Look before you leap: A confidence-based method for selecting species criticality while avoiding negative populations in τ-leaping

Christian A. Yates[1,a)] and Kevin Burrage[2,b)]
[1]*Centre for Mathematical Biology, Mathematical Institute, University of Oxford, 24-29 St Giles', Oxford OX1 3LB, United Kingdom*
[2]*Computing Laboratory and Oxford Centre for Integrative Systems Biology, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom and Mathematics Department, QUT, Brisbane, 4001, Australia*

(Received 20 September 2010; accepted 24 January 2011; published online 25 February 2011)

The stochastic simulation algorithm was introduced by Gillespie and in a different form by Kurtz. There have been many attempts at accelerating the algorithm without deviating from the behavior of the simulated system. The crux of the explicit τ-leaping procedure is the use of Poisson random variables to approximate the number of occurrences of each type of reaction event during a carefully selected time period, τ. This method is acceptable providing the leap condition, that no propensity function changes "significantly" during any time-step, is met. Using this method there is a possibility that species numbers can, artificially, become negative. Several recent papers have demonstrated methods that avoid this situation. One such method classifies, as critical, those reactions in danger of sending species populations negative. At most, one of these critical reactions is allowed to occur in the next time-step. We argue that the criticality of a reactant species and its dependent reaction channels should be related to the probability of the species number becoming negative. This way only reactions that, if fired, produce a high probability of driving a reactant population negative are labeled critical. The number of firings of more reaction channels can be approximated using Poisson random variables thus speeding up the simulation while maintaining the accuracy. In implementing this revised method of criticality selection we make use of the probability distribution from which the random variable describing the change in species number is drawn. We give several numerical examples to demonstrate the effectiveness of our new method. © *2011 American Institute of Physics*. [doi:10.1063/1.3554385]

I. INTRODUCTION

Many biological processes are inherently stochastic and, in recent years, the modeling of such mechanisms has become a particularly important field. These stochastic processes take place over widely varying spatial scales, from interactions between organisms[1] on the macroscale through cell migration[2] on the mesoscale right down to gene regulatory networks[3] on the microscale, with a myriad of other subcellular and supercellular processes in between.[4] These systems are modeled through a series of interactions, be they locusts sensing each other and reacting to each others' movements, cells migrating from one area of a domain to another, or molecules reacting with each other. For consistency throughout the rest of this paper, we will use the terminology of a chemical system, referring to these interactions as "reactions" and to interacting groups as "species," although the algorithms outlined here could equally well be applied to systems that do not represent chemical reactions. In many cases the "reactive collisions" associated with these "chemical species" can be modeled using the stochastic simulation algorithm (SSA).

Introduced by Gillespie[5,6] (and in a different form by Kurtz[7]), the SSA can be used to simulate a well-stirred system of $N \geq 1$ chemical species, $\{S_1, \ldots, S_N\}$, interacting stochastically through M reaction channels, $\{R_1, \ldots, R_M\}$. We denote the state of the system at time t by the vector, $\boldsymbol{X}(t) = (X_1(t), \ldots, X_N(t))$, where $X_i(t)$ represents the number of molecules of S_i at time t. We aim to simulate the evolution of $\boldsymbol{X}(t)$ given some initial condition, $\boldsymbol{X}(t_0) = \boldsymbol{x_0}$. The dynamics of each reaction channel, R_j, (for $j = 1, \ldots, M$) are completely characterized by a propensity function, $a_j(\boldsymbol{x})$, and a stoichiometric vector, $\boldsymbol{\nu}_j = (\nu_{1j}, \ldots, \nu_{Nj})$. The propensity functions are such that $a_j(\boldsymbol{x})dt$ defines the probability that, given $\boldsymbol{X}(t) = \boldsymbol{x}$, one R_j reaction will "fire" during the next infinitesimal time interval, dt. ν_{ij} represents the change in the population of S_i caused by one "firing" of the R_j reaction channel. For each reaction channel, R_j, the state change vector, $\boldsymbol{\nu}_j$, is defined once at the beginning of simulation, but the propensity functions will usually change dynamically with the state vector, $\boldsymbol{X}(t)$, and thus require updates at each time-step.

One implementation of this mathematically exact procedure for simulating the progression of a system is known as the direct method.[6] Given a system at time, t, in state, $\boldsymbol{X}(t)$, a time interval, τ, until the next reaction occurs, is generated. Along with it a reaction, with index, j, is chosen to occur at time, $t + \tau$. The changes in the numbers of molecules caused

a)Electronic mail: yatesc@maths.ox.ac.uk. URL: http://people.maths.ox.ac.uk/yatesc/.
b)Electronic mail: kevin.burrage@gmail.com. URL: http://www.comlab.ox.ac.uk/people/kevin.burrage/.

by the firing of the reaction channel, R_j, are implemented, the propensity functions are altered accordingly and the time is updated, ready for the next (τ, j) pair to be selected. A method for the implementation of this algorithm is given by Gillespie.[6]

The SSA records the exact history of a system, stepping between individual reaction events in the finest detail. This approach is simultaneously the Herculean strength and the Achilles' heel of the SSA. The construction of every reaction event gives a complete history of $\boldsymbol{X}(t)$, but considering every reaction is time consuming and computationally expensive.

If some losses in accuracy are acceptable then a faster, but approximate, method, the explicit τ-leaping algorithm,[8] is available. Instead of stepping gingerly between individual reactions, the accelerated method leaps temeritously through time, allowing several reaction events from each channel to occur between updates of the state vector. This accelerated method is not, however, completely reckless as it is constrained by the so-called "leap condition."[8] This states that all leaps should be sufficiently small that the change in the state vector will be so slight that no propensity functions have their values changed significantly during a leap. The number of firings, k_j, of each reaction channel, R_j, during a leap can be approximated using samples of the Poisson random variable, $\mathcal{P}(a_j(\boldsymbol{x})\tau)$, for sufficiently small τ. This provides us with a formula by which to update the state vector,

$$\boldsymbol{X}(t+\tau) = \boldsymbol{x} + \sum_{j=1}^{M} \boldsymbol{\nu}_j \mathcal{P}(a_j(\boldsymbol{x})\tau). \tag{1}$$

The accuracy of this approximation depends on the extent to which the leap condition is observed. If propensity functions change appreciably during the leap then we will be simulating reactions from the wrong Poisson distribution. What exactly constitutes an appreciable change in a propensity function and consequently how to choose τ is the subject of much debate.[8–11] The original[8] method bounds the change in each propensity function by $\varepsilon a_0(\boldsymbol{x})$, where $0 < \varepsilon \ll 1$ is known as the error control parameter. Throughout the rest of this paper and in all numerical simulations we fix $\varepsilon = 0.03$, a canonical value used by Cao *et al.*[10,12] This bound on the change in propensity function gives a method of bounding τ and completes the description of the basic explicit τ-leaping algorithm (see Cao *et al.*[6] and Gillespie[8,10]).

Since the Poisson random variable can have arbitrarily large sample values, it is possible that a reaction channel may fire so many times that it will deplete one or more of its reactants by a larger number than are actually available. One method of preventing this unphysical behavior, presented by Tian and Burrage[13] and independently by Chatterjee *et al.*[14] is known as binomial τ-leaping. This relies on approximating the number of firings, k_j, of a reaction channel, R_j, by a sample from a binomial distribution whose values are naturally bounded. Due to the poor correspondence between the Poisson and binomial distributions in the case of small species numbers, x_i, (the case with which we will most often be concerned when attempting to avoid negative populations) and overcautious restrictions on the number of reactions that can fire, the binomial τ-leap method tends to be less accurate than the SSA and the more recent "modified Poisson τ-leaping" procedure,[12] both of which circumvent the problem of negative species.

The modified Poisson τ-leaping procedure relies on identifying reactions that, if fired more than a predefined "critical" number of times, will cause one or more of their contingent species populations to become negative. The number of "allowed" firings of a reaction, R_j, will be denoted by L_j. Reactions are partitioned into critical and noncritical subsets with, at most, one of the critical reactions being able to fire per leap, by a process akin to the exact SSA. This ensures that the reactant species depleted by the critical reaction do not become negative. The more critical reactions we have, the slower the simulation becomes. Defining critical reactions should, therefore, be an important part of any algorithm. However, the size of the critical number of firings, n_c, is difficult to determine and Cao *et al.*[12] suggest a value between 2 and 20.

Widely varying reaction rate constants can cause reaction channels to have very different propensity functions and hence probabilities of driving the species population negative. Even when reactions have the same reaction rate constants, differing numbers of molecules undergoing the same reaction will have widely varying probabilities of becoming negative. If we choose n_c to be small then we run the unacceptable risk of ignoring reactions that have a high probability of their species populations becoming negative. However, choosing n_c to be too large risks classifying reactions, which have near-infinitesimal probabilities of becoming negative, as critical, slowing the simulation. Including rates of reaction that may vary over orders of magnitude and the fact that species numbers may be repopulated during a reaction complicates matters still further. A method for deciding the criticality of a reaction based on the *probability* of any one of its *contingent species* becoming negative would be preferable.

Using estimations of the current propensity functions, we derive the precise probability distribution for $\Delta_\tau X_i$, which describes the change in reactant species, i, as the sums and/or differences of Poisson m-lets.[15] Using this distribution we choose a confidence level and determine whether the probability (that the sample drawn from the random variable will be larger than the number of reactants remaining) falls outside this interval. Species for which the probability of becoming negative is high enough to fall outside the confidence interval will be marked as critical as will the reactions that depend upon them. This method will give us much more controlled consistency when deciding reaction criticality. The speed of the algorithm should be increased as we will no longer mark reactions that do not have a significant probability of driving one of their contingent species negative as critical.

