Abstract:
The explicit tau-leaping procedure attempts to speed up the stochastic simulation of a chemically reacting system by approximating the number of firings of each reaction channel during a chosen time increment tau as a Poisson random variable. Since the Poisson random variable can have arbitrarily large sample values, there is always the possibility that this procedure will cause one or more reaction channels to fire so many times during tau that the population of some reactant species will be driven negative. Two recent papers have shown how that unacceptable occurrence can be avoided by replacing the Poisson random variables with binomial random variables, whose values are naturally bounded. This paper describes a modified Poisson tau-leaping procedure that also avoids negative populations, but is easier to implement than the binomial procedure. The new Poisson procedure also introduces a second control parameter, whose value essentially dials the procedure from the original Poisson tau-leaping at one extreme to the exact stochastic simulation algorithm at the other; therefore, the modified Poisson procedure will generally be more accurate than the original Poisson procedure. (C) 2005 American Institute of Physics.

Avoiding Negative Populations in Explicit Poisson Tau-Leaping

Yang Cao [a)]
Dept. of Computer Science, Univ. of California, Santa Barbara, Santa Barbara, CA 93106

Daniel T. Gillespie [b)]
Dan T Gillespie Consulting, Castaic, CA 91384

Linda R. Petzold [c)]
Dept. of Computer Science, Univ. of California, Santa Barbara, Santa Barbara, CA 93106

Abstract: The explicit tau-leaping procedure attempts to speed up the stochastic simulation of a chemically reacting system by approximating the number of firings of each reaction channel during a chosen time increment τ as a Poisson random variable. Since the Poisson random variable can have arbitrarily large sample values, there is always the possibility that this procedure will cause one or more reaction channels to fire so many times during τ that the population of some reactant species will be driven negative. Two recent papers have shown how that unacceptable occurrence can be avoided by replacing the Poisson random variables with binomial random variables, whose values are naturally bounded. This paper describes a modified Poisson tau-leaping procedure that also avoids negative populations, but is easier to implement than the binomial procedure. The new Poisson procedure also introduces a second control parameter, whose value essentially dials the procedure from original Poisson tau-leaping at one extreme to the exact stochastic simulation algorithm (SSA) at the other; therefore, the modified Poisson procedure will generally be more accurate than the original Poisson procedure.

To appear in *Journal of Chemical Physics* (MS # A5.05.211)

[a)]Electronic mail: ycao@engineering.ucsb.edu

[b)]Electronic mail: gillespiedt@mailaps.org

[c)]Electronic mail: petzold@engineering.ucsb.edu

I. INTRODUCTION

We consider a well-stirred system of N chemical species $\{S_1,\ldots,S_N\}$ undergoing M chemical reactions $\{R_1,\ldots,R_M\}$. The current state of the system is specified by the vector $\mathbf{x}=(x_1,\ldots,x_N)$, where x_i is the current number of S_i molecules in the system. Each reaction channel R_j is characterized by its propensity function $a_j(\mathbf{x})$ and its state-change vector $\boldsymbol{\nu}_j=(\nu_{1j},\ldots,\nu_{Nj})$; here, $a_j(\mathbf{x})\,dt$ gives the probability that the system will experience an R_j reaction in the next infinitesimal time dt, and ν_{ij} is the change in the number of S_i molecules caused by one R_j reaction.

A mathematically exact procedure for simulating the evolution of this system is the stochastic simulation algorithm (SSA), which advances the system in time from one reaction event to the next.[1] The simplest implementation of the SSA is the so-called "direct method", which goes as follows:

1. In state $\mathbf{x}$ at time t, evaluate all the propensity functions, and also their sum $a_0(\mathbf{x})\equiv\sum_{j=1}^{M}a_j(\mathbf{x})$.
2. Generate a time increment τ as a sample of the exponential random variable with mean $1/a_0(\mathbf{x})$.
3. Generate a reaction index j as a sample of the point probability function $a_j(\mathbf{x})/a_0(\mathbf{x})$ $(j=1,\ldots,M)$.
4. Update $t\leftarrow t+\tau$ and $\mathbf{x}\leftarrow\mathbf{x}+\boldsymbol{\nu}_j$.
5. Record $(t,\mathbf{x})$ if desired. Return to 1, or else stop.

Carrying out steps 2 and 3 here is mathematically straightforward: We draw two random samples r_1 and r_2 of the unit-interval uniform random variable, and then compute $\tau=\left(1/a_0(\mathbf{x})\right)\ln(1/r_1)$, and j as the smallest positive integer for which $\sum_{j'=1}^{j}a_{j'}(\mathbf{x})$ exceeds $r_2 a_0(\mathbf{x})$.

Although the SSA is mathematically exact (assuming the definition of the propensity functions accurately reflects the dynamics of the system), the task of explicitly simulating each and every reaction event often makes the SSA too slow for practical implementation. A faster but approximate stochastic simulation procedure is the explicit Poisson tau-leaping algorithm.[2] The basic idea of this procedure is to advance the system by a *pre-selected* time increment τ (in contrast to the generated time increment τ in the SSA), which is large enough that many reaction events occur in that time, but nevertheless small enough that *no propensity function value is likely to change "significantly" as a consequence of those reaction events*. The latter restriction is called

the *leap condition*. One strategy for satisfying it is to require that the expected change in each propensity function during a leap be bounded by $\varepsilon a_0(\mathbf{x})$, where ε $(0 < \varepsilon \ll 1)$ is the *error control parameter*. One way to estimate the largest value of τ that meets this particular requirement is as follows:[3] First compute

$$f_{jj'}(\mathbf{x}) \triangleq \sum_{i=1}^{N} \frac{\partial a_j(\mathbf{x})}{\partial x_i} \nu_{ij'} \quad (j, j' = 1, \ldots, M), \tag{1}$$

and

$$\left.\begin{aligned} \mu_j(\mathbf{x}) &\triangleq \sum_{j'=1}^{M} f_{jj'}(\mathbf{x})\, a_{j'}(\mathbf{x}) \\ \sigma_j^2(\mathbf{x}) &\triangleq \sum_{j'=1}^{M} f_{jj'}^2(\mathbf{x})\, a_{j'}(\mathbf{x}) \end{aligned}\right\} \quad (j = 1, \ldots, M); \tag{2}$$

then take

$$\tau = \min_{j \in [1,M]} \left\{ \frac{\varepsilon\, a_0(\mathbf{x})}{\left|\mu_j(\mathbf{x})\right|}, \frac{\varepsilon^2 a_0^2(\mathbf{x})}{\sigma_j^2(\mathbf{x})} \right\}. \tag{3}$$

