# A finite difference method for estimating second order parameter sensitivities of discrete stochastic chemical reaction networks

Elizabeth Skubak Wolf and David F. Anderson

## Additional information on J. Chem. Phys.

# A finite difference method for estimating second order parameter sensitivities of discrete stochastic chemical reaction networks

Elizabeth Skubak Wolf[a)] and David F. Anderson[b)]

*Department of Mathematics, University of Wisconsin, Madison, Wisconsin 53706, USA*

We present an efficient finite difference method for the approximation of second derivatives, with respect to system parameters, of expectations for a class of discrete stochastic chemical reaction networks. The method uses a coupling of the perturbed processes that yields a much lower variance than existing methods, thereby drastically lowering the computational complexity required to solve a given problem. Further, the method is simple to implement and will also prove useful in any setting in which continuous time Markov chains are used to model dynamics, such as population processes. We expect the new method to be useful in the context of optimization algorithms that require knowledge of the Hessian. © *2012 American Institute of Physics*. [http://dx.doi.org/10.1063/1.4770052]

## I. INTRODUCTION

Stochastic models are commonly used to simulate and analyze chemical and biochemical systems, in particular when the abundances of the constituent molecules are small and ordinary differential equations cease to provide a good description of system behavior. The most common modeling choice is to use a continuous time Markov chain (CTMC), which we represent via the stochastic equation (1), but is often described in the biology literature via the chemical master equation, and simulated using Gillespie's algorithm[1,2] or the next reaction method.[3,4]

Parameter sensitivity analysis is a valuable tool in this setting as it provides a quantitative method for understanding how perturbations in the parameters affect different response functions of interest. Further, often the only means of determining the parameters for these models is experimentally. If the model provides a reasonable approximation of the system, the sensitivities can be used to analyze the identifiability of given parameters.[5] They can also suggest an experimental design in which more resources, which are often limited, can be spent determining the more sensitive parameters.

While first derivative sensitivities have been much studied, less focus has been given to finding reasonable algorithms for the computation of sensitivities of higher order, particularly in the discrete state setting. Second derivative sensitivities (the Hessian), however, are also useful. For example, they provide concavity information which is necessary for finding roots or extrema of an expectation. Additionally, in a more general optimization setting, the Hessian can be used to improve upon a simple steepest-descent method. Newton and quasi-Newton methods, for instance, use an approximate Hessian to choose the direction in which to step in the next iterate of the optimization, using curvature to find a more direct path to a local minimum than can be achieved by using the gradient alone. When the Hessian is positive semi-definite, these methods achieve a fast rate of local convergence. Additionally, trust-region based optimization methods can also be markedly improved by including a Hessian estimate.[6–9] Developing algorithms that successfully integrate these optimization methods in the chemical reaction network and CTMC setting, for example in the context of parameter estimation, is a topic of current research which depends critically on having an efficient method for approximating second derivatives.

We introduce here a method for the computation of the second order sensitivities of stochastically modeled biochemical reaction networks that is a nontrivial extension of the coupled finite difference method developed in Ref. 10. The proposed method produces an estimate with a significantly lower variance than existing methods, so that it requires much less central processing unit (CPU) time to produce an approximation within a desired tolerance level. Additionally, the paths generated can also be re-used to compute first derivatives of the system for use in any optimization algorithm. While biochemical reaction networks will be the setting for this paper, the proposed method is also applicable to a wide variety of continuous time Markov chain models, such as those used in queueing theory and the study of population processes.

The outline of the paper is as follows. In Sec. II, we precisely describe the model and problem under consideration. In Sec. III, we present the new method and give a simple algorithm for implementation. In Sec. IV, we provide several numerical examples to compare the new method with existing methods, including finite differencing with common random numbers, finite differencing with the common reaction path method, and second order likelihood transformations. Finally, in Sec. V, we provide some conclusions and discuss avenues for future work.

## II. THE FORMAL MODEL

Suppose we have a system of *d* chemical species undergoing *M* reactions, each with a given propensity function

---

[a)]Electronic mail: skubak@math.wisc.edu.
[b)]Electronic mail: anderson@math.wisc.edu.

$\lambda_k : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ (known as an intensity function in the mathematical literature) and reaction vector (transition direction) $\zeta_k \in \mathbb{R}^d$. We can model this system as a continuous time Markov chain using the random time change representation[11–13]

$$X_t = X_0 + \sum_{k=1}^{M} Y_k \left( \int_0^t \lambda_k(X_s)ds \right) \zeta_k, \qquad (1)$$

where the $Y_k$ are independent, unit-rate Poisson processes and $X_0$ is the initial state. We assume, without loss of generality, that the state space $\mathcal{S}$ is some subset of $\mathbb{Z}_{\geq 0}^d$. That is, the abundances of the constituent species are always non-negative integers. Note that the chemical master equation (forward equation in the language of probability) for the above model is

$$\frac{d}{dt}P_{X_0}(x,t) = \sum_{k=1}^{M} P_{X_0}(x - \zeta_k, t)\lambda_k(x - \zeta_k)1_{\{x - \zeta_k \in \mathbb{Z}_{\geq 0}^d\}}$$

$$- P_{X_0}(x, t) \sum_{k=1}^{M} \lambda_k(x),$$

where $P_{X_0}(x, t)$ is the probability of being in state $x \in \mathbb{Z}_{\geq 0}^d$ at time $t \geq 0$ given an initial condition of $X_0$.

Intuitively, the random time change representation (1) can be understood as follows. Let $R_k(t) := Y_k(\int_0^t \lambda_k(X_s)ds)$. Then $R_k(t)$ counts the number of times the $k$th reaction has occurred up to time $t$, and $R_k(t)\zeta_k$ is the change in the system due to these reactions. The representation (1) then shows that the process at time $t$ is simply its initial value plus the total change up to time $t$ due to each of the $M$ reactions. To understand the counting processes $R_k(t) = Y_k(\int_0^t \lambda_k(X_s)ds)$, picture a realization of the unit-rate Poisson process $Y_k$ as being determined by points on $\mathbb{R}_{\geq 0}$ giving the jump times of $Y_k$.[14] For example, we could have
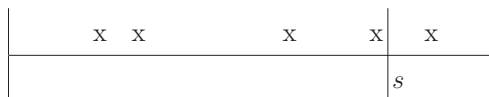


where the "X" marks correspond to the jump times of the Poisson process. At time zero begin at the origin, and as time increases, travel to the right. At a given time $s$ on the line, the value $Y_k(s)$ is how many points we have passed up to and including $s$. For example, in the picture above, $Y_k(s) = 4$. The propensity function $\lambda_k$ then indicates how fast we travel on this line. If $\lambda_k(X_s)$ is very large and $t_1 > t_0$, we expect $\int_0^{t_1} \lambda_k(X_s)ds$ to be much larger than $\int_0^{t_0} \lambda_k(X_s)ds$, so that $Y_k(\int_0^{t_1} \lambda_k(X_s)ds)$ is much larger than $Y_k(\int_0^{t_0} \lambda_k(X_s)ds)$; we have traveled past many points along the line between times $t_0$ and $t_1$, and so were "moving quickly." At the other extreme, if $\lambda_k(X_s)$ is zero between $t_0$ and $t_1$, then $\int_0^{t_1} \lambda_k(X_s)ds = \int_0^{t_0} \lambda_k(X_s)ds$, so that $Y_k(\int_0^{t_1} \lambda_k(X_s)ds) = Y_k(\int_0^{t_0} \lambda_k(X_s)ds)$; in this case we did not travel anywhere on the line, but were "stopped." For further information, intuition, and a derivation of this representation, see Refs. 12, 13, and 15.

Now we suppose that the propensities are dependent on some vector of parameters $\theta$; for instance, $\theta$ may represent a subset of the system's mass action kinetics constants. We then consider a family of models $X_t(\theta)$, parameterized by $\theta$, with stochastic equations

$$X_t(\theta) = X_0(\theta) + \sum_{k=1}^{M} Y_k \left( \int_0^t \lambda_k(\theta, X_s(\theta))ds \right) \zeta_k. \qquad (2)$$

Letting $f$ be some function of interest, for example, the abundance of some molecule, we define

$$J(\theta) := \mathbb{E} f(\theta, X_t(\theta)).$$

This paper is concerned with finding an efficient computational method for the approximation of the second partial derivatives of $J$,

$$\frac{\partial^2}{\partial \theta_j \partial \theta_i} J(\theta).$$

There are several existing methods for the approximation of such sensitivities. The likelihood ratio (LR) method proceeds analytically by moving the derivative inside the expectation.[6,16] The variance of such estimators, however, is often prohibitive. In Sec. IV, we include numerical results from the LR method for comparison. The general method of infinitesimal perturbation (IP) also proceeds by moving the derivative inside the expectation.[17] However, the usual IP methods do not in general apply for any stochastically modeled chemical reaction network in which the firing of one reaction can prevent the firing of another. For example, consider the reaction pair

$$A \to \emptyset, \quad A \to B,$$

when only one A molecule is present. See Ref. 18 or the Appendix of Ref. 17.

Finite difference methods for approximating these sensitivities start with the simple observation that for smooth functions $J$, we may approximate second partial derivatives by perturbing the parameter vector in both relevant directions, so that

$$\frac{\partial^2}{\partial \theta_j \partial \theta_i} J(\theta)$$

$$= \frac{J(\theta + (e_i + e_j)\epsilon) - J(\theta + e_i\epsilon) - J(\theta + e_j\epsilon) + J(\theta)}{\epsilon^2}$$

$$+ O(\epsilon), \qquad (3)$$

where $e_i$ is the vector with a 1 in the $i$th position and 0 elsewhere. Thus, for second derivatives, finite difference methods require up to four simulated paths to produce one estimate, as opposed to the LR method, which requires only one path per estimate. When coupling methods are used with the finite difference, however, the variance of the estimates produced are usually significantly lower than LR, as demonstrated in Sec. IV, so that finite different methods often provide much more effective estimators.

