An improved finite state projection algorithm for the numerical solution of the chemical master equation with applications

S. MacNamara[1]   K. Burrage[2]   R. B. Sidje[3]

Abstract

Recently, Munsky and Khammash suggested the Finite State Projection (FSP) algorithm for the numerical solution of the Chemical Master Equation, which provides a discrete and stochastic modelling framework for chemical kinetics. The important question of whether or not the algorithm is guaranteed to terminate is not addressed in the original work. We show that the well-known explosive birth process provides a counter example. We also give sufficient criteria for a model to be suitable for the FSP technique. We demonstrate the FSP technique on three novel applications. Results are presented for: (i) the Schlögl reactions; (ii) another example from Gillespie's celebrated book; and (iii) models for the role that dimerization plays in reducing noise in simple gene regulatory networks. Finally, we augment the dimerization model to include tetramers and show that this enhances the noise reduction properties of the network.

Contents

1 Introduction

Computational and mathematical models of cellular processes promise great benefits in important fields such as molecular biology and medicine. Increasingly, researchers are incorporating the fundamentally discrete and stochastic nature of the biochemical processes into the mathematical models that are intended to represent them [1]. This has led to the formulation of models for genetic networks as continuous time, discrete state, Markov processes [7, 15], giving rise to the so-called chemical master equation (CME) that governs the evolution of the associated probability density. While promising many insights, the CME is computationally intensive, especially as the dimension of

the model grows. This article focuses on the recently suggested FSP algorithm [10], designed for the numerical solution of the chemical master equation, and describes situations for which it is appropriate as well as demonstrates its utility to some new applications.

The rest of this section sets up the mathematical framework of the CME that is used to model chemical kinetics and Section 2 reviews the FSP algorithm for its numerical solution. Section 3 describes how a careful enumeration of the states within the projection can improve the efficiency of the technique and discusses the conditions under which the algorithm is guaranteed to terminate, whereas Section 4.1 applies the algorithm to the Schlögl reactions, and models of dimerization.

1.1 Chemical kinetics as Markov processes

Many different kinds of mathematical equations have been proposed to model gene regulatory networks, such as boolean networks or PDEs—Burrage et al. [6] provide a good review. This article works within the framework of the CME [7, 15].

In this article a biochemical system consists of $N \geq 1$ different kinds of chemical species $\{S_1, \ldots, S_N\}$, interacting via $M \geq 1$ chemical reactions $\{R_1, \ldots, R_M\}$. It is assumed that the mixture has constant volume, is homogeneous and that the mixture is at thermal equilibrium. The system is modeled as a temporally homogeneous, continuous time, discrete state, Markov process. The state of the system is defined by the number of copies of each different kind of chemical species. Thus the state $\boldsymbol{x} \equiv (x_1, \ldots, x_N)^T$ is a vector of non-negative integers where x_i is the number of copies of species S_i. Each possible configuration of the system defines a distinct vector and so must be interpreted as a state in the Markov chain, thus defining the state space. Transitions between states occur when (and only when) a reaction occurs. Associated with each reaction R_j is a *stoichiometric* vector $\boldsymbol{\nu}_j$, of the

same dimension as the state vector, that defines the way the state changes when the reaction occurs: if the system is in state $\boldsymbol{x}$ and reaction j occurs, then the system transitions to state $\boldsymbol{x} + \boldsymbol{\nu}_j$. Associated with each state is a set of M *propensities*, $\alpha_1(\boldsymbol{x}), \ldots, \alpha_M(\boldsymbol{x})$, that determine the relative chance of each reaction occurring if the system is in state $\boldsymbol{x}$. The propensities are defined by the requirement that, given $\boldsymbol{x}(t) = \boldsymbol{x}$, $\alpha_j(\boldsymbol{x})dt$ is the probability of reaction j occurring in the next infinitesimal time interval $[t, t + dt)$, where the dependence on time has now been made explicit.

1.2   The chemical master equation

Having defined the Markov chain we now consider the probability density associated with it. Let the probability of being in state $\boldsymbol{x}$ at time t be denoted by $P(\boldsymbol{x}; t)$ and consider the way that this changes over time. For each state $\boldsymbol{x}$, the previous description of the model implies that this probability satisfies the following discrete, parabolic, PDE,

$$\frac{\partial P(\boldsymbol{x}; t)}{\partial t} = \sum_{j=1}^{M} \alpha_j(\boldsymbol{x} - \boldsymbol{\nu}_j) P(\boldsymbol{x} - \boldsymbol{\nu}_j; t) - P(\boldsymbol{x}; t) \sum_{j=1}^{M} \alpha_j(\boldsymbol{x}).$$

