing the z exponentials is $O(zs^3)$. The remaining overhead cost is broken into two parts: first, the cost of storing the n columns of $\mathbf{\Phi}(\tau)$ each with s non-zero elements is $O(ns)$. Second, the cost of all K matrix–vector updates is dominated by the cost of multiplying a $w \times s$ matrix by a w element vector or $O(Ksw)$. The total complexity of the multiple time interval FSP algorithm is then $O(zs^3) + O(ns) + O(Ksw)$. As K increases, smaller matrices will be necessary, but the rate at which s decreases will vary from one chemical system to the next. In general, for a small number of time intervals, s is large and the total cost is dominated by the exponential computations (first term). Conversely, for a large number of time intervals, the cost is dominated by the overhead (second two terms).

The following section illustrates the use of this algorithm through a simplified model of the heat shock response in *Escherichia coli*.

5. 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 [22]. A simplified version of this system consists of three biochemical species that interact according to a set of three reactions,

$$s_1 \to s_2, \quad s_2 \to s_1, \quad \text{and} \quad s_2 \to s_3, \tag{24}$$

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 [8,23] as well as the FSP with the multiple time-scale model reduction [20,21]. Here we use this model in order to illustrate our current computational algorithm.

In the toy heat shock model, the propensity functions are given by

$$a_1 = c_1[s_1], \quad a_2 = c_2[s_2], \quad \text{and} \quad a_3 = c_3[s_2]. \tag{25}$$

The parameters for the reaction rates and the initial configuration are

$$c_1 = 10, \quad c_2 = 4 \times 10^4, \quad c_3 = 2, \quad s_1(0) = K = 2000, \quad s_2(0) = s_3(0) = 0. \tag{26}$$

Our FSP solution to this problem has used the following enumeration scheme;

$$i(s_1, s_2, s_3) = s_3(K+1) + s_2 + 1,$$

which has the inverse

$$\mathbf{x}_i = \begin{bmatrix} s_1(i) \\ s_2(i) \\ s_3(i) \end{bmatrix} = \begin{bmatrix} K - \operatorname{mod}(i-1, K+1) - \operatorname{floor}((i-1)/(K+1)) \\ \operatorname{mod}(i-1, K+1) \\ \operatorname{floor}((i-1)/(K+1)) \end{bmatrix},$$

where mod(x, y) is the remainder after dividing x by y, and floor(x) rounds x down to the nearest integer. With this enumeration scheme and the propensity functions and stoichiometry from (24) and (25), one can form the infinitesimal generator $\mathbf{A}$ as

$$\mathbf{A}_{ij} = \left\{ \begin{array}{cc} -c_1 s_1(i) - c_2 s_2(i) - c_3 s_2(i) & \text{for } (i = j) \\ c_1 s_1(j) & \text{for } j \text{ s.t. } \mathbf{x}_i = \mathbf{x}_j + [-1, 1, 0]^{\mathrm{T}} \\ c_2 s_2(j) & \text{for } j \text{ s.}t.\ \mathbf{x}_i = \mathbf{x}_j + [1, -1, 0]^{\mathrm{T}} \\ c_3 s_2(j) & \text{for } j \text{ s.t. } \mathbf{x}_i = \mathbf{x}_j + [0, -1, 1]^{\mathrm{T}} \\ 0 & \text{Otherwise} \end{array} \right\}. \tag{27}$$

For the initial conditions in (26), the reachable configuration set is the set of all configurations such that $s_1 + s_2 + s_3 = K$. For $K = 2000$, one can show that the number of points in this set is

$$\sum_{s_3=0}^{K} \sum_{s_2=0}^{K-s_3} 1 = \sum_{s_3=0}^{K} s_3 = 2,001,000,$$

and therefore, the full chemical master equation is too large to be solved exactly, and an approximation is necessary.

