

Efficient computation of parameter sensitivities of discrete stochastic chemical reaction networks

Muruhan Rathinam, Patrick W. Sheppard, and Mustafa Khammash

Citation: *J. Chem. Phys.* **132**, 034103 (2010); doi: 10.1063/1.3280166

View online: <http://dx.doi.org/10.1063/1.3280166>

View Table of Contents: <http://jcp.aip.org/resource/1/JCPSA6/v132/i3>

Published by the [AIP Publishing LLC](#).

Additional information on J. Chem. Phys.

Journal Homepage: <http://jcp.aip.org/>

Journal Information: http://jcp.aip.org/about/about_the_journal

Top downloads: http://jcp.aip.org/features/most_downloaded

Information for Authors: <http://jcp.aip.org/authors>

ADVERTISEMENT



**Run your GPU
code 2X faster.
Try a Tesla K20 GPU
accelerator today.
Free.**

Efficient computation of parameter sensitivities of discrete stochastic chemical reaction networks

Muruhan Rathinam,^{1,a)} Patrick W. Sheppard,^{2,b)} and Mustafa Khammash^{2,c)}

¹*Department of Mathematics and Statistics, University of Maryland, Baltimore County 1000 Hilltop Circle, Baltimore, Maryland 21250, USA*

²*Department of Mechanical Engineering, University of California, Santa Barbara Engineering II Bldg., Santa Barbara, California 93106-5070, USA*

(Received 20 August 2009; accepted 9 December 2009; published online 15 January 2010)

Parametric sensitivity of biochemical networks is an indispensable tool for studying system robustness properties, estimating network parameters, and identifying targets for drug therapy. For discrete stochastic representations of biochemical networks where Monte Carlo methods are commonly used, sensitivity analysis can be particularly challenging, as accurate finite difference computations of sensitivity require a large number of simulations for both nominal and perturbed values of the parameters. In this paper we introduce the common random number (CRN) method in conjunction with Gillespie's stochastic simulation algorithm, which exploits positive correlations obtained by using CRNs for nominal and perturbed parameters. We also propose a new method called the common reaction path (CRP) method, which uses CRNs together with the random time change representation of discrete state Markov processes due to Kurtz to estimate the sensitivity via a finite difference approximation applied to coupled reaction paths that emerge naturally in this representation. While both methods reduce the variance of the estimator significantly compared to independent random number finite difference implementations, numerical evidence suggests that the CRP method achieves a greater variance reduction. We also provide some theoretical basis for the superior performance of CRP. The improved accuracy of these methods allows for much more efficient sensitivity estimation. In two example systems reported in this work, speedup factors greater than 300 and 10 000 are demonstrated. © 2010 American Institute of Physics.

[doi:[10.1063/1.3280166](https://doi.org/10.1063/1.3280166)]

I. INTRODUCTION

Stochastic models of chemical reaction networks have gained significant attention recently due in large part to the increasing appreciation of the important role that stochastic effects and intrinsic noise have on biological networks.^{1,2} These networks involve some molecular species, which are present only in small copy numbers so that the discrete and stochastic nature of the system cannot be neglected. In fact in the presence of nonlinearities, the continuous deterministic model described by reaction rate equations may not even capture the average behavior of these systems correctly.

Chemical reaction models typically depend on a set of kinetic parameters whose values are often unknown or fluctuate due to an uncertain environment, such as is the case for gene regulatory networks. Even small changes to the parameters may significantly alter the system output, and thus it is critical to characterize such effects. Parametric sensitivity analysis studies the change in system outputs to variations in kinetic parameters and is an indispensable analysis technique in the study of kinetic models. It is instrumental in deducing system properties, such as robustness in an uncertain environment. In large networks, sensitivity analysis can pinpoint

critical or rate limiting pathways and aid in reduced order modeling, or in the biological context, guide drug targeting.

Sensitivity analysis may focus on the effects of finite or infinitesimal perturbations of certain parameters. In deterministic chemical kinetics the infinitesimal sensitivities are computed easily via the integration of the linearization of the reaction rate equations. If one is interested in effects of larger perturbations for which linear approximation is not adequate, one typically recomputes the solution to the reaction rate equations for the perturbed parameter values. In the stochastic setting the simplest and most common method for finite perturbations is via Monte Carlo simulations to compute a finite difference. The approach here is to characterize the sensitivity to a finite perturbation h of a parameter c about a nominal value $c=c_0$ via a finite difference, such as $(E[f(X(T, c_0+h))] - E[f(X(T, c_0))])/h$, of the expected values. Here f is a function of interest of the final state $X(T)$ of the chemical system. One uses Monte Carlo simulations to estimate the expected values via sample means. The simplest approach would be to use two independent streams of random numbers to generate samples of $X(T, c_0)$ and $X(T, c_0+h)$, the so called independent random number (IRN) approach. A recent study used this approach in combination with the Fisher information matrix to generate several different sensitivity measures.³ However the use of IRNs may result in a statistical estimator with large variance, thereby increasing the computational effort as large samples may be

^{a)}Electronic mail: muruhan@umbc.edu.

^{b)}Electronic mail: sheppard@engineering.ucsb.edu.

^{c)}Electronic mail: khammash@engineering.ucsb.edu.

required. In this paper we show how the approach of using the same stream of common random numbers (CRNs) to generate samples of $X(T, c_0)$ and $X(T, c_0 + h)$ can typically result in an estimator with low variance and thus requires far fewer samples and hence yields increased computational efficiency. We also propose a new method of computing sensitivities called the *common reaction path* (CRP) method that uses CRNs in the particular setting of *random time change* (RTC) representation of Markov processes.⁴ We show that both the CRN and CRP methods can dramatically reduce computation time over IRN, with CRP performing better than CRN. In addition to numerical evidence, we also provide a theoretical explanation (but not a rigorous proof) as to why CRP is expected to perform better.

The finite difference $(E[f(X(T, c_0 + h))] - E[f(X(T, c_0))])/h$ characterizes sensitivity to the perturbation h . When h is sufficiently small, one may treat the finite difference estimate as an approximation to the infinitesimal sensitivity given by the partial derivative $(\partial/\partial h)E[f(X(T, c_0 + h))]$ at $h=0$. Instead of using the finite difference, one may directly compute the infinitesimal sensitivity. One such approach is based on the Girsanov measure transformation,⁵ and another employs polynomial chaos methods.⁶

The contents of the paper are organized as follows. Section II provides a brief introduction to stochastic chemical kinetics. The RTC representation of discrete stochastic chemical systems is presented in Sec. III, which is used to formulate a variant of the stochastic simulation algorithm (SSA) that handles computations corresponding to individual reaction paths in Sec. III B. Section IV discusses Monte Carlo finite difference schemes using IRNs and CRNs in the context of accuracy of estimates. Section IV B introduces the CRP algorithm, which exploits the RTC representation in conjunction with CRNs to achieve higher efficiencies. All three methods are compared via numerical examples in Sec. V.

II. STOCHASTIC CHEMICAL KINETICS

Let us consider a chemical reaction system consisting of n chemical species whose evolution in continuous time is discrete and stochastic. Under the well stirred assumption⁷ the random state of the system at time t is characterized by the n dimensional vector $X(t, \omega)$ whose i th entry $X_i(t, \omega)$ is a non-negative integer corresponding to the number of molecules of the i th species at time t . Here ω captures the randomness and refers to an element of the sample space Ω consisting of sample trajectories. In what follows we sometimes drop ω for simplicity of presentation. The system consists of M reaction channels whose firings transition the system from one state to another, changing the population by a discrete amount given by the *stoichiometric vectors* ν_j , $j=1, \dots, M$. Each reaction channel has associated with it a *propensity function* $a_j(X(t), c)$, $j=1, \dots, M$, which is typically a function of the system state and one or more kinetic parameters contained in the vector c . The propensity function a_j is defined by the prescription that conditioned on being in state $X(t)$ at time t , $a_j(X(t), c)\delta t$ gives the probability for the j th reaction to occur in the infinitesimally small time interval

$(t, t + \delta t]$. The functional form for the propensity functions is usually combinatorial (and hence polynomial) in nature and is obtained from physical reasoning.⁷ Alternative forms using rational functions are also found in literature.⁸

It follows that this system is a continuous time Markov process whose probability mass function evolves in time according to the chemical master equation (CME).⁷ The analytical solution of the CME is usually intractable, but generating exact sample trajectories of the system is easy using a SSA such as the direct and first reaction methods presented by Gillespie⁷ or the next reaction method presented by Gibson and Bruck.⁹

III. RANDOM TIME CHANGE REPRESENTATION

In this section we describe the RTC description of a stochastic chemical process. In general such a description is possible for any Markov process whose state space is an integer lattice.⁴ We provide a physical interpretation that will help visualize the RTC representation from Ref. 4 in the context of chemical reactions.

In the RTC description, one may envisage each reaction channel to be carrying its own internal clock, which runs at a rate that equals its propensity function. In other words the *internal times* $S_j(t, \omega)$ for the reaction channels $j=1, \dots, M$ are defined by

$$S_j(t, \omega) = \int_0^t a_j(X(s, \omega)) ds, \quad j = 1, \dots, M. \quad (1)$$

It should be noted that although we refer to $S_j(t, \omega)$ as the internal times in order to aid the interpretation, the S_j are in fact dimensionless quantities. The key point of Ref. 4 is that *viewed from their respective internal times the reaction channels will fire as though they are independent unit rate Poissons*. This is made mathematically precise by the following equation:

$$X(t, \omega) = X(0, \omega) + \sum_{j=1}^M \nu_j Y_j(S_j(t, \omega), \omega), \quad (2)$$

where Y_1, \dots, Y_M are independent unit rate Poisson processes corresponding to the reaction channels. The advantage of this representation is that the “driving noise” processes are described independently of the state and the particular characteristics of the system, namely, ν_j and a_j . The equation above holds pathwise, i.e., for each realization ω . For a given realization $Y_j(\cdot, \omega)$, $j=1, \dots, M$ of the driving noise processes the computation of the state process X is a deterministic procedure specified according to the above equation. Thus the total number of times the reaction channel j fires between times 0 and t is given by evaluating the unit rate Poisson process Y_j at the random internal time $S_j(t)$. We shall describe this computation in more detail later.