Using this tau-selection procedure, the *explicit Poisson tau-leaping algorithm* goes as follows:[2,3]

1. In state $\mathbf{x}$ at time t, evaluate all the propensity functions, and their sum $a_0(\mathbf{x}) \equiv \sum_{j=1}^{M} a_j(\mathbf{x})$.
2. Using Eqs. (1) - (3), compute the largest time step τ that is not likely to result in any propensity function changing its value by more than $\varepsilon a_0(\mathbf{x})$.
3. If the τ value chosen in step 2 is less than some small multiple (say 10) of $1/a_0(\mathbf{x})$, then *reject* it and execute instead a moderate number (say 100) of successive single-reaction SSA steps before again attempting a tau-leap. Alternatively, if τ is larger than the chosen small multiple of $1/a_0(\mathbf{x})$, then *accept* it and proceed to step 4.
4. For each $j = 1, \ldots, M$, generate k_j as a sample of the Poisson random variable with mean $a_j(\mathbf{x})\tau$.
5. Update $t \leftarrow t + \tau$ and $\mathbf{x} \leftarrow \mathbf{x} + \sum_{j=1}^{M} k_j \boldsymbol{\nu}_j$.
6. Record $(t, \mathbf{x})$ if desired. Return to step 1, or else stop.

In this procedure, k_j represents the number of times reaction R_j fires in time $[t, t+\tau)$. The Poisson approximation to k_j in step 4 is justified theoretically by the fact that, to the extent that $a_j(\mathbf{x})$ remains constant over the next τ – i.e., to the extent that the

leap condition is satisfied – the number of R_j events that will occur in that next τ will by definition be the Poisson random variable with mean $a_j(\mathbf{x})\tau$. The caveat in step 3 is inserted because $1/a_0(\mathbf{x})$ is the mean time step to the next reaction event in the exact SSA, so if satisfying the leap condition restricts τ to only a few multiples of that time then it would be computationally more efficient (and also more accurate) to step according to the SSA. And since such a restriction to a small τ would likely persist for awhile, it seems reasonable to continue stepping according to the SSA for some time before again engaging the somewhat elaborate tau-selection procedure of Eqs. (1) - (3) in the hope of making a tau-leap.

The forgoing tau-leaping procedure has been shown capable of giving an acceptably accurate simulation that is substantially faster than the SSA for many "not-too-stiff" systems – i.e., systems in which the difference between the characteristic time scales of the fastest and slowest dynamical modes is not too large. But a potential problem with the procedure is that, since the Poisson random variable can have arbitrarily large sample values, we always run the risk that the Poisson approximation to k_j may result in reaction R_j firing so many times that more molecules of one of its reactants will be consumed in the τ-leap than are actually available. When that happens, step 5 may produce a negative population for that reactant species, which is unacceptable.

In the next section, we will review how a recently proposed "binomial" tau-leaping strategy manages to avoid simulating negative populations. After that we will present a "modified" Poisson tau-leaping procedure that resolves the negative population problem rather more easily, and in the process provides for increased accuracy relative to original Poisson tau-leaping.

II. BINOMIAL TAU-LEAPING

Recently, Tian and Burrage [4], and independently Chatterjee, et al. [5], proposed a way to avoid negative molecular populations in explicit tau-leaping. Their idea is to further approximate k_j as a *binomial* random variable, one that has the same mean $a_j(\mathbf{x})\tau$ as the original Poisson random variable, but whose upper limit parameter is deliberately chosen to keep k_j from being so large that more reactant molecules are consumed than are actually available.

We recall that the binomial random variable with parameters p $(0 < p < 1)$ and L (any positive integer) has mean Lp and variance $Lp(1-p)$, and its sample values range over all the integers in $[0, L]$. For the binomial random variable with mean $a_j(\mathbf{x})\tau$ and upper limit L_j, the parameter p will thus be given by $p = a_j(\mathbf{x})\tau / L_j$. The condition $p < 1$, which is required for a nonnegative binomial probability, then requires that

$$\tau < \frac{L_j}{a_j(\mathbf{x})}. \tag{4}$$

This is an *additional restriction* on the size of the leap variable τ – one that must be imposed in addition to the restrictions imposed by the leap condition through the tau-selection procedure (1) - (3). (The leap condition is just as necessary for binomial tau-leaping as it is for Poisson tau-leaping.) Although the mean $a_j(\mathbf{x})\tau$ of the resulting binomial estimate of k_j is, by construction, the same as the mean of the Poisson estimate, the variance of the binomial estimate of k_j will be $a_j(\mathbf{x})\tau[1-a_j(\mathbf{x})\tau/L_j]$, which is less than the variance $a_j(\mathbf{x})\tau$ of the Poisson estimate, as was noted by Chatterjee *et al.* [5]

To use the binomial tau-leaping procedure, we must choose for each reaction channel R_j a value for the parameter L_j, the maximum number of permitted firings of R_j during τ. Tian and Burrage [4] and Chatterjee *et al.*[5] use basically the same recipe for doing this: For the reaction $S_1 \to S_2$ they take $L_j = x_1$; for the reaction $S_1 + S_2 \to S_3 + S_4$ they take $L_j = \min(x_1, x_2)$; for the reaction $S_1 + S_1 \to S_2$ they take L_j to be the greatest integer in $x_1/2$; etc. In general, for any unimolecular or bimolecular reaction R_j, L_j is assigned the value[5]

$$L_j = \min_{i=1,\ldots,N}^{(\nu_{ij}<0)} \left\lfloor \frac{x_i}{|\nu_{ij}|} \right\rfloor, \tag{5}$$

where the square brackets denote the "greatest integer in" operation. Notice that the minimization in (5) is taken over *only* those species that get *decreased* in an R_j reaction.

But it should be noted in passing that there is some artificiality in restricting k_j to be less than or equal to the value (5), because that restriction is quite often *not* obeyed in the actual evolution of the system. For example, in the case of the two reaction channels $S_1 \rightleftarrows S_2$, restricting the total number of forward reaction events in the next τ to x_1 and the total number of backward reaction events to x_2 ignores the fact that far more of both reactions might actually occur in time τ; because, in the absence of other reaction channels involving these two species, these two reactions actually observe the less restrictive conditions that the number of forward reactions minus the number of backward reactions must be $\le x_1$, and the number of backward reactions minus the number of forward reactions must be $\le x_2$. In general, requiring $k_j \le L_j$ will be *overly restrictive* if there are other reactions present that can *increase* the populations of the consumed R_j reactants.