The remainder of this paper will be structured as follows: In Sec. II, we briefly outline the modified τ-leaping procedure of Cao *et al.*[12] which is the current gold standard for avoiding negative populations. We also explicitly describe the method of Cao *et al.*[10] for efficient step size selection. In Sec. III, we introduce a revised method which increases confidence in our choice of reaction criticality. Numerical simulations that demonstrate the improved efficiency and accuracy of our

method of criticality selection will be performed in Sec. IV and the implications of our adapted techniques will be discussed in Sec. V.

II. BACKGROUND

A. Some useful results from binomial τ-leaping

We have already informally introduced L_j as the number of firings of a reaction, R_j, before it sends one of its contingent species negative, providing no new molecules of that reactant species are created. More formally this has been expressed as[10,12]

$$L_j = \min_{i=1,\ldots,N;\nu_{ij}<0} \left\lfloor \frac{x_i}{-\nu_{ij}} \right\rfloor, \tag{2}$$

where $\lfloor x \rfloor$ represents the smallest integer in x.

Tian and Burrage[13] introduce Property 1:

Property 1. If $\mathcal{P}_1 = \mathcal{P}(\lambda_1)$ and $\mathcal{P}_2 = \mathcal{P}(\lambda_2)$ are two independent Poisson random variables with means λ_1 and λ_2, respectively, then $\mathcal{P}_1 + \mathcal{P}_2$ is also a Poisson random variable, $\mathcal{P}(\lambda_1 + \lambda_2)$, with mean $\lambda_1 + \lambda_2$.

This property can be found in any basic text on univariate discrete distributions, Johnson *et al.*[15] for example. We also introduce two new properties employed in Sec. III.

Property 2. If $\mathcal{P}_1 = \mathcal{P}(\lambda_1)$ and $\mathcal{P}_2 = \mathcal{P}(\lambda_2)$ are two independent Poisson random variables with means λ_1 and λ_2, respectively, then their difference $\mathcal{P}_1 - \mathcal{P}_2$ is given by a Skellam distribution, with mean $\lambda_1 - \lambda_2$ and variance $\lambda_1 + \lambda_2$. The probability mass function (PMF) of the Skellam distribution takes the form:

$$\Pr(k; \lambda_1, \lambda_2) = e^{-(\lambda_1+\lambda_2)} \left(\frac{\lambda_1}{\lambda_2}\right)^{k/2} I_{|k|}(2\sqrt{\lambda_1\lambda_2}), \tag{3}$$

where $I_k(z)$ is the modified Bessel function of the first kind.

The proof of Property 2 is given in Appendix A. Using Property 1 recursively in combination with Property 2, we can also show that the sum and/or difference of arbitrarily many Poisson distributions is also Skellam distributed.

If $\mathcal{P}^m$ is Poisson distributed but with support $0, m, 2m, 3m \ldots$, for $m \in \mathbb{Z}$, instead of integer support, then we call it a Poisson m-let, i.e., $m = 1$ gives the usual Poisson distribution or Poisson singlet, $m = 2$ gives a Poisson doublet etc.

Property 3. If $\mathcal{P}^m = m \times \mathcal{P}(\lambda_1)$ and $\mathcal{P}^n = n \times \mathcal{P}(\lambda_2)$ are a Poisson m-let and a Poisson n-let, respectively, where $m, n \in \mathbb{Z}$, then their sum $\mathcal{P} = \mathcal{P}^m + \mathcal{P}^n$ is distributed as a Poisson stopped sum distribution.

The proof of Property 3 is given in Appendix B. Since in the statement of Property 3 we assume $m, n \in \mathbb{Z}$, the formulae apply equally well to negative values of m and n as to positive values, and hence, to differences as well as sums of Poisson m-lets. It is also a trivial extension to prove that the sum or difference of Poisson stopped sum distributions is also a Poisson stopped sum distribution. From the individual Poisson m-lets we can find the probability generating function (PGF) of the Poisson stopped sum distribution and from the PGF we can find the PMF, and hence, the cumulative mass function (CMF).

B. Modified Poisson τ-leaping

It is possible, in the original formulation of the τ-leaping algorithm, that the Poisson approximation to k_j in Eq. (1) might result in so many firings of a reaction, R_j, that the population of a contingent species becomes negative.

The modified Poisson τ-leaping algorithm[12] arose from the observation that, typically, negative populations are caused by multiple firings of reactions that are only a few firings away from exhausting the numbers of one or more of their reactants. The algorithm introduces a second control parameter, n_c, a positive integer, usually set between 2 and 20, such that any reaction within n_c firings of exhausting one of its contingent species is classified as critical. We denote this method of criticality-selection as the CGP method. This method selects τ so that, at most, one firing occurs between all the critical reactions, using an adapted version of the SSA. This restriction means that negative populations are almost completely avoided. However, because Poisson random variables are still being used, there is still a small possibility (due to the noncritical reactions) of negative species populations occurring. On these rare occasions a leap can simply be rejected and repeated with a reduced τ.[12] An algorithm for the modified τ-leaping method is given in Cao *et al.*[12]

C. CGP method for τ-selection

The CGP τ-selection procedure attempts to choose τ so that leaps are as large as possible. It is also less computationally intensive than previous methods.[8,12] This, in combination with larger leap sizes, causes simulation speeds to increase. The CGP method's key idea is to bound the *relative* change in *species numbers* explicitly. By bounding the expected change in species numbers the method implicitly bounds the expected change in propensity functions in such a manner that the leap condition is still approximately preserved.

In the τ-selection method of Gillespie[8] the leap condition is realized by bounding the absolute value of the change in each propensity function, $\Delta_\tau a_j(\boldsymbol{x})$, in time-step, τ, by a small fraction, ε, of the sum of the propensity functions, $a_0(\boldsymbol{x})$. This does not limit the changes of the propensity functions in a particularly uniform manner.[10] The aim of the leap condition is to ensure that every propensity function remains practically constant during a leap. This can be achieved more consistently if we bound the relative changes in propensity functions as follows:

$$|\Delta_\tau a_j(\boldsymbol{x})| \leq \max\{\varepsilon a_j(\boldsymbol{x}), c_j\}, \quad j = 1, \ldots, M, \tag{4}$$

where c_j is the minimum amount by which the propensity function, $a_j(\boldsymbol{x})$, for the reaction, R_j, can change. This second argument of the maximization eliminates the probability of τ tending to zero as the size of a single propensity function tends to zero.[10]

The relative changes in the propensity functions can be bounded approximately by bounding the relative changes in their corresponding reactions' contingent species. Instead of choosing τ using condition (4) we can use

$$|\Delta_\tau X_i| \leq \max\{\varepsilon_i x_i, 1\}. \tag{5}$$

Here the second argument of the maximization [in analogy to condition (4)], the smallest amount by which a species number can change, is used to avoid τ approaching zero. For each reactant species, S_i, we find the highest order reaction in which S_i is a reactant and choose ε_i appropriately according to the rules given in Ref. 10. Considering a rearrangement of Eq. (1):

$$\Delta_\tau X_i = \sum_{j \in J_{\text{ncr}}} \nu_{ij} \mathcal{P}_j(a_j(\boldsymbol{x})\tau), \tag{6}$$

where J_{ncr} is the set of noncritical reactions, we can bound $\Delta_\tau X_i$ according to condition (5) by choosing an appropriate value of τ. The interpretation of Eq. (6) will determine how stringently the leap condition is adhered to.

The CGP τ-selection method uses the mean and variance of the distribution of $\Delta_\tau X_i$ given by Eq. (6) to place a bound on τ as follows. The mean and variance can be calculated straightforwardly as

$$\langle \Delta_\tau X_i \rangle = \sum_{j \in J_{\text{ncr}}} \nu_{ij} a_j(\boldsymbol{x})\tau, \tag{7}$$

$$\text{var}\{\Delta_\tau X_i\} = \sum_{j \in J_{\text{ncr}}} \nu_{ij}^2 a_j(\boldsymbol{x})\tau. \tag{8}$$

The bound in Eq. (5) on $|\Delta X_i|$ and hence the bound on the change in the propensity functions given by Eq. (4) may be considered "substantially satisfied"[8,10,11] if

$$|\langle \Delta_\tau X_i \rangle| \leq \max\{\varepsilon_i x_i, 1\}, \tag{9}$$

$$\sqrt{\text{var}\{\Delta_\tau X_i\}} \leq \max\{\varepsilon_i x_i, 1\}, \tag{10}$$

are simultaneously satisfied. Substituting Eqs. (7) and (8) into the above bounds on $\Delta_\tau X_i$ [Eqs. (9) and (10), respectively] gives two separate bounds on τ, both of which we require to be satisfied. Define

$$\hat{\lambda}_i = \sum_{j \in J_{\text{ncr}}} \nu_{ij} a_j(\boldsymbol{x}), \tag{11}$$

$$\hat{\sigma}_i^2 = \sum_{j \in J_{\text{ncr}}} \nu_{ij}^2 a_j(\boldsymbol{x}), \tag{12}$$

for each reactant species, i. The CGP τ-selection procedure for noncritical reactions is given by

$$\tau' = \min_{i \in I_{\text{ncr}}} \left\{ \frac{\max\{\varepsilon_i x_i, 1\}}{|\hat{\lambda}|}, \frac{\max\{\varepsilon_i x_i, 1\}^2}{\hat{\sigma}_i^2} \right\}, \tag{13}$$

where I_{ncr} is the set of noncritical species, i.e., those species which participate (either as reactants or products) in noncritical reactions, R_j, such that $j \in J_{\text{ncr}}$.

