Journal of Computational Physics 346 (2017) 1–13

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Coupling sample paths to the thermodynamic limit in Monte Carlo estimators with applications to gene expression

Ethan Levien [*], Paul C. Bressloff

Department of Mathematics, University of Utah, Salt Lake City, UT 84112, USA

ARTICLE INFO

Article history:
Received 8 September 2016
Received in revised form 27 May 2017
Accepted 31 May 2017
Available online 13 June 2017

Keywords:
Chemical reaction networks
Monte Carlo
Variance reduction
Piecewise deterministic Markov process
Gillespie algorithm

ABSTRACT

Many biochemical systems appearing in applications have a multiscale structure so that they converge to piecewise deterministic Markov processes in a thermodynamic limit. The statistics of the piecewise deterministic process can be obtained much more efficiently than those of the exact process. We explore the possibility of coupling sample paths of the exact model to the piecewise deterministic process in order to reduce the variance of their difference. We then apply this coupling to reduce the computational complexity of a Monte Carlo estimator. Motivated by the rigorous results in [1], we show how this method can be applied to realistic biological models with nontrivial scalings.

1. Introduction

Large stochastic biochemical reaction networks are a popular modeling framework for investigating cellular processes [2], but the complexity and population sizes involved in realistic models pose major computational challenges. However, when there is a separation of scales, such models lend themselves to a number of model reduction techniques that are useful for course grained analysis. One example occurs when there is a separation in species abundances [3,4]. If some subset of chemical species in a reaction network are extremely abundant, then reaction channels involving those species will generally occur much faster than reactions involving less abundant species. Another examples occurs when the model parameters vary over many orders of magnitude. For example, even if a species is in a very low abundance, it is possible that the reaction rates are such that a certain reaction involving this species occurs on a different timescale than other reactions involving the same species. One approach to analyzing the qualitative properties of such a multiscale model involves rescaling the system and taking a thermodynamic limit to obtain a piecewise deterministic Markov process (PDMP). A number of recent studies have provided rigorous errors bounds for this type of reduction [5,3,6,7]. While the PDMP yields useful information about stochastic effects of the rare species, quantitative information about the stochastic fluctuations of the abundant species is lost. On the other hand, in many systems, particularly those with feedback between the rare and abundant chemical species, there is an interest in quantifying the stochastic effects due to these fluctuations [8]. A common method for resolving these fluctuations is the diffusion approximation. While the diffusion approximation is often thought to be computationally advantageous, recent work on classically scaled population models has shown that this method yields only moderate computational gains [9]. Moreover, the error between the PDMP and the exact model is fixed. However, it is sometimes desirable to control this quantity, especially when the separation of scales is only moderate.

[*] Corresponding author.
E-mail addresses: levien@math.utah.edu (E. Levien), bressloff@math.utah.edu (P.C. Bressloff).

http://dx.doi.org/10.1016/j.jcp.2017.05.050
0021-9991/© 2017 Elsevier Inc. All rights reserved.

An alternative to multiscale reduction techniques is to develop methods for accelerating stochastic simulation algorithms such as the Gillespie algorithm [10–12]. For example, there have been numerous studies of the method of τ-leaping in an effort to accelerate simulations of continuous time Markov chains [13,14]. More recently, multi-level methods that couple τ-leaping approximations at different resolutions have been used to reduce variances in Monte Carlo estimators [15,16,9]. Variance reduction techniques that utilize probabilistic couplings have also appeared earlier in the context of stochastic differential equations (SDEs) and Markov Chain Monte Carlo methods [17,18]. While there has been some work that leverages multiscale reduction techniques for Monte Carlo estimators [19], to our knowledge the idea of using these techniques directly as a variance reduction tool has not been studied.

In this paper we explore the idea of coupling reduced models to exact models as a variance reduction tool for Monte Carlo estimators. We develop a new efficient Monte Carlo estimator for multiscale chemical reaction networks that are sufficiently near a thermodynamic limit. The key insight is that, since only a small fraction of the degrees of freedom of the PDMP are stochastic, one can efficiently compute statistics of the process. On the other hand, if one wants to resolve demographic noise in the full model it is necessary to perform a large number of Monte Carlo simulations. By coupling the full stochastic model to the PDMP one can reduce the variance by a factor inversely related to the system size, and hence a smaller number of simulations need to be performed to achieve a given error tolerance. For practical applications the desired error tolerance of the Monte Carlo estimator scales with this factor. Hence, the coupled Monte Carlo estimator has the potential to speed up computations by a fractional power of the error tolerance. Our results extend the idea of variance reduction developed in [15,16], and provide a new computational application of the theory developed in previous work on PDMP approximations, or partial thermodynamic limits [5,3,6].

The paper is organized as follows. In section 2 we introduce some background material related to mathematical modeling chemical process. In section 3 we derive an approximation by taking a thermodynamic limit of a multi-scale system. We also discuss our approach to simulating this process. Our approach to variance reduction is introduced in section 4, and in section 4.1 we present an algorithm for coupling the exact process to the multi-scale approximation. Finally, we apply our method to models of gene expression in section 5 where it is shown that significant computational gains can be made in comparison to crude Monte Carlo methods.

2. Background

2.1. Stochastic chemical reaction networks in the classical setting

We consider a system involving d chemical species, denoted $\mathcal{X} = \{\mathcal{X}_i\}_{i\in\mathcal{I}}$ with $\mathcal{I} = \{1, 2, \dots, d\}$. The species interactions are prescribed by p reaction channels, denoted $\mathcal{R} = \{\mathcal{R}_j\}_{j\in\mathcal{J}}$ with $\mathcal{J} = \{1, 2, \dots, p\}$. Let x_i be the number of $\mathcal{X}_i$ and set $x = (x_1, \dots, x_d)$. Then the j-th reaction takes the form

$$\mathcal{R}_j : \sum_{i=1}^{d} K_{j,i}^{\text{in}} \mathcal{X}_i \longrightarrow \sum_{i=1}^{d} K_{j,i}^{\text{out}} \mathcal{X}_i,$$

where $K_{j,i}^{\text{in}}, K_{j,i}^{\text{out}}$ are known as *stochiometric coefficients*. When such a reaction occurs the state x is changed according to

$$x_i \to x_i + K_{j,i}, \quad K_{j,i} = K_{j,i}^{\text{out}} - K_{j,i}^{\text{in}}.$$

More complicated multi-step reactions can always be decomposed into these fundamental single-step reactions with appropriate stochiometric coefficients. In practice, most reactions involve collisions between pairs of molecules, so that $\sum_i K_{j,i}^{\text{in}} = 1$ or 2. In the so-called *classical setting*, the abundances of each species are assumed to be the same order of magnitude. Therefore, if all the abundances are large then we can describe the evolution of the system by a set of deterministic kinetic equations involving the scaled variables $z_i = x_i/S$. Here S is a dimensionless quantity representing the system size, which in gene networks is typically taken to be the characteristic number of proteins. Alternatively, it could represent some volume scale factor. For a set of p reactions, the kinetic equations take the form (strictly speaking in the thermodynamic limit $S \to \infty$)

$$\frac{dz_i}{dt} = \sum_{j=1}^{p} K_{j,i} \left[\kappa_j \prod_{l=1}^{d} z_l^{K_{j,l}^{\text{in}}} \right] \equiv \sum_{j=1}^{p} K_{j,i} \bar{\alpha}_j(z), \tag{2.1}$$

where κ_j is a constant that depends on the probability that a collision of the relevant molecules actually leads to a reaction. The product term is motivated by the idea that in a well-mixed container there is a spatially uniform distribution of each type of molecule, and the probability of a collision depends on the probability that each of the reactants is in the same local region of space. Ignoring any statistical correlations, the latter is given by the product of the individual components. The S independent functions $\bar{\alpha}_j$ are known as transition intensities or *propensities*. These intensities are the leading order term of the scaled transition intensities for a finite population ($S < \infty$), which can be derived combinatorially. These are given by

$$\alpha_j(xS^{-1}) = \frac{\kappa_j}{S^{\sum_i K^{\text{in}}_{j,i}-1}} \prod_{i=1}^{l} \frac{x_i!}{(x_i - K^{\text{in}}_{j,i})!} = \bar{\alpha}_j(xS^{-1}) + \mathcal{O}(S^{-1}). \tag{2.2}$$