Since we are interested in the sensitivity with respect to parameters we include the dependence on parameters c and write the equation as follows:

$$X(t, \omega, c) = X(0, \omega, c) + \sum_{j=1}^M \nu_j Y_j(S_j(t, \omega, c), \omega), \quad (3)$$

where

$$S_j(t, \omega, c) = \int_0^t a_j(X(s, \omega, c), c) ds, \quad j = 1, \dots, M. \quad (4)$$

Note that the ν_j are independent of parameters while the propensities a_j are dependent on the parameters: $a_j = a_j(x, c)$. It is also important to note that the noise processes Y_j are Poisson with unit rate in their respective internal time frames, and do not depend on c explicitly. We shall mainly be interested in situations where the initial condition is independent of c and deterministic: $X(0, \omega, c) = x_0$. Equation (3) enables us to couple two processes $X(\cdot, \cdot, c_1)$ and $X(\cdot, \cdot, c_2)$ corresponding to different parameter values. In other words they are represented as functions of the same sample space and this allows for direct comparison. For instance the question “how does a given sample trajectory vary when the parameter is perturbed?” is mathematically well posed.

A. Pathwise computations based on random time change representation

Given a realization $Y_1(\cdot, \omega), \dots, Y_M(\cdot, \omega)$ of the noise we wish to solve for $X(\cdot, \omega, c)$ from Eq. (3). This may be done as follows. Let us denote the random internal jump times of the Poisson process Y_j by I_i^j , where $j = 1, \dots, M$ and $i = 1, 2, \dots$. Thus

$$I_1^j < I_2^j < I_3^j \dots$$

for each j . For a value t of physical time if $S_j(t, \omega, c) = I_i^j(\omega)$, then the i th firing of the j th reaction channel will occur at that time t . Let us denote this physical time at which the i th firing of the j th reaction channel occurs by $T_i^j(\omega, c)$. Thus by definition $S_j(T_i^j(\omega, c), \omega, c) = I_i^j(\omega)$. Let us also introduce $T_i(\omega, c)$ and $J_i(\omega, c)$ for $i = 1, 2, \dots$ as the random times at which the i th reaction event of any type occurs and the random type of this reaction channel, respectively. Thus J_i is a number in $\{1, 2, \dots, M\}$. It is clear that the collection (T_i, J_i) for $i = 1, 2, \dots$ carries the same information as the collection T_i^j for $i = 1, 2, \dots$ and $j = 1, \dots, M$. It is also clear that either one of the above will uniquely determine the trajectory $X(\cdot, \omega, c)$. Thus we focus on the computation of T_i and J_i .

An important point to note is that $S_j(t, \omega, c)$ is piecewise linear in t . In fact for $T_i \leq t < T_{i+1}$,

$$S_j(t, \omega, c) = S_j(T_i, \omega, c) + a_j(X(T_i, \omega, c), c)(t - T_i),$$

$$j = 1, \dots, M.$$

This allows for easy computation.

To further facilitate the computation we define $I_+^j(t, \omega, c)$ for $j = 1, \dots, M$ by

$$I_+^j(t, \omega, c) = \min\{I_l^j(\omega) | S_j(t, \omega, c) < I_l^j, l = 0, 1, 2, \dots\},$$

$$j = 1, \dots, M.$$

In words, $I_+^j(t)$ is the internal time of the next firing of reaction channel j at physical time t . It is convenient to keep track of I_+^j .

Assuming T_1, \dots, T_i and J_1, \dots, J_i are known for some i we compute T_{i+1} and J_{i+1} as follows. First note that knowing this information we also know $I_+^j(T_i)$ for $j = 1, \dots, M$ and $X(T_i)$. It follows that

$$T_{i+1} = T_i + \min \left\{ \frac{I_+^j(T_i) - S_j(T_i)}{a_j(X(T_i))} \mid j = 1, \dots, M \right\}. \quad (5)$$

To see this, first observe that when the physical time equals T_i , the internal times of the processes are given by $S_j(T_i)$. Second during $t \in [T_i, T_{i+1})$ the internal times $S_j(t)$ are increasing at the constant respective rates $a_j(X(T_i))$. Third the next internal times of firing of the reactions are given by $I_+^j(T_i)$. Thus the elapsed physical time $T_{i+1} - T_i$ before the next firing of a reaction is the minimum of $(I_+^j(T_i) - S_j(T_i)) / (a_j(X(T_i)))$ taken over j . Furthermore J_{i+1} is the index of the minimum. Thus $T_{i+1}(\omega, c) = I_{i+1}^{J_{i+1}}(\omega, c)$. It must be remarked that the minimum is unique except for a set of ω with probability zero. This is because given T_i , the $I_+^j(T_i)$ are continuously jointly distributed.

The above reasoning also gives us the first jump time T_1 . It is given by

$$T_1(\omega, c) = \min\{I_1^j / a_j(x_0, c) \mid j = 1, \dots, M\} \quad (6)$$

and $J_1(\omega, c)$ is the index of the minimum. Thus $T_1(\omega, c) = I_1^{J_1}(\omega, c)$. The pathwise computations described above are illustrated visually for a simple example in Fig. 1.

B. The random time change algorithm for simulation of stochastic chemical systems

The above methods of pathwise computation allow us to write an alternative algorithm for the exact simulation of stochastic chemical systems, which we term as the RTC simulation algorithm. The algorithm uses variables S_j , I_+^j and indices k_j for $j = 1, \dots, M$. Also we assume that M streams of unit exponential random numbers E_i^j (for $j = 1, \dots, M$ and $i = 1, 2, \dots$) are available. These random numbers represent the (internal) times between successive firings of the unit rate Poisson processes Y_j and are related to the internal firing times I_i^j by $I_{i+1}^j - I_i^j = E_i^j$. Note that if we can generate M independent streams of uniform random numbers in $[0, 1)$, then we can easily convert these to M independent streams of unit rate exponentials.

In what follows, k_j is the index into the j th stream of exponential numbers, S_j is the current internal time of j th reaction channel, and I_+^j is the internal time at which the next firing of reaction channel j occurs. Note that explicit dependence on parameters c is omitted to simplify notation.

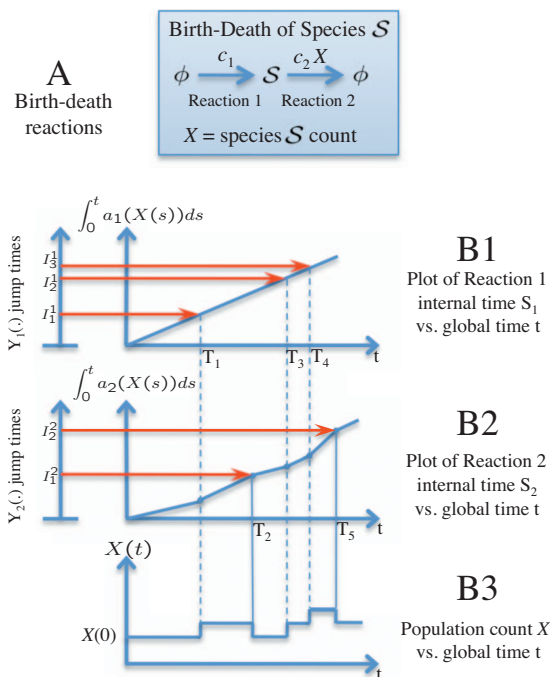


FIG. 1. RTC representation of a birth-death example. Panel A shows the two reactions describing the birth-death of species S . The propensity for reaction 1 is $a_1(X)=c_1$, which is independent of the population count X . Panels B1–B3 describe the RTC representation for the birth-death example. For this example, three time clocks can be envisioned. One maintains global time for an observer watching the reaction system. Two other clocks maintain internal times S_1 and S_2 for reactions 1 and 2, respectively. The rate of each internal clock is given by the propensity of the corresponding reaction: $a_i(X(s))$, $i=1,2$. The integral of each internal clock rate gives that respective clock's time S_i (plotted in panels B1 and B2 against global time, t). Randomness is generated through two independent unit rate Poisson processes, Y_1 and Y_2 , one for each reaction. The jump times for Y_1 and Y_2 can be generated ahead of time, and these in turn determine the path of the entire process as follows: At time $t=0$ the population count starts at $X(0)$ and remains there until the firing of the next reaction at time T_1 (B3). As global time flows past $t=0$, the internal times also flow each according to their respective rate: $a_i(X)$. In between reactions, $a_i(X)$ will be constant and hence, its integral, the clock time S_i , will be piecewise linear. A reaction fires when its internal time coincides with the jump time for its corresponding Poisson process. These are shown by the red, horizontal arrows emanating from the jump times J_i^j . In this example, this happens first for reaction 1 at global time T_1 , when $\int_0^{T_1} a_1(X(s))ds = J_1^1$. When a reaction fires, the population X changes according to the reaction stoichiometry. This change in X will subsequently affect the rates of reaction clocks whose propensity depends on X (reaction 2 in this example). The process proceeds until a final time is reached. The advantage of this random time representation is that the driving randomness is decoupled from the state.