This chemical master equation may be written in an equivalent matrix-vector form so that the evolution of the probability density $\boldsymbol{p}(t)$ (which is a vector of probabilities $P(\boldsymbol{x}; t)$, indexed by the states $\boldsymbol{x}$) is described by a system of linear, constant coefficient, ordinary differential equations:

$$\dot{\boldsymbol{p}}(t) = \boldsymbol{A}\boldsymbol{p}(t),$$

where the matrix $\boldsymbol{A} = [a_{ij}]$ is populated by the propensities and represents the *infinitesimal generator* of the Markov process as defined by Stewart [14], for example. The rows and columns of the matrix are indexed by the states, so the states have now been implicitly enumerated. For $i \neq j$, the non-negative a_{ij} entry of the matrix gives the propensity for the system

to transition to state i, given that it is in state j, and the diagonal terms are $a_{jj} = -\sum_{i \neq j} a_{ij}$, which means the matrix has zero column sum and so probability is conserved. Given an initial density $\boldsymbol{p}(0)$, the solution at time t is the familiar matrix-exponential function

$$\boldsymbol{p}(t) = \exp(t\boldsymbol{A})\boldsymbol{p}(0)\,, \tag{1}$$

where the exponential of a bounded operator is usually defined via a Taylor series:

$$\exp(t\boldsymbol{A}) = \boldsymbol{I} + \sum_{n=1}^{\infty} \frac{(t\boldsymbol{A})^n}{n!}\,. \tag{2}$$

The numerical solution of (1), for the special class of matrices arising in biological applications, is the focus of this article, and numerical methods for linear ODEs [4] are closely related. The matrix exponential is well studied and many numerical methods are surveyed by Moler and Van Loan [9]. Since biological models may involve huge state spaces, special techniques are necessary. In particular, the FSP algorithm [10] proposed to *truncate* the system to a finite subsystem that captures enough of the information in the model while remaining tractable and is justified by the accompanying approximation theorems. This also seems intuitive from a biological point of view since a cell will only ever have a finite number of molecules of each species.

2   Review of the FSP algorithm

In the FSP algorithm the matrix in (1) is replaced by $\boldsymbol{A}_k$ where

$$\boldsymbol{A} = \left(\begin{array}{c|c} \boldsymbol{A}_k & * \\ \hline * & * \end{array} \right). \tag{3}$$

That is, $\boldsymbol{A}_k$ is a $k \times k$ submatrix of the true operator $\boldsymbol{A}$. The states indexed by $\{1,\ldots,k\}$ then form the *finite state projection*, denoted by $\boldsymbol{X}_k$. The FSP

algorithm then takes the form

$$\boldsymbol{p}(t_f) \approx \exp(t_f \boldsymbol{A}_k)\boldsymbol{p}_k(0) \,, \tag{4}$$

which is an approximation of (1) at the final time t_f. Here we use the subscript k to denote the truncation just described and note that a similar truncation is applied to the initial distribution. Munsky and Khammash [10] then consider the column sum

$$\Gamma_k = \mathbb{1}^T \exp(t_f \boldsymbol{A}_k)\boldsymbol{p}_k(0) \,, \tag{5}$$

where $\mathbb{1} = (1, \ldots, 1)^T$ with appropriate length. Normally the exact solution (1) would be a proper probability vector with unit column sum; however, due to the truncation, the sum Γ_k may be less than one because in the approximate system the probability sum condition is no longer conserved. Munsky and Khammash [10] showed that as k is increased, Γ_k increases too, so that the approximation gradually improves. Additionally, Munsky and Khammash show [10, Theorem 2.2] that, for small $\epsilon > 0$, if $\Gamma_k \geq 1 - \epsilon$ then

$$\begin{pmatrix} \exp(t_f \boldsymbol{A}_k)\boldsymbol{p}_k(0) \\ \boldsymbol{0} \end{pmatrix} \leq \boldsymbol{p}(t_f) \leq \begin{pmatrix} \exp(t_f \boldsymbol{A}_k)\boldsymbol{p}_k(0) \\ \boldsymbol{0} \end{pmatrix} + \epsilon \mathbb{1} \,.$$

This is the basis of the FSP algorithm outlined in Algorithm 1. Note that $\boldsymbol{X}_0$ is used for the set of states forming the initial projection, $\boldsymbol{X}_k$ for the projection at the kth step, $\boldsymbol{A}_k$ for the corresponding approximating matrix, and $\boldsymbol{p}_k(0)$ for the corresponding approximate initial distribution.

In the original example the state space projection is expanded simply by increasing k. More generally the FSP allows expanding the projection in a way that respects the reachability of the model. If, from state $\boldsymbol{x}$, reaction j occurs with strictly positive probability, we say that the state $\boldsymbol{x} + \boldsymbol{\nu}_j$ is *reachable* in a single step. Expanding by reachability refers to computing the subset of the state space that is reachable by a sequence of first one or less reactions, then two or less reactions, and so on. Thus, depending on the way the states are enumerated, expanding by reachability may mean that

---

Algorithm 1 FSP$(\boldsymbol{A}, \boldsymbol{p}_0(0), t_f, \epsilon, \boldsymbol{X}_0)$

1: $\boldsymbol{A}_0 := \mathrm{submatrix}(\boldsymbol{X}_0)$
2: $\Gamma_0 := \mathbb{1}^T \exp(t_f \boldsymbol{A}_0)\boldsymbol{p}_1(0)$
3: **repeat**
4:     increment k
5:     $\boldsymbol{X}_k := \mathrm{expand}(\boldsymbol{X}_{k-1})$
6:     $\boldsymbol{A}_k := \mathrm{submatrix}(\boldsymbol{X}_k)$
7:     $\Gamma_k := \mathbb{1}^T \exp(t_f \boldsymbol{A}_k)\boldsymbol{p}_k(0)$
8: **until** $\Gamma_k \geq 1 - \epsilon$
9: **return** $\exp(t_f \boldsymbol{A_k})\boldsymbol{p}_k(0)$

---

the principal submatrix of the true operator is not simply the intersection of the first k rows and columns.