In our setting, Eq. (3) suggests an approximation of the form

$$\frac{\partial^2}{\partial \theta_j \partial \theta_i} J(\theta) \approx \mathbb{E}\left(\frac{f(\theta, X_t(\theta + (e_i + e_j)\epsilon)) - f(\theta, X_t(\theta + e_i\epsilon)) - f(\theta, X_t(\theta + e_j\epsilon)) + f(\theta, X_t(\theta))}{\epsilon^2}\right). \tag{4}$$

The Monte Carlo estimator for (4) with $R$ estimates is then

$$D_R(\epsilon) = \frac{1}{R}\sum_{\ell=1}^{R} d_{[\ell]}(\epsilon), \tag{5}$$

where

$$d_{[\ell]}(\epsilon)$$
$$:= \epsilon^{-2}\big[f(\theta, X_{t,[\ell]}(\theta + (e_i + e_j)\epsilon)) - f(\theta, X_{t,[\ell]}(\theta + e_i\epsilon))$$
$$- f(\theta, X_{t,[\ell]}(\theta + e_j\epsilon)) + f(\theta, X_{t,[\ell]}(\theta))\big],$$

where, for example, $X_{t,[\ell]}(\theta)$ is the $\ell$th path simulated with parameter choice $\theta$. Note that if the four relevant processes are computed independently, which we will call the independent random numbers (IRN) method, the variance of the estimator $D_R(\epsilon)$ is $R^{-1}\mathrm{Var}(d(\epsilon)) = O(R^{-1}\epsilon^{-4})$. The goal of any coupling method in this context is to lower the variance of $d(\epsilon)$ by correlating the relevant processes.

We will demonstrate via example that the method presented here lowers the variance of the numerator of $d(\epsilon)$ to $O(\epsilon)$, thereby lowering the variance of $d(\epsilon)$ to $O(\epsilon^{-3})$, yielding $\mathrm{Var}(D_R(\epsilon)) = O(R^{-1}\epsilon^{-3})$. The proof of this fact follows from work in Ref. 10. For several non-trivial examples, however, the method gives even better performance, lowering the variance of $d(\epsilon)$ another order of magnitude to $O(\epsilon^{-2})$. In contrast, every other coupling method we attempted[19] yielded an asymptotic variance for $d(\epsilon)$ of $O(\epsilon^{-3})$ at best, and in general were much less efficient than the method being proposed here. Theoretical work, and a discussion of conditions on the model guaranteeing the faster rate of convergence, will be presented in a follow-up paper.

For ease of exposition and notation, we have described finite differences using the forward difference (4). Our formal construction will also use the forward difference. In practice, however, it is no more difficult to use the central second difference,

$$\epsilon^{-2}[f(\theta, X_t(\theta + (e_i + e_j)\epsilon/2)) - f(\theta, X_t(\theta + (e_i - e_j)\epsilon/2))$$
$$- f(\theta, X_t(\theta + (e_j - e_i)\epsilon/2)) + f(\theta, X_t(\theta - (e_i + e_j)\epsilon/2))], \tag{6}$$

which has a bias of only $O(\epsilon^2)$; this is what we have implemented in our numerical examples.

## III. COUPLING THE FINITE DIFFERENCE

The goal of any coupling of the finite difference is to reduce the variance of the estimator produced by somehow ensuring that the four paths needed in (3) remain close together. The common random numbers (CRN) coupling achieves this goal by reusing the uniform random numbers in an implementation of Gillespie's direct algorithm.[2] Implicit in Eq. (2) is the common reaction path (CRP) coupling,[20] which assigns

a stream of random numbers to each $Y_k$ which are then used to produce the required realizations of the stochastic processes.

The method presented here, on the other hand, forces the paths to share reactions; often two or more of the four paths have the same reaction occur at the same point in time. Further, the method often naturally "recouples" the processes during the course of a simulation.[10] These facts allow the paths to remain closer than is possible by only sharing random numbers, and so the method consistently produces an estimate with lower variance, often significantly so. We provide numerical evidence for this comparison of methods in Sec. IV; we also briefly revisit this discussion at the end of Sec. III B.

### A. Review of first derivatives

The main idea of the method presented here is most easily seen in the context of first derivatives, where only correlated pairs of runs are required to approximate the first finite difference

$$\epsilon^{-1}(J(\theta + e_i\epsilon) - J(\theta)).$$

The main idea of the coupling presented in this paper is illustrated in the following toy example. Suppose we wish to study the difference between two Poisson processes $Z_1$, $Z_2$ with rates 13.1 and 13, respectively. One way would be to use independent, unit rate Poisson processes $Y_1$, $Y_2$ and write

$$Z_1(t) = Y_1(13.1t) \text{ and } Z_2(t) = Y_2(13t). \tag{7}$$

Then $\mathbb{E}(Z_1(t) - Z_2(t)) = 0.1t$ and $\mathrm{Var}(Z_1(t) - Z_2(t))$ $= \mathrm{Var}(Y_1(13.1t)) + \mathrm{Var}(Y_2(13t)) = 26.1t$.

We would like to lower this variance: instead, write

$$Z_1(t) = Y_1(13t) + Y_2(0.1t) \text{ and } Z_2(t) = Y_1(13t).$$

Then we still have $\mathbb{E}(Z_1(t) - Z_2(t)) = \mathbb{E}Y_2(0.1t) = 0.1t$ as needed, but now $\mathrm{Var}(Z_1(t) - Z_2(t)) = \mathrm{Var}(Y_2(0.1t)) = 0.1t$ instead.

In general, we want to consider the difference of two processes $Z_1$, $Z_2$ with intensities $A$ and $B$ (the intensities may be functions of time, but this is not important to the main idea). In this case, we would write

$$Z_1 = Y_1 + Y_2 \text{ and } Z_2 = Y_1 + Y_3,$$

where $Y_1$, $Y_2$, and $Y_3$ are independent unit-rate Poisson processes that have intensities

$$m := \min\{A, B\}, \ A - m, \text{ and } B - m,$$

respectively. In other words, we have split the counting processes $Z_1$ and $Z_2$ into three sub-processes. One of these, $Y_1$, is shared between $Z_1$ and $Z_2$, so that at the time at which $Y_1$ jumps, both of the original processes jump. These shared jumps lower the variance of the difference $Z_1 - Z_2$ to only $A$

$+ B - 2m = |A - B|$, rather than the $A + B$ which would result from using only two processes similarly to (7).

This is precisely the idea needed for the computation of first derivatives, as in Ref. 10, which is discussed in Sec. III C in more detail. This idea will also serve as the basis for our coupling method for second derivatives.

## B. Construction of the coupling

For the computation of second derivatives, rather than correlated pairs of runs, correlated quartets of runs are used. We suppose we have the four CTMCs of (4), with $i, j \in \{1, \ldots, M\}$ fixed, which for convenience of exposition we order as

$$X_t(\theta + (e_i + e_j)\epsilon), \ X_t(\theta + e_i\epsilon), \ X_t(\theta + e_j\epsilon), \ X_t(\theta). \quad (8)$$

We also assume that their initial conditions are equal (i.e., they are equal at $t = 0$), to some value $X_0(\theta)$. For each of the four processes above, there is an associated propensity for each of the $M$ reaction channels. For example, the propensity of the $k$th reaction channel of the first process (the one with parameter choice $\theta + (e_i + e_j)\epsilon$) is

$$\lambda_{k,1} := \lambda_k(\theta + (e_i + e_j)\epsilon, \ X_t(\theta + (e_i + e_j)\epsilon)).$$

Similarly rename the propensity of the $k$th reaction channel of the second process (parameter choice $\theta + e_i\epsilon$) by $\lambda_{k,2}$, the third process as $\lambda_{k,3}$, and the fourth process as $\lambda_{k,4}$, as per our ordering (8). Note that these propensities are dependent on $\theta$ and $X_t(\theta)$, but we will drop either or both of these dependencies in our notation when they are not relevant to the current discussion.

Next, we introduce a coupling of these four processes that will produce an estimator (5) with low variance. The main idea is similar to that in Sec. III A as well as in Ref. 10 in that it rests on splitting a counting process into sub-processes, to be shared among the four CTMCs (8). With this goal in mind, we create a sub-process to allow the 1st and 2nd processes of (8) to jump simultaneously, one to allow the 1st and 3rd to jump simultaneously, one for the 2nd and 4th, and one for the 3rd and 4th. Additionally, we create a sub-process that allows all four to jump simultaneously. As in the first derivative setting, the rates of these sub-processes will involve minimums of the original CTMCs. Finally, we also require four additional sub-processes to make up any "leftover" propensity of the original CTMCs.

