Detailed Comparison between StochSim and SSA

Zhen Liu [*] Yang Cao[*†]

Abstract

Morton-Firth and Bray's stochastic simulator (StochSim) and Gillespie's stochastic simulation algorithm (SSA) are two important methods for stochastic modeling and simulation of biochemical systems. They have been widely applied to different biological problems. A key question is discussed here: Are these two methods equivalent? This paper compares these two methods using fundamental probability analysis. Our analysis clearly shows that, when the time step in the StochSim is chosen very small, the StochSim can be viewed as a first-order approximation to the SSA. Our analysis also explains why the SSA is usually much more efficient than the StochSim for biochemical systems. However, when multistate species present in a system, the StochSim clearly shows its advantage. We use complexity analysis to explain this advantage. The Hybrid SSA (HSSA) is proposed to combine the advantages of both the StochSim and the SSA. When the populations of the multistate species are small, the HSSA is very efficient. Numerical experiments are presented to verify the analysis.

1 Introduction

Traditionally models of biochemical systems are formulated using reaction rate equations (RREs) that are deterministic and continuous. In recent years, more and more attention has been paid to the stochastic and discrete modeling and simulation methods due to concerns over stochastic effects resulting from the small numbers of reacting molecules in microscopic systems.[1–4]

In this paper we are concerned with two fundamental stochastic simulation methods that have been successfully applied to many biological systems. One is the famous Gillespie's Stochastic Sim-

[*]Department of Computer Science, Virginia Tech, Blacksburg, VA 24061

[†]Author to whom correspondence should be addressed. Email: ycao@cs.vt.edu.

ulation Algorithm (SSA),[5,6] an essentially exact simulation method that follows the same distribution as that rules the chemical master equations[5] (CME). Progress[7–9] has been made to improve the implementation of the SSA. Approximation methods, particularly the tau-leaping methods,[10] have been proposed to improve the efficiency at a little expense of accuracy. The development of accurate and efficient approximation algorithms for SSA is still undergoing.

The other stochastic simulation method is the stochastic simulator (StochSim) developed by Morton-Firth and Bray.[11–13] The StochSim has been successfully applied in the stochastic modeling of the bacterial chemotaxis. An important feature of StochSim is its object-oriented nature. It treats all reacting molecules as individual objects with their own properties, such as conformation states, velocity, spatial information, etc. Such a special feature makes the StochSim extendable for handling special situations such as the multistate variables or spatially inhomogeneous stochastic simulation. However, the StochSim has exhibited low efficiency in systems containing large number of molecules.

An important question should be raised naturally: Are these two methods equivalent? This question was first discussed in Shimizu and Bray,[13] which showed an equivalence of the physical assumptions of the two methods. However, even though both methods are based on equivalent fundamental physics assumptions, mathematically they may still be different, particularly in their computational efficiency. Later another comparison[14] was published to show efficiency differences between them. That comparison was mostly based on a particular model. Theoretical comparison is still in need to understand their differences in accuracy and efficiency. In this paper we present a detailed comparison, based on fundamental probability analysis, for the two algorithms (with a little modification made for the StochSim). We limit our analysis under the well-stirred (spatially homogeneous) assumption. Our analysis reveals that if the time step in the StochSim is selected very small, the probability used in the StochSim is a first-order approximation to the corresponding one in the SSA. In other words, they are NOT equivalent but can be very close. According to the analysis, the SSA is more efficient in general. However, when multistate variables are involved in a system, the StochSim can be much more efficient. We explain this from complexity analysis and

numerical experiments. We believe that this detailed comparison will help to better understand the connection between the StochSim and the SSA and to develop efficient algorithms that can combine the advantages of them. As an initial attempt, we propose a hybrid strategy for cases where the populations of the multistate species are low.

The outline of this paper is as follows. In Section 2 we briefly review the SSA and the StochSim. In Section 3 we present a detailed comparison between the two algorithms. In Section 4 we discuss the special situation when multistate species are involved in the system. Numerical experiments are presented in Section 5.

2 Background

2.1 SSA

Suppose the system involves $\mathcal{N}_\mathcal{S}$ molecule species $\{S_1, \ldots, S_{\mathcal{N}_\mathcal{S}}\}$. The state vector is denoted by $X(t) = (X_1(t), \ldots, X_{\mathcal{N}_\mathcal{S}}(t))$, where $X_i(t)$ is the number of molecules of species S_i at time t. $\mathcal{M}$ reaction channels $\{R_1, \ldots, R_\mathcal{M}\}$ 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* $\nu_j = (\nu_{1j}, \ldots, \nu_{\mathcal{N}_\mathcal{S},j})$: $a_j(x)dt$ 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 molecule 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 $a_0(x) = \sum_{j=1}^{\mathcal{M}} a_j(x)$. 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. For a system with a large $\mathcal{M}$ value, we can assume that the stoichiometric matrix is sparse, which is usually true for a large chemical reacting system. Then with the best simulation strategy, the time complexity in a single step can be estimated by $O(\log(\mathcal{M}))$.[7,8,15–17]

2.2 StochSim

The StochSim is an object-oriented algorithm. In initialization, objects for all molecules in the system are created, along with a number of *pseudo-molecule objects*. Then a look-up table is constructed to describe the reaction possibilities for all reaction channels. The rows of the table list the first reactant, and the columns list the second reactant (if the second reactant is a pseudo-molecule, it represents a mono-molecule reaction). The corresponding entry in the table shows the probability for the two molecules to have a reaction. If there are multiple reaction channels between the two molecules, this entry saves the sum of the probabilities of all involved reaction channels.