In summary, given a fixed final time t_f and a starting state, the FSP algorithm gradually expands the projection around the initial state, via reachability. This method was recently improved to a Krylov based approach [5], by adapting Sidje's Expokit codes [12, 13] and that is the method we employ. Apart from Figure 1, which was done in MATLAB, all numerical experiments were conducted on the HPC located at the University of Queensland and running the Linux operating system. The system is known as 'Storm' and is an SGI Altix with 64 Itanium 2 CPU and 120 Gbytes of memory. However, only a single processor was used. The codes were developed in Fortran and compiled with the Intel ifort compiler.

3 Existence and structure of the projection

Section 3.3 shows that a suitable finite state projection does not always exist. In those cases where it does, we describe how exploiting the reachable structure in the Markov model and also tracking the region where most of

the probability density is concentrated, gives a more efficient algorithm.

3.1   Efficient enumerations of the state space

The state space of the Markov process associated with the CME may be stored in memory as a multidimensional array, with a particular state indexed by the number of molecules of each species. Often, only some fraction of the full state space is reachable and so a sparse storage technique is desirable. One approach is to use 'hand crafted' enumerations, as do Burrage et al. [5], while an alternative that was used here, is to store the states in a hash table. A Fortran hash table code of Brent's [2] was adapted for this purpose. This allowed efficient sparse storage and extraction of principal submatrices, as required in the next section.

3.2   A dynamic, nested projection

As observed by Burrage et al. [5], adaptively tracking the support of the distribution can speed up the algorithm significantly. Although empirically this approach gave good results, providing theoretical bounds for the accuracy of the approximation is difficult. Alternatively, there is a slight variation of that implementation, intermediate between the original FSP [10] and the inexact Krylov approach used by Burrage et al. [5], which allows a natural extension of the FSP theorems, so that error analysis is straightforward. In this variation the sequence of matrices A_k used at each integration step are still principal submatrices of A that correspond to the region of the state space where most of the support is concentrated but exact matrix vector products are used in the Arnoldi process. Requiring the conserved probability to be greater than $1 - \epsilon t / t_f$ at the step computing the solution at time t, and repeatedly applying the FSP theorems then gives a global error bound of ϵ at the final time t_f. We tested the approach in one preliminary numerical exper-

iment to provide evidence that the approach is worthwhile. The method was applied to the system of Michaelis–Menten enzyme kinetics [5], on which previous methods have been tested. The test problem was to compute the CME solution at $t_f = 20\,\text{s}$ with an initial condition of 100 substrates and 1000 enzymes. The original FSP [10] takes more than $200\,\text{s}$ to complete, while the adaptive method described here takes only $91\,\text{s}$. The differences between the computed solutions were about 10^{-6} and 10^{-5} in the 1-norm and ∞-norm respectively. Note that the FSP tolerance was set to $\epsilon := 10^{-5}$ in both cases which is consistent with the difference as measured by the ∞-norm. For this experiment the adaptive tracking was not performed at each step but only periodically and the nested projection was arbitrarily set to be about 40% of the size of the full state space. In the future a more sophisticated, robust implementation with an automated step-size control strategy is planned but even this simple strategy led to an improvement.

3.3   Existence counter example

Munsky and Khammash [10] demonstrated their new FSP algorithm on a Markov model (of the Pap-pili epigenetic switch) with a countably infinite state space, and suggest their method as a way of dealing with models with infinite or very large state spaces. Gillespie [7] also formulates some models of chemical reaction kinetics with infinite state spaces, and two of these are considered in the next section. In the case of finite models the existence of a suitable FSP is guaranteed and so the algorithm is guaranteed to terminate. Munsky and Khammash [10] seemed to imply that this would also be true for infinite models but this is not the case in general, as shown by the following example.

Consider the explosive birth process described, for example, by Norris [11, p.90]. Perhaps the simplest example for our purposes would be a continuous time, Markov process, on a discrete set of states indexed by the natural numbers $i \in \mathcal{N}$, such that i transitions to $i + 1$ with propensity i^2. In this
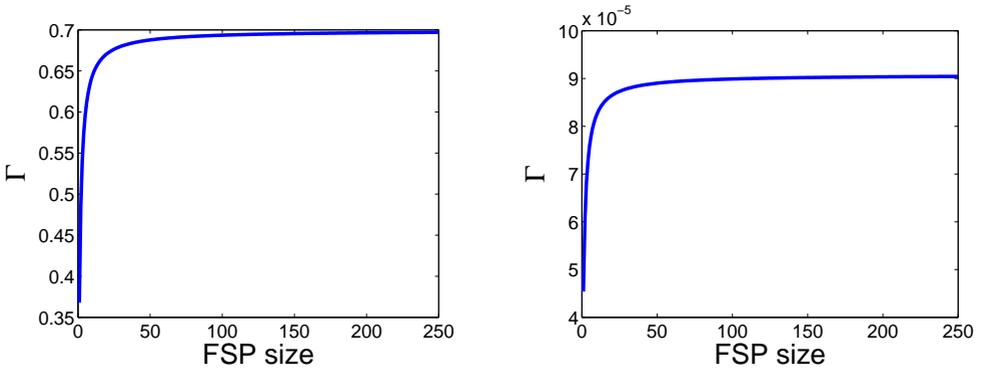
FIGURE 1: The FSP accuracy criteria for the explosive birth process described above, at $t_f = 1\,\mathrm{s}$ (left) and $t_f = 10\,\mathrm{s}$ (right). In each case the conserved probability is converging to a limit strictly less than unity.