Applying the original finite state projection method allows one to significantly reduce the order of the problem and achieve a manageable solution at least for small time intervals ($t \leqslant 300$ s). For our FSP implementation, we seek a one-norm error in the total distribution of less than 10^{-3}. In order to populate the initial configuration set $\mathbf{X}_{J_0}$ for the FSP, we have used twenty runs of the Stochastic Simulation Algorithm and recorded the maximum values for species s_2 and s_3 as m_2 and m_3, respectively. We then chose $\mathbf{X}_{J_0}$ to include all configurations such that $s_2 \leqslant 1.2m_2$ and $s_3 \leqslant 1.2m_3$. Although these SSA runs take a significant amount of time to complete, they provide a good estimate of the region needed in the projection space, and can significantly reduce the number of subsequent FSP expansion steps. We implemented this method on the heat shock model for three final times 100, 200, and 300 seconds. For $t_f = 100$ s, the SSA runs yielded $m_2 = 9$ and $m_3 = 120$, and $\mathbf{X}_{J_0}$ contained 1595 configurations where $s_2 \leqslant 10$ and $s_3 \leqslant 144$. For larger time intervals, more configurations are necessary: for $t_f = 200$ s, the SSA runs yielded $m_2 = 9$ and $m_3 = 215$, and $\mathbf{X}_{J_0}$ contained 2849 configurations where $s_2 \leqslant 10$ and $s_3 \leqslant 258$. For $t_f = 300$ s, the SSA runs yielded $m_2 = 9$ and $m_3 = 310$, and $\mathbf{X}_{J_0}$ contained 1595 configurations where $s_2 \leqslant 10$ and $s_3 \leqslant 372$. For each of the time intervals, the solution on the initial projection $\mathbf{X}_{J_0}$ exceeded our accuracy requirements, and a single FSP step sufficed. The solid lines in Fig. 4 present the probability distributions for the number of s_3 molecules at 100, 200 and 300 s, and the top section of Table 1 summarizes the computational requirements necessary to achieve these results with this particular implementation of the FSP solution.

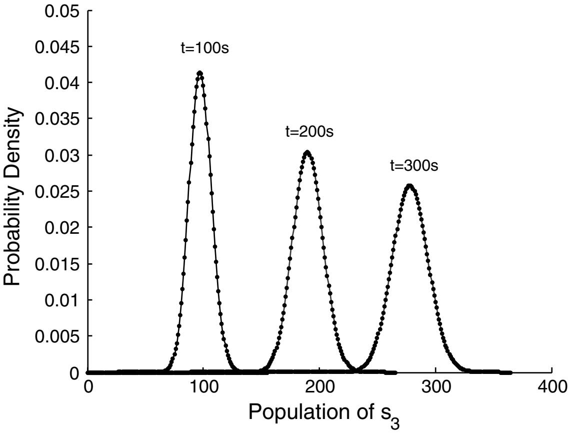

Fig. 4. Probability distribution of the population of the σ_{32}-RNAP complex at 100, 200 and 300 s as computed with the basic FSP algorithm (solid line) and with the multiple time interval FSP algorithm (dotted line).

Table 1
Comparison of the computational requirements of the basic FSP solution and the multiple time interval FSP (MTI-FSP) algorithm for the toy heat shock example for three different final times: $t_f = 100$, 200, and 300 s

FSP solution with 20 SSA runs to find $\mathbf{X}_{J_0}$ (with optimal choice of $\mathbf{X}_{J_0}$)

t_f (s)	# Exp's	Size of $\mathbf{X}_{J_0}$	C_{Total}[a] (s)	Error $\|\cdot\|_1$	C_{SSA}(s)	$C_{\exp}$ (s)
100	1 (1)	1595 (1430)	164 (27)	2.4×10^{-4} (9.61×10^{-4})	128	37 (27)
200	1 (1)	2849 (2585)	462 (165)	3.7×10^{-4} (9.04×10^{-4})	249	213 (165)
300	1 (1)	4103 (3641)	1003 (437)	4.6×10^{-4} (9.67×10^{-4})	354	649 (437)
1000	1 (1)	≈11,000		Exceeds machine memory		

Multiple time interval FSP Algorithm

t_f (s)	# Exp's	Size of $\mathbf{X}'_{J_k}s$	C_{Total}(s)	Error $\|\cdot\|_1$	C_{SSA}(s)	$C_{\exp}$ (s)	C_{OH}(s)
100	46	121	8.1	3.7×10^{-4}	1.4	2.9	3.8
200	89	121	17.8	4.6×10^{-4}	1.3	5.7	10.8
300	116	132	37.8	1.6×10^{-4}	1.4	12.5	24.0
1000	218	144	212	2.3×10^{-4}	1.3	27.3	183.4

For each method, different computational costs are given: C_{Total} is the total computational expense; C_{SSA} is the cost of the 20 SSA runs that have been used to initialize the set $\mathbf{X}_{J_0}$ for the first time interval; $C_{\exp}$ is the cost of exponentiating the matrices during the solution process; and in the case of the MTI-FSP method, C_{OH} is the overhead cost for the solution scheme. For the original FSP algorithm (top table), the results in parenthesis correspond to trials run with an *a priori* choice of $\mathbf{X}_{J_0}$. For the MTI-FSP algorithm results (bottom table), we have used a non-optimal interval size of one second; different interval sizes may provide the same or better accuracy at a lower computational cost.

