Simulation Algorithms for Continuous Time Markov Chain Models

H.T. Banks[1], Anna Broido[2], Brandi Canter[3], Kaitlyn Gayvert[4], Shuhua Hu[1], Michele Joyner[3], Kathryn Link[5]

[1]Center for Research in Scientific Computation
Center for Quantitative Sciences in Biomedicine
North Carolina State University
Raleigh, NC 27695-8212 USA

[2]Department of Mathematics
Boston College
Chestnut Hill, MA 02467-3806 USA

[3]Department of Mathematics and Statistics
East Tennessee State University
Johnson City, TN 37614-70663 USA

[4]Department of Mathematics
State University of New York at Geneseo
Geneseo, NY 14454 USA

[5]Department of Mathematics
Bryn Mawr College
Bryn Mawr, PA 19010-2899 USA

December 14, 2011

Abstract

Continuous time Markov chains are often used in the literature to model the dynamics of a system with low species count and uncertainty in transitions. In this paper, we investigate three particular algorithms that can be used to numerically simulate continuous time Markov chain models (a stochastic simulation algorithm, explicit and implicit tau-leaping algorithms). To compare these methods, we used them to analyze two stochastic infection models with different level of complexity. One of these models describes the dynamics of Vancomycin-Resistant Enterococcus (VRE) infection in a hospital, and the other is for the early infection of Human Immunodeficiency Virus (HIV) within a host. The relative efficiency of each algorithm is determined based on computational time and degree of precision required. The numerical results suggest that all three algorithms have similar computational efficiency for the VRE model due to the low number of species and small number of transitions. However, we found that with the larger and more complex HIV model, implementation and modification of tau-Leaping methods are preferred.

Key Words: stochastic simulation algorithm, explicit tau-leaping method, implicit tau-leaping method, VRE, HIV.

1 Introduction

Deterministic approaches involving ordinary differential equations to approximate large number discrete populations with a continuum, though widely used, have proven less useful when applied (often with little or no justification!) to small sample sizes. To address this issue, continuous time Markov chain (CTMC) models are often used when dealing with low species or population counts. There are a variety of stochastic algorithms that can be employed to simulate CTMC models. However, it appears that none of these algorithms is universally efficient in many problems of interest.

There are a plethora of applications in which the questions we investigate here arise. In addition to the infection models we use for illustration, similar stochastic models arise in just-in-time production networks, manufacturing and delivery, logistic/supply chains, and multi-scale (large/small) population models as well as network models in communications and security. A typical example is the agricultural (pork) production system investigated in [2]. There a stochastic transport model was used to study the impact of disturbances (introduction of diseases and other disruptions in the network) in production systens such as that depicted in Figure 1.

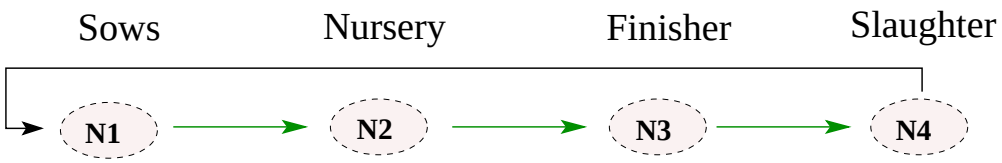

Figure 1: Aggregated agricultural network model.

The schematic represents a simplified swine production network with four levels of production nodes: (i) growers/sows (N1), (ii) nurseries (N2), (iii) finishers (N3), and (iv) processing plants/slaughterhouses (N4). At the grower or sow farms (N1), new piglets are born and weaned approximately three weeks after birth. The three-week old piglets are moved to nursery farms (N2) to mature for another seven weeks. They are then transferred to the finisher farms (N3) where they grow to full market size. This takes approximately twenty weeks. Once they reach market weight, the matured pigs are moved to the processors (slaughterhouses) (N4).

Such pork production industries operate under a "just-in-time" philosophy often employed in many manufacturing systems. In particular, feedstock and animals are grown in different areas and the animals are moved with precise timing from one farm to another, depending on their age. Any shocks propagate rapidly through such systems if conditions are conducive. For example, interruption to feed supply has much larger impact when farms have minimal surplus supplies. The maturity-based just-in-time movement of animals between farms serves as another vulnerability. Stopping movement (transportation disruptions due to weather, work stoppages due to illness, etc) of animals to and from a farm with animals infected by disease will have disruptive effects that

quickly spread throughout the system. Other interruptions occur when nurseries supplying farms have nowhere to send animals as they mature if the farms have not cleared their current animals for some reason. This will cause finishers and slaughterhouses to have supply interrupted. Randomness seen in the stochastic network model originates from random movement of discrete "individuals" from node to node. Analysis (see [2]) shows that, due to an averaging effect, these random effects become less important as the system size (number N of "individuals") increases. We observe that an application of such a stochastic transportation model to describe the system behavior should account for size of groups in which pigs are transported between nodes. If thousands of pigs are moved at a time, an appropriate notion of an "individual" in the context of the model might be a thousand pigs. Then treating each group of a thousand animals as unit would lead to a marked increase in magnitude of stochastic fluctuations seen at the "population" level. As a result, scaling in the model may result in vastly different stochastic fluctuations in model simulations. Thus one must exercise care in how data and "units" are formulated in modeling populations.

To further illustrate these practical modeling concerns, we consider the results related to our investigations below of Vancomycin-Resistant Enterococcus (VRE) transmission among patients in hospital intensive care units. Figure 2 depicts the results obtained when comparing stochastic model simulations with corresponding deterministic ordinary differential equation formulations for the total number of beds $N = 37$ (left panel) and $N = 3700$ (right panel). The figure reveals that behaviors of the model simulations are quite different when treating one 3700 bed unit as 37 bed units in 100 hospitals if we could argue (not often plausible) that the units are similar in patient and health care worker routines. This offers rather clear warnings for the indiscriminate use of limiting deterministic ordinary differential equations in place of Markov chain models to study small population count systems.

Fundamental to the investigation of such systems is the ability to efficiently simulate the systems in the context of inverse problems, parameter estimation, sensitivity, control, etc. Computational methods abound for the corresponding deterministic limiting (as population size increases) differential equations. While a number of stochastic simulation algorithms exist, they are often difficult to use in the contexts mentioned above.