Note that the factor of S ensures that the probability of a reaction depends on the probability that the interacting molecules are in the same local region of space, whereas the combinatorial factor takes into account the fact the probability that r molecules of the same species are within a reaction radius is proportional to $x_i(x_{-}1)\dots(x_i - r + 1)$, which reduces to x_i^r in the large S limit. For a more detailed discussion of the physical principles underlying chemical reactions see [20]. In terms of α_j, the unscaled transition probabilities are given by

$$\mathbb{P}(\text{one } j \text{ transition in time } dt | X(t) = x) = S\alpha_j(xS^{-1})\,dt\,.$$

Given this notation, it is straightforward to write down the corresponding chemical master equation for finite S, which takes into account intrinsic fluctuations in the number of molecules (demographic noise). Setting $P(x,t) = \mathbb{P}(x(t) = x | x(0) = x_0)$, the chemical master equation is

$$\frac{dP(x,t)}{dt} = S\sum_{j=1}^{p} \alpha_j((x - K_j)/S)P(x - K_j, t) - \alpha_j(x/S)P(x,t). \tag{2.3}$$

If we let the random variable $X_i(t)$ denote the number of molecules in species i, then an explicit representation of $X_i(t)$ is given by

$$X_i(t) = X_i(0) + \sum_{j=1}^{p} R_j(t)K_{j,i} \tag{2.4}$$

where $R_j(t)$ denotes the number of times reaction j has occurred by time t, and is a jump process for which the jump rate is locally given by $S\alpha_j(X(t)/S)$. It can be proved that if $\Pi(t)$ is a unit rate Poisson process, a representation of $R_j(t)$ is given by the *time change representation* [21],

$$R_j(t) = \Pi\left(\int_0^t S\alpha_j(X(s)/S)ds\right).$$

It follows that the expected number of reactions will grow linearly with S. Moreover, using the law of large numbers for Poisson processes,

$$\lim_{S\to\infty} \frac{\Pi(ST)}{S} = T,$$

one has

$$\lim_{S\to\infty} \frac{R_j(t)}{S} = \int_0^t \bar{\alpha}_j(z(s))ds$$

provided $\alpha_j(X(t)/S) = \mathcal{O}(1)$. This fact can be used to derive deterministic equations (2.1) from (2.4).

Algorithm 1 Simulation of fully stochastic model.

```
Initialize $X(0)$ and set $t_0 = 0$ and $k = 0$.
while $t_k \le T$ do
    Compute $\Lambda = \sum_{j=1}^{p} \alpha_j(X(t_k))$
    Generate random numbers $\tau \sim \text{Exp}(\Lambda)$ and $r \sim \text{Unif}(0,1)$
    $j_{k+1} := \min_m \left\{m : \sum_{j=1}^{m} \alpha_j(X(s)) < r\Lambda\right\}$
    $t_{k+1} := t_k + \tau$
    $k \to k+1$
    $X(t_k) = X(t_{k-1}) + K_{j_k}$
```

For the propose of simulation it is common to construct the aggregate process, $\sum_{j=1}^{p} R_j(t)$, instead of simulating the individual terms in the sum. To do this, ones notes that if $t_1, t_2, \dots$ are the jump times of $X_i(t)$, then the interjump times satisfy

$$\mathbb{P}(t_k - t_{k-1} < t | X(t_{k-1}) = x) = 1 - \exp\left\{-t\sum_{j=1}^{p} \alpha_j(x)\right\}.$$

In other words, $t_k - t_{k-1}$ is exponentially distributed with a rate given by the aggregate jump rate $\sum_{j=1}^{p} \alpha_j(x)$. The reaction that fires after the k-th jump can then be selected by generating a uniform random variable r over $[0, 1]$. In particular, given that $X(t_{k-1}) = x$, the index j_k of the k-th reaction is given by

$$j_k = \min \left\{ j : \sum_{m=1}^{j} \alpha_m(x/S) < r \sum_{m=1}^{p} \alpha_m(x) \right\}$$

This procedure for generating samples paths of $X(t)$ is known as *Gillespie's algorithm* and a detailed description is given in Algorithm 1

2.2. Monte Carlo simulations in the classical setting

Suppose we wish to approximate some statistics of the scaled state variables $Z(t) = X(t)/S$ using a crude Monte Carlo estimator (MCE) $\widehat{Q}_{\text{crude}}(M)$ with M realizations of the process. That is,

$$\widehat{Q}_{\text{crude}}(M) = \frac{1}{M} \sum_{j=1}^{M} f(Z_{[j]}(T)). \tag{2.5}$$

Here and elsewhere we use the convention that the subscript $[j]$ indicates a specific realization of a process. In order for $\widehat{Q}_{\text{crude}}(M)$ to approximate $\mathbb{E}[f(Z(T))]$ to $\mathcal{O}(\varepsilon)$ in the sense of confidence intervals (i.e. the standard deviation of $\widehat{Q}_{\text{crude}}(M)$ is $\mathcal{O}(\varepsilon)$), we require $M = \mathcal{O}(\varepsilon^{-2}\text{Var}(f(Z(T)))$ [22]. Here we have used the fact that $\mathbb{E}[f(Z(T))] = \mathcal{O}(1)$.

Now suppose we seek an approximation of $\mathbb{E}[f(Z(T))]$ for some fixed value of S. If S is extremely large, $Z(T)$ is almost deterministic and any information that distinguishes $f(Z(T))$ from $f(z(t))$ will be inversely proportional to a power of S. As a consequence, if ε is too large, then all of the interesting information about the stochastic model will be lost in the sample noise. This observation leads us to conclude that the limiting behavior in S of any numerical method for approximating $\mathbb{E}[f(Z(T))]$ should not be studied independently of ε, and it is therefore import to quantify the relationship between these two parameters. With this in mind, we introduce the dimensionless parameter $\delta > 0$ defined as the ratio

$$\delta = -\frac{\ln \varepsilon}{\ln S}.$$

Roughly speaking, if the ratio is large the accuracy of our estimate relative to the $\mathcal{O}(S)$ fluctuations will be high. δ can be interpreted as a relative accuracy with respect to the noise in the model, which is in contrast to the absolute measure of accuracy ε. It should be emphasized that δ is introduced for the propose of analysis and in practice ε is usually selected based on other considerations. However, as we are interested in studying the asymptotic complexity in terms of S, we will take the relative accuracy δ to be fixed. Rearranging the expression for δ yields the formula for ε:

$$\varepsilon = \varepsilon(S) = S^{-\delta}.$$

This expression previously appeared in work on classically scaled models, where the efficiency of many existing Monte Carlo methods was studied in terms of δ [9]. In a similar spirit, we will judge the effectiveness of the methods proposed in this paper in terms of this parameter.

