Stochastic chemical kinetics and the *total* quasi-steady-state assumption: application to the stochastic simulation algorithm and chemical master equation

Alberto M. Bersani *
Department of Mathematical Methods and Models, "La Sapienza" University, Rome, Italy

Kevin Burrage †, Shev MacNamara ‡, and Roger B. Sidje §
Advanced Computational Modelling Centre and The Department of Mathematics, The University of Queensland, Australia.

(Dated: May 9, 2007)

Recently the application of the quasi-steady-state approximation (QSSA) to the Gillespie algorithm was suggested for the purpose of speeding up stochastic simulations of chemical systems that involve both relatively fast and slow chemical reactions [Rao and Arkin, J. Chem. Phys. **118**, 4999 (2003)]. Improved numerical efficiency is obtained by respecting the vastly different time scales characterizing the system and then by advancing only the slow reactions exactly, based on a suitable approximation to the fast reactions. We considerably extend the work of Rao and Arkin by applying it to numerical methods for the direct solution of the Chemical Master Equation (CME) and in particular to the Krylov Finite State Projection algorithm. In addition we extend the mathematical framework of the approximation scheme and point out some important connections to the literature on the (deterministic) total QSSA (tQSSA) and place the stochastic analogue of the QSSA (in all forms) within the more general framework of aggregation of Markov processes, which naturally suggests a family of related numerical methods. We apply the new methods to Michaelis-Menten enzyme kinetics, competitive inhibition, and a component of the λ-phage genetic switch and two further models of enzyme kinetics arising in the mitogen-activated-protein kinase cascade: a model of the dual phosphorylation scheme and the Goldbeter-Koshland switch. Overall we report dramatic improvements by applying the tQSSA to the CME-solver.

I. INTRODUCTION

Chemical kinetics are often modeled by ordinary differential equations (ODEs), but under some circumstances – for example, when some species are present in small numbers[1,2] – a discrete and stochastic framework is more appropriate.[3] Such a framework is provided by continuous-time, discrete-state Markov processes. Markov models of the bacteriophage λ life cycle have been a flagship for the success of this approach.[4]

A very popular method for studying and simulating intrinsic noise is the Stochastic Simulation Algorithm (SSA).[5,6] However the SSA can become too slow in the presence of large molecular populations or rate constants, thus motivating the τ-Leap approximation,[7] accelerated leap methods[8–10] and more generally, multiscale methods for simulating biochemical kinetics.[11,12]

In the presence of both fast and slow reactions the quasi-steady-state approximation (QSSA) has been one such multiscale method that has recently received much attention for the purpose of speeding up the SSA.[13–16] Here we investigate its application to the direct solution of the chemical master equation (CME), which describes the evolution of the probability mass function associated with the SSA. Significantly we are able to adapt a CME-solver, based on Krylov methods,[17–19] by incorporating a type of QSSA and thus take advantage of the multiscale nature of the systems being studied.

This paper is organized as follows. First we discuss the mathematical framework of the CME and then review previous works that have applied the QSSA to the SSA. We then give an analysis of the QSSA as applied to the CME and distinguish among different forms of the QSSA, with particular reference to the Michaelis-Menten system. This leads to the development of some novel numerical methods for the solution of the CME and the details of their implementation are discussed. We report the results of applying these new methods to various models of enzyme kinetics and finally give a discussion of the strengths and limitations of this work.

II. BACKGROUND TO MODELS OF BIOCHEMICAL KINETICS

In this section a review of the way that chemical kinetics are modeled as Markov processes is given as well as a summary of the numerical methods that are commonly used to study them.

A. Chemical kinetics as Markov processes

The framework of the CME[3,6] is now described. In this paper a biochemical system consists of $N \geq 1$ different kinds of chemical species $\{S_1, \ldots, S_N\}$, inter-

*bersani@dmmm.uniroma1.it
†kb@maths.uq.edu.au
‡shev@maths.uq.edu.au
§rbs@maths.uq.edu.au

acting via $M \geq 1$ chemical reactions $\{R_1, \ldots, R_M\}$. It is assumed that the mixture has constant volume, is homogeneous and that it is at thermal equilibrium. The system is modeled as a temporally homogeneous, continuous-time, discrete-state, Markov process. While macro-molecular crowding effects leading to anomalous diffusion can be significant when describing processes on the membrane of a cell or within a cell,[20,21] this framework has proved to be successful in a number of biological settings.[4] The state of the system is defined by the number of molecules of each 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, Ω. Transitions between states occur 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 a 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 $[t, t + dt)$.

B. The SSA and leap methods

In a seminal work the SSA[5,6] was suggested for simulating a continuous-time, discrete state Markov process exactly, in the sense of faithfully sampling paths from the model with an appropriate distribution. The SSA simulates the system one reaction at a time. At each step, it samples from an exponential distribution, the waiting time until the next reaction occurs, and, from a uniform distribution, to determine the reaction number based on the relative sizes of the propensity functions. However, in situations where there are large numbers of some of the chemical species or large propensities, the time step may become very small and the SSA becomes too slow. The (Poisson) τ-Leap approximation[7] was proposed, which speeds up the simulation by leaping forward through a much larger interval in time, with the number of times a reaction fires being drawn from the Poisson distribution. Following this idea, the mid-point τ-Leap method,[7] implicit τ-Leap method[22] and Poisson Runge-Kutta method[11] have been introduced. Poisson random variables are nonnegative but unbounded so without very careful step size strategies[23] it is possible that such a procedure may predict negative numbers of some molecular species. In another approach to avoiding negative molecular numbers, Tian and Burrage[10] sample from the binomial distribution, since binomial random variables have a finite range and may well approximate a Poisson random variable for certain parameter ranges.

C. The chemical master equation

Rather than simulating a path through the Markov process, we can, given an initial condition $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$, directly compute the probability of being in state $\boldsymbol{x}$ at time t, $P(\boldsymbol{x}; t)$, and consider the way that this changes over time. It can be shown that for each state $\boldsymbol{x}$, the previous description of the model implies that this probability satisfies the following discrete PDE,

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

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, with $a_{jj} = -\sum_{i \neq j} a_{ij}$. This means the matrix has zero column sum and so probability is conserved. Given an initial distribution $\boldsymbol{p}(0)$, the solution at time t is

$$\boldsymbol{p}(t) = \exp(t\boldsymbol{A})\boldsymbol{p}(0), \tag{2}$$

where the exponential of a bounded operator is usually defined via a Taylor series: $\exp(t\boldsymbol{A}) = \boldsymbol{I} + \sum_{n=1}^{\infty} \frac{(t\boldsymbol{A})^n}{n!}$. We note that the matrix exponential is well studied[24] and numerical methods for linear ODEs[25] are closely related. There are some technical considerations when the system is infinite.[19] For example, the operator may be unbounded and the power-series representation is not appropriate.[26] Also, the well-known explosive birth process[27] provides an example of an infinite model for which the algorithms used in this paper may not terminate. However for biological applications, physically reasonable models should be finite and bounded. The Finite State Projection (FSP) algorithm[28] uses a truncated version of the full operator, which is always finite and bounded, and which provides an approximation to the behaviour of the model.

III. THE FSP ALGORITHM

In the FSP algorithm the matrix in (2) is replaced by $\boldsymbol{A}_k$ where

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

i.e. $\boldsymbol{A}_k$ 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*, which will be denoted by $\boldsymbol{X}_k$. The FSP algorithm replaces (2) with the approximation

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

The subscript k denotes the truncation just described and note that a similar truncation is applied to the initial distribution. Munsky and Khammash[28] then consider the column sum

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

where $\mathbb{1} = (1, ..., 1)^T$ with appropriate length. Normally the exact solution (2) 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[28] showed that as k is increased, Γ_k increases too, so that the approximation is gradually improved. Additionally it is shown in Theorem 2.2 of Ref. 28 that if

$$\Gamma_k \geq 1 - \epsilon \tag{6}$$

for some pre-specified tolerance ϵ, then we have

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

Algorithm 1 summarizes the FSP. Note that $\boldsymbol{X}_0$ is used for the set of states forming the initial projection, $\boldsymbol{X}_k$ is the projection at the k^{th} step, $\boldsymbol{A}_k$ is the corresponding approximating matrix and $\boldsymbol{p}_k(0)$ is the corresponding approximate initial distribution.

In the original implementation the state-space projection is expanded simply by increasing k. More generally the FSP allows expanding the states in a way that respects the reachability of the model.[28]

ALGORITHM 1: FSP$(\boldsymbol{A}, \boldsymbol{p}_0(0), t_f, \epsilon, \boldsymbol{X}_0)$
$\boldsymbol{A}_0 := \mathit{submatrix}(\boldsymbol{X}_0)$;
$\Gamma_0 := \mathbb{1}^T \exp(t_f \boldsymbol{A}_0)\boldsymbol{p}_1(0)$;
for $k := 1, 2, ...$ **until** $\Gamma_k \geq 1 - \epsilon$ **do**
  $\boldsymbol{X}_k := \mathit{expand}(\boldsymbol{X}_{k-1})$;
  $\boldsymbol{A}_k := \mathit{submatrix}(\boldsymbol{X}_k)$;
  $\Gamma_k := \mathbb{1}^T \exp(t_f \boldsymbol{A}_k)\boldsymbol{p}_k(0)$;
endfor
return $\exp(t_f \boldsymbol{A_k})\boldsymbol{p}_k(0)$;

This method was recently improved to a Krylov-based approach,[17–19] by adapting Sidje's Expokit codes.[29,30]

IV. THE KRYLOV FSP ALGORITHM

We now describe a very efficient modification of the FSP that uses Krylov methods, inexact matrix-vector products and adaptively tracking the support of the distribution. As well as being a matrix-free approach, it allows the concurrent expansion of the state-space and evaluation of the exponential.

A. Krylov-based exponential computation

The Krylov FSP outlined in Algorithm 2 is based around Expokit.[29,30] It converts the problem of exponentiating a large sparse matrix to that of exponentiating a small, dense matrix in the Krylov subspace. The dimension m of the Krylov subspace is typically small, and $m = 30$ was used in this implementation (it may be changed adaptively during the FSP process but this is not considered here). The Krylov approximation to $\exp(\tau \boldsymbol{A})\boldsymbol{v}$ being used is

$$\beta \boldsymbol{V}_{m+1} \exp(\tau \overline{\boldsymbol{H}}_{m+1})\boldsymbol{e}_1 \tag{7}$$

where $\beta \equiv \|\boldsymbol{v}\|_2$, $\boldsymbol{e}_1$ is the first unit basis vector, and $\boldsymbol{V}_{m+1}$ and $\overline{\boldsymbol{H}}_{m+1}$ are the orthonormal basis and upper Hessenberg matrix resulting from the well-known Arnoldi process. The exponential in the smaller subspace is computed via the diagonal Padé approximation with degree $p = 6$, together with scaling and squaring (implemented in a way that is slightly more efficient than usual[29]).

B. Embedded exponential computation

Rather than simply being a mere substitution of *MATLAB*'s *expm* in the original FSP algorithm with the Krylov-based variant as one may think at first, there is actually a deeper improvement to be stressed. Unlike the original FSP algorithm that repeatedly computes $\exp(t_f \boldsymbol{A}_k)\boldsymbol{p}_k(0)$ with the *same* t_f, until $\boldsymbol{A}_k$ is sufficiently large, the new solver uses the embedded scheme (with vectors padded with zeros to be of consistent sizes as appropriate)

$$\boldsymbol{p}(t_f) \approx \exp(\tau_K \boldsymbol{A}_K) \ldots \exp(\tau_0 \boldsymbol{A}_0)\boldsymbol{p}(0)\,, \quad t_f = \sum_{k=0}^{K} \tau_k, \tag{8}$$

so that

$$\boldsymbol{p}(t_{k+1}) = \boldsymbol{p}(t_k + \tau_k) = \exp(\tau_k \boldsymbol{A}_k)\boldsymbol{p}(t_k), \tag{9}$$

where the $\{\tau_k\}$ are step-sizes and K denotes the total number of steps needed. Literally, the improved FSP scheme (8) is evaluated from right to left, harnessing the built-in step-by-step integration procedure of Expokit, with the special feature that the matrix changes between these internal integration steps.