III. INCORPORATING CONFIDENCE MEASURES FOR CRITICALITY SELECTION

Consider a simple degradation reaction, $S_1 \xrightarrow{c_1} \phi$. The expected number of firings for this reaction is $\lambda = x_1 c_1 \tau$. Taking an arbitrary value for the rate constant multiplied by the time-step, e.g., $c_1\tau = 0.1$, we find that the probability of species numbers becoming negative, when the number of firings of the reaction is drawn from a Poisson distribution with mean, λ, is drastically different depending on the number of molecules of the species, x_1. For example, if $x_1 = 10$, the probability of the species becoming negative is approximately 10^{-8}. However, if $x_1 = 2$, the probability increases to 10^{-3}, five orders of magnitude larger. Choosing the critical species number to be $n_c = 1$, our second example is not flagged as critical despite the high probability of the population becoming negative. Conversely, choosing n_c to be larger, i.e., $n_c = 10$, the species which have very small probabilities of becoming negative are labeled as critical and their dependent reactions are restricted unnecessarily, slowing the simulation.

Even this simplified view presents problems for the CGP method of determining criticality. This example ignores the issues of differing reaction rates and the fact that reactant species may be repopulated during a leap, making it less likely for their population to become extinct. Different reactant species require levels of criticality dependent on their population size and on the rates of reactions that deplete and add to their numbers. The criticality of each reaction will depend on the criticality of its contingent species. This hints that taking a single critical value for all reactions is not appropriate.

If we are to draw a Poisson random number of firings of a reaction channel we would like to ensure that the probability that we take any of the contingent species' populations negative is less than some threshold value, $P_c = \alpha/100$, for example. This is the basis for a one-sided confidence interval. If the probability of the population becoming negative goes above P_c then we regard this reaction as critical. This way we eliminate reactions being unnecessarily labeled as critical when, in fact, they can undergo Poisson τ-leaping with an infinitesimally small probability of sending the populations of their contingent reactant species negative.

In order to find the probability of a reactant species' population becoming negative we consider the distribution $\Delta_\tau X_i$ of its change. A problem occurs in that, the value of τ is unknown and cannot be chosen until we have decided which reactions are critical and which are not. This "chicken and egg" situation can be resolved by invoking the leap condition, which states that the propensities should not change significantly between steps. This implies their sum should not change too much, and hence, the size of the time-step should, on average, be approximately the same. We can, therefore, evaluate $\Delta_\tau X_i$ using the value of the previous time-step, τ. Consider the distribution (6) of the random variable $\Delta_\tau X_i$:

$$\Delta_\tau X_i = \sum_{j \in J_{\text{ncr}}} \nu_{ij} \mathcal{P}_j(a_j(\boldsymbol{x})\tau).$$

Property 3 states that the sum of Poisson m-lets is given by a Poisson stopped sum distribution. This is precisely the situation when considering the distribution of $\Delta_\tau X_i$, where m is replaced by ν_{ij} and we assume, that the Poisson random variables, $\mathcal{P}_j(a_j(\boldsymbol{x})\tau)$, describing the number of firings of the reactions dependent on reactant, S_i, are uncorrelated. The $2q$ parameters (where q is the highest order of the reaction that reactant S_i is involved in) of the Poisson stopped sum

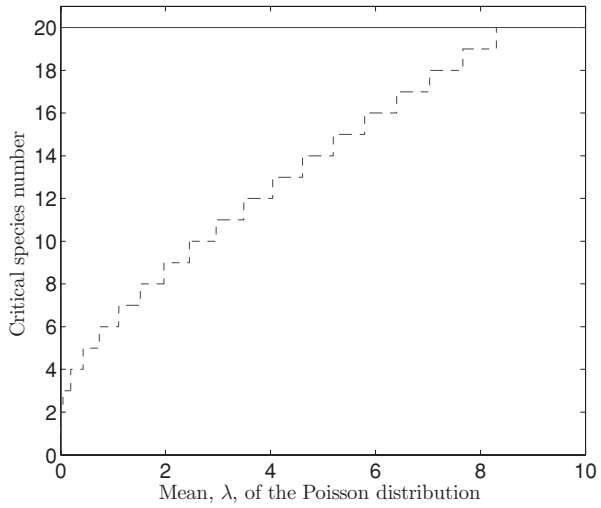


FIG. 1. An example critical curve for purely linear species depletions with criticality level $\alpha = 0.001$. The values on the x-axis represent the mean, λ, of the Poisson distribution depleting the species, i, and the dashed curve represents the value of x_i above which species, i, is not classified as critical. The full line represents the effective criticality curve of the CGP method ($n_c = 20$).

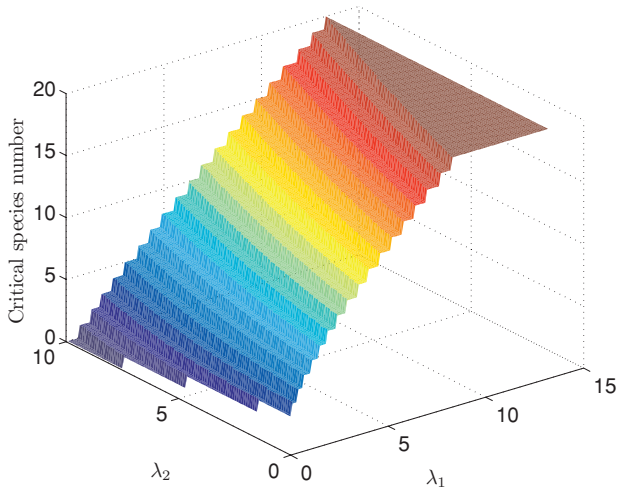


FIG. 2. An example critical surface for a Skellam distribution with criticality level $\alpha = 0.001$. The values on the x-axis represent the mean, λ_1, of the Poisson distribution depleting the species whereas the values on the y-axis represent the mean, λ_2, of the Poisson distribution regenerating the species. The surface represents the value of x_i above which species, S_i, is not classified as critical.

distribution, given in Eq. (6), all depend linearly on τ. Given a predefined confidence interval, we can use the PMF of this distribution to calculate the CMF, and hence, to calculate a critical surface for species numbers, below which species will be defined as critical.

Figure 1 displays such a criticality surface for the simplest case, $\Delta_\tau X_i = -\sum \mathcal{P}_j(a_j(\boldsymbol{x})\tau)$, where species i is depleted by linear reaction channels (i.e., $\nu_{ij} = -1$). It also shows for comparison the effective criticality curve for the CGP method of τ-selection, with $n_c = 20$. When simulating our revised method for values of the parameter that would give species criticality values above 20 we take the species criticality value to be exactly 20. This threshold is necessitated by the fact that we cannot calculate the value of the species criticality curve for all possible (infinitely many) values of the parameter of the Poisson distribution. As such, we assign a maximum criticality threshold of 20 which, as the maximum value of n_c considered by Cao *et al.*[12] has previously been shown to be robust in avoiding negative populations in τ-leaping.

In a more complicated scenario, in which a species undergoes linear degradation and creation reactions during each τ-leap, the species is depleted by a Poisson distribution with mean, λ_1, but replenished by a Poisson distribution with mean, λ_2. The random variable describing the number of molecules of the species that are depleted is drawn from a Skellam distribution with mean, $\lambda_1 - \lambda_2$, and variance, $\lambda_1 + \lambda_2$, given by $-\Delta_\tau X_i = \mathcal{P}_1(\lambda_1) - \mathcal{P}_2(\lambda_2)$. For the two parameters of this Skellam distribution, λ_1 and λ_2, we can calculate a critical *surface*, below which species will be considered critical (see Fig. 2). This criticality surface is sufficient for all sets of linear reactions. Similar curves/surfaces can be calculated for nonlinear reactions using the more general Poisson stopped sum distributions.

This revised method of deciding the criticality eliminates the problem of reactions that have an infinitesimally small probability of driving their reactant species populations negative being labeled as critical. It also ensures that reactions that have a high probability of sending reactant species negative are always labeled as critical. This allows us to approximate the number of firings of more reaction channels using Poisson random variables and fewer using the adapted SSA technique.

An algorithm for the implementation of the confidence-based methods is outlined as follows.

1. Initialize the time, $t = t_0$, and the species numbers, $\boldsymbol{X}(t_0) = \boldsymbol{x}_0$.
2. Evaluate the propensity functions, $a_j(\boldsymbol{x})$, and their sum, $a_0(\boldsymbol{x}) = \sum_{j=1}^{M} a_j(\boldsymbol{x})$, and identify which reactions are currently critical using the method described in this section (Sec. III).
3. With a prespecified value of ε, compute a candidate leap time, τ', using the method of Cao *et al.*[10] as described in Sec. II C. τ' is the largest permissible time-step for noncritical reactions.
4. If the value of τ' chosen is less than some small multiple (say 2) of $1/a_0$ then reject it and implement a number (say 100) of successive steps of the SSA before returning to step (2). If τ is larger than our chosen small multiple of $1/a_0$ then accept it and proceed to step (5).
5. Compute the sum, $a_0^c(\boldsymbol{x})$, of the critical propensity functions. Generate a second candidate leap time, τ'', in a method akin to the SSA, as a sample of an exponential random variable with mean, $1/a_0^c(\boldsymbol{x})$. τ'' is a tentative estimate of the time to the next critical reaction.
6. Take $\tau = \min\{\tau', \tau''\}$. For all noncritical reactions, R_j, generate k_j as a sample of the Poisson random variable with mean, $a_j(\boldsymbol{x})\tau$.
   (a) If $\tau' < \tau''$, set $k_j = 0$ for all critical reactions (i.e., no critical reactions fire).

(b) If $\tau'' < \tau'$, use *rand*, a uniform random number between 0 and 1, to generate a reaction index, j_c, with probability, $a_{j_c}(\boldsymbol{x})/a_0^c(\boldsymbol{x})$, in proportion to its propensity function, i.e., find j_c, such that

$$\sum_{j'=1}^{j_c} a_{j'}(\boldsymbol{x}) < a_0^c(\boldsymbol{x}) rand < \sum_{j'=1}^{j_c+1} a_{j'}(\boldsymbol{x}). \tag{14}$$

Set $k_{j_c} = 1$, and for all other critical reactions set $k_j = 0$.