The goal of this paper is to illustrate how widely performances may vary for some stochastic algorithms when compared on two stochastic infection models and to demonstrate how one might perform computational studies to aid in selection of appropriate algorithms. Specifically, we examine three commonly used algorithms: a stochastic simulation algorithm (SSA), and explicit and implicit tau-leaping methods. One of the models used to demonstrate the efficiency of these three algorithms describes progression of a Vancomycin-resistant enterococcus (VRE) infection in a hospital unit. The other model describes the dynamics of HIV during the early stage of infection, in which the target cells are still at very high level while the infected cells are at very low level.

The outline of the remainder of this paper is as follows. In Section 2 we give short descriptions of a stochastic simulation algorithm, as well as the explicit and implicit tau-

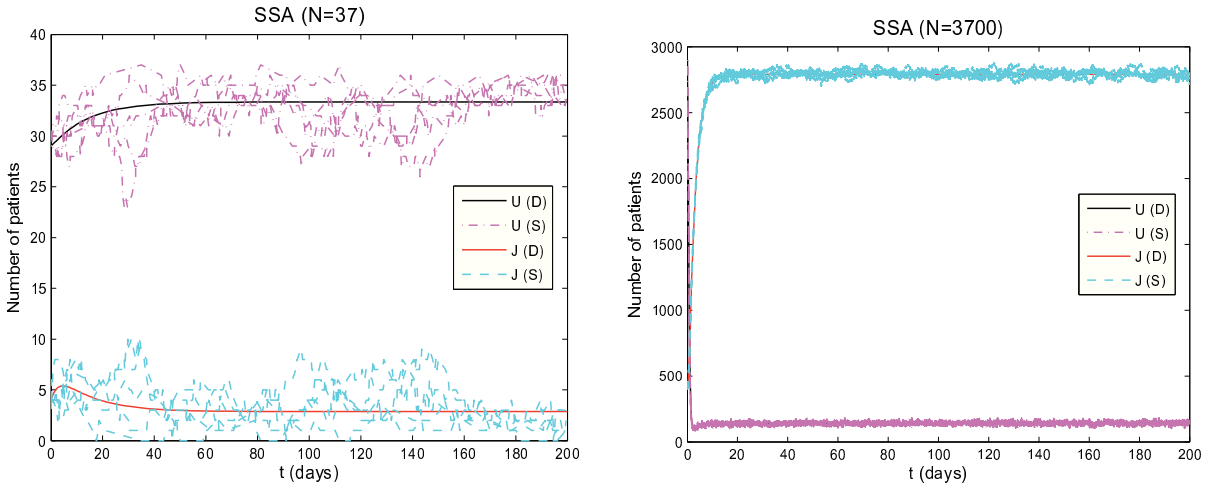

Figure 2: Results from VRE model simulations: Graphs in the left column are for uncolonized patients (U) and colonized patients in isolation (J) and N=37 and the ones in right column are for uncolonized patients (U) and colonized patients in isolation (J) and N=3700. The (D) and (S) in the legend denote the solutions obtained with the deterministic VRE model and stochastic VRE model introduced below, respectively, where the stochastic results are obtained with the SSA.

leaping algorithms. In Section 3 we apply these three stochastic algorithms to the VRE and HIV models and compare their computational efficiency. We conclude the paper in Section 4 with some summary remarks.

2 Simulation Algorithms

In this section, three computational algorithms for solving stochastic systems will be examined, the stochastic simulation algorithm, the explicit tau-leaping method and the implicit tau-leaping method. Outlines for implementing each algorithm will be given along with motivations for the algorithm and discussions about when one might want to use one algorithm over another.

Unless otherwise indicated, a capital letter is used throughout to denote a random variable, a bold capital letter is for a random vector, and their corresponding small letters are for their realizations.

2.1 Stochastic Simulation Algorithm

The stochastic simulation algorithm (SSA), also known as the *Gillespie algorithm* [8], is the standard method employed to simulate continuous time Markov Chain models. The SSA was first introduced by Gillespie in 1976 to simulate the time evolution of the stochastic formulation of chemical kinetics, a process which takes into account that molecules

come in whole numbers as well as the inherent degree of randomness in their dynamical behavior. However, in addition to simulating chemically reacting systems, the Gillespie algorithm has become the method of choice to numerically simulate stochastic models arising in a variety of other biological applications [1, 2, 12, 13, 16, 17].

Two mathematically equivalent procedures were originally proposed by Gillespie, the "Direct method" and the "First Reaction method". Both procedures are exact procedures rigorously based on the chemical master equation [8]; however, the direct method is the method typically implemented due to its efficiency. Likewise, this is the method employed in this paper. The direct method can be described for a general system by assuming $\mathbf{X} = (X_1, X_2, ..., X_n)^T$ represents the state variables of the system where $X_i(t)$ denotes the number in state X_i at time t (X_i may be the number of patients, cells, species, etc). Furthermore, it is assumed l transitions (often referred to as *reaction channels* in the biochemistry literature) are possible with associated transition rates (often referred to as *propensity functions* in the biochemistry literature) represented by λ_i, $i = 1, ..., l$. Given this terminology, the direct method for the Gillespie algorithm can be described by the following procedure:

Step 1. Initialize the state of the system $\mathbf{x}_0$;

Step 2. For the given state $\mathbf{x}$ of the system, calculate the transition rates $\lambda_i(\mathbf{x})$, $i = 1, ..., l$;

Step 3. Calculate the sum of all transition rates, $\lambda = \sum_{i=1}^{l} \lambda_i(\mathbf{x})$;

Step 4. Simulate the time, τ, until the next transition by drawing from an exponential distribution with mean $1/\lambda$;

Step 5. Simulate the transition type by drawing from the discrete distribution with probability Prob(transition $= i) = \lambda_i(\mathbf{x})/\lambda$. Generate a random number r_2 from a uniform distribution and choose the transition as follows: If $0 < r_2 < \lambda_1(\mathbf{x})/\lambda$, choose transition 1; if $\lambda_1(\mathbf{x})/\lambda < r_2 < (\lambda_1(\mathbf{x}) + \lambda_2(\mathbf{x}))/\lambda$ choose transition 2, and so on;

Step 6. Update the new time $t = t + \tau$ and the new system state;

Step 7. Iterate steps 2-6 until $t \geq t_{stop}$.

