The Adaptive Explicit-Implicit Tau-Leaping Method with Automatic Tau Selection

Yang Cao*

Department of Computer Science, 660 McBryde Hall, Virginia Tech, Blacksburg, VA 24061

Daniel T. Gillespie

Dan T. Gillespie Consulting, 30504 Cordoba Place, Castaic, CA 91384

Linda R. Petzold

Department of Computer Science, University of California, Santa Barbara, Santa Barbara, CA 93106

Abstract

The existing tau-selection strategy, which was designed for explicit tau-leaping, is here modified to apply to implicit tau-leaping, allowing for longer steps when the system is stiff. Further, an adaptive strategy is proposed that identifies stiffness and automatically chooses between the explicit and the (new) implicit tau-selection methods to achieve better efficiency. Numerical testing demonstrates the advantages of the adaptive method for stiff systems.

*Author to whom correspondence should be addressed. Email: ycao@cs.vt.edu.

1 Introduction

In recent years, concerns over stochastic effects resulting from the small numbers of certain reactant molecules in microscopic systems[1–4] have called for accurate and efficient stochastic simulation methods. The fundamental simulation method is Gillespie's Stochastic Simulation Algorithm (SSA).[5,6] Although progress[7,8] has been made to improve the efficiency of implementations of the SSA, as an exact procedure that simulates every reaction, it is necessarily inefficient for most realistic problems. The tau-leaping method[9] has been proposed to improve the efficiency. By using a Poisson approximation, the tau-leaping method can "leap over" many fast reactions and approximate the stochastic behavior of the system very well. The tau-leaping method makes a natural connection between the SSA in the discrete stochastic regime and the explicit Euler method applied to the chemical Langevin equation in the continuous stochastic regime and to the reaction rate equation (RRE) in the continuous deterministic regime. In this sense, the tau-leaping method is ideal for multiscale stochastic simulation.

Stiffness reflects the presence of multiple timescales, the fastest of which are stable. Stiffness is a well-known challenge in the deterministic simulation of chemically reacting systems. For stiff systems, the stepsize of explicit methods must be restricted to maintain numerical stability. In the case of discrete stochastic simulation using the explicit tau-leaping method, which limits to the explicit Euler method as the population of each chemical species becomes very large, the stepsize must be similarly restricted to maintain numerical stability.[10] Implicit tau-leaping methods have been proposed to solve this problem, in particular the implicit tau method[11] and the trapezoidal tau method.[12] Convergence and stability properties for fixed stepsizes of the explicit and implicit tau methods have been studied.[13] Real-world applications will require the adaptive selection of the stepsize τ. This has been studied recently in Cao et. al.[14] Although these strategies have led to practical and efficient τ-selection for nonstiff discrete stochastic systems, when applied to stiff problems they result in unnecessarily small τ values. A τ-selection formula that reflects the enhanced stability of implicit methods is thus needed.

Another approach to dealing with stiffness in the accelerated simulation of discrete stochastic systems is to make use of a stochastic version of the quasi steady state or partial equilibrium assumptions.[15–17] In the deterministic case, the quasi steady state approximation assumes that on the time scale of interest, the instantaneous rates of change for some intermediate species are approximately equal to zero. The partial equilibrium approximation assumes that some fast reactions are always in equilibrium. In many cases these two assumptions are equivalent. The quasi-steady state approximation focuses on the state, while the partial equilibrium approximation concentrates on the reactions. The quasi steady state approximation was extended to the stochastic quasi steady state approximation (SQSSA),[18] while the partial equilibrium approximation was extended to the slow-scale SSA method.[19] Both were shown to be very useful in accelerating stochastic simulation. Since they are very similar, we will focus on the partial equilibrium approximation. When a system is in a partial equilibrium state, the system dynamics is determined mainly by the slow reaction channels. This observation leads to the slow-scale SSA method.[19] In the present work, we will apply this observation to the implementation of implicit tau-leaping methods to yield an efficient τ-selection formula for stiff systems.

A complex system may not always remain in a partial equilibrium state. When the system is not in partial equilibrium, it is necessary to simulate the fast reaction channels accurately to reflect the corresponding dynamical change. However, when the fast reaction channels reach the partial equilibrium state, it is more efficient to focus on the slow-scale reaction channels. This can be achieved by keeping dynamic lists of fast and slow reaction channels and verifying equilibrium conditions during the simulation. However, the frequent house-keeping operations can be computationally expensive and can impact the simulation efficiency. Moreover, when the system exhibits modes between the fast and slow ones, the partial equilibrium method is not applicable. Here we propose a somewhat less rigorous, but more practical method. By comparing the stepsizes given by the implicit τ-selection formula and the explicit τ-selection formula, our method dynamically switches between implicit and explicit tau-leaping methods without explicitly distinguishing the fast and slow scales. This switching strategy of comparing the stepsizes given by the explicit and implicit methods has been successful in the numerical solution of ODEs. Interested readers are referred to L. Petzold,[20] which outlines the strategy for the automatic explicit/implicit code LSODA[21] that is available in Mathematica.

The outline of this paper is as follows. In Section 2 we briefly review the SSA method and tau-leaping methods. In Section 3 we introduce the tau-selection formula for stiff systems. In section 4 the adaptive explicit-implicit tau-leaping method is described. Numerical experiments are presented in Section 5.

2 Simulation Algorithms for Chemical Kinetics

2.1 SSA and Tau-leaping Methods

Suppose the system involves N molecular species $\{S_1, \ldots, S_N\}$. The state vector is denoted by $X(t) = (X_1(t), \ldots, X_N(t))$, where $X_i(t)$ is the number of molecules of species S_i at time t. M reaction channels $\{R_1, \ldots, R_M\}$ are involved in the system. Assume that the system is well-stirred and in thermal equilibrium. The dynamics of reaction channel R_j is characterized by the *propensity function* a_j and by the *state change vector* $\nu_j = (\nu_{1j}, \ldots, \nu_{Nj})$: $a_j(x)dt$ gives the probability that one R_j reaction will occur in the next infinitesimal time interval $[t, t+dt)$, and ν_{ij} gives the change in the S_i molecular population induced by one R_j reaction.

The dynamics of the system can be simulated by the SSA method.[5,6] With $X(t) = x$, let $a_0(x) = \sum_{j=1}^{M} a_j(x)$. On each step, SSA generates two random numbers r_1 and r_2 in $U(0,1)$, the uniform distribution on the interval $(0,1)$. The time for the next reaction to occur is given by $t + \tau$, where τ is given by

$$\tau = \frac{1}{a_0(x)} \log(\frac{1}{r_1}). \tag{1}$$

The index j for the next reaction is given by the smallest integer satisfying

$$\sum_{l=1}^{j} a_l(x) > r_2 a_0(x). \tag{2}$$

The system states are updated by $X(t+\tau) = x + \nu_j$. The simulation proceeds to the next occurring time, until it reaches the final time.

In principle, the SSA could be used to simulate all of the chemical species and reactions. But because it must proceed one reaction at a time, it is much too slow for most practical problems. Gillespie[9] has proposed a scheme called *tau-leaping* to accelerate the SSA. The basic idea of the tau-leaping method is to ask the question: How many times does each reaction channel fire in each subinterval? In each step, the tau-leaping method can proceed with many reactions. This is achieved at the cost of some accuracy. Define

$$\begin{aligned} K_j(\tau; x, t) \quad &= \quad \text{the number of times, given } X(t) = x, \text{ that reaction channel } R_j \\ &\quad\quad \text{will fire in the time interval } [t, t+\tau) \quad (j = 1, \ldots, M). \end{aligned} \tag{3}$$

Tau-leaping assumes the **Leap Condition**: *For the current state x, require τ to be small enough that the change in the state during $[t, t+\tau)$ will be so small that no propensity function will suffer an appreciable change in its value.* $K_j(\tau; x, t)$ is then well approximated by the Poisson random variable with mean and variance $a_j(x)\tau$.

$$K_j(\tau; x, t) = P(a_j(x)\tau) \quad (j = 1, \ldots, M). \tag{4}$$

The basic tau-leaping method proceeds as follows: Choose a value for τ that satisfies the Leap Condition. Generate for each $j = 1, \ldots, M$ a sample value k_j of the Poisson random variable $P(a_j(x)\tau)$, and update the state by

$$X(t+\tau) = x + \sum_{j=1}^{M} k_j \nu_j. \tag{5}$$

If the populations of all reactant species are sufficiently large, the Poisson random variable $P(a_j(x)\tau)$ can be approximated by its mean $a_j(x)\tau$, and the basic (explicit) tau-leaping formula becomes the explicit Euler formula for the deterministic reaction rate equation. But the explicit Euler formula is known to be inefficient when applied to stiff problems. The explicit tau-leaping formula has the same difficulty. The implicit tau formula[11] has been proposed to handle the stiffness, and is given by

$$\hat{X}^{(it)}(t+\tau) = x + \sum_{j=1}^{M} \nu_j \left[P(a_j(x)\tau) - \tau a_j(x) + \tau a_j \left(\hat{X}^{(it)}(t+\tau) \right) \right]. \tag{6}$$

Newton's method is used to solve (6) for $X^{(it)}(t+\tau)$. Note that here the $X^{(it)}(t+\tau)$ are floating point values. In the simulation, we change them to integers by rounding the quantity in brackets on the right side of (6) to the nearest integer. But to simplify the analysis, here we will use (6) as written. It has been demonstrated[11] that the implicit tau formula allows much larger stepsizes than the explicit tau formula, when applied to stiff stochastic systems. A trapezoidal tau formula has also been proposed[12] as follows:

$$\hat{X}^{(tr)}(t+\tau) = x + \sum_{j=1}^{M} \nu_j \left[P(a_j(x)\tau) - \frac{\tau}{2} a_j(x) + \frac{\tau}{2} a_j \left(\hat{X}^{(tr)}(t+\tau) \right) \right]. \tag{7}$$

The difference between (6) and (7) is only the coefficients of the $a_j(x)$ and $a_j(\hat{X})$ terms. But (7) has better accuracy when applied to some stiff problems.[12] Implementation details are similar.