7. If there is a negative component in $\boldsymbol{x} + \sum_{j=1}^{M} k_j \boldsymbol{v}_j$, reduce τ' by half and return to step (4). Otherwise update $\boldsymbol{x} = \boldsymbol{x} + \sum_{j=1}^{M} k_j \boldsymbol{v}_j$ and $t = t + \tau$.
8. If $t < t_{\text{final}}$, the desired stopping time, then go to step (2). Otherwise stop.

IV. NUMERICAL EXAMPLES

In order to test our revised τ-leaping method we have applied it to four different systems of reactions. The first is a set of 100 isomerization reactions, the second a similar set of 100 dimerization reactions. The third is the standard LacZ/LacY system[16] using the model developed by Kierzek[17–19] as employed by Tian and Burrage[13] and Cao *et al.*[10,12] to test the accuracy and efficiency of their τ-leaping algorithms. The fourth and final comparison is a system simulating cell migration in a one-dimensional domain using a position-jump process (see Baker *et al.*[2]). When calculating a critical curve/surface in these numerical simulations we consider the distribution of the change in species number due solely to reactions that deplete the species number:

$$\sum_{\nu_{ij}<0, j\in J_{\text{ncr}}} \nu_{ij} \mathcal{P}_j(a_j(\boldsymbol{x})\tau). \tag{15}$$

In the comparisons that follow we will denote this the "confidence-based (CB)-depletion" method, indicating that the distribution we consider for the change in species is purely from reactions which deplete the species number. This method will use a criticality curve similar to that shown in Fig. 1. We recognize that this does not take into account the possible repopulation of the species by reactions for which $\nu_{ij} > 0$ and, as such, in this method, we will be overly cautious in our classification of critical reactions. For comparison, we also consider the distribution of the change in species number due to those reactions that deplete and those that regenerate the species number:

$$\sum_{j\in J_{\text{ncr}}} \nu_{ij} \mathcal{P}_j(a_j(\boldsymbol{x})\tau). \tag{16}$$

We denote this as the "CB-regeneration" method. This method will employ a criticality curve similar to that shown in Fig. 2. In general, the "regeneration" method will be more efficient than the "depletion" method, since by considering possible regeneration of species numbers during reactions we will obtain a tighter bound on the necessary level of criticality for each species and will classify fewer reactions, unnecessarily, as critical. However, the depletion method may be simpler to implement since the cumulative distribution function (CDF) of the Poisson distribution has a closed form and the CDF of the Skellam distribution does not. This makes calculating the criticality curve for the depletion method, using the CDF of the Poisson distribution simpler than for the regeneration method, where the probability density function of the Skellam distribution must be summed for each value of the parameter to calculate the desired curve.

In the following simulations we compare the times of the direct method implementation of the SSA, the CGP with an extensive range of values of the criticality parameter, $n_c = 0 \ldots 20$ (approximately the range of values suggested by Cao *et al.*[12]), and the CB methods with confidence level, $\alpha = 1 \times 10^{-7}$. The accuracy of the CB methods is found to be comparable to that of the CGP method in all simulations (figures not shown). All simulations detailed in this section were written in C++ performed on a "no name" brand machine with 4 AMD (Advanced MicroDevices) Phenom(tm) II 945 processors (3GHz clock speed, 2MB L2 cache, 6MB L3 cache), 64-bit kernel running Ubuntu Linux 10.04 LTS.

A. Isomerization loops

In this, somewhat contrived, set of reactions there are 100 reactant species and 100 reactions. These reactions form two isomerization loops. In both loops every species decays to the next until the last species decays back to the first:

$$\begin{aligned} S_1 &\xrightarrow{c_1} S_2, \\ &\vdots \\ S_{50} &\xrightarrow{c_{50}} S_1, \\ S_{51} &\xrightarrow{c_{51}} S_{52}, \\ &\vdots \\ S_{100} &\xrightarrow{c_{100}} S_{51}. \end{aligned} \tag{17}$$

All the isomerization reactions occur with the same rate constant, $c_i = 0.1$, for $i = 1 \ldots 100$. Initially each species in the first isomerization loop has 10 molecules and each species in the second loop has 1000 [i.e., $S_i(0) = 10$ for $i = 1 \ldots 50$ and $S_i(0) = 1000$ for $i = 51 \ldots 100$]. We simulate over the time interval, $t \in [0, 10]$, and repeat the simulations 1000 times. The results of these simulations are given in Table I.

The CB methods are the fastest in this situation with the regeneration method being slightly faster than the deple-

TABLE I. A comparison, for model (17), of the total central processing unit (CPU) time, the number of steps taken, and the number of reactions deemed to be critical over a common time interval ($t \in [0, 10]$) from a common initial condition [$S_i(0) = 10$ for $i = 1 \ldots 50$ and $S_i(0) = 1000$ for $i = 51 \ldots 100$] for 1000 repeats using the SSA and three τ-leaping methods. None of the τ-leaping methods required the rejection of any time-steps due to negative species numbers. We choose the criticality parameter, $n_c = 11$, as the median value of the range 2 to 20 suggested by Cao *et al.* (Ref. 12).

	CPU time (s)	Steps	Critical reactions
SSA	72.93	50 507 096	N/A
CGP ($n_c = 11$)	18.93	264 192	8 232 365
CB-depletion	2.77	31 080	66 544
CB-regeneration	2.74	29 461	45 493

TABLE II. A comparison, for model (17), of the total CPU time, the number of steps taken, and the number of reactions deemed to be critical over a common time interval ($t \in [0, 10]$) from a common initial condition [$S_i(0) = 10$ for $i = 1 \ldots 100$] for 1000 repeats using the SSA and three τ-leaping methods. The SSA only marginally out-performs the CB methods, but both the CB method and the SSA drastically out-perform the CGP method.

	CPU time (s)	Steps	Critical reactions
SSA	1.40	1 001 664	N/A
CGP ($n_c = 11$)	37.75	529 277	33 053 214
CB-depletion	2.65	34 017	127 148
CB-regeneration	2.41	31 666	83 179

tion method, as expected. There are sufficiently many of each species to make τ-leaping more effective than the SSA, but there are a number of species which are below the criticality threshold, $n_c = 11$, of the CGP τ-leaping method. The CB methods determine that these reaction channels are not, in fact, critical as they have a minuscule probability of sending their contingent species negative. We can alter the initial conditions to show that the CB methods perform consistently well in comparison to the SSA and the CGP τ-leaping method.

For example, if we set the initial number of each species to be 10 [i.e., $S_i(0) = 10$ for $i = 1 \ldots 100$], then we see that the SSA out-performs all three τ-leaping algorithms (see Table II). It out-performs the CGP τ-leaping method by more than an order of magnitude, but only marginally out-performs the CB methods.

If we set the initial number of each species to be 1000 [i.e., $S_i(0) = 1000$ for $i = 1 \ldots 100$] we see that the CB methods are nearly two orders of magnitude faster than the SSA and of comparable (or slightly increased) efficiency with the CGP method (see Table III). For this reaction system the CB methods demonstrate more versatility than the traditional one-size-fits-all approach of the CGP algorithm.

Figure 3 demonstrates that the CB methods perform consistently well in comparison to the CGP technique over a range of values of the criticality parameter, n_c. At worst the speeds of the two algorithms are comparable and at best the CB algorithms are considerably faster than the CGP algorithm. These plots demonstrate that the CB methods are a robust choice in the face of differing initial conditions and crucially their performance is independent of any tuning parameters.

TABLE III. A comparison, for model (17), of the total CPU time, the number of steps taken, and the number of reactions deemed to be critical over a common time interval ($t \in [0, 10]$) from a common initial condition [$S_i(0) = 1000$ for $i = 1 \ldots 100$] for 1000 repeats using the SSA and three τ-leaping methods.

	CPU time (s)	Steps	Critical reactions
SSA	127.32	100 004 782	N/A
CGP ($n_c = 11$)	1.38	3830	0
CB-depletion	1.37	3818	0
CB-regeneration	1.36	3789	0