But there is another side to this coin, which turns out to be rather more troublesome: Requiring $k_j \le L_j$ *will not be restrictive enough* if there are other reactions present that can *decrease* the populations of the consumed R_j reactants; because, if there are two or more reaction channels with a *common* consumed reactant, we must take care that the total number of firings of *all* those reaction channels should not consume more molecules of the common reactant than are available. This requirement is clearly recognized by both Tian and Burrage [4] and Chatterjee et al.[5], but they address it in different ways.

Chatterjee et al.[5] propose to handle the problem by generating a binomial k_j subject to the limit (5) for each of the consuming reactions *in succession*, decreasing the common reactant population on the right side of (5) appropriately after each k_j is chosen. But there is a bias in this strategy that makes its outcome dependent on the arbitrary order in which the reactions are considered: Earlier considered reactions will tend to fire more often than later considered reactions; indeed, later considered reactions will not be allowed to fire at all if the earlier considered reactions have used up all the molecules of the common reactant. Chatterjee et al.[5] try to correct this bias by randomly changing the order in which the reactions are considered from one leap to the next.

Tian and Burrage[4] take a more analytical approach. They prove theorems for constraining the sum of two independent Poisson random variables that allow them to do the following: If two reactions R_1 and R_2 both consume one molecule of a common reactant species, and if there are only L molecules of that species present, then Tian and Burrage generate the number of times k_1 and k_2 that those two channels fire subject to the *constraint* $k_1 + k_2 \le L$. Since this is done in a way that treats the two reaction channels equitably, there is no bias. Tian and Burrage state that this procedure can be extended to more than two reactions, although they do not give detailed instructions for doing that. But there would appear to be other situations remaining to be addressed. For instance, if reaction R_2 in the aforementioned example consumed *two* molecules of the common species (as happens in a dimerization), then the constraint would read $k_1 + 2k_2 \le L$, and such a linear combination constraint is not covered by the sum constraint theorems of Tian and Burrage[4]. Or, if in some bimolecular reaction, one of the two reactants is also a consumed reactant in a second reaction while the other consumed reactant is also a consumed reactant in a third reaction, then the constraints on the numbers of times each of those three reactions could fire would be complicated even to write down, much less develop theorems for.

It thus appears that the problem of multiple reactions with common consumed reactants poses issues for the binomial tau-leaping strategy that have not yet been fully resolved. And writing a general binomial tau-leaping program that reliably handles all situations that could possibly arise would seem to be a very challenging task. In the following section, we describe a *modified Poisson* tau-leaping procedure that resolves the negativity problem without having to address these particular issues.

III. MODIFIED POISSON TAU-LEAPING

If the leap condition is *strictly* obeyed, in the sense that we never leap by a τ that changes the value of any propensity function by a "significant amount", we would arguably never drive any reactant population negative; because, a change from positive to negative in the value of any propensity function is arguably always "significant", even if it happens to be smaller than the bound $\varepsilon a_0(\mathbf{x})$ that is imposed by the tau-selection procedure (1) - (3). In other words, if $a_j(\mathbf{x})$ goes from a positive value to zero (or less) in a time leap τ, we cannot fairly regard it as "staying approximately constant" during τ,

so we have no justification for approximating the number of R_j firings during τ as a Poisson random variable, much less a binomial random variable. The approximation simply requires a *smaller value of* τ than the one proposed by Eqs. (1) – (3). This remedy is clearly involved in the binomial approach described in the preceding section, where the τ-value suggested by Eqs. (1) - (3) occasionally gets *reduced* by binomial condition (4).

Therefore, one obvious if unsophisticated way to avoid negative populations in Poisson tau-leaping would be to simply *not accept* any τ that produces a negative species population, and to keep trying again using smaller values of τ, reduced say by a factor of $\frac{1}{2}$, until no negative populations are obtained.

But while one or two applications of this "try again" procedure during a simulation run should be tolerable, frequent applications are not only annoyingly inefficient, but also indicative of compromised accuracy. The first step toward developing a better strategy is to recognize that negative values of a consumed reactant are likely to arise only when the population of that reactant is already small. For example, the single reaction $S_1 \to S_2$ will rarely be a problem for Poisson tau-leaping if $x_1 \geq 20$; because, in the case $x_1 = 20$ for example, no τ-selection procedure that is truly consistent with the leap condition should allow more than 8 firings of that reaction (otherwise the propensity function would suffer a "significant" change during the leap of more than 40%), and the probability that a Poisson random variable with mean 8 will give a sample value that is greater than 20 is only about 10^{-4}. But if we scale this situation down by a factor of 10, taking $x_1 = 2$ and using a Poisson random variable with mean 0.8, the probability of getting a sample value greater than x_1 increases by a factor of about 500.

It therefore seems prudent to monitor the populations of the consumed reactants for each reaction channel during a Poisson tau-leaping simulation, and to flag any reaction channel as being "critical" if it is currently in danger of exhausting any of its reactants. Taking a cue from the binomial strategy described in Sec. II, we propose to call R_j a *critical reaction* if L_j, as computed from formula (5), is found to be less than or equal to some critical value n_c. The value assigned to n_c is discretionary, but typically it might be something between 2 and 20. Of course, since the species populations change as the system evolves in time, the roster of critical reactions will have to be regularly updated as the simulation proceeds. But note that any reaction whose propensity function happens to be *zero* should *not* be placed on the critical reactions list; because, since such a reaction would have zero probability of firing, it would be incapable of driving any species population negative.

The following *modified Poisson tau-leaping procedure* incorporates the forgoing strategy in a way that ensures that no more than one firing of a critical reaction can occur in a single τ-leap. That makes it impossible for any critical reaction to produce a negative species population count. The theoretical justification for each step in this modified tau-leaping procedure will be explained in detail in the section that follows.