RTC simulation algorithm.

- (1) **Initialization** Set $t=0$, $T_0=0$, $X(T_0)=x_0$; $S_j=0$, $k_j=1$ and $I_+^j=E_+^j$ for $j=1, \dots, M$

General step $T_i, X(T_i), k_j, I_+^j$ and S_j are known.

- (2) Exit if terminal condition is reached, otherwise continue.
 (3) Calculate propensity functions $a_j(X(T_i))$ for $j=1, \dots, M$.
 (4) Compute $T_{i+1}: T_{i+1} = T_i + \min \left\{ \frac{I_+^j - S_j}{a_j(X(T_i))} \mid j=1, \dots, M \right\}$.
 Let j^* be the index of the minimum in above equation.
 (5) Set $X(T_{i+1}) = X(T_i) + \nu_{j^*}$.
 (6) For $j=1, \dots, M$ set $S_j \leftarrow S_j + a_j(X(T_i))(T_{i+1} - T_i)$.
 (7) Increment k_{j^*} .
 (8) Set $I_+^{j^*} \leftarrow I_+^{j^*} + E_{k_{j^*}}^{j^*}$.
 (9) Increment i and return to step 1.

As there are several widely used exact SSAs available, it is worthwhile discussing here where the RTC simulation al-

gorithm fits into the present landscape. Gibson and Bruck's next reaction method,⁹ Anderson's modified next reaction method,¹⁰ and our proposed RTC method are all stochastically equivalent to Gillespie's direct method as well as first reaction method⁷ in that they are all exact simulation methods. The main differences are algorithmic and are outlined next.

Gillespie's direct method differs from the rest of the methods mentioned in that it explicitly generates a uniform random number to determine which reaction channel fired, whereas all the other methods make this determination by taking the minimum of M different firing times. Gibson and Bruck's next reaction method resembles Gillespie's first reaction method in that it keeps in memory M IRNs at every step, one per each reaction channel. However, following the initialization step in which M random numbers are drawn to set the firing times of each reaction, the next reaction method only generates *one random number* at each iteration because the next firing times for all reactions besides the one that fired are reused. This feature is shared with the modified next reaction and RTC methods. Gillespie's first reaction method differs from these three methods in that it discards the unused $M-1$ random numbers from the previous step and generates another M random numbers.

The difference between Anderson's modified next reaction method and Gibson and Bruck's next reaction method is algorithmically insignificant but conceptually significant. The first works with internal times (which naturally arise in the RTC representation) while the latter with physical time. The advantage of this conceptual framework also allowed Anderson¹⁰ to efficiently simulate systems with time dependent propensities.

While the RTC is most similar to the modified reaction method, it differs from it in a subtle but important way: RTC draws from M independent, parallel streams of random numbers E_+^1, E_+^2, \dots (for $j=1, \dots, M$) corresponding to each reaction channel rather than drawing them from a single stream. This tweak to the algorithm makes a difference when doing sensitivity analysis. In particular, it ensures that keeping the same M parallel streams of random numbers is equivalent to keeping the same paths for the processes Y_j . See Appendix A for an illustration.

The next reaction method as it originally appeared in Ref. 9 makes use of dependency graphs and priority queues. These increase efficiency compared to the direct method by updating only those propensities that change following a particular reaction firing and by using specific data structures to make update operations faster. While the modified next reaction method and our proposed RTC method do not employ these tools, they could easily be incorporated to improve efficiency.

IV. MONTE CARLO BASED SENSITIVITY ANALYSIS OF STOCHASTIC CHEMICAL NETWORKS

Now we consider the problem of using Monte Carlo simulation to estimate the stochastic analog to the sensitivity coefficient of deterministic dynamical systems.¹¹ Typically one is interested in the sensitivity of the expected value of a function f of the state $X(T)$ at some final time T ,

$E[f(X(T, c_0))]$, to perturbations in a nominal parameter c_0 . We may characterize the sensitivity by the finite difference

$$\frac{E[f(X(T, c_0 + h, \omega))] - E[f(X(T, c_0, \omega))]}{h},$$

where h is a variation in the parameter c_0 . As h approaches 0 this limits to the partial derivative

$$\frac{\partial}{\partial h} E[f(X(T, c_0 + h, \omega))]$$

evaluated at $h=0$. It is useful to define the random variable Z whose expected value is what we seek to compute,

$$Z = \frac{f(X(T, c_0 + h, \omega)) - f(X(T, c_0, \omega))}{h}. \quad (7)$$

Since X is a stochastic process whose distribution cannot typically be obtained analytically, one can compute a sample of independent realizations of Z and estimate $E(Z)$. Any exact SSA—the first reaction and direct methods,⁷ the next reaction method,⁹ the modified next reaction method,¹⁰ or the RTC simulation algorithm of Sec. III B—is suitable to generate independent samples of $f(X(T, c_0))$ and of $f(X(T, c_0 + h))$ to estimate $E(Z)$. At first it would seem natural to generate the sample of $f(X(T, c_0 + h))$ to be independent of the sample of $f(X(T, c_0))$. This approach is referred to as the IRN method. Here we describe the overall structure of such an algorithm.

Algorithm for sensitivity estimation using IRN.

- (1) For $i=1$ to N_{tr} , where N_{tr} is the number of trajectories to be simulated:
- (2) Initialize random number generator with random seed.
- (3) Let $c=c_0$ and run any version of the SSA algorithm to compute: $X(T, c_0)$. All random numbers needed are generated by successive calls to a random number generator.
- (4) Let $c=c_0+h$ and use same version of SSA algorithm to compute $X(T, c_0+h)$. All random numbers needed are generated independent of the numbers used in step 2 by successive calls to a random number generator. This can be accomplished by continuing to draw from the same stream without resetting the state, or by reinitializing the random number generator with a different random seed.
- (5) Compute sensitivity for the i th trajectory:
 $Z_i = f(X(T, c_0 + h)) - f(X(T, c_0)) / h$.
- (6) End For loop in i .
- (7) Compute sample mean and sample standard deviation of $\{Z_i | i=1, \dots, N_{tr}\}$.

Although this procedure is straightforward and easy to implement, one must simulate the system a great many times N_{tr} in order to generate accurate estimates using IRN. To see this, let us briefly recall the problem of estimating $E(Z)$ from an independent sample of random variables $Z_1, \dots, Z_{N_{tr}}$, all having the same distribution as Z .

The standard estimator of $E(Z)$ is the sample mean $\bar{Z} = (Z_1 + \dots + Z_{N_{tr}}) / N_{tr}$. The accuracy of the estimator \bar{Z} may be measured in terms of its standard deviation. The standard deviation of \bar{Z} is equal to

$$\sigma_{\bar{Z}} = \sigma_Z / \sqrt{N_{tr}},$$

where σ_Z is the standard deviation of Z . The relative standard error (RSE) in the estimation [assuming $E(Z) \neq 0$] is

$$\text{RSE} = \frac{\sigma_Z}{\sqrt{N_{tr}} |E(Z)|}.$$

In the sensitivity estimation problem at hand, Z is the sensitivity from Eq. (7). For notational simplicity set $Y_1 = f(X(T, c_0))$ and $Y_2 = f(X(T, c_0 + h))$. Hence $Z = (Y_2 - Y_1) / h$. Thus its standard deviation is

$$\sigma_Z = \sqrt{\text{var}(Y_1) + \text{var}(Y_2) - 2\text{cov}(Y_1, Y_2)} / |h| \quad (8)$$

and the RSE is

$$\text{RSE} = \frac{\sqrt{\text{var}(Y_1) + \text{var}(Y_2) - 2\text{cov}(Y_1, Y_2)}}{\sqrt{N_{tr}} |E(Y_2) - E(Y_1)|}. \quad (9)$$

Thus the RSE in general depends on $\text{cov}(Y_1, Y_2)$, h , and N_{tr} . In the IRN method, $X(T, c_0)$ and $X(T, c_0 + h)$ are independent random variables and hence $\text{cov}(Y_1, Y_2) = 0$.

Increasing h reduces RSE; however if one aims to estimate the infinitesimal sensitivity via the finite difference, then a large h leads to a bias. Thus if the goal is to estimate the infinitesimal sensitivity, the proper selection of h in the finite difference scheme equation (7) involves a tradeoff between its variance and its bias: h must be small to reduce the bias, yet large enough to keep the estimator variance in check. An appropriately chosen sequence of diminishing h values can yield optimal convergence rates (to the infinitesimal sensitivity).^{12,13} However this investigation is not the focus of this paper.

Given that h is determined by other considerations, the only option left to reduce the RSE using the IRN algorithm is to increase N_{tr} , i.e., to generate more independent samples of $f(X(T, c_0))$ and $f(X(T, c_0 + h))$. In Sec. IV A and IV B, we provide ways to decrease RSE by increasing $\text{cov}(Y_1, Y_2)$.

It is worth noting here that other finite difference schemes, such as the central difference approximation, could be used in place of Eq. (7). The approach presented below is also applicable to these estimators, but we only consider the scheme in Eq. (7) throughout for the sake of simplicity.