Formally, define $R_{k,[b_1,b_2,b_3,b_4]}$ as a counting process, where $b_\ell \in \{0, 1\}$. A jump of $R_{k,[b_1,b_2,b_3,b_4]}$ indicates that the $\ell$th process in the ordering (8) jumps by reaction $k$ if and only if $b_\ell = 1$, for $\ell \in \{1, 2, 3, 4\}$. For example, $R_{k,[1,1,0,0]}(t)$ counts the number of times the $k$th reaction has fired simultaneously for the first and second processes of (8) (but the third and fourth did not fire), whereas $R_{k,[1,0,1,0]}(t)$ counts the number of times the $k$th reaction has fired simultaneously for the first and third processes of (8) (but the second and fourth did not fire). Define the propensity of $R_{k,[b_1,b_2,b_3,b_4]}$ by $\Lambda_{k,[b_1,b_2,b_3,b_4]}$,

so that in the random time change representation (2),

$$R_{k,[b_1,b_2,b_3,b_4]}(t) = Y_{k,[b_1,b_2,b_3,b_4]} \left( \int_0^t \Lambda_{k,[b_1,b_2,b_3,b_4]}(s)ds \right), \quad (9)$$

where the $Y$'s are independent unit-rate Poisson processes and where the propensities are

$$\Lambda_{k,[1,1,1,1]} = \lambda_{k,1} \wedge \lambda_{k,2} \wedge \lambda_{k,3} \wedge \lambda_{k,4},$$

$$\Lambda_{k,[1,1,0,0]} = \lambda_{k,1} \wedge \lambda_{k,2} - \Lambda_{k,[1,1,1,1]},$$

$$\Lambda_{k,[0,0,1,1]} = \lambda_{k,3} \wedge \lambda_{k,4} - \Lambda_{k,[1,1,1,1]},$$

$$\Lambda_{k,[1,0,1,0]} = (\lambda_{k,1} - \lambda_{k,1} \wedge \lambda_{k,2}) \wedge (\lambda_{k,3} - \lambda_{k,3} \wedge \lambda_{k,4}),$$

$$\Lambda_{k,[0,1,0,1]} = (\lambda_{k,2} - \lambda_{k,1} \wedge \lambda_{k,2}) \wedge (\lambda_{k,4} - \lambda_{k,3} \wedge \lambda_{k,4}),$$
$$\qquad (10)$$

$$\Lambda_{k,[1,0,0,0]} = (\lambda_{k,1} - \lambda_{k,1} \wedge \lambda_{k,2}) - \Lambda_{k,[1,0,1,0]},$$

$$\Lambda_{k,[0,1,0,0]} = (\lambda_{k,2} - \lambda_{k,1} \wedge \lambda_{k,2}) - \Lambda_{k,[0,1,0,1]},$$

$$\Lambda_{k,[0,0,1,0]} = (\lambda_{k,3} - \lambda_{k,3} \wedge \lambda_{k,4}) - \Lambda_{k,[1,0,1,0]},$$

$$\Lambda_{k,[0,0,0,1]} = (\lambda_{k,4} - \lambda_{k,3} \wedge \lambda_{k,4}) - \Lambda_{k,[0,1,0,1]},$$

where we define the notation $a \wedge b := \min\{a, b\}$. The proposed coupling is then given by the following:

$$X_t(\theta + (e_i + e_j)\epsilon)$$
$$= X_0(\theta) + \sum_k \zeta_k (R_{k,[1,1,1,1]}(t) + R_{k,[1,1,0,0]}(t)$$
$$\quad + R_{k,[1,0,1,0]}(t) + R_{k,[1,0,0,0]}(t)),$$

$$X_t(\theta + e_i\epsilon)$$
$$= X_0(\theta) + \sum_k \zeta_k (R_{k,[1,1,1,1]}(t) + R_{k,[1,1,0,0]}(t)$$
$$\quad + R_{k,[0,1,0,1]}(t) + R_{k,[0,1,0,0]}(t)),$$
$$\qquad (11)$$

$$X_t(\theta + e_j\epsilon)$$
$$= X_0(\theta) + \sum_k \zeta_k (R_{k,[1,1,1,1]}(t) + R_{k,[0,0,1,1]}(t)$$
$$\quad + R_{k,[1,0,1,0]}(t) + R_{k,[0,0,1,0]}(t)),$$

$$X_t(\theta) = X_0(\theta) + \sum_k \zeta_k (R_{k,[1,1,1,1]}(t) + R_{k,[0,0,1,1]}(t)$$
$$\quad + R_{k,[0,1,0,1]}(t) + R_{k,[0,0,0,1]}(t)).$$

A few comments are in order. First, note that, for example, the marginal process $X_t(\theta + (e_i + e_j)\epsilon)$ above involves all the counting processes in which $b_1 = 1$. Second, each of these marginal processes $X_t(\cdot)$ have the same distribution as the original, uncoupled, processes since the transition rates of the marginal processes have remained unchanged. This can be checked by simply summing the rates of the relevant counting processes, which are all those $\Lambda_{k,[b_1,b_2,b_3,b_4]}$ in which a given $b_\ell = 1$. Third, if $f$ is linear, for example if we are estimating the abundance of a particular molecule, many of the $R_{k,[b_1,b_2,b_3,b_4]}$ are completely cancelled if we now construct the difference (4). An example of this will be shown in Sec. IV A. Fourth, even if $i = j$, the coupling requires two different copies of the process $X_t(\theta + e_i\epsilon)$, one taking the role of $X_t(\theta + e_i\epsilon, t)$ and the other $X_t(\theta + e_j\epsilon)$.

As discussed at the beginning of this section, the CRN and CRP methods attempt to reduce the variance of the estimator (5) by reusing random numbers for each of the four nominal processes. However, as discussed in Ref. 10 in the setting of first derivatives, this will often lead to a decoupling over long enough time periods. Hence, the variance of the CRN and CRP estimators will often eventually converge to a variance of the same order of magnitude as the estimator constructed using independent samples. This behavior is demonstrated by example in the current setting of second derivatives in Sec. IV. The double coupled method presented here *recouples* the four relevant processes every time they are near each other, which, by contrast, does not occur in either CRN or CRP. We refer the interested reader to Ref. 10, Sec. 3.1 for a more thorough discussion of this idea.

## C. An alternative derivation

The coupling described in Sec. III B can be derived in an alternate way, which explains why the method is termed "double coupled." We could first couple the first and second processes of (8) using the coupled finite difference method,[10] and then couple the third and fourth in the same manner. For example, using the $\lambda_{k,\ell}$ as defined in Sec. III B, the first two processes in (8) are constructed as

$$X(\theta + (e_i + e_j)\epsilon, t) = X_0(\theta) + \sum_k (R_{k,[1,1]} + R_{k,[1,0]})\zeta_k,$$

$$X(\theta + e_i\epsilon, t) = X_0(\theta) + \sum_k (R_{k,[1,1]} + R_{k,[0,1]})\zeta_k,$$

(12)

where $R_{k,[b_1,b_2]} = Y_{k,[b_1,b_2]}(\int_0^t \Lambda_{k,[b_1,b_2]}(s)ds)$ are defined analogously to (9) and where

$$\Lambda_{k,[1,1]}(s) = \lambda_{k,1}(s) \wedge \lambda_{k,2}(s),$$

$$\Lambda_{k,[1,0]}(s) = \lambda_{k,1}(s) - \lambda_{k,1}(s) \wedge \lambda_{k,2}(s),$$

$$\Lambda_{k,[0,1]}(s) = \lambda_{k,2}(s) - \lambda_{k,1}(s) \wedge \lambda_{k,2}(s).$$

As in (11), the processes defined in (12) jump together as often as possible: they share the sub-processes $R_{k,[1,1]}$, each of which runs at a propensity equal to the minimum of the respective propensities of the two original processes. We then expect the variance of the first finite difference $[f(\theta, X_t(\theta + (e_i + e_j)\epsilon)) - f(\theta, X_t(\theta + e_i\epsilon))]\epsilon^{-1}$ to be small since the two processes of (12) will remain approximately the same whenever they jump simultaneously via $R_{k,[1,1]}$.

Now note that, together, the two processes (12) can be viewed as a new CTMC with dimension $2d$, twice that of that of the original process. The third and fourth processes in (8) can be similarly coupled, giving us *two* $2d$-dimensional CTMCs. Finally, we couple these new processes into a single CTMC of dimension $4d$, in precisely the same manner of Ref. 10. This construction leads to the same process as given in (11). The details are left to the interested reader.

## D. Algorithms for simulation of (11)

We present two algorithms for the pathwise simulation of Eq. (11). The first corresponds to the next reaction method of

Ref. 4, whereas the second corresponds to an implementation of Gillespie's direct method.[1,2] As usual, it will be problem specific as to which algorithm is most efficient.

Below, rand(0,1) indicates a uniform[0,1] random variable, independent from all previous random variables. Recall that if $U \sim \text{rand}(0,1)$, then $\ln(1/U)/\lambda$ is exponentially distributed with parameter $\lambda > 0$. Also recall that even if $i$ and $j$ are equal, the processes $X(\theta + e_i\epsilon)$ and $X(\theta + e_j\epsilon)$ are still constructed separately. Define the set

$$B := \{[1, 1, 1, 1], [1, 1, 0, 0], [0, 0, 1, 1], [1, 0, 1, 0],$$

$$[0, 1, 0, 1], [1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]\}$$

and note that it will often be convenient to use a *for* loop, from 1 to 9, to enumerate over the vectors in $B$.

ALGORITHM—Modified next reaction method applied to (11).

**Initialization**: Set $X(\theta + (e_i + e_j)\epsilon) = X(\theta + e_i\epsilon) = X(\theta + e_j\epsilon) = X(\theta) = X_0$ and $t = 0$; for each $k \in \{1, \ldots, M\}$ and each $b \in B$, set $T_{k,b} = 0$ and $P_{k,b} = \ln(1/u_{k,b})$ for $u_{k,b} \sim \text{rand}(0,1)$.

**Repeat** the following steps:

(i)   For each $k$, set

$$\lambda_{k,1} = \lambda_k(\theta + (e_i + e_j)\epsilon, X(\theta + (e_i + e_j)\epsilon)),$$

$$\lambda_{k,2} = \lambda_k(\theta + e_i\epsilon, X(\theta + e_i\epsilon)),$$

$$\lambda_{k,3} = \lambda_k(\theta + e_j\epsilon, X(\theta + e_j\epsilon)),$$

$$\lambda_{k,4} = \lambda_k(\theta, X(\theta)),$$

and use to set each of the nine variables $\Lambda_{k,b}$ as above in (10).