The Krylov FSP method inherits Expokit's automatic step-size control strategy to ensure the accuracy of the Krylov approximation but in order to ensure accuracy

of the FSP approximation too, an additional check is required at each step. More precisely it is required that

$$\Gamma_k = \mathbb{1}^T \boldsymbol{p}(t_k) \geq f(t_k) \equiv 1 - \epsilon \frac{t_k}{t_f}. \qquad (10)$$

If this fails, then the time-step τ_k is repeatedly halved until the criterion is met. Equation (10) is a natural generalization of (6), and represents the repeated application of the FSP Theorems,[28] at intermediate time steps. In order to meet the global accuracy requirement (6), the local accuracy requirements (10) must be more stringent.

The new algorithm expands the FSP projection if and only if the previous step did not initially satisfy (10). By default, in the implementation used for this paper, the projection is expanded in level sets of reachability,[28] by ten steps at a time, and is initialized to a minimum size of 2500, although these parameters can be adjusted to suit the problem.

ALGORITHM 2: KrylovFSP($\boldsymbol{A}, \boldsymbol{p}_0(0), t_f, \epsilon, \boldsymbol{X}_0, m, tol$)
$t_0 := 0$; $\hat{\boldsymbol{X}}_0 := \boldsymbol{X}_0$;
$\boldsymbol{A}_0 := submatrix(\boldsymbol{X_0})$; $\hat{\boldsymbol{A}}_0 := \boldsymbol{A}_0$;
expandFSP := FALSE ;
for $k := 0, 1, 2, ...$ **until** $t_k = t_f$ **do**
  $[\boldsymbol{V}_{m+1}, \overline{\boldsymbol{H}}_{m+1}] := Arnoldi\ (\hat{\boldsymbol{A}}_k, \boldsymbol{p}_k(t_k), m)$;
  repeat { enforce numerical accuracy }
    $\tau_k :=$ step-size ;
    $\boldsymbol{p}_k(t_k) := \beta \boldsymbol{V}_{m+1} \exp(\tau_k \overline{\boldsymbol{H}}_{m+1}) \boldsymbol{e}_1$;
    err := numerical-error-estimate ;
  until err $\leq 1.2\ tol$;
  $\Gamma_k := \mathbb{1}^T \boldsymbol{p}_k(t_k)$;
  while $\Gamma_k < 1 - \epsilon \frac{t_k + \tau_k}{t_f}$ **do**
    { enforce FSP criterion }
    expandFSP := TRUE ;
    $\tau_k := \frac{1}{2} \tau_k$;
    $\boldsymbol{p}_k(t_k) := \beta \boldsymbol{V}_{m+1} \exp(\tau_k \overline{\boldsymbol{H}}_{m+1}) \boldsymbol{e}_1$;
    $\Gamma_k := \mathbb{1}^T \boldsymbol{p}_k(t_k)$;
  endwhile
  if expandFSP
    $\boldsymbol{X}_{k+1} := expand(\boldsymbol{X}_k)$;
    $\hat{\boldsymbol{X}}_{k+1} := select(\boldsymbol{X}_{k+1})$;
    $\boldsymbol{A}_{k+1} := submatrix(\boldsymbol{X}_{k+1})$;
    $\hat{\boldsymbol{A}}_{k+1} := submatrix(\hat{\boldsymbol{X}}_{k+1})$;
    expandFSP := FALSE ;
  endif
  $t_{k+1} := t_k + \tau_k$;
endfor
return $\boldsymbol{p}(t_k)$

C. Inexact matrix-vector product

The Krylov FSP is capable of simultaneously keeping track of a pair of projections, $\boldsymbol{X}_k$ and $\hat{\boldsymbol{X}}_k$, one nested inside the other: $\hat{\boldsymbol{X}}_k \subseteq \boldsymbol{X}_k$. The larger projection, $\boldsymbol{X}_k$, grows monotonically and represents the original FSP projection that would be required if the current time were to be the final time. Inside this is a second projection, $\hat{\boldsymbol{X}}_k$, essentially capturing a suitably large proportion of the support of the probability distribution at that instant. This nested projection is used to define another submatrix $\hat{\boldsymbol{A}}_k$ within $\boldsymbol{A}_k$, which is used to form the approximation (9), i.e., $\hat{\boldsymbol{A}}_k$ is used in the matrix-vector products required by the Arnoldi process, making it an inexact Krylov method.[31,32] The nested projection need not grow monotonically; indeed it may also shrink, as seen in the Michaelis-Menten enzyme kinetics model in Section VI F. The smaller this nested projection, the smaller the corresponding submatrix $\hat{\boldsymbol{A}}_k$, making the algorithm highly efficient in such circumstances. However for some models the computational overhead of tracking $\hat{\boldsymbol{X}}_k$ is more expensive than the pay-off of using a smaller matrix, $\hat{\boldsymbol{A}}_k$.

V. APPLICATION OF THE QSSA TO THE SSA

Dramatic improvements to computational efficiency were reported at the cost of only an acceptable loss in accuracy, by applying the QSSA to the SSA.[13–16] The idea behind these algorithms is to advance only the 'slow' reactions exactly, based on a suitable approximation to the 'fast' reactions, which rapidly reach a quasi-equilibrium, allowing large time steps to be taken.

In Ref. 15 a multiscale approach is adopted, similar to that advocated by Burrage *et. al.*[11] and a mixture of three modeling regimes are used: ODEs, stochastic differential equations (SDEs) and the SSA, depending on how many molecules there are. The choice of which regime is to be used varies dynamically with the evolution of the system state.

These approximations can also be considered from the perspective of the CME. Associated with each of the new approximations is the evolution of some probability mass function, which satisfies some approximate CME of lower dimension than the true CME governing the full system, and this reduction in the dimension of the model is where computational savings can be achieved. In a good example of such an approach Lötstedt and Ferm[12] use a Fokker-Planck approximation.

VI. APPLICATION OF THE QSSA TO THE CME

Motivated by the previously cited successful examples of the application of the QSSA to the SSA, we now develop the QSSA in the context of numerical methods for the solution of the CME. Previous works[12,33] have considered related ideas but the methods presented here are based on Krylov methods combined with aggregation.

The CME is itself a system of ODEs so at first glance one may consider merely applying the QSSA from the ODE setting in the usual way. However in the context of the CME, each ODE now governs the *probability* of a particular state, not the concentration of a particular species, and the state space is generally enormous. These differences mean that it is worth first considering how to customize the QSSA for application to the CME.

First we present a general analysis, roughly summarizing Rao and Arkin. We begin by partitioning the set of reactions. Let $R_f \subset \{R_1, \ldots, R_M\}$ denote the subset of fast reactions and let R_s denote the rest. The CME (1) can then be re-written by splitting the right hand side into two parts – one for the fast reactions and one for the slow reactions. For example, the fast reactions gives rise to the following 'fast CME',

$$\frac{\partial P(\boldsymbol{x};t)}{\partial t} = \sum_{j\in R_f} \alpha_j(\boldsymbol{x}-\boldsymbol{\nu}_j)P(\boldsymbol{x}-\boldsymbol{\nu}_j;t) - P(\boldsymbol{x};t)\sum_{j\in R_f}\alpha_j(\boldsymbol{x}). \tag{11}$$

An analogous 'slow CME' arises for the slow reactions and summing the two recovers (1). Next we define an induced partition of the species. A species is deemed to be fast if its population is changed by at least one fast reaction; otherwise the species is deemed to be slow. The state vector is likewise partitioned into $\boldsymbol{x} = [\boldsymbol{z}, \boldsymbol{y}]$, where $\boldsymbol{z}$ corresponds to the vector of fast species and $\boldsymbol{y}$ to the slow species and similarly for the stoichiometric vectors, $\boldsymbol{\nu} = [\boldsymbol{\nu}^z, \boldsymbol{\nu}^y]$. With this, we introduce the conditional distribution

$$P(\boldsymbol{z}, \boldsymbol{y}) = P(\boldsymbol{z}|\boldsymbol{y})P(\boldsymbol{y})$$

and using the chain rule of differentiation the CME can be re-written as

$$P(\boldsymbol{y};t)\frac{\partial P(\boldsymbol{z}|\boldsymbol{y};t)}{\partial t} + P(\boldsymbol{z}|\boldsymbol{y};t)\frac{\partial P(\boldsymbol{y};t)}{\partial t} = \sum_j \alpha_j([\boldsymbol{z},\boldsymbol{y}]-[\boldsymbol{\nu}^z,\boldsymbol{\nu}^y])P(\boldsymbol{z}-\boldsymbol{\nu}_j^z|\boldsymbol{y}-\boldsymbol{\nu}_j^y;t)P(\boldsymbol{y}-\boldsymbol{\nu}_j^y;t) - P(\boldsymbol{z}|\boldsymbol{y};t)P(\boldsymbol{y};t)\sum_j \alpha_j([\boldsymbol{z},\boldsymbol{y}])\ . \tag{12}$$

So far everything has been exact, but we now introduce the approximation

$$\frac{\partial P(\boldsymbol{z}|\boldsymbol{y})}{\partial t} \approx 0, \qquad P(\boldsymbol{z}|\boldsymbol{y}) \approx \hat{P}(\boldsymbol{z}|\boldsymbol{y})\ .$$

This approximation is intended to be the analogue of the QSSA in the deterministic setting. The approximation on the left is intended to capture the intuition that, given the fixed values of the slow species, the fast species rapidly settle down to equilibrium. In order to find $\hat{P}(\boldsymbol{z}|\boldsymbol{y})$, we set the left hand side of (11) to zero and solve the resulting equation. It is sometimes possible to solve this analytically since the resulting equation is simpler than the full CME and in particular only involves fast reactions, with the values of the slow species being fixed. Substituting this approximation into (12) gives rise to a reduced CME that involves only the slow reactions, and in which the propensities of these slow reactions are modified according to the choice of $\hat{P}(\boldsymbol{z}|\boldsymbol{y})$.

Very similar general principles to the above analysis underlay all of the works in which the QSSA is applied to the CME[13–16] but they all differ slightly in the precise way that $\hat{P}(\boldsymbol{z}|\boldsymbol{y})$ is defined and then computed, and in the way that the propensity functions are then modified based on this. We now give a more concrete analysis of our own.