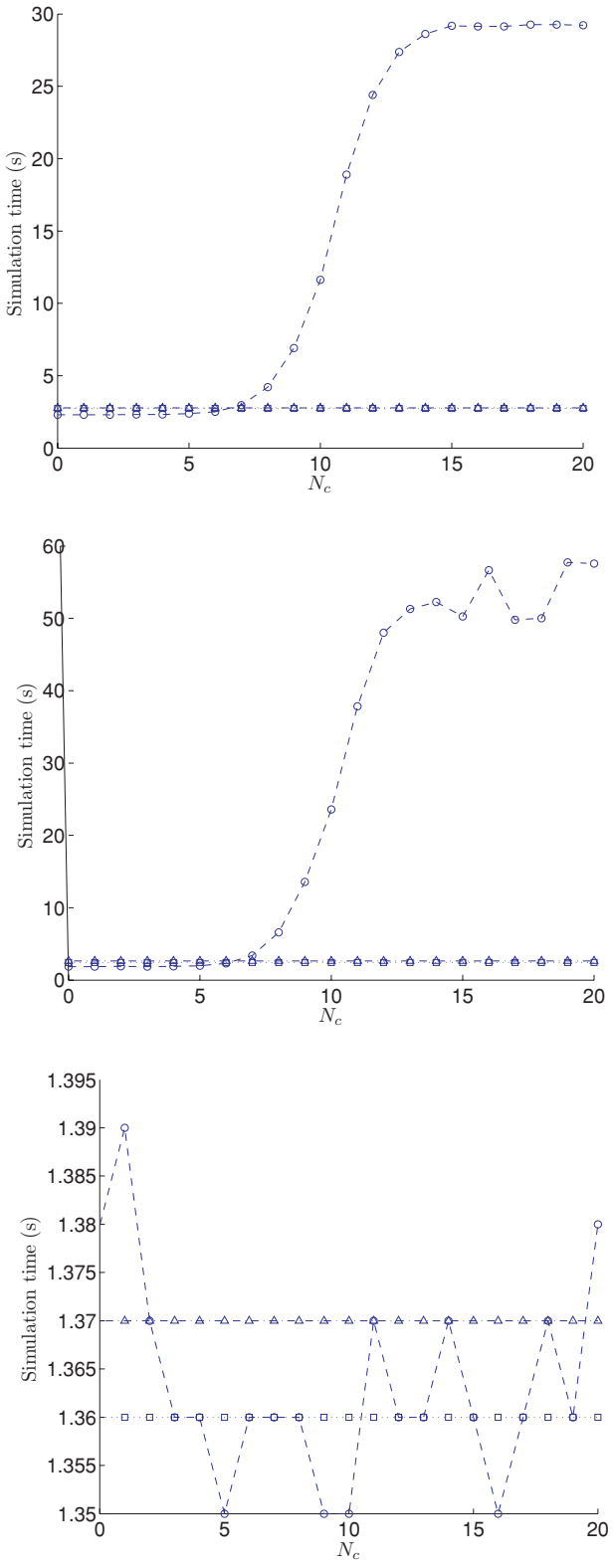


FIG. 3. A comparison of simulation times for 1000 repeats of each of the three τ-leaping methods for each of the three initial conditions described in Tables I, II, and III: (a) $S_i(0) = 10$ for $i = 1 \ldots 50$ and $S_i(0) = 1000$ for $i = 51 \ldots 100$, (b) $S_i(0) = 10$ for $i = 1 \ldots 100$, (c) $S_i(0) = 1000$ for $i = 1 \ldots 100$, over the range of values of the parameter $n_c = 0 \ldots 20$. Circles joined by a dashed line correspond to the CGP method, triangles joined by a dotted-dashed line to the CB-depletion method, and squares joined by a dotted line to the CB-regeneration method. In (a) and (b), for lower values of n_c, the simulation times of all three algorithms are comparable, however, for larger values of n_c, the CB methods are significantly faster. In (c) all three methods are approximately comparable independent of the value of n_c.

TABLE IV. A comparison, for model (18), of the total CPU time, the number of steps taken, and the number of reactions deemed to be critical over a common time interval ($t \in [0, 10]$) from a common initial condition [$S_i(0) = 10$ for $i = 1 \ldots 50$ and $S_i(0) = 1000$ for $i = 51 \ldots 100$] for 1000 repeats using the SSA and three τ-leaping methods. None of the τ-leaping methods required the rejection of any time-steps due to negative species numbers.

	CPU time (s)	Steps	Critical reactions
SSA	39 706	3 463 378 894	N/A
CGP ($n_c = 11$)	398	2 468 173	123 345 100
New	320	972 062	13 492 169
New	313	898 838	7,982 245

TABLE V. A comparison, for model (18), of the total CPU time, the number of steps taken, and the number of reactions deemed to be critical over a common time interval ($t \in [0, 10]$) from a common initial condition [$S_i(0) = 1000$ for $i = 1 \ldots 100$] for 1000 repeats using the SSA and three τ-leaping methods.

	CPU time (s)	Steps	Critical reactions
SSA	70 402	2 516 518 253	N/A
CGP ($n_c = 11$)	1923	19 979 926	0
CB-depletion	1917	19 980 419	0
CB-regeneration	1888	19 979 825	0

B. Dimerization loops

In order to show that the CB methods can be applied straightforwardly to systems of nonlinear reactions we have altered the above reaction system, to be dimerizations (instead of isomerizations) where two molecules of each reactant species are required for each reaction. These reactions form two dimerization loops. In each loop one molecule of a species combines with another of the same species and decays to two molecules of the next species until molecules of the last species decay back to the first:

$$\begin{aligned} S_1 + S_1 &\xrightarrow{c_1} 2S_2 \\ S_2 + S_2 &\xrightarrow{c_2} 2S_3 \\ &\vdots \\ S_{50} + S_{50} &\xrightarrow{c_{50}} 2S_1 \\ S_{51} + S_{51} &\xrightarrow{c_{51}} 2S_{52} \\ S_{52} + S_{52} &\xrightarrow{c_{52}} 2S_{53} \\ &\vdots \\ S_{100} + S_{100} &\xrightarrow{c_{100}} 2S_{51}. \end{aligned} \tag{18}$$

All the dimerization reactions occur with the same rate constant, $c_i = 0.1$, for $i = 1 \ldots 100$. Initially each species in the first dimerization loop has 10 molecules and each species in the second loop has 1000 molecules [i.e., $S_i(0) = 10$ for $i = 1 \ldots 50$ and $S_i(0) = 1000$ for $i = 51 \ldots 100$]. We use this set of reactions to test the speed and reliability of the algorithms. We simulate over the time interval [0, 10] and repeat the simulations 1000 times.

Clearly the CB methods are faster than the CGP in the situation described above (see Table IV) for similar reasons to those given for the isomerization loop system. Altering the initial conditions we will see that the CB methods perform consistently well in comparison to the SSA and the CGP τ-leaping method. If we set the initial number of each species to be large for both dimerization loops; 1000, for example, [i.e., $S_i(0) = 1000$ for $i = 1 \ldots 100$] then we see that all three τ-leaping methods are well over an order of magnitude faster than the SSA. Both CB methods remain marginally more efficient than the CGP method (see Table V). Figure 4 demonstrates the consistently good performance of the CB method in comparison to the CGP technique as n_c varies for the dimerization loop described above.

C. LacY and LacZ expression model

The genetic regulation of the lac operon was one of the earliest complex genetic regulatory mechanisms to be described in detail. The lac operon consists of a promoter, a terminator, a regulator, an operator, and three structural genes. The two most important of these structural genes are LacY and LacZ, necessary for lactose catabolism.[16] The other constituents of the lac operon control the expression of LacY and LacZ via a series of interactions with transcription regulators in the cytosol. The number of transcription regulators in the system may be as low as ten and these transcription regulators bind to a single "molecule" of the DNA regulatory region.[16] Clearly, stochastic fluctuations of molecule numbers and time intervals will be important in such a system and the near-zero copy numbers of some species make this system a good test of τ-leaping methods aiming to avoid negative populations. The LacZ/LacY model as developed by Kierzek,[19] consists of 18 reactant species, one product (a result of the catabolization of lactose which does not take part, as a reactant, in any of the system's reactions) and 22 reaction channels. A list of reaction channels and rates of the chemical kinetics can be found in Tian and Burrage.[13]

Running the SSA for this reaction set from $t = 0$ to $t = 2000$ (the time interval used for comparisons of algorithms in Tian and Burrage[13]) takes over 6 min on our computer. Since running large numbers of repeats will require a large amount of computer time, we instead run the SSA from $t = 0$ to $t = 1000$ to find an "initialization state" for the system and run 1000 repeats of each simulation algorithm from $t = 1000$ to $t = 1001$, as in Cao *et al.*[10] The numbers of molecules of each species in the initialization state are given in Table VI.

The results in Table VII demonstrate that the CGP method is the quickest, closely followed by the CB-depletion and CB-regeneration τ-leaping algorithms. This result

TABLE VI. Initial numbers of molecules of each of the 19 species in the LacZ/LacY model taken from one run of the SSA from $t = 0$ to $t = 1000$.

Species	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Number	0	28	0	1	0	22	0	1	0	392
Species	S_{11}	S_{12}	S_{13}	S_{14}	S_{15}	S_{16}	S_{17}	S_{18}	S_{19}	
Number	59	57	1661	662	17 760	15 832	6 814 877	155	78 727 744	

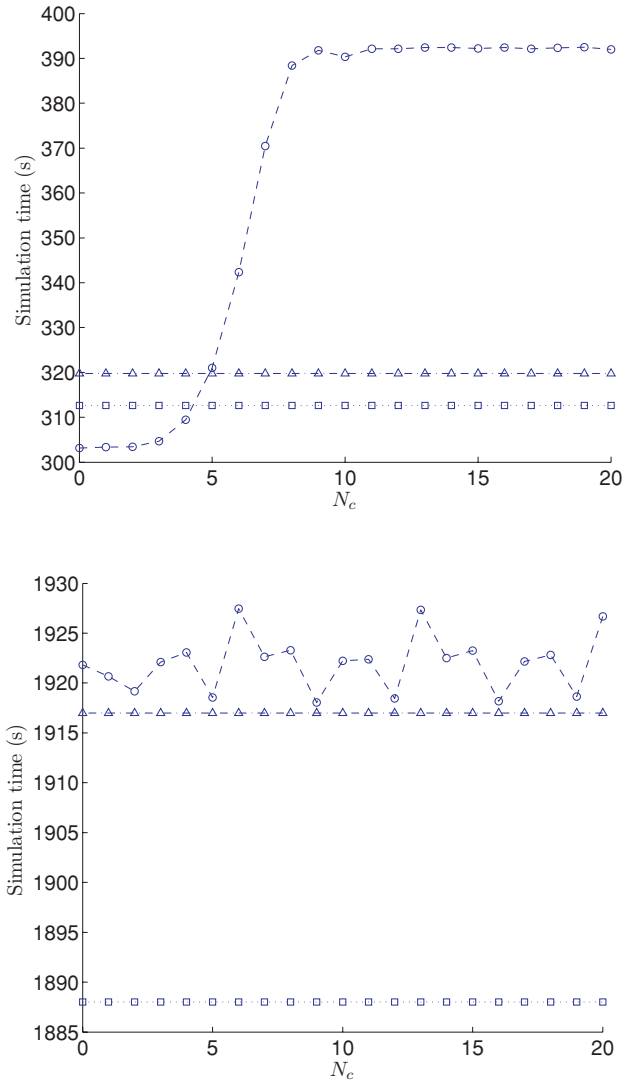


FIG. 4. A comparison of simulation times for 1000 repeats of each of the three τ-leaping methods for each of the initial conditions described in Tables IV and V: (a) $S_i(0) = 10$ for $i = 1 \ldots 50$ and $S_i(0) = 1000$ for $i = 51 \ldots 100$ and (b) $S_i(0) = 1000$ for $i = 1 \ldots 100$, over a range of values of the parameter, $n_c = 0 \ldots 20$. Legend is as described in the caption of Fig. 3. In (a) we see that for lower values of n_c, the CGP method does slightly better than the CB methods, however, for larger values of n_c, the CB methods are significantly faster. In (b) the CB methods are marginally faster than the CGP method independent of the value of n_c.

remains true for a range of values of the criticality parameter n_c (see Fig. 5). Both the τ-leaping methods are an order of magnitude faster than the SSA. Very few of the species have the sorts of numbers of molecules for the CB methods of criticality-selection to make a considerable difference over the CGP method (see Table VI). Some are too large, meaning that both CGP and CB methods will classify the dependent reactions as noncritical, and some are too small, meaning that both methods are likely to classify the dependent reactions as critical. The CB methods turn out to be slightly slower since they do not gain any advantage from differential classifications of criticality (see critical reactions column of Table VII). However, all three τ-leaping methods are still considerably faster than the SSA over the chosen time-period (see Table VII).