A. Using common random numbers to reduce the sensitivity estimator variance

The easiest and most common method to achieve variance reduction is to introduce dependence among the random variables being estimated by using CRNs in simulations.¹⁴

The concept behind CRN is simple. By using CRNs for the simulation of both $X(T, c_0)$ and $X(T, c_0 + h)$, one introduces nonzero covariance between the two processes. Positive covariance is not guaranteed in general, but when h approaches zero, under some mild assumptions, $\text{var}(Y_1 - Y_2)$ approaches zero as well (see Appendix C), which is equivalent to $\text{cov}(Y_1, Y_2)$ approaching $\text{var}(Y_1)$.

Here we present a straightforward implementation of CRN as applied to a SSA such as Gillespie's direct method.⁷ When one computes the sensitivity with the direct method SSA, implementation of the method of CRN is achieved by

using the same stream of uniform random numbers that generate r_1 and r_2 at each step, i.e., the random variables used to determine the time increment until the next reaction firing and the random type of reaction channel which fires, respectively. The practical implementation of CRN can be achieved multiple ways; here we reseed the random number generator prior to computing $X(T, c_0 + h)$ with the same initial seed used to simulate $X(T, c_0)$ at each step.

To present the algorithm for sensitivity estimation employing CRN more generally, we define the functions $\text{rand}()$ to return a uniform random number in the interval $[0,1)$ and $\text{seed}(w)$ to initialize $\text{rand}()$ with seed w . The function $\text{time}(\text{NULL})$ returns the current system time for the purpose of generating a random seed.

Algorithm for sensitivity estimation using CRN.

- (1) For $i=1$ to N_{tr}
- (2) Generate random seed: $w=\text{time}(\text{NULL})$.
- (3) Seed the random number generator: $\text{seed}(w)$.
- (4) Let $c=c_0$ and run any version of the SSA algorithm to compute $X(T, c_0)$. Here all random numbers needed are generated by successive calls to $\text{rand}()$.
- (5) Reseed random number generator using same w as before: $\text{seed}(w)$.
- (6) Let $c=c_0+h$ and run the same version of the SSA algorithm to compute $X(T, c+h)$. All random numbers needed are again generated by successive calls to $\text{rand}()$.
- (7) Compute sensitivity for i th trajectory:
 $Z_i = f(X(T, c_0 + h)) - f(X(T, c_0)) / h$.
- (8) End For loop in i .
- (9) Compute sample mean and sample standard deviation of $\{Z_i | i=1, \dots, N_{tr}\}$.

B. The common reaction path method for sensitivity estimation

It is apparent from the above discussion that the sensitivity estimates obtained with any of the popular SSA variants (the first reaction and direct methods,⁷ the next reaction method,⁹ and the modified next reaction method¹⁰) will be improved by using CRN rather than IRN. In this section we present a particular form of CRN method, namely, the use of CRNs in the context of the RTC simulation algorithm presented in Sec. III B, which we refer to as the CRP algorithm. The rationale for this terminology is as follows. The sequence of samples taken from the j th stream of random numbers during simulation, which determine the internal jump times of that reaction channel, can be interpreted as the *reaction path* of the j th reaction channel. Collectively, the set of M reaction paths uniquely determines the evolution of the state, as detailed previously in Sec. III A.

In the following we assume the existence of M independent streams of unit rate exponential random numbers. Suppose the j th stream is accessed by $E=\text{rande}(j)$, and j th stream is seeded by $\text{seed}(j, w)$. Let N_{tr} be the number of trajectories that will be generated. The algorithm described here randomly seeds each stream before each new trajectory is generated using the current system clock as the seed. This is to ensure independence of the streams. In practice other implementations may be used so long as care is taken to preserve independence of the streams.

The CRP method, like the CRN method, also requires a

viable way of sharing the reaction paths between simulations of the processes with parameters c_0 and c_0+h . One way to implement this is to generate a trajectory of $X(c_0)$ and to store the random numbers E_j^i in an M dimensional array, making sure that they are ordered exactly as they were generated during simulation and into the row corresponding to their reaction channel. Then a trajectory of $X(c_0+h)$ would be generated using the numbers from the array in the identical order. However from a practical standpoint, this approach has several disadvantages, including possible storage bottlenecks when used for simulations with many reaction channels and/or jump events. A more practical approach which we take here is to seed each stream and then store the seed prior to generating a trajectory for $X(c_0)$, and then reseed each stream with the identical seeds before generating $X(c_0+h)$.

CRP sensitivity analysis algorithm.

- (1) For $i=1$ to N_{tr} .
- (2) Seed the streams: For $j=1, \dots, M$, $w(j)=\text{time}(\text{NULL})$, $\text{seed}(j, w(j))$.
- (3) Let $c=c_0$ and run RTC simulation algorithm to compute $X(T, c_0)$. Here the E_1^j, E_2^j, \dots for $j=1, \dots, M$ are generated by successive calls to $\text{rande}(j)$.
- (4) Reseed the streams with same $w(j)$ as before: $\text{seed}(j, w(j))$ for $j=1, \dots, M$.
- (5) Let $c=c_0+h$ and run algorithm RTC to compute $X(T, c_0+h)$. Here the E_1^j, E_2^j, \dots for $j=1, \dots, M$ are generated by successive calls to $\text{rande}(j)$.
- (6) Compute sensitivity for i th trajectory:
 $Z_i = f(X(T, c_0 + h)) - f(X(T, c_0)) / h$.
- (7) End For loop in i .
- (8) Compute sample mean and sample standard deviation of $\{Z_i | i=1, \dots, N_{tr}\}$.

It is clear how to run this algorithm for any general number of p parameter values $c_0, c_0+h_1, \dots, c_0+h_p$ instead of $p=1$ as described above. First, it must be noted that CRP is a special case of CRN. Second it is important to point out the distinction between the CRP algorithm and the more general CRN algorithm. If CRN is used with Gillespie's direct SSA, although the same set of random numbers is used to generate $X(t, c_0)$ and $X(t, c_0+h)$, the internal jump times for each reaction channel will likely be different. When CRN is used with either the next reaction method or the modified next reaction method, at each step in the simulation the random number drawn from the single common stream may set the internal time for a different reaction than that in the unperturbed simulation (see Appendix A for an illustration). The resulting deviations in the internal jump times for each channel will likely limit the positive correlation at each replicate. In contrast, by sharing M independent streams (one per reaction channel) in the CRP algorithm, one ensures that each of the reaction paths will be identical between the perturbed and unperturbed processes. This we believe more tightly couples the processes thus, increasing the covariance compared to using CRN with the other simulation algorithms.

Our numerical examples demonstrate that CRP tends to achieve a lower variance than the other CRN estimators discussed. This is further borne out by a continuity property enjoyed by CRP estimators (and not shared by the other

CRN estimators), which more tightly couples the nominal and perturbed trajectories to achieve the observed reduction in variance. The plot of estimator variances as a function of perturbation size h for three examples shown in Fig. 5 justifies our intuition that the CRP estimator has lower variance. Furthermore, the trajectories perturb in a more continuous manner with respect to the parameters c in the CRP method than in the CRN method implemented with any existing SSA variants. While the states at any fixed time are integers and hence cannot be expected to change continuously with respect to parameters c (for any fixed sequence of random numbers), the times T_n of reaction events typically change continuously in the CRP method while only piecewise continuously in a general CRN method. In Appendix B we explore this continuity property of CRP and CRN in conjunction with Gillespie's direct method.

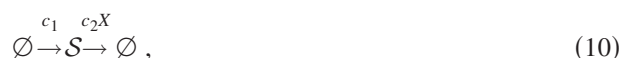
In Appendix C we show under some modest assumptions that the variance of the CRP and CRN estimators approaches 0 as the perturbation h approaches 0. In numerical examples we observe that the variances behave as $\mathcal{O}(h)$ for small h (see Fig. 5). In contrast the variance of the IRN estimator does not approach 0, but rather a nonzero value as $h \rightarrow 0$. This observation leads to the following conclusions. The ratio of the variances of the CRP and CRN estimators is constant for small h while the ratio of CRP (or CRN) and IRN estimator variances depends on h . As h gets smaller, both CRP and CRN become increasingly better than IRN and thus the advantage can be made arbitrarily large although all methods become increasingly computationally expensive. Moreover, the advantage (or disadvantage) of CRP over CRN remains constant for all sufficiently small h . Our numerical examples suggest that both CRP and CRN (in conjunction with Gillespie's direct method) perform much better than IRN for h small enough to estimate infinitesimal sensitivity. Also we observe in Sec. V that CRP performs better than CRN in each example. Although the advantage depends on the specific problem, in all examples the advantage is substantial, as even a factor as small as two may translate to a savings of hours or even days of computation time in larger problems.

V. NUMERICAL EXAMPLES

We provide three examples in this section to compare the performance of IRN, CRN with Gillespie's direct method, and CRP.