(ii)  For each $k$ and $b \in B$, set

$$\Delta t_{k,b} = \begin{cases} (P_{k,b} - T_{k,b})/\Lambda_{k,b} , & \text{if } \Lambda_{k,b} > 0 \\ \infty & , \quad \text{else} \end{cases}.$$

(iii) Set $\Delta = \min_{k,b}\{\Delta t_{k,b}\}$ and let $\mu := k$ and $\nu := b = [b_1, b_2, b_3, b_4]$ be the indices where the minimum is achieved.

(iv)  Set $t = t + \Delta$.

(v)   Update state vector variables $X(\theta + (e_i + e_j)\epsilon)$, $X(\theta + e_i\epsilon)$, $X(\theta + e_j\epsilon)$, $X(\theta)$ by adding $\zeta_\mu$ to the $\ell$th process if and only if $b_\ell = 1$ in $\nu$.

(vi)  For each $k$ and $b \in B$, set $T_{k,b} = T_{k,b} + \Delta \cdot \Lambda_{k,b}$.

(vii) Set $P_{\mu,\nu} = P_{\mu,\nu} + \ln(1/u)$ where $u \sim \text{rand}(0,1)$.

(viii) Return to (i) or quit.

ALGORITHM—Gillespie's direct method applied to (11).

**Initialization**: Set $X(\theta + (e_i + e_j)\epsilon) = X(\theta + e_i\epsilon) = X(\theta + e_j\epsilon) = X(\theta) = X_0$ and $t = 0$.

**Repeat** the following steps:

(i) For each $k$, set

$$\lambda_{k,1} = \lambda_k(\theta + (e_i + e_j)\epsilon, X(\theta + (e_i + e_j)\epsilon)),$$

$$\lambda_{k,2} = \lambda_k(\theta + e_i\epsilon, X(\theta + e_i\epsilon)),$$

$$\lambda_{k,3} = \lambda_k(\theta + e_j\epsilon, X(\theta + e_j\epsilon)),$$

$$\lambda_{k,4} = \lambda_k(\theta, X(\theta)),$$

and use to set each of the nine variables $\Lambda_{k,b}$ as above in (10).

(ii) Let $\Lambda_0 = \sum_k \sum_b \Lambda_{k,b}$ and $u \sim \text{rand}(0, 1)$, and set

$$\Delta = \ln(1/u)/\Lambda_0.$$

(iii) Set $t = t + \Delta$.

(iv) Let $u \sim \text{rand}(0, 1)$ and use to select $(\mu, v) \in \{(k, b) : k \in \{1, \ldots, M\}, b \in B\}$ where each pair $(k, b)$ is selected with probability $\lambda_{k,b}/\Lambda_0$.[21]

(v) Update state vector variables $X(\theta + (e_i + e_j)\epsilon)$, $X(\theta + e_i\epsilon)$, $X(\theta + e_j\epsilon)$, $X(\theta)$ by adding $\zeta_\mu$ to the $\ell$th process if and only if $b_\ell = 1$ in $v$.

(vi) Return to (i) or quit.

## IV. NUMERICAL EXAMPLES

In this section, we compare the double coupled method with the following existing methods:

(a) the usual independent random numbers estimator in which the processes of (4) are simulated independently, also referred to as the crude Monte Carlo method,

(b) the common random numbers approach in which the processes of (4) are simulated given the same stream of random numbers using Gillespie's direct algorithm,[22]

(c) the common reaction path method of Ref. 20 in which the processes of (4) are coupled by reusing each $Y_k$ of (2),

(d) the double coupled method proposed here (CFD2, where the CFD stands for "coupled finite difference") which implements the coupling (11),

(e) a Girsanov transformation or likelihood ratio method in which the computed weight function is used as a control variate (see Ref. 16 and Chapters V.2 and VII.3 of Ref. 6).

All methods except (b) were simulated using the next reaction algorithm, modified as necessary. We also note that the first four methods use the second finite difference, which has some bias (see Sec. II); recall that to reduce this bias we actually simulate the centered difference (6), which is accomplished in the same way as the forward difference but with the parameters shifted. The LR method is the only one of the four methods we use here that is unbiased; its high variance, however, typically makes the method unusable. Finally, when discussing performance, we will refer to $R$ of (5) as the number of estimates.

### A. A simple birth process

Consider a pure birth process $A \to 2A$. Here, $\zeta = 1$, and denoting by $X_t$ the number of $A$ molecules at time $t$, we assume a propensity function $\lambda(\theta, X_t(\theta))$, so that in the random time change representation,

$$X_t(\theta) = X_0 + Y\left(\int_0^t \lambda(\theta, X_s(\theta))ds\right),$$

where, as usual, $Y$ is a unit-rate Poisson process.

Suppose we are interested in the second derivative of $\mathbb{E}X_t$ with respect to $\theta$ (so that $f(\theta, x) = x$). We double couple the processes as in (11), noting that we are in the special case when $i = j$. This does not change the main idea of the double coupling, but it requires us to distinguish the two nominal processes with the same parameter value $\theta + \epsilon$; we label them as $X_t^1(\theta + \epsilon)$ and $X_t^2(\theta + \epsilon)$. Ordering as in (8), and noting that since there is only one reaction we may drop the subscript $k$, we find that

$$\lambda_1 = \lambda(\theta + 2\epsilon, X_t(\theta + 2\epsilon)),$$

$$\lambda_2 = \lambda(\theta + \epsilon, X_t^1(\theta + \epsilon)),$$

$$\lambda_3 = \lambda(\theta + \epsilon, X_t^2(\theta + \epsilon)),$$

$$\lambda_4 = \lambda(\theta, X_t(\theta)),$$

and use these to define the $\Lambda$'s as given in (10). The double coupled processes are then given as

$$X_t(\theta + 2\epsilon) = X_0(\theta) + R_{[1,1,1,1]}(t) + R_{[1,1,0,0]}(t)$$
$$+ R_{[1,0,1,0]}(t) + R_{[1,0,0,0]}(t),$$

$$X_t^1(\theta + \epsilon) = X_0(\theta) + R_{[1,1,1,1]}(t) + R_{[1,1,0,0]}(t)$$
$$+ R_{[0,1,0,1]}(t) + R_{[0,1,0,0]}(t),$$

$$X_t^2(\theta + \epsilon) = X_0(\theta) + R_{[1,1,1,1]}(t) + R_{[0,0,1,1]}(t)$$
$$+ R_{[1,0,1,0]}(t) + R_{[0,0,1,0]}(t),$$

$$X_t(\theta) = X_0(\theta) + R_{[1,1,1,1]}(t) + R_{[0,0,1,1]}(t)$$
$$+ R_{[0,1,0,1]}(t) + R_{[0,0,0,1]}(t).$$

Now that we have coupled the processes, note that when we consider the second difference (4) for the given $f$, which is linear, most of the sub-processes cancel. For example, since $R_{[1,1,0,0]}$ is present in both $X_t(\theta + 2\epsilon)$, which is positive in the difference, and in $X_t^1(\theta + \epsilon)$, which is negative, $R_{[1,1,0,0]}$ is not present in the second difference. One can easily check that the numerator of the difference (4) simplifies in this case to

$$X_t(\theta + 2\epsilon) - X_t^1(\theta + \epsilon) - X_t^2(\theta + \epsilon) + X_t(\theta)$$
$$= R_{[1,0,0,0]}(t) - R_{[0,1,0,0]}(t) - R_{[0,0,1,0]}(t) + R_{[0,0,0,1]}(t).$$
$$(13)$$

Note that the rates of the four remaining counting processes of (13) are usually relatively small; in fact, at any given time at least two of the four must have zero propensity, as can be seen by considering the possible values of the minima involved.

Suppose that $\lambda(\theta, X_t(\theta)) = \theta X_t(\theta)$ is simply a constant times the population at time $t$. We choose to estimate $\frac{\partial^2 \mathbb{E}X_t}{\partial\theta^2}$

TABLE I. Confidence intervals (95%) and computation time for each of the five methods, after 100 000 estimates, on the simple birth model of Sec. IV A (with linear propensity). An $\epsilon$ of 1/50 was used for the three finite difference methods. Actual value: 304.6.

| Method | Estimates | Approximation | No. of updates | CPU time (s) |
|--------|-----------|---------------|----------------|--------------|
| IRN    | 100 000   | $307 \pm 447$ | $\approx 3.7 \times 10^6$ | 38 |
| CRP    | 100 000   | $315 \pm 24$  | $\approx 3.7 \times 10^6$ | 49 |
| CRN    | 100 000   | $282 \pm 24$  | $\approx 3.3 \times 10^6$ | 32 |
| LR     | 100 000   | $311 \pm 20$  | $\approx 1.1 \times 10^6$ | 37 |
| CFD2   | 100 000   | $296 \pm 12$  | $\approx 1.2 \times 10^6$ | 22 |

at $t = 5$ and $\theta = 1/2$, with $X_0(\theta) = 1$. We use $\epsilon = 1/50$ for the finite difference methods. For simple examples such as this, one can solve for the derivative explicitly; in this case the actual value is 304.6.

As can be seen in the data in Table I, manifested in the width of the confidence interval, the variance of the double coupled estimator is smaller than that of the estimators given by the other methods. For instance, for the same number of estimates it gives a confidence interval of half the width of the CRP and CRN methods, which for this single-reaction model, though implemented differently, give equivalent estimators. Here and throughout, confidence intervals are constructed as $\pm 1.96\sqrt{v}$ where $v$ is the variance of the estimator (5).