2.2 Nonnegative Poisson Tau-Leaping and Tau Selection Formulas

It has been found in the simulation of certain systems in which some consumed reactant species are present in small numbers that the original Poisson tau-leaping method may drive some reactant populations negative. Several strategies have been proposed to get around this problem. Tian and Burrage,[22] and independently Chatterjee et al.,[23] proposed approximating the unbounded Poisson random numbers K_j with bounded binomial random numbers. But it turns out that it is usually not the unboundedness of the Poisson k_j's that produces negative populations, but rather the lack of coordination in tau-leaping between different reaction channels that separately decrease the population of a common species. An improvement[24] to the binomial tau-leaping has been proposed to use multinomial random numbers to generate K_j values. To solve the same problem using the original Poisson tau-leaping, Cao et al.[25] have proposed a different approach that resolves this difficulty and also establishes a smooth connection with the SSA.

The Nonnegative Poisson tau-leaping algorithm[25] is based on the fact that negative populations typically arise from multiple firings of reactions that are only a few firings away from consuming all the molecules of one of their reactants. To focus on those reaction channels, the modified tau-leaping algorithm introduces a second control parameter n_c, a positive integer that is usually set somewhere between 5 and 20. Any reaction channel with a positive propensity function that is currently within n_c firings of exhausting one of its reactants is then classified as a *critical* reaction. The modified algorithm chooses τ in such a way that no more than one firing of *all* the critical reactions can occur during the leap. Essentially, the algorithm simulates the *critical* reactions using an adapted (and thus not quite exact) version of the SSA, and the remaining *non-critical* reactions using the previously described Poisson tau-leaping method. Since no more than one firing of a critical reaction can occur during a leap, the probability of producing a negative population is reduced to nearly zero. On those rare occasions when a negative population does arise (from firings of some non-critical reaction), the leap can simply be rejected and repeated with τ reduced by half, or else the simulation can be started over using a larger value for n_c. The reasoning, examples and implementation details can be found in Ref.[25]

In order for tau-leaping to be practical, we need to have a procedure for quickly determining the largest value of τ that is compatible with the Leap Condition. Gillespie[9] originally proposed that the Leap Condition could be considered satisfied if the expected change in each propensity function $a_j(x)$ during the leap were bounded by $\epsilon a_0(x)$, where ϵ is an error control parameter ($0 < \epsilon \ll 1$). Later, Gillespie and Petzold[26] showed that the largest value of τ that satisfies this requirement can be estimated by requiring that the *mean* and *standard deviation* of the expected change in $a_j(x)$ in the time period τ be bounded by $\epsilon a_0(x)$ for all j. In a more recent work, Cao et al.[14] proposed an improvement of this tau selection formula. The new tau-selection formula is given by

$$\tau = \min_{i \in I_{rs}} \left\{ \frac{\max\{\epsilon x_i/g_i, 1\}}{|\mu_i(x)|}, \frac{\max\{\epsilon x_i/g_i, 1\}^2}{\sigma_i^2(x)} \right\}, \tag{8}$$

where I_rs is the set of indices of all *reactant species*, g_i is given by a formula which guarantees that bounding the relative change of states is sufficient for bounding the relative change of propensity functions, and μ_i, σ_i are given by

$$\mu_i(x) \triangleq \sum_{j \in J_{ncr}} \nu_{ij} a_j(x), \quad \forall i \in I_{rs}, \tag{9a}$$

$$\sigma_i^2(x) \triangleq \sum_{j \in J_{ncr}} \nu_{ij}^2 a_j(x), \quad \forall i \in I_{rs}, \tag{9b}$$

where J_{ncr} is the set of indices of all *non-critical reactions*.

3 Tau Selection Formula for Stiff Systems

For a stiff problem solved with an implicit tau method, the tau selection formula (8) can generate unnecessarily small stepsizes. For example, suppose we have a system that consists of three reactions as follows:

$$S_1 \underset{c_2}{\overset{c_1}{\rightleftharpoons}} S_2 \xrightarrow{c_3} S_3, \tag{10}$$

where the propensities are given by $a_1(x) = c_1x_1$, $a_2(x) = c_2x_2$, $a_3(x) = c_3x_2$ and $c_1, c_2 >> c_3$. Thus the first two reactions are much faster than the third. We also assume that the first two reaction channels are near equilibrium, which is represented mathematically by $c_1x_1 \approx c_2x_2$. According to the slow-scale SSA,[19] the dynamics of this system can be approximated by the reduced system:

$$\hat{S}_t \xrightarrow{\hat{c}_3} S_3, \tag{11}$$

where the population of the virtual species S_t is given by $X_t = X_1+X_2$, and the reaction rate is determined by $\hat{c}_3 = \frac{c_3c_1}{c_1+c_2}$. If one has to apply a tau-leaping method to the reduced system, the τ-selection formula (8) generates a stepsize $\hat{\tau} = \frac{\epsilon}{\hat{c}_3}$. However, for the original system, the same τ-selection formula generates a stepsize close to $\tau_{orig} = \dfrac{\epsilon^2}{\max\{c_1^2, c_2^2\}}$. It is easy to verify that τ_{orig} is much smaller than $\hat{\tau}$. The smaller stepsize τ_{orig} is actually appropriate for explicit tau-leaping methods, where the stepsize of such a system will be limited by the stability requirement of the explicit method. However, properly chosen implicit tau-leaping methods do not have such a stability limitation, and the much larger $\hat{\tau}$ is the appropriate stepsize.

The reason that τ_{orig} is much smaller than $\hat{\tau}$ is due to the fact that the τ-selection formula (8) is based on the criteria that the relative change of each propensity function be bounded by ϵ, which can be formulated as:

$$|\Delta a_j(x)| \leq \epsilon a_j(x), \quad j = 1, \ldots, M. \tag{12}$$

However, if the fast reaction channels are near a partial equilibrium state, as pointed out in the slow-scale SSA theory, it is the *average* values of the propensities of the fast reaction channels that affect the dynamics of the slow reaction channels. The relative change of the propensities for those equilibrium reaction channels may be large, but that only affects the accuracy of the variance for the fast variables. The dynamics of the system is determined mainly by those reaction channels that are not in equilibrium. Thus in the stepsize control, we can ignore those reaction channels that are in partial equilibrium. One consequence of this relaxed requirement is that the variance of those species populations that get changed by the equilibrium reactions may not be accurate. For instance, if we require only that the relative change of the propensities (or state variables) *resulting from* R_3 be bounded in the above example, the τ value will be given by $\tau_{new} = \frac{\epsilon}{c_3}$. Note that τ_{new} is still smaller than $\hat{\tau}$ (because $\hat{c}_3 < c_3$), but it is much larger than τ_{orig}.

3.1 Partial Equilibrium Condition

To handle the simulation for a general model, we need an algorithm. The first important question is: How should we determine whether or not some reaction channels are in partial equilibrium? This question is complicated in general. Here we focus only on reversible reaction pairs (Note that in many biochemical systems, partial equilibrium is reached in reversible reaction pairs based on the *detailed balance* assumption. The general situation will be a topic for future research.). Let us consider a pair of reversible reaction channels R_+ and R_- to be in partial equilibrium if the corresponding propensities $a_+(x)$ and $a_-(x)$ are relatively close to each other. Because of the fluctuation in the system, they will not be exactly equal to each other. But their difference should be much smaller than each of them. We formulate this condition as:

$$|a_+(x) - a_-(x)| \leq \delta \min\{a_+(x), a_-(x)\}, \tag{13}$$

where δ is a small positive number. This is a natural generalization from the concept of equilibrium in a deterministic system. In practice we usually choose δ to be around 0.05. Our experience indicates that it gives reasonably good results. Another way is to use the formula (11) in Samant and Vlachos,[27] in which an extra term was introduced to account for randomness of propensities due to the fluctuation. Currently, for simplicity we only use (13).