The whole simulation time interval is equally divided into discrete time steps. At each time step, the StochSim proceeds with the following steps.

1. Randomly select the first reactant from all real molecule objects using uniform distribution.

2. Randomly select the second reactant from all real molecule objects, except the one just selected in step 1 * and pseudo-molecule objects using uniform distribution. With the two objects selected in steps 1 and 2, a possible reaction is determined.

3. Search in the look-up table for the possible reaction between the two selected objects. If no corresponding entry is found, StochSim concludes that no reaction occurs in this time step. Otherwise a uniform random number in $(0, 1)$ is generated and compared with the probability

*This is a modification of the original StochSim, in which the second step may select the same molecule as in the first step. Consider a dimerization reaction between two molecules of the same species. It is not possible to have a reaction with only one molecule selected twice. Thus we believe this situation should be excluded. We have also changed the probability calculation formula for the look-up table to reflect this change.

retrieved from the table. If the probability from the table is smaller, StochSim concludes that no reaction occurs in this time step. Otherwise there is a reaction between these two molecules. If there is only one possible reaction channel between these two molecules, it is simple. Otherwise, the code selects one of the reaction channels between the two molecules in a similar way as the equation (2) in the SSA.

4. Update the system accordingly and proceed the simulation to the next time step.

The numbers of real molecules and pseudo-molecules are fixed, and the timestep Δt and the entries in the table are pre-calculated before the simulation. Let N be the number of all real molecules in the system, N_0 be the number of pseudo-molecules, k_{1j} be a mono-molecule rate constant, k_{2j} be a bi-molecule rate constant, Δt be a fixed but small time step, N_A be the Avogadro constant, and V be the volume of the system. The probabilities in the look-up table are calculated with the following formula.

For a mono-molecule reaction,

$$p_{1j} = \frac{k_{1j} N (N + N_0 - 1) \Delta t}{N_0}. \tag{3}$$

For a bi-molecule reaction,

$$p_{2j} = \frac{k_{2j} N (N + N_0 - 1) \Delta t}{2 N_A V}. \tag{4}$$

In order to treat mono-molecule reactions similarly as bi-molecule reactions, pseudo-molecules are introduced with a fixed population N_0. N_0 is calculated to make the maximum possibilities of mono-molecule reactions equal to that of the bi-molecule reactions. Let k_{1max} be the maximum reaction rate of all mono-molecule reactions, k_{2max} be the maximum reaction of all bi-molecule reactions. The formula of N_0 is given by

$$N_0 = Round\left(2 N_A V \times \frac{k_{1max}}{k_{2max}}\right), \tag{5}$$

where $Round(x)$ represents the positive integer nearest to x. The time step Δt is calculated before the simulation. The criteria is that the maximum probability in the look-up table is smaller than a

probability constant $MAXP$. The choice of $MAXP$ is important but tricky. As it is not the major topic of this paper, interested readers may refer to a review article by Chatterjee and Vlachos.[17] In the real implementation of the StochSim,[18] $MAXP$ is set as 0.599, and the corresponding formula is given[18] by

$$\Delta t = \frac{MAXP}{k_{1max}N + \frac{k_{2max}N(N+N_0-1)}{2N_AV}}. \tag{6}$$

Note that the fixed N limit the application of this algorithm. For example, in a system with a binding reaction

$$A + B \leftrightarrow C, \tag{7}$$

the total number of molecules will change after the firing of this reaction. In the real implementation[18] of the StochSim, to solve this problem N is chosen larger than the actual total population and certain number of dummy species are introduced to compensate the change of total number of molecules. The reaction (7) can be read as

$$A + B \leftrightarrow C + Dummy. \tag{8}$$

For the simplification, in the following analysis we will still assume that N is the total population.

3 Comparison of SSA and StochSim

The implementation details of the SSA and the StochSim are quite different. However, it was stated in Shimizu and Bray[13] that these two methods are based on equivalent fundamental physics assumptions. Pettigrew and Resat[14] also showed that these two methods generated similar distribution plots for a particular model. In this paper we analyze the StochSim from a different angle. Instead of directly comparing it with the SSA, we first compare the SSA with a simple simulation procedure. Suppose that we equally divide the time interval into many small time slices Δt. Because the probability that the R_j reaction channel will fire in the next infinitesimal time interval $[t, t+dt)$ is given by $a_j(x)dt$, when Δt is sufficiently small the probability that one R_j reaction will occur in the time interval $[t, t+\Delta t)$ can be *approximated* by $a_j(x)\Delta t$. We can then implement the following simulation procedure.