In terms of ε, the expected number of computations, or *complexity*, of the MCE $\widehat{Q}_{\text{crude}}(M)$ is

$$\mathsf{C}_{\text{crude}} = \mathcal{O}(\varepsilon^{-2}\mathsf{C}_X \text{Var}(f(Z(T))),$$

where C_X is the complexity of generating the sample $Z(T)$:

$$\mathsf{C}_X = \mathbb{E}[\text{ \# of computations to simulate } Z(T)].$$

In the classical setting, it can be shown that $\text{Var}(f(Z(T)) = \mathcal{O}(S^{-1})$, which corresponds to the stochastic model approaching a deterministic limit at a rate inversely proportionally to the size of the system [9]. On the other hand, $\mathsf{C}_X = \mathcal{O}(S)$, so that in the classical setting, the contribution of the complexity from the simulation of the path cancels with the variance of the path, and we obtain

$$\mathsf{C}_{\text{crude}} = \mathcal{O}(\varepsilon^{-2}).$$

Essentially, when a stochastic model approaches a deterministic limit, variance reduction in terms of the system size is "for free". This applies not only to the crude MCE, but any Monte Carlo method applied to a biochemical model in the classical setting. We refer to [9] for a rigorous analysis of Monte Carlo methods in the classical setting. The crucial observation that motivates the developments in this paper is that in the multiscale setting, where some species abundances do not scale with S, it is generally not possible to bound $\text{Var}(f(Z(T))$ in terms of S, and the complexity of any Monte Carlo method increases by an order of magnitude.

3. The multiscale approximation

For biochemical networks of interest the classical assumption that a deterministic system is obtained in the scaling limit $S \to \infty$ is a major oversimplification. Instead, one is often interested in *multi-scale* systems [5,23,24]. Multi-scale models are characterized by the fact that the reaction propensities α_j, and hence the time between stochastic events may vary over many orders of magnitude for different reaction channels. As a result, it is not useful to approximate the system by a deterministic process, since the natural scalings of the model in S may not converge to a deterministic process. This motivates a *multiscale approximation* of the fully stochastic model, in which only a fraction of the species are taken to evolve continuously.

In order to describe the multiscale approximation, we begin by assuming that $X_i(t) = O(M_i(S))$ for some nondecreasing function $M_j(s)$. This is a generalization of the classical setting discussed earlier where $M_i(S) = S$ for each i. We then introduce the decomposition of the species indices $\mathcal{I} = (\mathcal{I}^L, \mathcal{I}^H)$ where

$$\mathcal{I}^L = \{i : M_i = \mathcal{O}(1)\}$$

and $\mathcal{I}^H = \mathcal{I} \setminus \mathcal{I}^L$. That is, $\mathcal{I}^L$ and $\mathcal{I}^H$ index the low and high copy species respectively. In this context, the high copy species are any species for which the abundances grow with S. This decomposition is generally derived from physical constraints on the model, but there is usually some freedom in how the scaling factors M_i are selected. The induced decomposition of the state variables is given by $X(t) = (X^L(t), X^H(t))$, which we rescale by setting $Z_i^H(t) = X_i^H(t)/M_j$ to obtain $Z(t) = (X^L(t), Z^H(t))$. Note that the meaning of an entry in $Z(t)$ now depends on whether the corresponding species is in high or low copy. If $Z_i(t) \in Z^H(t)$ then Z_i can be thought of as dimensionless concentration that tracks the relative abundance of a species, while if $Z_i(t) = X_i(t) \in X^L(t)$, $X_i(t)$ counts the number of species.

The separation of scales in the species abundances, along with the fact that the rate constants κ_j may vary over many orders of magnitude, induces a decomposition of the reactions: $\mathcal{J} = (\mathcal{J}^L, \mathcal{J}^H)$. As with the species, $\mathcal{J}^L$ and $\mathcal{J}^H$ are indices of the reactions for which the rates are $\mathcal{O}(1)$ and $\mathcal{O}(G_j(S))$ respectively. Unlike M_j, $G_j(S)$ is a function of S that may be monotonically increasing or decreasing. For $j \in \mathcal{J}^L$ the corresponding propensity is given by $\alpha_j(Z(t))$, whereas for $j \in \mathcal{J}^H$ we include a factor of $G_j = G_j(S)$ so that the propensity is $G_j\alpha_j(Z(t))$. Note that this implies $R_j(t)$ is $\mathcal{O}(G_j(S))$ in terms of S, which suggest that for any physical meaningful scaling $G_j(S)/M_i(S)$ is bounded in S. This is because if this ratio is not controlled, the terms $R_j(t)/M_i(S)$ and hence the random variables

$$Z_i^H(t) = X_i^H(0)/M_i(S) + \sum_{j \in \mathcal{J}^H} R_j(t)/M_i(S)$$

grow infinitely large as S grows. We do however allow that $G_j(S)/M_i(S) \to 0$, which implies the term $R_j(t)/M_i(S)$ becomes negligible for large S. It will be important to keep track of the terms for which this ratio does not vanish, hence we define

$$\mathcal{J}_i^H = \{j : \lim_{S\to\infty} G_j(S)/M_i(S) > 0\}.$$

Roughly speaking, the reactions indexed by $\mathcal{J}_i^H$ make a non-negligible contribution to the evolution of the i-th high copy variable for large S. Finally, it should be emphasized that in practice S is often fixed, and the selection of the scaling factors M_j and G_j is not based on whether certain quantities actually change with the size of the system. Instead these are selected so that the limiting system obtained below is useful. A more extensive discussion of this topic is provided in [24].

We now return to the asymptotic regime in which $S \to \infty$ and write down the process $\bar{Z}$ for which $Z = (Z^H, X^L) \to \bar{Z} = (\bar{Z}^H, \bar{X}^L)$. Here the convergence is in probability. This limit has been investigated rigorously in [24], and it was established that the dynamics of $\bar{Z}$ are given by the piecewise deterministic process (PDMP),

$$\frac{d}{dt}\bar{Z}_i^H(t) = \sum_{j \in \mathcal{J}_i^H} K_j \bar{\alpha}_j(\bar{Z}^H(t), \bar{X}^L(t)) \tag{3.1}$$

in between jumps, while $\bar{X}_i^L$ is given by the counting process

$$\bar{X}^L(t) = \bar{X}^L(0) + \sum_{j \in \mathcal{J}^L} \bar{R}_j(t) K_j \tag{3.2}$$

where $\bar{R}_j(t)$ are counting processes counting the number of jumps in each reaction channel. The reader should compare (3.1) and (3.2) to (2.1) and (2.4) respectively.

Sample paths of the process $\bar{Z}(t)$ can be generated by a modification of the Gillespie algorithm, known as the *true jump method*. We simplify notation by letting

$$\Lambda(\bar{Z}(t)) = \sum_{j \in \mathcal{J}^L} \bar{\alpha}_j(\bar{Z}(t)) \tag{3.3}$$

denote the aggregate jump rate. In order to derive the true jump method, one first notes that the interjump times, $t_k - t_{k-1}$, satisfy

$$\mathbb{P}(t_k - t_{k-1} < t | X^L(t_{k-1}) = x) = 1 - \exp\left\{ - \int_0^t \Lambda(Z^H(s + t_{k-1}), x) ds \right\}. \tag{3.4}$$

Note that we cannot simply generate $t_k - t_{k-1}$ directly from exponentially distributed random variables as is done in Gillespie's algorithm, instead one must solve the ODE (3.1) between jumps using some appropriate discretization of the continuous process and use the solution to obtain t_k. It follows from (3.4) that $\int_{t_{k-1}}^{t_k} \Lambda(\bar{Z}(s))ds \sim \text{Exp}(1)$. To understand how this statement relates to the next jump time in Gillespie's algorithm, note that if $\Lambda = \Lambda(z)$ is a constant, properties of exponentially distributed random variables can be used to deduce that $t_{k-1} - t_k \sim \text{Exp}(\Lambda)$. We can now state an explicit representation of t_k in terms of a minimization problem involving $S_k \sim \text{Exp}(1)$. This is given by

$$t_k = \inf_u \left\{ u > 0 : \int_{t_{k-1}}^{u} \Lambda(\bar{Z}(t'))dt' = S_k \right\}.$$