case the chain explodes with probability 1 at a finite time. Since an infinite set of states are reachable in finite time, no finite subset will do for the FSP algorithm, and it does not converge when run on an example such as this. This provides a counter example and we see that the FSP algorithm does not always terminate, as shown in Figure 1. For a birth process like this example the probability of explosion increases with the final time t_f so the plot on the right of Figure 1 shows that much less probability is conserved by a finite projection than the plot on the left, which explains the large difference in the scales for Γ on these two plots. Criteria for a process to be non-explosive are also identified by Norris [11]. From a computational point of view, the simplest of these is that the diagonal of the matrix be bounded.

While on the subject of applying the FSP to infinite models, another technicality is identified. In the original FSP article the exponential of the linear operator is defined via the usual power series as do we in (2), but in the case of unbounded operators—and the model of the Pap-pili epigenetic switch that Munsky and Khammash use may be considered an example of this—it is not clear that this is well-defined. Kato [8, p.481], for example, discusses this

issue. However, we emphasize that, from the computational point of view, the FSP is a very reasonable approach. In particular, physically reasonable models arising in biological applications should be finite and bounded. Despite this, infinite models have been formulated previously, perhaps because a bound on the number of molecules in the system is not known. In such cases the truncated version of the full operator used by the FSP algorithm is always finite and bounded, and provides an approximation to the behaviour of the model.

4   Applications

Some novel applications of the Krylov FSP algorithm [5], to models that had previously been studied only via Gillespie's stochastic simulation algorithm (SSA) [7], are now presented.

4.1   The Schlögl reactions

In this section the FSP technique is applied to a pair of models of chemical kinetics described by Gillespie [7]. Gillespie felt that neither would be amenable to a numerical solution since the state space is infinite and so would require a corresponding infinite set of memory. As an example of the utility of the FSP algorithm we successfully apply it to these models, showing that the associated CME may indeed be solved to arbitrary precision with only a finite projection. Although these models are infinite, and thus subject to the considerations discussed in the previous section, the issue of existence, is resolved affirmatively merely by running the FSP algorithm. Consider the CME associated with the two chemical reaction systems described in Table 1. The solutions of the associated CME, as well as sample trajectories, are depicted in Figures 2, 3, 4 and 5. These are consistent with the simulations shown by Gillespie [7, Figures 6-17 and 5-4]. The solutions in Figure 3 show

Table 1: Reactions and rate constants for two examples from Gillespie's [7]. The B_i species are buffered and assumed constant in both examples. *Left:* Description of the Schlögl reactions [7, p.520]. The parameters were chosen to match that of Gillespie [7, Figure 6–17, p.525] so that the numbers of species B_1 and B_2 are $N_1 = 10^5$ and $N_2 = 2 \times 10^5$ respectively. *Right:* Description of a set of example reactions [7, p.362]. The parameters were chosen to match that of Gillespie [7, Figure 5–4, p.371] so that the numbers of species B_1, B_2 and B_3 were all 10^4 in simulation (a) and 10^6 in simulation (b).

Schlögl reactions		
1	$B_1 + 2X \xrightarrow{3.0E-7} 3X$	1	$B_1 + X \xrightarrow{0.025} X + X$
2	$B_1 + 2X \xleftarrow{1.0E-4} 3X$	2	$B_1 + X \xleftarrow{20.0} X + X$
3	$B_2 \xrightarrow{1.0E-3} X$	3	$X + X \xrightarrow{80.0} B_2 + B_3$
4	$B_2 \xleftarrow{3.5} X$	4	$X + X \xleftarrow{0.00025} B_2 + B_3$