Simulation Procedure 3.1 *Suppose at time t the system is at $X(t) = x$ and the probability that one R_j reaction will occur in the time interval $[t, t + \Delta t)$ is given by $a_j(x)\Delta t$. In each Δt time slice, a random number r is generated uniformly on interval $[0,\ 1)$ and compared with $a_0(x)\Delta t$. If $a_0(x)\Delta t > r$, one reaction will fire in this small time slice Δt and the reaction channel index j can be selected the same as (2) in the standard SSA procedure. We let the time proceed to $t + \Delta t$ and update the state variable by $X(t + \Delta t) = x + \nu_j$. Otherwise, no reaction will fire in this time interval. We just let the time proceed to $t + \Delta t$ with no state variable update.*

One can easily see that this simulation procedure is not exact. If Δt is chosen larger than $\frac{1}{a_0(x)}$, the *probability* $a_0(x)\Delta t$ will be greater than 1, and that is not allowed. Will everything be okay if $\Delta t < \frac{1}{a_0(x)}$? No. Note that the simulation procedure 3.1 implies that $1 - a_0(x)\Delta t$ is the probability that no reaction will occur in the next time interval Δt. However, one can easily derive that the probability that no reaction will fire in the next Δt for any $\Delta t > 0$ is

$$\mathrm{e}^{-a_0(x)\Delta t} = 1 - a_0(x)\Delta t + \frac{1}{2}\left(a_0(x)\Delta t\right)^2 + O((\Delta t)^3). \tag{9}$$

We can see that $1-a_0(x)\Delta t$ is the first-order approximation of (9) and the leading term for the error is given by $\frac{1}{2}\left[a_0(x)\Delta t\right]^2$. The leading term shows that if $a_0(x)\Delta t \leq 0.1$, which gives $\Delta t \leq \frac{0.1}{a_0(x)}$, the error of the first-order approximation can be estimated by 0.005. This Δt may be small enough so that the first-order approximation is acceptable and the histogram generated from the simulation procedure 3.1 will be close to that generated by the SSA. However, if Δt has to be one magnitude smaller than $\frac{1}{a_0(x)}$, the chance that no reaction will fire in the next time step $[t,\ t+\Delta t)$ is relatively high. Thus before one reaction really fires in the simulation procedure 3.1, there will be several (around 10) steps that no reaction fires at all. This situation can be illustrated in Figure 1.

The computational cost of a dynamic simulation algorithm is composed of two parts: the average computational cost for each time step and the total number of time steps in a simulation. The computational costs for the SSA and the simulation procedure 3.1 in each step are almost the same. But there are much more steps in the procedure 3.1. Thus we can conclude that the simulation procedure 3.1 is an approximation to the SSA with higher computational cost. Obviously it is not

a good strategy in stochastic simulation.

Next we compare the simulation procedure 3.1 and the StochSim. Let the fixed time steps in the StochSim and the procedure 3.1 both be Δt. We have the following theorem.

Theorem 3.1 *When Δt is sufficiently small, the procedure of the StochSim follows the same probability as the simulation procedure 3.1.*

PROOF. In the StochSim, two objects are randomly selected in the first two steps. The probability that these two could have a reaction is given by (3) or (4) if Δt is small.

Let us first look at a mono-molecule reaction R_j with a reactant S_i. In the assumption of simulation procedure 3.1, the probability that one R_j reaction will occur in the next time interval $[t, t+\Delta t)$ is given by $a_j(x)\Delta t$, where $a_j(x) = c_j x_i$. On the other hand, in the StochSim process R_j is selected when a S_i object is selected in the first step and a pseudo molecular object is selected in the second step. The probability that one S_i molecule is selected in the first step is $\frac{x_i}{N}$. The probability that one pseudo molecule is selected in the second step is $\frac{N_0}{N+N_0-1}$. Multiply them with the equation (3). The probability that an R_j reaction will fire in the next time interval $[t,\ t+\Delta t)$ in the StochSim is thus given by

$$\frac{x_i}{N} \cdot \frac{N_0}{N+N_0-1} \cdot \frac{k_{1j} N(N+N_0-1)\Delta t}{N_0} = k_{1j} x_i \Delta t. \tag{10}$$

Note that $c_j = k_{1j}$ for a mono-molecule reaction (See Ref[5]). Thus the StochSim and the procedure 3.1 follow the same probability for a mono-molecule reaction.

For a bi-molecule reaction R_j between reactants S_i and S_k, the procedure 3.1 assumes that the probability that one R_j reaction will fire in the time interval $[t, t+\Delta t)$ is given by $a_j(x)\Delta t$, where $a_j(x) = c_j x_i x_k$. In the StochSim, R_j may fire only if S_i and S_k are selected in the first two steps. The probability that one S_i molecule is selected in the first step is $\frac{x_i}{N}$, while the probability that one S_k molecule is selected in the second step is $\frac{x_k}{N+N_0-1}$. The probability that S_k is selected in the first step and S_i is selected in the second step is the same. Thus the probability that one S_i molecule and one S_k molecule are selected in the first two steps is $\frac{2x_i x_k}{N(N+N_0-1)}$. Multiply it

with the equation (4). The probability that an R_j reaction will fire in the next time interval Δt in StochSim is given by

$$\frac{2x_i x_k}{N(N+N_0-1)} \cdot \frac{k_{2j} N(N+N_0-1)\Delta t}{2N_A V} = \frac{k_{2j}}{N_A V} x_i x_k \Delta t. \tag{11}$$

Note that $c_j = \frac{k_{2j}}{N_A V}$ for a bi-molecule reaction between two different species (See Ref[5]). Again we see that the StochSim and the procedure 3.1 follow the same probability.