While a number of methods exist for approximating t_k, the approach we will take is based on solving an ODE for the time variable which can be derived as follows. First, introduce the variable $\tau(s) > t_{k-1}$. Then, setting $\int_{t_{k-1}}^{\tau(s)} \Lambda(\bar{Z}(t'))dt' = s$ and differentiating with respect to s yields the equation

$$\tau'(s) = \frac{1}{\Lambda(\bar{Z}(\tau(s)))}.$$

Solving this equation between 0 and S_k with $\tau(0) = t_{k-1}$, yields $\tau(S_k) = t_k$. Finally, $\bar{Z}^H(t_k)$ is found by setting $z(s) = \bar{Z}^H(\tau(s))$ and applying the chain rule to get an ODE for $z(s)$. To summarize, t_k along with the state of the continuous variable at t_k is given by $(\bar{Z}^H(t_k), t_k) = (z(S_k), \tau(S_k))$ where

$$\begin{cases} z'(s) = \sum_{j \in \mathcal{J}^H} \alpha_j(z(s), \bar{X}^L(t_{k-1}))K_j / \Lambda(z(s), \bar{X}^L(t_{k-1})) \\ \tau'(s) = 1/\Lambda(z(s), \bar{X}^L(t_{k-1})) \\ z(0) = \bar{Z}^H(t_{k-1}), \quad \tau(0) = t_{k-1}. \end{cases} \tag{3.5}$$

This method for computing t_k is known as the CHV method, and was recently proposed in [25] where a more detailed derivation and discussion of (3.5) can be found.

Once the time t_k is computed, the selection of the next reaction is essentially the same as in Gillespie's algorithm, except that one only selects from the reaction in $\mathcal{J}^L$. The details of this procedure to compute $\bar{Z}(t)$ with an accuracy of h are given in (2).

Algorithm 2 Simulation of the multiscale approximation.

```
Select an accuracy $h$. Initialize $\bar{Z}(0)$ and set $t_k = 0$ and $k = 0$.
while $t_k \leq T$ do
    Generate a random number $S_k \sim \text{Exp}(1)$.
    Let $(z(S_k), \tau(S_k))$ be the numerical solution to (3.5) with $t_0 = t$ and $z_0 = \bar{Z}^H(t_k)$.
    $\bar{Z}(t_k + \tau(S_k)) = z(S_k)$
    $\Lambda = \sum_{j \in \mathcal{J}^L} \alpha_j(\bar{Z}(\tau))$
    Generate a random number $r \sim \text{Unif}(0, 1)$
    $j_k := \min_i\{i : \sum_{j \in \mathcal{J}^L : j < i} \alpha_j(\bar{Z}(S_k)) < r\Lambda\}$
    $t_{k+1} := t_k + \tau(S_k)$
    $k \to k + 1$
    $\bar{X}(t_k) = \bar{X}(t_{k-1}) + K_{j_k}$
Perform numerical integration to obtain $\bar{Z}(T)$.
```

While other algorithms for generating samples paths of the multiscale approximation exist [6], we have found this method to be effective and leave a detailed comparison of the different methods to a future study. The import point to note is that the complexity of generating a sample path of the multiscale approximation does not grow with the system size. This means that is much easier to compute statistics of $\bar{Z}(t)$ than $Z(t)$ when S is large.

4. Variance reduction in the multiscale setting

As noted at the end of Section 2, for multiscale models we can generally not bound the sample variance in terms of the system size and the asymptotic complexity of Monte Carlo methods picks up a factor of S. For example,

$$\mathsf{C}_{\text{crude}} = \mathcal{O}(\varepsilon^{-2-1/\delta}).$$

While a great deal of information about the exact model is still contained in the thermodynamic limit $(\bar{Z}^H, \bar{X}^L)$, it is not being used in the computations. Again, we emphasize how this contrasts with the classical setting, where information about the deterministic limit is used to accelerate the converge of an MCE without any additional work. Generally speaking, our goal is to understand how information about the thermodynamic limit can be used in the multiscale setting.

The coupled Monte Carlo estimator we will introduce is based on the idea of variance reduction via a probabilistic coupling of the exact process with an approximate process. In our case the approximate process will be the PDMP $\bar{Z}(t)$. This idea has proven to be very useful in multilevel Monte Carlo methods [16,18] where different τ-leaping approximations are coupled. To construct an MCE in the present setting we note that

$$\mathbb{E}[f(Z(T))] = \mathbb{E}[f(Z(T)) - f(\bar{Z}(T))] + \mathbb{E}[f(\bar{Z}(T))] \tag{4.1}$$

Two observations allow us to use this decomposition to obtain statistics of $\mathbb{E}[f(Z(T))]$ more efficiently than the crude MCE (2.5). First, statistics of $\bar{Z}(T)$ can be obtained much more efficiently than statistics of the exact process when S is even moderately large relative to the abundance of the rare species. These statistics can be obtained either by a MCE using Algorithm 2 to generate the sample paths, or by non-Monte Carlo based methods which are difficult to apply to the exact process. Second, we can reduce the number of simulations we need of the full process by coupling the processes $Z(t)$ and $\bar{Z}(t)$ in a way that reduces the variance of the difference $f(Z_i(T)) - f(\bar{Z}_i(T))$. By coupling, we mean the construction of a random variable $(Z(t), \bar{Z}(t))$ for which $Z(t)$ and $\bar{Z}(t)$ are correlated, but $Z(t)$ and $\bar{Z}(t)$ have the same marginal distributions as the original processes. We will develop this coupling in section 4.1, but first let us explore in greater depth the implications of (4.1).

For the second term in (4.1) let us assume we can produce an approximation $\widehat{Q}_Z(h) \approx \mathbb{E}[f(\bar{Z}_i(T))]$ satisfying

$$\mathbb{E}[|\widehat{Q}_Z(h) - \mathbb{E}[f(\bar{Z}_i(t))]|] = \mathcal{O}(h)$$

and $\mathrm{Var}(\widehat{Q}_Z(h)) = \mathcal{O}(h^2)$ with $h < \varepsilon$. We will also need an approximate path to estimate the term $f(Z_i(T)) - f(\bar{Z}_i(T))$. This can be obtained from Algorithm 2. Then an $\mathcal{O}(\varepsilon)$ estimator $\widehat{Q}_{\mathrm{coupled}}(M_1, h)$ of (4.1) can be constructed by summing the estimator

$$\widehat{Q}_{(Z,\bar{Z})}(M_1) = \frac{1}{M_1}\sum_{j=1}^{M_1}(f(Z_{[j]}(T)) - f(\bar{Z}_{[j]}(T)))$$

and the approximation $\widehat{Q}_Z(h)$:

$$\widehat{Q}_{\mathrm{coupled}}(M_1, h) = \widehat{Q}_{(Z,\bar{Z})}(M_1) + \widehat{Q}_Z(h).$$

Technically the terms $\bar{Z}_{i,[j]}(T)$ in the first expression are computed to $\mathcal{O}(h)$ from Algorithm 2; however, we have surpassed the dependence on h since it plays no role in the analysis. Setting

$$V_{(Z,\bar{Z})}(T) = \mathrm{Var}(f(Z_i(T)) - f(\bar{Z}_i(T))), \tag{4.2}$$

the variance of the coupled MC estimator, V_{coupled}, is simply

$$V_{\mathrm{coupled}} = M_1^{-1}V_{(Z,\bar{Z})} + \mathrm{Var}(\widehat{Q}_Z(h)) \sim M_1^{-1}V_{(Z,\bar{Z})} \tag{4.3}$$

where we are assuming the first term is leading order in S, meaning that the limiting factor in reducing the variance is the simulation of the coupled path. The methods of this paper are obviously applicable when $\widehat{Q}_Z(h)$ is computed with non-Monte Carlo based methods, and hence $\mathrm{Var}(\widehat{Q}_Z(h)) = 0$, but also apply when this term is estimated from Monte Carlo simulations of the PDMP. This is because simulations of $\bar{Z}(t)$ are much cheaper than simulations of $X(t)$ and hence $\mathrm{Var}(\widehat{Q}_Z(h))$ can be made $o(\varepsilon)$ at a negligible cost. Of course for an order ε estimator, we require $V_{\mathrm{coupled}} = O(\varepsilon^2)$ so that

$$M_1 = \varepsilon^{-2}V_{(Z,\bar{Z})}. \tag{4.4}$$

It is now clear that for $\widehat{Q}_{\mathrm{coupled}}$ to be preferable over $\widehat{Q}_{\mathrm{crude}}$, it must be that (1) $V_{(Z,\bar{Z})}$ is small and (2) $\widehat{Q}_Z(h)$ is cheap to generate. (2) is true because statistics of $\bar{Z}(t)$ does not require implementing the large number of stochastic events needed for $X(t)$, while (1) is the topic of the next section.