A. Birth-death process

To illustrate how the CRN and CRP methods are used in the sensitivity analysis of discrete chemical systems, we first consider the simple example of a one species birth-death process,



in which a chemical species \mathcal{S} , whose population is denoted by X , is created at constant rate c_1 and decays proportionally to its current population at rate c_2X . The solution for the evolution of the state $X(t)$ is known exactly for this

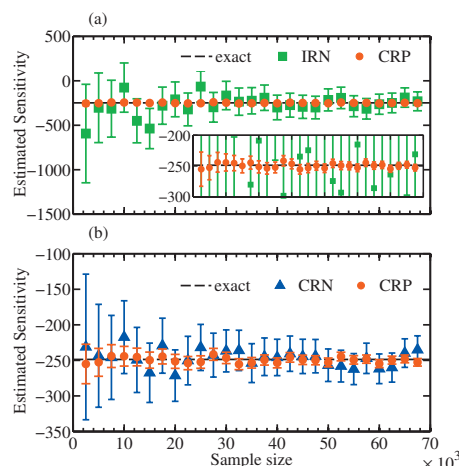


FIG. 2. Estimated sensitivities of $E(X(T, c))$ at final time $T=115$ to changes in parameter c_2 for the birth-death process computed by (a) CRP and IRN methods and (b) IRN and CRN methods. Estimates were computed from independent samples of various sizes with fixed $h=5 \times 10^{-4}$ for each of the methods. (a) The CRP sensitivity estimator exhibits significantly lower variance than those in the IRN. The inset rescales the axes to show detail of the CRP estimates.

example.¹⁵ For deterministic initial condition $X(t_0, \omega, c) = x_0$, the distribution at time t is the sum of independent Binomial and Poisson random variables with distributions $\mathcal{B}(N, p)$ and $\mathcal{P}(\lambda)$, respectively, where $N = x_0$, $p = e^{-c_2 t}$, and $\lambda = (c_1/c_2)(1 - e^{-c_2 t})$. The expected value and variance of $X(t, c)$ are thus computed exactly as follows:

$$E[X(t, c)] = Np + \lambda = x_0 e^{-c_2 t} + (c_1/c_2)(1 - e^{-c_2 t}), \quad (11)$$

$$\begin{aligned} \text{var}(X(t, c)) = Np(1 - p) + \lambda = x_0 e^{-c_2 t}(1 - e^{-c_2 t}) \\ + (c_1/c_2)(1 - e^{-c_2 t}). \end{aligned} \quad (12)$$

For this example, consider the initial condition $x_0=0$ and parameters $c_1=2.5$, $c_2=0.1$.

The proposed CRP algorithm was used to estimate the sensitivities of the population of X at time $T=115$ to changes in the death rate, c_2 . The sensitivity also was computed via Gillespie's direct algorithm⁷ using both IRNs and CRNs. Figures 2–4 show comparisons between the sensitivity estimates obtained by each method.

Figure 2 shows results of the three sensitivity estimators computed by each method using different numbers of independent samples for fixed $h=5 \times 10^{-4}$. The CRP estimator converges to the exact sensitivity coefficient (-248.7) much faster and with much lower variance than the IRN estimator [Fig. 2(a)]. CRP also outperforms estimates computed using CRN, shown in Fig. 2(b).

In Fig. 3(a), $N_{\text{tr}}=10^5$ samples were used by the CRP and IRN methods to estimate the sensitivity coefficient from different sized perturbations h to c_2 . While the exact sensitivity coefficient lies within the confidence intervals for both estimators, the CRP estimator demonstrates far lower variance for small h compared to the IRN method. As seen in Fig. 3(b) the CRP estimator is also more accurate than the CRN estimator, achieving tighter confidence intervals especially for smaller h . When larger magnitude h was used, however, the sample paths of $X(c)$ and $X(c+h)$ lose their strong posi-

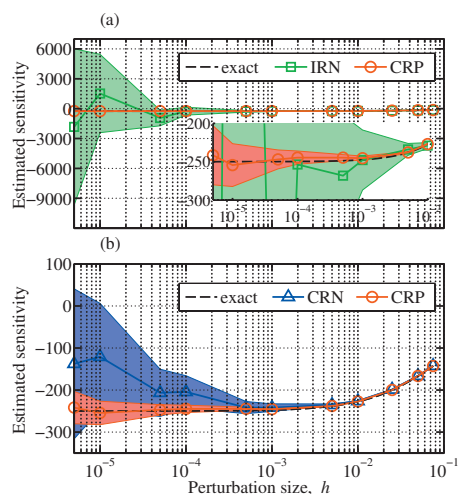


FIG. 3. (a) Estimated sensitivity coefficient for the birth-death process with respect to changes in the death rate c_2 computed by IRN and CRP methods. Insert: rescaled axes detail the low variance estimates found using the CRP method. (b) Sensitivity estimates computed by the CRN and CRP methods. Estimates, plotted as markers, were computed using $N_{tr}=10^5$ trajectories simulated by the algorithm indicated. The black dashed line corresponds to the finite difference approximation computed from the exact solution in Eq. (12). Shaded regions indicate the 95% confidence intervals for the estimates.

tive correlation and the advantage is decreased. These results are seen more clearly by examining the RSE and the speedup factor of each method, shown in Figs. 4(a) and 4(b).

RSE computed from Eq. (9) compares the standard deviation of the estimator relative to the process mean. As demonstrated in Fig. 4(a), the RSE of the estimates for each method is low for large h and increases as the magnitude of h diminishes. RSE first exceeds one and begins a steep ascent at approximately $h=10^{-3}$ for the IRN estimator and near $h=7 \times 10^{-5}$ for the CRN estimator. In comparison RSE of the CRP estimator stays below one for h values as low as 5×10^{-6} .

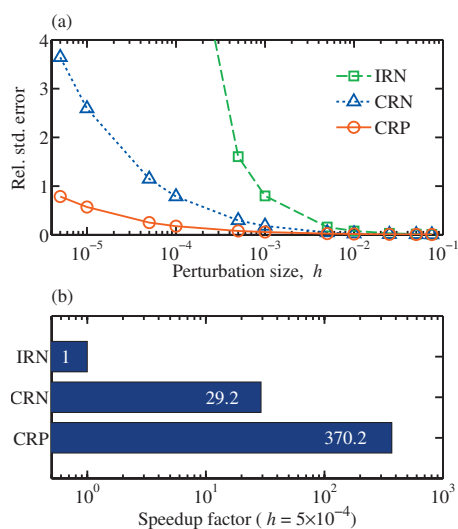


FIG. 4. (a) RSE and (b) computational speed up for birth-death process sensitivity computations. (a) The RSE, computed from 1.25×10^5 samples, is substantially lower for both the CRN and CRP estimators than the IRN estimator for all h . (b) With h fixed and the IRN computation time set as the reference value, the CRP method speeds up the sensitivity computation by a factor of 370, which is more than one order of magnitude higher than the speedup achieved by using the CRN method.

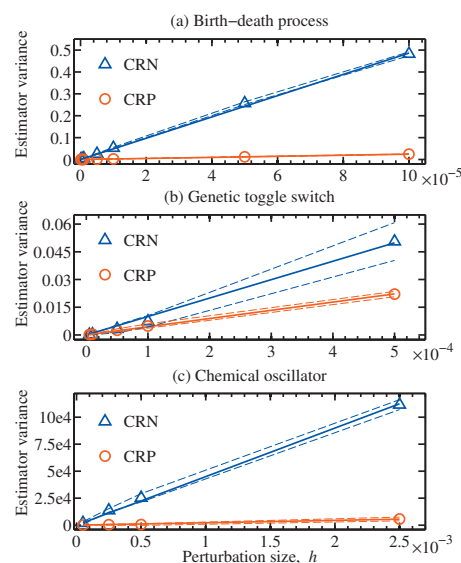


FIG. 5. Variance of the difference estimator, $X(T, c_0+h) - X(T, c_0)$, for decreasing perturbation size h shown for the (a) birth-death process, (b) genetic toggle switch, and (c) chemical oscillator numerical examples. The data points are the variances of each method estimated from many independent samples, the dashed lines indicate the 68% (one standard deviation) confidence intervals of the estimates, and the solid lines are least-squares linear regressions for the estimates. Although the variance at a fixed h depends upon the problem, the variance of the CRP and CRN estimators decrease linearly as $h \rightarrow 0$ for each numerical example considered. For all examples greater variance reduction was observed in the CRP estimator than CRN, and the (problem dependent) ratio of variances is constant for small h .

The practical benefit of computing sensitivity with CRN or CRP instead of IRN is more clearly seen by the computational speedup of Fig. 4(b). The actual computation times depend on the architecture and implementation, therefore we define the *speedup factor* for a method as the time required to compute the sensitivity within a desired accuracy using that method divided by the time required using IRN. Using this metric, the sensitivity estimates with the desired standard deviation are computed nearly 30-fold faster with CRN and more than 350-fold faster with CRP than with IRN.

Note that it is proven in Appendix C that for both CRN and CRP the estimator variance approaches 0 as h approaches 0. In fact the behavior is $\mathcal{O}(h)$ as verified numerically in Fig. 5(a).

Table I(a) lists the CPU time and number of unique uniform random numbers generated as two measures of the computational cost required of estimating the sensitivity coefficient using each method. The IRN and CRN methods take nearly the same amount of CPU time to compute the estimate from a given number of samples, while it takes $\approx 6\%$ longer to generate the CRP estimate. The reduction in variance far exceeds the minimal increase in computational effort; however, the confidence interval for the CRP estimate from 10^4 samples (14.1) is only roughly 5% that of the IRN estimator (283.1) and 27% that of the CRN estimator (52.3).

Additionally, because the CRN and CRP methods reuse random numbers between simulations, fewer unique random numbers must be generated. The CRN method uses half the amount of random numbers as IRN, while CRP uses only one-quarter of the number of random numbers. It should be noted that the CRN and CRP algorithms used to generate

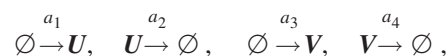
TABLE I. Numerical statistics for the sensitivity analysis performed for (a) the birth-death example and (b) the genetic toggle switch example. The confidence intervals included with the estimate correspond to two standard deviations (approximately 95%). The “Random numbers” column gives the quantity of unique random numbers actually used during simulation for each method.