2.2 Tau-Leaping Methods

Since the SSA method keeps track of each transition, it can be impractical to implement for certain applications due to the computational time required. As a result, Gillespie proposed an approximate procedure, the tau-leaping method, which accelerates the computational time while only sustaining a small loss in accuracy [9]. Instead of taking incremental steps in time, keeping track of $\mathbf{X}(t)$ at each time step as in the SSA method, the tau-leaping method *leaps* from one subinterval to the next, approximating how many

transitions take place during a given subinterval. It is assumed that the value of the leap, τ, is small enough that there is no significant change in the value of the transition rates along the subinterval $[t, t+\tau]$. This condition is known as the *leap condition*. The tau-leaping method thus has the advantage of simulating many transitions in one *leap* while not losing significant accuracy, resulting in a speed up in computational time. In this paper, we consider two tau-leaping methods, an explicit and an implicit tau-leaping method.

2.2.1 An Explicit Tau-Leaping Method

The explicit tau-leaping method is based on an explicit formulation for the update in number of species $\mathbf{X}$ at time $t+\tau$, given $\mathbf{X}(t) = \mathbf{x}$. The basic explicit tau-leaping method approximates K_j, the number of times a transition j is expected to occur within the time interval $[t, t+\tau]$, by a Poisson random variable $P_j(\lambda_j(\mathbf{x}), \tau)$ with mean (and variance) $\lambda_j(\mathbf{x})\tau$. Once the number of transitions are estimated, the approximate number of species, known as the *tau-leaping approximation*, of $\mathbf{X}$ at time $t+\tau$ is given by the formula

$$\mathbf{X}(t+\tau) = \mathbf{x} + \sum_{j=1}^{l} P_j(\lambda_j(\mathbf{x}), \tau)\mathbf{v}_j \tag{2.1}$$

with $\mathbf{v}_j = (v_{1j}, ..., v_{nj})^T$ where v_{ij} represents the change in state variable X_i caused by transition j [6]. However, as mentioned previously, the process for selecting τ is critical in the tau-leaping method. If τ is chosen too small, tau-leaping will essentially stop, leading to the standard SSA algorithm; on the other hand, if the value of τ is too large, the leap condition may not be satisfied, possibly causing significant inaccuracies in the simulation. In this paper, we use a τ-selection procedure based on the algorithm in [6]. For alternative procedures for selecting τ, we refer the reader to references [6, 9, 10].

Let $\Delta X_i = X_i(t+\tau) - x_i$ with x_i being the ith component of $\mathbf{x}$, $i = 1, 2, \ldots n$, and ϵ be an error control parameter with $0 < \epsilon \ll 1$. In the given τ-selection procedure, τ is chosen such that

$$\Delta X_i \leq \max\left\{\frac{\epsilon}{g_i}x_i, 1\right\},\ i = 1, ..., n, \tag{2.2}$$

which evidently requires the relative change in X_i to be bounded by $\dfrac{\epsilon}{g_i}$ except that X_i will never be required to change by an amount less than 1. The value of g_i in (2.2) is chosen such that the relative changes in all the transition rates will be bounded by ϵ.

The tau-leaping method employed in this paper also includes modifications developed by Cao, et al. [5] to avoid the possibility of negative populations. When utilizing a tau-leaping method instead of the exact SSA method, as discussed previously, estimates are made about how many times a transition has occurred during the leap-interval. From the estimate of the number of transitions and how each transition effects the state variables, an estimate is obtained for the number of species in each state, X_i, at the end of the

leap-interval. In some instances, if a population or number of species is small at the beginning of the leap-interval, the estimate of the state variable after numerous transitions may result in a negative population. To avoid this situation, Cao, et al. [5, 6] introduced another control parameter, n_c, a positive integer (normally set between 2 and 20) which is used to separate transitions into two classes, critical transitions or noncritical transitions. A transition j is deemed critical if after n_c of these transitions, there is a danger in one of the state variables involved in the transition reaching zero. An estimate for the maximum number of times L_j, $j = 1, ...l$ that transition j can occur before reducing one of the state variables involved in the transition to 0 (or less) is calculated by

$$L_j = \min_{\{1 \le i \le n; \nu_{ij} < 0\}} \left[\frac{x_i}{|v_{ij}|} \right]$$

with the brackets indicating the floor function. If L_j is less than the control parameter n_c, then the reaction is deemed critical. All critical transitions are then restricted to a single transition during the leap period reducing the probability of a negative population to nearly zero. All the remaining noncritical transitions use the traditional tau-leaping method. The algorithm for the modified explicit tau-leaping method is given below.

Step 1. Given $\mathbf{X}(t) = \mathbf{x}$, identify all critical transitions by first estimating the maximum number of times, L_j, that a transition can occur before causing a negative population where

$$L_j = \min_{\{1 \le i \le n; \nu_{ij} < 0\}} \left[\frac{x_i}{|v_{ij}|} \right]$$

with the brackets indicating the floor function. A transition is considered critical if $L_j < n_c$. (In our calculations, we set n_c =10)

Let

$$J_{cr} = \{j \in \{1, ..., l\} | j \text{ is a critical transition}\}$$

and

$$J_{ncr} = \{j \in \{1, ..., l\} | j \text{ is a noncritical transition}\}.$$

Step 2. Choose a value for the error control parameter ϵ. (In our calculations, we set ϵ = 0.03). Then, compute τ_1 so each transition rate λ_j, $j = 1, ..., l$ is bounded by ϵ, according to the following definitions:

$$\tau_1 = \min_{1 \le i \le n} \left\{ \frac{\max\{\epsilon x_i / g_i, 1\}}{|\hat{\mu}_i(\mathbf{x})|}, \frac{\max\{\epsilon x_i / g_i, 1\}^2}{\hat{\sigma}_i^2(\mathbf{x})} \right\} \tag{2.3}$$

where $\hat{\mu}_i(\mathbf{x}) = \sum_{j \in J_{ncr}} v_{ij} \lambda_j(\mathbf{x})$, $\hat{\sigma}_i^2(\mathbf{x}) = \sum_{j \in J_{ncr}} v_{ij}^2 \lambda_j(\mathbf{x})$, and g_i as described in the text, $i = 1, ..., n$.

Step 3. Determine whether tau-leaping is appropriate by comparing τ_1 to $1/\lambda$. If τ_1 is less than some multiple of $1/\lambda$ (chosen to be 10 in our calculations), then abandon tau-leaping and execute a set number of single transition SSA steps (chosen to be 100 in our calculations) and return to step 2. Otherwise proceed.