4.1. A coupling algorithm

We now construct an algorithm for simulating a coupled process $(Z, \bar{Z})$ that keeps Z close to $\bar{Z}$, thereby minimizing the variance $V_{(Z,\bar{Z})}$. The technique for coupling the two process is based around a decomposition of the counting processes $R_j(t)$ and $\bar{R}_j(t)$ (when $j \in \mathcal{J}^L$) into a counting process that is common to both process, and a process which accounts for the fact that the rates may not be the same. Explicitly, we introduce the decompositions

$$\begin{aligned} R_j(t) &= R_{j,1}(t) + R_{j,2}(t) \\ \bar{R}_j(t) &= R_{j,1}(t) + R_{j,3}(t) \end{aligned} \tag{4.5}$$

where the counting processes $R_{j,1}(t)$ count jumps that occur in both the exact model and the multiscale approximation, while $R_{j,2}(t)$ and $R_{j,3}(t)$ account for the fact that the jump rates in the exact model differ from those in the approximation. Note that technically the equalities in (4.5) are in distribution. We let the jumps of $R_{j,1}(t)$ occur at rates equal to the minimum of the rates of $R_j(t)$ and $\bar{R}_j(t)$:

$$\alpha_{j,1}(Z(t), \bar{Z}(t)) = \min\{\alpha_j(Z(t)), \alpha_j(\bar{Z}(t))\}.$$

In order for $R_{j,1}(t) + R_{j,2}(t)$ to have the same distribution as $R_j(t)$, we require

$$\begin{aligned} \alpha_j(Z(t)) &= \alpha_{j,1}(Z(t), \bar{Z}(t)) + \alpha_{j,2}(Z(t)) \\ \alpha_j(\bar{Z}(t)) &= \alpha_{j,1}(Z(t), \bar{Z}(t)) + \alpha_{j,3}(Z(t)) \end{aligned}$$

and hence

$$\begin{aligned} \alpha_{j,2}(Z(t), \bar{Z}(t)) &= \alpha_j(Z(t)) - \min\{\alpha_j(Z(t)), \alpha_j(\bar{Z}(t))\} \\ \alpha_{j,2}(Z(t), \bar{Z}(t)) &= \alpha_j(\bar{Z}(t)) - \min\{\alpha_j(Z(t)), \alpha_j(\bar{Z}(t))\}. \end{aligned}$$

To simplify notation, we once again introduce the aggregate jump rate

$$\Lambda(Z(t), \bar{Z}(t)) = \sum_{j \in \mathcal{J}^L} \alpha_{j,1}(Z(t), \bar{Z}(t)) + \alpha_{j,2}(Z(t), \bar{Z}(t)) + \alpha_{j,3}(Z(t), \bar{Z}(t)) + \sum_{j \in \mathcal{J}^H} G_j \alpha_j(Z(t)) \tag{4.6}$$

We can now express the jump times of the coupled process $(Z(t), \bar{Z}(t))$ as

$$\mathbb{P}(t_k - t_{k-1} < t | X(t_{k-1}) = x) = 1 - \exp\left\{ -\int_0^t \Lambda(Z(t_{k-1}), \bar{Z}(s)) ds \right\}.$$

Using this equation, the jump times t_k can be computed in the exact same manner as they were for the process $\bar{Z}$ in Algorithm 2. The only modification is that the aggregate jump rate now includes all the coupled rates, as well as the rates of the jumps in $\mathcal{J}^H$ evaluated on the exact process. Explicitly, replacing (3.3) by (4.6) in (3.5) we obtain

$$\begin{cases} z_i'(s) = \sum_{j \in \mathcal{J}_i^H} \bar{\alpha}_j(z(s), \bar{X}^L(t_{k-1})) K_j \Lambda(Z^H(t_{k-1}), X^L(t_{k-1}), z(s), X^L(t_{k-1}))^{-1}, \\ \tau'(s) = \Lambda(Z^H(t_{k-1}), X^L(t_{k-1}), z(s), X^L(t_{k-1}))^{-1} \\ z(0) = \bar{Z}^H(t_{k-1}), \quad \tau(0) = t_{k-1}, \end{cases} \tag{4.7}$$

so that $(\bar{Z}^H(t_k), t_k) = (z(S_k), \tau(S_k))$. In order to simulate the coupled process, we need to select the reaction that occurs at t_k, as well as the specific term in the decompositions (4.5) that fires. We will continue to use j_k to denote the k-th reaction, and introduce the index $l_k = 1, 2, 3$ to specify the term in the decomposition (4.5). Given a uniform random variable r, j_k is given by the familiar minimization problem

$$j_k = \min_j \left\{ j : \sum_{m=1}^{j} \mathbf{1}_{\{m \in \mathcal{J}^L\}} \sum_{l=1}^{3} \alpha_{m,l}(Z(t_{k-1}), \bar{Z}(t_k)) + G_j \mathbf{1}_{\{m \in \mathcal{J}^H\}} \alpha_m(Z(t_{k-1})) < r \Lambda(Z(t_{k-1}), \bar{Z}(t_k)) \right\}. \tag{4.8}$$

If $j_k \in \mathcal{J}^L$, then we need to compute l_k which can be computed using the same value of r:

$$l_k = \min_j \left\{ l : \sum_{m=1}^{l} \alpha_{j_k,l}(Z(t_{k-1}), \bar{Z}(t_k)) < r \sum_{m=1}^{3} \alpha_{j_k,m}(Z(t_{k-1}), \bar{Z}(t_k)) \right\} \tag{4.9}$$

What we have described is essentially the true jump method applied to the coupled process $(Z(t), \bar{Z}(t))$, and a detailed description of this procedure is provided in Algorithm 3.

Algorithm 3 Simulation of coupled process.

```
Initialize $\bar{Z}(0)$ and set $t_k = 0$ and $k = 0$.
while $t_k \leq T$ do
    Generate a random number $S_k \sim \mathrm{Exp}(1)$.
    Let $(z, \tau) = (z(S_k), \tau(S_k))$ be the solution to (4.7) with $t_0 = t$ and $z_0 = \bar{Z}^H(t_k)$.
    Set $\bar{Z}^H(t_k + \tau) = z$
    Generate a random number $r \sim \mathrm{Unif}(0, 1)$
    Compute $j_k$ using (4.8).
    $k \to k + 1$.
    $t_k := t_{k-1} + \tau$
    if $j_k \in \mathcal{J}^L$ then Compute $l_k$ using (4.9).
        if $l_k = 1$ then Set $X^L(t_k) = X^L(t_{k-1}) + K_{j_k}$ and $\bar{X}^L(t_k) = \bar{X}^L(t_{k-1}) + K_{j_k}$
        else if $l_k = 2$ then Set $X^L(t_k) = X^L(t_{k-1}) + K_{j_k}$
        else if $l_k = 3$ then Set $\bar{X}^L(t_k) = \bar{Z}^L(t_{k-1}) + K_{j_k}$
    else
        Set $Z^H(t_k) = Z^H(t_{k-1}) + K_{j_k} S^{-1}$
```

Table 1
The rate constants, $G_j \kappa_j$, and propensities for the model (5.1) with $S = 10^2$.

j	$\alpha_j(Z(t))$	$G_j \kappa_j$	κ_j	G_j
1	$\kappa_1 X_2(t)$	1.3×10^2	1.3	S
2	$\kappa_2 Z_1(t)$	1.0	1.0	1
3	$\kappa_3 X_2(t)$	5.0	5.0	1
4	$\kappa_4 X_3(t)$	4.0	4.0	1

5. Application to modeling gene expression

We now demonstrate the effectiveness of our method on two models of stochastic gene expression.