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

From Table 1 it can be seen that the results generated using the above choice of $\mathbf{X}_{J_0}$ significantly exceed the accuracy requirement of $\varepsilon = 10^{-3}$, suggesting that a different routine for choosing and expanding the projection set may have allowed a more efficient computation. In order to provide a better benchmark for later comparisons with the new multiple time interval FSP method, we have used the results from above to find an *a posteriori* choice for $\mathbf{X}_{J_0}$. In order to retain accuracy in the solution to an error of $\varepsilon = 10^{-3}$ at a time $t_f = 100$ s, we need to include the 1430 configurations where $s_3 \leqslant 129$ and $s_2 \leqslant 10$; for $t_f = 200$ s, we need to include 2585 configurations where $s_3 \leqslant 234$ and $s_2 \leqslant 10$; and for $t_f = 300$ s, we need include 3641 configurations where $s_3 \leqslant 330$ and $s_2 \leqslant 10$. The results in parenthesis in the top section of Table 1 summarize the computational requirements and accuracy results for FSP solutions with these choices for the projection. While these new projections are still meet the specified accuracy requirements, they take far less time to compute.

We have also applied the multiple time interval FSP algorithm to the toy heat shock model in order to compute the probability distributions at 100, 200 and 300 s. We have used a non-optimal interval size of one second. In the first time interval, we initialize the projection with 20 SSA runs same as above. For each subsequent time interval, we use an initial projection set that is the same size as that required for the previous time interval, translated to the current region of the configuration space as described in Section 3. In all time intervals we have used an FSP expansion method based on that described in Section 2.1 as follows. For a given projection set $\mathbf{X}_{J_k}$, and a given initial configuration $\mathbf{x}_i \in \mathbf{X}_{J_k}$, let $G_\mu(\tau)$ denote the probability that a system beginning in $\mathbf{x}_i$ would leave the set $\mathbf{X}_{J_k}$ via the μth reaction during the time interval of length τ. For each μ, if $G_\mu(\tau) > \varepsilon/M$, then we expand $\mathbf{X}_{J_{k+1}}$ to also include the configurations that can be reached via one μth reaction from $\mathbf{X}_{J_k}$. In this manner the projection is expanded in every direction in which a significant portion of the distribution is moving.

Fig. 4 shows that there is no discernible difference between the results of the basic FSP algorithm and those of the multiple time interval FSP algorithm. For a comparison of computational efforts, the bottom portion of Table 1 provides the maximum matrix size, number of matrices, and computational time required for the multiple time interval 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 3.3, for a final time of 300, the reduction is a factor of about 11.6. Furthermore, these reductions are in comparison to the FSP benchmark in which the projection was known *a priori*; compared to the full FSP algorithm implementation, reductions of a factor of 20 have been achieved.

Of course, while the accuracy of the multiple time interval FSP is guaranteed, the efficiency of the algorithm depends upon our chosen interval size as discussed above. Fig. 5 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 intervals (bottom axis) and the interval length (top axis). As we use more time intervals, the probability distribution has less time to disperse between one interval and the next, and the required matrix exponentials are smaller as shown in Fig. 5(top) and 5(middle). However, because the matrix dimension is a discrete integer quantity, this decrease is stepwise rather than smooth, and a large range of interval lengths may require the same matrix size. If an interval 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 an interval 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. 5(top) and 5(middle).

In order to show how these concerns affect the computation, we have broken the total computational cost in Fig. 5(bottom) into three components. The first cost is that of computing the matrix exponentials; the second cost is the combined cost of storing the vectors $\{\mathbf{E}_i\}$ and then updating the solution from one interval to the next; and the third cost is the cost of initializing the first projection set with a set of 20 SSA runs. For $t_f = 300$ s, this tradeoff is optimized for 360 time intervals corresponding to a interval length of $\tau \approx 0.83$ s. To obtain the solution with this time interval, the algorithm needed to compute 122 matrix exponentials of size 121×121 or smaller, and the computation takes about 31.4 s.