Step 4. Compute the sum of all critical transition rates

$$\lambda^c = \sum_{j \in J_{cr}} \lambda_j(\mathbf{x}).$$

Generate a *second candidate* time leap, τ_2 as a sample of the exponential random variable with mean $1/\lambda^c$.

Step 5. Let $\tau = \min\{\tau_1, \tau_2\}$. Approximate the number of transitions within the time interval, K_j, as a sample of the Poisson random variable with mean $\lambda_j(\mathbf{x})\tau$ for all $j \in J_{ncr}$. For all critical transitions, define K_j as follows:

- If $\tau = \tau_1$, set $K_j = 0$ for all $j \in J_{cr}$ (no critical transitions occur).
- If $\tau = \tau_2$, let j_c be a sample of the integer random variable with point probabilities $\lambda_j(\mathbf{x})/\lambda^c$ for $j \in J_{cr}$. Set $K_{j_c} = 1$ (j_c indicates the only critical transition which occurs) and $K_j = 0$ for $j \in J_{cr}, j \neq j_c$ (only one critical transition occurs).

Step 6. If there is a negative component in $\mathbf{x} + \sum_j K_j \mathbf{v}_j$, reduce τ_1 by half and return to step 3. Otherwise leap by replacing the time, $t = t + \tau$ and then update the new system state,

$$\mathbf{x}(t + \tau) = \mathbf{x} + \sum_j K_j \mathbf{v}_j.$$

Step 7. Iterate steps 1-6 until $t \geq t_{stop}$.

2.2.2 An Implicit Tau-Leaping Method

In many applications, such as the HIV model explained in Section 3.2, problems of "stiffness" may arise. Rathinam et al. [15] explored the nature of stiffness in discrete stochastic systems and demonstrated that an implicit tau-leaping method (similar to implicit Euler methods for ordinary differential equations) is capable of taking large time steps for stiff, discrete systems, producing accurate results for such systems while significantly reducing the computational time when compared to explicit tau-leaping methods [11]. The implicit tau-leaping method replaces the explicit update formula given in equation (2.1) by an implicit tau-leaping formula given by

$$\mathbf{X}(t + \tau) = \mathbf{x} + \sum_{j=1}^{l} \left(P_j(\lambda_j(\mathbf{x}), \tau) - \lambda_j(\mathbf{x})\tau + \lambda_j(\mathbf{X}(t + \tau))\tau \right) \mathbf{v}_j.$$

Note that the above formula typically gives a non-integer vector for $\mathbf{X}(t+\tau)$. To overcome this difficulty, Rathinam et al. [15] proposed a two-stage process given by

$$\widetilde{\mathbf{X}} = \mathbf{x} + \sum_{j=1}^{l} \Big(P_j(\lambda_j(\mathbf{x}), \tau) - \lambda_j(\mathbf{x})\tau + \lambda_j(\widetilde{\mathbf{X}})\tau \Big) \mathbf{v}_j, \tag{2.4}$$

and

$$\mathbf{X}(t+\tau) = \mathbf{x} + \sum_{j=1}^{l} \Big[\!\Big[P_j(\lambda_j(\mathbf{x}), \tau) - \lambda_j(\mathbf{x})\tau + \lambda_j(\widetilde{\mathbf{X}})\tau \Big]\!\Big] \mathbf{v}_j, \tag{2.5}$$

where $[\![z]\!]$ denotes the nearest non-negative integer corresponding to a real number z.

The implicit tau-leaping method does not have a stability limitation as does the explicit tau-leaping (i.e., the relative changes in all the transition rates are bounded by ϵ) due to the implicitness of the scheme. In [7] the stepsize for the stiff system is chosen to bound the relative changes of those transition rates resulting from the non-equilibrium reactions by ϵ, thus, a larger stepsize is allowed. However, as remarked by the authors in [7], it is generally difficult to determine whether or not a reaction is in partial equilibrium, and the partial equilibrium condition is only formulated in [7] for those reversible reaction pairs for some biochemical systems. To overcome this difficulty, in this paper we use (2.3) to choose τ_1 for the implicit tau-leaping method but with a larger ϵ to allow a possible large time stepsize.

To avoid the possibility of negative populations (i.e., to ensure (2.4) has non-negative solution), the algorithm for implicit tau-leaping is implemented in the same way as that for the explicit tau-leaping method except the update for states (i.e., replace (2.1) by (2.4) and (2.5)).

3 Numerical Examples

In this section, we report on applying the SSA, explicit and implicit tau-leaping algorithms to two stochastic infection models in the literature. One is used to describe VRE infection in a hospital. The other describes the early HIV infection within a host. All three algorithms were coded in Matlab, and all the simulation results were run on a linux machine with a 2GHz Intel Xeon Processor with 8GB of RAM total.

The following notation will be used throughout the subsequent discussions: $\mathbb{Z}^n$ is the set of n-dimensional column vectors with integer components, and $\mathbf{e}_i \in \mathbb{Z}^n$ is the ith unit column vector, that is, the ith entry of $\mathbf{e}_i$ is 1 and all the other entries are zeros, $i = 1, 2, \ldots, n$.

3.1 VRE Model

We adopt the stochastic model developed in [13] for the dynamics of VRE infection in a hospital to demonstrate the computational efficiency of SSA, explicit and implicit tau-

leaping methods. In this model, the dynamics of VRE infection is modeled as a continuous time Markov chain, and a constant population is assumed so that the hospital remains full for all time; that is, the overall admission rate equals the overall discharge rate. Let N denote the total number of beds available in the hospital, and $\{\mathbf{X}^N(t), t \geq 0\}$ be a continuous time Markov chain with $\mathbf{X}^N = (X_1^N, X_2^N, X_3^N)^T$, where the meanings of the random variables X_i^N, $i = 1, 2, 3$ are given in Table 1.

Variables	Description
$X_1^N(t)$	Number of uncolonized patients at time t in a hospital with N beds
$X_2^N(t)$	Number of VRE colonized patients at time t in a hospital with N beds
$X_3^N(t)$	Number of VRE colonized patients in isolation at time t in a hospital with N beds

Table 1: State variables for the VRE model.