For each method, we also include the CPU time that was required for the simulation, as well as the number of updates made to the system state (the number of times a reaction vector is added to the state vector). The latter is a useful comparison tool, as it provides a measure of the amount of work the method requires, but is not influenced by differences in implementation (such as use of Gillespie vs next reaction algorithms). These differences, on the other hand, often affect CPU time. We do not also provide a random number count for each method, but note here that except for CRN this number is equal to the number of system updates. For CRN, which uses Gillespie's algorithm, two random numbers are used per system update. Finally, the CPU time will certainly vary by machine; all tests described in this section were run in MATLAB on a Windows machine with a 1.6 GHz processor.

## B. mRNA transcription and translation

We now examine the performance of the proposed method on a more realistic model. In the following model of gene transcription and translation, mRNA is being created, and then translated into protein, while both the mRNA and the protein may undergo degradation (where here the constants are in the sense of mass action kinetics, so, for example, protein is being created at a rate of $\gamma$ times the number of mRNA molecules):

$$\emptyset \underset{\theta}{\overset{2}{\rightleftarrows}} M \overset{\gamma}{\to} M + P, \qquad P \overset{1}{\to} \emptyset.$$
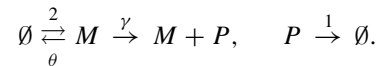
We assume initial concentrations of zero mRNA and protein molecules. The stochastic equation for this model is

$$X(\theta, t) = Y_1(2t)\begin{pmatrix} 1 \\ 0 \end{pmatrix} + Y_2\left(\int_0^t \theta X_M(\theta, s)ds\right)\begin{pmatrix} -1 \\ 0 \end{pmatrix}$$
$$+ Y_3\left(\int_0^t \gamma X_M(\theta, s)ds\right)\begin{pmatrix} 0 \\ 1 \end{pmatrix}$$
$$+ Y_4\left(\int_0^t X_P(\theta, s)ds\right)\begin{pmatrix} 0 \\ -1 \end{pmatrix},$$

where $X = \begin{pmatrix} X_M \\ X_P \end{pmatrix}$ gives the numbers of the mRNA and protein molecules respectively. Note that we have moved the parameter $t$ from the subscript for notational convenience.

In Subsection IV B 1, we compute the second derivative of the expected number of protein molecules with respect to $\theta$, while in Subsection IV B 2, we compute the mixed partial of this same quantity with respect to both $\theta$ and $\gamma$. In Subsection IV B 3, we compute the second derivative of the *square* of the expected number of protein molecules with respect to $\theta$.

### 1. Second derivative of protein abundance with respect to $\theta$

Suppose we would like to estimate the second derivative of the expected number of protein molecules with respect to $\theta$ at a time of $t = 30$ and $\theta = \frac{1}{4}$. Additionally, we fix $\gamma = 10$ and $X_0 = 0$. One can analytically find that $\frac{\partial^2}{\partial\theta^2}\mathbb{E}X_P(30) = 2496$.

First, Table II gives simulation data as in the previous examples, with two different perturbations, $\epsilon$, of $\theta$ used. Note

TABLE II. Confidence intervals (95%) for each of the finite difference methods for the computation in the mRNA and protein model of Subsection IV B 1. Note that the bias of the second finite difference can be seen when $\epsilon = 1/20$ (the actual value is 2496). Also note that, though for a fixed number of estimates the CFD2 method is not the fastest method, it achieves a much smaller confidence interval. The number of updates and computational time for a fixed number of estimates are essentially independent of $\epsilon$ and so the reported values, here and throughout, are the average of the values for the two choices of $\epsilon$.

| Method | Estimates | $\epsilon = 1/20$ | $\epsilon = 1/100$ | No. of updates | CPU time (s) |
|--------|-----------|-------------------|--------------------|----------------|--------------|
| CRN    | 1000      | $2682 \pm 1192$   | $5950 \pm 19\,123$  | $\approx 1.26 \times 10^7$ | 46 |
| CRP    | 1000      | $2758 \pm 569$    | $-2630 \pm 9268$    | $\approx 1.27 \times 10^7$ | 70 |
| CFD2   | 1000      | $2655 \pm 129$    | $2640 \pm 1001$     | $\approx 4.68 \times 10^6$ | 48 |
| CRN    | 10 000    | $2453 \pm 369$    | $1505 \pm 6120$     | $\approx 1.27 \times 10^8$ | 457 |
| CRP    | 10 000    | $2783 \pm 179$    | $2627 \pm 2937$     | $\approx 1.27 \times 10^8$ | 672 |
| CFD2   | 10 000    | $2601 \pm 40$     | $2352 \pm 282$      | $\approx 4.68 \times 10^7$ | 483 |
| CRN    | 40 000    | $2386 \pm 188$    | $1069 \pm 2984$     | $\approx 5.07 \times 10^8$ | 1829 |
| CRP    | 40 000    | $2745 \pm 89$     | $3593 \pm 1468$     | $\approx 5.07 \times 10^8$ | 2739 |
| CFD2   | 40 000    | $2582 \pm 20$     | $2512 \pm 147$      | $\approx 1.87 \times 10^8$ | 1931 |

TABLE III. Confidence intervals (95%) for the LR method (d) for the computation in the mRNA transcription model computation of Subsection IV B 1. Even though this method is fastest per estimate, note that the variance (and so the width of the confidence interval) is large.

| Estimates | Approximation | No. of updates | CPU time (s) |
|---|---|---|---|
| 1000 | $2150 \pm 2258$ | $\approx 4.20 \times 10^6$ | 14 |
| 10 000 | $2429 \pm 729$ | $\approx 4.19 \times 10^7$ | 135 |
| 40 000 | $2176 \pm 404$ | $\approx 1.68 \times 10^8$ | 540 |

the trade-off between bias and precision: a larger epsilon implies the second finite difference has a larger bias, but, since there is an $\epsilon^2$ in the denominator of the estimator, the variance of the estimator is smaller; for small epsilon it is vice versa. Table III shows the relevant data for the LR method.

Perhaps more illustrative is Table IV, which compares the numbers of estimates and system updates as well as the time required to achieve a 95% confidence interval of a set width. These data give a good idea of the efficiency of the methods, as often one desires the estimate within a given tolerance. We can see that the double coupled method is approximately 25 times faster than CRP, 73 times faster than the often used CRN method, over 100 times faster than the LR method, and over 125 times faster than IRN. Note also that the double coupled method requires drastically fewer estimates to achieve the same confidence, so that, even though the computation of one double coupled estimate requires more time than most of the other methods, as can be seen in Table II, the lower variance leads to very large time savings.

Finally, in Figure 1 we include a plot of the variance of the different estimators versus time. Note that the scales on the plots are very different. The plots corresponding to finite difference methods all appear to converge; the limiting value for the double coupled method, however, is over 20 times smaller than the CRP method, and over 170 times smaller than the CRN and IRN methods. Note also that, as time increases, the CRN variance tends to the same value as the IRN method; this is expected, since we expect the processes to decouple. The variance for CRP behaves similarly, converging to a number of approximately the same order of magnitude as the IRN method, though the value itself is significantly lower in this four-reaction model. The plot for the LR method scales quadratically, as is expected by the form of the estimator (see Chapter VII.3 of Ref. 6). This shows that, for moderate and large times, the double coupled method quickly becomes much more efficient then the other estimators.

TABLE IV. Required estimates, updates, and computational time needed for 95% confidence intervals of $\pm 120$ for all five methods on the computation of the mRNA transcription model computation of Subsection IV B 1. An $\epsilon$ of 1/20 was used for the finite difference methods.

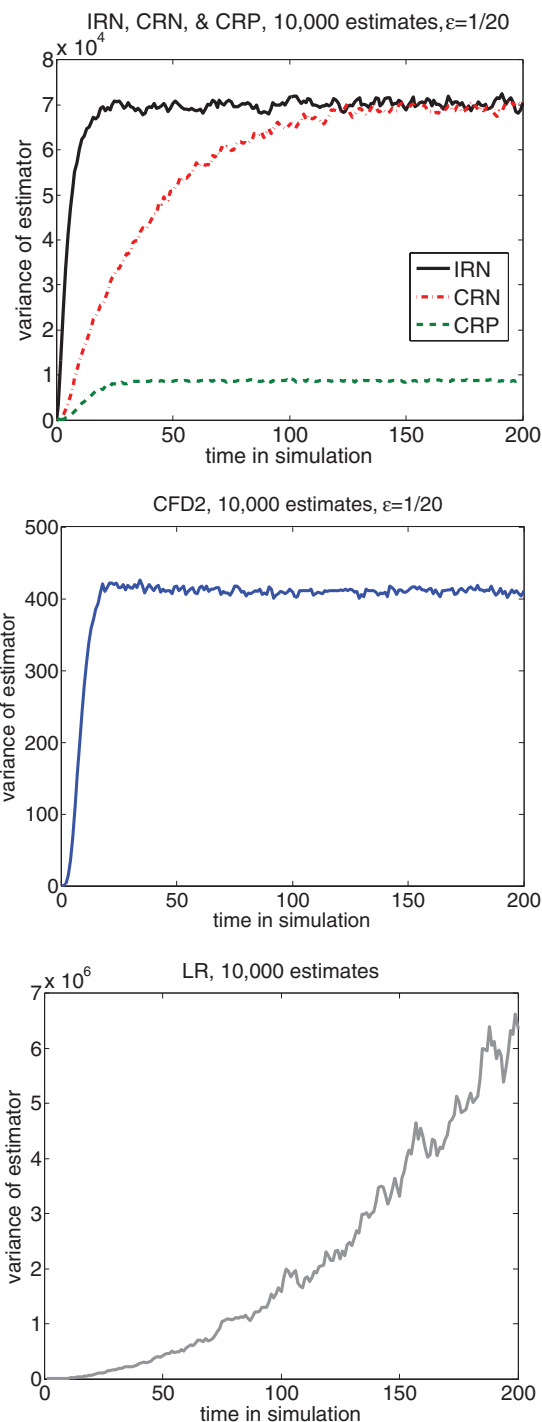| Method | Estimates | Approximation | No. of updates | CPU time (s) |
|---|---|---|---|---|
| LR | 495 000 | $2506 \pm 120$ | $\approx 2.1 \times 10^9$ | 6619 |
| IRN | 190 000 | $2617 \pm 120$ | $\approx 2.4 \times 10^9$ | 7657 |
| CRN | 98 100 | $2572 \pm 120$ | $\approx 2.6 \times 10^8$ | 4489 |
| CRP | 22 200 | $2532 \pm 120$ | $\approx 2.8 \times 10^8$ | 1533 |
| CFD2 | 1150 | $2565 \pm 120$ | $\approx 5.8 \times 10^6$ | 61 |



FIG. 1. Variance versus time of the estimators of the five different methods, applied to the calculation of $\frac{\partial^2}{\partial \theta^2} \mathbb{E} X_P(\theta, t)$ in the mRNA transcription model of Subsection IV B 1. Note that the scales are vastly different.