1. In state $\mathbf{x}$ at time t, evaluate all the propensity functions and their sum $a_0(\mathbf{x}) \equiv \sum_{j=1}^{M} a_j(\mathbf{x})$.
2. Identify the *currently critical reactions*, namely those reaction channels R_j for which $a_j(\mathbf{x}) > 0$ and $L_j \le n_c$, where L_j is as defined in Eq. (5).
3. Using a *modified* version of Eqs. (1) - (3), compute the largest time step τ' that is not likely to result in any propensity function changing its value by more than $\varepsilon a_0(\mathbf{x})$. The modification here is that *the index j' in Eqs. (1) and (2) should now run over only the non-critical reactions*. If there are no non-critical reactions (i.e., if all the reactions are critical), then ignore Eqs. (1) - (3) and put $\tau' = \infty$.
4. If the τ' value chosen in step 2 is less than some small multiple (say 10) of $1/a_0(\mathbf{x})$, then *reject* it and execute instead a moderate number (say 100) of successive single-reaction SSA steps before again attempting a tau-leap. Alternatively, if τ' is larger than the chosen small multiple of $1/a_0(\mathbf{x})$, then *accept* it and proceed to step 5.
5. Compute the *sum* $a_0^c(\mathbf{x})$ of the propensity functions of the *critical* reactions. Generate τ'' as a sample of the exponential random variable with mean $1/a_0^c(\mathbf{x})$.

6a. If $\tau' < \tau''$: Take $\tau = \tau'$. For all the *critical* reactions R_j, set $k_j = 0$. For all the *non-critical* reactions R_j, generate k_j as a sample of the Poisson random variable with mean $a_j(\mathbf{x})\tau$.

6b. If $\tau'' \le \tau'$: Take $\tau = \tau''$. Generate j_c as a sample of the integer random variable with point probabilities $a_j(\mathbf{x})/a_0^c(\mathbf{x})$, where j runs over the index values of the *critical* reactions only. Set $k_{j_c} = 1$, and for all the other critical reactions set $k_j = 0$. For all the *non-critical* reactions R_j, generate k_j as a sample of the Poisson random variable with mean $a_j(\mathbf{x})\tau$.

7. Update $t \leftarrow t + \tau$ and $\mathbf{x} \leftarrow \mathbf{x} + \sum_{j=1}^{M} k_j \boldsymbol{\nu}_j$.
8. If any component of $\mathbf{x}$ is now negative, *undo* step 7, replace $\tau' \leftarrow \tau'/2$, and return to step 6.
9. Record $(t, \mathbf{x})$ if desired. Return to step 1, or else stop.

IV. RATIONALE FOR THE MODIFIED POISSON TAU-LEAPING PROCEDURE

Step 2 in the modified Poisson tau-leaping algorithm determines which reactions are currently critical. This step has been interposed between the first two steps of the original Poisson tau-leaping algorithm. But two changes have been introduced in step 3: First, the tau-value produced by the selection procedure (1) - (3) has been labeled τ' instead of τ. And second, Eqs. (1) and (2) have been modified, in a sense simplified, in that the

index j' now runs over *only* the *non-critical* reactions. (But the index j in Eqs. (1) - (3) still runs over *all* the reactions.)

As regards the first of these two changes, τ' is now only a *candidate* for the actual tau-leap. Step 5 will produce a second candidate τ'', and step 6 will then choose as the actual time leap τ the smaller of τ' and τ''. We will explain in detail why this is done momentarily.

The justification for the change in the range of the index j' in Eqs. (1) and (2) is this: Since there will be no more than one firing among all the critical reactions, we need be concerned only with propensity function changes that are caused by potentially multiple firings of the *non-critical* reactions. An examination of the derivation of Eqs. (1) - (3) in Ref. 3 will reveal that the index j in those equations specifies the reaction whose propensity function change is being estimated, while the index j' specifies the reaction whose firings are *causing* those changes. Since we are concerned here only with changes caused by the non-critical reactions, then j' can now be restricted to the non-critical reactions. But the index j still needs to runs over *all* the reactions since, for reasons that will be explained momentarily, we must ensure that the propensity functions of the *critical* reactions will not be substantially changed during a leap by the firings of the non-critical reactions.

Step 4 is exactly the same as the third step in ordinary Poisson tau-leaping. It essentially abandons tau-leaping in favor of the exact SSA whenever the tau-selection procedure (1) - (3) produces a value on the order of the expected time to the next reaction.

To understand the logic behind steps 5 and 6, first note that if the non-critical reactions were *not* firing, then the procedure specified in step 5 to generate τ'' would make it the time to the *next* firing of a *critical* reaction (cf. step 2 of the SSA in Sec. I); likewise, the procedure used in step 6b to generate j_c would make it the index of the next-firing critical reaction (cf. step 3 of the SSA). But if firings of the non-critical reactions induce changes in the values of the propensity functions of the critical reactions, this SSA logic is no longer exact. That is why we must let j in Eqs. (1) - (3) run over the critical reactions as well as the non-critical reactions. For then, the τ'-selection procedure in step 3 should ensure that the changes in the propensity functions of the critical reactions caused by firings of the non-critical reactions during the leap will not be "significant". Then, to a first approximation, those changes can be ignored. And then we will have a logical basis for regarding τ'' and j_c as reasonably good *approximations* to the time to and the index of the next firing critical reaction.

In steps 6a and 6b, the two alternatives $\tau' < \tau''$ and $\tau'' \le \tau'$ are considered separately. If $\tau' < \tau''$, then no critical reaction will fire during $[t, t+\tau']$, since the earliest critical reaction fires at the later time $t+\tau''$; therefore, a leap by $\tau = \tau'$ proceeds according to the recipe described in step 6a. Alternatively, if $\tau'' \le \tau'$, then a leap by τ'' would be *allowed* by the leap condition (since the leap condition actually allows a leap by the larger amount τ'), and that leap would carry us to the occurrence of the next critical reaction, R_{j_c}; therefore, a leap by $\tau = \tau''$ proceeds according to the recipe described in step 6b.

Notice that in no case can more than one critical reaction occur in a leap, and whatever critical reaction channel does fire (once) will necessarily have a positive propensity function. Thus, it will be impossible under this procedure for the firing of a *critical* reaction to produce a negative species population. Of course, multiple firings of the *non-critical* reactions could still produce negative populations. But that becomes less and less probable as n_c is assigned larger and larger values. Step 8 is introduced to take care of this usually improbable eventuality. Arguably, step 8 should also have appeared in the *original* tau-leaping procedure (in Sec. I) between its last two steps. And steps 7 and 9 of the modified Poisson tau-leaping algorithm are exactly those last two steps of the original Poisson tau-leaping algorithm.

V. TUNING THE PARAMETER n_c

If the value of the parameter n_c were taken to be *zero*, then no reaction channels would ever be identified as "critical" in step 2. In that case, the modified Poisson tau-leaping procedure would reduce to the *original* Poisson tau-leaping procedure.