the characteristic bimodal behaviour of the Schlögl reactions for intermediate initial numbers of species X, but not for very large or very small initial numbers. From top to bottom, the CME solutions presented in Figure 3 took 9 s, 240 s, 248 s and 554 s to compute, with final FSP sizes of about 1001, 2239, 2259 and 2801, for the initial numbers of species X as 0, 238, 258, and 800, respectively. For the second model taken from Gillespie [7, p.362], the CME solutions are shown in Figure 5. For part (a), for both initial conditions (initial numbers of species X were 0 and 70), the CME solution took less than 3 s to compute, with final FSP sizes of about 1100. For part (b), with initial numbers of species X at zero, the solution still took less than 3 s to compute with a final FSP size of around 3000, while with initial numbers of 7000, it took 335 s to compute the solution with a final FSP size of about 17000.
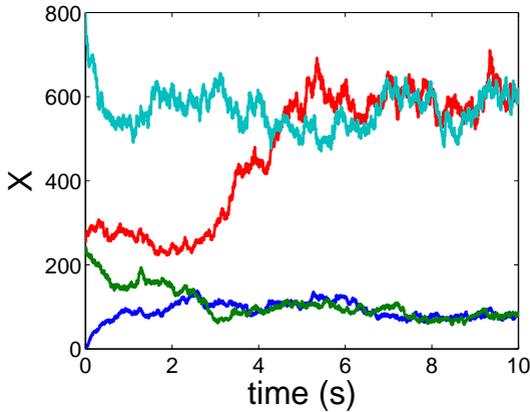
FIGURE 2: Four sample trajectories generated by the Stochastic Simulation Algorithm (SSA), for the Schlögl reactions, to match those given by Gillespie [7, Figure 6-17]. Initial numbers of species X were 0 (Blue), 238 (Green), 258 (Red), and 800 (Aqua).

4.2   Exploring noise reduction via dimerization

A study comparing models of gene regulatory networks, with and without dimerization [3], found that dimerization serves to buffer against stochastic fluctuations. These two models are defined in Table 2 and the result of an application of the FSP is shown in Figure 6.