Note that the equilibrium criteria (13) may fail when some reactants in the reversible reaction channels are present with a very small population. In that situation, due to stochastic fluctuation, the population of these reactants may drop to zero, which makes the propensities $a_+(x)$ or $a_-(x)$ suddenly become zero. (13) will not be satisfied in that situation. But when the population of the corresponding reactant becomes nonzero, (13) will be satisfied then. This will lead to frequent switches for this pair of reversible reaction channels between equilibrium and non-equilibrium states and is not desired in the simulation. We are working to develop more robust criteria for identifying quasi-equilibrium situations.

3.2 Tau Selection Formula for Implicit Methods

If all reaction channels in a system are in equilibrium, the system should remain in equilibrium because there is no other reaction channel to drive it away from the equilibrium state. Thus, once the system reaches equilibrium, the stepsize τ can be chosen to be very large. In this case, the inherent randomness may lead to relatively large changes for the propensities. But the implicit method will keep the simulation trajectory close to the equilibrium manifold. A reasonable stepsize control strategy doesn't have to consider the corresponding fluctuations. Similarly, when only some of the reaction channels are in equilibrium, the dynamics of the system will be driven by the reaction channels that are not in equilibrium. Thus the stepsize control for stiff system should *limit the relative changes resulting from the nonequilibrium reaction channels to within ϵ.*

The original τ-selection formula (8) restricts the relative change of the state variables due to all reaction channels. To find the proper τ value for the implicit methods in the presence of partial equilibrium subsystems, we need only further restrict the index set used in the τ-selection formula (8). The original index set is given by J_{ncr}, the set of indices of all *non-critical reactions.* Let J_{ne} denote the set of indices of the reaction channels that are not in partial equilibrium. Then $J_{necr} = J_{ne} \cap J_{ncr}$ represents the set of indices of all reaction channels that are neither critical nor in partial equilibrium. The τ-selection formula for the implicit methods is then given by

$$\tau^{(im)} = \min_{i \in I_{rs}} \left\{ \frac{\max\{\epsilon x_i/g_i, 1\}}{|\mu_i^{(im)}(x)|}, \frac{\max\{\epsilon x_i/g_i, 1\}^2}{[\sigma_i^{(im)}(x)]^2} \right\}, \tag{14}$$

where $\mu_i^{(im)}$, $\sigma_i^{(im)}$ are given by

$$\mu_i^{(im)}(x) \triangleq \sum_{j \in J_{necr}} \nu_{ij} a_j(x), \quad \forall i \in I_{rs}, \tag{15a}$$

$$[\sigma_i^{(im)}(x)]^2 \triangleq \sum_{j \in J_{necr}} \nu_{ij}^2 a_j(x), \quad \forall i \in I_{rs}. \tag{15b}$$

For a system where all reaction channels are in equilibrium, the above formula (14) will give $\tau = \infty$. Of course in practice the stepsize will always have an upper limit, for example, the simulation end time T.

Note that here we do not verify whether the fast reactions are MUCH faster than the other reactions. This is different from the slow-scale SSA method. In the slow-scale SSA, the difference between the time scales is very important. This is because in the slow-scale SSA method, fast reaction channels are represented by their behavior at $t = \infty$ in the virtual fast process. Their dynamical changes will not be simulated by the slow-scale SSA simulation. However, implicit tau-leaping methods generate the number of reactions K_j for all reaction channels. The dynamical changes resulting from those reaction channels in equilibrium are still simulated. They only get ignored when estimating the stepsize τ in formula (14). Thus if some reaction channels in equilibrium are not much faster than some non-equilibrium reaction channels, with the stepsize calculated from only the non-equilibrium reaction channels, the fluctuation resulting from those reaction channels that are slow but in equilibrium, will still be well simulated in the implicit tau-leaping method.

4 Adaptive Tau-Leaping Method

If a problem is known to be stiff, the implicit tau-leaping methods can be applied along with the stiff τ-selection formula (14). If a problem is nonstiff, it is better to use explicit tau-leaping because it is cheaper per step. What if we do not have this knowledge before the simulation? Or what if the system presents dynamical behavior such that in one time period it is stiff but in another time period it is nonstiff? How do we judge whether or not a system is stiff in a simulation? With the two τ-selection formulas, we can easily construct an adaptive simulation strategy to automatically switch between explicit and implicit tau-leaping methods: If the τ (in the following it is denoted by $\tau^{(ex)}$ since it is used for explicit tau-leaping method) calculated from formula (8) is much smaller than $\tau^{(im)}$ calculated from formula (14), we consider the system to be stiff and use an implicit tau-leaping method with stepsize $\tau^{(im)}$. Otherwise, it is more efficient to use the explicit tau-leaping method with stepsize $\tau^{(ex)}$. Combining this strategy with the nonnegative tau-leaping method, we have the new algorithm as follows:

Adaptive Tau-Leaping Algorithm:

1. In state x at time t, identify the currently critical reactions. This is done by first estimating for each reaction R_j with $a_j(x) > 0$ the maximum number of times L_j that R_j can fire before exhausting one of its reactants:[22,23]

$$L_j = \min_{i \in [1,N];\, \nu_{ij} < 0} \left[\frac{x_i}{|\nu_{ij}|} \right]. \tag{16}$$

   Here the minimum is taken over only those index values i for which $\nu_{ij} < 0$, and the brackets denote "greatest-integer-in". Any reaction R_j with $a_j(x) > 0$ is deemed *critical* if $L_j < n_c$. (We will normally take $n_c = 10$.)

2. With a value chosen for ϵ (we normally take ϵ in the range 0.03 to 0.05), compute the *candidate* time leaps $\tau^{(ex)}$ by using the explicit tau-selection formula (8) and $\tau^{(im)}$ by using the implicit tau-selection formula (14). To calculate (14), formula (13) is applied to select partial equilibrium reaction channels.

3. If $\tau^{(im)}$ is larger than $N_{stiff} * \tau^{(ex)}$, the system is considered to be stiff. An implicit tau method such as implicit tau-leaping or trapezoidal tau-leaping is chosen and $\tau_1 = \tau^{(im)}$. Otherwise, an explicit tau-leaping is chosen and $\tau_1 = \tau^{(ex)}$. Usually we choose $N_{stiff} =$ 100.

4. If τ_1 is less than some small multiple (which we usually take to be 10) of $1/a_0(x)$, abandon tau-leaping temporarily, execute some modest number (which we usually set as 100 if the previous step uses the SSA or the explicit tau-leaping method, or 10 if the previous step uses the implicit tau-leaping methods) of single-reaction SSA steps, and return to step 1. Otherwise, proceed to step 5.

5. Compute the sum $a_0^c(x)$ of the propensity functions of all the *critical* reactions. Generate a *second candidate* time leap τ_2 as a sample of the exponential random variable with mean $1/a_0^c(x)$. As thus computed, τ_2 tentatively estimates the time to the next *critical* reaction.

6. Choose τ and set the number of firings k_j of each reaction R_j accordingly:

   (a) If $\tau_2 > \tau_1$, take $\tau = \tau_1$. For all critical reactions R_j set $k_j = 0$ (no critical reactions will fire during this leap). For all non-critical reactions R_j, generate k_j using the tau-leaping method chosen in step 3 with stepsize τ.

   (b) If $\tau_2 \leq \tau_1$, take $\tau = \tau_2$. Generate j_c as a sample of the integer random variable with point probabilities $a_j(x)/a_0^c(x)$, where j runs over the index values of the *critical* reactions only. (The value of j_c identifies the next critical reaction, the *only* critical reaction that will fire in this leap.) Set $k_{j_c} = 1$, and for all other critical reactions R_j set $k_j = 0$. For all the non-critical reactions R_j, if the explicit method was chosen in step 3, or if the implicit method was chosen in step 3 but $\tau_2 \leq \tau^{(ex)}$, use the explicit tau-leaping method, generating k_j as a sample of the Poisson random variable with mean $a_j(x)\tau$; otherwise generate k_j using the implicit tau-leaping method with stepsize τ.

7. If there is a negative component in $x + \sum_j k_j \nu_j$, reduce τ_1 by half, and return to step 4. Otherwise, leap by replacing $t \leftarrow t + \tau$ and $x \leftarrow x + \sum_j k_j \nu_j$; then return to step 1, or else stop.

This method can efficiently switch among the SSA, the explicit tau-leaping method and the implicit tau-leaping method. However, note that it cannot efficiently handle situations in which some species are present with a very small population but are involved in some very fast reaction channels. Since we treat the corresponding reaction channel as *critical*, it will be handled by an SSA type simulation. This is a drawback of this algorithm, as compared to slow-scale SSA. Further improvement is still under research.

5 Numerical Examples

We tested our method on two examples.