In any small time interval of length Δt, the process $\{\mathbf{X}^N(t), t \geq 0\}$ jumps from state $\mathbf{x}^N$ to $\mathbf{x}^N + \mathbf{v}_j$ with probability $\lambda_j(\mathbf{x}^N)\Delta t + o(\Delta t)$, that is,

$$\text{Prob}\left\{\mathbf{X}^N(t+\Delta t) = \mathbf{x}^N + \mathbf{v}_j \mid \mathbf{X}^N(t) = \mathbf{x}^N\right\} = \lambda_j(\mathbf{x}^N)\Delta t + o(\Delta t),\ j = 1, 2, \ldots, l, \quad (3.1)$$

where $\mathbf{x}^N = (x_1^N, x_2^N, x_3^N)^T \in \mathbb{Z}^3$, $\mathbf{v}_j \in \mathbb{Z}^3$, and λ_j is the transition rate for transition j given in the second column of Table 2 (see [13] for details on the rationale behind in deriving these transition rates). From this table, we see that there are five transition rates

Reactions	$\lambda_j(\mathbf{x}^N)$	$\mathbf{v}_j$
$X_1^N \to X_2^N$	$m\mu_1 x_1^N + \beta x_1^N(x_2^N + (1-\gamma)x_3^N)$	$-\mathbf{e}_1 + \mathbf{e}_2$
$X_3^N \to X_2^N$	$m\mu_2 x_3^N$	$\mathbf{e}_2 - \mathbf{e}_3$
$X_2^N \to X_1^N$	$(1-m)\mu_2 x_2^N$	$\mathbf{e}_1 - \mathbf{e}_2$
$X_3^N \to X_1^N$	$(1-m)\mu_2 x_3^N$	$\mathbf{e}_1 - \mathbf{e}_3$
$X_2^N \to X_3^N$	αx_2^N	$-\mathbf{e}_2 + \mathbf{e}_3$

Table 2: Transition rates $\lambda_j(\mathbf{x}^N)$ as well as the corresponding state changes $\mathbf{v}_j$ for the stochastic VRE model.

(i.e., $l = 5$), and the corresponding states changes $\mathbf{v}_j$ are listed in the third column of this table.

3.1.1 Numerical Results

Numerical results were obtained by applying the SSA, explicit tau-leaping and implicit tau-leaping to the stochastic VRE model (3.1) with transition rates given in Table 2. We compare the computational time of the SSA, explicit and implicit tau-leaping methods with different values of N. All the simulations were run for the time period $[0, 200]$ days with parameter values given by

$$m = 0.04, \quad \beta = 0.001, \quad \gamma = 0.58, \quad \alpha = 0.29, \quad \mu_1 = 0.16, \quad \mu_2 = 0.08,$$

and initial conditions $\mathbf{X}^N(0) = N\left(\frac{29}{37}, \frac{4}{37}, \frac{4}{37}\right)^T$. That is, all the simulations start with the same initial density $\left(\frac{29}{37}, \frac{4}{37}, \frac{4}{37}\right)^T$.

For the tau-leaping methods, the value of g_i is found to be $g_i = 4, i = 1, 2, 3$, and the value of ϵ is set as 0.3 for the implicit tau-leaping. This is chosen based on the simulation results so that the computational time is comparatively short without compromising the accuracy of the solution.

Figure 3 depicts the computational time of each algorithm for an average of five typical simulation runs with N varying from $37, 185, 370, 1850, 3700, 18500, 37000$. From this

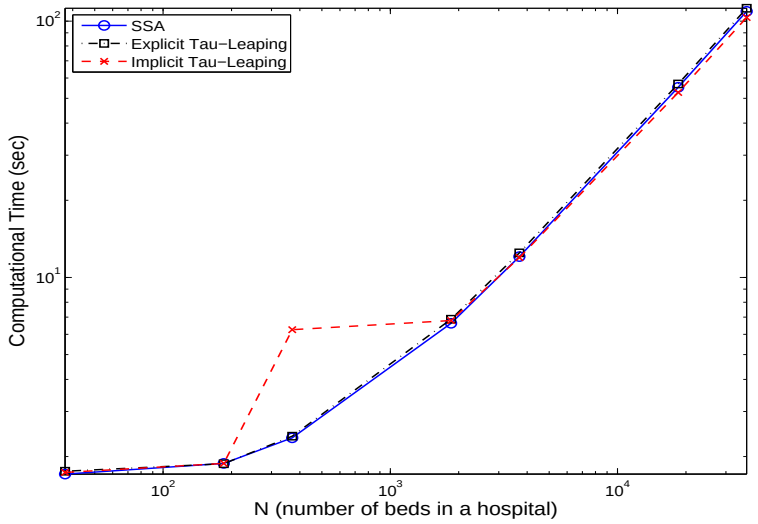

Figure 3: Comparison of computational time of different algorithm (SSA, Explicit Tau-Leaping and Implicit Tau-Leaping) for an average of five typical simulation runs.

figure, we see that the computational time for all the algorithms increases as the value N increases. This is expected for the SSA as the mean time stepsize for the SSA is the inverse of the sum of all transition rates, which increases as N increases (roughly proportional to N^2 as can be seen from the transition rates illustrated in Table 2). For the explicit tau-leaping method we found that for all the N that we tried, the value of τ_1 is often less than $10/\lambda$, which implies that the SSA is implemented for most of the time as opposed to the tau-leaping method (based on the algorithm in Section 2.2.1). This also explains why the SSA and the explicit tau-leaping perform similarly. The same thing is also observed for the implicit tau-leaping method when $N = 37$ and 185. However, when N increases to 370, we found that the implicit tau-leaping method required significantly more time, and we also found that its time stepsize in this case is still not significantly higher than those of the SSA. We note that systems of nonlinear equations need to be solved for the implicit tau-leaping method. Hence, the computational time in this case is expected to be higher than those for the other two methods (this can be observed in the figure). When

N continues to increase, we see that the computational times for the implicit tau-leaping are similar to those of the SSA and the explicit tau-leaping methods. This is because the time stepsize becomes significantly higher than the those of these two methods which compensates for the time consuming solving of systems of nonlinear equations. As we can see from (2.3) and the transition rates illustrated in Table 2, if $\epsilon x_i/g_i > 1$, then the first term inside the minimum sign of (2.3) is roughly proportional to $1/N$ while the second term is roughly proportional to 1. The simulation results show that the first term inside of the minimum sign of (2.3) is smaller than the second term when N increases to 1850 and above. Hence, the time stepsize for the implicit tau-leaping method decreases as N increases when $N \geq 1850$. This implies the computational time for the implicit tau-leaping method increases as N increases. Based on the above discussions, we see that the SSA performs similarly to the tau-leaping methods (and may be slightly better in some cases). Due to its simplicity and accuracy, the SSA is the best choice for this particular problem.