### 2. Mixed partial derivative of protein abundance

We compare the five methods in the estimation of $\frac{\partial^2}{\partial \gamma \partial \theta} \mathbb{E} X_P(30)$ at $\theta = 1/4$ and $\gamma = 10$, which can be calculated exactly to be $-31.8$. Table V shows the approximations and computational complexity of these methods using 5000 estimates.

Note that in this example, the LR method outperforms all methods, except CFD2, with respect to computation time.

TABLE V. Confidence intervals (95%) and computational complexity for all five methods, after 5000 estimates, for the computation of the mixed partial derivative in the mRNA transcription model as in Subsection IV B 2. An $\epsilon$ of 1/25 was used for the finite difference methods. Additionally, results from the LR method with CPU time approximately that of CFD2 are included for comparison. Actual value: $-31.8$.

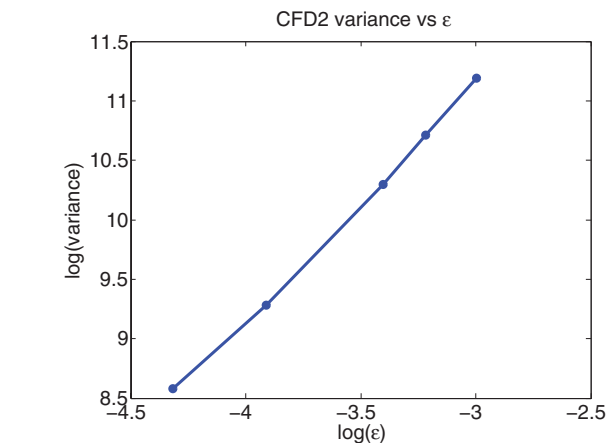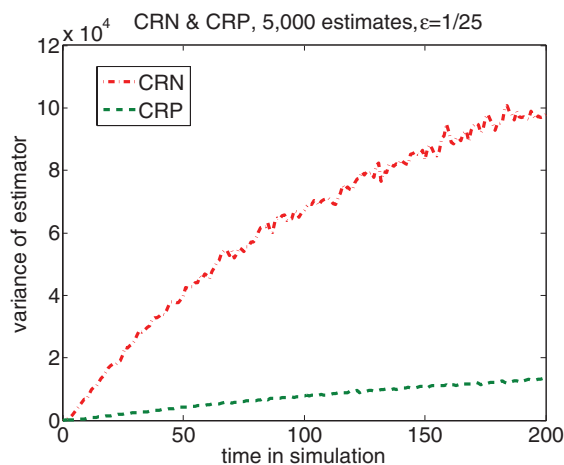| Method | Estimates | Approximation | No. of updates | CPU time (s) |
|---|---|---|---|---|
| IRN | 5000 | $607 \pm 923$ | $\approx 8.43 \times 10^7$ | 264 |
| CRN | 5000 | $191.5 \pm 330$ | $\approx 8.42 \times 10^7$ | 273 |
| CRP | 5000 | $21.0 \pm 96$ | $\approx 8.41 \times 10^7$ | 365 |
| CFD2 | 5000 | $-33.8 \pm 4$ | $\approx 2.25 \times 10^7$ | 238 |
| LR | 5000 | $-15.4 \pm 113$ | $\approx 2.10 \times 10^7$ | 73 |
| LR | 17 000 | $-61.8 \pm 68$ | $\approx 6.72 \times 10^7$ | 234 |



FIG. 3. This log-log plot of variance versus epsilon (100 000 estimates) for the mRNA transcription model computation of Subsection IV B 3 suggests that the CFD2 method gives an estimator of $O(\epsilon^{-2})$ even though the function $f$ of the system state is nonlinear: the slope of the best fit line is 1.98.

Thus, for comparison, we have also included the results of a test using the LR method in which the CPU time is approximately the same as CFD2; note that the confidence interval for the CFD2 method is much smaller. Figure 2 shows variance plots of the CRN and CRP, and CFD2 methods over time in simulation.

### 3. Second derivative of the square of protein abundance with respect to $\theta$

We also calculate, from the mRNA transcription model of Example 4.1, $\frac{\partial^2}{\partial \theta^2}\mathbb{E}(X_P(t)^2)$ at $t = 5$ and $\theta = \frac{1}{4}$, with $\gamma = 10$ and $X_0 = 0$. Note here we are considering a function $f$ of the state space which is nonlinear.

In Figure 3, we plot the log of the variance of the numerator of the estimator (6) versus the log of epsilon. Since we expect, for the double coupled CFD2 method, that this variance $V(\epsilon)$ should scale like $C\epsilon^p$ for some constants $C$ and $p$, we see that the slope of $\log(V(\epsilon)) = \log(C) + p \log(\epsilon)$ from our simulations will suggest the value of $p$. This plot suggests that $p = 2$; since the numerator of the estimator is then divided by $\epsilon^2$ in $d(\epsilon)$, this suggests a final variance of $O(R^{-1}\epsilon^{-2})$ for the estimator (5) as discussed in Sec. II.

For comparison, the slope of this log-log plot for the IRN method is zero, as the variance of the numerator does not depend on epsilon, giving a final variance of $O(R^{-1}\epsilon^{-4})$. The slopes for the associated log-log plots for the CRN and CRP estimators will vary with time (discussed further in Subsection IV D 1).

The general behavior of the variances over time for the IRN, CRN, and CRP methods can be seen in Figure 4.

### C. Quadratic decay

In order to demonstrate that the $O(\epsilon^2)$ convergence rate seen in the previous examples does not universally hold, we consider a pure decay process of a population $X_t$, so that the sole reaction has $\zeta = -1$ and quadratic propensity $\lambda(\theta, X_t(\theta)) = \theta X_t(\theta)(X_t(\theta) - 1)$, and calculate $\frac{\partial^2 \mathbb{E}X_t(\theta)}{\partial \theta^2}$ with $\theta = 1$ and with initial population $X_0(\theta) = 2000$. Figure 5 gives a log-log plot of variance versus epsilon at time 0.001. Since it suggests $p = 1$, this demonstrates that, in this case, the double coupled method provides only $O(R^{-1}\epsilon^{-3})$ convergence as discussed in Sec. II, showing that rate to be sharp.
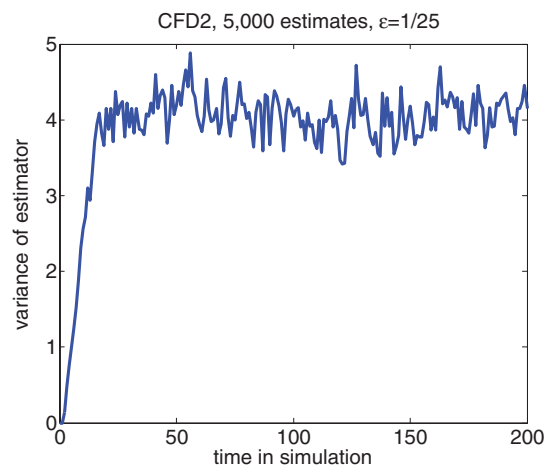


FIG. 2. Plots of variance over time for the CRN and CRP methods, and the CFD2 method, in computing the mixed partial derivative of the mRNA transcription model of Subsection IV B 2. Note the very different scales. For comparison, the IRN method plateaus at a variance of approximately $5 \times 10^6$.
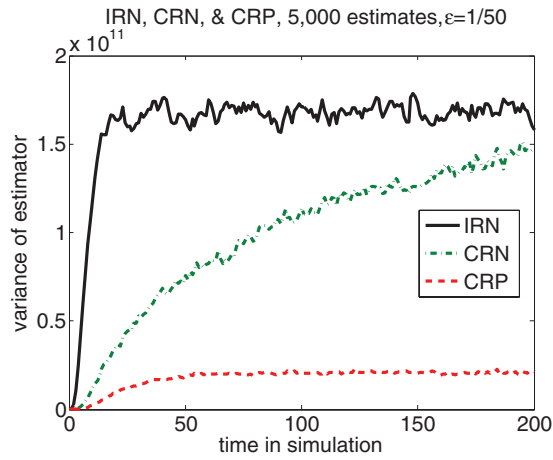
FIG. 4. Plot of variance over time for 5000 estimates of the IRN, CRN, and CRP methods for the mRNA transcription model computation of Subsection IV B 3. The variance of the CFD2 method is too small to be seen at this scale; at time 200 it is approximately $3.5 \times 10^8$.

As demonstrated in Table VI and in Figure 6, however, the double coupled method is still significantly more efficient than existing methods on this model.