At the other extreme, if the value of n_c were taken so *large* that *every* reaction channel were always deemed critical, then the computation of τ' via Eqs. (1) - (3) in step 3 would never be performed (τ' would always be assigned the value ∞), and step 6b would always be selected. But no Poisson random numbers would have to be generated in step 6b, since there would be no non-critical reactions. The modified Poisson tau-leaping procedure would then reduce to the *exact SSA*. Of course, the simulation would then be very slow; therefore, we should always try to take n_c "as small as possible". A *too* small value for n_c would be signaled by the need to use the try-again procedure of step 8 more often than we would like.

These considerations show that, by adjusting the value of the parameter n_c, we can cause the modified Poisson tau-leaping procedure to perform anywhere between the original Poisson tau-leaping procedure ($n_c = 0$) and the exact SSA ($n_c = \infty$). This should give us added flexibility in finding a satisfactory compromise between simulation speed and simulation accuracy. Experience thus far suggests that a value for n_c somewhere between 5 and 15 will usually be optimal. But it would appear that, so long as n_c is large enough that step 6b is *sometimes* selected, the modified Poisson tau-leaping procedure should be *somewhat* more accurate than the original Poisson tau-leaping procedure.

VI. NUMERICAL TESTS

To test our modified Poisson tau-leaping procedure, we have applied it, along with the original Poisson tau-leaping procedure, the binomial tau-leaping procedure, and the exact SSA, to the LacY/LacZ reaction model of Kierzek[6]. This model was used by Tian and Burrage[4] to test their binomial tau-leaping procedure because they found that simulating this model using ordinary Poisson tau-leaping regularly produced negative populations. The LacY/LacZ model has 19 species and 22 reactions. We simply list the

reactions in Table 1, and refer to Kierzek[6] for an explanation of the underlying biology, and to Tian and Burrage[4] for a broader discussion of the model.

The results of our comparison runs are shown in Table 2. In evaluating these results, it should be kept in mind that the exact SSA run required 3938 s of CPU time, and simulated 1.66×10^9 individual reaction events. For error tolerance $\varepsilon = 0.03$, the *original Poisson* tau-leaping simulation took 5.15×10^5 leaps; however, over 10^4 of those leaps produced a negative species population. Whenever a negative species population was encountered, the precipitating leap was immediately undone and repeated with τ decreased by a factor of ½. In every case this was sufficient to resolve the negative population problem. But the presence of so many obvious errors in leaping suggests that less obvious errors are probably slipping through, and hence that the simulation is not being done as accurately as we might wish. Increasing ε to 0.05 reduced the number of leaps in original Poisson tau-leaping by 38%, but more than doubled the number of rejections due to negative populations.

No τ-rejections were encountered in the binomial tau-leaping runs or the modified Poisson tau-leaping runs. For $\varepsilon = 0.03$ the *binomial* run took 50% more leaps than the original Poisson run, demonstrating that avoiding negative populations generally requires taking smaller leaps. But surprisingly, increasing ε to 0.05 did not reduce the number of binomial leaps by very much; we shall explain the reason for this shortly.

All of the *modified Poisson* tau-leaping runs used $n_c = 10$, so that a reaction channel was deemed "critical" whenever it was within 10 firings of exhausting any one of its reactants. The modified Poisson procedure took 23% more leaps than the original Poisson procedure for $\varepsilon = 0.03$, and 29% more leaps for $\varepsilon = 0.05$, but fewer leaps in both cases than the binomial procedure. Of course, these results could be changed either way by suitably changing n_c, since taking $n_c = 0$ would turn the modified Poisson procedure into the original Poisson procedure, and taking $n_c = \infty$ would turn it into the SSA.

We investigated the relative accuracies of the three tau-leaping methods for the LacY/LacZ model by running ensembles of 10,000 runs using each method (with $\varepsilon = 0.03$) over a short time interval, and then comparing the final population distributions of the 19 species with those obtained in a like ensemble of SSA runs. For most species, both the binomial and the modified Poisson procedures gave noticeably more accurate distributions than the original Poisson procedure. But there was no clear winner in accuracy between the binomial and modified Poisson procedures, since for some species the binomial results were slightly more accurate while for other species the modified Poisson results were slightly more accurate.

To gain more insight into how the individual tau-leaping procedures actually functioned, we repeated the three long $\varepsilon = 0.03$ tau-leaping runs of Table 2 and plotted for each the τ-values that were used on every thousandth leap over a representative time interval. Figure 1 shows the results for the *original Poisson* tau-leaping run. The open squares in this plot identify τ-values that were originally twice as large, but got reduced to avoid negative populations. As an aside, we note that further testing revealed that practically all of the τ-values lying on the up-sloping limiting line in Fig. 1 were

determined by the second (σ_j) argument on the right hand side of Eq. (3), while the τ-values (at least those represented by solid dots) below that line were determined by the first (μ_j) argument on the right hand side of Eq. (3).

Figure 2 shows the τ-values used on every thousandth leap of the *binomial* tau-leaping run. The limiting plateau that kicks in shortly after time $t = 600$ was found to arise from the binomial condition (4) as it applies to reaction R_{21}: The reaction rate for R_{21} is 431, so if there are n LacZlactose molecules, the right hand side of condition (4) evaluates to $n/(431 \cdot n) \approx 0.0023$, which is precisely the level of the plateau. Increasing the error tolerance ε from 0.03 to 0.05 has no effect on this plateau value, and that explains why there is so little difference between the run times of the binomial procedure for those two values of ε.

Figure 3 shows the τ-values used on every thousandth leap of the *modified Poisson* tau-leaping run. In this plot, a solid dot indicates that the leap contained *no* firings of a critical reaction (step 6a of the algorithm), while an open circle indicates that the leap contained *one* firing of a critical reaction (step 6b of the algorithm). The appearance of more open-circled *low* τ-values than occurred in the original Poisson run in Fig. 1 and the binomial run in Fig. 2 shows the modified Poisson procedure "being careful" not to leap over more than one firing of a critical reaction. But the appearance of points (and open circles) *above* the 0.0023 limit of Fig. 2 shows that not leaping over more than one firing of a critical reaction can often be done using a τ-value that is actually *larger* than what would be allowed by the binomial condition (4). This illustrates the point made in the paragraph following Eq. (5), that the binomial condition (4) can be overly restrictive when the number of molecules of a species that gets consumed in one reaction can be increased by some other reaction. In this case, LacZlactose gets consumed by the limiting reaction R_{21}, but it also gets produced by reaction R_{20}.