A. Operator splitting in the CME

We focus on the so-called 'slow-scale' approximation[13] and place this in a matrix-framework. Similar to the above, in Section III of Ref. 13 Cao *et. al.* define a *partition* of the set of chemical reactions and likewise an induced partition of the species. The fast reactions induce a 'fast partition' of the state space, with two states being in the same subset of this partition if and only if (either) one can be reached from the other via a sequence of fast reactions. Each subset gives rise to a so-called 'virtual fast process', defined in Section IV of Ref. 13, which consists of the subsystem obtained when the slow reactions are turned off. In the context of the CME, (1) has been split into two, the 'fast part' of which is equation (11). The same splitting may be expressed conveniently in matrix notation as:

$$\boldsymbol{A} = \boldsymbol{A}_f + (\boldsymbol{A} - \boldsymbol{A}_f) \equiv \boldsymbol{A}_f + \boldsymbol{A}_s.$$

Here $\boldsymbol{A}_f$ corresponds to the fast reactions associated with (11), and similarly $\boldsymbol{A}_s$ corresponds to the slow reactions. Both $\boldsymbol{A}_f$ and $\boldsymbol{A}_s$ are required to be infinitesimal generators of a Markov process by themselves, a property deliberately preserved in order for them to be amenable to further analysis.[13]

The matrix $\boldsymbol{A}_f$ is block diagonal, with blocks corresponding to subsets of the fast partition. Thus each block governs a virtual fast process. Similarly, the slow reactions induce a (different) partition under which $\boldsymbol{A}_s$ is also block diagonal. Corresponding to each block of $\boldsymbol{A}_f$ is a stationary solution and later, we will see how these can be used for $\hat{P}$. Thus, in general, $\boldsymbol{A}_f$ has multiple zero eigenvalues, corresponding to distinct eigenvectors, and similar remarks apply to $\boldsymbol{A}_s$.

We would like to exploit the splitting to take advantage of the time-scale separation in the chemical system. In the special case that the split operators were to commute exactly, we would have $e^{\boldsymbol{A}} = e^{h\boldsymbol{A}_s}e^{h\boldsymbol{A}_f}$ and the system would really be reducible to two separate subsystems that could be treated independently. This suggests we develop an approximation by supposing that the operators *almost* commute. We examine this prospect via the Baker-Campbell-Hausdorff formula.

B. Applying the Baker-Campbell-Hausdorff formula to the CME

The Baker-Campbell-Hausdorff (BCH) formula states:

$$\begin{aligned} e^{\boldsymbol{A}} &= e^{\boldsymbol{A}_s+\boldsymbol{A}_f} \\ &= e^{\boldsymbol{A}_s}e^{\boldsymbol{A}_f} + \frac{1}{2}[\boldsymbol{A}_s,\boldsymbol{A}_f] + \frac{1}{12}\big[\boldsymbol{A}_s,[\boldsymbol{A}_s,\boldsymbol{A}_f]\big] + \dots \end{aligned}$$

The BCH formula allows us to establish a connection to the slow-scale approximation.[13] Consider taking a small time step h so that:

$$\begin{aligned} \boldsymbol{p}(t+h) &= e^{h\boldsymbol{A}}\boldsymbol{p}(t) \\ &= e^{h\boldsymbol{A}_s}e^{h\boldsymbol{A}_f}\boldsymbol{p}(t) + \frac{h}{2}[\boldsymbol{A}_s,\boldsymbol{A}_f]\boldsymbol{p}(t) \\ &+ \frac{h}{12}\big[\boldsymbol{A}_s,[\boldsymbol{A}_s,\boldsymbol{A}_f]\big]\boldsymbol{p}(t) + \dots \\ &\approx e^{h\boldsymbol{A}_s}e^{h\boldsymbol{A}_f}\boldsymbol{p}(t) \\ &\approx e^{h\boldsymbol{A}_s}\boldsymbol{A}_{f_\infty}\boldsymbol{p}(t)\ . \end{aligned} \tag{13}$$

The top line is the distribution that must be approximated and then sampled from in order to take a small time step h in a simulation algorithm for the CME, such as the slow-scale SSA.[13] The slow-scale approximation to this distribution is in two stages, represented by the second last and third last lines above. First, the time step h is required to be sufficiently small for the contribution of higher order terms, involving commutators, to be negligible. Second, the time step h is required to be sufficiently large that the fast reactions almost reach equilibrium, so that $e^{h\boldsymbol{A}_f}$ is approximated by its stationary solution (in fact, as already observed there are multiple stationary solutions). Thus, in the second last line, we introduce $\boldsymbol{A}_{f_\infty} \equiv \lim_{t\to\infty} e^{t\boldsymbol{A}_f}$. We have just recovered the two constraints on the size of the time step h made by the slow-scale SSA:[13] h must be be small enough that only a single slow-scale reaction occurs over the interval but large enough that the fast reactions essentially reach equilibrium. The time-scale separation exhibited by models appropriate for the QSSA provides for the existence of such a h.

Consider taking N steps with (13):

$$\begin{aligned} \boldsymbol{p}(Nh) &= (e^{h\boldsymbol{A}_s}\boldsymbol{A}_{f_\infty})\dots(e^{h\boldsymbol{A}_s}\boldsymbol{A}_{f_\infty})\boldsymbol{p}(0) \\ &= (e^{h\boldsymbol{A}_s}\boldsymbol{A}_{f_\infty})\times \\ &\quad (\boldsymbol{A}_{f_\infty}e^{h\boldsymbol{A}_s}\boldsymbol{A}_{f_\infty})\dots(\boldsymbol{A}_{f_\infty}e^{h\boldsymbol{A}_s}\boldsymbol{A}_{f_\infty})\boldsymbol{p}(0)\ . \end{aligned}$$

There are N factors in brackets in each equality. The second equality follows because $\boldsymbol{A}_{f_\infty}$ is a projection matrix so $\boldsymbol{A}_{f_\infty}^2 = \boldsymbol{A}_{f_\infty}$. The first $N-1$ steps use the new update rule

$$\boldsymbol{p}(t+h) = (\boldsymbol{A}_{f_\infty}e^{h\boldsymbol{A}_s}\boldsymbol{A}_{f_\infty})\boldsymbol{p}(t)\ , \tag{14}$$

while only the last step uses (13). This motivates us to change the update rule from (13) to (14). Indeed with this change our approximation becomes a better analogue of the slow-scale approximation because the action of $\boldsymbol{A}_{f_\infty}$ is the analogue of (14) in step 3 of the slow-scale algorithm.[13] As an aside we remark that an alternative way to derive (14) is to apply a similar QSSA-based argument to the Strang splitting.[34] We now introduce the approximation $\boldsymbol{A}_{f_\infty}e^{h\boldsymbol{A}_s}\boldsymbol{A}_{f_\infty} \approx e^{h\hat{\boldsymbol{A}}_{\boldsymbol{s}}}$ into (14) giving our penultimate approximation to (2):

$$e^{t\hat{\boldsymbol{A}}_{\boldsymbol{s}}}\boldsymbol{p}(0)\ . \tag{15}$$

The most obvious choice for $\hat{\boldsymbol{A}}_{\boldsymbol{s}}$ is $\boldsymbol{A}_{f_\infty}\boldsymbol{A}_s\boldsymbol{A}_{f_\infty}$. This is the Krylov approximation, where the projection of the exponential is approximated by the exponential of the projection.[35] However the purpose of this last step is to recover the Markov property, so that the approximation is amenable to further analysis and computation. Since the previous choice may not be Markov we make the ansatz that $\hat{\boldsymbol{A}}_{\boldsymbol{s}} \equiv \boldsymbol{A}_{f_\infty}\boldsymbol{A}_s\boldsymbol{A}_{f_\infty} + \alpha(\boldsymbol{A}_{f_\infty} - I)$, which is Markov for suitably large α.[36] We will see in the next section that with this ansatz we already have enough information to form a very useful numerical approximation and so a more precise specification of $\hat{\boldsymbol{A}}_{\boldsymbol{s}}$ is not required. In summary, we have approximated one Markov process, governed by the CME represented by $\boldsymbol{A}$, with another Markov process, governed by the CME represented by $\hat{\boldsymbol{A}}_s$.

The idea behind the last step is that knowledge of the values of the slow species of the current state is sufficient for the approximate model, since we always approximate the fast species by an equilibrium distribution, conditional on the slow species, anyway. Thus we think of combining states that have the same values of the slow species into one big state (*aggregation*). We now make the connection to aggregation of Markov chains.[37]

C. The QSSA as a form of aggregation

Briefly, we introduce the aggregation and disaggregation operators, $\boldsymbol{E}$ and $\boldsymbol{F}$.[36] Given the state space, of size n_A, and some partition of this into n_B subsets, we define $\boldsymbol{E} \in \mathbb{R}^{n_B\times n_A}$ such that $\boldsymbol{E}_{i,j} = 1$, if state j is in subset i and $\boldsymbol{E}_{i,j} = 0$ otherwise. We are then free to choose any $\boldsymbol{F} \in \mathbb{R}^{n_A\times n_B}$ with nonnegative entries, unit column sum, and such that $\boldsymbol{F}_{i,j}^T \neq 0$ if and only if $\boldsymbol{E}_{i,j} \neq 0$. Usually we think of $n_B \ll n_A$. The pair of operators always have the properties (in both the discrete-time and continuous-time setting) that $\boldsymbol{EF} = \boldsymbol{I}$, $\boldsymbol{FE}$ is a projection matrix, and $\boldsymbol{EAF}$ also represents a Markov process whenever $\boldsymbol{A}$ does. The technique of aggregation was introduced so that the former could be used as an approximation to the latter, with the dual computational advantages of smaller dimension (a matrix of dimension n_B as opposed to n_A), while still preserving the Markov property.

We choose $\boldsymbol{E}$ to combine states according to the partition of the state space into virtual fast processes and we choose $\boldsymbol{F}$ so that its columns record the equilibrium solutions of these fast processes. With this choice

$$\boldsymbol{A}_{f_\infty} = \boldsymbol{FE}\ .$$

(In general we are free to try many choices for $\boldsymbol{E}$ – a linear combination reflecting a conservation law, for example – and for $\boldsymbol{F}$. In the special case of the QSSA though, once we settle on $\boldsymbol{E}$, then $\boldsymbol{F}$ is always defined by the above constraint.) Thus,

$$\begin{aligned}\boldsymbol{E}\hat{\boldsymbol{A}}_s &= \boldsymbol{E}\Big(\boldsymbol{A}_{f_\infty}\boldsymbol{A}_s\boldsymbol{A}_{f_\infty} + \alpha(\boldsymbol{A}_{f,\infty} - I)\Big) \\ &= (\boldsymbol{EF})(\boldsymbol{EA}_s\boldsymbol{F})\boldsymbol{E} + 0 \\ &= \boldsymbol{BE}\,,\end{aligned}$$

where we have introduced $\boldsymbol{B} \equiv \boldsymbol{EA}_s\boldsymbol{F}$. (In fact $\boldsymbol{EA}_f = 0$ so $\boldsymbol{B} = \boldsymbol{EAF}$, which is the conventional approximation used when the technique of aggregation is applied.) Equivalently,

$$\boldsymbol{E}e^{t\hat{\boldsymbol{A}}_s} = e^{t\boldsymbol{B}}\boldsymbol{E}\,. \tag{16}$$

Importantly, this explicitly gives a more efficient way to compute (15). We recover disaggregated solutions as $\boldsymbol{F}e^{t\boldsymbol{B}}\boldsymbol{Ep}(0)$, which is our final approximation to (2). This is mathematically equivalent to approximating (15) by $\boldsymbol{A}_{f_\infty}e^{t\hat{\boldsymbol{A}}_s}\boldsymbol{p}(0)$ but computationally preferable. Thus we have achieved our goal of placing the application of the QSSA to the CME, along with the slow-scale approximation,[13] within the framework of aggregation.

In Refs. 15,16 the number of reactions is used as the measure of computational complexity, but we point out that from the perspective of the CME the dimension of the problem is another relevant measure. Often the number of species, N, is regarded as the dimension of the problem and this is effectively reduced after the application of the QSSA. Importantly the preceding analysis shows that this results in a reduction in the dimension of the matrix: $\boldsymbol{B}$ is typically much smaller than $\boldsymbol{A}$.