The Fano factor, defined as the ratio of the variance to the mean, is used by Bundschuh et al. [3] to characterize and compare the fluctuations arising in the pathways with and without dimerization. The Fano factors of the CME solutions presented in Figure 6 are about 6.55 and 4.15, for the pathways with and without dimerization, respectively. Each is more than unity, indicating larger fluctuations than the Poisson process, and the monomeric pathway shows increased fluctuations compared to the dimeric pathway. Both of these results are consistent with the original findings but show that the same trend also holds for the transient solutions, although with
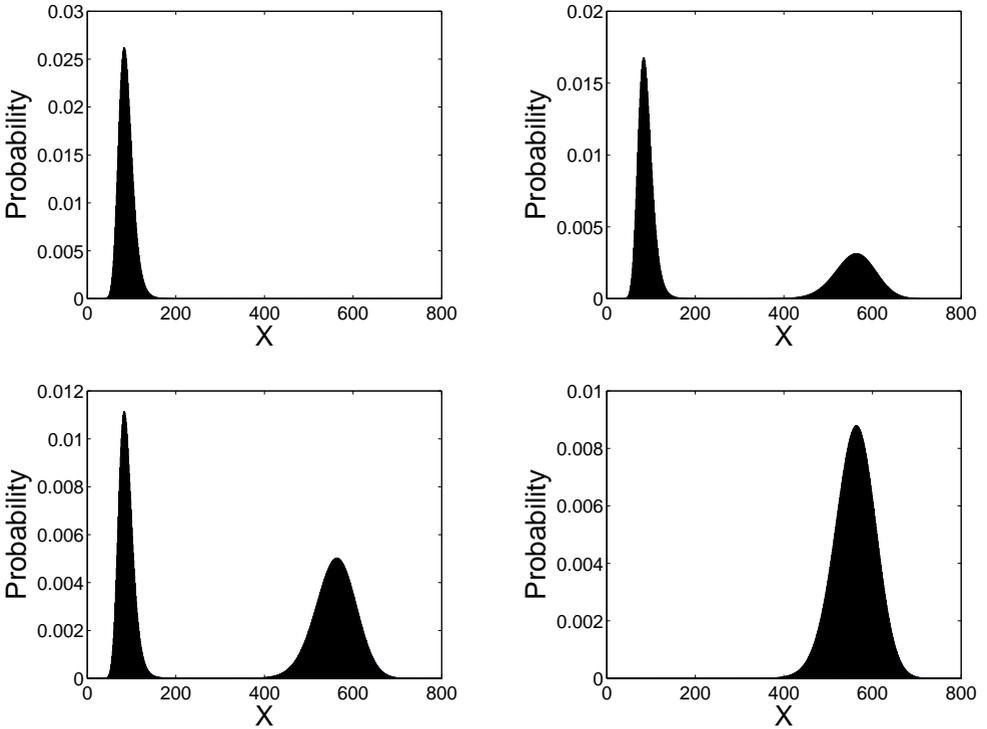
FIGURE 3:   CME solutions, for the Schlögl reactions at $t_f = 10\,\text{s}$, computed to within a tolerance of 10^{-5} via the FSP, with initial conditions matching SSA plots in Figure 2. Initial numbers of species X were, from top-left to bottom-right: 0, 238, 258 and 800.
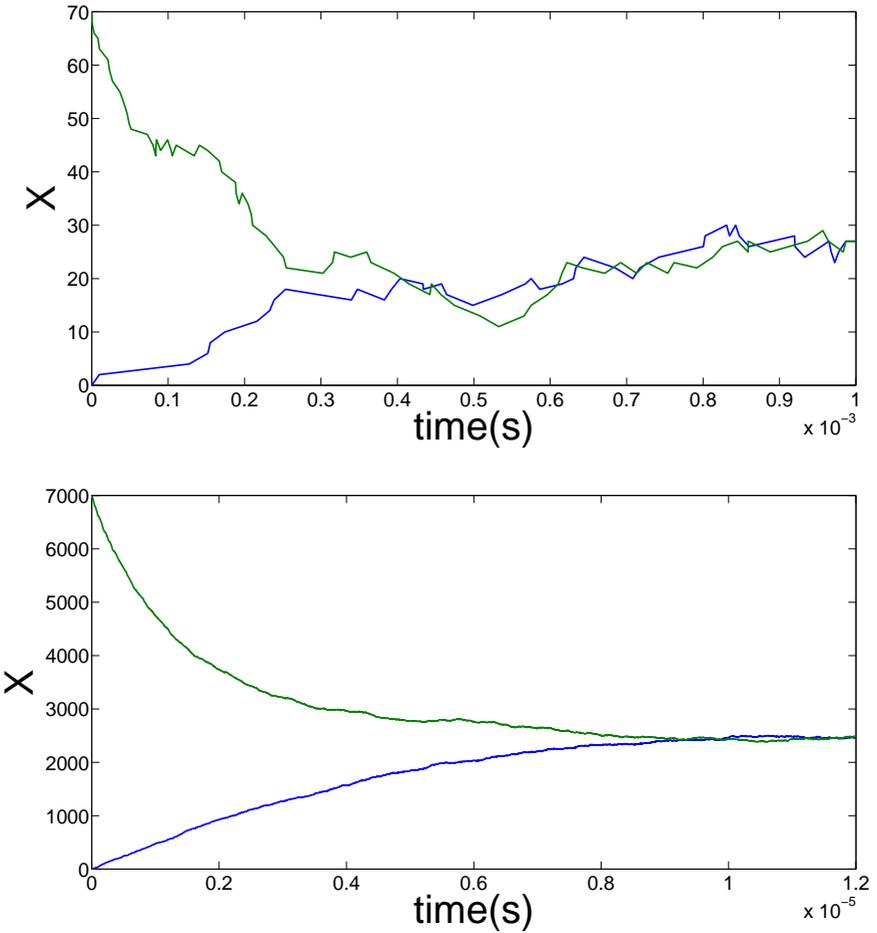
FIGURE 4: Four sample trajectories, generated by the Stochastic Simulation Algorithm (SSA), for the model given by Gillespie [7, p.362]. Parameters for top and bottom as in Table 1, (a) and (b), respectively. Initial numbers of species X were, for the top, 0 (blue) and 70 (green), and, for the bottom, 0 (blue) and 7000 (green).
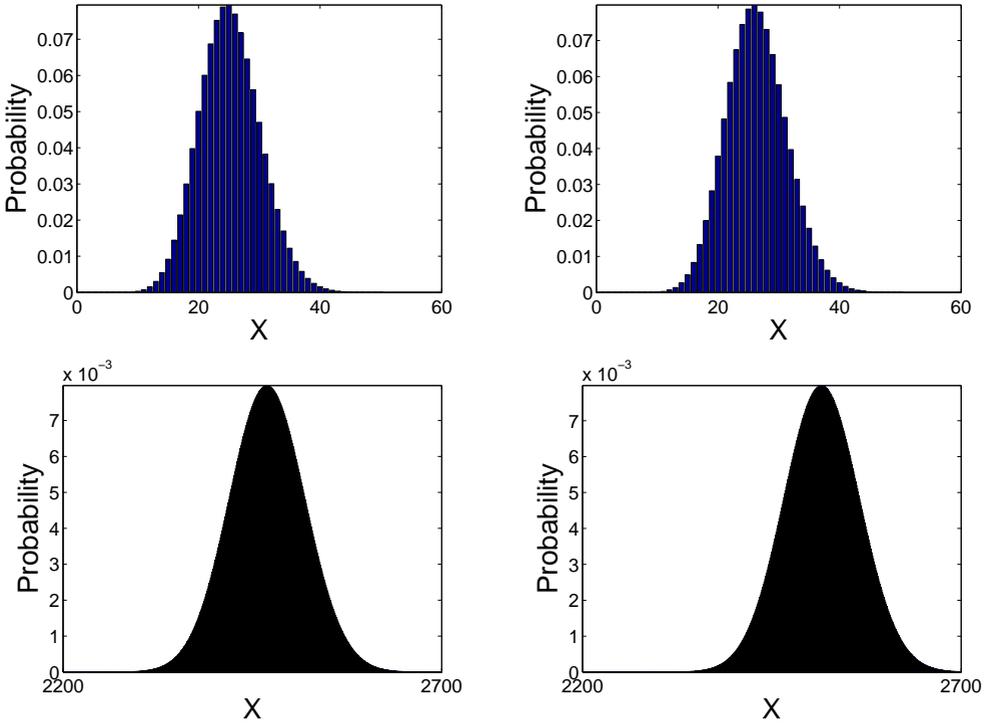
FIGURE 5: CME solutions, for the model given by Gillespie [7, p.362], computed to within a tolerance of 10^{-5} via the FSP, with initial conditions matching SSA plots in Figure 4. CME solutions computed at at $t_f = 1$ millisecond (top two plots) and at $t_f = 12$ microseconds (bottom two plots). Initial numbers of species X were, from top-left to bottom-right: 0, 70, 0 and 7000.