Samples generated	Method	(a) Birth-death example		$h=5 \times 10^{-4}$	(b) Toggle switch example		$h=1 \times 10^{-3}$
		Estimated sensitivity	Random numbers	CPU time (s)	Estimated sensitivity	Random numbers	CPU time (s)
...	Exact	-248.7	1.19
10 000	IRN	-306.4 ± 283.1	$2.20 \times 10^{+07}$	1.8	231.5 ± 573.4	$2.53 \times 10^{+07}$	3.3
	CRN	-250.8 ± 52.3	$1.10 \times 10^{+07}$	1.8	-1.1 ± 8.0	$1.27 \times 10^{+07}$	3.2
	CRP	-251.0 ± 14.1	$5.52 \times 10^{+06}$	1.9	-1.3 ± 4.6	$6.34 \times 10^{+06}$	3.7
25 000	IRN	-245.4 ± 179.0	$5.51 \times 10^{+07}$	4.4	215.1 ± 362.6	$6.33 \times 10^{+07}$	8.2
	CRN	-231.8 ± 33.1	$2.76 \times 10^{+07}$	4.4	-2.5 ± 5.0	$3.16 \times 10^{+07}$	7.9
	CRP	-247.0 ± 8.9	$1.38 \times 10^{+07}$	4.7	0.5 ± 2.9	$1.59 \times 10^{+07}$	9.2
50 000	IRN	-343.8 ± 126.6	$1.10 \times 10^{+08}$	8.8	92.6 ± 256.4	$1.27 \times 10^{+08}$	16.1
	CRN	-243.3 ± 23.4	$5.52 \times 10^{+07}$	8.8	-1.2 ± 3.6	$6.33 \times 10^{+07}$	15.8
	CRP	-248.0 ± 6.3	$2.76 \times 10^{+07}$	9.3	0.6 ± 2.1	$3.18 \times 10^{+07}$	18.5
75 000	IRN	-349.4 ± 103.4	$1.65 \times 10^{+08}$	13.1	78.5 ± 209.4	$1.90 \times 10^{+08}$	24.1
	CRN	-244.6 ± 19.1	$8.28 \times 10^{+07}$	13.2	-0.8 ± 2.9	$9.49 \times 10^{+07}$	23.8
	CRP	-247.5 ± 5.2	$4.14 \times 10^{+07}$	13.9	1.0 ± 1.7	$4.77 \times 10^{+07}$	27.9
100 000	IRN	-326.5 ± 89.5	$2.20 \times 10^{+08}$	17.5	23.7 ± 181.3	$2.54 \times 10^{+08}$	32.1
	CRN	-247.0 ± 16.6	$1.10 \times 10^{+08}$	17.6	-1.1 ± 2.5	$1.27 \times 10^{+08}$	31.9
	CRP	-248.7 ± 4.5	$5.52 \times 10^{+07}$	18.6	0.9 ± 1.5	$6.37 \times 10^{+07}$	37.4

these results were programed with ease of implementation as the primary concern. Because only the seeds and states of the random number generator were reused between simulations, each CRN was actually generated twice. An alternative implementation that stores random numbers generated between sequential simulations may be able to exploit this property for modest improvements in computation time. However, this clearly will have no effect on the variance properties.

B. Genetic toggle switch

A more interesting example of a stochastic biochemical system is that of the genetic toggle switch.¹⁶ This system consists of two repressor-promoter gene pairs that interact to form a bistable switch for a given set of parameters, and is a prototype for mutually inhibitory genetic circuits exhibiting bistability. The synthesis of one gene product represses the production of the other, and the stochastic nature of the system enables the system to randomly switch between states for which one product is present in large quantities and the other is nearly completely absent. Bistable stochastic genetic switches are of particular relevance because they arise naturally in biology¹⁷ and have also have been constructed in synthetic biological networks.¹⁶ Here we consider a simplified stochastic version of the model presented in Ref. 16. The model describes two species, U and V , whose respective populations are described by the state vector, $X(t, c) = [X_1(t, c) X_2(t, c)]'$. The system transitions between states according to four reactions,



with the following propensity functions a_j and stoichiometry vectors ν_j associated with each reaction channel j :

$$a_1(X(t), c) = \frac{\alpha_1}{1 + X_2^\beta}, \quad a_2(X(t), c) = X_1,$$

$$a_3(X(t), c) = \frac{\alpha_2}{1 + X_1^\gamma}, \quad a_4(X(t), c) = X_2, \quad (13)$$

$$\nu_1 = [1 \quad 0]', \quad \nu_2 = [-1 \quad 0]', \quad \nu_3 = [0 \quad 1]', \quad \nu_4 = [0 \quad -1]'. \quad (14)$$

For this example, we analyze the system having the nominal set $c = c_0$ of model parameters,

$$\alpha_1 = 50, \quad \alpha_2 = 16, \quad \beta = 2.5, \quad \gamma = 1,$$

beginning at $T=0$ with initial conditions $X_1(0)=X_2(0)=0$.

The probability densities for the toggle switch example can be solved to an arbitrarily high degree of accuracy using the finite state projection (FSP) algorithm.¹⁸ The sensitivity of $E(X_1(t))$ for $t=10$ with respect to perturbations in parameter α_1 was computed using the IRN, CRN, and CRP methods. The results are presented in comparison with the exact distributions as computed by the FSP in Figs. 5(b), 6, and 7 and Table I(b). Figure 5(b) numerically verifies that the variances of the CRN and CRP estimators are of $\mathcal{O}(h)$ as $h \rightarrow 0$, indicating that these methods should achieve significant variance reduction for small enough h .

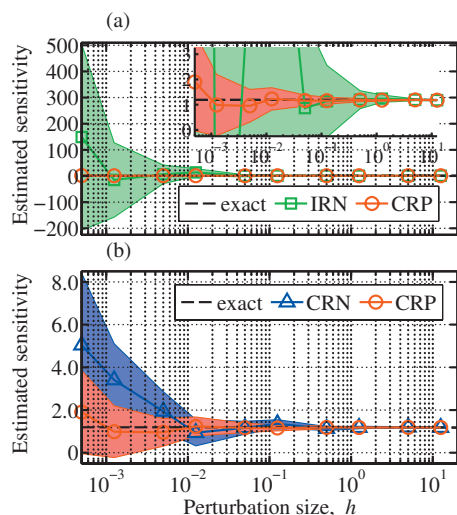


FIG. 6. Sensitivity of $E(X_1(T, c))$ at $T=10$ with respect to changes in α_1 for genetic toggle switch example computed by (a) the IRN and CRP methods and (b) the CRN and CRP methods. Estimates were computed from 10^5 samples simulated using the algorithm indicated. The exact sensitivity coefficient as computed from the distributions obtained by the FSP corresponds to the black dashed line. The CRN and CRP methods achieve much lower variance estimates than the IRN method for small h , with CRP having a slight advantage over the CRN estimator.

Figure 6(a) shows that the CRP estimator outperformed the IRN method in computing the sensitivity coefficient by achieving significantly lower variance, especially at small h . The CRP estimator has slightly lower variance than the CRN estimates for all h [Fig. 6(b)]. Unlike the previous example, the discrepancy between the CRN and CRP methods is not as large. This is further illustrated in the plots for RSE and computational speedup, shown in Fig. 7. The RSE of CRP is roughly 60% that of CRN, and the speedup is over double compared to CRN. However the CRN and CRP methods calculate estimates of comparable accuracy over 5000- and

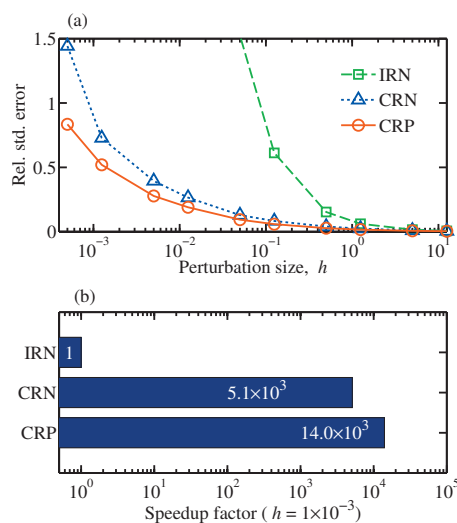


FIG. 7. Relative error and computational speedup for genetic toggle switch sensitivity computations. (a) The relative error (computed from 10^5 samples) is substantially lower for both the CRN and CRP estimators than the IRN estimator. (b) Using either the CRN or CRP methods will compute estimates of a given accuracy over 5000 times faster than using IRN, which is used as the basis for comparison. The CRP method is roughly twice as fast as the CRN method for this example, with a speedup factor of 14 000 over IRN.

TABLE II. Model reactions and parameters for chemical oscillator example.

Reactions	Parameter	Value
$P_a \xrightarrow{c_1} P_a + \text{mRNA}_a$	c_1	50.0
$P_a \xrightarrow{\alpha_a c_1} P_a - A + \text{mRNA}_a$	c_2	0.01
$P_r \xrightarrow{c_2} P_r + \text{mRNA}_r$	c_3	500.0
$P_r \xrightarrow{\alpha_r c_2} P_r - A + \text{mRNA}_r$	c_4	100.0
$\text{mRNA}_a \xrightarrow{c_3} \text{mRNA}_a + A$	c_5	20.0
$\text{mRNA}_r \xrightarrow{c_4} \text{mRNA}_r + R$	c_6	0.0
$A + R \xrightleftharpoons{c_5} A_R$	c_7	20.0
$P_a + A \xrightleftharpoons{c_6, c_7} P_a - A$	c_8	0.0
$P_r + A \xrightleftharpoons{c_8, c_9} P_r - A$	c_9	1.0
$A \xrightarrow{c_{11}} \emptyset$	c_{10}	100.0
$R \xrightarrow{c_{12}} \emptyset$	c_{11}	1.0
$\text{mRNA}_a \xrightarrow{c_{13}} \emptyset$	c_{12}	0.2
$\text{mRNA}_r \xrightarrow{c_{14}} \emptyset$	c_{13}	10.0
$A_R \xrightarrow{c_{15}} R$	c_{14}	0.5
	c_{15}	10.0
	α_a	10.0
	α_r	5000

14 000-fold faster than the IRN, respectively, making them much more efficient than the IRN method. We note that although the advantage in speedup using CRP over CRN is much smaller here (2.7) compared to the previous example (12.7) that this advantage is still significant in cases where simulations can take hours or days to compute.