5.1 Stiff Decaying Dimerization Example

The stiff decaying dimerization model was originally proposed in Ref.[9] and modified in Ref.[11] The stiff decaying dimerization model consists of three species S_1, S_2 and S_3 and four reaction channels:

$$\begin{aligned} S_1 &\xrightarrow{c_1} 0 \\ S_1 + S_1 &\underset{c_2}{\overset{c_3}{\rightleftharpoons}} S_2 \\ S_2 &\xrightarrow{c_4} S_3. \end{aligned} \tag{17}$$

This reaction was studied with nonstiff reaction rates[9] and stiff reaction rates.[11] In Rathinam et. al.,[11] the stiff reaction rates were chosen as $c_1 = 1,\ c_2 = 10,\ c_3 = 1000,\ c_4 = 0.1$, and initial condition was set as $x_1(0) = 400$, $x_2(0) = 798$ and $x_3(0) = 0$. This initial condition was chosen so that the reversible reaction pair

$$S_1 + S_1 \underset{c_2}{\overset{c_3}{\rightleftharpoons}} S_2 \tag{18}$$

is close to partial equilibrium. Here we have used the initial condition $x_1(0) = 10,000$, $x_2(0) = 0$ and $x_3(0) = 0$ so that initially the above reversible reaction pair is not in partial equilibrium. We simulated the system using the adaptive explicit-implicit tau-leaping method from $t_0 = 0$ to $T_f = 4$ with $\epsilon = 0.05$, and obtained the results shown in Figure 1. We focus here on the behavior of the stepsizes. In the beginning, the adaptive method chooses SSA for a very short time as shown in Figure 2. We call this short time interval the "SSA regime". Then explicit tau-leaping is chosen to simulate the fast reaction channel accurately (as shown in Figure 3). We call this time interval the "explicit tau regime". Most of the simulation is performed in the so-called "stiff regime" during which $\tau^{(im)}$ is used and the trapezoidal tau-leaping method is applied. The tau values over the whole time interval are shown in Figure 4. We can see a "jump" between the explicit and implicit methods. That is due to the switch process. We choose the implicit method when it can use a stepsize 100 times larger than that allowed by the explicit tau-leaping method. When it cannot maintain such a high efficiency, we switch back to the explicit tau-leaping method. Another interesting behavior is that after $t \approx 0.8$, the τ value is almost fixed at $\tau = 0.05$. We note that in this simulation, $\epsilon = 0.05$, $c_1 = 1$, and R_1 is a simple decay process. Applying the τ-selection formula on this reaction channel yields $\tau = 0.05$. Thus as shown in Figure 4, the stepsizes are eventually limited by one of the slow-scale reaction channels.

For this example, $10,000$ runs took $139,817$ seconds CPU time using the exact SSA throughout, $2,461$ seconds CPU time using the explicit tau-leaping method throughout with $\epsilon = 0.05$, and 112 seconds for the adaptive explicit-implicit tau-leaping method with $\epsilon = 0.05$. We also ran the two tau-leaping simulations with the error tolerance reduced by one-half. $10,000$ runs took $11,710$ seconds CPU time for the explicit tau-leaping method with $\epsilon = 0.025$, but only 72 seconds CPU time for the adaptive explicit-implicit tau-leaping method with $\epsilon = 0.025$. Table 1 shows the accuracy of the mean and deviation for each variable calculated from the ensembles. Note that the standard deviation of X_1 (a fast variable) is not accurate for the adaptive explicit-implicit tau-leaping method. This can be improved by the down-shifting strategy.[11,28] When applying this strategy, we applied the adaptive switch mechanism until very close to the final time. Then we stopped the switch mechanism and used only the explicit tau-leaping. In particular we used the criteria

$$t + \tau^{(im)} < T_f - 100(\tau^{(ex)} + 1/a_0) \tag{19}$$

to decide whether or not to turn on the switch mechanism. This condition is to make sure that at least 100 steps of explicit tau-leaping or SSA will be applied before the output of states. If condition (19) is satisfied, we turn on the switch mechanism; otherwise, we turn it off and the system will switch between the explicit tau-leaping and SSA. Since this turning-off occurs for only a relatively short time, it has little effect on the simulation efficiency. When we apply the down-shifting strategy, $10,000$ runs took 133 seconds CPU time for the adaptive explicit-implicit tau-leaping method with $\epsilon = 0.05$. and 106 seconds CPU time with $\epsilon = 0.025$. The tau values over the whole time interval, when down-shifting strategy is applied, are shown in Figure 5. We can see that the tau values jump when the simulation is close to the end. That is due to the turning-off of the adaptive switch mechanism. The accuracy of the mean and standard deviation for each variable is also shown in Table 1. We can see that the accuracy of the standard deviation is much improved for X_1. We also compared the histogram distance[29] between the SSA ensemble and the ensembles generated by the adaptive explicit-implicit tau-leaping method ($\epsilon = 0.025$) with or without the down-shifting strategy (Figure 6). The histogram distance between the SSA ensemble and the ensemble generated by the adaptive method with down-shifting (both 10,000 runs) is 0.1208, which is around the self-noise[29] due to the stochastic fluctuation. Thus the ensemble generated by the the adaptive method ($\epsilon = 0.025$) with down-shifting strategy is considered accurate enough.

5.2 A Simple Gene Expression Example

This model was originally proposed in Bundschuh et. al.[30] It describes a simple negative feedback in gene expression. The model is given as

$$\begin{array}{ccc} D^* & \xrightarrow{c_1} & D+M+R \\ M & \xrightarrow{c_2} & M+P \\ M & \xrightarrow{c_3} & \emptyset \\ P & \xrightarrow{c_1} & \emptyset \end{array} \tag{20}$$

with three pairs of fast reversible reactions.

$$D + R \underset{c_5}{\overset{c_{-5}}{\rightleftharpoons}} D^* \tag{21}$$

$$P + P \underset{c_6}{\overset{c_{-6}}{\rightleftharpoons}} P_2 \tag{22}$$

$$D + P_2 \underset{c_7}{\overset{c_{-7}}{\rightleftharpoons}} Q \tag{23}$$

where D and D^* denotes DNA in activated or inactivated state, M denotes mRNA, R denotes RNA polymerase, P denotes the protein and P_2 denotes the dimer of P. The parameters are given in Table 2. The initial condition is set as $x_{D^*} = 1$ and all the rest are zero. The problem time interval is from $t = 0$ to $T = 50,000$. Note that the populations of D and D^* switch between 0 and 1. The two pairs (21) and (23) are considered always near equilibrium. But since the population is low, they are also critical reactions. Our algorithm cannot handle this situation efficiently right now. Thus we use slow-scale SSA[19] to treat these two pairs and model them via the Michaelis-Menten kinetics.[31–33] Only the reaction pair (22) is subject to the equilibrium check.

In the numerical comparison, we apply slow-scale SSA to treat (21) and (23), and use the explicit tau-leaping method for the rest. The selection mechanism will always select SSA because the automatically generated stepsize for explicit tau-leaping is too small. However, the situation changes when the adaptive explicit-implicit tau-leaping is applied. When the pair (22) reaches partial equilibrium, the code chooses the implicit tau-leaping method with a much larger stepsize. Otherwise, it chooses SSA. In our experiments, the CPU time for 10000 simulation runs using SSA is 49,641 seconds; the CPU time for 10,000 simulation runs using adaptive explicit tau-leaping (with $\epsilon = 0.015$) is 51,487 seconds, while the time for 10,000 simulation runs using adaptive (explicit-implicit) tau-leaping method is 1,876 seconds with error tolerance $\epsilon = 0.015$ (down-shifting is applied). The slight increase of CPU time from SSA to the adaptive explicit tau-leaping is due to the overhead of the selection mechanism since SSA is always chosen. The probability distributions of P corresponding to the runs with the two methods (SSA and adaptive implicit-explicit tau-leaping) are shown in Figure 7. We can see that the results given by the adaptive tau-leaping method is very accurate. However, the simulation is about twenty-five times faster for this simple case.

To capture more details in the simulation, we collected simulation data in a typical simulation trajectory. Among the total 66,030 simulation steps, the implicit tau-leaping method was chosen in only 3,241 steps. However, in the total problem time interval of $50,000$ seconds the implicit tau-leaping method was used for $48,152$ seconds problem time. Only $1,848$ seconds of problem time was simulated by the 62,789 SSA steps. The average stepsize taken by the implicit tau-leaping method is 14.86 seconds, while the average stepsize taken by SSA is only 0.03 seconds. The implicit tau-leaping method is much more efficient than the SSA.

6 Discussion and Conclusion

In this paper we proposed a tau-selection formula for implicit tau-leaping methods to simulate stiff systems. Combined with the previously proposed tau-selection formula for explicit methods, we presented the adaptive explicit-implicit tau-leaping method. Through two simple examples, We demonstrate that the new method appears much more efficient than the explicit tau-leaping method for stiff systems.