Finally, we made comparison simulations of the simple model system

$$S_1 \xrightarrow{c_1} S_2 \xrightarrow{c_2} S_3 , \tag{6}$$

with $c_1 = 10$, $c_2 = 0.1$, and initial populations $x_1(0) = 9$, $x_2(0) = 2 \times 10^4$, $x_3(0) = 0$. More specifically, we made four sets of 10^5 simulation runs from time 0 to time 0.1, using the SSA, the original Poisson tau-leaping method, the binomial tau-leaping method, and the modified Poisson tau-leaping method. All three tau-leaping simulations had $\varepsilon = 0.03$, and the modified Poisson run had additionally $n_c = 10$.

Table 3 shows for each simulation set the CPU time for all 10^5 runs, the average number of steps per run, and the average number of step-rejections per run. (A "step" is one reaction for the SSA, and one leap for the three tau-leaping procedures.) The histogram distributions for $x_2(0.1)$ and $x_3(0.1)$ were found to be practically indistinguishable among the four simulation sets, but as Fig. 4 shows, marked differences were found in the final state distributions for species S_1: The original Poisson and binomial runs produced $x_1(0.1)$ distributions that differ noticeably from that of the SSA run, whereas the modified Poisson runs gave a distribution for $x_1(0.1)$ that is practically

indistinguishable from that of the SSA runs. The modified Poisson runs required on average over twice as much CPU time, and over three times as many steps per run, as the other two tau-leaping runs. But still, the average CPU time for the modified Poisson method was less than one-third of that for the SSA, and the average number of steps per run for the modified Poisson method was less than 4% of that for the SSA.

The increased accuracy of the modified Poisson tau-leaping method over the other two tau-leaping methods for this simple example is due to the fact that the number of S_1 molecules was always less than n_c, so reaction R_1 was always treated as a critical reaction. The lesson here is that, by taking care not to leap over more than one R_1 reaction in this simple model, we will get results that are practically as accurate as the SSA but in less time. Of course, we might not always need that level of accuracy. But the modified Poisson method gives us the option of obtaining it, simply by choosing the value of the parameter n_c.

VII. CONCLUSIONS

We have shown that the modified Poisson tau-leaping procedure described in Secs. III and IV avoids the negative population problems of original Poisson tau-leaping, and can be made to perform anywhere "between" the original tau-leaping procedure and the exact SSA simply by tuning the parameter n_c between 0 and ∞. Therefore, modified Poisson tau-leaping appears to represents a clear improvement over original Poisson tau-leaping.

As compared to the recently proposed binomial tau-leaping procedure[4,5] for avoiding negative populations, the modified Poisson procedure seems to offer several advantages. First, although both procedures make use of the rather arbitrary values L_j in Eq. (5), those L_j values do not get "quantitatively propagated" in the modified Poisson procedure as they do in the binomial procedure. The only purpose served by the L_j values in the modified Poisson procedure is to decide which reaction channels should be put on the critical reaction list. We need not even compute L_j for any reaction R_j that we are confident will never be a critical reaction.

Second, the modified Poisson procedure never has to worry about two or more reactions with a common consumed reactant "colluding" to drive the population of that common reactant negative. (This assumes that all potentially colluding reactions will be on the critical list, but that should always be so.) This follows from the fact that in a modified Poisson tau-leap, there can never be more than one firing among all the critical reactions. In contrast, binomial tau-leaping in principle allows any reaction to fire enough times in a leap that its propensity function could actually be brought to zero. But whenever *multiple* firings of a reaction channel bring its propensity function to zero, there is always a possibility that the leap condition will have been "violated in spirit", since such a change in the value of a propensity function is arguably always "significant" regardless of what Eqs. (1) - (3) might suggest. And since violations of the leap condition generally imply quantitative inaccuracies in the leap, then even though a binomial tau-

leap will never lead to negative populations, it may sometimes be accompanied by an unanticipated degradation in accuracy.

In most practical cases, such as the LacY/LacZ model considered in Sec. VI, we expect that the modified Poisson and binomial procedures will have comparable accuracies for comparable run times. But the modified Poisson procedure will be much easier to program than the binomial procedure if the latter is required to take proper account of all possible "collusions" among reaction channels with common consumed reactants. Thus, we believe that its simplicity, reliability, and tunable accuracy give modified Poisson tau-leaping a practical edge over binomial tau-leaping.

ACKNOWLEDGEMENTS: This work was supported in part by the U.S. Air Force Office of Scientific Research and the California Institute of Technology under DARPA award No. F30602-01-2-0558, and in part by the Department of Energy under DOE award No. DE-FG02-04ER25621. YC and LP received additional support from the National Science Foundation under NSF award No. CTS-0205584, and from the Institute for Collaborative Biotechnologies though grant DAAD19-03-D-0004 from the U.S. Army Research Office. DG received additional support through the Molecular Sciences Institute from Sandia National Laboratories and the Department of Energy's "Genomes to Life" Program.

REFERENCES

[1] D. T. Gillespie, J. Comput. Phys. **22**, 403 (1976); J. Phys. Chem. 81, 2340 (1977); Physica A **188**, 404 (1992).

[2] D. T. Gillespie, J. Chem. Phys. **115**, 1716 (2001).

[3] D. T. Gillespie and L. R. Petzold, J. Chem. Phys. **119**, 8229 (2003).

[4] T. Tian and K. Burrage, J. Chem. Phys. **121**, 10356 (2004).

[5] A. Chatterjee, D. Vlachos, and M. Katsoulakis, J. Chem. Phys. **122**, 024112 (2005).

[6] A. M. Kierzek, Bioinformatics **18**, 470 (2002).

FIGURE CAPTIONS

Fig. 1. Showing, for a simulation of the LacY/LacZ reactions in Table 1 using the *original Poisson* tau-leaping method with $\varepsilon = 0.03$, the tau-values that were used every thousandth leap. Open squares indicate tau-values that had been reduced by a factor of ½ from the values that had first been proposed by the tau-selection procedure (1) - (3), in order to avoid negative populations.

Fig. 2. Showing, for a simulation of the LacY/LacZ reactions using the *binomial* tau-leaping method with $\varepsilon = 0.03$, the tau-values that were used every thousandth leap.