For a bi-molecule reaction R_j between two S_i molecules, the procedure 3.1 assumes that the probability that one R_j reaction will fire in the time interval $[t, t+\Delta t)$ is given by $a_j(x)\Delta t$, where $a_j(x) = \frac{1}{2} c_j x_i (x_i - 1)$. In the StochSim, R_j may fire only if S_i is selected in both steps. The probability that one S_i molecule is selected in the first step is $\frac{x_i}{N}$, while the probability that a different S_i molecule is selected in the second step is $\frac{x_i - 1}{N + N_0 - 1}$. Similar to the case of bi-molecule reaction between two different species, these two molecules can be selected with different order. Thus the probability that two S_i molecules are selected in the first two steps is $\frac{2x_i(x_i-1)}{N(N+N_0-1)}$. Multiply it with the equation (4). The probability that an R_j reaction will fire in the next time interval Δt in StochSim is given by

$$\frac{2x_i(x_i-1)}{N(N+N_0-1)} \cdot \frac{k_{2j} N(N+N_0-1)\Delta t}{2N_A V} = \frac{k_{2j}}{N_A V} x_i(x_i-1)\Delta t. \tag{12}$$

Note that $c_j = \frac{2k_{2j}}{N_A V}$ [†] for a bi-molecule reaction between the same species (See Ref.[5]). Again we see that the StochSim and the procedure 3.1 follow the same probability. Thus we have the theorem. □

From this theorem we can see that when Δt is sufficiently small, the StochSim can also be viewed as a first-order approximation to the SSA. However, there are some differences between the procedure 3.1 and the StochSim. Their computational costs are proportional to $\frac{1}{\Delta t}$. In order to have a high efficiency, it is preferred to have a large Δt. However, as demonstrated in the comparison between the SSA and the procedure 3.1, both the procedure 3.1 and the StochSim are only accurate when Δt is selected sufficiently small. Moreover, in the StochSim procedure all p_{1j}'s

[†]The relation between c_j and k_{2j} in this case is different from the case of bi-molecule reaction between two different species, as pointed out in Ref.[5]

in (3) and p_{2j}'s in (4) must be smaller than 1. They add extra restrictions on Δt. These extra restrictions can be very tight.

To see this point, let us consider the following example.

Example 1: Suppose there are three species S_1, S_2 and S_3 but only one reaction channel R_1 between S_1 and S_2.

$$S_1 + S_2 \xrightarrow{c} S_3 + S_2. \tag{13}$$

Let $X_1 = M_0$, $X_2 = 1$ and $X_3 = 0$, where $M_0 >> 1$. Let $c = 1$. Thus the propensity function for R_1 is $a_1(x) = x_1 x_2$. According to the SSA, the mean time for the next R_1 reaction to fire is given by

$$\tau_{SSA} = \frac{1}{a_1(x)} = \frac{1}{M_0}. \tag{14}$$

In the simulation procedure 3.1, the corresponding simulation time step Δt can be given by

$$\Delta t < \frac{0.1}{a_1(x)} = \frac{1}{10M_0}. \tag{15}$$

For the StochSim, as shown in (4)

$$p_2 = \frac{k_2 N(N + N_0)\Delta t}{2N_A V} < 1.$$

Here we already know that $c = \frac{k_2}{N_A V} = 1$. Thus the StochSim should satisfy the restriction

$$\frac{N(N + N_0)\Delta t}{2} < 1, \tag{16}$$

which gives

$$\Delta t < \frac{2}{N(N + N_0)}. \tag{17}$$

Here N is the total number of molecules. $N = M_0 + 1$. N_0 is the total number of pseudo-molecules. Since there is no mono-molecule reaction, $N_0 = 0$. Then $\Delta t < \frac{2}{(M_0+1)^2}$. When M_0 is large, this restriction on Δt is much tighter than (15). This example demonstrates a situation where the StochSim is much less efficient than the procedure 3.1 although Theorem 3.1 states that they both follow the same probability when Δt is sufficiently small.

Let $M_0 = 10^4$, equations (14) and (17) imply that for this example the number of steps the StochSim needs is about $5,000$ times as what the SSA needs. The efficiency is very low. Of course this example is an extreme case. The low efficiency of the StochSim often arises in situations where the reaction rates related to the species with large populations are relatively small, or there are reactions between species with small populations and species with large populations. In that case, from equation (6) the time step in the StochSim will be much smaller than the mean time step in the SSA.

According to the above analysis, if the computational costs in a single step are comparable for the SSA, the procedure 3.1 and the Stochsim, we can conclude that the StochSim is less or equally efficient than the procedure 3.1, which is about one magnitude slower than the SSA. For multiscale cases such as Example 1, the efficiency of the StochSim can be much lower than that of the SSA.