This new method may lead to a multiscale stochastic simulation method for general biochemical systems. But there are still some issues to be resolved for the most realistic and efficient implementation. The most important is the accuracy for 'fast' variables. According to our numerical experiments, the new method generates accurate distributions for the slow variables. It may generate errors in the variance of the fast variables. This is known for the implicit tau-leaping method[11] and the multiscale SSA method.[28] As we have seen in the numerical experiments, the "down-shifting" strategy[11,28] can be used to improve that. However, rigorous analysis is still needed with regard to the implementation details. Another issue concerns the equilibrium condition. In this paper we have presented a simple test for whether a subsystem (mostly a pair of reversible reaction channels) is in equilibrium. We are aware that the corresponding theory requires further development. Also more numerical tests on complex systems need to be done to verify and improve the current algorithm. We plan to address these problems in our future research.

Acknowledgments

Support for LP and YC was provided by the U. S. Department of Energy under DOE award No. DE-FG02-04ER25621, by the National Science Foundation under NSF award, CCF-0326576 and NSF/NIGMS GM078993, by the Institute for Collaborative Biotechnologies through grant DAAD19-03-D-0004 from the U. S. Army Research Office, and by NIH GM075297 and NIH GM078989. Support for DG was provided by the University of California under Consulting Agreement 054281A20 with the Computer Science Department of its Santa Barbara campus; and by the California Institute of Technology, through Consulting Agreement 21E-1079702 with the Beckman Institute's Biological Network Modeling Center, and through Consulting Agreement 102-1080890 with the Control and Dynamical Systems Department pursuant to NIH Grant R01 GM078992 from the National Institute of General Medical Sciences.

References

[1] H. McAdams, A. Arkin. Stochastic mechanisms in gene expression. *Proc. Natl. Acad. Sci. USA*, 94:814–819, 1997.

[2] A. Arkin, J. Ross and H. McAdams. Stochasitc kinetic analysis of developmental pathway

bifurcation in phage λ-infected escherichia coli cells. *Genetics*, 149:1633–1648, 1998.

[3] H.H. McAdams and A. Arkin. It's a noisy business! *Trends in Genetics*, 15(2):65–69, 1999.

[4] N. Fedoroff and W. Fontana. Small numbers of big molecules. *Science*, 297:1129–1131, 2002.

[5] D. Gillespie. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.*, 22:403–434, 1976.

[6] D. Gillespie. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81:2340–61, 1977.

[7] M. Gibson and J. Bruck. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem. A*, 104:1876, 2000.

[8] Y. Cao, H. Li and L. Petzold. Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *J. Chem. Phys.*, 121:4059–67, 2004.

[9] D. Gillespie. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 115:1716, 2001.

[10] Y. Cao, L. Petozld, M. Rathinam and D. Gillespie. The numerical stability of leaping methods for stochastic simulation of chemically reacting systems. *J. Chem. Phys*, 121:12169–78, 2004.

[11] M. Rathinam, L. Petzold, Y. Cao, D. Gillespie. Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method. *J. Chem. Phys.*, 119:12784–94, 2003.

[12] Y. Cao and L. Petzold. Trapezoidal tau-leaping formula for the stochastic simulation of chemically reacting systems. *Proceedings of Foundations of Systems Biology in Engineering (FOSBE 2005)*, pages 149–152, 2005.

[13] M. Rathinam, L. Petzold, Y. Cao, D. Gillespie. Consistency and stability of tau leaping schemes for chemical reaction systems. *SIAM Multiscale Modeling*, 4:867–895, 2005.

[14] Y. Cao, D. Gillespie and L. Petzold. Efficient Stepsize Selection for the Tau-Leaping Method. *J. Chem. Phys.*, 124:044109, 2006.

[15] J. D. Ramshaw. Partial chemical equilibrium in fluid dynamics. *Phys. Fluid*, 23, 1980.

[16] M. Rein. The partial-equilibrium approximation in reacting flows. *Phys. Fluids A*, 4, 1992.

[17] L. A. Segel and M. Slemrod. The quasi-steady-state assumption: a case study in perturbation. *SIAM Review*, 31, 1989.

[18] C. Rao and A. Arkin. Stochastic chemical Kinetics and the quasi steady-state assumption: application to the Gillespie algorithm. *J. Chem. Phys.*, 118:4999–5010, 2003.

[19] Y. Cao, D. Gillespie and L. Petzold. The slow-scale stochastic simulation algorithm. *J. Chem. Phys.*, 122:014116, 2005.

[20] L. Petzold. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential-equations. *SIAM J. Sci. & Stat. Comput.*, 4:136–148, 1983.

[21] K. Radhakrishnan and A. C. Hindmarsh . Description and Use of LSODE, the Livermore Solver for Ordinary Differential Equations. LLNL report UCRL-ID-113855, December 1993.

[22] T. Tian and K. Burrage. Binomial leap methods for simulating stochastic chemical kinetics. *J. Chem. Phys.*, 121:10356–64, 2004.

[23] A. Chatterjee, D. G. Vlachos and M. A. Katsoulakis. Binomial distribution based tau-leap accelerated stochastic simulation. *J. Chem. Phys.*, 122:024112, 2005.

[24] M. Pettigrew and H. Resat. Multinomial tau-leaping method for stochastic kinetic simulation. *J. Chem. Phys.*, 126:084101, 2007.

[25] Y. Cao, D. Gillespie and L. Petzold. Avoiding negative populations in explicit tau leaping. *J. Chem. Phys.*, 123:054104, 2005.

[26] D. Gillespie and L. Petzold. Improved leap-size selection for accelerated stochastic simulation. *J. Chem. Phys.*, 119:8229–34, 2003.

[27] A. Samant, D. G. Vlachos. Overcoming stiffness in stochastic simulation stemming from partial equilibrium: A multiscale Monte Carlo algorithm. *J. Chem. Phys.*, 123:144114, 2005.

[28] Y. Cao, D. Gillespie and L. Petzold. Multiscale stochastic simulation algorithm with stochastic partial equilibrium assumption for chemically reacting systems. *J. Comput. Phys.*, 206:395–411, 2005.

[29] Y. Cao and L. Petzold. Accuracy limitations and the measurement of errors in the stochastic simulation of chemically reacting systems. *J. Comput. Phys.*, 212:6–24, 2006.

[30] R. Bundschuh, F. Hayot and C. Jayaprakash. The role of dimerization in noise reduction of simple genetic networks. *J. Theor. Biol.*, 220:261–269, 2003.

[31] L. Michaelis and M. L. Menten. *Biochem. Z*, 49:333–369, 1913.

[32] K. M. Plowman. *Enzyme Kinetics.* New York, McGraw-Hill, 1971.

[33] Y. Cao, D. Gillespie and L. Petzold. Accelerated Stochastic Simulation of the Stiff Enzyme-Substrate Reaction. *J. Chem. Phys.*, 123:144917, 2005.

List of Figure Captions

Figure 1: The dynamics of the stiff decaying dimerization model (17) in one SSA simulation. S_1 decreases and S_2 increases quickly in the transient period before $t = 0.125$. After that all species change slowly in the stiff regime.

Figure 2: The tau values in the "SSA regime" for the stiff decaying dimerization model (17) simulated by the adaptive algorithm ($\epsilon = 0.05$). Due to the low population of S_2 in this time interval, the simulation used SSA.

Figure 3: The tau values in the "explicit tau regime" for the stiff decaying dimerization model (17) simulated by the adaptive algorithm ($\epsilon = 0.05$). The population of S_2 is large enough for the explicit tau-leaping method. But the implicit tau-leaping method is not efficient enough, yet.

Figure 4: The tau values in the whole time interval [0, 4] for the stiff decaying dimerization model (17) simulated by the adaptive algorithm without the down-shifting strategy ($\epsilon = 0.05$). 'o' denotes that the corresponding simulation is done by implicit tau-leaping while '*' denotes that the corresponding simulation is done by explicit tau-leaping. The SSA regime is too short to be seen. The "jump" after $t = 0.125$ shows the switch from the explicit tau-leaping method to the implicit tau-leaping method.

Figure 5: The tau values in the whole time interval [0, 4] for the stiff decaying dimerization model (17) simulated by the adaptive algorithm with the down-shifting strategy ($\epsilon = 0.05$). 'o' denotes that the corresponding simulation is done by implicit tau-leaping while '*' denotes that the corresponding simulation is done by explicit tau-leaping. Compared to Figure 4, we can see a "jump back" near the final time, which shows a turning-off of the switch mechanism by the down-shifting strategy.

Figure 6: The histogram distance generated from ensembles (10,000 runs) by SSA (denoted by line) and by the adaptive explicit-implicit tau-leaping method with (denoted by 'o') and without (denoted by '*') the downshifting strategy (both for $\epsilon = 0.025$) for the population of species S_1 at the final time $T_f = 4$ in the stiff decaying dimerization model (17). The histogram obtained with the downshifting strategy is much closer to the SSA histogram than the one without the strategy.