Fig. 3. Showing, for a simulation of the LacY/LacZ reactions using the *modified Poisson* tau-leaping method with $\varepsilon = 0.03$ and $n_c = 10$, the tau-values that were used every thousandth leap. A solid point indicates that the leap occurred without the firing of any critical reaction (step 6a), while an open circle indicates that the leap occurred with one critical reaction firing once (step 6b).

Fig. 4. Showing, for four sets of 10^5 simulation runs of reactions (6) from the initial state $(x_1, x_2, x_3) = (9, 20000, 0)$ to time 0.1, using the exact SSA and the three tau-leaping methods, histograms of the final distributions of the S_1 population. The three tau-leaping methods all used $\varepsilon = 0.03$, and the modified Poisson tau-leaping method also used $n_c = 10$. (The final state distributions of the S_2 and S_3 populations for the four simulation methods were practically indistinguishable.)

	Reaction channel	*Reaction rate*
R_1	PLac + RNAP $\rightarrow$ PLacRNAP	0.17
R_2	PLacRNAP $\rightarrow$ PLac + RNAP	10
R_3	PLacRNAP $\rightarrow$ TrLacZl	1
R_4	TrLacZl $\rightarrow$ RbsLacZ + PLac + TrLacZ2	1
R_5	TrLacZ2 $\rightarrow$ TrLacYl	0.015
R_6	TrLacYl $\rightarrow$ RbsLacY + TrLacY2	1
R_7	TrLacY2 $\rightarrow$ RNAP	0.36
R_8	Ribosome + RbsLacZ $\rightarrow$ RbsRibosomeLacZ	0.17
R_9	Ribosome + RbsLacY $\rightarrow$ RbsRibosomeLacY	0.17
R_{10}	RbsRibosomeLacZ $\rightarrow$ Ribosome + RbsLacZ	0.45
R_{11}	RbsRibosomeLacY $\rightarrow$ Ribosome + RbsLacY	0.45
R_{12}	RbsRibosomeLacZ $\rightarrow$ TrRbsLacZ + RbsLacZ	0.4
R_{13}	RbsRibosomeLacY $\rightarrow$ TrRbsLacY + RbsLacY	0.4
R_{14}	TrRbsLacZ $\rightarrow$ LacZ	0.015
R_{15}	TrRbsLacY $\rightarrow$ LacY	0.036
R_{16}	LacZ $\rightarrow$ dgrLacZ	6.42×10^{-5}
R_{17}	LacY $\rightarrow$ dgrLacY	6.42×10^{-5}
R_{18}	RbsLacZ $\rightarrow$ dgrRbsLacZ	0.3
R_{19}	RbsLacY $\rightarrow$ dgrRbsLacY	0.3
R_{20}	LacZ + lactose $\rightarrow$ LacZlactose	9.52×10^{-5}
R_{21}	LacZlactose $\rightarrow$ product + LacZ	431
R_{22}	LacY $\rightarrow$ lactose + LacY	14

Table 1: Reaction channels and rates for the LacZ/LacY model of Kierzek[6].

	Original Poisson			Binomial		Mod. Poisson ($n_c = 10$)	
ε	Time (s)	Leaps	Rejects	Time (s)	Leaps	Time (s)	Leaps
0.03	57	5.15×10^5	10493	89	7.75×10^5	72	6.31×10^5
0.05	36	3.20×10^5	21968	85	7.73×10^5	47	4.13×10^5

Table 2: CPU time and total number of leaps taken for one simulation run of the LacY/LacZ model over a common time interval from a common initial condition, using three different tau-leaping methods and two different values of the error control parameter ε. Also shown for the original Poisson tau-leaping run is the number of times during the run that the selected value of τ had to be rejected because it produced a negative population; such leaps were undone and repeated with τ reduced by a factor of ½. The corresponding exact SSA run required 3938 s of CPU time, and took 1.66×10^9 steps (individual reactions).

	SSA	Original Poisson	Binomial	Modified Poisson
Total CPU Time (s)	35.3	4.5	4.1	10.2
Avg Steps Per Run	204.7	2.0	2.0	6.7
Avg. Rejects Per Run	0	0.13	0	0

Table 3: Total CPU time, average number of steps per run, and average number of step rejections per run, for sets of 10^5 simulations from time 0 to time 0.1 of the model (6) using the SSA, original Poisson tau-leaping, binomial tau-leaping, and modified Poisson tau-leaping.

Fig 1

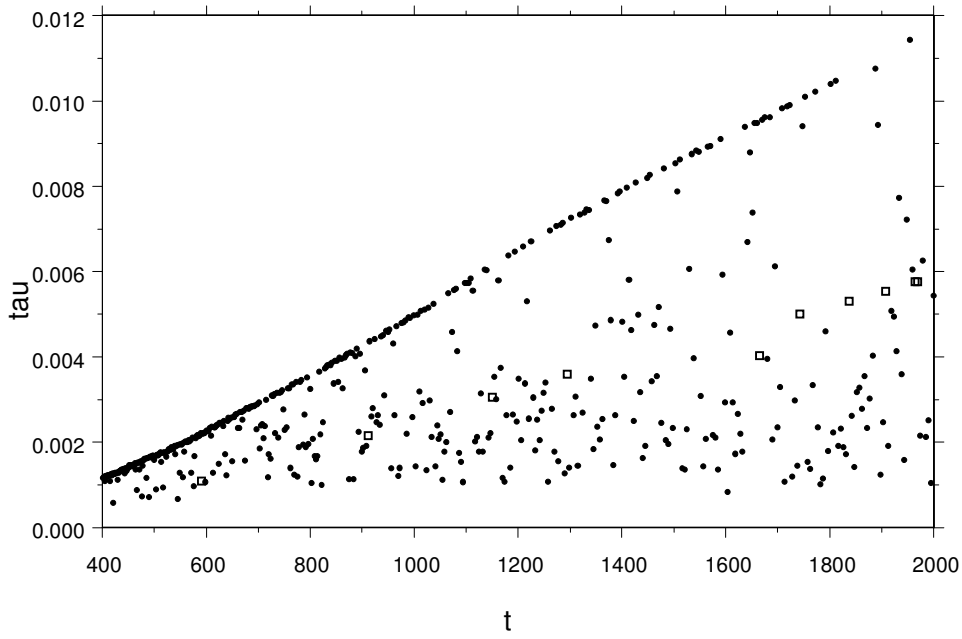

0.012
0.010
0.008
0.006
0.004
0.002
0.000
tau
400
600
800
1000
1200
1400
1600
1800
2000
t