On the first page of Ref. 14, Goutsias summarizes the two approximations that Rao and Arkin make in developing their algorithm, which are very similar to those underlying all of the algorithms just discussed[13–16] and the validity and interpretation of these is commented on in footnote 8 of Ref. 13, in which it is emphasized that the components of a Markov process are, in general, not Markov. We point out here that these assumptions are closely related to the technique of aggregation and the case where they hold exactly is known as 'ordinary lumpability' of Markov chains,[38,39] a special case that can arise when reducing the size of the Markov model. This amounts to (16) holding, with $\hat{\boldsymbol{A}}_s$ replaced by an appropriate $\boldsymbol{A}$, representing the larger Markov process in question, and considered under an appropriate aggregation scheme, $\boldsymbol{E}$. (In general, given $\boldsymbol{A}$ and $\boldsymbol{E}$, there need not exist such a $\boldsymbol{B}$.)

D. Computation of quasi-stationary distributions

We now outline three strategies for obtaining the stationary solutions of the fast operator, which we need to define the columns of $\boldsymbol{F}$.

The first approach is described in Appendix A of Ref. 13, where a recursive formulation for the stationary solution may be derived by making the ansatz that the model satisfies the special criterion of *detailed balance*.[3] We identify a formula in this way for the Michaelis-Menten model. One problem with this approach is that a naïve use may lead to underflow, for some parameter ranges. Beginning the computation nearby the peak of the solution has been suggested in order to cope with this.[13]

While being able to find a formula is useful from the point of view of further analysis, a computational strategy could be automated and could cope with larger models for which simple formulas can not be obtained. We suggest automatic identification of the blocks, $\boldsymbol{A}_{\boldsymbol{f_i}}$, of the fast operator, via the reachability structure of the model, and then to use off-the-shelf methods for solving the matrix equation $\boldsymbol{A}_{f_i}x_i = \boldsymbol{0}$. Two good choices would be the LU factorization (for relatively small, dense blocks) or the power method (for larger, sparse blocks). Each block $\boldsymbol{A}_{\boldsymbol{f_i}}$ is guaranteed to have a non-trivial solution corresponding to the stationary solution because it is an infinitesimal generator of a Markov process. We set $\hat{P}(\boldsymbol{z}|\boldsymbol{y}_i) \equiv x_i(\boldsymbol{z})$. Additional considerations apply when $\boldsymbol{A}_{f_i}$ is the result of truncation, such as in the FSP, and so is not quite a true generator. In such cases it may not have an invariant solution at all.

For the LU factorization, we have chosen to use the DGESV subroutine freely available as part of LAPACK.[40] Note that a naïve use of such a solver will not return the desired result– it may simply return the trivial solution, or not return at all due to the singular nature of the blocks. In order to overcome this one can use the trick suggested by Chan[41] of adding a rank one matrix. i.e. solve $(\boldsymbol{A}_{\boldsymbol{f_i}} + \alpha e_j e_j^T)\tilde{x}_i = e_j$ and set $x_i = \frac{\tilde{x}_i}{\alpha\tilde{x}_i}$. Although mathematically the choice of the index j is arbitrary, numerically it helps to set j to correspond to the column of $\boldsymbol{A}_{\boldsymbol{f_i}}$ with diagonal entry of minimal magnitude. Note that this trick involves changing just a single entry in the matrix and so is extremely computationally cheap.

In order to be sure of converging to the desired result there are some well-known restrictions on the power method. By consideration of the Gerschgorin disks associated with the operator, the choice of $\alpha \equiv 2 * \max|a_{jj}|$ ensures that the use of the power method with the shifted operator $(\boldsymbol{A}_{\boldsymbol{f_i}} + \alpha I)$ converges to the correct eigenvalue. A good choice for the initial vector would be $[1, \ldots, 1]^T$ because it is guaranteed to have a nonzero projection onto the stationary solution, and also has the desirable property of being orthogonal to all other eigenvectors.

E. The QSSA-based CME-solver

ALGORITHM 3: QSSA CME-solver($\boldsymbol{A}, t_f, \boldsymbol{p}(0), [R_f, R_s]$)
$[\boldsymbol{B}, \boldsymbol{Ep}(0), \boldsymbol{F}]$=Preprocess($\boldsymbol{A}$, $\boldsymbol{p}(0)$, $[R_f, R_s]$) ;
$[\boldsymbol{q}(t_f)]$=KrylovFSP ($\boldsymbol{B}$, $\boldsymbol{Ep}(0)$, t_f, ϵ, $\boldsymbol{X}_0$, m, tol) ;
return $\boldsymbol{Fq}(t_f)$;

The QSSA-based CME-solver is outlined in Algorithm 3, which evaluates (13) via the technique in (16). We emphasize that, after application of the QSSA, the reduced model can still be implemented computationally as a Krylov FSP, although now with modified propensity functions. The preprocessing stage computes the aggregated version of the initial state, and then computes the way in which the propensity functions of the remaining slow reactions should be modified. This latter step is described in the previous section. Note that neither $\boldsymbol{E}$ nor $\boldsymbol{F}$ need ever be formed explicitly and it is only their action that is computed. The QSSA-based CME-solver computes $e^{t_f \boldsymbol{B}}\boldsymbol{Ep}(0)$ and then, in the last line of Algorithm 3, there is a post-processing step, which is mathematically equivalent to multiplication of the resulting distribution by $\boldsymbol{F}$. If the approximation were exact we would have that $\boldsymbol{Fq}(t_f) = \boldsymbol{p}(t_f)$.

We point out that the remarks made in Refs. 13,16 about the SSA carry over to the QSSA-based CME-solver in a way that is completely analogous. First, in many cases it is only the aggregated distributions that are of interest and so some modest computational savings may be made by skipping the post processing step. Secondly, in line with previous authors, we compare the distributions for a particular species as a measure of the accuracy of the approximation. For example, in the case of the Michaelis-Menten enzyme kinetics, the accuracy is assessed in terms of how well the (marginal) distribution for the number of products is approximated. Third, often experimental data is so difficult to obtain that there is really only enough information to parameterize an aggregated model, such as the QSSA anyway.

F. The tQSSA-based CME-solver

Having discussed the interpretation of the QSSA in the stochastic setting, and in particular its application to the CME, we now distinguish between the total (tQSSA) and standard (sQSSA) versions, in analogy with the deterministic setting. The original papers detailing the tQSSA in the ODE setting[42–46] introduce it via the example of Michaelis-Menten kinetics, the same example that Rao and Arkin use to introduce their QSSA in the stochastic setting,[16] so we follow this lead. From the perspective of aggregation the total and standard versions may be regarded as corresponding to special choices of $\boldsymbol{E}$ in Algorithm 3.

The tQSSA for Michaelis-Menten enzyme kinetics

This model was used to demonstrate application of the QSSA to the SSA[14,16] so it is a good test problem for the tQSSA CME-solver. We use the same initial state and rate constants here. It involves an enzyme, E, that gradually catalyzes the conversion of all available substrate, S, into a product, P, via an intermediate complex, C. There are four chemical species, $[S,\ E,\ C,\ P]^T$, and three chemical reactions, which are described in Table I. The first two reactions are deemed to be 'fast' and the last to be 'slow'. The state space of the Michaelis-

	Reaction	Propensity
1	S + E $\longrightarrow$ C	$c_1 \times S \times E$
2	S + E $\longleftarrow$ C	$c_2 \times C$
3	C $\longrightarrow$ P + E	$c_3 \times C$

TABLE I: Description of the stochastic Michaelis-Menten enzyme kinetics scheme as in Ref. 16. Rate constants: $c = [1.0,\ 1.0,\ 0.1]^T$. Initial state: $[100,\ E_I,\ 0,\ 0]^T$. Example (i): $E_I = 10$, $t_f = 30$. Example (ii): $E_I = 1000$, $t_f = 20$.

Menten scheme is depicted in Figure 1. The initial state is $[S_I, E_I, 0, 0]$, and is represented by the state in the bottom left corner. The system is subject to the following pair of conservation laws: $E_I = E+C$ and $S_I = S+C+P$. Within the same row, reactions represent the reversible formation and dissociation of the complex, while transitions from one row to the next row higher up represent the irreversible formation of another product. The support of the distribution gradually moves up the rows, becoming more concentrated, which can be exploited for numerical advantage.[17–19] We distinguish two cases for the initial state, depending on whether the enzyme is in excess or not: (i) $S_I > E_I$ and (ii) $S_I \leq E_I$. The state space for (i) includes all states in Figure 1, and the size is $\frac{(E_I+1)(E_I+2)}{2} + (E_I+1)(S_I - E_I)$. In case (ii) the state space consists of only the 'triangle' formed by retaining the states in the row marked by an asterisk (*) and those above it. The total number of states is $\frac{(S_I+1)(S_I+2)}{2}$.

The tQSSA aggregates states with the same value of the 'total substrate' variable, $S_T \equiv S + C$. These aggregated states are depicted as 'blocks' in Figure 1. We think of the aggregation operator for the tQSSA as having the action $\boldsymbol{E}_t[S, E, C, P] \equiv [S_T]$, although this is a slight abuse of notation since $\boldsymbol{E}_t$ acts on the set of all states, not just one at a time. The initial state is replaced by the aggregated initial state $S_{T_I} = S_I + C_I$ and thus the size of the state space of the aggregated model is only $S_I + 1$. In order to recover the distribution for the rest of the species we use the conditional distribution $P(S, C, E, P \,|\, S_T)$. Observing the conservation laws, it is enough to know just $P(C \,|\, S_T)$. In general, obtaining exact knowledge of $P(C \,|\, S_T)$ would be as difficult a numerical problem as solving the full CME, however in analogy with the tQSSA in the deterministic setting we may approximate it by assuming that the formation and dissoci-

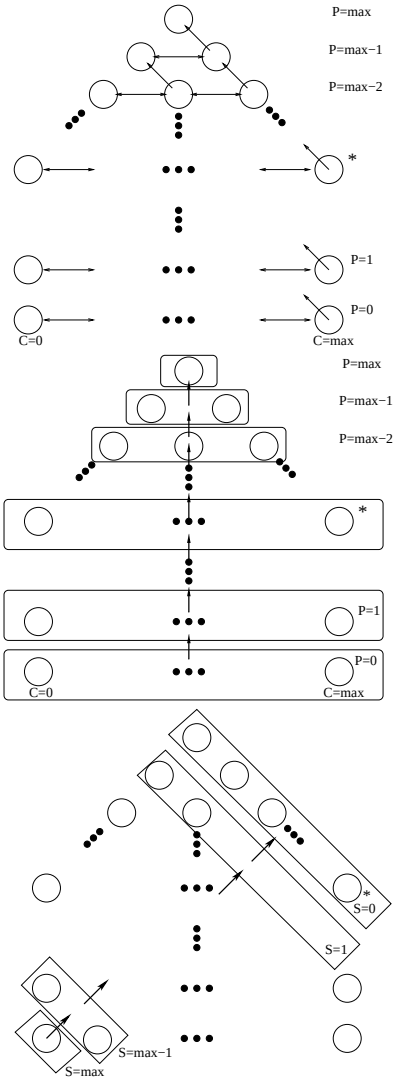

FIG. 1: *Top:* Schematic representation of the state space of the Michaelis-Menten model in the stochastic regime. The state at the top is an absorbing state corresponding to the situation where all substrates have been converted to products so the number of products is at a maximum ($P = \max = S_I$). *Middle:* Interpretation of the tQSSA, which aggregates states with the same value of the total substrate, $S_T \equiv S + C$. *Bottom:* Interpretation of the sQSSA, which aggregates states with the same number of free substrates.

ation of the complex species is at quasi-equilibrium, given the value of the total substrate. That is, we ignore the slow reactions that give rise to transitions outside of the block and just concentrate on the fast reactions within the block, and use the equilibrium distribution of this virtual fast process[13] as our approximation. That is we make the approximation $P(C \,|\, S_T) \approx \hat{P}(C \,|\, S_T)$, where $\hat{P}(C \,|\, S_T) \equiv x_{S_T}(C)$ is obtained via the non-trivial solution to $\boldsymbol{A}_{f,S_T} x_{S_T} = 0$ and where $\boldsymbol{A}_{f,S_T}$ represents the virtual fast process for the particular block in question. Details are given in Appendix A. We can now compute the modified propensities that complete the specification of the aggregated model: the propensity for the reaction $S_T \to S_T - 1$ is $c_3 \mathbb{E}[C|S_T]$.