Comparison among the computational times and number of unique random numbers that were needed for the simulations, listed in Table I(b), shows consistent trends with what was seen in the birth-death example. Namely, the IRN and CRN methods had comparable computation times, while the CRP method required $\approx 16\%$ more CPU time to compute an estimate from the same number of samples.

C. Chemical oscillator

As another example, consider the model of a chemical oscillator¹⁹ in which one repressor protein R and one activator protein A are under the control of their respective promoters P_r and P_a . Protein A is able to bind with both P_a and P_r to significantly enhance the transcription of mRNA_a and mRNA_r and subsequent synthesis of proteins A and R . The repressor protein R inhibits this activity by forming the intermediate complex A_R with protein A before inducing its degradation. This system is capable of inducing periodic oscillations. In total the model consists of nine chemical species participating in 14 chemical reactions, whose propensities depend on 17 kinetic parameters. The reactions and parameters for the model are listed in Table II. The population of each species was set to zero initially, except for promoters P_a and P_r , which were set to one. The sensitivity of protein A at $T=20$ was estimated using the CRN and CRP methods as described previously.

The estimator variances as a function of h are shown for this example in Fig. 5(c). The variances for both the CRN and CRP estimators behave as $\mathcal{O}(h)$ for small h , consistent with the prior two examples and again confirming the proof in Appendix C. Additionally, the ratio of CRP variance to CRN variance indicates that CRP performs much better than CRN in this example, resembling the significant advantage in variance reduction observed previously in example VA.

VI. CONCLUSIONS AND FUTURE WORK

We have demonstrated two efficient alternatives to the familiar approach of using IRNs in Monte Carlo finite difference sensitivity analysis of discrete stochastic chemical networks: the CRN and CRP methods. Sensitivity analysis with CRN can be readily implemented with any version of the popular SSAs by merely running the perturbed and unperturbed trajectories with the same stream of random numbers. The CRP method uses CRNs in the setting of RTC representation of jump Markov processes. The implementation of the CRP algorithm is also relatively easy; the only additional requirement is the availability of a modern random number generator which supports multiple, independent streams of random numbers. Both methods result in an estimator with reduced variance, thereby requiring fewer samples to achieve the same accuracy, although the CRP method performs better than CRN by achieving greater variance reduction.

We showed via numerical examples that both CRP and CRN achieve significant computational speedup over IRN (factors of over 10 000 were observed in one example). The CRP method outperformed the CRN method in all the examples. We also demonstrated that typically in the CRP method the times of reaction events perturb continuously when the parameters are perturbed while in CRN they perturb only piecewise continuously. This we believe is the reason for the greater covariance between the perturbed and unperturbed trajectories and hence the reduced variance of the finite difference estimator in the case of CRP over CRN.

Ongoing work involves direct (without using finite differences) Monte Carlo estimation of infinitesimal sensitivities using the CRP approach as well as theoretical investigations of the CRP approach.

ACKNOWLEDGMENTS

The authors wish to acknowledge financial support from the National Science Foundation under Grant Nos. NSF-DMS-0610013, NSF-IGERT DGE02-21715, NSF-ECCS-0835847, and NSF-ECCS-0802008, by the National Institutes of Health under Grant No. R01-GM04983, and by the Institute for Collaborative Biotechnologies through Grant No. DAAD19-03-D-0004 from the U.S. Army Research Office.

APPENDIX A: THE SIGNIFICANCE OF USING M PARALLEL STREAMS OF RANDOM NUMBERS

Consider the example depicted in Fig. 1. If CRN method is chosen in conjunction with the modified next reaction method proposed by Anderson,¹⁰ the internal times I_1^1, I_2^1, I_3^1 and I_1^2, I_2^2 will be generated from a single stream of unit rate

exponentials. Suppose the first five numbers of this single stream are 0.52, 0.2, 1.1, 0.9, and 1.2. For the situation depicted in the figure, the order in which the reaction channels 1 and 2 fire is 1, 2, 1, 1, 2,.... Thus the internal times are assigned values in the order $I_1^1, I_1^2, I_2^1, I_3^1$ and I_2^2 , and their values are (keeping in mind that $I_0^j=0$ for $j=1,2$)

$$I_1^1 = 0.52, \quad I_1^2 = 0.2, \quad I_2^1 = I_1^1 + 1.1 = 1.62,$$

$$I_3^1 = I_2^1 + 0.9 = 2.52, \quad I_2^2 = I_1^2 + 1.2 = 1.4.$$

Now if keeping the *same single stream* of random numbers 0.52, 0.2, 1.1, 0.9, 1.2,...., one recomputes the modified next reaction method for a different choice of parameter values c_1, c_2 since the propensities change, the order in which the reactions fire might change. Suppose this order changes to 1, 2, 1, 2, 1.... Then the internal times will be assigned the values

$$I_1^1 = 0.52, \quad I_1^2 = 0.2, \quad I_2^1 = I_1^1 + 1.1 = 1.62,$$

$$I_2^2 = I_1^2 + 0.9 = 1.1, \quad I_3^1 = I_2^1 + 1.2 = 2.82.$$

Thus even though the same stream of random numbers are used, the reaction paths as specified by the values of I_i^j are assigned with different values. In the CRP method, by keeping M parallel streams of (unit rate exponential) random numbers, we can ensure that the same values of I_i^j are used when a simulation is repeated with different parameter values. To see this suppose that in the same example, two parallel streams (one for each reaction channel) of unit rate exponential random numbers are used. Suppose the first few numbers of this double stream are 0.43, 1.3, 0.7,... and 0.9, 1.1, 0.1,.... By using the first stream exclusively for reaction channel 1 and the second for reaction channel 2 we ensure the following assignment of values for the internal times I_i^j irrespective of parameter values and thus the exact order in which reactions occur:

$$I_1^1 = 0.43, \quad I_2^1 = I_1^1 + 1.3 = 1.73, \quad I_3^1 = I_2^1 + 0.7 = 2.43, \dots$$

and

$$I_1^2 = 0.9, \quad I_2^2 = I_1^2 + 1.1 = 2.0, \quad I_3^2 = I_2^2 + 0.1 = 2.1, \dots$$

APPENDIX B: CONTINUITY PROPERTIES OF CRP AND CRN

It can be shown that in the CRP method the reaction times T_n are continuous functions of the parameters c in most cases,²⁰ while for CRN implemented in conjunction with other methods, the reaction times T_n are only piecewise continuous in parameters c . Here we use a simple example to show why this is the case for CRP and for CRN implemented in conjunction with Gillespie's direct method.

Let T_n denote the time of the n th reaction event and J_n denote the type of the n th reaction event (J_n is a number in the set $\{1, 2, \dots, M\}$). Let us denote the stream of uniform random number pairs used in Gillespie's direct method by U_n and V_n , where U_n decides the time of n th reaction event and V_n its type. Then we may write

$$T_n(c) = T_{n-1}(c) + \frac{\log(1/U_n)}{a_0(X_{n-1}(c), c)}$$

and $J_n(c)$ is the smallest number J in $\{1, 2, \dots, M\}$ such that

$$\sum_{j=1}^J a_j(X_{n-1}(c), c) > V_n a_0(X_{n-1}(c), c).$$

In addition it follows that

$$X_n(c) = X_{n-1}(c) + \nu_{J_n(c)}.$$

Once we fix the sequence of numbers U_n, V_n , the above three deterministic equations govern the evolution of T_n and X_n for $n=1, 2, \dots$. Assume the initial condition $X_0=x_0$ (deterministic) and set $T_0=0$. For simplicity let us focus on the birth-death example with $a_1(x, c)=c_1$ and $a_2(x, c)=c_2x$. Then

$$T_1(c) = \frac{\log(1/U_1)}{c_1 + c_2x_0},$$

which is a continuous function of both c_1 and c_2 . Suppose c_1 is varied. It is easy to see that in the neighborhood of a typical value of c_1 , $T_1(c)$ and $J_1(c)$ are continuous functions (the latter being constant). However when c_1 is varied across some critical value $c_1=c^*$, $J_1(c)$ will jump. Here c^* satisfies $c^*=V_1(c^*+c_2x_0)$ or equivalently,

$$c^* = \frac{c_2x_0V_1}{1-V_1}.$$

This causes $X_1(c)$ to jump across $c_1=c^*$ as well. In fact when $c_1 < c^*$, $J_1(c)=2$ and $X_1(c)=x_0-1$. (We have assumed $x_0 > 0$.) When $c_1 > c^*$ we have $X_1(c)=x_0+1$. It follows that $T_2(c)$ is given by

$$T_2(c) = T_1(c) + \frac{\log(1/U_2)}{c_1 + c_2(x_0+1)}$$

for $c_1 > c^*$ and

$$T_2(c) = T_1(c) + \frac{\log(1/U_2)}{c_1 + c_2(x_0-1)}$$

for $c_1 < c^*$ indicating a discontinuous change across $c_1=c^*$. Likewise J_2 jumps across $c_1=c^*$. However in addition to the discontinuity at $c_1=c^*$ inherited from that of $J_1(c)$, there will be additional discontinuities in $J_2(c)$. One additional discontinuity occurs at $c_1=c^{**}$ given by

$$c^{**} = \frac{c_2(x_0+1)V_2}{1-V_2}$$

if $[c_2(x_0+1)]/(1-V_2) > c^*$ and another one at $c_1=c^{***}$ given by

$$c^{***} = \frac{c_2(x_0-1)V_2}{1-V_2}$$

if $[c_2(x_0-1)]/(1-V_2) < c^*$. Depending on how c^*, c^{**}, c^{***} are ordered (clearly $c^{**} < c^{***}$) one or both c^{**} and c^{***} will be discontinuities. Likewise $X_2(c)$ and $T_3(c)$ both will have discontinuities at the same c_1 values as $J_2(c)$.