3.2 HIV Model

We adopted the stochastic model developed in [3] for the dynamics of early HIV infection within a host to demonstrate the computational efficiency of the SSA, explicit and implicit tau-leaping methods. In this model the dynamics of HIV infection are modeled as a continuous time Markov chain. Let ν denote the volume of blood (unit: μl-blood), and $\{\mathbf{X}^\nu(t), t \geq 0\}$ be a pure jump Markov process with $\mathbf{X}^\nu = (X_1^\nu, X_2^\nu, \ldots, X_7^\nu)^T$, where the meanings of random variables X_i^ν, $i = 1, 2, \ldots, 7$ are stated in Table 3. In any small

Variables	Description
$X_1^\nu(t)$	number of non-infected activated CD4+ T-cells in ν μl-blood at time t
$X_2^\nu(t)$	number of infected activated CD4+ T-cells in ν μl-blood at time t
$X_3^\nu(t)$	number of non-infected resting CD4+ T-cells in ν μl-blood at time t
$X_4^\nu(t)$	number of infected resting CD4+ T-cells in ν μl-blood at time t
$X_5^\nu(t)$	number of RNA copies of infectious free virus in ν μl-blood at time t
$X_6^\nu(t)$	number of HIV-specific effector CD8+ T-cells in ν μl-blood at time t
$X_7^\nu(t)$	number of HIV-specific memory CD8+ T-cells in ν μl-blood at time t

Table 3: State variables for the stochastic HIV model.

time interval of length Δt, the process $\{\mathbf{X}^\nu(t), t \geq 0\}$ jumps from state $\mathbf{x}^\nu$ to $\mathbf{x}^\nu + \mathbf{v}_j$ with probability $\lambda_j(\mathbf{x}^\nu)\Delta t + o(\Delta t)$. That is,

$$\text{Prob}\left\{\mathbf{X}^\nu(t+\Delta t) = \mathbf{x}^\nu + \mathbf{v}_j \mid \mathbf{X}^\nu(t) = \mathbf{x}^\nu\right\} = \lambda_j(\mathbf{x}^\nu)\Delta t + o(\Delta t), \quad j = 1, 2, \ldots, l, \qquad (3.2)$$

where $\mathbf{x}^\nu = (x_1^\nu, x_2^\nu, \ldots, x_7^\nu)^T \in \mathbb{Z}^7$, $\mathbf{v}_j \in \mathbb{Z}^7$, and λ_j is the transition rate for the jth transition given in the second column of Table 4 (the interested readers can refer to [3] for details in deriving these transition rates). From this table, we see that there are 19 transition rates (i.e., $l = 19$), and the corresponding states changes $\mathbf{v}_j$ are listed the third column of this table.

Reactions	$\lambda_j(\mathbf{x}^\nu)$	$\mathbf{v}_j$
Death of uninfected activated CD4+ T cells	$d_{T1}x_1^\nu$	$-\mathbf{e}_1$
Elimination of infected activated CD4+ T cells by HIV-specific effector CD8+ T cells	$\delta_E \frac{x_6^\nu}{\nu} x_2^\nu$	$-\mathbf{e}_2$
Death of uninfected resting CD4+ T cells	$d_{T2}x_3^\nu$	$-\mathbf{e}_3$
Death of infected resting CD4+ T cells	$d_{T2}x_4^\nu$	$-\mathbf{e}_4$
Natural clearance of virus	cx_5^ν	$-\mathbf{e}_5$
Death of HIV-specific effector CD8+ T cells	$\left(d_{E1} + d_E \frac{x_2^\nu}{x_2^\nu + k_d}\right) x_6^\nu$	$-\mathbf{e}_6$
Death of HIV-specific memory CD8+ T cells	$d_{E2}x_7^\nu$	$-\mathbf{e}_7$
Infection of activated CD4+ T cells	$\beta_{T1} \frac{x_5^\nu}{\nu} x_1^\nu$	$-\mathbf{e}_1 + \mathbf{e}_2 - \mathbf{e}_5$
Infection of resting CD4+ T cells	$\beta_{T2} \frac{x_5^\nu}{\nu} x_3^\nu$	$-\mathbf{e}_3 + \mathbf{e}_4 - \mathbf{e}_5$
Differentiation from uninfected activated CD4+ T cells to uninfected resting CD4+ T cells	$\gamma_T x_1^\nu$	$-\mathbf{e}_1 + \mathbf{e}_3$
Differentiation from infected activated CD4+ T cells to infected resting CD4+ T cells	$\gamma_T x_2^\nu$	$-\mathbf{e}_2 + \mathbf{e}_4$
Differentiation from HIV-specific effector CD8+ T cells to HIV-specific memory CD8+ T cells	$\gamma_E \frac{x_1^\nu + x_2^\nu}{x_1^\nu + x_2^\nu + k_\gamma} x_6^\nu$	$-\mathbf{e}_6 + \mathbf{e}_7$
Activation of uninfected resting CD4+ T cells	$\left(a_T \frac{x_5^\nu}{x_5^\nu + k_V} + a_A\right) x_3^\nu$	$n_T\mathbf{e_1} - \mathbf{e}_3$
Activation of infected resting CD4+ T cells	$\left(a_T \frac{x_5^\nu}{x_5^\nu + k_V} + a_A\right) x_4^\nu$	$n_T\mathbf{e_2} - \mathbf{e}_4$
Activation of HIV-specific memory CD8+ T cells	$a_E \frac{x_5^\nu}{x_5^\nu + k_V} x_7^\nu$	$n_E\mathbf{e_6} - \mathbf{e}_7$
Production of new virions simultaneous with the death of infected activated CD4+ T cells	$\delta_V x_2^\nu$	$-\mathbf{e}_2 + n_V\mathbf{e}_5$
Birth of uninfected resting CD4+ T cells	$\left(\zeta_T \frac{k_s}{x_5^\nu + k_s}\right) \nu$	$\mathbf{e}_3$
Birth of HIV-specific effector CD8+ T cells	$\nu\zeta_E + b_{E1} \frac{x_2^\nu}{x_2^\nu + k_{b1}} x_6^\nu$	$\mathbf{e}_6$
Proliferation of HIV-specific memory CD8+ T cells	$b_{E2} \frac{k_{b2}}{x_7^\nu + k_{b2}} x_7^\nu$	$\mathbf{e}_7$

Table 4: Transition rates $\lambda_j(\mathbf{x}^\nu)$ as well as the corresponding state changes $\mathbf{v}_j$ for the stochastic HIV model.