The tQSSA is the most natural aggregation to consider in the stochastic regime. In the deterministic setting of ODEs however, the tQSSA was not the first form of the QSSA to be considered, with the change of variables introduced only later in order to greatly extend the parameter regime over which the QSSA was valid.[46] Instead the standard QSSA (sQSSA) was considered first so we consider a stochastic analogue of this too.

The sQSSA makes the same approximation without introducing the change of variables, which in the stochastic setting corresponds to aggregating states with the same number of free substrates. The bottom of Figure 1 depicts this interpretation of the sQSSA. One problem is that the states within a block are connected by the relatively slow reactions, so this approximation corresponds to the counter-intuitive assumption that the slow reactions almost reach equilibrium before the fast reactions take effect. Furthermore, the equilibrium solution to the virtual process that each block gives rise to is simply a delta distribution on the state at the top of the block. The modified propensity functions would be based solely on this state but since it has no complex species, it does not transition to any other block. Thus, by itself, this stochastic version of the sQSSA is not a sensible approximation. This observation provides another way of understanding why the tQSSA can be more successful than the sQSSA in the deterministic setting, when the ODE model of chemical reactions is viewed as a limiting case of the Markov model.[47]

There are also other approximations such as the 'reverse' QSSA[42,46] that could be effective. The rQSSA is extensively studied in Refs. 48,49 where it is shown that the Michaelis-Menten approximation is nothing more than the zero-order approximation of the system, when we make an asymptotic expansion of the solutions S and C in terms of the parameter $\epsilon \equiv (E_I/(K_M + S_I))$ and that, consequently, the Michaelis-Menten approximation is valid not only when $E_I/S_I \ll 1$, but also when $(E_I/(K_M + S_I)) \ll 1$. Here $K_M = \frac{c_2+c_3}{c_1}$. This further motivates the introduction of the tQSSA to the CME as it too may be valid over a wider parameter range.

We now consider Rao and Arkin's interpretation of the QSSA applied to the CME associated with Michaelis-Menten enzyme kinetics. Although the total substrate variable $S_T = S + C$ is explicitly introduced on page 5002 in Ref. 16, there is no connection to the tQSSA. In fact, eq. (17) in Ref. 16 is obtained by performing an asymptotic expansion of the probability P in terms of the perturbing term $\epsilon := \frac{e_0}{s_0}$. This mechanism, with the same parameter, was suggested by Heineken *et al.*,[50] in a deterministic framework, in order to show that the sQSSA can be considered as the zero-order approximation of the system. The asymptotic expansion proposed by Rao and Arkin is valid only if $\frac{e_0}{s_0} \ll 1$. Thus their approximation cannot be classified as an example of the tQSSA. As a confirmation of this fact, in Figure 1 of Ref. 16, it is re-

ported that the approximation performs well in the case where substrates are in excess of enzymes but performs poorly when the situation is reversed. This is in contrast to the deterministic setting where the tQSSA was introduced precisely because it performed better for the case where enzymes were in excess. As a partial justification of the poor performance of their approximation in the stochastic setting, when enzymes are in excess, Figure 3 of Ref. 16 shows that, in the deterministic setting, the QSSA also performs poorly. This highlights the need to distinguish between the various forms of the QSSA and make connections to the deterministic literature because, for example, the tQSSA would perform well in both regimes.

Lastly, we point out that if the closely related slow-scale approximation[13] were applied to the Michaelis-Menten scheme, it would be classified as an example of the tQSSA, because the partition into virtual fast processes is the same as the partition induced by the introduction of the total substrate, S_T. In particular each virtual fast process corresponds to a block in Figure 1.

VII. NUMERICAL RESULTS

In this section we compare the accuracy and efficiency of two numerical methods for the solution of the CME: (A) the Krylov FSP for the full CME and (B) the tQSSA-based CME-solver. Unless otherwise stated, all numerical experiments used FORTRAN with the Intel 'ifort' compiler, and were conducted on an SGI Altix with 64 Itanium 2 CPUs and 120 GB of memory running the Linux operating system. However only a single processor was used. Since the true solution is not available we assess the accuracy of the tQSSA by comparison with the Krylov FSP, with strict tolerances. By default the Krylov FSP is called with (Expokit, FSP) tolerances of $(10^{-8}, 10^{-5})$ but bear in mind that the method is guaranteed to be at least this accurate – the actual results may be better. Note that the post-processing step is not included in the runtimes reported here and that the accuracy of the tQSSA-CME-solver is assessed according to the conditional distributions, as described in Section VI E.

As test problems we use the models that Rao and Arkin[16] considered as well as additional models of enzymatic networks that form important building blocks of the mitogen-activated-protein kinase cascade:[44] double phosphorylation,[43] and the Goldbeter-Koshland switch.[45]

A. Michaelis-Menten enzyme kinetics

First we apply the Krylov FSP to the full CME for Michaelis-Menten enzyme kinetics and compare this with Figure 1 of Ref. 16. This provides a comparison of the SSA with the Krylov CME-solver for the purpose of estimating moments of the CME. Rao and Arkin report using 50,000 SSA simulations to generate this Figure, from which they estimate the mean and variance. There are two examples in the Figure with initial states matching examples (i) and (ii) of Table I but different time parameters of $t_f = 160$ and $t_f = 70$, respectively. The first example has a state space size of 1056 and the Krylov FSP will compute the CME solution in under 3 seconds, whereas 50,000 simulations with the SSA takes more than 50 seconds. Thus the Krylov FSP will estimate the CME solution and its moments more accurately and more efficiently. The second example is about five times larger with 5,151 states in all, and a norm that is about two orders of magnitude larger than example (i). While the CME-solver takes longer than 50,000 simulations, it is more accurate, and for the same accuracy it is more efficient, similar to the results of previous studies.[18] The tQSSA-based CME-solver is quite accurate for this second example, being of the order of 10^{-4} in the 1-norm. This is consistent with intuition derived from the deterministic tQSSA. In particular we do not encounter the trouble that Rao and Arkin report for this example. In fact, as remarked above, although Rao and Arkin explicitly introduce the total substrate $S_T \equiv S + C$ in the derivation of their approximation, the final, numerical implementation, can not be interpreted as a tQSSA in the context of the CME.

Example		runtime(s)	$\|\cdot\|_1$	$\|\cdot\|_2$	$\|\cdot\|_\infty$
(i)	$\boldsymbol{B}$	0.6 ($\boldsymbol{A}$: 0.5)	7E-3	2E-3	4E-4
(ii)	$\boldsymbol{B}$	0.6 ($\boldsymbol{A}$: 130)	3E-4	9E-5	3E-5

TABLE II: Comparison of Krylov FSP ($\boldsymbol{A}$) and tQSSA ($\boldsymbol{B}$) for the Michaelis-Menten model with parameters given in Table I.

The tQSSA reduces the model to a one-dimensional, pure death process. Thus the resulting matrix $\boldsymbol{B}$ is bidiagonal. Further details are given in Appendix A.

Ex	$\|\boldsymbol{A}\|_2$	$\|\boldsymbol{A}_f\|_2$	$\|\boldsymbol{A}_s\|_2$	$\|\boldsymbol{C}\|_2$	$\|\boldsymbol{A}\boldsymbol{p}(0)\|_2$	$\|\boldsymbol{C}\boldsymbol{p}(0)\|_2$	$\|\boldsymbol{B}\|_2$
(i)	1.7E3	1.7E3	1.7	1.7E2	1.4E2	1.4E2	1.97
(ii)	1.9E5	1.9E5	19.1	1.8E4	1.4E5	1.4E4	19.1

TABLE III: Norms of operators in Michaelis-Menten model. $\boldsymbol{C} \equiv [\boldsymbol{A}_s, \boldsymbol{A}_f]$.

The results of applying the tQSSA-based CME-solver with this $\boldsymbol{B}$, to the two examples in Table I, are recorded in Table II, which shows that it is an extremely good approximation. The accuracy is measured by comparing the conditional distributions for the products. Example (ii) shows considerable savings in runtime, while example (i) is really too small to see this. The tQSSA is more accurate for example (ii) where the enzymes are in excess, which is to be expected since the increased population of enzymes increases the propensity of the fast reactions making the assumptions underlying the tQSSA even more appropriate.

The norms of the operators involved in the examples in Table I are given in Table III. The norm of the reduced operator $\boldsymbol{B}$ is at least two orders of magnitude less than that of the full model $\boldsymbol{A}$ and this is where some of the computational savings are being made. Also, $\frac{\|[\boldsymbol{A}_s,\boldsymbol{A}_f]\|_2}{\|\boldsymbol{A}\|_2}$ is small, which is consistent with analysis via the BCH formula. Further experiments show that $\|[\boldsymbol{A}_s,\boldsymbol{A}_f]\boldsymbol{p}(t)\|_2$ becomes much smaller for larger t. This suggests using the full Krylov FSP for a very brief initial transient, and then switching to the tQSSA for the rest of the computation, would increase the accuracy of the tQSSA without much extra cost. Numerical experiments combining the algorithms confirm this for example (ii): using the full Krylov FSP for an initial transient of $t = 1.0$ and then switching to the tQSSA for the rest of the integration only increases the runtime to about 10 seconds but gives significantly greater accuracy of 10^{-6} in the 1-norm.

B. Double phosphorylation

This is an example of a fully competitive reaction scheme, with substrates competing for a common enzyme, and arises in the double phosphorylation of MAPK by MAPKK.[43,51] It can be thought of as one Michaelis-Menten scheme feeding into another and so there is a natural choice for the tQSSA in which two new 'total substrate' variables are introduced: $S_{T_i} \equiv S_i + C_i$ for $i = 1, 2$. It may also be thought of as a special case of the model for competitive inhibition considered later,[43] in which $P_1 \equiv S_2$. There are six chemical species, $[S_1,\ E,\ C_1,\ S_2,\ C_2,\ P]^T$, and six chemical reactions, described in Table IV, which gives parameters for six examples considered here. It is subject to two conservation laws: $S_{1_I} = S_1 + C_1 + S_2 + C_2 + P$ and $E_I = E + C_1 + C_2$ and has an absorbing state that is always eventually reached. The pair of reversible reactions 1 and 2, and the pair 4 and 5, are deemed to be fast and the remainder slow. We consider the model from first the deterministic

	Reaction	Propensity
1	$S_1 + E \longrightarrow C_1$	$c_1 \times S_1 \times E$
2	$S_1 + E \longleftarrow C_1$	$c_2 \times C_1$
3	$C_1 \longrightarrow S_2 + E$	$c_3 \times C_1$
4	$S_2 + E \longrightarrow C_2$	$c_4 \times S_2 \times E$
5	$S_2 + E \longleftarrow C_2$	$c_5 \times C_2$
6	$C_2 \longrightarrow P + E$	$c_6 \times C_2$

TABLE IV: Description of the dual phosphorylation enzyme kinetics scheme. Examples (a), (b) and (c) have $c = [0.2,\ 1.0,\ 0.6,\ 0.2,\ 1.0,\ 0.5]^T$[45] with initial states $[100,\ E_I,\ 0,\ 0,\ 0,\ 0]^T$, where $E_I = 1000, 100, 10$ with a corresponding choice of $t_f = 2, 2.5, 20$. Examples (d), (e) and (f) match (a), (b) and (c), respectively, except they have different rate constants: $c = [1.0,\ 1.0,\ 0.1,\ 1.0,\ 1.0,\ 0.1]^T$.

and then the stochastic perspective.