5.1. A simple genetic model

A toy model of stochastic gene expression involves three species

$\mathcal{X}_1 = \mathrm{M}$	Protein monomer
$\mathcal{X}_2 = \mathrm{G}$	Gene in on state
$\mathcal{X}_3 = \mathrm{G}^*$	Gene in off state,

and the reaction network is given by

$$\begin{aligned} &G \overset{\kappa_1}{\to} G + M \\ &M \overset{\kappa_2}{\to} \emptyset \\ &G \underset{\kappa_4}{\overset{\kappa_3}{\rightleftharpoons}} G^*. \end{aligned} \tag{5.1}$$

When the gene is in state G, the protein is produced at a rate proportional to $S\kappa_1$, while when in state G* the protein is not produced. For example, G* could represent the presence of a repressor occupying the RNA polymerase binding site, thereby preventing the transcription of the gene [26,27]. This model was originally proposed in [28] to investigate the implications of stochastically in gene expression for haploinsufficiency. Note that the complex mechanism of transcription is viewed as a "back box" represented by the production rate $S\kappa_1$. The motivation for taking the rate of protein production to scale with S is that the number of proteins in the system is typical very large. The propensities of this model as well as some physically relevant values of the rate constants are provided in Table 3.

While the network is simple, it provides a useful test case for the methods developed in this paper. In particular, a natural multiscale decomposition follows immediately from the physical interpretation of the model, leading to a very simple thermodynamic limit. Since X_2 and X_3 are binary values, and S is defined to be the typical number of proteins in the system, we have $\mathcal{I}^H = \{1\}$ with $M_i = S$. It then follows from Table 1 that $\mathcal{J}^H = \{1, 2\}$. This implies that the continuous part of the dynamics in the multiscale approximation is given by

$$\frac{d}{dt}\bar{Z}_1(t) = \kappa_1 \bar{X}_2(t) ds - \kappa_2 \bar{Z}_1(t)$$

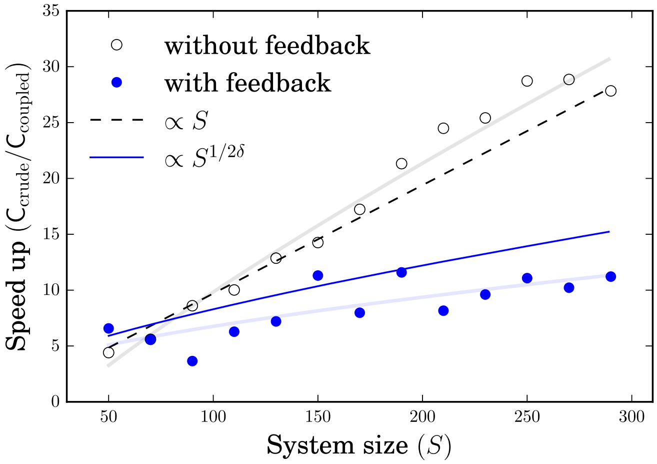

Fig. 1. The observed speedup of the coupled estimator, $\mathsf{C}_{\text{crude}}/\mathsf{C}_{\text{coupled}}$ measured in the total number of events needed to obtain an $\mathcal{O}(S^{-0.7})$ estimate of $\mathbb{E}[Z(10)]$ with initial data $Z(0) = (Z_A(0), X_B(0), X_C(0)) = (1, 0, 1)^T$. The faded dashed lines indicate the best fits for $aS^d + b$ using a non-linear least squares fit to determine a, b and d. The darker lines display the predicted speedups $aS + b$ and $aS^{1/2\delta} + b$ with least squares fits for a and b. Monte Carlo simulations were performed by generating samples until $\sigma_m m^{-1/2} \widehat{Q}_m^{-1} < \varepsilon$ where m is the number of samples generated, σ_m is the sample standard deviation and $\widehat{Q}_m$ is the estimate using those samples. We used a Python implementation of the LSODA algorithm to perform the integration step in Algorithm 2. We have confirmed that the estimates are within the expected confidence intervals. The rate constants used are from Tables 1 and 2.

Table 2
The rate constants and propensities for the model (5.2) with $S = 10^2$.

j	$\alpha_j(Z(t))$	$G_j\kappa_j$	κ_j	G_j
1	$\kappa_1 X_2(t)$	1.3×10^2	1.3	S
2	$\kappa_2 Z_1(t)$	1.0	1.0	1
3	$\kappa_3 X_2(t) Z_1(t)$	5.0	5.0	1
4	$\kappa_4 X_3(t)$	4.0	4.0	1

A bound on the complexity of the estimator can be derived from the fact that error between the coupled and exact process in L^2 is $\mathcal{O}(S^{-1})$ [1]. This implies $V_{(Z,\bar{Z}_h)} = \mathcal{O}(\varepsilon^{1/\delta})$. Recalling (4.4), we have $M_1 = \varepsilon^{1/\delta - \delta}$, and since the asymptotic number of computations to generate a sample of $(Z, \bar{Z})$ is equal to $\mathsf{C}_Z = \mathcal{O}(S)$, we obtain the asymptotic complexity

$$\mathsf{C}_{\text{coupled}} \sim M_1 \mathsf{C}_Z = \mathcal{O}(\varepsilon^{-\delta}).$$

This implies $\mathsf{C}_{\text{crude}}/\mathsf{C}_{\text{coupled}} = S$, which is independent of δ and therefore one benefits from using the coupling technique for any value of δ. In practice this means that in a sufficiently large systems, the coupling method will produce computational gains regardless of the desired accuracy. Note that this is exactly the complexity $\mathsf{C}_{\text{crude}}$ in the classical setting. In this sense the coupling gives us the variance reduction that we get *for free* in the classical setting. This result is confirmed by numerical experiments performed over a range of system sizes, see Fig. 1.

If we drop the assumption that the dynamics of the gene do not depend on the concentration of the protein (that is, we add feedback), then we can obtain a similar limiting systems, but the complexity analysis must be revisited. For example, suppose we modify the network so that the concentration of the protein catalyses the transition of the gene into the off state:

$$\begin{aligned} G &\xrightarrow{\kappa_1} G + M \\ M &\xrightarrow{\kappa_2} \emptyset \\ M + G &\xrightarrow{\kappa_3} M + G^* \\ G^* &\xrightarrow{\kappa_4} G. \end{aligned} \tag{5.2}$$

Not that this new network can be studied under the same scaling as the network without feedback, see Table 2. The continuous evolution of $\bar{Z}_1(t)$ is therefore unchanged from the previous model.