3.2.1 Numerical Results

Numerical results were obtained by applying the SSA, explicit tau-leaping and implicit tau-leaping methods to the stochastic HIV model (3.2) with transition rates given in Table 4. We will compare the computational times of the SSA, explicit and implicit tau-leaping methods with different values of ν. All the simulations were run for the time period

$[0, 100]$ days with parameter values given in Table 5 (adapted from Table 2 in [4]) and initial conditions

$$\mathbf{X}^{\nu}(0) = \nu(5, 1, 1400, 1, 10, 5, 1)^T. \tag{3.3}$$

Thus all the simulations start with the same initial concentrations $(5, 1, 1400, 1, 10, 5, 1)^T$.

Params	Value	Params	Value	Params	Value	Params	Value
d_{T1}	0.02	β_{T1}	10^{-2}	γ_T	0.005	n_T	2
a_T	0.008	k_V	0.1ν	a_A	10^{-12}	δ_V	0.7
δ_E	0.01	ζ_T	7	k_s	$10^2\nu$	d_{T2}	0.005
β_{T2}	10^{-6}	n_V	100	c	13	ζ_E	0.001
b_{E1}	0.328	k_{b1}	0.1ν	d_E	0.25	k_d	0.5ν
d_{E1}	0.1	a_E	0.1	n_E	3	γ_E	0.01
k_γ	$10\ \nu$	b_{E2}	0.001	k_{b2}	$100\ \nu$	d_{E2}	0.005

Table 5: The values of parameters used in the simulation.

For the implicit tau-leaping method, the value of ϵ is set as 0.12. This is chosen (based on multiple simulation results) so that the computational time is comparatively short without compromising the accuracy of the solution. In addition, the value of g_i in the tau-leaping methods is found to be

$$g_i = \begin{cases} 4, & i = 2, 3, 4, 5, 6, \\ 3, & i = 1, 7. \end{cases}$$

Figure 4 depicts the computational time of each algorithm for an average of five typical simulation runs with ν varying from $10, 50, 10^2, 2 \times 10^2, 5 \times 10^2, 10^3$ for the SSA and $10, 50, 10^2, 2 \times 10^2, 5 \times 10^2, 10^3, 10^4, 10^5, 10^6, 5 \times 10^6$ for the explicit and implicit tau-leaping. From this figure, we see that the computational times for the SSA increase as the values ν increase. This is again expected as the mean time stepsize for the SSA is the inverse of the sum of all transition rates, which increases as ν increases (roughly proportional to ν, as can be seen from the transition rates illustrated in Table 4). In addition, even with $\nu = 10^3$, it took the SSA more than 8000 seconds for one sample path (for this reason we did not run any simulations for the SSA with ν greater than 10^3). Hence, it is impractical to implement the SSA if one desires to run this HIV model for a typical human (generally having 5×10^6 μl-blood). This is not surprising due to the large value of uninfected resting CD4+ T cells (as can be seen from the initial condition (3.3)).

From Figure 4 we also see that the computational time for the explicit tau-leaping method increases as the value ν increases from 10 to 50, and decreases as ν increases to 100. Its computational time decreases dramatically as the value of ν increases from 100 to 10^4, and then stabilizes there for $\nu \geq 10^4$. This is related to the formula for τ_1. As we can see from (2.3) and transition rates in Table 4, if $\epsilon x_i / g_i > 1$ then the first term inside the minimum sign of (2.3) is roughly of the same order for all the values of ν while the second term is roughly proportional to the value of ν. In addition, we found from

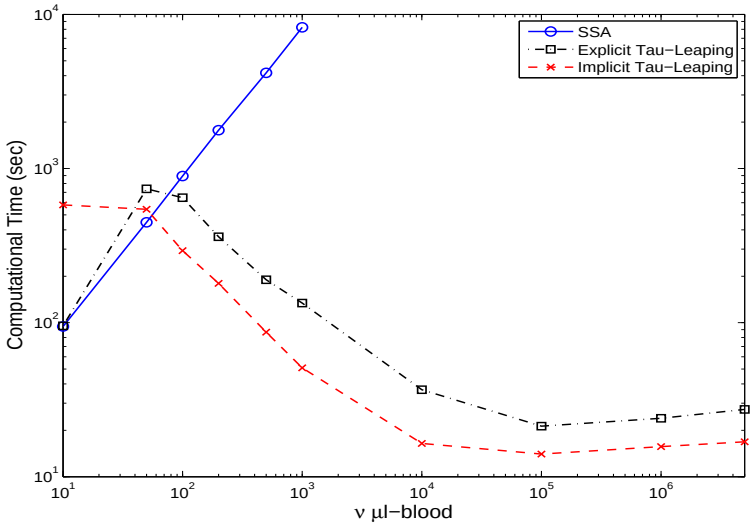

Figure 4: Comparison of computational time of different algorithm (SSA, Explicit Tau-Leaping and Implicit Tau-Leaping) for an average of five typical simulation runs.

the simulation results that the first term inside the minimum sign of (2.3) is much larger than the corresponding second term until $\nu = 10^4$, and becomes smaller than the second term when $\nu \geq 10^4$. Hence, τ_1 increases as ν increases when $\nu \leq 10^4$ and has roughly similar values for all the cases when $\nu \geq 10^4$. This agrees with the observation that the computational time decreases dramatically as the value of ν increases from 100 to 10^4, and then stabilizes for $\nu \geq 10^4$. The increase of computational time as ν increases from 10 to 50 is because τ_1 is so small that a large number of the SSA steps are implemented instead of tau-leaping.

We also observe from Figure 4 that the computational time for the implicit tau-leaping method decreases as ν increases when $\nu \leq 10^4$ and then stabilizes for $\nu \geq 10^4$; this can be attributed to the same reason as in the explicit tau-leaping method. In addition, we see that the computational times for implicit tau-leaping are significantly higher than those of the SSA and explicit tau-leaping at $\nu = 10$. This is because in this case the implicit tau-leaping is implemented for many times (solving the system of nonlinear equations in each implicit tau-leaping step is costly) and the time stepsizes are not significantly larger than those of the other two methods.