Extrapolating from Table 1(top) suggests that a regular FSP solution at $t = 1000$ s would require inclusion of more than 11,000 configurations. Unfortunately, the memory requirement to exponentiate a $11{,}000 \times 11{,}000$ 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 intervals, and Fig. 6 plots the size of the largest exponentiated matrix, the number of matrix exponentials, and the computational time as a function of the number of time intervals. For

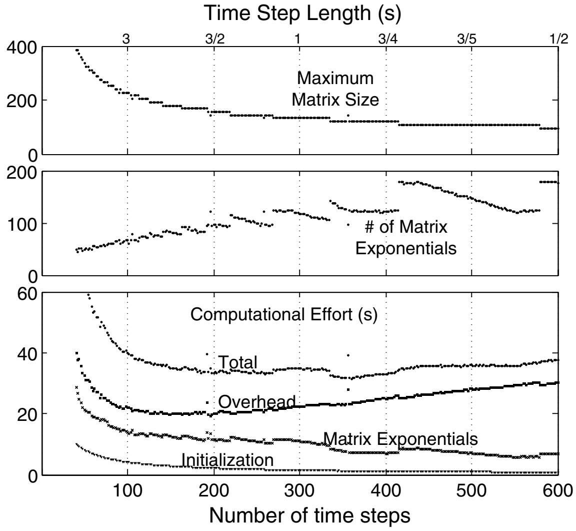

Fig. 5. Trade off between more and fewer time intervals in the multiple time interval FSP (MTI-FSP) algorithm solution for the toy heat shock model at a final time of $t_f = 300$ s. The following are plotted as function of the number of intervals: (top) the size of the largest required matrix exponential computation, (middle) the number of matrix exponential computations performed, (bottom) the computational time required for the MTI-FSP algorithm split into three components: the smallest is the cost of using 20 SSA to initialize the projection for the first time interval, the next smallest is total cost of computing matrix exponentials, and the largest is the remaining overhead costs (primarily data storage and retrieval). All computations have been performed in Matlab 7.2 on a Dual 2 GHz PowerPC G5.

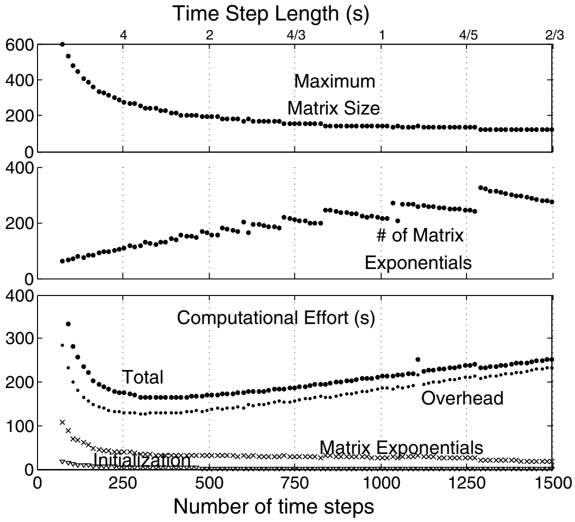

Fig. 6. Trade off between more and fewer time intervals in the multiple time interval FSP (MTI-FSP) algorithm solution for the toy heat shock model at a final time of $t_f = 1000$ s. See also Fig. 5.

$t_f = 1000$ s, the computational tradeoff is optimized for 360 time intervals corresponding to a interval length $\tau \approx 2.8$. To obtain the solution with this time interval, the algorithm needed to compute 131 matrix exponentials of size 228 × 228 or smaller, and the total computation takes about 164 s.

6. 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 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 interval FSP (MTI-FSP) algorithm, which is essentially an incremental approach to solving the original FSP.

The MTI-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 super-position 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 interval. By directly controlling the error at each time-interval, we can control the final error.

In addition to presenting the MTI-FSP algorithm, we have also developed a new approach to choosing and expanding the configuration space for an FSP analysis. While in the original FSP algorithm [16], the initial projection space was chosen arbitrarily, here we now utilize a few computationally inexpensive stochastic simulation runs to initialize the projection. Later, by systematically tracking the directions in which the probability measure leaks out of the projection space, we can more efficiently expand the projection space to recapture more of that probability measure.