TABLE VII. A comparison, for the LacZ/LacY model of Kierzek (Ref. 18) of the total CPU time, the number of steps taken, and the number of reactions deemed to be critical over a common time interval ($t \in [1000, 1001]$) from a common initial condition (see Table VI) for 1000 repeats using the SSA and three τ-leaping methods.

	CPU time (s)	Steps	Critical reactions
SSA	254.32	448 740 253	N/A
CGP ($n_c = 11$)	24.56	3 904 290	13 662 309
CB-depletion	26.73	3 902 985	13 588 212
CB-regeneration	27.09	3 905 514	13 608 898

D. A further example from biology

The final system we consider is a one-dimensional position-jump model of cell migration.[2,20] The premise of the simulation is the time evolution of discrete cell density given an initial profile. The stochastic motion of individual cells is modeled across a domain with zero-flux boundary conditions. We divide the unit domain [0, 1] into $b = 50$ intervals of equal length and implement migration by allowing cells to move left and right from their current interval with constant transition rate, $d = 1$. We assume there is no cell death or division on the time-scale of interest so that, in combination with the zero-flux boundary conditions, the total cell numbers remain constant. Essentially, this example reduces to a system of $b = 50$ species reacting through $2 \times b - 2 = 98$ reaction channels (cells moving left and right from each interval except for at the end intervals where cells are only allowed to move in one direction to implement zero-flux boundary conditions). The number of cells in each interval corresponds to the number of molecules of each species. As such, all reactions are isomerizations and propensity functions depend only on the transition rate, d, and the number of cells in the corresponding interval. We can write the reaction system as follows:

$$S_1 \xrightarrow{d} S_2$$

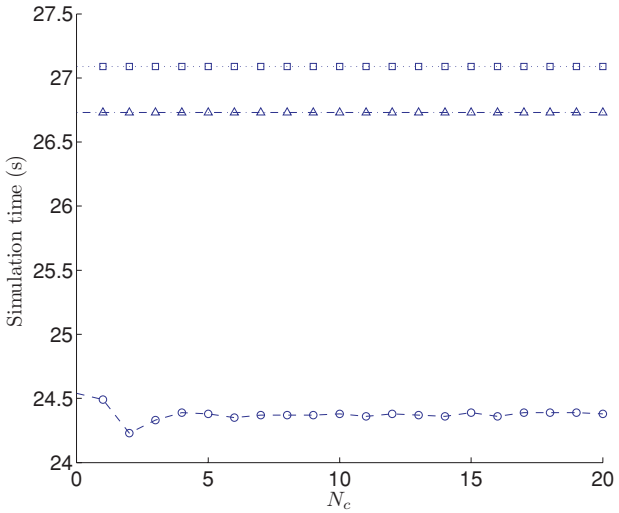


FIG. 5. A comparison of simulation times for 1000 repeats of each of the three τ-leaping methods for the LacZ/LacY model over a range of values of the parameter, $n_c = 0 \ldots 20$. For all values of n_c, the CGP method is marginally quicker than either of the CB methods.

TABLE VIII. A comparison, for model (19), of the total CPU time, the number of steps taken, and the number of reactions deemed to be critical over a common time interval ($t \in [0, 2000]$) from a common initial condition [$S_1(0) = 5000$ and $S_i(0) = 0$ for $i = 2 \ldots 50$] for 1000 repeats using the SSA and three τ-leaping methods.

	CPU time (s)	Steps	Critical reactions
SSA	34834	2.72×10^9	N/A
CGP ($n_c = 11$)	7542	1.71×10^8	5.20×10^8
CB-depletion	7310	1.66×10^8	6.86×10^8
CB-regeneration	7160	1.65×10^8	3.31×10^8

$$\begin{aligned} S_1 &\overset{d}{\leftarrow} S_2 \overset{d}{\rightarrow} S_3 \\ &\vdots \\ S_{b-2} &\overset{d}{\leftarrow} S_{b-1} \overset{d}{\rightarrow} S_b \\ S_b &\overset{d}{\rightarrow} S_{b-1}. \end{aligned} \tag{19}$$

We consider a variety of conditions to determine how the algorithms perform against each other. In the first example we place 5000 cells in the first interval of the domain [i.e., $S_1(0) = 5000$ and $S_i(0) = 0$ for $i = 2 \ldots, 50$]. The system is run over the time interval $[0, 2000]$ and results are given in Table VIII.

Table VIII demonstrates the improved efficiency of the CB methods in comparison to the CGP method. The simulation time is reduced by approximately 3%–5%. This is evidenced by the smaller number of steps taken by the CB methods. Although it appears that the CGP has fewer critical reactions, this is an artifact of our recording procedure: critical reactions are not recorded during the time the τ-leaping simulations spend implementing the SSA [when the proposed time-step, τ', given by noncritical reactions is smaller than twice the average time-step of the SSA, $2/a_0$, (see step 4 of the algorithm given in Sec. III)]. The CGP algorithm spends far more time implementing the SSA, as evidenced by the increased simulation time and the number of steps in comparison to the CB methods. Similar reasoning applies to the results in Table IX.

Figure 6 displays the size of the time-step, τ, against the time it was chosen. Early in the simulations there is usually at least one species with low molecular numbers (i.e., an interval with a small number of cells) and the CGP method marks a large proportion of reactions dependent on low copy number species as being critical, despite their minuscule probability of driving the species number negative. This results in a choice of small τ [see Fig. 6(a)]. However, the CB methods are free to classify reactions dependent on low copy number species as noncritical. Far fewer small values of τ are selected in the initial stages of the simulations using the CB methods

TABLE IX. A comparison, for model (19) (now with $b = 100$ intervals), of the total CPU time, the number of steps taken, and the number of reactions deemed to be critical over a common time interval ($t \in [0, 2000]$) from a common initial condition [$S_1(0) = 10000$ and $S_i(0) = 0$ for $i = 2 \ldots 100$] for 100 repeats using the SSA and three τ-leaping methods.

	CPU time (s)	Steps	Critical reactions
SSA	13538	4.00^9	N/A
CGP ($n_c = 11$)	4497	3.27×10^7	5.53×10^8
CB-depletion	3851	2.78×10^7	5.96×10^8
CB-regeneration	3762	2.71×10^7	2.79×10^8

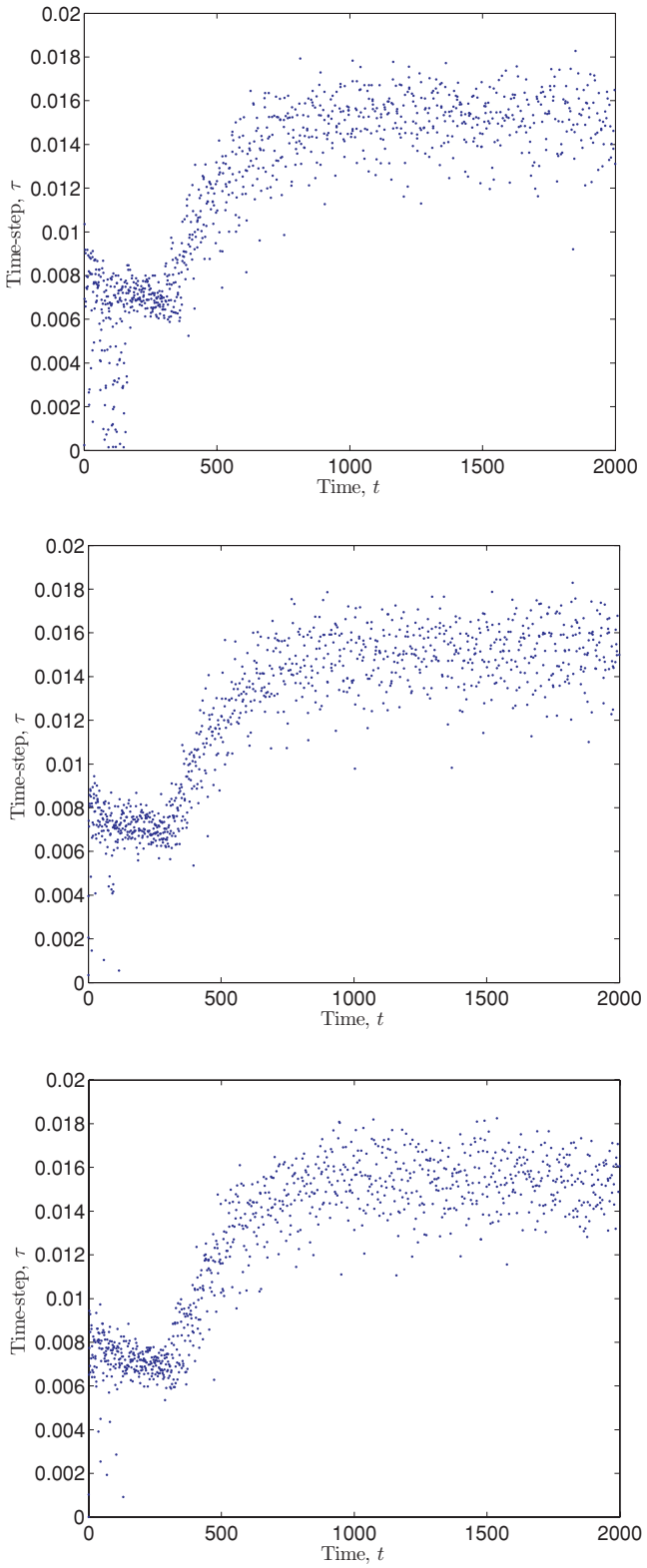


FIG. 6. For a simulation of the one-dimensional cell migration reactions of model (19) the values of the chosen time-step, τ, vs the time it was chosen, for (a) the CGP τ-selection procedure, (b) the CB-depletion procedure, and (c) the CB-regeneration procedure. τ values were taken every 1000th leap (up to a maximum of 1000 samples). Other conditions are as given in Table VIII.