Based on the above discussions, we know that for smaller values of ν (less than 100) the SSA is the choice due to its simplicity, accuracy and efficiency. However, for larger values of ν the tau-leaping methods are definitely the choice with implicit tau-leaping performing better than explicit tau-leaping, as is expected due to the stiffness of the system (large variations in both parameter values and state variables).

4 Concluding Remarks

In this paper we summarized comparison studies on the computational efficiency of the SSA, explicit and implicit tau-leaping methods for two distinct stochastic models with different levels of complexity. Simulation results reveal that for both models the computational times of the SSA increase as the sample size (number of beds N in the VRE model and volume of blood ν in the HIV model) increases. This is because the mean time step-size for the SSA is the inverse of the sum of the transition rates, which increases as the sample size increases. In addition, the results suggest these three algorithms have comparable computational times for the VRE model because of the low number of species and small number of transitions, and thus the SSA is the best choice for this problem because of its simplicity and accuracy. However, for the HIV model both tau-leaping methods have significantly lower computational cost than the SSA except when the sample size ν is very small (e.g., less than 100 μl-blood). In addition, the implicit tau-leaping method has lower computational cost than the explicit tau-leaping method when the sample size is sufficiently large (primarily due to the stiffness of the underlying system).

Note that the stochastic HIV model in this paper describes the early infection where the number of uninfected resting T cells is very large (on order of 1000 cells per μl-blood). This explains why it required the SSA more than 8000 seconds to run one sample path with $\nu = 1000$ μl-blood. If we assume that an average person has 5×10^6 μl-blood, it is impractical for the SSA to run even one sample path at this scale. The numerical results demonstrate that the dynamics of uninfected resting CD4+ T cells could be well approximated by ordinary differential equations even with $\nu = 10$ μl-blood (see [3] for details). In addition, Table 5 reveals that there is a large variation between the values of the parameters. Thus, the HIV model in this paper is multi-scaled in both states and time. There are some hybrid simulation methods (also referred to as multi-scale approaches; interested readers can see [14] for an overview of these methods) specifically designed for multi-scale systems. The basic idea of these hybrid method is to partition the system into two subsystems, one containing fast transitions and the other containing slow transitions. Then the two subsystems are simulated iteratively by using numerical integration of ordinary differential equations (or stochastic differential equations) and stochastic algorithms (such as the SSA), respectively. Although these algorithms are very attractive, they are most challenging to implement and require major levels of user intervention. This is the primary reason we did not pursue these methods in our current efforts.

Acknowledgements

This paper results from a REU program held at North Carolina State University in the summer of 2011, with the support provided by the National Science Foundation under grants DMS-0636568, DMS-0552571, and DMS-1063010, by the National Security Agency under grant H9823-10-1-0252, and by the NC State REU in Mathematics: Modeling and Industrial Applied Mathematics. This research was also supported in part by Grant Num-

ber NIAID R01AI071915-09 from the National Institute of Allergy and Infectious Diseases.

References

[1] L.J.S. Allen, *An Introduction to Stochastic Processes with Applications to Biology*, Chapman & Hall/CRC, Boca Raton, FL, 2011.

[2] P. Bai, H.T. Banks, S. Dediu, A.Y. Govan, M. Last, A.L. Lloyd, H.K. Nguyen, M.S. Olufsen, G. Rempala and B.D. Slenning, Stochastic and deterministic models for agricultural production networks, *Mathematical Biosciences and Engineering*, 4 (2007), 373–402.

[3] H.T. Banks, S. Hu, M. Joyner, A. Broido, B. Canter, K. Gayvert and K. Link, A comparison of computional efficiencies of stochastic algorithms in terms of two infection models, CRSC-TR11-13, Center for Research in Scientific Computation, North Carolina State University, Raleigh; *Math. Biosci. Engr.*, submitted.

[4] H.T. Banks, M. Davidian, S. Hu, G. Kepler and E.S. Rosenberg, Modelling HIV immune response and validation with clinical data, *Journal of Biological Dynamics*, 2 (2008), 357–385.

[5] Y. Cao, D.T. Gillespie and L.R. Petzold, Avoiding negative populations in explicit Poisson tau-leaping, *The Journal of Chemical Physics*, 123 (2005), 054104.

[6] Y. Cao, D.T. Gillespie and L.R. Petzold, Efficient step size selection for the tau-leaping simulation method, *The Journal of Chemical Physics*, (124) 2006, 044109.

[7] Y. Cao, D.T. Gillespie and L.R. Petzold, Adaptive explicit-implicit tau-leaping method with automatic tau selection, *The Journal of Chemical Physics*, 126 (2007), 224101.

[8] D.T. Gillespie, A General Method for Numerically Simulating the stochastic time evolution of coupled chemical reactions, *The Journal of Computational Physics*, 22 (1976), 403–434.

[9] D.T. Gillespie, Approximate accelerated stochastic simulation of chemically reacting systems, *The Journal of Chemical Physics*, 115 (2001), 1716–1733.

[10] D.T. Gillespie and L.R. Petzold, Improved leap-size selection for accelerated stochastic simulaton, *The Journal of Chemical Physics*, 119 (2003), 8229–8234.

[11] D.T. Gillespie and L.R. Petzold, Stochastic simulation of chemical kinetics, *Annual Review of Physical Chemistry*, 58 (2007), 25–55.

[12] H.H. McAdams and A. Arkin, Stochastic mechanisms in gene expression, *PNAS*, 94 (1997), 814–819.

[13] A.R. Ortiz, H.T. Banks, C. Castillo-Chavez, G. Chowell and X. Wang, A deterministic methodology for estimation of parameters in dynamic Markov chain models, *Journal of Biological Systems*, 19 (2011), 71–100.

[14] J. Pahle, Biochemical simulations: stochastic, approximate stochastic and hybrid approaches, *Brief Bioinform*, 10 (2009), 53–64.

[15] M. Rathinam, L.R. Petzold, Y. Cao and D.T. Gillespie, Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method, *The Journal of Chemical Physics*, 119 (2003), 12784–12794.

[16] E. Renshaw, *Modelling Biological Populations in Space and Time*, Cambridge Univ. Press, 1991.

[17] D.J. Wilkinson, Stochastic modelling for quantitative description of heterogeneous biological systems, *Nature Reviews Genetics*, 10 (2009), 122–133.