We have demonstrated the MTI-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 1000 s, the original FSP algorithm would require a configuration space of over 11,000 points, which is too large to manage. However, we can now solve the problem using matrices less than one thirtieth of the size.

This time stepping approach is just one of the many mutually beneficial improvements that are quickly expanding the ability of the FSP to directly approximate the solution of 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.

Acknowledgements

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 US Army Research Office.

References

[1] D.T. Gillespie, A rigorous derivation of the chemical master equation, Physica A 188 (1992) 404–425.

[2] D. McQuarrie, "Stochastic approach to chemical kinetics, J. Appl. Probab. 4 (1967) 413–478.

[3] van Kampen, Stochastic Processes in Physics and Chemistry, third ed., Elsevier, 2001.

[4] D.T. Gillespie, Exact stochastic simulation of coupled chemical reactions, J. Phys. Chem. 81 (May) (1977) 2340–2360.

[5] E. Haseltine, J. Rawlings, Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics, J. Chem. Phys. 117 (Jul) (2002) 6959–6969.

[6] C.V. Rao, A.P. Arkin, Stochastic chemical kinetics and the quasi-steadystate assumption: application to the gillespie algorithm, J. Chem. Phys. 118 (Mar) (2003) 4999–5010.

[7] Y. Cao, D.T. Gillespie, L.R. Petzold, Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems, J. Comp. Phys. 206 (July) (2005) 395–411.

[8] Y. Cao, D. Gillespie, L. Petzold, The slow-scale stochastic simulation algorithm, J. Chem. Phys. 122 (Jan) (2005).

[9] H. Salis, Y. Kaznessis, Accurate hybrid stochastic simulation of a system of coupled chemical or biological reactions, J. Chem. Phys. 112 (054103) (2005).

[10] D.T. Gillespie, Approximate accelerated stochastic simulation of chemically reacting systems, J. Chem. Phys. 115 (Jul) (2001) 1716–1733.

[11] D.T. Gillespie, L.R. Petzold, Improved leap-size selection for accelerated stochastic simulation, J. Chem. Phys. 119 (Oct) (2003) 8229–8234.

[12] M. Rathinam, L.R. Petzold, Y. Cao, D.T. Gillespie, Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method, J. Chem. Phys. 119 (Dec) (2003) 12784–12794.

[13] T. Tian, K. Burrage, Binomial leap methods for simulating stochastic chemical kinetics, J. Chem. Phys. 121 (Dec) (2004) 10356–10364.

[14] Y. Cao, D.T. Gillespie, L.R. Petzold, Avoiding negative populations in explicit poisson tau-leaping, J. Chem. Phys. 123 (054104) (2005).

[15] Y. Cao, L.R. Petzold, Accuracy limitations and the measurement of errors in the stochastic simulation of chemically reacting systems, J. Comp. Phys. 212 (Feb) (2006) 6–24.

[16] B. Munsky, M. Khammash, The finite state projection algorithm for the solution of the chemical master equation, J. Chem. Phys. 124 (044104) (2006).

[17] B. Munsky, A. Hernday, D. Low, M. Khammash, Stochastic modeling of the pap-pili epigenetic switch, in: Proceedings of the FOSBE, August 2005, pp. 145–148.

[18] B. Munsky, M. Khammash, A reduced model solution for the chemical master equation arising in stochastic analyses of biological networks, in: Proceedings of the 45th IEEE CDC, Dec 2006.

[19] K. Burrage, M. Hegland, S. Macnamara, R. Sidje, A krylov-based finite state projection algorithm for solving the chemical master equation arising in the discrete modelling of biological systems, in: Proceedings of the A.A. Markov 150th Anniversary Meeting, 2006.

[20] S. Peles, B. Munsky, M. Khammash, Reduction and solution of the chemical master equation using time-scale separation and finite state projection, J. Chem. Phys. 125 (Nov) (2006).

[21] B. Munsky, S. Peles, M. Khammash, Stochastic analysis of gene regulatory networks using finite state projections and singular perturbation, in: Proceedings of the 26th American Control Conference, July 2007.

[22] H. El Samad, H. Kurata, J. Doyle, M. Khammash, C. Gross, Surviving heat shock: Control strategies for robustness and performance, PNAS 102 (8) (2005) 27362741.

[23] H. El Samad, M. Khammash, L. Petzold, D. Gillespie, Stochastic modeling of gene regulatory networks, Int. J. Robust Nonlinear 15 (2005) 691–711.