We begin with a brief review of the ODE version of the model, following Pedersen *et. al.*[43] The full model is described by the following six DEs:

$$\begin{aligned}
\frac{dS_1}{dt} &= c_2C_1 - c_1S_1E \\
\frac{dE}{dt} &= (c_2 + c_3)C_1 + (c_5 + c_6)C_2 - c_1S_1E - c_4S_2E \\
\frac{dC_1}{dt} &= c_1S_1E - (c_2 + c_3)C_1 \\
\frac{dS_2}{dt} &= c_3C_1 + c_5C_2 - c_4S_2E \\
\frac{dC_2}{dt} &= c_4S_2E - (c_5 + c_6)C_2 \\
\frac{dP}{dt} &= c_6C_2\ . \qquad (17)
\end{aligned}$$

Using the two conservation laws, (17) can be reduced to just four DEs. The sQSSA then makes the approximation that $\frac{dC_i}{dt} \approx 0$, for $i = 1, 2$, while the tQSSA makes the same approximation after first introducing the change of variables mentioned above. Either approach reduces the system to just two DEs, although they are parameterized by more complicated propensity functions. In particular the tQSSA gives rise to a cubic polynomial that must be solved at each step of a numerical integration scheme. Figure 2 presents solutions to the model and results of the tQSSA for example (c). The approximation is extremely accurate and further experiments for examples (a) and (b) show similar results. Numerical experiments using *MATLAB*'s built-in ODE solvers for example (b) gave the following results. Using the stiff solver *ode15s* was slightly faster than using *ode45* and using Cardan's formula, for the solution of a cubic, was more than twice as fast as *fzero*. Surprisingly it was found that there was not much difference between the runtimes taken to solve the full and reduced (tQSSA) systems. Indeed solving the full system was slightly faster with solutions being obtained in a matter of seconds. This raises questions about the worth of the tQSSA approximation for this model from a numerical point of view in the deterministic ODE setting.

Figure 3 shows a simulation of a stochastic trajectory. It appears that the stochastic and deterministic dynamics are roughly consistent, with the time taken to reach equilibrium, and the molecular populations, being comparable.

Table V compares the tQSSA with the full Krylov FSP. For each example we choose a value of t_f that occurs at an 'interesting' stage of the dynamics of the process, roughly just before the peak in the population of the second complex, C_2, and still far from equilibrium. The results are significant but it is anticipated that choosing larger values of t_f would favour the tQSSA even more. Examples (a), (b) and (c) show the effect of reducing the number of enzymes. For examples (a) and (b) almost all of the computational time of the tQSSA is spent preprocessing, with less than one second being needed to solve the reduced system. For both examples, the Krylov FSP eventually works on a matrix of size 4,517,885, which is about 98% of the full model (of size 4,598,126).

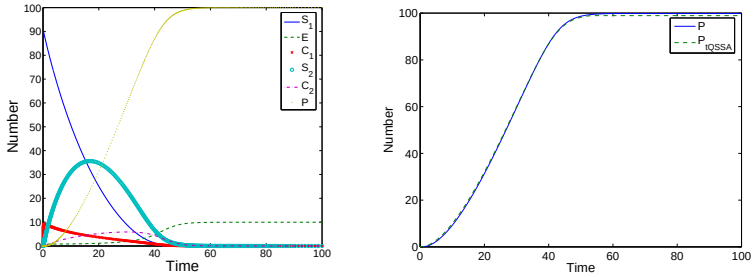

FIG. 2: Solutions to the deterministic double phosphorylation model for example (c) in Table IV. *Left:* Full solution. *Right:* tQSSA solution for the products. The approximation is very good, although it slightly overestimates the population of products for very early times and then slightly underestimates the population at equilibrium.

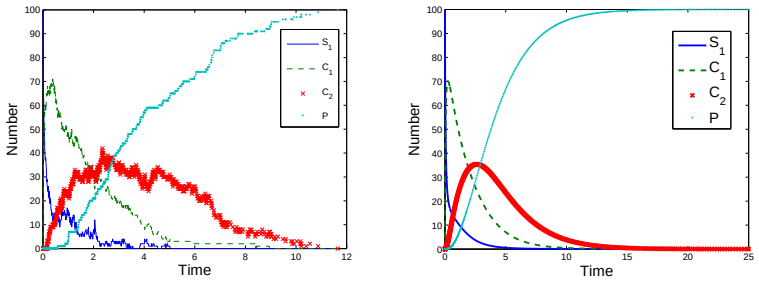

FIG. 3: Dynamics of the stochastic and deterministic models of dual phosphorylation for example (b) in Table IV. *Left:* A stochastic trajectory (via the SSA). The process is absorbed at $t_f \approx 12$. *Right:* Solution to the corresponding deterministic model. The system is just beginning to settle down by $t = 12$.

Example		runtime (s)	$\|\cdot\|_1$	$\|\cdot\|_2$	$\|\cdot\|_\infty$	n	$\mathbb{E}(P)$
(a)	$\boldsymbol{A}$	7,446				4,517,885	29.4
	$\boldsymbol{B}$	356	4E-2	8E-3	3E-3	5,151	29.7
(b)	$\boldsymbol{A}$	1,414				4,517,885	28.2
	$\boldsymbol{B}$	353	0.6	0.1	4E-2	5,151	31.8
(c)	$\boldsymbol{A}$	60				270,272	31.4
	$\boldsymbol{B}$	5	0.3	5E-2	1E-2	5,007	33.3
(d)	$\boldsymbol{A}$	7,567				1,782,721	1.745
	$\boldsymbol{B}$	144	2E-3	1E-3	6E-4	5,151	1.749
(e)	$\boldsymbol{A}$	1,227				1,869,423	2.1
	$\boldsymbol{B}$	151	9E-2	4E-2	2E-2	5,007	2.2
(f)	$\boldsymbol{A}$	5				32,967	1.6
	$\boldsymbol{B}$	1.2	6E-2	3E-2	2E-2	5,007	1.7

TABLE V: Comparison of Krylov FSP ($\boldsymbol{A}$) and tQSSA ($\boldsymbol{B}$) for the double phosphorylation model, with examples as in Table IV. The accuracy of the tQSSA is assessed in terms of the conditional distribution for the products, P, the mean of which is recorded in the last column. For each method, n is the size of the projection used.

The enzymes are well in excess for example (a), which gives the best results for these rate constants. The speed-up is more than an order of magnitude while maintaining reasonable accuracy. Two visualizations of the solution are provided in Figures 4 and 5.

The enzymes and substrates are balanced in example (b), which shows a speed-up of about a factor of four. This is less than the last example, as is the accuracy. This behaviour is to be expected of the tQSSA as we decrease the number of enzymes, based on our experience in the deterministic setting. The runtime for the full CME-solver is reduced for this example. This is also to be expected, since the increase in the parameter t_f is only slight but the reduction in the size of the propensities (due to the decreased enzymes) is significant, thus allowing Expokit to take larger time steps. The runtime of the tQSSA is comparable because a similar amount of preprocessing is required. Despite having a significant error in the 1-norm, the CME solutions in Figure 6 appear very similar.

The enzymes have been reduced so much in example (c) that the substrates are now in excess and the system is truly competitive. Compared to the previous examples, we expect that the much larger value of t_f would be favourable towards the tQSSA, while the reduced number of enzymes would be unfavourable. Overall the accuracy is intermediate between the first and second examples, while the speed up is about an order of magnitude. For this smaller example, most of the computational time required by the tQSSA CME-solver is spent on building the state space, (not on preprocessing), which is in contrast to the previous two examples. The CME solutions obtained from the two methods show a similar comparison as in example (b). However, two visualizations of the full CME solution, provided in Figure 7, appear quite

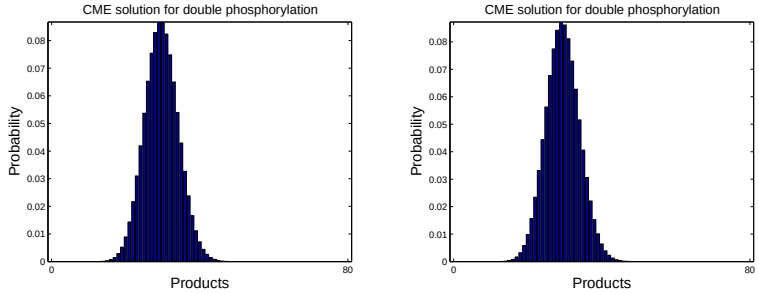

FIG. 4: The CME solution, visualized in terms of the conditional distribution for the products for example (a) in Table IV. *Left:* tQSSA (***B***). *Right:* Krylov FSP (***A***). Visually, the two methods appear almost indistinguishable although according to the 1-norm the approximation is only of the order of 10^{-2}.

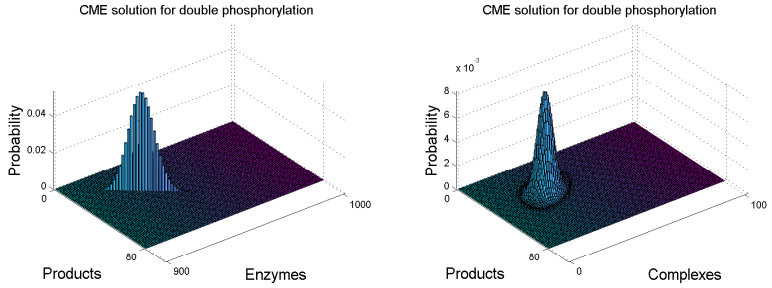

FIG. 5: The (true) CME solution for example (a) in Table IV, computed with the full Krylov FSP. The support is concentrated in a small fraction of the total state space; this is exploited for numerical advantage by the Krylov FSP. *Left:* Products compared to enzymes. *Right:* Products compared to the first complex, C_1.

different to those for the previous examples.

Examples (d), (e) and (f) show the effects of changing the rate constants to be in line with a coupled pair of Michaelis-Menten schemes taken from Ref. 16. This change makes the problem even more suitable to the tQSSA as the difference between the propensities for fast and slow reactions is more pronounced. Thus examples (d), (e) and (f) show better accuracy than examples (a), (b) and (c), respectively and a glance at $\mathbb{E}(P)$ shows the systems have proceeded much more quickly towards equilibrium. Compared to (a) and (b) the projection size is significantly reduced in (d) and (e), respectively, but this is not reflected in the runtimes because the new rate constants force Expokit to use smaller step sizes. This stiffness is overcome through aggregation and the tQSSA for these examples is about twice as fast.

C. Competitive inhibition

This model consists of two Michaelis-Menten enzyme kinetic models, catalyzed by the same enzyme.[16] There are seven chemical species, $[S_1,\ E,\ C_1,\ P_1,\ S_2,\ C_2,\ P_2]^T$, and six chemical reactions, which are described in Table VI. It is subject to three conservation laws: $S_{1_I} = S_1 + C_1 + P_1$, $S_{2_I} = S_2 + C_2 + P_2$ and $E_I = E + C_1 + C_2$. The fast reactions are 1, 2, 4 and 5.