Fig 2

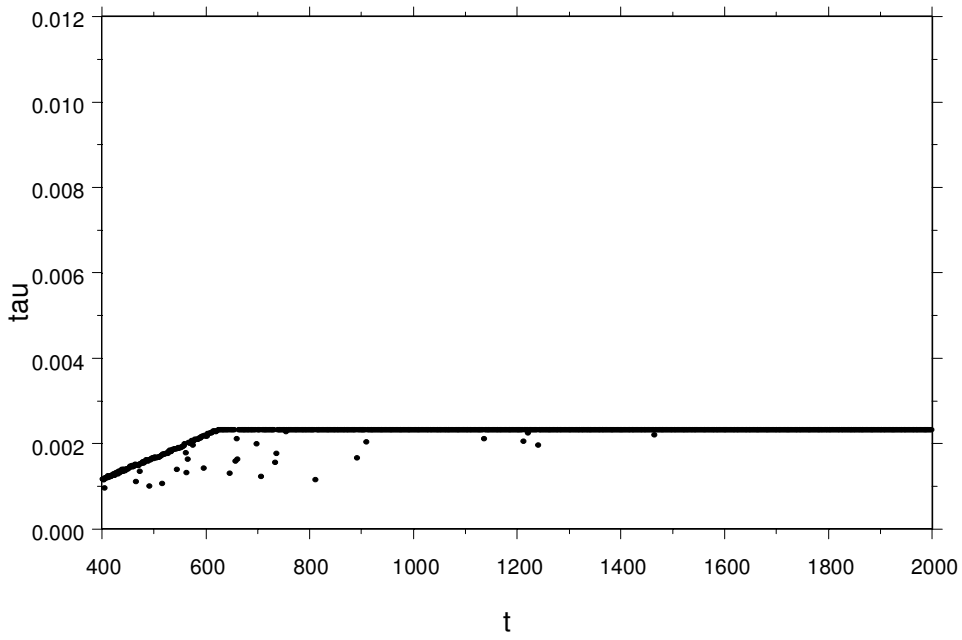

0.012
0.010
0.008
0.006
0.004
0.002
0.000
tau
400
600
800
1000
1200
1400
1600
1800
2000
t

Fig 3

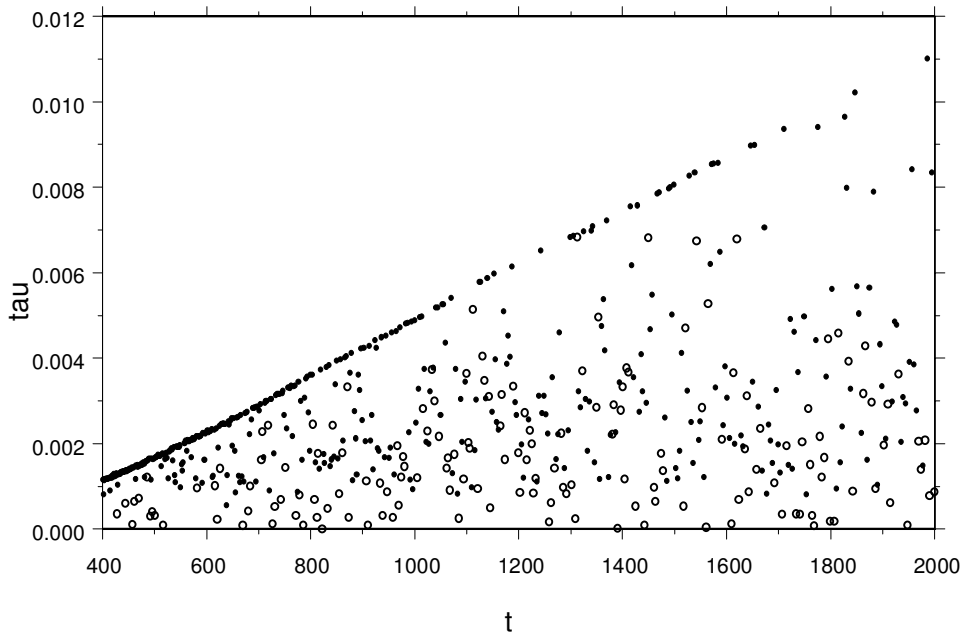
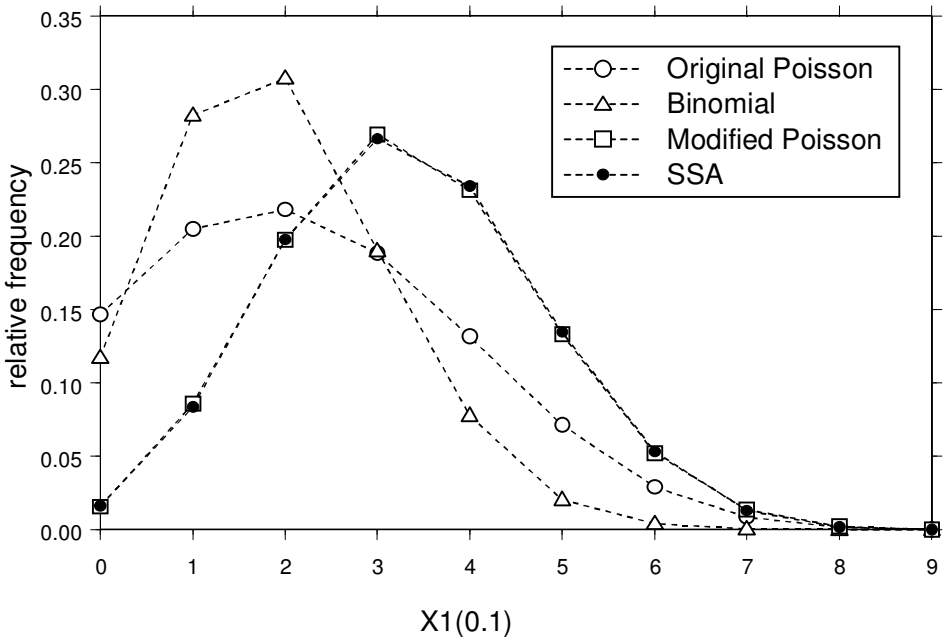

0.012
0.010
0.008
0.006
0.004
0.002
0.000
tau
400
600
800
1000
1200
1400
1600
1800
2000
t

Fig 4


0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
relative frequency
0
1
2
3
4
5
6
7
8
9
X1(0.1)
Original Poisson
Binomial
Modified Poisson
SSA