Thus we conclude that for a typical stream of random numbers, the reaction firing times T_n are only piecewise continuous in the parameters c , and the number of discontinuities in T_n increases with n .

On the other hand let us examine what happens when the RTC algorithm is used again with the aid of the birth-death example. Suppose E_1^j, E_2^j, \dots for $j=1, 2$ is a given double stream of unit rate exponential random numbers. Then we obtain the following formulas for T_n and J_n :

$$T_n(c) = T_{n-1}(c) + \min \left\{ \frac{E_+^j(T_{n-1}(c)) - S_j(T_{n-1}(c))}{a_j(X_{n-1}(c), c)} \right\}_{j=1, \dots, M}$$

and $J_n(c)$ equals the j value at which the minimum occurs. Furthermore as before, $X_n=X_{n-1}+\nu_{J_n}$. Again note that $X_0=x_0$ and $T_0=0$. In particular, we have

$$T_1(c) = \min \left\{ \frac{E_1^1}{c_1}, \frac{E_1^2}{c_2x_0} \right\}.$$

As in the case of CRN with Gillespie's direct method, $T_1(c)$ will be continuous in c (because the minimum of M different continuous functions is still a continuous function) and $J_1(c)$ will undergo a jump as c_1 is varied across the value $c_1=c^*$ given by $E_1^1/c^*=E_1^2/(c_2x_0)$, or equivalently

$$c^* = \frac{E_1^1c_2x_0}{E_1^2}.$$

First note that this jump occurs at a point when both reactions fire simultaneously. In other words $T_2=T_1$ at $c_1=c^*$. Also $J_1(c)=2$, $X_1(c)=x_0-1$ for $c_1 < c^*$ and $J_1(c)=2$, $X_1(c)=x_0+1$ for $c_1 > c^*$. Let us examine what happens for $c_1=c^*(1+\epsilon)$ when $\epsilon > 0$ is arbitrarily small. The internal times elapsed by T_1 are given by

$$S_1(T_1) = c_1T_1 = E_1^1$$

and

$$S_2(T_2) = c_2x_0T_1 = \frac{E_1^1c_2x_0}{c_1} \approx E_1^2(1-\epsilon),$$

where we have used $1/(1+\epsilon) \approx 1-\epsilon$. Then T_2 is given by

$$T_2(c) = T_1(c) + \min \left\{ \frac{E_2^1}{c_1}, \frac{\epsilon E_1^2}{c_2(x_0+1)} \right\}.$$

Thus for ϵ near 0, the next reaction to fire will be $J_2=2$ and $T_2=T_1+\mathcal{O}(\epsilon)$. Similarly examining $c_1=c^*(1-\epsilon)$ for arbitrarily small $\epsilon > 0$, we see that $J_2=1$ and $T_2=T_1+\mathcal{O}(\epsilon)$. This shows that T_2 remains continuous across $c_1=c^*$ even though the order of reactions changes from $J_1=1$, $J_2=2$ for $c_1 > c^*$ to $J_1=2$, $J_2=1$ for $c_1 < c^*$.

The critical reason for the continuity property of CRP is that in instances when one reaction is close to firing but loses out to a competing reaction, the reaction times are rescaled rather than reset, ensuring that it will next fire soon. When parameters change, the two reactions swap order of firing with two successive firings occurring very close in physical time. In Gillespie's direct method that a reaction "lost out" to

another reaction—no matter how close to firing it was—is forgotten after the winning reaction fires. Thus when reaction firing orders change, the times for the subsequent reactions also undergo a discontinuous change.

There are exceptions to this continuity property of reaction times T_n in CRP. This happens when the firing of one reaction changes the propensity of a competing reaction to zero, thus preventing it from firing immediately thereafter. A detailed investigation of this phenomenon is beyond the scope of this paper. However our reasoning shows that typically the number of discontinuities of T_n are fewer (if any at all) in CRP when compared to CRN with Gillespie's direct method.

APPENDIX C: ACCURACY OF THE STATISTICAL ESTIMATION OF SENSITIVITY

In Sec. IV A it was noted that methods using CRN (or CRP) decreased the variance of the estimator especially for small h . Typically for the CRP as well as CRN methods when h is sufficiently small, the covariance $\text{cov}(Y_1, Y_2)$ is positive and close to $\text{var}(Y_1)$ and equivalently $\text{var}(Y_1 - Y_2)$ is close to 0. However for the independent sample SSA when c_0 and $c_0 + h$ are close $\text{var}(Y_1 - Y_2)$ is close to $2\text{var}(Y_1)$.

We shall show that $\text{var}(Y_1 - Y_2)$ approaches 0 as h approaches 0. To see this, first note that in CRN and CRP methods, for a given fixed set of CRNs, as $h \rightarrow 0$, $X(T, c_0 + h) \rightarrow X(T, c_0)$ with probability 1. We shall now show that $E[(f(X(T, c_0 + h)) - f(X(T, c_0)))^2] \rightarrow 0$. Suppose that $E(f(X(T, c))^4)$ exists and is continuous in c in a closed ball centered at $c_0 \in \mathbb{R}^p$. Let M be the maximum value of $E(f(X(T, c))^4)$ in this closed ball. Suppose $c_0 + h$ belongs to this ball. Then using the inequality $(a - b)^4 \leq 2^4(a^4 + b^4)$ one obtains that

$$E((f(X(T, c_0 + h)) - f(X(T, c_0)))^4) \leq 2^5 M.$$

Let $Y(h) = f(X(T, c_0 + h)) - f(X(T, c_0))$ be a family of random variables indexed by h . It follows the family $(Y(h))^2$ is bounded in L^2 (two norm) and hence is uniformly integrable.²¹ Since as $h \rightarrow 0$, $X(T, c_0 + h) \rightarrow X(T, c_0)$ with probability 1, it follows that $(Y(h))^2 \rightarrow 0$ with probability 1. By uniform integrability, we obtain that $E((Y(h))^2) \rightarrow 0$, proving that $E[(f(X(T, c + h)) - f(X(T, c)))^2] \rightarrow 0$.

¹H. McAdams and A. Arkin, *Proc. Natl. Acad. Sci. U.S.A.* **94**, 814 (1997).

²M. Thattai and A. van Oudenaarden, *Proc. Natl. Acad. Sci. U.S.A.* **98**, 8614 (2001).

³R. Gunawan, Y. Cao, L. Petzold, and F. J. Doyle, *Biophys. J.* **88**, 2530 (2005).

⁴S. N. Ethier and T. G. Kurtz, *Markov Processes: Characterization and Convergence* (Wiley, New York, 1986).

⁵S. Plyasunov and A. P. Arkin, *J. Comput. Phys.* **221**, 724 (2007).

⁶D. Kim, B. J. Debuschere, and H. N. Najm, *Biophys. J.* **92**, 379 (2007).

⁷D. Gillespie, *J. Phys. Chem.* **81**, 2340 (1977).

⁸C. Rao and A. Arkin, *J. Chem. Phys.* **118**, 4999 (2003).

⁹M. A. Gibson and J. Bruck, *J. Phys. Chem.* **104**, 1876 (2000).

¹⁰D. F. Anderson, *J. Chem. Phys.* **127**, 214107 (2007).

¹¹A. Varma, M. Morbidelli, and H. Wu, *Parametric Sensitivity in Chemical Systems* (Cambridge University Press, Cambridge, 1999).

¹²P. W. Glynn, *Proceedings of the Winter Simulation Conference* (ACM, New York, 1989), p. 90.

¹³P. L'Ecuyer, *Ann. Operat. Res.* **39**, 121 (1992).

¹⁴P. Glasserman and D. D. Yao, *Manage. Sci.* **38**, 884 (1992).

¹⁵M. Rathinam and H. El-Samad, *J. Comput. Phys.* **224**, 897 (2007).

¹⁶T. Gardner, C. R. Cantor, and J. J. Collins, *Nature (London)* **403**, 339 (2000).

¹⁷U. Bhalla and R. Iyengar, *Science* **283**, 381 (1999).

¹⁸B. Munsky and M. Khammash, *J. Chem. Phys.* **124**, 044104 (2006).

¹⁹J. M. G. Vilar, H. Y. Kueh, N. Barkai, and S. Leibler, *Proc. Natl. Acad. Sci. U.S.A.* **99**, 5988 (2002).

²⁰Discontinuity in jump times occurs only when in a sample path two reactions happen to fire simultaneously and the firing of one of these reactions causes the propensity of the other to go to zero.

²¹D. Williams, *Probability with Martingales* (Cambridge University Press, Cambridge, 1991).