	Reaction	Propensity
1	$S_1 + E \longrightarrow C_1$	$c_1 \times S_1 \times E$
2	$S_1 + E \longleftarrow C_1$	$c_2 \times C_1$
3	$C_1 \longrightarrow P_1 + E$	$c_3 \times C_1$
4	$S_2 + E \longrightarrow C_2$	$c_4 \times S_2 \times E$
5	$S_2 + E \longleftarrow C_2$	$c_5 \times C_2$
6	$C_2 \longrightarrow P_2 + E$	$c_6 \times C_2$

TABLE VI: Description of the competitive inhibition enzyme kinetics scheme.[16] Initial state: $[50,\ 30,\ 0,\ 0,\ 60,\ 0,\ 0]^T$, $t_f = 10$, $c = [1.0,\ 1.0,\ 0.1,\ 1.0,\ 1.0,\ 0.1]^T$.

An analysis, very similar to that of the double phosphorylation model, may be applied, again introducing two total substrate variables, so that the reduced model has the species $[S_{T_1},\ S_{T_2},\ P_1,\ P_2]$. Table VII shows a speed-up of more than an order of magnitude, while maintaining good accuracy. Almost half of the runtime of the tQSSA is consumed by preprocessing. The full model uses a projection of size $\approx 800,000$ but the tQSSA reduces this to ≈ 3000. The means compare favourably:

	$\mathbb{E}(S_{T_1})$	$\mathbb{E}(S_{T_2})$	$\mathbb{E}(P_1)$	$\mathbb{E}(P_1)$
A	35.60	43.92	13.40	16.08
B	36.57	43.88	13.43	16.12

The means of P_1 and P_2 are slightly over-estimated, as expected, so the behaviour of the tQSSA for this example

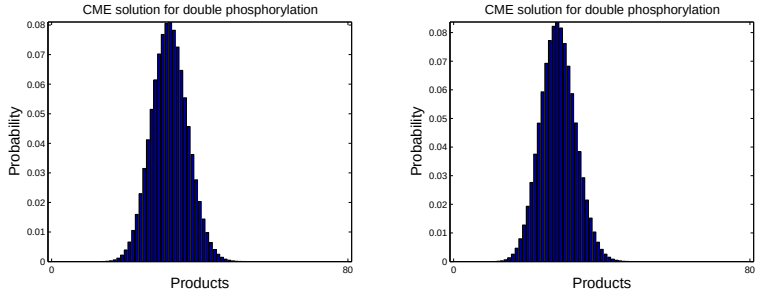

FIG. 6: Comparison of the tQSSA (*left*) with the Krylov FSP (*right*) for example (b) in Table IV. The tQSSA has 'shifted' the distribution to the right, although only slightly, which is consistent with our intuition that the tQSSA over estimates how quickly the reactions progress towards equilibrium.

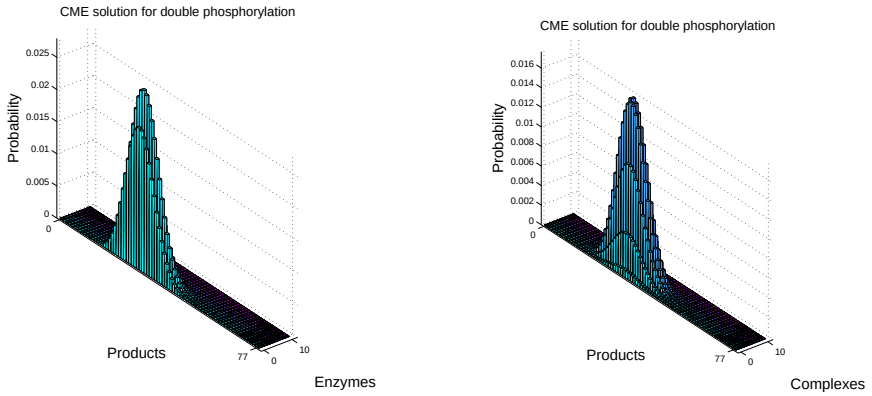

FIG. 7: The (true) CME solution for example (c) in Table IV. *Left:* Products compared to enzymes. *Right:* Products compared to the first complex, C_1.

is similar to example (b) in Figure 6 where the distribution had a similar spread but was 'shifted' to the right.

	runtime(s)	$\|\cdot\|_1$	$\|\cdot\|_2$	$\|\cdot\|_\infty$
$\boldsymbol{B}$	22 ($\boldsymbol{A}$: 540)	9E-3	2E-3	7E-4

TABLE VII: Results for competitive inhibition. Accuracy of the tQSSA assessed in terms of the distribution for P_2.

D. Goldbeter-Koshland switch

The Goldbeter-Koshland switch[52] consists of a pair of Michaelis-Menten enzyme kinetic models, catalyzed by different enzymes, in which the product of the one forms the substrate of the other, and vice-versa. There are six chemical species, $[S,\ E_1,\ C_1,\ P,\ E_2,\ C_2]^T$, and six chemical reactions, which are described in Table VIII. It is subject to three conservation laws: $S_I = S + C_1 + P + C_2$, $E_{1_I} = E_1 + C_1$ and $E_{2_I} = E_2 + C_2$. Reactions 1, 2, 4 and 5 are fast. The full model uses a projection of size $\approx 170,000$ but the tQSSA drastically reduces this to ≈ 100, giving a speed-up of more than an order of magnitude as shown in Table IX. Almost 90% of the runtime of the tQSSA is spent preprocessing. Again, the means compare well: $\mathbb{E}(S_{T_1})$ and $\mathbb{E}(S_{T_2})$ are 51.10 and 48.91, respecitvely, under $\boldsymbol{A}$, while they are 51.09 and 48.91, respectively, under $\boldsymbol{B}$.

	Reaction	Propensity
1	$S + E_1 \longrightarrow C_1$	$c_1 \times S \times E_1$
2	$S + E_1 \longleftarrow C_1$	$c_2 \times C_1$
3	$C_1 \longrightarrow P + E_1$	$c_3 \times C_1$
4	$P + E_2 \longrightarrow C_2$	$c_4 \times P \times E_2$
5	$P + E_2 \longleftarrow C_2$	$c_5 \times C_2$
6	$C_2 \longrightarrow S + E_2$	$c_6 \times C_2$

TABLE VIII: The Goldbeter-Koshland switch.[52] Initial state: $[100,\ 100,\ 0,\ 0,\ 100,\ 0]^T$, $t_f = 20$, $c = [1.0,\ 1.0,\ 0.1,\ 1.0,\ 1.0,\ 0.1]^T$.

	runtime(s)	$\|\cdot\|_1$	$\|\cdot\|_2$	$\|\cdot\|_\infty$
$\boldsymbol{B}$	21 ($\boldsymbol{A}$: 700)	1E-3	2E-4	7E-5

TABLE IX: Comparison of methods for the Goldbeter-Koshland switch. Accuracy of the tQSSA assessed in terms of the distribution for S_{T_1}.

E. Simplified λ-phage switch

This is a simplified model of the P_R promoter[14,16] described in Table X. The extension of the tQSSA is not as straight-forward because unlike the previous examples, this model is not decomposable into Michaelis-Menten building blocks. The last two reactions corresponding to the reversible dimerization are fast so we introduce $S_T \equiv 2D + M$, as the total number of monomers (in dimer or free form), and suggest this as the tQSSA, regarding previous works[14,16] as examples of this. This model is larger than the previous ones and thus more computationally challenging for a CME-solver.

	Reaction	Propensity
1	RNA $\longrightarrow$ RNA+M	c_1 RNA
2	M $\longrightarrow \emptyset$	c_2 M
3	DNA.D $\longrightarrow$ RNA +DNA.D	c_3 DNA.D
4	RNA $\longrightarrow \emptyset$	c_4 RNA
5	DNA+D $\longrightarrow$ DNA.D	c_5 DNA D
6	DNA.D $\longrightarrow$ DNA +D	c_6 DNA.D
7	DNA.D + D $\longrightarrow$ DNA.2D	c_7 DNA.D D
8	DNA.2D $\longrightarrow$ DNA.D +D	c_8 DNA.2D
9	M+M $\longrightarrow$ D	$\frac{1}{2}c_9$ M(M-1)
10	D $\longrightarrow$ M+M	c_{10} D

TABLE X: Description of the Goutsias model of regulated transcription. Avogadro's number is $A = 6.0221415 \times 10^{23}$ and V is the volume of the cell, fixed in this example to $10^{-15}L$. Rate constants: $c_1 = 0.043$, $c_2 = 0.0007$, $c_3 = 0.0715$, $c_4 = 0.0039$, $c_5 = \frac{0.012\times 10^9}{AV}$, $c_6 = 0.4791$, $c_7 = \frac{0.00012\times 10^9}{AV}$, $c_8 = 0.8765 \times 10^{-11}$, $c_9 = \frac{0.05\times 10^9}{AV}$, $c_{10} = 0.5$ We use the same initial state of Ref. 14, with $[M,\ D,\ RNA,\ DNA,\ DNA.D,\ DNA.2D]^T = [2,\ 6,\ 0,\ 2,\ 0,\ 0]^T$.

An analytic formula for the stationary distribution of the reversible dimerization can be used.[14,16] Table XI shows a speed up of a few orders of magnitude, while the accuracy is quite reasonable: the means of RNA, DNA, $DNA.D$ and $DNA.2D$ agree to three significant figures. Also, $\mathbb{E}(S_T) = 25.74$, under $\boldsymbol{A}$, while $\mathbb{E}(S_T) = 25.73$ under $\boldsymbol{B}$. The full CME requires a projection size of over two million, while the tQSSA reduces this to 100,000 so the reduction in dimension through aggregation is significant.

	runtime(s)	$\|\cdot\|_1$	$\|\cdot\|_2$	$\|\cdot\|_\infty$
$\boldsymbol{B}$	10 ($\boldsymbol{A}$: 1820)	5E-3	3E-4	1E-4

TABLE XI: Comparison of methods for Goutsias' model of the λ-phage switch[14] with $t_f = 200$. Accuracy of the tQSSA assessed in terms of aggregated solutions: $\left\|\boldsymbol{E}e^{t_f\boldsymbol{A}}v - e^{t_f\boldsymbol{B}}(\boldsymbol{E}v)\right\|$.

VIII. DISCUSSION

One of the virtues of the new tQSSA CME-solver is that it can be automated, in contrast to its ODE counterpart or previous, semi-analytic approaches. Furthermore, the methods for obtaining the equilibrium distributions of the fast operator could equally well be incorporated into a modified SSA. This would avoid the need for the analysis given in previous works,[13,15,16] at the cost of a modest (off-line) computational overhead.

While some previous works have relied on visual comparisons of samples of stochastic trajectories, and perhaps also on comparisons of estimates of moments, this work has allowed another method of assessing the accuracy of the approximations. In particular it allows a comparison of the numerical solution to the approximate CME that the methods do satisfy with the numerical solution of the true CME that they are intended to satisfy. For example, Figures 4 and 6 show that although a visual comparison of the distributions can be quite reasonable, the 1-norm of the error may be significant. Thus we must be cautious of being too optimistic based on visual comparisons alone. Of course order properties[9,11] provide another reasonable approach.

As noted in previous works[18] a type of exact τ-Leap may be implemented by repeatedly applying a CME-solver and then sampling from the result. Numerical experiments with the Michaelis-Menten model show that when we scale c_1 and c_2 by 1000, it is actually competitive to use such a leap method based on an embedded tQSSA-based CME-solver, instead of the SSA. However it must be acknowledged that this scaling exaggerates the difference between fast and slow reactions to such an extent that it is a contrived example.