4 The Multi-State Situation

4.1 Multistate Species

Although for a multiscale system the efficiency of the StochSim is much lower than that of the SSA, the StochSim could have advantages for some systems if its computational cost in a single step is much lower than that of the SSA. A typical situation is when multi-state species are involved. Multi-state species often appear in biological systems where one molecule may change its characteristics depending on changes on its many binding sites, such as phosphorylation or methylation. If a molecule has 10 binding sites, depending on the states of all these binding sites, one molecule may exhibit $2^{10} = 1024$ different states. When it has 20 binding sites, the number of possible states rises to a million. If multistate species can combine in many different ways, the combinatorial complexity may lead to a very large system size.[19–21] In the structure of the traditional SSA simulation, each possible state should be assigned with an independent state variable. If each of them may react with other species in the system, $\mathcal{N}_\mathcal{S}$, the number of species, and $\mathcal{M}$, the number of reaction channels, may become dramatically large. Note that the lowest computational cost of the SSA in a single

step is $O(\log(\mathcal{M}))$.[7,8,15–17] If a multistate species is involved with many reaction channels, we have at least $\mathcal{M} = O(N_M)$, where N_M denotes the number of possible states for the multistate species. N_M can be very large, which results in a large M. The computational cost of the SSA could be very high in this case. However, since the StochSim treats molecules as individual objects, it need not introduce a large number of species and reaction channels. Its computational cost for a single step does not change with N_M. Thus when N_M is large, the computational cost in each step of the StochSim can be much smaller than that of the SSA, which may compensate the extra cost in the total number of steps as analyzed in the previous section. In that case, the StochSim shows advantages over the SSA.

Example 2: Consider a system with three types of species X, Y and E, where E is an enzyme with $1,000$ different states (10 binding sites). Each state has different characteristics for its enzyme ability. And the population of E is small. Three types of reactions are considered.

$$X \xrightarrow{E_n} Y, \tag{18}$$

$$E_n \longrightarrow E_{n+1}, \tag{19}$$

$$E_{n+1} \longrightarrow E_n, \tag{20}$$

where E_n represents E in state n. For the SSA to simulate this system, because each state of E is formulated as an individual species and each of the three reaction types (18-20) will be correspondingly extended to around 1,000 reaction channels, we have $\mathcal{N}_\mathcal{S} = 1,002$ and $\mathcal{M} \approx 3,000$. However, the StochSim can group all these reaction channels into just three channels (18-20) by treating the 1,000 reaction channels as one. When an E molecule is picked in the first two steps of the StochSim, it must have been in a particular state. When one reaction fires and this molecule changes its state, the StochSim procedure only needs to change the corresponding state for this molecule.

4.2 The Hybrid SSA

For many biological systems, the populations of species are of multiscale. Most species are present with large or moderate populations and do not have the multistate problem as discussed in Section 4.1. There are a few species with multistate characteristics. If the multistate species present with small populations, it is more efficient to treat each multi-state molecule as an independent object. We call this strategy the hybrid SSA (HSSA). The simulation procedure of the HSSA is very similar to that of the standard SSA. The difference lies in the classification of species and reaction channels. A system contains two types of species: normal species and multistate species. In the HSSA, normal species remain the same as in the standard SSA while each molecule of the multistate species is stored as an indexed object. Then this system contains three types of indexed reactions.

1. Reactions among normal species. This type remains the same as in the standard SSA.

2. Reactions involved with only one multistate object. They include mono-molecule reactions of a multistate object and bi-molecule reactions between a multistate object and a non-multistate species.

3. Reactions between two multistate objects.

In the simulation, the total propensity function $a_0(x)$ is the sum of all reactions. The time step τ and the reaction index j are calculated using (1) and (2). If the index j points to the first type, the system is updated as in the standard SSA. Otherwise, the firing of a reaction will cause a state change of one or two multistate objects. Each involved object changes its corresponding state and updates its rate constants. When a new multistate molecule is produced, a new object is created into the system. When an existing multistate molecule degrades, the corresponding object is eliminated from the system. Thus the numbers of species and reaction channels vary. For certain systems, the dynamical changes of the numbers of species and reaction channels may be very large. Such examples can be found in Lok and Brent[22] where the chemical network is updated dynamically and the standard SSA is applied to the dynamically varying chemical network. If the multistate

species are generated and eliminated with small populations, the HSSA strategy will show good efficiency.

Let us consider Example 2 again. Assume that the total population of E is E_T. As discussed before, the standard SSA will have 1002 species and about 3000 reaction channels. However, if we apply the HSSA method, each E molecule is treated as a multistate object. For each E object the three reaction channels (18-20) have local copies. Thus $\mathcal{N}_{\mathcal{S}} = 2 + E_T$ and $\mathcal{M} = 3E_T$. When E_T is small, the efficiency gain is great. When E_T increases, the efficiency gain decreases. When E_T reaches around $1,000$, it will be less efficient than the standard SSA.

5 Numerical Experiments

In this section we present numerical experiments for the comparison of the SSA and the StochSim.