## D. Genetic toggle switch

Finally, we consider a model of a genetic toggle switch that also appeared in Refs. 20 and 10,

$$\emptyset \underset{1}{\overset{\lambda_1}{\rightleftarrows}} A, \qquad \emptyset \underset{1}{\overset{\lambda_2}{\rightleftarrows}} B,$$
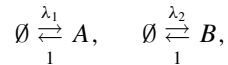
where

$$\lambda_1(t) = \frac{b}{1 + X_B(t)^\beta} \quad \text{and} \quad \lambda_2(t) = \frac{a}{1 + X_A(t)^\alpha},$$
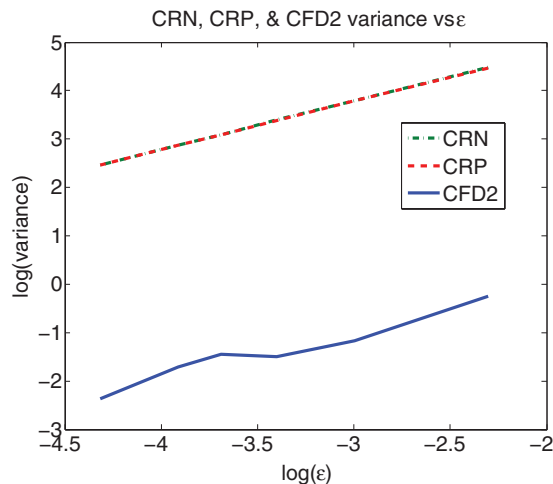


FIG. 5. This log plot of variance versus epsilon (each point computed to the first of 300 000 estimates or a confidence of $\pm 10$) for the decay model of Subsection IV C suggests that the CFD2 method gives an estimator of only $O(\epsilon^{-3})$: the slope of the best fit line is approximately 0.97. While the CRN and CRP methods also give an estimator of this same rate (the slope of the best fit lines are both $\approx 0.99$, and in fact the lines are on top of each other), the variance of the estimates from CRN and CRP are significantly higher than those from the CFD2 method, as can be seen by the wide gap between the above curves.

TABLE VI. Estimates, $\epsilon$ used, and updates and computational time needed for the given 95% confidence intervals for all methods for $\frac{\partial^2 \mathbb{E} X_t(\theta)}{\partial \theta^2}$ at $t = 0.001$ for the quadratic decay model of Subsection IV C. The upper half of the table shows the relevant results after the simulation of 10 000 estimates. The lower half of the table shows the results of simulations run until the estimate had a confidence interval of a desired width. The IRN and LR methods were unable to achieve these precisions due to memory constraints. Note again the equivalence of the CRN and CRP methods on a single reaction model.

| Method | $\epsilon$ | Estimates | Approximation | No. of updates | CPU time (s) |
|--------|------------|-----------|---------------|----------------|--------------|
| LR | n/a | 10 000 | $1240 \pm 1070$ | $\approx 1.3 \times 10^7$ | 9 |
| IRN | 1/20 | 10 000 | $555 \pm 218$ | $\approx 4.8 \times 10^7$ | 35 |
| CRN | 1/20 | 10 000 | $585 \pm 52$ | $\approx 4.0 \times 10^7$ | 30 |
| CRP | 1/20 | 10 000 | $584 \pm 52$ | $\approx 4.0 \times 10^7$ | 30 |
| CFD2 | 1/20 | 10 000 | $592 \pm 5$ | $\approx 1.4 \times 10^7$ | 90 |
| CRN | 1/20 | 272 000 | $588 \pm 10$ | $\approx 1.1 \times 10^9$ | 813 |
| CRP | 1/20 | 271 000 | $589 \pm 10$ | $\approx 1.1 \times 10^9$ | 862 |
| CFD2 | 1/20 | 1950 | $592 \pm 10$ | $\approx 2.7 \times 10^6$ | 17 |
| CRN | 1/50 | 169 500 | $543 \pm 50$ | $\approx 6.8 \times 10^8$ | 511 |
| CRP | 1/50 | 169 000 | $515 \pm 50$ | $\approx 6.8 \times 10^8$ | 510 |
| CFD2 | 1/50 | 1800 | $605 \pm 50$ | $\approx 2.5 \times 10^6$ | 16 |

and where $X_A(t)$ and $X_B(t)$ denote the number of gene products from two interacting genes. Note that each gene product inhibits the growth of the other.

We take parameter values of $b = 50$, $\beta = 2.5$, $a = 16$ and will differentiate with respect to $\alpha$. Note that this model does *not* follow mass action kinetics, or have linear propensities. In Subsection IV D 1, we consider a second derivative of $\mathbb{E} X_B$ at a fixed time, while in Subsection IV D 2, we consider a second derivative of the expected *time average* of $X_A$ up to a given time, which is a functional of the path of $X_A$ rather than simply $X_A$ at some terminal time.

### 1. Second derivative of abundance of B with respect to $\alpha$

We estimate $\frac{\partial^2 \mathbb{E} X_B(\alpha,t)}{\partial \alpha^2}$ at $\alpha = 1$ and at two times, 5 and 400. In Figure 7, we plot the log of the variance of the numerator of the estimator (6), using CFD2, versus the log of the perturbation epsilon. As in Subsection IV B 2, the plot clearly suggests that $p = 2$. We also plot the same quantity using CRP and CRN. These slopes, on the other hand, vary with time. For small times both slopes are close to one, but as time increases the slopes decrease, until, for very large times, they are close to zero. This corresponds with the fact that for large times the variances of the CRP and CRN estimates converge to values on the order of the IRN estimate variance, which, as previously noted, is independent of the value of epsilon. The general behavior of the variances over time can be seen in Figure 8, where it is seen that CFD2 has a variance that is 16 times lower than CRN and 36 times lower than CRP. Further, we note that for this model the CRP method outperforms the CRN method for small times, while for larger times CRN outperforms CRP.
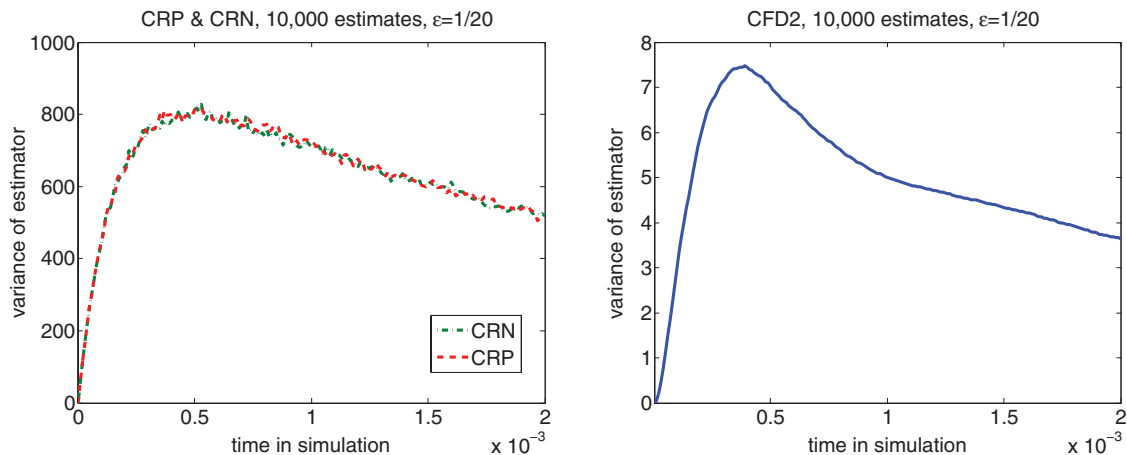
FIG. 6. The behavior over time of the variance of the estimates of the CRN, CRP, and CFD2 methods on the quadratic decay model of Subsection IV C. Note that the variance for the CFD2 method is 100 times smaller than the other two methods, which, as expected, act the same on this model. An $\epsilon$ of 1/20 was used and 10 000 estimates were run. The plot of the IRN variance is similar in shape but with a peak variance of $3.1 \times 10^4$.
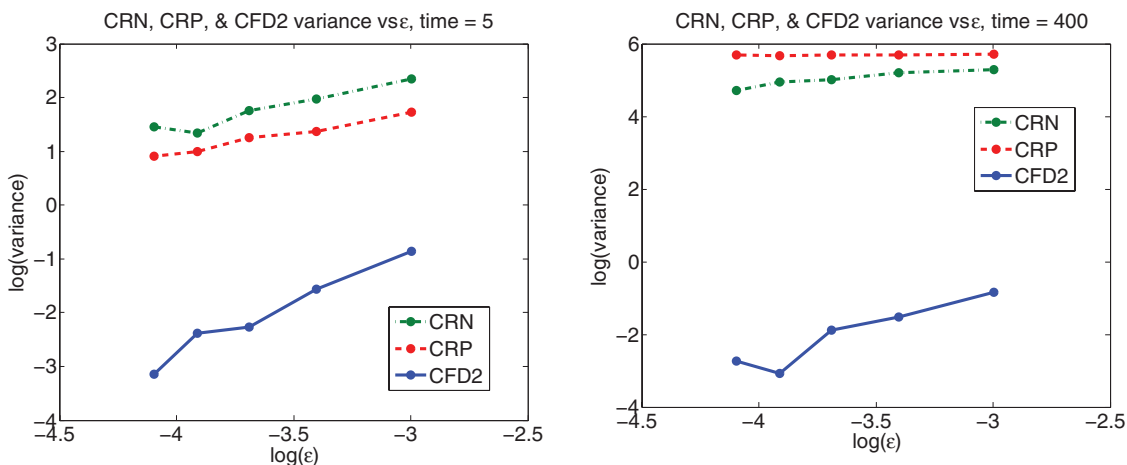