We now make some remarks about the limitations of our approach. First, it is important to be able to identify fast and slow reactions within a system, so the QSSA is a technique that is well-suited only to a certain class of problems that exhibit very different time-scales, with some chemical reactions having much greater propensities than others. Even when the model is amenable to a type of QSSA further research into how to identify the optimal splitting, and then how to automate this, is needed.

Secondly, the inherently high-dimensional nature of the CME provides a challenge for all numerical methods. Recently considerable progress has been made and modest-sized problems are becoming feasible via various techniques.[12,17,18,36] The tQSSA essentially reduces the dimension of the problem and thus provides yet another way to cope with the curse of dimensionality.

Thirdly, the certificate of accuracy that comes with solving the full CME is lost after application of the tQSSA, although numerical results indicate that the accuracy is acceptable. Also, the connection to the theory of aggregation provides a natural framework for developing approximations of higher quality and in particular as $\boldsymbol{E} \to \boldsymbol{I}$, the accuracy would be expected to improve (becoming exact in the limit). A simple strategy for im-

proving the accuracy in certain regions is thus not to aggregate in those regions, for example.

IX. CONCLUSIONS

We have developed an analogue of the methods in Rao and Arkin[16] for the numerical solution of the CME, as well as suggesting a natural mathematical framework for generalizing this to a family of approximations of increasing quality. In this context we have contributed further to the theoretical treatment of the approximations by making some important connections to the literature on aggregation and the tQSSA. In particular we have suggested a natural interpretation of the tQSSA in the stochastic setting. This has allowed a more thorough assessment of the accuracy of the previous numerical methods, as well as resulting in a CME-solver that is more computationally efficient. For the purpose of comparison with previous works, the new methods have been successfully demonstrated on all of the models that Rao and Arkin consider, namely the Michaelis-Menten enzyme kinetics, competitive inhibition and a component of the lambda phage genetic switch. In addition it has been demonstrated on a model of dual phosphorylation and the Goldbeter-Koshland switch. Overall the application of the tQSSA CME-solver was extremely successful since it dramatically reduces the size of the problem and speeds up the computation very considerably, while maintaining acceptable accuracy.

Acknowledgements

Prof. Kevin Burrage would like to thank the Australian Research Council (ARC) for his funding of a Federation Fellowship.

APPENDIX A: ANALYTIC FORMULA FOR THE MICHAELIS-MENTEN QUASI-EQUILIBRIUM DISTRIBUTIONS AND STRUCTURE OF $\boldsymbol{B}$

Recall that we have already defined the aggregation operator $\boldsymbol{E}_t$ in Section VI F as aggregating states with the same value of S_T. It remains to compute the equilibrium distributions of each block, from which we define the columns of $\boldsymbol{F}$. We can use the computational methods described in Section VI D but for this simple example we can also identify an analytic formula, by making the ansatz of *detailed balance*[3] and using the recursive strategy outlined in Appendix A of Ref. 13, for example. For each block corresponding to $S_T = 0, 1, \ldots, S_I$, let $\hat{P}(S|S_T)$ denote the probability of the state with S substrates, according to the quasi-equilibrium distribution. Then with $S_{min} \equiv \max(0,\ S_T - E_I)$ and $\hat{P}(S_{min}|S_T) \equiv 1$, $\hat{P}(\cdot|S_T)$ (not yet normalized) is defined recursively, for $S = S_{min}, \ldots, S_T - 1$, by:

$$\begin{aligned}\hat{P}(S+1|S_T) &= K\frac{S_T - S}{(E_I - [S_T - (S+1)])(S+1)}\hat{P}(S|S_T),\\ A_{S_T} &\equiv \Big(\sum_{S=S_{min}}^{S_T}\hat{P}(S|S_T)\Big)^{-1},\end{aligned}$$

where $K \equiv \frac{c_2}{c_1}$ is the equilibrium constant for the reversible reactions and A_{S_T} is a normalization constant. We use the normalized solution for $\boldsymbol{F}$. Previous studies[14,16] make the choice $K = 1$. With this choice the peak of the distribution is near S_{min}, so the above recursive evaluation of the formula is a reasonable approach, although we still normalize at each step of the recursion.

As noted in Section VI F we may use the previous result to compute the modified propensities, $\alpha(S_T \to S_T - 1)$, for the reduced model as:

$$c_3\mathbb{E}[C|S_T] = c_3 A_{S_T}\sum_{S=S_{min}}^{S_T-1}(S_T - S)\hat{P}(S|S_T),$$

for $S_T = S_I, S_I - 1, \ldots, 1$. Here we have used the identity $\hat{P}(C|S_T) \equiv \hat{P}(S|S_T)$ for $C = S_T - S$, which follows from the conservation law. Enumerating the states in the same order (i.e. increasing order of the number of products), the matrix $\boldsymbol{B}$ is bidiagonal, of size $S_I + 1$, with

$$\begin{aligned}b_{ii} &\equiv -b_{i+1,i}\\ b_{i+1,i} &\equiv \alpha(S_I - i + 1 \to S_I - i)\,.\end{aligned}$$

The last column is all zeros, corresponding to the absorbing state, and the eigenvalues are the diagonal terms: zero together with the $\alpha(S_I - i + 1 \to S_I - i)$. For both examples in Table I the size of $\boldsymbol{B}$ is only 101, and in general the size will grow only linearly with S_I instead of quadratically as for $\boldsymbol{A}$. In fact a matrix of size only 101 is small enough that it is quite competitive to simply form the exponential in full in *MATLAB* for example. However, forming the exponential in full is not an approach that will scale to larger models.

[1] W. J. Blake, M. Kærn, C. R. Cantor, and J. J. Collins, Nature **422**, 633 (2003).
[2] N. Federoff and W. Fontana, Science **297**, 1129 (2002).
[3] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry* (Elsevier Science, 2001).
[4] A. Arkin, J. Ross, and H. McAdams, Genetics **149**, 1633 (1998).
[5] D. T. Gillespie, J. Phys. Chem. **81**, 2340 (1977).
[6] D. T. Gillespie, *Markov Processes: An Introduction for Physical Scientists* (Academic Press, Harcourt Brace Jo-

vanovich, 1992).
[7] D. T. Gillespie, J. Chem. Phys. **115**, 1716 (2001).
[8] K. Burrage, S. Mac, and T. Tian, Lecture Notes in Control and Inform. Sci. **341**, 359 (2006).
[9] M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie, Multiscale Model. Simul. **4**, 867 (2005).
[10] T. Tian and K. Burrage, J. Chem. Phys. **121**, 10356 (2004).
[11] K. Burrage, T. Tian, and P. Burrage, Prog. Biophys. Mol. Biol. **85**, 217 (2004).
[12] P. Lötstedt and L. Ferm, Multiscale Model. Simul. **5**, 593 (2006).
[13] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **122**, 014116 (2005).
[14] J. Goutsias, J. Chem. Phys. **122**, 184102 (2005).
[15] E. L. Haseltine and J. B. Rawlings, J. Chem. Phys. **117**, 6959 (2002).
[16] C. V. Rao and A. P. Arkin, J. Chem. Phys. **118**, 4999 (2003).
[17] K. Burrage, M. Hegland, S. MacNamara, and R. Sidje, in 150^{th} *Markov Anniversary Meeting, Charleston, SC, USA*, edited by A. Langville and W. Stewart (Boson Books, 2006), pp. 21–38.
[18] S. MacNamara, K. Burrage, and R. Sidje, Multiscale Model. Simul. (submitted).
[19] S. F. MacNamara, R. B. Sidje, and K. Burrage, in *Computational Techniques and Applications Conference* (James Cook University, Townsville, Australia, submitted 2006).
[20] M. Elowitz, M. Surette, P. Wolf, J. Stock, and S. Leibler, J. Bacteriol. **181(1)**, 197 (1999).
[21] D. Nicolau, Jr., K. Burrage, R. G. Parton, and J. Hancock, Molecular and Cellular Biology **26**, 313 (2006).
[22] M. Rathinam, L. Petzold, Y. Cao, and D. Gillespie, J. Chem. Phys. **119**, 12784 (2003).
[23] D. T. Gillespie and L. R. Petzold, J. Chem. Phys. **119**, 8229 (2003).
[24] C. Moler and C. Van Loan, SIAM Review **45(1)**, 3 (2003).
[25] K. Burrage, *Parallel and sequential methods for ordinary differential equations* (Oxford University Press, Oxford, 1995).
[26] T. Kato, *Perturbation theory for linear operators* (Springer-Verlag, 1976).
[27] J. R. Norris, *Markov chains* (Cambridge, 1997).
[28] B. Munsky and M. Khammash, J. Chem. Phys. **124**, 044104 (2006).
[29] R. B. Sidje, ACM Trans. Math. Software **24**, 130 (1998, www.expokit.org).
[30] R. B. Sidje and W. J. Stewart, Comput. Statist. Data Anal. **29**, 345 (1999).
[31] A. Bouras and V. Frayssé, SIAM J. Matrix Anal. Appl. **26(3)**, 660 (2005).
[32] V. Simoncini and D. Szyld, SIAM J. Sci. Comput. **25**, 454 (2003).
[33] S. Peleš, B. Munsky, and M. Khammash, J. Chem. Phys. **125**, 204104 (2006).
[34] G. Strang, SIAM J. Numer. Anal. **5**, 506 (1968).
[35] Y. Saad, Siam J. Numer. Anal. **29**, 209 (1992).
[36] M. Hegland, C. Burden, L. Santoso, S. MacNamara, and H. Booth, Journal of computational and applied mathematics (2006).
[37] W. J. Stewart, *Introduction to the Numerical Solution of Markov Chains* (Princeton University Press, 1994).
[38] C. J. Burke and M. Rosenblatt, The Annals of Mathematical Statistics pp. 1112–1122 (1958).
[39] J. G. Kemeny and J. L. Snell, *Finite Markov Chains* (Springer, 1976).
[40] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, et al., *LAPACK Users' Guide* (Society for Industrial and Applied Mathematics, Philadelphia, PA, 1999), 3rd ed., ISBN 0-89871-447-8 (paperback).
[41] R. H. Chan, Numerische Mathematik **51**, 143 (1987).
[42] J. A. M. Borghans, R. J. De Boer, and L. A. Segel, Bulletin of Mathematical Biology **58**, 43 (1996).
[43] M. G. Pedersen, A. M. Bersani, and E. Bersani, Bulletin of Mathematical Biology **69**, 443 (2007).
[44] M. G. Pedersen, A. M. Bersani, and E. Bersani, Accepted for publication in J. Math. Chem. (Preprint Me.Mo.Mat. N. 6/2006).
[45] M. G. Pedersen, A. M. Bersani, E. Bersani, and G. Cortese, Accepted for publication in Mathematics and Computers in Simulation (Proceedings 5th MATHMOD Conference, ARGESIM Report n. 30, Vienna University of Technology Press, 2006).
[46] A. R. Tzafrifri, Bulletin of Mathematical Biology **65**, 1111 (2003).
[47] T. G. Kurtz, J. Chem. Phys. **57**, 2976 (1972).
[48] L. A. Segel, Bulletin of Mathematical Biology **50**, 579 (1988).
[49] L. A. Segel and M. Slemrod, SIAM Review **31**, 446 (1989).
[50] F. Heineken, H. Tsuchiya, and R. Aris, Mat. Biosciences **1** (1967).
[51] C.-Y. F. Huang and J. E. Ferrell, Jr., Proc. Natl. Acad. Sci. **93**, 10078 (1996).
[52] A. Goldbeter and D. Koshland, PNAS **78**, 6840 (1981).