[see Figs. 6(b) and 6(c)] allowing the simulations to run to completion in a shorter time (see Table VIII).

The difference between the criticality classification methods is even more stark when we simulate the same reaction model with slightly altered conditions. Consider doubling the spatial resolution of the interval (by doubling the number of intervals and halving the interval width), i.e., $b = 100$. Consequently, in order to maintain the same steady state cell numbers in each interval, we also double the initial number of cells, i.e., $S_1(0) = 10000$ and $S_i(0) = 0$ for $i = 2 \ldots 100$. We now have a reaction system of 100 chemical species reacting through 198 reaction channels. Table IX highlights the improved efficiency of the CB methods in comparison to the CGP method while at the same time it maintains its advantage over the SSA. This further demonstrates the robustness of the CB methods as a parameter-independent choice for stochastic acceleration.

Figure 7 shows the analogous comparison of time-step, τ, to Fig. 6 for the revised conditions described above. The CGP method is overly restrictive in its choice of time-step for an even larger proportion of the simulation than in Fig. 6. This trend continues as we increase the spatial resolution of the system until the CGP method is overly restrictive in time-step choice for the duration of the simulation, increasing the discrepancy in times between the CGP and CB τ-leaping methods.

As with the other model systems, in Fig. 8, we provide a comparison of the three τ-leaping algorithms for varying values of the criticality parameter, n_c. The CB methods are shown to be either comparable or superior to the CGP method as n_c varies.

As a final demonstration of the efficiency of the CB method, Fig. 9 shows the average time taken per simulation for varying numbers of cells initially positioned in the first interval of 50. The CB-depletion method is quicker for all such initial conditions and, in some cases, substantially so.

V. DISCUSSION

The CGP method for determining criticality takes a one-size-fits-all approach by allocating a critical value, n_c, for L_j values below which the reaction channel, j, is classified as critical. As demonstrated in Sec. III, by considering a simple degradation reaction, the speed and accuracy of the CGP method is dramatically dependent on the value of this parameter. If chosen too high it can be overly restrictive. If chosen too low there is a danger of species numbers becoming negative. We have presented a more versatile method of classifying criticality of reaction channels. We consider the *species* numbers and calculate how likely it is that a species will be driven negative given the reactions that deplete and replenish its numbers. We can then classify an individual species as critical and consequently all reactions dependent on that species are also classified as critical. Our confidence-based approach allows us to be more discriminatory about which species, and hence reactions, are considered critical. Only those species having a prespecified probability of becoming negative are classified as critical. This flexible approach leads to the classification of fewer reactions as critical, enabling the simulation to select more and larger time-steps from the noncritical regime. This method does not face the problem of having to tune the value of n_c for the relevant simulation, so we need not worry that we will send the population of a reaction species negative through

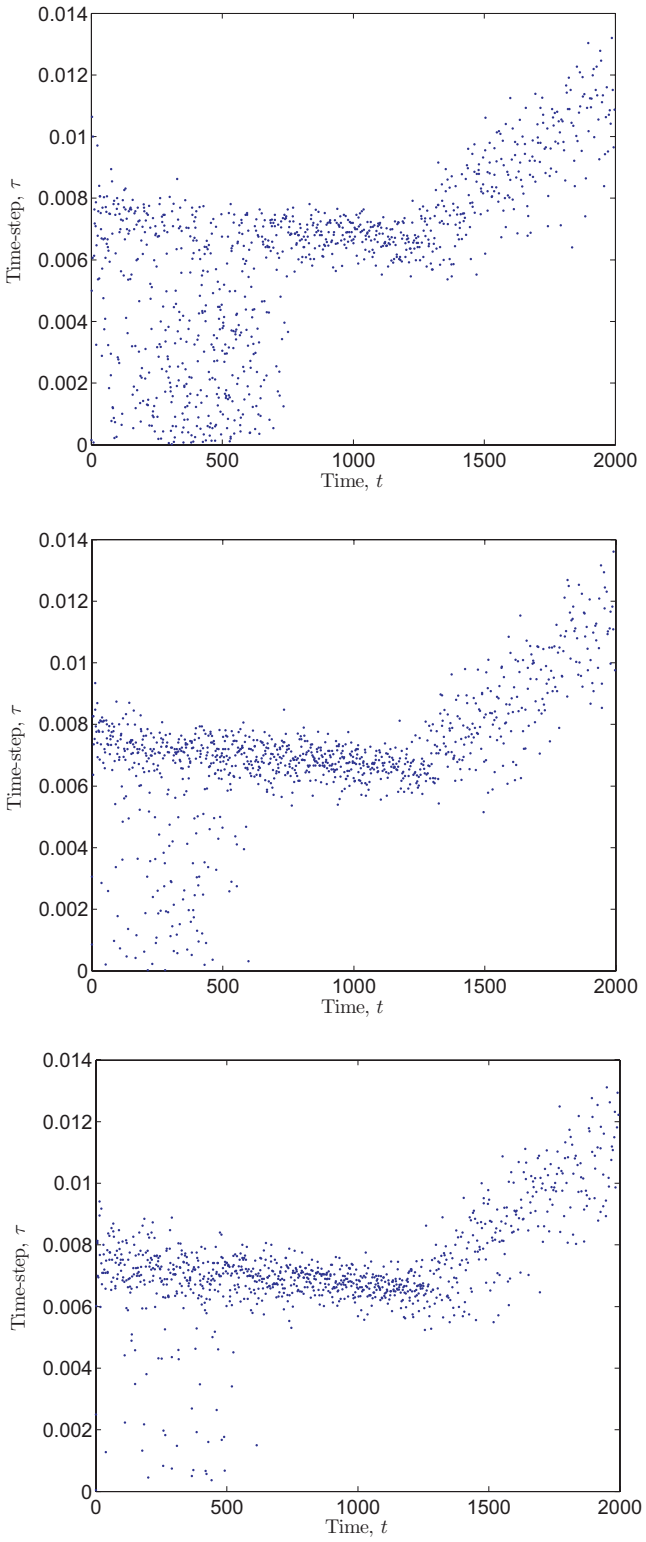
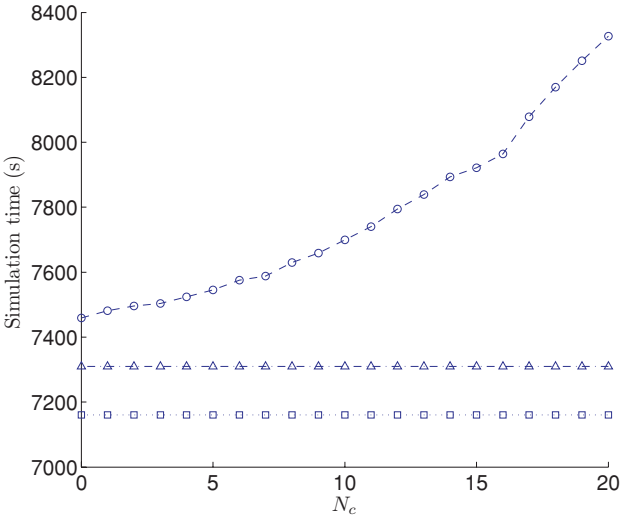


FIG. 7. For a simulation of the one-dimensional cell migration reactions of model (19) the values of the chosen time-step, τ, vs the time it was chosen, for (a) the CGP τ-selection procedure, (b) the CB-depletion procedure, and (c) the CB-regeneration procedure. τ values were taken every 1000th leap (up to a maximum of 1000 samples). Other conditions are as in Table IX.



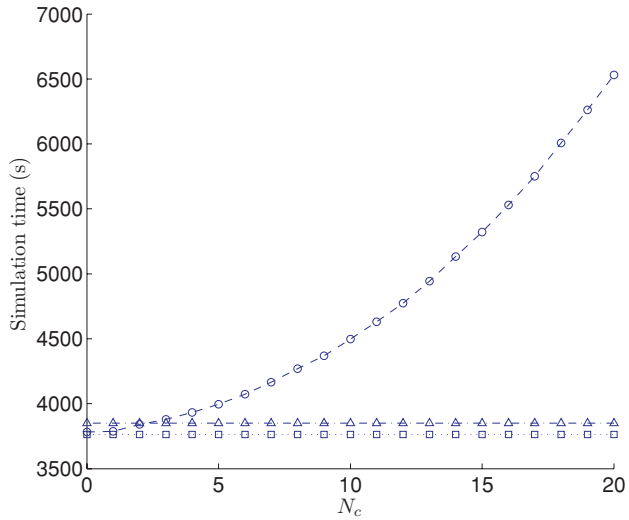


FIG. 8. A comparison of simulation times for each of the three τ-leaping methods for both of the initial conditions described above: (a) $b = 50$, $S_1(0) = 5000$ and (b) $b = 100$, $S_1(0) = 10000$ over a range of values of the parameter, $n_c = 0 \ldots 20$. Legend is as described in the caption of Fig. 3. In (a) the CB methods are significantly faster than the CGP method. Times given are for 1000 repeats of the simulation. The higher the value of n_c, the more efficient the CB methods become in comparison to the CGP method. In (b) we see that for very low values of n_c, the CGP and CB methods are of comparable speed, however, for larger values of n_c, the CB methods are significantly faster. Times given are for 100 repeats of the simulation.