5.1 Bacteria Chemotaxis Model

The first example we use is the bacteria chemotaxis model on which the StochSim has been applied successfully. This model contains 10 reaction channels (see the supplementary material) with the assumption that some fast reaction channels always remain in equilibrium. The enzyme ($TTWWAA$) plays an important role in this model. It has multiple states and can participate in various types of reactions such as phosphorylation, methylation, and binding with other chemicals. The StochSim models each molecule of the enzyme $TTWWAA$ as an object with many different states. To implement the SSA, we have to transform different states of the enzyme TTWWAA into different species. Each species corresponds to one state of the enzyme. By doing so, we increase the number of reactants and reactions in the system. The resulted model (see the supplement material) contains 28 reaction channels.

We listed the means and variances of the total population of the active $TTWWAA$ in Table 1. The corresponding histograms are shown in Figure 2. We can see that the results from the StochSim and the SSA are very close. But the average tau value of SSA is 217 times larger than

the Δt for StochSim, while the simulation time of SSA is 31 times faster than that of StochSim. Note that here for the fair comparison, the StochSim has been rewritten in C language to improve the efficiency. For the original StochSim package, $10,000$ simulations took $219,280$ seconds CPU time, which is 5.6 times slower than our simple implementation in C language.

	Mean	*Variance*	*Simulation time*	*Average Stepsize*
StochSim	210.82	202.83	*39,202s*	3.6×10^{-5}
SSA	210.12	207.82	*1,268s*	7.9×10^{-3}

Table 1: Comparison of the means, variances, simulation times and average stepsizes by SSA and StochSim on Chemotaxis for 10,000 runs.

5.2 Efficiency Comparison on a Simple Example

We implemented the StochSim, the SSA and the HSSA methods on a modified version of Example 2. The reactions are listed below

$$\begin{aligned} E_n + X &\xrightarrow{k_1} E_n + Y \\ Y &\xrightarrow{k_2} X \\ E_n &\xrightarrow{k_3} E_{n+1} \\ E_n &\xrightarrow{k_4} E_{n-1} \end{aligned} \tag{21}$$

where E_n represents the state n in species E and $1 \leq n \leq 1,000$. The reaction rate $k_1 = 5 \times 10^3 n$ [‡], where n is the index of the corresponding state of E. $k_2 = 1$ and $k_3 = k_4 = 0.1$. First we fix the initial population of $X = 100$ and $Y = 0$ and increase E_T, the population of E, from 1 to $10,000$.

The CPU time for different methods are listed in Table 2. For this simple example, when the population of E is small, the HSSA is the most efficient, and the StochSim is a little more efficient than the SSA. As E_T increases by a factor of 10, the CPU time for the SSA increases with a factor between 6 and 10. For the StochSim, when $E_T < 100$, the CPU time increased little. After $E_T > 100$, the CPU time shows superlinear increase. The CPU time of the HSSA shows even more

[‡] k_1 is much larger than other reaction rates because it is a bi-molecule reaction rate.

obvious superlinear increase trend. As E_T becomes large, the HSSA becomes the slowest method.

Population of E	*1*	*10*	*100*	*1,000*	*10,000*
SSA	*0.81*	*6.93*	*58.29*	*324*	*2,013*
StochSim	*0.31*	*0.34*	*0.90*	*11.42*	*666*
HSSA	*0.013*	*0.11*	*6.64*	*425*	*22,122*

Table 2: The CPU time comparison among the SSA, the StochSim, and the HSSA for 1,000 runs of the model (21)

The simulation results can be explained with complexity analysis. The CPU time is decided by the computational cost in a single step multiplied by the total number of steps. For SSA, as E_T increases, the propensity values $a_j(x) = O(E_T)$ except the second reaction channel in (21). The computational cost in a single step of the SSA does not change with E_T. The number of steps is of order $O(\frac{1}{\tau})$. Because the τ value is of order $O(\frac{1}{a_0})$, the number of total simulation steps is then of order $O(a_0)$. But for this example, $a_0(x) = \sum_j a_j(x)$ is of order $O(E_T)$. Thus the CPU time is of order $O(E_T)$. For the StochSim, when E_T increases, the computational cost in a single step does not change much. The CPU time is proportional to $\frac{1}{\Delta t}$. From (17) we know that the CPU time is of order $O(N^2)$, where N is the total population in the system. In this example, $N = 100 + E_T$. When $E_T < 100$, N does not change much as E_T increases. Thus the CPU time does not increase much. But when $E_T > 100$, N increases linearly with E_T. Thus the CPU time increases quadratically with E_T. For the HSSA, since it is still the SSA method, the number of steps should be the same as the situation in the standard SSA. Thus the number of the steps for the HSSA is of order $O(E_T)$. However, its computational cost in a single step is $O(\mathcal{M})$. When E_T is small, $\mathcal{M}$ is small. This cost is small. When E_T increases, each new object will have its own copy of reaction channels. We have $\mathcal{M} = O(E_T)$. Thus the computational cost in a single step is $O(E_T)$. Multiplying the computational cost in a single step and the total number of steps, we know that the CPU time for the HSSA is of order $O(E_T^2)$.

From this example we can see that the HSSA works well when E_T is small. When the total

population increases, the efficiency of the StochSim drops quadratically. But with a large E_T, the StochSim still shows advantages for systems involved with multistate species. Further improvement of HSSA is in need to combine the advantages of the StochSim and the SSA.