TABLE 2: Description of the reactions for two pathways, one with and one without dimerization, in a simple gene regulatory networks of Bundschuh et al. [3]. The first six reactions are common to both models. The number of DNA molecules is fixed at 1 and the number of RNA polymerases is fixed at $R = 30$. Parameters: $C = 2.0$, $k_1 = 0.0078$, $k_2 = 0.043$, $k_3 = 0.0039$, $k_4 = 0.0007$, $k_5 = 0.038$, $k_6 = 0.3$. Monomeric pathway: $k_7 = 0.00526\sqrt{C}$, $k_8 = 7.75/\sqrt{C}$, $k_9 = 0.2/\sqrt{C}$, $k_{10} = 0.2\sqrt{C}$. Dimeric Pathway: $k_7 = 0.025$, $k_8 = 0.5$, $k_9 = 0.012$, $k_{10} = 0.9$.

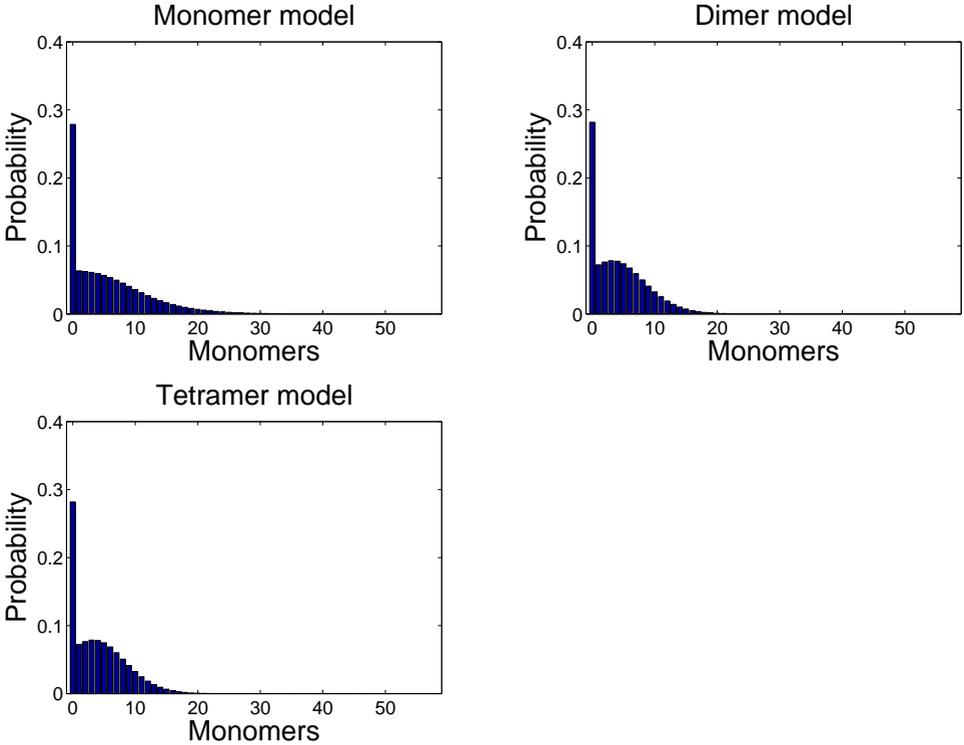|	Monomeric pathway		Dimeric pathway	
	reactions	propensities	reactions	props
1	$D^* \longrightarrow D + M + R$	k_1	$D^* \longrightarrow D + M + R$	k_1
2	$M \longrightarrow M + P$	$k_2 M$	$M \longrightarrow M + P$	$k_2 M$
3	$M \longrightarrow \emptyset$	$k_3 M$	$M \longrightarrow \emptyset$	$k_3 M$
4	$P \longrightarrow \emptyset$	$k_4 P$	$P \longrightarrow \emptyset$	$k_4 P$
5	$D + R \longrightarrow D^*$	$k_5 R$	$D + R \longrightarrow D^*$	$k_5 R$
6	$D + R \longleftarrow D^*$	k_6	$D + R \longleftarrow D^*$	k_6
7	$D + P \longrightarrow T$	$k_7 P$	$2P \longrightarrow P_2$	$k_7 \binom{P}{2}$
8	$D + P \longleftarrow T$	k_8	$2P \longleftarrow P_2$	$k_8 P_2$
9	$T + P \longrightarrow Q$	$k_9 P$	$D + P_2 \longrightarrow Q$	$k_9 P_2$
10	$T + P \longleftarrow Q$	k_{10}	$D + P_2 \longleftarrow Q$	k_{10}

FIGURE 6: CME solutions at $t_f = 250\,\mathrm{s}$, from an initial condition with populations of all species set to zero, computed via an FSP to within a tolerance of 10^{-5}, and visualized as the conditional distributions for the number of monomers. *Top:* Solutions to Bundschuh et al. [3] models of simple gene regulatory networks, one with dimerization and one without. *Bottom:* Solution to the dimeric model, augmented with tetramers.