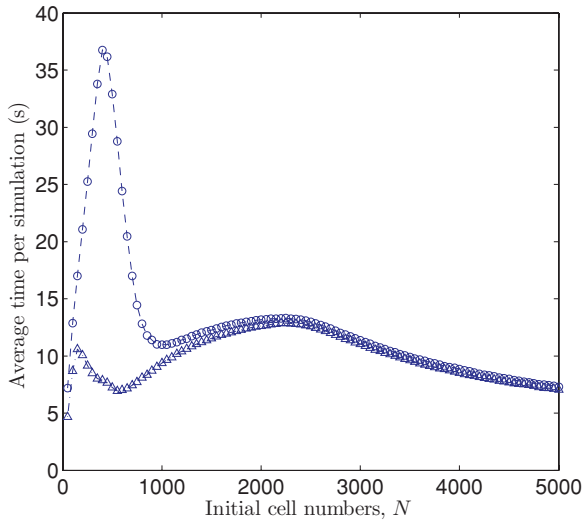


FIG. 9. Average CPU time taken given different numbers of cells initially positioned in the first interval on a domain partitioned into $b = 50$ intervals. The initial cell numbers range from $N = 50$ to $N = 5000$ in increments of 50. All times are averaged over 10 repeats. Circles with a dashed line represent the CGP method ($n_c = 11$) while triangles with a dotted-dashed line represent the CB-depletion method. The CB-regeneration method is of comparable speed to the CB-depletion method and for clarity is not shown.

incaution. This makes our method a consistent choice, robust to a wide variety of initial conditions.

Test simulations carried out on four model reaction systems have indicated that the CB τ-leaping procedures can be orders of magnitude faster than the CGP method in some situations. It appears that in all situations the CB methods will be at least of comparable speed to the CGP method with comparable accuracy. It is important to emphasize that the CB methods achieve this optimal or near-optimal performance without the necessity for the tuning parameter, n_c. This will obviate the need for running expensive test simulations to optimize the value of n_c.

We have found the accuracy of the CB methods to be comparable to that of the CGP methods in all of the simulations we have considered. An interesting further investigation of the accuracy of both τ-leaping strategies may be to consider a nonhomogeneous variation on the cell migration algorithm presented in Sec. IV D: The Fisher, Kolmogorov, Petrovski, Piskunov wave front, for example, has a speed which is extremely sensitive to small perturbations. Incorrect simulation of species numbers should have macroscopic consequences on the wave speed.[21,22]

Finally, we must acknowledge that one possible cause of erroneous classification of criticality which may occur in the CB methods (and hence a possible reduction in speed of the algorithms) is due to the fact that we are forced to use the time-step, τ, from the previous step of the algorithm in order to approximate the distribution in the change of species number for the current step. We justified this approximation, by appealing to the leap-condition: that propensity functions, and hence, the proposed time-step, τ', will not change significantly between reactions. However, there is likely to be some degree of change which will lead to erroneous classification. One possible way to circumvent this problem would be to iteratively choose τ and species criticality in a manner similar to that implemented by Harris and Clancy[23] when classifying reactions in their "partitioned leaping" approach: Start with the value of τ from the previous time-step and classify reactions as critical or otherwise based on this τ. Then update τ according to the criticality status of the reaction channels. Return and reclassify the reactions using this updated value of τ. Repeat this process until reaction classifications cease to change. Whether this iterative method of τ-selection would converge quickly and whether the advantaged gained from the precise classification of reactions would outweigh the cost of the iterative procedure remain unexplored areas for further investigation.

We suggest that our confidence-based method of species and hence reaction criticality classification is sufficient to increase the speed of many systems of reactions and will, at the very least, be a robust choice of algorithm in the face of widely varying species populations. The confidence-based method of choosing criticality will perform particularly well when there are some species with large numbers of molecules and some with smaller [formerly CGP subcritical (i.e., $S_i < n_c$)] numbers. This will confer the benefits of τ-leaping while avoiding the handicap of classifying too many reaction channels as critical and decelerating the algorithm.

ACKNOWLEDGMENTS

C.A.Y. is supported by the Engineering and Physical Sciences Research Council via the Systems Biology Doctoral Training Centre, University of Oxford and St. Catherine's College, University of Oxford.

APPENDIX A: PROOF OF PROPERTY 2

The probability that the difference of the two random variables, $\mathcal{P}_1 - \mathcal{P}_2$ (which we will now denote by $\mathcal{SK}$ in this derivation), takes the value k is the sum from $j = 0 \ldots \infty$ of the probability that the first random variable, $\mathcal{P}_1$, takes the value $k + j$ multiplied by the probability that the second random variable, $\mathcal{P}_2$, takes the value j. Using the definition of the probability density function of the Poisson random variable we can rewrite the above more mathematically as

$$\Pr(\mathcal{SK} = k) = \sum_{j=0}^{\infty} \frac{e^{-\lambda_1}\lambda_1^{(k+j)}}{(k+j)!} \times \frac{e^{-\lambda_2}\lambda_2^{j}}{j!}. \tag{A1}$$

This can be simplified in a few lines to

$$\begin{aligned}\Pr(\mathcal{SK} = k) &= e^{-[\lambda_1+\lambda_2]}\left(\frac{\lambda_1}{\lambda_2}\right)^{k/2} \sum_{j=0}^{\infty} \frac{(\lambda_1\lambda_2)^{(k/2+j)}}{(k+j)!\,j!}, \\ &= e^{-[\lambda_1+\lambda_2]}\left(\frac{\lambda_1}{\lambda_2}\right)^{k/2} I_k(2\sqrt{\lambda_1\lambda_2}),\end{aligned} \tag{A2}$$

where $I_k(z)$ is the modified Bessel function of the first kind.

APPENDIX B: PROOF OF PROPERTY 3

We prove Property 3 using PGFs. The PGF of the sum of a Poisson m-let and a Poisson n-let is given by

$$\begin{aligned}G(z) &= E(z^{\mathcal{P}^m+\mathcal{P}^n}) \\ &= G_{\mathcal{Q}_1}(z^m)G_{\mathcal{Q}_2}(z^n) \\ &= e^{\lambda_1(z^m-1)} \times e^{\lambda_2(z^n-1)}, \\ &= \exp\{\lambda_1 z^m + \lambda_2 z^n - (\lambda_1+\lambda_2)\},\end{aligned} \tag{B1}$$

where $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are Poisson distributions with means λ_1 and λ_2. The first equality makes use of the assumption that $\mathcal{P}^m$ and $\mathcal{P}^n$ are independent. The last line of Eq. (B1) is the PGF of a Poisson stopped sum distribution.

[1] C. A. Yates, R. Erban, C. Escudero, I. D. Couzin, J. Buhl, I. G. Kevrekidis, P. K. Maini, and D. J. T. Sumpter, Proc. Natl. Acad. Sci. U.S.A. **106** (14), 5464 (2009).
[2] R. E. Baker, C. A. Yates, and R. Erban, Bull. Math. Biol. **72** (3), 719 (2010).
[3] R. Erban, I. G. Kevrekidis, D. Adalsteinsson, and T. C. Elston, J. Chem. Phys. **124** (8), 084106 (2006).
[4] A. Arkin, J. Ross, and H. H. McAdams, Genetics **149** (4), 1633 (1998).
[5] D. T. Gillespie, J. Comput. Phys. **22** (4), 403 (1976).
[6] D. T. Gillespie, J. Phys. Chem. **81** (25), 2340 (1977).
[7] T. G. Kurtz, J. Chem. Phys. **57**, 2976 (1972).
[8] D. T. Gillespie, J. Chem. Phys. **115**, 1716 (2001).
[9] D. F. Anderson, J. Chem. Phys. **128**, 054103 (2008).
[10] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **124** (4), 044109 (2006).
[11] D. T. Gillespie and L. R. Petzold, J. Chem. Phys. **119**(16), 8229 (2003).
[12] Y. Cao, D. T. Gillespie, and L. R. Petzold, J. Chem. Phys. **123**, 054104 (2005).
[13] T. Tian and K. Burrage, J. Chem. Phys. **121**, 10356 (2004).
[14] A. Chatterjee, D. G. Vlachos, and M. A. Katsoulakis, J. Chem. Phys. **122**, 024112 (2004).
[15] N. L. Johnson, S. Kotz, and A. W. Kemp, *Univariate discrete distributions* (Wiley, New York, 2005).
[16] J. E. Krebs, E. S. Goldstein, and S. T. Kilpatrick, *Lewin's Genes X* (Jones & Bartlett, Sudbury, MA, 2010).
[17] T. A. Carrier and J. D. Keasling, J. Theor. Biol. **201** (1), 25 (1999).
[18] A. M. Kierzek, Methods Biochem. Anal **18** (3), 470 (2002).
[19] A. M. Kierzek, J. Zaim, and P. Zielenkiewicz, J. Biol. Chem. **276** (11), 8165 (2001).
[20] H. G. Othmer and A. Stevens, SIAM J. Appl. Math. **57**(4), 1044 (1997).
[21] E. Brunet and B. Derrida, Phys. Rev. E **56** (3), 2597 (1997).
[22] J. S. Hansen, B. Nowakowski, and A. Lemarchand, J. Chem. Phys. **124**, 034503 (2006).
[23] L. A. Harris and P. Clancy, J. Chem. Phys. **125**, 144107 (2006).