6 Conclusion

In this paper we analyze the theoretical foundation of the StochSim and compare the SSA and the StochSim. We have demonstrated that when Δt is very small, the StochSim can be viewed as a first-order approximation of the SSA. In general the SSA is more efficient than StochSim, especially when dealing with systems with a large N (total population). The StochSim has advantages when multistate species are involved in the system. We have proposed the HSSA method to improve the efficiency of the SSA in the multistate case. When the number of multistate molecules is small, the HSSA shows a high efficiency. When the number of multistate molecules increases, the large numbers of species and reaction channels still present a challenge for SSA type of methods. A good solution in this situation is still under research. The recent progress[20,21] on the simulation methods for the rule-based modeling has pointed to a good direction. We believe that the analysis presented here will help to derive an improved algorithm that can combine the advantages of both the StochSim and the SSA.

Acknowledgement: This work was supported by the National Science Foundation under award CCF-0726763.

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

[4] Elowitz MB, Levine AJ, Siggia ED and Swain PS. Stochastic gene expression in a single cell. *Science*, 297:1183–1186, 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. *To appear, J. Chem. Phys.*, 2004.

[9] McCollum JM, Peterson GD, Cox CD, Simpson ML and Samatova NF. The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Computational Biology and Chemistry*, 30:39–49, 2006.

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

[11] C. Morton-Firth, D. Bray. Predicting temporal fluctuations in an intracellular signalling pathway. *J. theor. Biol.*, 192:117–128, 1998.

[12] C. Morton-Firth, T. Shimizu , D. Bray. A free-energy-based stochastic simulation of the tar receptor complex. *J. Mole. Bio.*, 286:1059–1074, 1999.

[13] T. Shimizu and D. Bray. *Computational cell biology - the stochastic approach in Foundations of Systems Biology (Kitano, H. ed.)*. MIT Press, Cambridge, MA, 2001.

[14] Pettigrew MF, and H Resat. Modeling Signal Transduction Networks: A comparison of two Stochastic Kinetic Simulation Algorithms. *J. Chem. Phys.*, 123:114707, 2005.

[15] T Schulze. Kinetic monte carlo simulations with minimal searching. *Phys. Rev. E*, 65(3):036704, Feb 2002.

[16] H. Li and L. Petzold. Logarithmic Direct Method for DiscreteStochastic Simulation of Chemically Reacting Systems. technical report, 2006.

[17] A Chatterjee and D Vlachos. An overview of spatial microscopic and accelerated kinetic Monte Carlo methods. *J. Computer-Aided Materials Design*, 14:253–308, 2007.

[18] D Bray's group. http://www.pdn.cam.ac.uk/groups/comp-cell/StochSim.html.

[19] W. Hlavacek, J. Faeder, M. Blinov, R. Posner, M. Hucka and W. Fontana. Rules for Modeling Signal-Transduction Systems. *Sci. STKE*, 2006:re6, 2006.

[20] J Yang, M Monine, J Faeder and W Hlavacek. Kinetic monte carlo method for rule-based modeling of biochemical networks, 2007. arXiv.org:0712.3773.

[21] V Danos, J Feret, W Fontana1, J Krivine. Scalable simulation of cellular signaling networks. *Lect. Notes Comput. Sci*, 4807:139–157, 2007.

[22] Lok L and Brent R. Automatic generation of cellular reaction networks with Moleculizer 1.0. *Nat Biotechnol*, 23:131–136, 2005.

Supplement Material

Chemotaxis Model

The bacteria chemotaxis model we used in this paper is based on the activity of $TTWWAA$, a complex protein composed of two molecules of each of *Tar*, *CheW* and *CheA*. The complex has active or inactive states and its activity is determined by its methyl and aspartate binding sites. In general, the more methyl groups are bound to it, the more active the complex is. On the other hand, when aspartate is bound to the receptor Tar, the activation level of the complex is repressed. Activated $TTWWAAs$ can be autophosphorylated and the phosphate complexes can react with proteins *CheY* and *CheB*, producing *CheYp* (phosphate *CheY*) and *CheBp* (phosphate *CheB*). *CheYp* can bind to the motor complex of the cell flagella, increase the probability that the motor switches from counter clock-wise (CCW) rotation to clock-wise (CW) rotation, and finally decide the swimming pattern of the bacteria. *CheBp* and *CheR* act as regulators on the methylation level of the complex. Both of them can bind to TTWWAA and assist removing from or adding methyl groups onto the complex in relatively slow rates. The change of methylation level helps to maintain the activity level of the complex.

The chemotaxis model we consider here is a simplified version.[12] The StochSim treats $TTWWAA$ as a multistate species with five properties listed below:

1. Whether or not it is bound with *CheR* (R or N_R)
2. Whether or not it is bound with *CheBp* (B or N_B)
3. Level of methylation (from 0 to 4)
4. Active or inactive (A or I)
5. Whether or not it is bound to *aspartate* (asp or N_{asp})

The above five properties determine the state of a $TTWWAA$ molecule. For example, property $N_RB2I?$ means that the complex is not bound to *CheR* but bound to *CheBp*, with a methylation level 2, in an inactive state, and may or my not be bound with aspartate.