FIG. 7. These log-log plots (25 000 estimates) of variance versus epsilon for computation on the gene toggle model as in Subsection IV D 1, at two different times, suggest that the double coupled method gives an estimator of $O(\epsilon^{-2})$ even though two of the intensities are nonlinear: the slope of the best fit line for the CFD2 method is approximately 2 (=1.97) at both times. The slope for the CRP and CRN methods, on the other hand, are approximately 0.74 and 0.90 respectively at time 5, but are only around 0.03 and 0.49 at time 400.
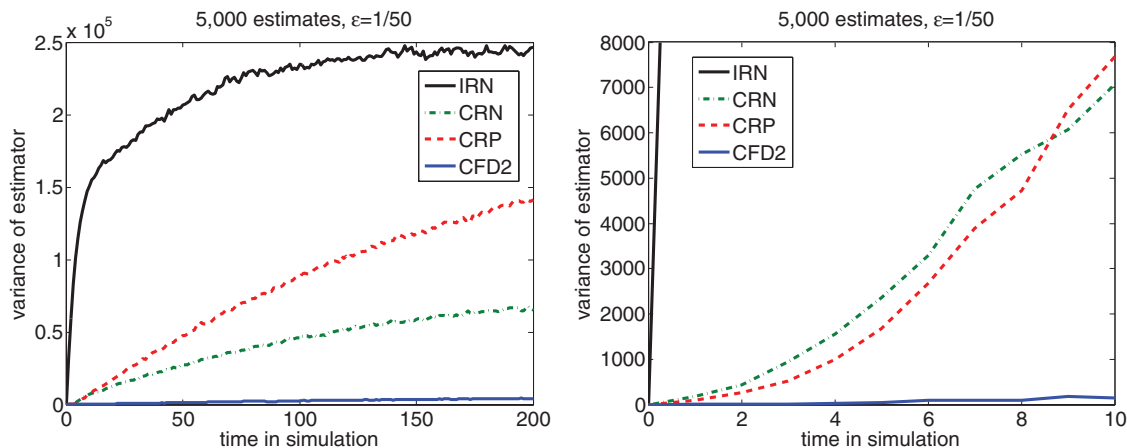


FIG. 8. Plots of variance over time for 10 000 estimates of the finite difference methods for the gene toggle model as in Subsection IV D 1; the figure at left includes time up to 200, while the figure at right provides a close-up view of the plot for times less than 10. The value of the CFD2 variance at time 200 is approximately 4000, while the CRN variance is approximately 65 000.

TABLE VII. Confidence intervals (95%) and computational complexity for each of the methods, after 100 000 estimates, for the time average computation at $t = 30$ on the gene toggle model of Subsection IV D 2. An $\epsilon$ of 1/50 was used.

| Method | Estimates | Approximation | CPU time (s) |
|--------|-----------|---------------|--------------|
| IRN    | 100 000   | $-13.8 \pm 621$ | 2240 |
| CRN    | 100 000   | $-274 \pm 146$  | 1441 |
| CRP    | 100 000   | $-215 \pm 107$  | 3035 |
| CFD2   | 100 000   | $-222 \pm 26$   | 2722 |

### 2. Second derivative of time average of abundance of A with respect to α

Finally, while this was not discussed in the paper, we include an example computing a sensitivity of a path functional. That is, the quantity we wish to study is a function of the path of the process $X(s)$ for $s \leq t$, rather than just the terminal value $X(t)$. The only difference in implementation is the need to compute this quantity during the simulation of the path (or

to store the path for the computation after its simulation). Table VII shows the estimates of $\frac{\partial^2}{\partial \alpha^2} \mathbb{E}\, t^{-1} \int_0^t X_A(s)ds$ at $t = 30$ using the various finite difference methods, demonstrating the advantage of the double coupled method for these path functional quantities as well. Additionally, Figure 9 shows that the overall behavior of the variances of the three finite difference methods remains the same as in the previous examples.

## V. CONCLUSIONS AND FUTURE WORK

We have introduced a new, efficient method for the computation of second derivative sensitivities for discrete biochemical reaction networks. Through several numerical examples we have demonstrated its advantage over existing methods, both in simple scenarios and in more realistic systems, including several examples in which the system contained nonlinear propensities, or in which the relevant quantity to be studied involved a nonlinear function $f$ or even a path functional of the system state. Future work will include proving analytical bounds on the variance of the estimator given by the new method and exploring conditions in which a better convergence rate is achieved, as well as finding efficient algorithms to simultaneously compute all of the second order sensitivities of models with a large number of parameters. Another avenue of future work will involve incorporating algorithms for the computation of second derivatives into the optimization methods discussed in the introduction in the context of parameter estimation.
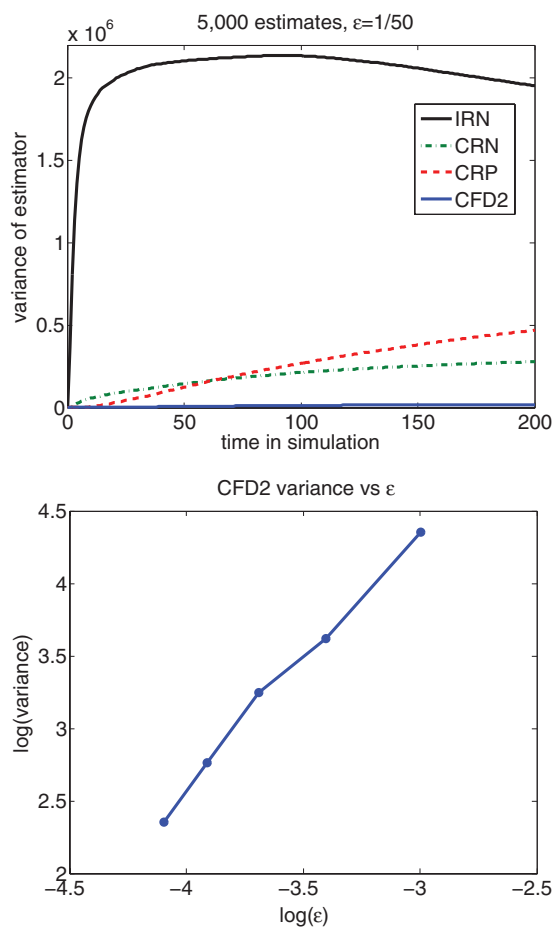
FIG. 9. At top, a plot of variance over time for 5000 estimates for the finite difference methods for the path functional computation on the gene toggle model in Subsection IV D 2. The value of the CFD2 variance at time 200 is approximately 19 000, while the CRN variance is approximately 282 000. At bottom, a log-log plot (2000 estimates) of variance versus epsilon for this computation suggests that the double coupled method gives an estimator converging faster than $O(\epsilon^{-3})$ in this computation as well: the slope of the best fit line for the CFD2 method is approximately 1.78.

[1]D. T. Gillespie, "A general method for numerically simulating the stochastic time evolution of coupled chemical reactions," J. Comput. Phys. **22**, 403 (1976).

[2]D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," J. Phys. Chem. **81**, 2340 (1977).

[3]M. Gibson and J. Bruck, "Efficient exact stochastic simulation of chemical systems with many species and many channels," J. Phys. Chem. A **105**, 1876 (2000).

[4]D. F. Anderson, "A modified next reaction method for simulating chemical systems with time dependent propensities and delays," J. Chem. Phys. **127**, 214107 (2007).

[5]M. Komorowski, M. J. Costa, D. A. Rand, and M. P. H. Stumpf, "Sensitivity, robustness, and identifiability in stochastic chemical kinetics models," Proc. Natl. Acad. Sci. U.S.A. **108**, 8645 (2011).

[6]S. Asmussen and P. W. Glynn, *Stochastic Simulation: Algorithms and Analysis* (Springer, 2007).

[7]J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed. (Springer, New York, 2006).

[8]J. Rawlings and J. Ekerdt, *Chemical Reactor Analysis and Design Fundamentals* (Nob Hill Publishing, 2002).

[9]J. Spall, "Feedback and weighting mechanisms for improving Jacobian estimates in the adaptive simultaneous perturbation algorithm," IEEE Trans. Autom. Control **54**, 1216 (2009).

[10]D. F. Anderson, "An efficient finite difference method for parameter sensitivities of continuous time Markov chains," SIAM J. Numer. Anal. **50**, 2237 (2012).

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

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

[13]D. F. Anderson and T. G. Kurtz, "Continuous time Markov chain models for chemical reaction networks," in *Design and Analysis of Biomolecular Circuits: Engineering Approaches to Systems and Synthetic Biology*, edited by H. Koeppl *et al.* (Springer, 2011), pp. 3–42.

[14]The gaps between points are unit exponential random variables.

[15]T. G. Kurtz, "Representation and approximation of counting processes," in *Advances in Filtering and Optimal Stochastic Control*, Lecture Notes in Control and Information Sciences Vol. 42 (Springer, Berlin, 1982), pp. 177–191.

[16]S. Plyasunov and A. P. Arkin, "Efficient stochastic sensitivity analysis of discrete event systems," J. Comp. Phys. **221**, 724 (2007).

[17]P. Sheppard, M. Rathinam, and M. Khammash, "A pathwise derivative approach to the computation of parameter sensitivities in discrete stochastic chemical systems," J. Chem. Phys. **136**, 034115 (2012).

[18]P. Glasserman, *Gradient Estimation Via Perturbation Analysis* (Kluwer Academic, 1991).

[19]We do not provide a full list of the less efficient couplings, of which there are many.

[20]M. Rathinam, P. W. Sheppard, and M. Khammash, "Efficient computation of parameter sensitivities of discrete stochastic chemical reaction networks," J. Chem. Phys. **132**, 034103 (2010).

[21]This is the usual generation of a discrete random variable found in every instance of Gillespie's algorithm.

[22]At each step, the first random number determines the time of the next reaction and the second determines which occurs; the reactions were listed in a fixed order as given in this paper.