Figure 7: The histogram distance generated from ensembles (10,000 runs) by SSA (solid

line with 'o') and by the adaptive explicit-implicit tau-leaping method (dashed line with '+') with the downshifting strategy (for $\epsilon = 0.015$) for the population of protein (P) at the final time $T_f = 50,000$ in the negative feedback gene expression model (20). The histogram distance is within the self-distance.

List of Tables

Method	error tolerance	Mean			Standard Deviation		
		x_1	x_2	x_3	x_1	x_2	x_3
SSA		617.1	1904.7	1230.2	24.43	31.99	29.48
Explicit Tau	0.05	617.3	1905.3	1230.0	30.08	33.60	29.73
	0.025	617.1	1904.9	1230.8	25.33	32.63	30.44
Adaptive Method	0.05	616.7	1901.0	1235.2	11.58	30.49	29.58
without down-shifting	0.025	617.0	1902.1	1232.0	12.12	30.97	29.50
Adaptive Method	0.05	616.1	1901.2	1234.7	30.07	33.64	29.64
with down-shifting	0.025	616.5	1902.7	1231.9	25.77	32.87	29.45

Table 1: Accuracy comparison for different methods and error tolerances.

Parameter	Value	Parameter	Value
c_1	0.0078	c_2	0.043
c_3	0.0039	c_4	0.0007
c_5	0.038	c_{-5}	0.3
c_6	0.05	c_{-6}	0.5
c_7	0.012	c_{-7}	0.9

Table 2: The parameter values in the gene expression model. These parameters are obtained from Bundschuh2003 et. al.[30] (except that c_6 is restored to 0.05 as in Arkin et. al.[2]

).

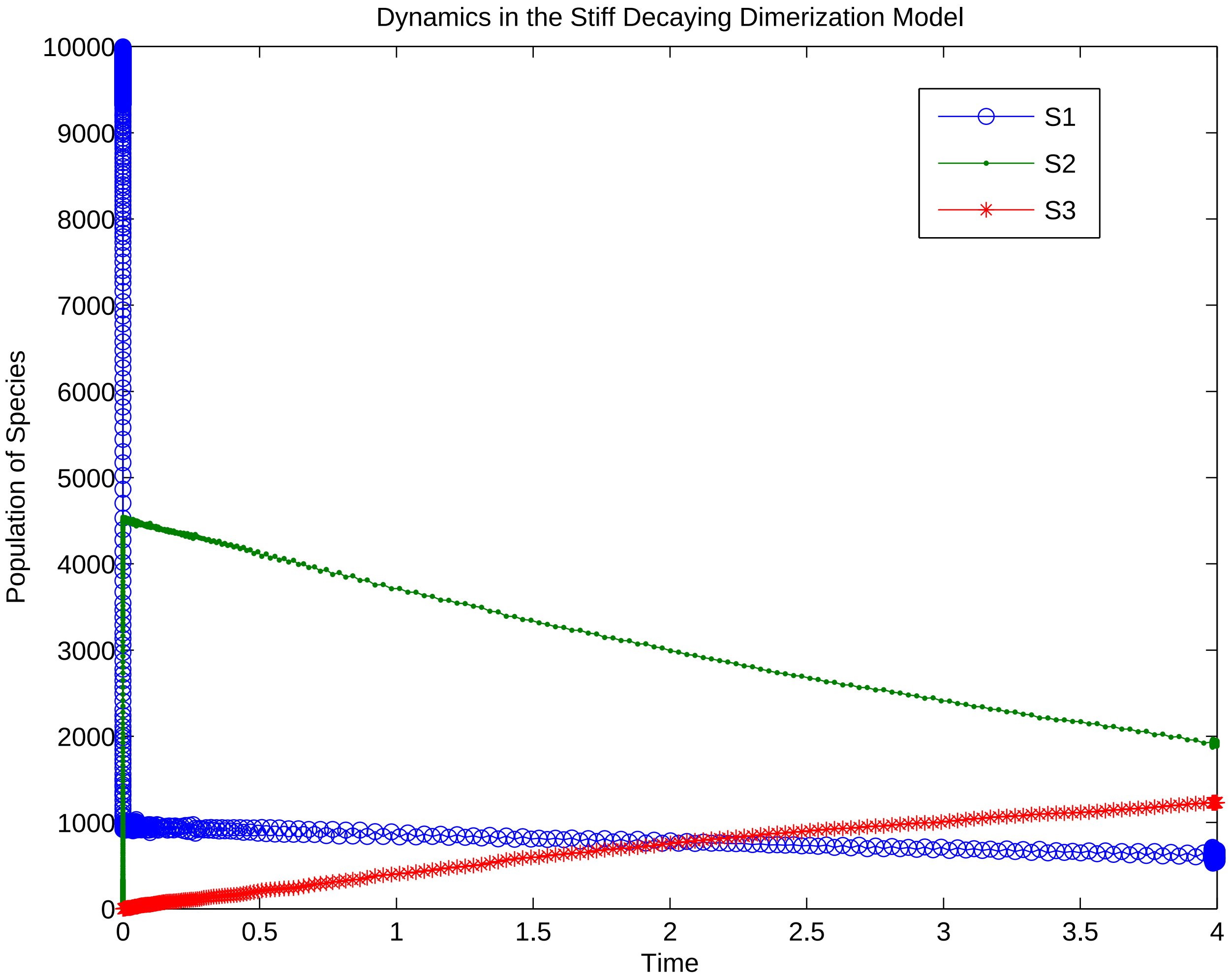
Dynamics in the Stiff Decaying Dimerization Model
Population of Species
Time
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
0
0.5
1
1.5
2
2.5
3
3.5
4
S1
S2
S3

Figure 1:

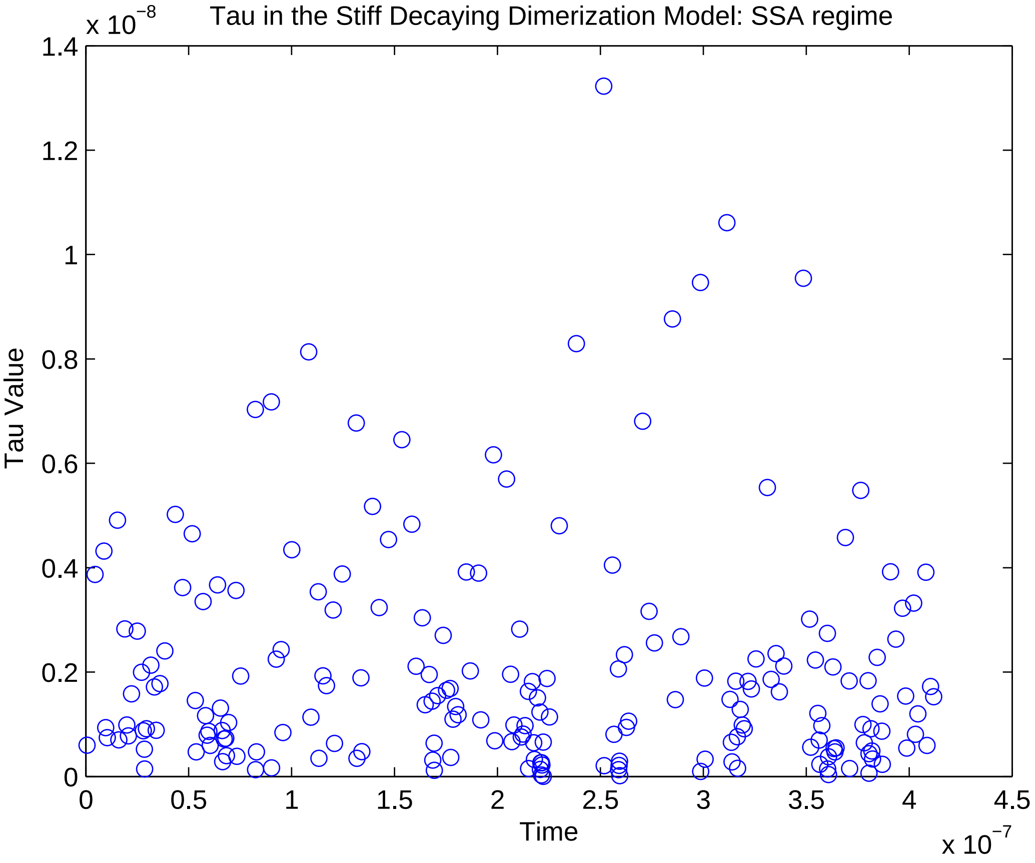
Tau in the Stiff Decaying Dimerization Model: SSA regime
x 10^{-8}
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Tau Value
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
Time
x 10^{-7}

Figure 2:

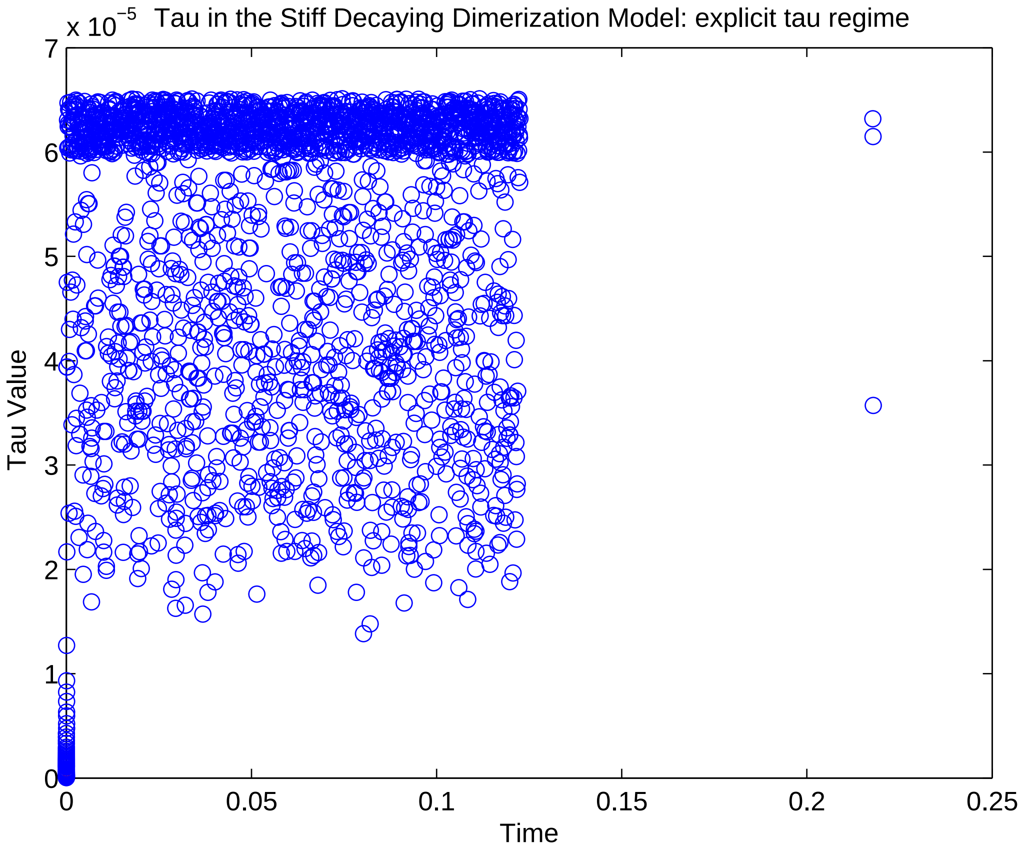
x 10^−5
Tau in the Stiff Decaying Dimerization Model: explicit tau regime
Tau Value
Time
0
1
2
3
4
5
6
7
0
0.05
0.1
0.15
0.2
0.25

Figure 3:

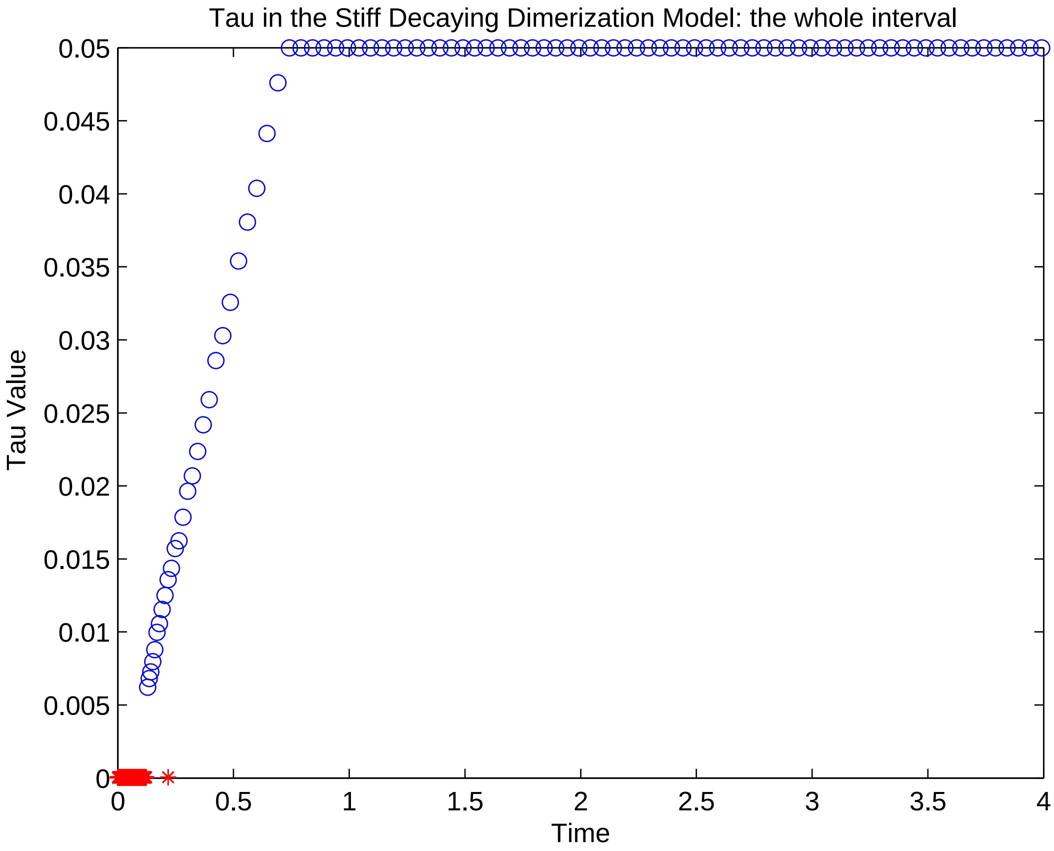
Tau in the Stiff Decaying Dimerization Model: the whole interval
Tau Value
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
0.5
1
1.5
2
2.5
3
3.5
4
Time

Figure 4:

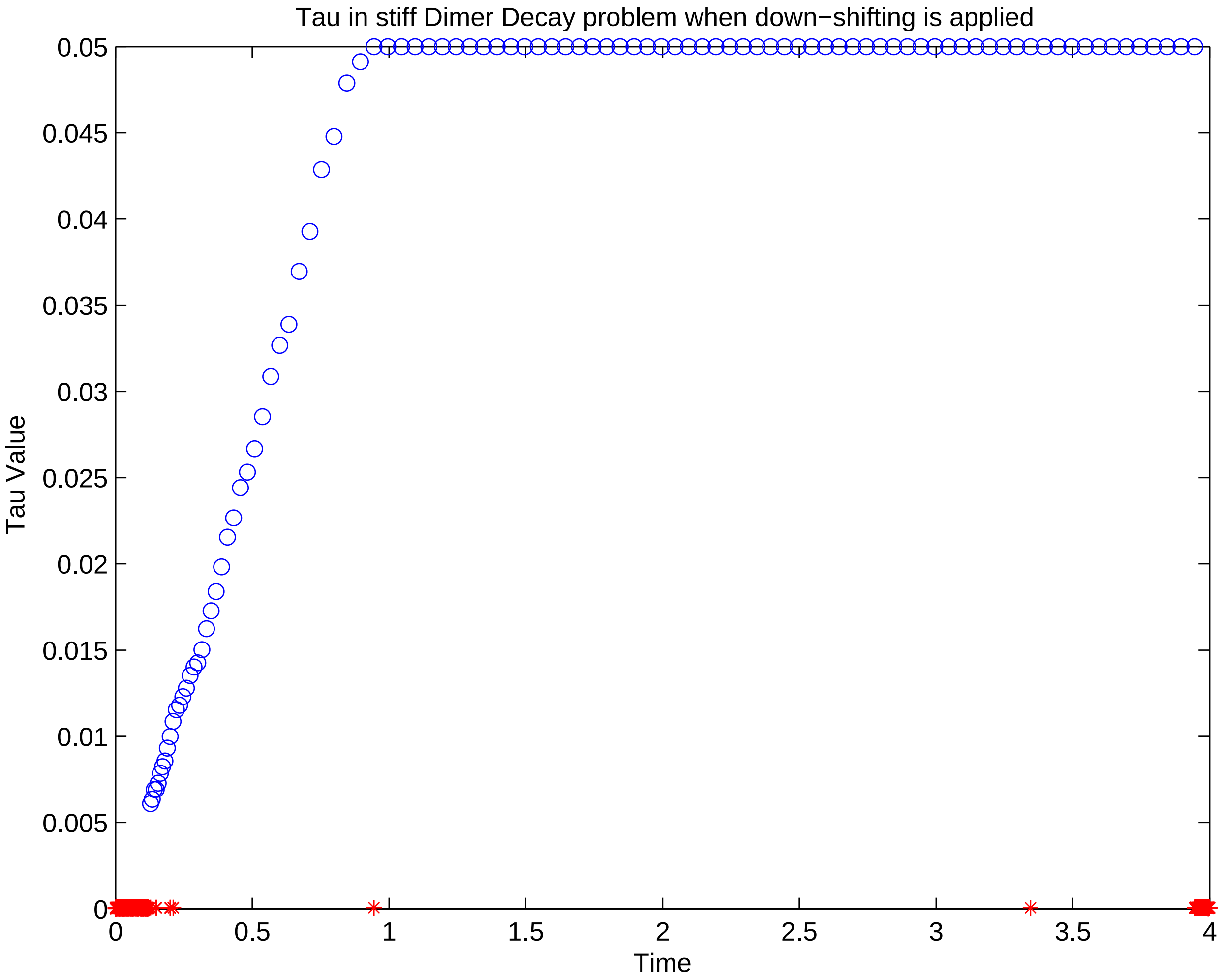
Tau in stiff Dimer Decay problem when down−shifting is applied
Tau Value
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
Time
0
0.5
1
1.5
2
2.5
3
3.5
4