Because the activation of $TTWWAA$ and the binding of aspartate are much faster than other reactions, these two processes are treated as fast reactions and assumed to reach partial equilibrium all the time. For all the other reactions, Table 3 provides a detailed list.

Description	Reaction	Multistate Reactant Properties	Effects
CheBp Binding	$Bp + TTWWAA \leftrightarrow TTWWAA$	$N_R N_B ? A ?$	$N_R B ? A ?$
Demethylation	$TTWWAA \rightarrow TTWWAA + Bp$	$N_R B 1 ? ?$	$N_R N_B 0 ? ?$
	$TTWWAA \rightarrow TTWWAA + Bp$	$N_R B 2 ? ?$	$N_R N_B 1 ? ?$
	$TTWWAA \rightarrow TTWWAA + Bp$	$N_R B 3 ? ?$	$N_R N_B 2 ? ?$
	$TTWWAA \rightarrow TTWWAA + Bp$	$N_R B 4 ? ?$	$N_R N_B 3 ? ?$
CheR Binding	$R + TTWWAA \leftrightarrow TTWWAA$	$N_R N_B ? A ?$	Replace $RN_B ? A ?$
Methylation	$TTWWAA \rightarrow TTWWAA + R$	$RN_B 0 ? ?$	$N_R N_B 1 ? ?$
	$TTWWAA \rightarrow TTWWAA + R$	$RN_B 1 ? ?$	$N_R N_B 2 ? ?$
	$TTWWAA \rightarrow TTWWAA + R$	$RN_B 2 ? ?$	$N_R N_B 3 ? ?$
	$TTWWAA \rightarrow TTWWAA + R$	$RN_B 3 ? ?$	$N_R N_B 4 ? ?$

Table 3:

The above is the model implemented in the StochSim. We translated it into the corresponding representation for SSA, with the same partial equilibrium for the fast reactions. We treat every state of TTWWAA as one species. Table 4 provides a list of all the other converted reactions. In the list, the subscripts of a TTWWAA complex refers to its methylation level. Note that complexes with different subscripts belong to different species.

Description	Reaction
CheBp Binding	$Bp + TTWWAA_0 \leftrightarrow TTWWAA_0B$
	$Bp + TTWWAA_1 \leftrightarrow TTWWAA_1B$
	$Bp + TTWWAA_2 \leftrightarrow TTWWAA_2B$
	$Bp + TTWWAA_3 \leftrightarrow TTWWAA_3B$
	$Bp + TTWWAA_4 \leftrightarrow TTWWAA_4B$
Demethylation	$TTWWAA_1B \rightarrow TTWWAA_0 + Bp$
	$TTWWAA_2B \rightarrow TTWWAA_1 + Bp$
	$TTWWAA_3B \rightarrow TTWWAA_2 + Bp$
	$TTWWAA_4B \rightarrow TTWWAA_3 + Bp$
CheR Binding	$R + TTWWAA_0 \leftrightarrow TTWWAA_0R$
	$R + TTWWAA_1 \leftrightarrow TTWWAA_1R$
	$R + TTWWAA_2 \leftrightarrow TTWWAA_2R$
	$R + TTWWAA_3 \leftrightarrow TTWWAA_3R$
	$R + TTWWAA_4 \leftrightarrow TTWWAA_4R$
Methylation	$TTWWAA_0R \rightarrow TTWWAA_1 + R$
	$TTWWAA_1R \rightarrow TTWWAA_2 + R$
	$TTWWAA_2R \rightarrow TTWWAA_3 + R$
	$TTWWAA_3R \rightarrow TTWWAA_4 + R$

Table 4:

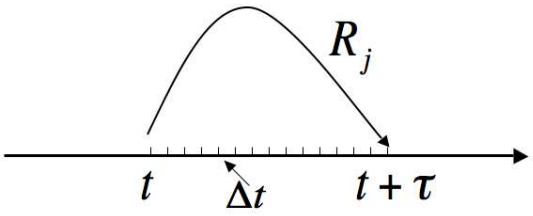

Figure 1: The time step comparison between the SSA and the simulation procedure 3.1. To ensure the accuracy, for each reaction corresponding to one step τ in the SSA, there must be several (around 10) steps that in the time interval Δt there is no reaction firing in the simulation procedure 3.1.

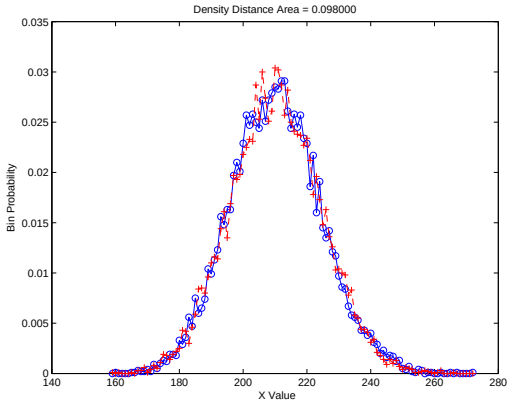

Figure 2: The histograms of the active TTWWAA simulated by the SSA(blue) and the StochSim(red).