Note that we are assuming the physical (unscaled) rate constants are the same as those for the simpler model considered above, but the structure of the model has changed. This corresponds to the physical assumption that the switching depends on the concentration of the protein and not the total number of proteins in the system. We have proven (see [1]) that for models of this type we have the larger L^2 error of $\mathcal{O}(S^{-1/2})$, and hence the slower converge

$$\mathsf{C}_{\text{coupled}} \sim M_1 \mathsf{C}_Z = \mathcal{O}(\varepsilon^{-\delta - 1/2\delta}).$$

Table 3
The physically prescribed rate constants and a possible set of scaling exponents for (5.3) with $S = 10^2$.

j	$\alpha_j(Z(t))$	$G_j\kappa_j$	κ_j	G_j
1	$\kappa_1 X_3(t)$	4.3×10^{-2}	4.3	S^{-1}
2	$\kappa_2 Z_1(t)$	7.0×10^{-2}	7.1	S^{-1}
3	$\kappa_3 X_5(t)$	7.15×10^{-2}	7.15×10^{-2}	1
4	$\kappa_4 X_3(t)$	3.9×10^{-3}	3.9×10^{-3}	1
5	$\kappa_5 Z_2(t) X_4(t)$	1.99×10^{-2}	1.99×10^{-2}	1
6	$\kappa_6 X_5(t)$	4.79×10^{-1}	4.79×10^{-1}	1
7	$\kappa_7 X_5(t) Z_2(t)$	1.99×10^{-3}	1.99×10^{-1}	S^{-2}
8	$\kappa_8 X_6(t)$	8.7×10^{-12}	8.7×10^{-8}	S^{-2}
9	$\kappa_9 Z_1(t)(Z_1(t) - 1/S)$	8.3×10^{2}	8.3	S
10	$\kappa_{10} Z_2(t)$	5.5×10^{1}	5.5×10^{-1}	S

In contrast to the results stated for systems without feedback, we now have the δ dependent speed up $\mathrm{C}_{\text{crude}}/\mathrm{C}_{\text{coupled}} = S^{1/2\delta}$. As δ becomes large the coupling method becomes less effective. Note that this result does not say anything about the constant appearing in the $\mathcal{O}(\cdot)$ term, but does give us a rough idea of when we can expect the method to be effective. In the present context we have found that method is effective for moderate S provided $\delta < 1$. For example, in Fig. 1 we display results for $\delta = 0.7$. If $S = 10^2$, this corresponds to an accuracy of approximately $\varepsilon = 0.025$.

5.2. Goutsias's model of regulated transcription

In order to establish that our method is indeed applicable to more realistic models appearing in biology, we will examine a model of regulated gene transcription that involves six species,

$\mathcal{X}_1 = \mathrm{M}$	Protein monomer
$\mathcal{X}_2 = \mathrm{D}$	Transcription factor
$\mathcal{X}_3 = \mathrm{RNA}$	$mRNA$
$\mathcal{X}_4 = \mathrm{DNA}$	Unbound DNA
$\mathcal{X}_5 = \mathrm{DNA} \cdot \mathrm{D}$	DNA bound at one site
$\mathcal{X}_6 = \mathrm{DNA} \cdot 2\mathrm{D}$	DNA bound at two sites,

and is described by the reaction network

$$\begin{aligned} RNA &\xrightarrow{\kappa_1} RNA + M \\ M &\xrightarrow{\kappa_2} \emptyset \\ DNA \cdot D &\xrightarrow{\kappa_3} RNA + DNA \cdot D \\ RNA &\xrightarrow{\kappa_4} \emptyset \\ DNA + D &\underset{\kappa_6}{\overset{\kappa_5}{\rightleftharpoons}} DNA \cdot D \\ DNA{\cdot}D + D &\underset{\kappa_8}{\overset{\kappa_7}{\rightleftharpoons}} DNA{\cdot}2D \\ 2M &\underset{\kappa_{10}}{\overset{\kappa_9}{\rightleftharpoons}} D \end{aligned} \tag{5.3}$$

This model has previously been studied in [29,24]. Here, the process of translation produces protein monomers from mRNA in the first reaction, while the third reaction models the transcription of DNA. It is assumed that transcription occurs only when a transcription factor occupies one the binding sites. The last reaction models the dimerization of the protein monomer into the transcription factor, while remaining reactions incorporate the binding and unbinding of the transcription factors to the DNA and the degradation of various species.

By analogy with the previous examples, one can think of the binding and unbinding of the transcription factor as a source of switching between discrete states of the gene. This motivates the multi-scale approximation in which $\mathcal{I}^H = \{1, 2\}$ with $M_1 = M_2 = S$. Selecting an appropriate multi-scale decomposition of the reactions is much more subtle, and as noted earlier, there is some freedom in how we proceed. For the propose of illustration, suppose we select $\mathcal{J}^H = \{1, 2, 7, 8, 9, 10\}$ using the values of G_j in Table 3. This choice of scaling factors is motived by previous research on multiscale approximations of this model, see [24]. One then finds that the continuous dynamics of the multi-scale approximation are then given by

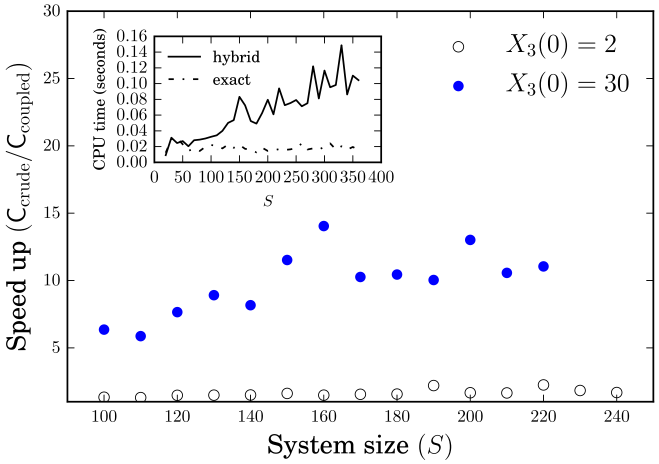

Fig. 2. The observed speedup of the coupled estimator, $\mathrm{C}_{\mathrm{crude}}/\mathrm{C}_{\mathrm{coupled}}$ measured in the total number of events needed to obtain an $\mathcal{O}(S^{-1})$ estimate of $\mathbb{E}[Z(5)]$ with initial data $Z(0) = (1, 1, X_3(0), 2, 0, 0)^T$. The faded line indicates the best fit. The implementation details are the same as in Fig. 1. The y axis begins at 1, so all of the simulations did produce speeds ups, although they are negligible for small values of $X_3(0)$. The inset shows the CPU time used to compute a representative sample path for each system size, and it can be seen that the cost of computing the hybrid paths is essentially constant. The rate constants used are from Table 3.

$$\frac{d}{dt}\bar{Z}_1(t) = 2\kappa_{10}\bar{Z}_2(t) - 2\kappa_9\bar{Z}_1(t)^2$$
$$\frac{d}{dt}\bar{Z}_2(t) = \kappa_9\bar{Z}_1(t)^2 - \kappa_{10}\bar{Z}_2(t).$$

Note that this system is decoupled from the stochastic dynamics of the low copy species, and hence the term $\mathbb{E}[\bar{Z}^H(T)]$ can be obtained via deterministic integration methods.

We have found that this particular limiting model works well with our method. Numerical results are presented in Fig. 2, where we see that computational improvements are clearly made. Fig. 2 also illustrates the role of the reaction channels involving RNA in the coupling. Recall that the coupled channels in Algorithm 3 are those for which $j \in \mathcal{J}^L$. In the present context, the channels involving RNA (channels 3 and 4) are the coupled channels, so intuitively one would expect that occurrences of theses reaction help in reducing the L^2 error. This is observed in our numerical results, where it can be seen that increasing the initial RNA significantly improves the effectiveness of the coupled estimator. When $X_3(0) = 2$ the production of RNA is very rare, and hence channels 3 and 4 rarely fire, making the coupling significantly less effective. In light of this observation, we expect that the optimal choice of scaling parameters depends not only on the model, but the specific problem, including the initial data. For example, is there another choice of scaling parameters for which the estimator performs better when there is no RNA initially in the system? We hope this question will be addressed in a future study.

6. Conclusions

Variance reduction in Monte Carlo estimators through probabilistic coupling has been used extensively in the scientific computing literature [30,31,16,17]. However, there has been little work exploring the application of simplified models to reduce variances in Monte Carlo estimators for complex chemical reaction networks. We have extended the idea of variance reduction to models with partial thermodynamic limits in which the qualitative behavior of the full stochastic model is well approximated by a PDMP. Such population models arise in the biological and chemical sciences whenever the population can be decomposed into a group of abundant species, and a group of rare species. The rare species often act as an environment that controls the dynamics of a large population, such as how the discrete state of a gene controls the production of a protein. Building on previous variance reduction techniques, we have constructed a coupling between the PDMP the thee exact process which significantly reduced the variance of their squared difference, and applied this to a Monte Carlo estimator. While bounds on the asymptotic complexity exist for simple scalings, we have shown how our method can be applied to more arbitrary scalings. In the future we hope to develop more systematic methods for determining the scaling that minimizes the complexity of our estimator.

Our results suggest that approximate stochastic models, such as the ones studied rigorously in [5,3,6] may be useful in the context of variance reduction for exact models. It would be particularly fruitful to extend our work to develop computational tools that are specifically tailored to spatial process. In particular, the reaction diffusion master equation (RDME) is a continuous time Markov chain approximation of reaction diffusion processes for which there is a great deal of interest in simulating efficiently [32,33]. Other future directions include extending the coupling to other model reductions, such as the quasi-steady state, an idea that was briefly explored in [16].

Acknowledgements

This work was supported by the National Science Foundation (DMS 1613048 and RTG 1148230).

References

[1] E. Levien, P.C. Bressloff, Coupling sample paths to the partial thermodynamic limit in stochastic chemical reaction networks, arXiv:1609.02502, http://arxiv.org/abs/1609.02502.

[2] P.C. Bressloff, Stochastic Processes in Cell Biology, Springer, 2014.

[3] A. Crudu, A. Debussche, O. Radulescu, Hybrid stochastic simplifications for multiscale gene networks, BMC Syst. Biol. 3 (1) (2009) 1.

[4] W. E, D. Liu, E. Vanden-Eijnden, Nested stochastic simulation algorithms for chemical kinetic systems with multiple time scales, J. Comput. Phys. 221 (1) (2007) 158–180, http://dx.doi.org/10.1016/j.jcp.2006.06.019.

[5] T. Jahnke, M. Kreim, Error bound for piecewise deterministic processes modeling stochastic reaction systems, Multiscale Model. Simul. 10 (4) (2012) 1119–1147.

[6] A. Chevallier, S. Engblom, Pathwise error bounds in multiscale variable splitting methods for spatial stochastic kinetics, arXiv:1607.00805, 2016.

[7] T.G. Kurtz, The relationship between stochastic and deterministic models for chemical reactions, J. Chem. Phys. 57 (7) (1972) 2976–2978.

[8] D.F. Anderson, T.G. Kurtz, Stochastic Analysis of Biochemical Systems, Stochastics in Biological Systems, vol. 1, Springer International Publishing, 2015, p. 90, http://www.springer.com/us/book/9783319168944.

[9] D.F. Anderson, D.J. Higham, Y. Sun, Computational complexity analysis for Monte Carlo approximations of classically scaled population processes, arXiv preprint, arXiv:1512.01588, https://arxiv.org/pdf/1512.01588.

[10] D.T. Gillespie, Exact stochastic simulation of coupled chemical reactions, J. Phys. Chem. 81 (1977) 2340–2361.

[11] D.T. Gillespie, Stochastic simulation of chemical kinetics, Annu. Rev. Phys. Chem. 58 (2007) 33–55.

[12] S. Zeiser, U. Franz, O. Wittich, V. Liebscher, Simulation of genetic networks modelled by piecewise deterministic Markov processes, IET Syst. Biol. 2 (2008) 113–135.

[13] Y. Cao, D.T. Gillespie, L.R. Petzold, Efficient step size selection for the tau-leaping simulation method, J. Chem. Phys. 124 (2006) 044109.

[14] D.F. Anderson, A. Ganguly, T.G. Kurtz, Error analysis of tau-leap simulation methods, Ann. Appl. Probab. (2011) 2226–2262.

[15] D.F. Anderson, D.J. Higham, Y. Sun, Complexity of multilevel Monte Carlo tau-leaping, SIAM J. Numer. Anal. 52 (6) (2014) 3106–3127.

[16] D.F. Anderson, D.J. Higham, Multilevel Monte Carlo for continuous time Markov chains, with applications in biochemical kinetics, Multiscale Model. Simul. 10 (1) (2012) 146–179.

[17] J.B. Goodman, K.K. Lin, Coupling control variates for Markov chain Monte Carlo, J. Comput. Phys. 228 (19) (2009) 7127–7136.

[18] M.B. Giles, Multilevel Monte Carlo methods, Acta Numer. 24 (2015) 259–328.

[19] A. Ganguly, D. Altintan, H. Koeppl, Jump-diffusion approximation of stochastic reaction dynamics: error bounds and algorithms, arXiv, 2014, 32 pp., http://arxiv.org/abs/1409.4303.

[20] N.G. Van Kampen, Stochastic Processes in Physics and Chemistry, vol. 1, Elsevier, 1992.

[21] S.N. Ethier, T.G. Kurtz, Markov Processes: Characterization and Convergence, Wiley, 1986.

[22] P. Glasserman, Monte Carlo Methods in Financial Engineering, vol. 53, Springer Science & Business Media, 2003.

[23] A. Duncan, R. Erban, K. Zygalakis, Hybrid framework for the simulation of stochastic chemical kinetics, J. Comput. Phys. 326 (1) (2016) 398–419.

[24] H.-w. Kang, T.G. Kurtz, L. Drive, Separation of time-scales and model reduction for stochastic reaction networks?, Ann. Appl. Probab. 23 (2003) (2010) 1–49, arXiv:1011.1672v1 [math.PR], 7 Nov 2010.

[25] R. Veltz, A new twist for the simulation of hybrid systems using the true jump method, arXiv [math], http://arxiv.org/abs/1504.06873.

[26] R. Karmakar, I. Bose, Graded and binary responses in stochastic gene expression, Phys. Biol. 1 (2004) 197–204.

[27] P. Thomas, N. Popović, R. Grima, Phenotypic switching in gene regulatory networks, Proc. Natl. Acad. Sci. USA 111 (19) (2014) 6994–6999, http://dx.doi.org/10.1073/pnas.1400049111, http://www.pnas.org/content/111/19/6994, http://www.ncbi.nlm.nih.gov/pubmed/24782538.

[28] D.L. Cook, A.N. Gerber, S.J. Tapscott, Modeling stochastic gene expression: implications for haploinsufficiency, Proc. Natl. Acad. Sci. USA 95 (26) (1998) 15641–15646, http://dx.doi.org/10.1073/PNAS.95.26.15641, http://www.ncbi.nlm.nih.gov/pubmed/9861023, http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC28097.

[29] J. Goutsias, Quasiequilibrium approximation of fast reaction kinetics in stochastic biochemical systems, J. Chem. Phys. 122 (18) (2005) 184102, http://dx.doi.org/10.1063/1.1889434, http://aip.scitation.org/doi/10.1063/1.1889434.

[30] D.F. Anderson, B. Ermentrout, P.J. Thomas, Stochastic representations of ion channel kinetics and exact stochastic simulation of neuronal dynamics, J. Comput. Neurosci. 38 (1) (2015) 67–82.

[31] D.F. Anderson, M. Koyama, An asymptotic relationship between coupling methods for stochastically modeled population processes, IMA J. Numer. Anal. 35 (4) (2015) 1757–1778.

[32] S. Isaacson, The reaction–diffusion master equation as an asymptotic approximation of diffusion to a small target, SIAM J. Appl. Math. 70 (1) (2009) 77–111, http://dx.doi.org/10.1137/070705039, http://epubs.siam.org/doi/abs/10.1137/070705039.

[33] S.A. Isaacson, D. Isaacson, Reaction-diffusion master equation, diffusion-limited reactions, and singular potentials, Phys. Rev. E, Stat. Nonlinear Soft Matter Phys. 80 (6) (2009).