significantly larger fluctuations. We also investigated, in a third model, how dimer and tetramer mechanisms compare (the former allowing dimerization in solution and the latter allowing both dimerization and tetramerization in solution) in terms of fluctuations. The dimeric pathway described in Table 2 was augmented to allow the formation of tetramers, increasing the number of chemical species by one and the number of reactions by two. In this new model the operator is bound by the tetramers, instead of the dimers. With the propensity functions for the formation and dissociation of tetramers being analogous to that for dimerization (and with the same constants) the CME solution at $t_f = 250\,\mathrm{s}$ was computed via an FSP algorithm. The Fano factor was about 4.01, which is slightly smaller than that of the dimeric model. So, at least for this choice of parameter values, the effect of introducing tetramers is to further reduce the fluctuations. This comparison highlights another reason why cells may introduce tetramers and, more generally, higher order multimers as a mechanism for controlling fluctuations in simple gene regulatory networks.

5   Conclusions

As well as making some observations about the situations for which the FSP algorithm is appropriate—and in particular identifying the criteria under which the algorithm is guaranteed to terminate—the FSP has been successfully demonstrated on three new applications. Although these new applications involve models of only a modest size, it shows that the CME approach is feasible for these and others, where previously this was felt not to be the case. In relation to dimerization, the results largely confirm the original findings of Bundschuh et al. [3], via a different approach. Also, transient solutions were considered, where previously the focus had been on equilibrium solutions. Furthermore, the dimeric pathway model has been extended here to explore the effects of introducing tetramers. This raises an interesting question about whether cells use the introduction of higher order multimers as a mechanism

for combating noise.

References

[1] A. Arkin and H. McAdams. It's a noisy business! Genetic regulation at the nanomolar scale. *Trends Genet.*, 15(2):65–69, 1999. doi:10.1016/S0168-9525(98)01659-X. C414

[2] R. P. Brent. Reducing the retrieval time of scatter storage techniques. *Communications of the ACM*, 16(2):105 –109, February 1973. doi:10.1145/361952.361964. C420

[3] R. Bundschuh, F. Hayot, and C. Jayaprakash. The role of dimerization in noise reduction of simple genetic networks. *J. Theor. Biol.*, 220:261–269, 2003. doi:10.1016/j.jtbi.2004.05.013. C425, C429, C430, C431

[4] K. Burrage. *Parallel and sequential methods for ordinary differential equations.* Oxford University Press, Oxford, 1995. C417

[5] K. Burrage, M. Hegland, S. MacNamara, and R. B. Sidje. A Krylov-based finite state projection algorithm for solving the chemical master equation arising in the discrete modelling of biological systems. In A. N. Langville and W. J. Stewart, editors, *150th Markov Anniversary Meeting, Charleston, SC, USA*, pages 21–38. Boson

Books, 2006.
http://www.bosonbooks.com/boson/nonfiction/mam/mam.html.
C419, C420, C421, C423

[6] K. Burrage, T. Tian, and P. Burrage. A multi-scaled approach for simulating chemical reaction systems. *Progress in Biophysics and Molecular Biology*, 85:217–234, 2004.
doi:10.1016/j.pbiomolbio.2004.01.014. C415

[7] D. T. Gillespie. *Markov Processes: An Introduction for Physical Scientists*. Academic Press, Harcourt Brace Jovanovich, 1992. C414, C415, C421, C423, C424, C425, C427, C428

[8] T. Kato. *Perturbation theory for linear operators*. Springer–Verlag, 1976. C422

[9] C. Moler and C. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Review*, 45(1):3–49, 2003.
http://www.siam.org/journals/sirev/45-1/41801.html. C417

[10] B. Munsky and M. Khammash. The finite state projection algorithm for the solution of the chemical master equation. *J. Chem. Phys.*, 124:044104, 2006. doi:10.1063/1.2145882. C415, C417, C418, C420, C421

[11] J. R. Norris. *Markov chains*. Cambridge, 1997. C421, C422

[12] R. B. Sidje. Expokit: A software package for computing matrix exponentials. *ACM Transactions on Mathematical Software*, 24(1):130–156, 1998. http://www.expokit.org. C419

[13] R. B. Sidje and W. J. Stewart. A numerical study of large sparse matrix exponentials arising in Markov chains. *Comput. Statist. Data Anal.*, 29:345 – 368, 1999. doi:10.1016/S0167-9473(98)00062-0. C419

[14] W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains.* Princeton University Press, 1994. C416

[15] N. G. van Kampen. *Stochastic Processes in Physics and Chemistry.* Elsevier Science, 2001. C414, C415

Author addresses

1. **S. MacNamara**, Advanced Computational Modelling Centre, & Dept. of Mathematics, University of Queensland, Brisbane, Australia.
   mailto:shev@maths.uq.edu.au

2. **K. Burrage**, Advanced Computational Modelling Centre, and Dept. of Mathematics, University of Queensland, Brisbane, Australia.

3. **R. B. Sidje**, Advanced Computational Modelling Centre, & Dept. of Mathematics, University of Queensland, Brisbane, Australia.