Figure 5:

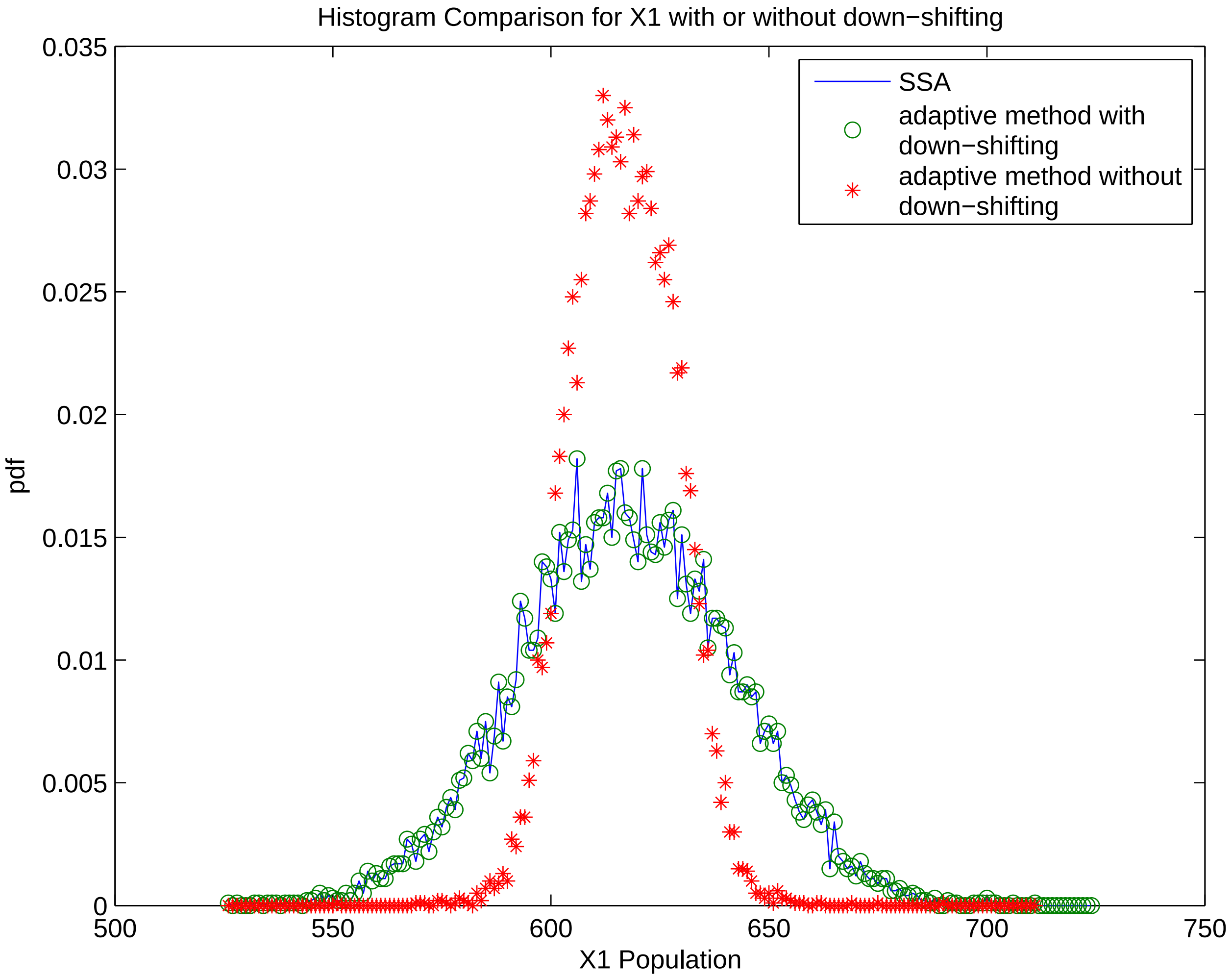
Histogram Comparison for X1 with or without down−shifting
pdf
X1 Population
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
500
550
600
650
700
750
SSA
adaptive method with down−shifting
adaptive method without down−shifting

Figure 6:

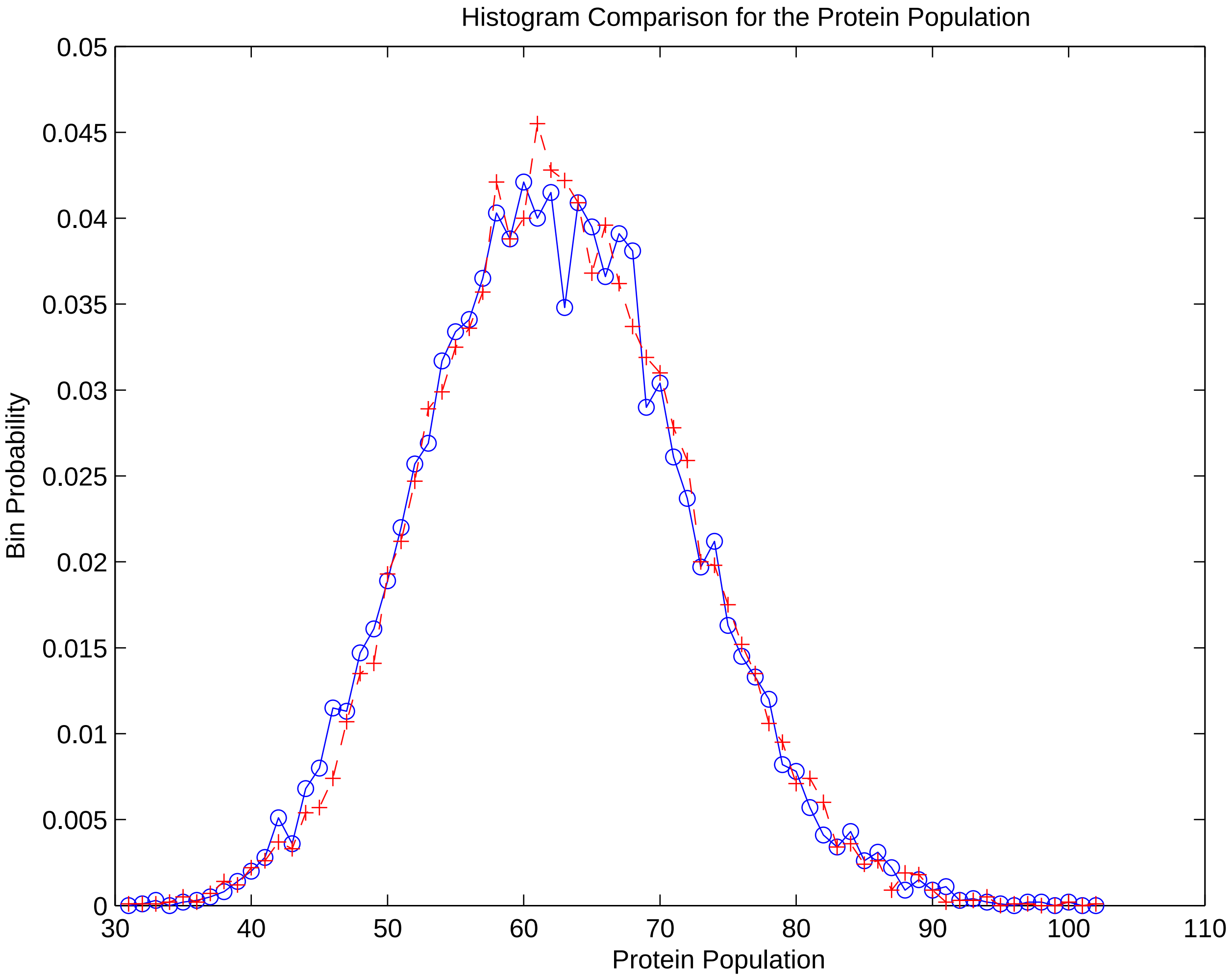
Histogram Comparison for the Protein Population
Bin Probability
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
30
40
50
60
70
80
90
100
110
Protein Population

Figure 7: