accuracy of the algorithm in comparison to the widely used SSA (Stochastic Simulation Algorithm).

1 Introduction

Kinetic Monte Carlo (KMC) simulations have been applied extensively to model the dynamic evolution of what may be broadly classified as reaction-diffusion systems.[1–15] Unlike continuum models, KMC simulations have the ability to accurately incorporate the stochastic nature of these processes. The history of so-called “rejection-free" Kinetic Monte Carlo methods in which all moves are accepted with a certain transition probability began in the mid-70s with two seminal papers published within a year of each other. The first, by Bortz, Kalos and Lebowitz (BKL),[16] became especially popular in the materials and physics communities and has been used extensively to model island growth, thin film growth, self-assembly, physical and chemical vapor deposition processes in the intervening decades. In such algorithms, the “events" considered in the system invariably tend to involve “jumps" of an entity (*e.g.*, an atom) from one lattice site to another lattice site nearby. The second paper, by Gillespie,[17] centered around the concept of a “well mixed" system, reminiscent of chemical systems, and introduced to the community as the, now extensively used, Stochastic Simulation Algorithm (SSA). In the intervening decades, SSA, and the scores of SSA-derivative algorithms that it inspired, have been extensively applied to the simulation of chemical systems and, especially, computational biology.[1–8] Both communities refer to these methods as Kinetic Monte Carlo but, as reported in a recent review of KMC techniques by Chatterjee and Vlachos,[18] they “rarely reference each other." This dual adoption of the same term by different communities to mean distinctly different implementations of the master equation is confusing to many newcomers to the field. In KMC, whether BKL- or Gillespie-based, a lattice-based approach is not necessary, but it makes the computation simpler.

Gillespie’s SSA[17,19] is an accurate method, but one that rapidly becomes computationally infeasible, even for modestly large systems, because of its inherent nature of executing one reaction in a time interval. This limitation has led to a deep literature of publications[20–43] focused on developing an accelerated Gillespie KMC scheme. One of the more popular schemes is the so-called "τ-leaping" algorithm originally devised by Gillespie.[21] τ-leaping uses the approach of "firing" (enacting) multiple reactions within a leap interval, τ. The choice of τ is selected in such a way that the rates of all reactions will not change by a significant amount within that interval, otherwise the algorithm will be inaccurate. This assumption/constraint causes the derivation of τ to be the most critical step in the algorithm.

The algorithms mentioned above apply to spatially homogeneous systems, which assume that particle diffusion is sufficiently fast that each particle has an equal probability of reacting at any given point of time. To tackle the systems in which diffusion plays an important part, spatial compartment-based KMC algorithms is employed by numerous investigators.[31,44–48] Such algorithms typically take care of the spatial aspect by dividing the domain into a set of well-mixed sub-volumes and hence model diffusion by moving particles from one sub-volume to its neighboring sub-volumes. However, such algorithms are unable to tackle the fate of individual particles *because of the well-mixed sub-volume assumption*. In other words, this is not an "explicit-atom" KMC approach. Unfortunately, there are some complex reaction-diffusion networks where interactions and movements at the atomic level are important in order to accurately depict their stochastic behavior. One important example of this can be found in semiconductor processing where modeling of dopant and damage profiles resulting from ion implantation[11–14,49–60] involves the representation of a variety of intrinsically off-lattice and atomic cluster-level events, such as the self-interstitials, dopants at interstitial locations, and more complex clusters composed of dopants, self-interstitials and vacancies arranged in a complex on- and off-lattice manner.

In this paper, we develop a new accelerated τ-leaping KMC algorithm to represent *off-lattice* reaction-diffusion networks, named the Cornell Off-Lattice Particle-based Simulator (COPS). The representation of reactions takes place at an atomic level; accordingly, both unimolecular and bimolecular reactions (two-body collisions) are modeled in the algorithm. The unique approach taken here is to derive an expression for the variable time step, τ, and execute multiple events within that time step. As already mentioned, the selection of the time step is the most critical step. Hence, a detailed explanation is provided for our derivation of time step expressions. The performance and accuracy of the algorithm are tested through two numerical examples.

There are very few particle-based stochastic simulators that work at a similar level to the new COPS algorithm. Of these, ChemCell,[61] MCell,[62] Smoldyn[63,64] and DADOS[11–14,57,60] are the best known off-lattice, stochastic, particle-based reaction-diffusion simulators.[65] To differentiate COPS from these approaches, we note that ChemCell and Smoldyn use a fixed (not variable) time step; MCell uses an adaptive time step. But only Smoldyn can handle reaction and diffusion in 1D, 2D and 3D. As we shall show in this paper, COPS uses the concept of a variable unbiased time step, one that responds to the local environment of the reaction-diffusion process. DADOS uses the SSA concept of firing one event at a time which severely limits the computational speed; in contrast, COPS has the ability to fire multiple events at a time, resulting in accelerated computational speed. In terms of applications to date, DADOS has been exclusively used in the field of semiconductor modeling and design, and the other three methods have been used almost exclusively in computational biology applications. COPS can be applied to any type of reaction-diffusion system. Finally, COPS has the advantage over MCell and ChemCell that it can handle 1D, 2D and 3D systems.

2 Previous Work

As mentioned above, stochastic algorithms have been developed to handle reaction-diffusion systems at either mesoscopic or microscopic levels. Algorithms developed at a microscopic level focus on the behavior of individual entities. These algorithms track the position and state of each individual particle; they provide an exact representation of the underlying process. Stochastic algorithms at the mesoscopic level essentially sacrifice accuracy for computational speed by discretizing the computational domain into subvolumes. Instead of treating particles individually, these algorithms keep track of the total number of particles of each species per subvolume. At either the microscopic or mesoscopic level, the reaction-diffusion system can be assumed to be either well-mixed (homogeneous) or not (heterogeneous).

In this section, some of the major previous work is explained by introducing some exact stochastic KMC algorithms at both microscopic and mesoscopic levels and for well-mixed and heterogeneous systems. Accelerated τ-leaping algorithms developed at each level are mentioned to lay the foundation for the development of the COPS algorithm for off-lattice particle-based systems.

2.1 Exact Stochastic Methods

2.1.1 Mesoscopic-level Methods

SSA was developed by Gillespie[17] in 1976 to simulate the temporal evolution of N molecular species $\{S_1, S_2, ...S_N\}$ in well-mixed systems, described at a mesoscopic level. The number of molecules of each species is given by state vector $X(t) = [X_1(t), ..., X_N(t)] = \mathbf{x}$. These species of molecules interact with each other through M reaction events $\{R_1, R_2, ...R_M\}$ in a fixed volume Ω. For each reaction event, i, the propensity function is defined as the probability that one event of type i will occur inside the volume Ω in an infinitesimal interval $[t, t+dt]$. The propensity of a reaction event is denoted by $a_i(t)$ and is given by $c_i h_i(\mathbf{x})$. Here, c_i is the specific probability rate constant for channel (reaction) R_i and $h_i(\mathbf{x})$ is the number of distinct combinations of R_i reactant molecules available in state $\mathbf{x}$. Step-by-step details of the SSA procedure are described in the Supplemental Information.

Bernstein[44] has developed a spatial extension of SSA to handle the heterogeneity of the system at a mesoscopic level. The algorithm discretizes space into cubic sub-volumes and each sub-volume is assumed to be well-mixed. The diffusion of the species is modeled in terms of species moving from one sub-volume to its neighboring sub-volumes and is modeled as a unimolecular reaction. The diffusive unimolecular reaction is denoted as $S_i^a \xrightarrow{d_i} S_i^b$ which represents the species S_i moving from sub-volume a to its neighboring sub-volume b with microscopic diffusive rate constant, d_i. The propensity of this reaction is defined as $d_i h_i^a$, where h_i^a is given by the number of species S_i in sub-volume a, and d_i is given by D_i/h^2. D_i is the diffusivity value of species S_i and

h is the length of a cubic sub-volume. The SSA procedure defined above is then implemented in the entire system, containing diffusive unimolecular reactions for each species in each sub-volume and M reaction events.

2.1.2 Microscopic-level Methods

In contrast to SSA's mesoscopic level description for well-mixed systems, there have been a couple of KMC algorithms relevant to the microscopic level. A rule-based modeling of biochemical networks was developed by Yang *et al.*[66] for well-mixed systems. In this approach, species are represented as structured objects, and interactions between species are represented as rules that operate on these objects to modify their properties. The explicit nature of their method means that they cannot simulate systems exhibiting complex interactions. To overcome this limitation, a "Network-Free" stochastic simulator, (NFsim), was introduced by Sneddon *et al.*[67] NFsim is the platform that efficiently simulates rule-based models and permits flexible coarse-graining of reaction mechanisms for well-mixed systems using arbitrary mathematical or conditional functions.

At a microscopic level for heterogeneous systems, DADOS[11–14,57,60] is the most popular off-lattice exact stochastic simulator in the field of semiconductor modeling. The algorithm layout is very similar to the SSA for well-mixed systems except the particles have their own spatial coordinates and they can react and move in an off-lattice fashion. In a DADOS simulator, reactions such as $S_1 + S_2 \leftrightarrow S_3$ are modeled by performing interactions (forward reaction) and events (reverse reaction). When particles S_1 and S_2 are within a so-called "capture volume"[11,12,60] of each other they interact, leading to the formation of S_3. On the other hand, the probability of species S_3 breaking-up is controlled by its event rate. Other type of events might include, for example, diffusional hops of mobile species, or emission of point defects from clusters.

2.2 τ-Leaping Methods

2.2.1 Mesoscopic-level Methods

As mentioned above, SSA's primary shortcoming is that every event firing is simulated explicitly. This imposes a considerable computational burden on the algorithm, particularly if one or more species have large populations. To accelerate the SSA, the concept of τ-leaping was introduced by Gillespie[21] in 2001. The approach begins by defining a quantity, $K_i(\tau)$, as the number of times reaction i fires during a time interval $[t, t+\tau]$. The key idea is to calculate a time step, τ, during which all the propensities remain essentially constant. In such a case, the reaction dynamics can be assumed to obey Poisson statistics and the number of firings is given by:

$$K_i(\tau) \approx \mathscr{P}(a_i(t)\tau). \tag{1}$$

where $P(a_i(t)\tau)$ is a Poisson random variable with mean and variance $a_i(t)\tau$. At each iteration, the value of the time step is selected and the system is updated by firing events multiple times given by equation (Eq. (1)). A proviso is added that, if the value of the time step is too small, some variant of SSA[21] is used instead to avoid operational difficulties.

The τ-leaping approach has been modified by various investigators[21–29,32] focusing on resolving the inherent issues in the algorithm. One of these issues is that the use of Poisson random variables can lead to negative populations because of the unconstrained nature of the variables. To finesse this, Tian and Burrage[25] originated a method using binomial random numbers since they are inherently constrained.

For heterogenous systems, spatial leaping approaches[45–47] involve grouping events (reaction and diffusion) by sub-volume, calculating a characteristic time interval τ leap for each sub-volume and then choosing the global time step. Every reaction and diffusion event can then fire multiple times within τ.

2.2.2 Microscopic-level Methods

Vlachos[68] and co-workers have implemented a τ-leaping approach to handle lattice-based systems described at a microscopic level and capable of handling heterogenous system. They used the BKL method[16] to calculate the transition probability of processes occuring in the system. The BKL method groups processes into classes of identical transition probability per unit time. These groups of processes are then termed as groups of reactions on which the τ-leap technique is implemented. Vlachos's approach used a temporal coarse-graining by using an acceleration factor in the expression for the time step τ. The value of the time step, τ, is thus chosen as that of the microscopic KMC method multiplied by an acceleration factor, f (where f $>$ 1). This approach thus has many of the features that we are looking for: It is an accelerated, microscopic-level approach capable of handling heterogenous systems. Of course, the key distinction between Vlachos's[68] τ-leap technique and the COPS algorithm developed here is that Vlachos's method[68] was intended for lattice-based systems, whereas COPS is specifically tailored to off-lattice systems.

3 COPS - A leaping approach for off-lattice particle-based reaction-diffusion systems

As mentioned in the previous work, DADOS is an exact stochastic simulator which was developed to handle off-lattice particle based systems. But it is inherently slow due to its use of an SSA-like algorithm. None of the leaping approaches mentioned in the previous work can be applied to tackle off-lattice particle-based systems. The closest is Vlachos's leaping technique[68] for lattice-based systems. Vlachos's algorithm does not, however, use an explicit expression for the time step which, as he has pointed out, can lead to possible errors.[68] The COPS algorithm has been designed to combine the efficiency of the τ-leaping approach and the accuracy afforded by an off-lattice representation. Thus, COPS is a microscopic-level representation whose novelty lies largely in the development of an explicit expression of the time step. Since we are taking into account the exact locations of the species, any off-lattice particle-based method (such as ours) will be computationally more expensive than a “sub-volume" method in which the simplification of a “well-mixed" assumption is invoked. However, the major goal of COPS is to develop a stochastic off-lattice algorithm which is more computationally efficient than a particle-based SSA, with all the inherent advantage that this implies over a simpler well-mixed assumption.

3.1 Overview

In this paper, a new algorithm is developed for the spatio-temporal evolution of an off-lattice system of N molecular species $\{S_1, S_2, ...S_N\}$ diffusing in a d-dimensional system (where d can be 1, 2 or 3). These molecular species interact with each other through M reaction events $\{R_1, R_2, ...R_M\}$ in a fixed volume Ω. We have chosen to model the diffusion of any species as a random walk along orthogonal directions, in a similar manner to the procedure adopted by DADOS.[11–14,57,60] A more general approach would be to allow a random walk in *any* random direction, but this is prohibitively expensive computationally.

For each species, 2d is the number of directions in which the species can diffuse, where d is the dimensionality of the system. For example, if d is 2, diffusion is modeled as a random walk along the orthogonal directions, +x, -x, +y and Ðy. Thus, if 2d is the number of diffusion events for one species, 2Nd will be the total number of diffusion events in the system, and M+2Nd is the total number of reaction and diffusion events combined. Hence, the diffusivity value of each species is denoted by $\{D_1, D_2, ..., D_N\}$.

For each event i, the propensity function is commonly defined as the probability that one event of type i will occur inside a volume Ω in an infinitesimal time interval $[t, t+dt]$. The propensity for reaction event i is defined in terms of two input parameters, a probability rate constant c_i and a reaction radius, r_i. The capture radius (or reaction radius) is defined in such a way that if the distance between any two reactant species is less than the capture radius, they will “react" with a rate defined by c_i. Our interpretation of the propensity follows the same approach used in DADOS, as described above.[11,12,60,63] However, COPS and DADOS differ in the complexity of their treatment of reactivity within the capture radius: In DADOS, all species within the capture radius react instantaneously; whereas, in COPS, they react with some probability rate constant, c_i.

The mathematical basis for our use of a dual input parameter definition of the propensity can be found through its correspondence to the so-called $\lambda - \rho$ model introduced by Oxford mathematicians, Erban and Chapman.[69] As can be seen in the section of their paper that explains the derivation of improved molecular-based models, Erban and Chapman's[69] definitions of λ and ρ are the same as our definitions of c_i and r_i, respectively. Erban and Champman present their algorithm as an improvement over Smoldyn and suggest that it might similarly improve MCell. The main difference between COPS and Erban and Chapman's algorithm[69] is that their $\lambda - \rho$ model is, like DADOS and other off-lattice models, written around the use of SSA, whereas COPS employs an accelerated tau-leaping based approach.

The propensity of a reaction event is denoted by $a_i(t)$, where subscript i refers to reaction R_i. Based on the input parameters defined above, $a_i(t)$ is defined as follows:

$$a_i(t) = c_i h_i(\mathbf{x}). \tag{2}$$

where c_i is the probability rate constant and $h_i(\mathbf{x})$ is the number of molecules for a unimolecular reaction, or the number of reactive pairs for a bimolecular reaction.[60,63] The number of reactive pairs is given by the number of pairs that have a distance less than the capture radius, r_i.

$$a_i^k(t) = d_i X_i^k. \tag{3}$$

X_i^k As mentioned above, the movement of species i with macroscopic diffusivity D_i is modeled as a random walk $S_i(x) \xrightarrow{d_i} S_i(x+h)$. Here x represents the coordinate of the species, h is the jump distance[60] for species i, and d_i is the microscopic diffusive rate constant given by $d_i = D_i/h^2$. The jump distance restricts the particle to move by no more than the diffusion length of the species. Hence, the jump distance is different for different species. The propensity (alternatively named as the "frequency" or the probability per unit time) of the random walk of any species in any direction depends on the population of that species which can diffuse in that direction, multiplied by a diffusive rate constant. Hence, the propensity of species S_i going from location x to location $x+h$ is given by d_iX_i, where X_i is given by the number of molecules that can diffuse in the $+x$ direction. Similarly, the general expression of propensity of any species S_i in k^{th} direction is given by $d_iX_i^k$, where d_i is the diffusive rate constant of species i and X_i^k is the number of molecules of S_i that can diffuse in k^{th} direction. The propensity of a diffusion event is denoted by $a_i^k(t)$, where subscript i denotes the type of diffusing species, and superscript k represents the direction of movement of that species.

3.2 Step-by-Step Procedure

1. **Initialization –** Initialize the system by setting the initial populations for N species and defining all the M reactions in which the species participate. Define a set of “global" parameters which includes values of suitable capture radii for each reaction and the “jump distance" for the movement of each species.

2. **Neighbor List –** Create a separate “neighbor list" for each reaction and diffusion event. Store the particles in each list. The well-established concept of a neighbor list is explained in the Supplemental Information.

3. **Propensities –** Calculate the value of propensity of each event at a given time. The overall propensity is given by sum of all the propensities.

4. **τ-selection –** A variable time step is used here, in contrast to the fixed time step used in other off-lattice reaction-diffusion algorithms. The expression for the time step will be derived in section Section 4.

5. **Switch Step –** A proviso, the “switch step," is added to the algorithm so that whenever the derived value of τ becomes less than a few multiples of the average time interval $(1/a_0)$ of an SSA approach,[21] the COPS algorithm will switch into SSA mode, since SSA would be

faster for such a condition. Thus, if the value of $\tau \leq \frac{f}{a_0}$, the algorithm switches to a standard SSA approach[21] to maintain computational efficiency. Any value from 1-10 can be used for f.[21]

6. **Execution of Events –** If the switch step condition does not hold (hopefully, the majority of situations), calculate the number of executions (firings) for each event. Binomial random numbers are used to calculate the number of firings for each event,[25,26,34] as given by

$$K_i(\tau) = \mathscr{B}\left(\frac{a_i \tau}{tot_i}, tot_i\right). \tag{4}$$

   where tot_i is the maximum number of permitted firings for event i during time step τ. The use of a binomial approach is explained in detail in the Supplemental Information. Execute the events.

7. **Updating –** Globally update the system. Update the time to t = t+τ.

8. **Check Point –** Return to step 2 unless t = $t_{desired}$

4 Evaluation of the time-step in COPS

The COPS algorithm uses the well-established concepts of neighbor list and binomial leaping. These concepts are described in the Supplemental Information. This section describes the selection of the time step τ. In keeping with the τ-leaping framework, the value of the time step is chosen in such a way that propensities of each event do not change by a significant amount.

The following leap condition[21] should satisfy the value of τ used

$$|a_i(t+\tau) - a_i(t)| \leq \varepsilon a_i(t), \quad \forall i = 1, 2...., M + 2Nd \tag{5}$$

where ε is a pre-specified error control parameter ($0 < \varepsilon \ll 1$). The expression sets the relative change in the propensities as being bounded by ε. This strategy can make the algorithm inefficient for small values of $a_i(t)$. This is because, if $a_i(t)$ is close to zero, the leap condition will force τ to approach zero which will make the leaping algorithm inefficient. However, we know that the minimum propensity change that can happen for any event is its probability rate constant c_i.[29] Hence, the right hand side of equation 6 will be given by the upper bound of $\varepsilon a_i(t)$ and c_i, as shown in the following expression:

$$|a_i(t+\tau) - a_i(t)| \leq \max\{\varepsilon a_i(t), c_i\}, \quad \forall i = 1, 2...., M + 2Nd \tag{6}$$

The important goal now is to derive the change of propensity of each event as a function of τ and then use this information in equation (Eq. (6)) to obtain a value for τ. The minimum of all the τ values thus obtained is chosen as the final value of τ. The procedure of obtaining τ values for each event is split into two parts: The first part involves calculating τ for each *reaction* event and then taking the minimum of all the reaction events. This ensuing τ value is denoted by $\tau_{reaction}$. The second part involves obtaining τ for each *diffusion* event and evaluating the minimum τ of all the diffusion events. This minimum τ value is denoted by $\tau_{diffusion}$. The minimum of $\tau_{reaction}$ and $\tau_{diffusion}$ gives us the final value of τ to use at a given time.

$$\tau = \min\{\tau_{reaction}, \tau_{diffusion}\}. \tag{7}$$

4.1 Derivation of spatial τ-selection

In this sub-section, we describe how the time step is derived in order to implement the strategy given in the sub-section above.

4.1.1 Minimum Time Step for Reactions ($\tau_{reaction}$)

To calculate $\tau_{reaction}$, we need to determine the expected value of the change in propensity for each reaction and then use the τ-selection criterion to obtain τ for each reaction. The final step involves taking the minimum value of every one of these τ values. Finding the expected value of the change is the most difficult aspect.

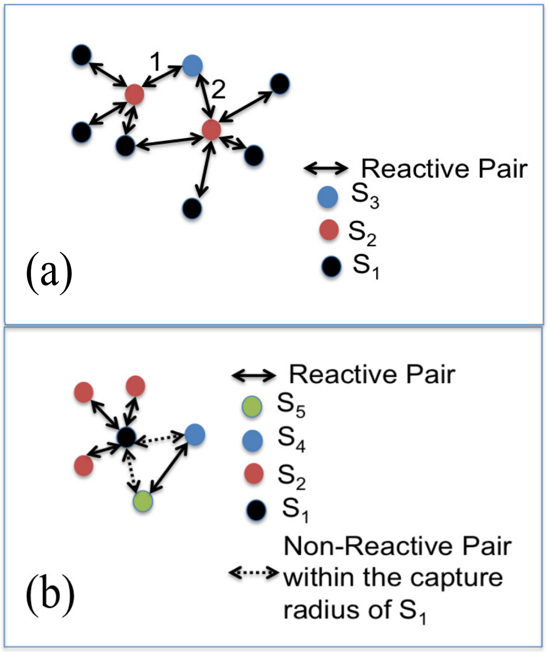

Figure 1: Illustrative examples of the impact factors defined to derive an expression for $\tau_{reaction}$. The impact factor due to *reactant-reactant* effects, p_{ij}, is explained through Figure 1(a); the impact factor due to *reactant-product* effects, q_{ij}, is explained by Figure 1(b).

The important thing that needs to be considered in calculating the expected change is the impact of other events on reaction i. The impact of one event on other events is derived by defining what we have termed an "impact factor." The impact factors ensure that the time step is not large enough to avoid unphysically large changes in the propensities which could violate the leap condition. For example, it restricted the movement of species so that the number of reactive pairs will not increase or decrease by a ε percent. The change in reaction i due to all the events is split into two components: The first is $\Delta a_{iR}(t)$, the change in propensity of the i^{th} reaction because of the execution of all the reactions. The second one is $\Delta a_{iD}(t)$ which is the change due to the diffusion of the species. For the leap condition to be true, the expected value of the total change should be less than $\varepsilon a_i(t)$. Hence, the summation of the expected value of the two components should also be less than $\varepsilon a_i(t)$.

$$|<\Delta a_{iR}(t)> + <\Delta a_{iD}(t)>| \leq \max\{\varepsilon a_i(t), c_i\}. \tag{8}$$

Two separate sets of reactants and products for each reaction are defined: The set S_{iR} is the set of reactants for the i^{th} reaction and S_{iP} is the set of products for the i^{th} reaction. For example, for the reaction $S_1 + S_2 \xrightarrow{k_1} S_3$, S_R is $\{S_1, S_2\}$ and S_P is $\{S_3\}$. The reaction i is affected by reaction j in the following two ways:

1. If reaction R_i has common reactants with reaction R_j, there will be a decrease in the propensity of reaction R_i because of the execution of reaction R_j. We call this eventuality the *reactant-reactant effect*, which is denoted by p_{ij}. p_{ij} is defined as the change in the number of occurrences of reaction R_i per occurrence of R_j. It has a negative value because of the decrease in the number of occurrences. To explain the effect, consider two reactions,

$$\begin{aligned} R_1 : S_1 + S_2 &\xrightarrow{k_1} S_4, \\ R_2 : S_2 + S_3 &\xrightarrow{k_2} S_5. \end{aligned}$$

   As reactions R_1 and R_2 share common reactants (S_2), there will be a decrease in the occurrences of reaction R_1 because of reaction R_2 and *vice versa*. But the decrease will be spatial in nature; it will decrease in regions where (S_1, S_2) reactive pairs has S_3 in the vicinity. In this case, factor p_{12} represents the total change in the occurrences of reaction R_1 because of occurrences of reaction R_2 divided by the total occurrences of reaction R_2. A pictorial representation of factor p_{ij} is shown in Figure 1(a) where one S_2 has three reactive pairs with S_1 and another S_2 has four reactive pairs with S_1. The total number of reaction pairs of type R_1 is seven. The lone S_3 can react with two S_2's. Hence the total number of reactive pairs of type 2 is two. Thus, firing the reactive pair labeled as 1 in the Figure 1(a) will decrease the number of reactive pairs of reaction R_1 by four, and firing reactive pair labeled as 2 will decrease the number of reactive pairs of reaction R_1 by three. Hence, p_{12} (the effect of 2 on 1) is $7/2 = 3.5$. In general, the set of common reactants for reaction i is denoted by S_i^{RR} which is given by $S_{iR} \cap S_{jR} \forall j = 1, ..M$.

2. Reaction R_i can also be affected if the reactants of reaction R_i share the products of reaction

R_j. We denote this impact factor as a *reactant-product effect*, q_{ij}. This factor has a positive value because there is an increase in the number of occurrences of R_i because of reaction R_j. Again, we consider two reactions:

$$\begin{aligned} R_1 : S_1 + S_2 &\overset{k_1}{\longrightarrow} S_3, \\ R_2 : S_4 + S_5 &\overset{k_2}{\longrightarrow} S_2. \end{aligned}$$

When reaction R_2 is executed, there will be an increase in S_2 molecules which can result in an increase of the number of occurrences of reaction R_1. The set representing the reactant-product effect is denoted by S_i^{RP}, which is given by $S_{iR} \cap S_{jP} \forall j = 1, ..M$. Figure 1(b) provides a pictorial representation of this effect: There are three reactive pairs of reaction 1 and one reactive pair of reaction 2 in the Figure 1(b). Executing the reactive pair of reaction 2 will lead to the formation of S_2, which is within the capture radius of S_1. Thus, it will increase the number of occurrences of reaction R_1 by 1. Hence, the value of q_{12} will be 1. The important thing to be addressed here is, while calculating the q_{ij}, we need to know the location of the products: For a bimolecular reaction with *one* product, the location of the lone product is assumed to be that of one the two reactants. For a bimolecular reaction with *two* products, the location of the two products is assumed to be the same as the location of the two reactants.

Using the above two factors, the change in the propensity of reaction R_i due to reaction R_j can be written as:

$$\Delta a_{iR}(t) = c_i(p_{ij}K_j(\tau) + q_{ij}K_j(\tau)). \tag{9}$$

where $K_j(\tau)$ is the number of firings for the reaction R_j and $\Delta a_{iR}(t)$ is the propensity change due to the reaction-reaction effect. Considering the expected value of the change, factors p_{ij} and q_{ij} are already averaged, and the average value of the number of firings, $K_j(\tau)$, is the propensity of the reaction R_j multiplied by the leap interval τ. Therefore,

$$< \Delta a_{iR}(t) > \; = \; c_i\tau(p_{ij}a_j(t) + q_{ij}a_j(t)). \tag{10}$$

Generalizing the above expression using the non-zero factor sets described in the above two subsections, we obtain

$$\begin{aligned} < \Delta a_{iR}(t) > &= c_i \tau (P^{RR} + Q^{RP}), \\ \text{where } P^{RR} &= \sum_{\substack{j=1 \\ j \in S_i^{RR}}}^{M} p_{ij} a_j(t) \text{ and } Q^{RP} = \sum_{\substack{j=1 \\ j \in S_i^{RP}}}^{M} q_{ij} a_j(t). \end{aligned} \tag{11}$$

The next step is to incorporate the expected change in the reaction propensity because of the diffusional movement of species ($< \Delta a_{iD}(t) >$). Consider the following reaction

$$R_1 : S_1 + S_2 \xrightarrow{k_1} S_3.$$

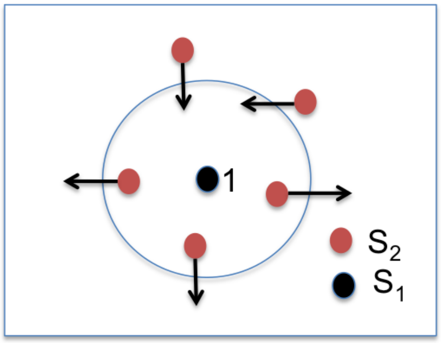

Figure 2: Illustrative example of the impact factor defined to derive a value for $\tau_{diffusion}$. Impact factors arise from the reaction-diffusion effect

As mentioned earlier, species can move in 2d directions. So the movement of species S_1 and S_2 in different directions can increase, or decrease, the number of occurrences of reaction R_1. As shown in Figure 2, some $\{S_1, S_2\}$ reactive pairs can become non-reactive if S_1 and S_2 move outside their capture radius. Similarly, some formerly non-reactive pairs can become reactive once species diffuse within their capture radius. Given that the impact factor is spatial in nature, there will be an increase in reactions in some regions, and a decrease in some other regions. The increase and decrease in reactions in the different regions is obtained using the concept of neighbor lists, described in the Supplemental Information. The impact factor is denoted by the *reaction-diffusion* *effect*, $r^k_{i,j}$, and is defined as the average increase in the number of occurrences of reaction R_i due to species S_j moving in a k^{th} direction. The value of the factor for the i^{th} reaction can be non-zero for the set S^{RD}_i which is given by $S_{iR} \cap S_j \forall j = 1...N$. The set S^{RD}_i contains species which can both react and diffuse. The propensity change in the i^{th} reaction because of species S_j moving in a k^{th} direction is written as:

$$\Delta a_{iD}(t) = c_i(r^k_{i,j} K^k_j(\tau)). \tag{12}$$

where K^k_j is the number of firings for species j moving in a k^{th} direction and $\Delta a_{iD}(t)$ is the propensity change because of the reaction-diffusion effect. Taking the average of equation (Eq. (12)) we obtain,

$$< \Delta a_{iD}(t) >= c_i \tau (r^k_{i,j} a^k_j). \tag{13}$$

Hence the general expression for the change due to diffusion is:

$$\begin{aligned} < \Delta a_{iD}(t) > &= c_i \tau R^{RD}, \qquad (14) \\ \text{where } R^{RD} &= \sum_{\substack{j=1 \\ j \in S^{RD}_i}}^{N} \sum_{k=1}^{2d} r^k_{i,j} a^k_j. \end{aligned}$$

The derived expected value of individual propensity changes can then be put into equation (Eq. (8)) to obtain an expression of τ for the i^{th} reaction

$$\begin{aligned} \tau c_i |(P^{RR} + Q^{RP} + R^{RD}|) &\leq \max\{\varepsilon a_i(t), c_i\}, \\ \tau &\leq \frac{\max\{\varepsilon a_i(t), c_i\}}{c_i(|P^{RR} + Q^{RP} + R^{RD}|)}. \qquad (15) \end{aligned}$$

The important thing to be noted in this expression is that the actual value of the individual changes is taken instead of the absolute value, which can lead to a faulty time step in some examples. As the defined impact factors can have a negative or positive sign depending upon the system, the denominator (the summation of the actual value of all the factors) can become very close to zero. In some cases, this results in an unrealistic value of the time step. This will become more clear if we consider the example of species S_1 and S_2 involved in a reaction $S_1 + S_2 \xrightarrow{k_1} 2S_2$. Ignoring the reaction-diffusion effect, we can have non-zero values of p_{11} and q_{11} at some time, t. If both the factors have approximately the same values, it is possible to obtain a very small value in the denominator of the time step expression in equation (Eq. (13)) because the factors have opposite signs. This, in turn, would lead to an unreasonably high value of the time step. This eventuality requires a modification to the procedure to make the time step expression in such a way that we always take the absolute value of each component which results in the change of propensity. Hence, we can use the following construct:

$$\text{If } c \leq \frac{1}{|a|+|b|} \quad \text{then} \quad c \leq \frac{1}{|a+b|}, \tag{16}$$

$$\text{As } |a+b| \quad \leq \quad |a|+|b|. \tag{17}$$

where a, b, c are non-zero real numbers. This concept is put into use for time step equation (Eq. (13)). The expected change of the propensity of reaction R_i is composed of three components, P^{RR}, Q^{RP} and R^{RD}. The components P^{RR} and R^{RD} consist of such sub-components, which can be positive or negative whereas component Q^{RP} is always positive. To avoid the problem of unrealistic time steps, we make use of the mathematical construct described above and write:

$$\begin{aligned} \text{if } \tau \quad &\leq \quad \frac{\max\{\varepsilon a_i(t), c_i\}}{c_i(P^{RR}_{abs} + Q^{RP}_{abs} + R^{RD}_{abs})} \text{ then } \tau \leq \frac{\max\{\varepsilon a_i(t), c_i\}}{c_i(|P^{RR} + Q^{RP} + R^{RD}|)}, \\ \text{where } P^{RR}_{abs} \quad &= \quad \sum_{\substack{j=1 \\ j \in S_i^{RR}}}^{M} |p_{ij}| a_j(t),\ Q^{RP}_{abs} = \sum_{\substack{j=1 \\ j \in S_i^{RP}}}^{M} q_{ij} a_j(t) \text{ and } R^{RD}_{abs} = \sum_{\substack{j=1 \\ j \in S_i^{RD}}}^{N} \sum_{k=1}^{2d} |r^k_{i,j}| a^k_j(t). \end{aligned}$$

The above improved formula will provide an expression for the time step for the R_i reaction. The time step for all M reactions are calculated in a similar manner, and the minimum value of all the time steps will be $\tau_{reaction}$:

$$\tau_{reaction} = \min_{i \in [1,M]} \left\{ \frac{\max\{\varepsilon a_i(t), c_i\}}{c_i(P^{RR}_{abs} + Q^{RP}_{abs} + R^{RD}_{abs})} \right\}. \tag{18}$$

4.1.2 Minimum Time Step for Diffusion ($\tau_{diffusion}$)

The procedure to determine $\tau_{diffusion}$ parallels the sub-section above for reactions and hence is not repeated here. Details are, however, provided in the Supplemental Information. Table 1 provides a summary of all the essential expressions derived for the time step related to diffusion.

Table 1: τ-selection formulae

$$tau = \min\{\tau_{reaction}, \tau_{diffusion}\},$$

$$\tau_{reaction} = \min_{i \in [1,M]} \left\{ \frac{\max\{\varepsilon a_i(t), c_i\}}{c_i(P^{RR}_{abs} + Q^{RP}_{abs} + R^{RD}_{abs})} \right\},$$

$$\text{where } P^{RR}_{abs} = \sum_{\substack{j=1 \\ j \in S^{RR}_i}}^{M} |p_{ij}| a_j(t),\ Q^{RP}_{abs} = \sum_{\substack{j=1 \\ j \in S^{RP}_i}}^{M} q_{ij} a_j(t) \text{ and } R^{RD}_{abs} = \sum_{\substack{j=1 \\ j \in S^{RD}_i}}^{N} \sum_{k=1}^{2d} |r_{i,jk}| a_{jk}(t),$$

$$\tau_{diffusion} = \min_{\substack{i \in [1,N] \\ k \in [1,2d]}} \left\{ \frac{\max\{\varepsilon a^k_i(t), d_i\}}{d_i \sum_{j=1}^{M} |\nu_{ij}| a_j(t)} \right\}.$$

5 Implementation Issues

5.1 Symmetric Nature of the Reactant-Reactant Effect (p_{ij})

The time step expression we derived here is complicated in nature and, consequently, involves coding a significant number of entities whose impact on computational speed has to be taken into consideration. The effect of propensities on each other in off-lattice, particle-based systems is completely spatial in nature and depends upon the process that is simulated. However, we have made an observation that can significantly reduce the calculations that need to be considered and, thus, improve computational efficiency. This observation tells us that the common reactant factor, p_{ij}, is derived from a *symmetric* matrix. The reactant-reactant effect, p_{ij}, is given by the total decrease in the number of occurrences of reaction R_i (P_{ij}) due to the reaction R_j, divided by total number of occurrences (X_j) of reaction R_j. The matrix P_{ij} is symmetric.

$$p_{ij} = \frac{P_{ij}}{X_j}. \tag{19}$$

Since the symmetric nature of P_{ij} is not intuitive, this is most easily explained through an example. Consider two biomolecular reactions which have common reactants:

$$R_1 : S_1 + S_2 \xrightarrow{k_1} S_3,$$
$$R_2 : S_2 + S_4 \xrightarrow{k_2} S_5.$$

If the matrix is symmetric, P_{12} should equal P_{21}. This can be confirmed by referring to Figure 3 which shows the reactant-reactant effect. In that figure, the value of P_{12} is obtained by calculating the decrease in the number of occurrences of reaction R_1 due to the execution of each occurrence of type R_2. Thus, firing the lone pair of $\{S_2, S_3\}$ will consume S_2, resulting in the decrease of three occurrences of reaction R_1. Hence the value of P_{12} is 3. The procedure to calculate P_{21} is performed in the same way, but in reverse. Now we see how firing reaction R_1 affects reaction R_2. There are three occurrences of reaction R_1. Firing each reaction will consume the only S_2 in the system which then decreases the number of occurrences of reaction R_2 by one. But, since we have three total occurrences of reaction R_1, the net effect for P_{21} will be 1+1+1=3. Thus P_{ij} is a symmetric matrix. This result is very useful when we have a number of reactions with common reactants, as this will cut the calculation of p_{ij} in half.

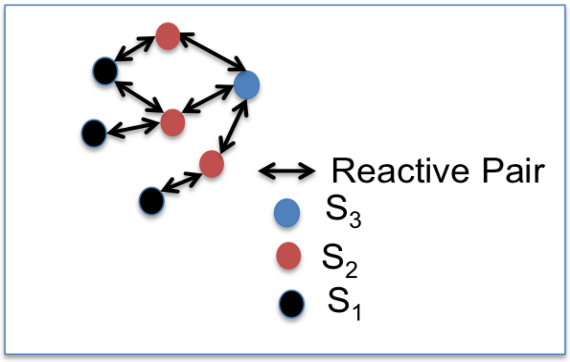

Figure 3: Illustrative example to explain the symmetric nature of impact factor, p_{ij}

5.2 Time Step Issue

Unlike the concept of individual propensities for each sub-volume in spatial leaping algorithms at mesoscopic level, the *overall* propensity of each event in its entire domain is defined in COPS. This may result in some technical issues, since dealing with the expected change in the diffusional propensity of the events is very challenging. Consider a simple system of two species that diffuse and react with each other.

$$R_1 : S_1 + S_2 \xrightarrow{k_1} S_3.$$

q_{11}, for this system is zero. The value of p_{11} will be non-zero, since firing one occurrence of reaction R_1 will lead to a decrease in the number of occurrences because of common reactants. To simplify the expression, we ignore the reaction-diffusion effect. Hence, from equation 19, we can write

$$\tau_{reaction} = \frac{\max\{\varepsilon a_1(t), k_1\}}{p_{11}k_1a_1(t)}. \tag{20}$$

To simplify the above equation, we assume $\max\{\varepsilon a_1(t), k_1\}$ is $\varepsilon a_1(t)$. Equation 27 is modified to,

$$\tau_{reaction} = \frac{\varepsilon}{p_{11}k_1}. \tag{21}$$

We further assume that $\tau_{reaction}$ is the minimum of $\tau_{reaction}$ and $\tau_{diffusion}$, hence the value of the time step is $\tau_{reaction}$. Now, if the value of the rate constant, $k_1 \ll 1$, the value of the time step is very much greater than 1. The important thing to note is that the above expression for the time step does not depend on the population of species, S_1 and S_2. If the population of both the species is relatively small, a high value of time step can result in an excessive number of firings for the movement of species.

The above issue regarding the movement of species can be resolved by restricting the number of firings of the diffusion events. The value of the time step is adjusted in such a way that it will move only ε percent of the number of available species in that time step.

6 Numerical Examples

We now test the algorithm by considering a couple of representative example test cases.

6.1 Fisher's Equation

Fisher's equation,[70,71] also known as the Fisher-KPP equation, is one of the simplest examples of a non-linear reaction-diffusion equation. It has been used, for example, to describe the propagation of an advantageous gene in a population[70] or the spatio-temporal evolution of species under the combined effects of diffusion and logistical growth.[71] The equation is perhaps best known for its traveling wave solutions. It is a two-component reaction-diffusion system with the following reaction:

$$R_1 : S_1 + S_2 \xrightarrow{k_1} 2S_2.$$

where S_1 and S_2 are species diffusing in a 1d system. The dynamics of traveling wave solutions arises by saturating one end of the boundary by species S_2 and distributing the S_1 over the rest of the domain. Assuming 'no flux' boundary conditions, the domain will start to fill with S_2 as S_1 and S_2 react due to the diffusion of both the species towards each other. Thus, the traveling waves of S_2 can be followed in time.

We consider a 1d domain with 'no flux' boundary conditions applied at each end. We start by saturating one end of the boundary by species S_2 and distributing S_1 in the rest of the boundary as mentioned in the above paragraph. S_1 and S_2 can diffuse in both +x and -x directions with a diffusivity, D, and can react with each other with a rate constant, k_1.

For this test case, the total concentration of species is taken as 120 molecules/meter and we choose the concentration of species S_2 to be 6 molecules/meter. The values of the diffusivity, D, and probability rate constant, k_1, are taken to be 0.001 m^2/s and 1 s^{-1}, respectively. The value of the capture radius and jump distance are taken as 0.2 m and 0.01 m, respectively. Using the above set of parameters, simulations were run for different domain sizes, keeping the concentration fixed. The COPS results are compared with predictions from SSA as a means to check the performance and accuracy of our newly developed algorithm.

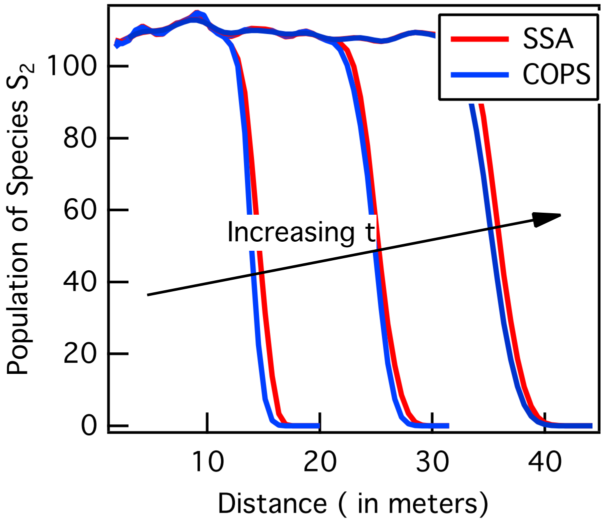

Figure 4: Comparison of snapshots of Fisher equation solutions from SSA and COPS at different times. The three snapshots correspond to time, t= 2, 4 and 6 seconds, respectively. Simulations are run for a domain size of 60 m.

Figure 4 shows three snapshots of the traveling wave of S_2 obtained by solving Fisher's equation using COPS , as well as those for the corresponding SSA solution. The snapshots are taken at t= 2, 4 and 6 seconds for domain size of length = 60 m. It can be clearly seen that COPS shows an almost exact match with SSA. This validates the qualitative accuracy of the COPS for this system. For a more quantitative assessment, we also calculated the wave front speed predicted by COPS and SSA. The wave front speed represents how fast the system domain is filled with species S_2. The distance until species S_2 is saturated increases with time and that distance is named the "saturation distance." The saturation distance is extracted at different times and plotted against time. The slope of this plot gives us the value of the speed of the wave front. Figure 5 shows the saturation distance profiles as a function of time for both COPS and SSA, and show considerable similarity. The value of wave front speed for COPS was found to be 5.1 m/s, which matches closely with the SSA value of 5.2 m/s.

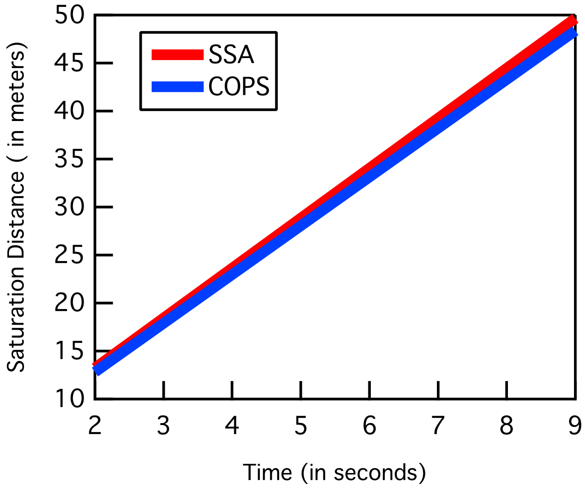

Figure 5: Fisher Equation example: Saturation distance versus time for both SSA (red) and COPS (blue). Simulations are run for a domain size of 60 m.

Figure 6 shows an analysis of the computational cost of COPS compared to SSA. As shown on a log scale in Figure 6(a), both algorithms initially take much the same time. There is no significant computational gain to use COPS for smaller populations as it is well known that SSA is computationally efficient under those conditions. But the relative inefficiency of the SSA algorithm can be clearly seen for larger populations, where there is a much more rapid increase in computational time for SSA compared to COPS as the population size increases. The comparison is clearer in Figure 6(b) where the ratio of the computational speed of COPS to SSA is plotted for different populations. The ratio of speeds increases linearly with population. COPS is about 3 times faster for a population size of 2400; whereas at higher populations, for example at a population value of 35000, COPS is about 50 times faster than SSA. This shows that COPS is very efficient at larger populations. This occurs without loss of accuracy relative to SSA.

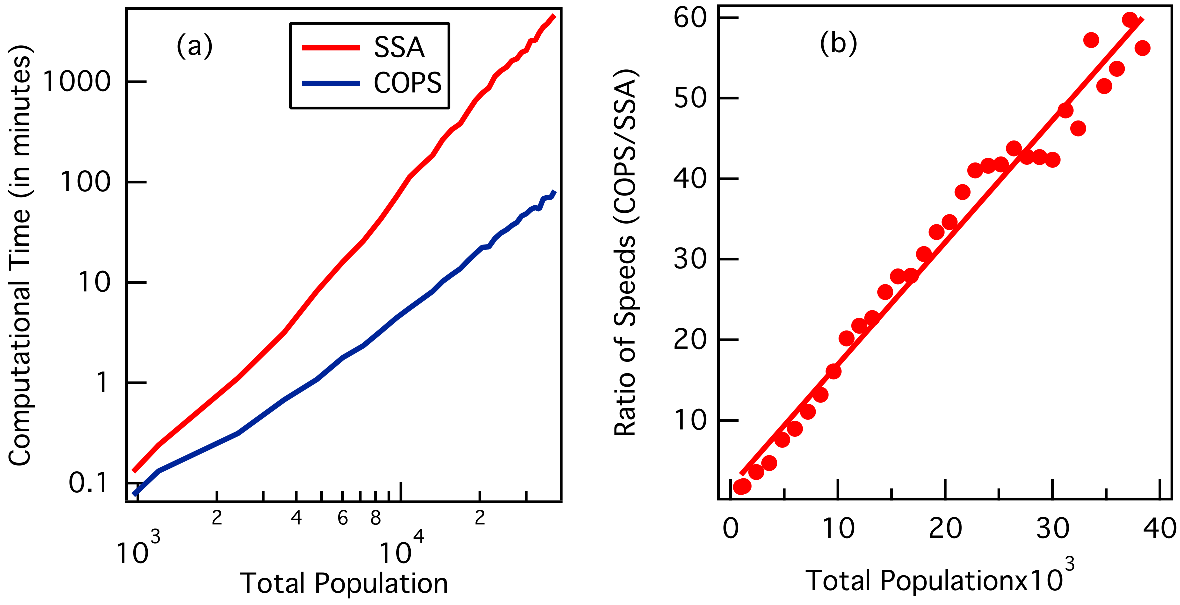

Figure 6: Comparison of computational speed between SSA and COPS. (a) Log-log plot of the computational time versus population size for SSA and COPS and (b) ratio of speeds of COPS to SSA for different populations.

6.2 Lotka-Volterra Equations

The results shown above for the Fisher's equation example suggests that COPS is as computationally efficient as SSA for systems with sufficiently high populations. In this second numerical example, we compare the efficiency of SSA and COPS for "rare event" type systems. In such systems, there is often an important event with a small rate constant. We might expect SSA to take longer to execute in this circumstance, as it will preferentially execute other events with higher rate constants. In contrast, COPS can be expected to be much more efficient than SSA, given that it has the ability to fire multiple events in a single time interval. This capability will results in the firing of rare events in fewer iterations for any given system.

Lotka-Volterra equations[72,73] are best known for describing the interactions between two species in an ecosystem, predators and prey. We model the equations as a stochastic reaction-diffusion system with predator and prey species involved in the following reactions:

$$\begin{aligned} R_1 : S_1 + S_2 &\xrightarrow{k_1} 2S_2, \\ R_2 : S_1 &\xrightarrow{k_2} 2S_1, \\ R_3 : S_2 &\xrightarrow{k_3} \phi. \end{aligned}$$

Here S_1 and S_2 are assumed to be prey and predator, respectively. The first reaction corresponds to predator-prey interactions. “Branching" of prey (the generation of offspring) is described by reaction 2. Death of predators is represented by the third reaction.

We consider a 2d system with periodic boundary conditions. Species S_1 and S_2 can diffuse in four directions perpendicular to each other with diffusivity value, D. The value of D is taken to be 1 m^2/s. The values of the rate constants k_1, k_2 and k_3 are set to be 0.2 s^{-1}, 10 s^{-1} and 10 s^{-1}, respectively. We start with an initial population of 1000 for both predator and prey. Both the species are uniformly distributed in a periodic 2d domain of length and width given by 20 m. The capture radius of the predator-prey interaction reaction and the jump distance of the species are assumed to be 3.5 m and 0.1 m, respectively. Simulations using SSA and COPS were run using the above set of parameters. The resulting dynamics of the system is explained below.

The spatio-temporal evolution of the system shows different patterns, some snapshots of which are shown in Figure 7. Figure 7(a) is a snapshot taken at time t=0 which shows predators and prey initially uniformly distributed throughout the domain. The high value of the capture radius results in a large number of predator-prey interactions which ultimately leads to a decrease in the number of prey and an increase in the number of predators. This is captured in Figure 7(b), showing fewer red dots (prey) and a plethora of blue dots (predators). As the number of prey starts to decrease, the predators start to die as there are fewer prey to eat. The high value of predator death rate chosen in this example leads to rapid depletion of predators. This, in turn, helps the population of prey to recover and leads to the creation of clusters of prey, as shown in Figure 7(c). However, as the size of prey clusters start to increase, the reaction involving predator-prey interactions becomes active

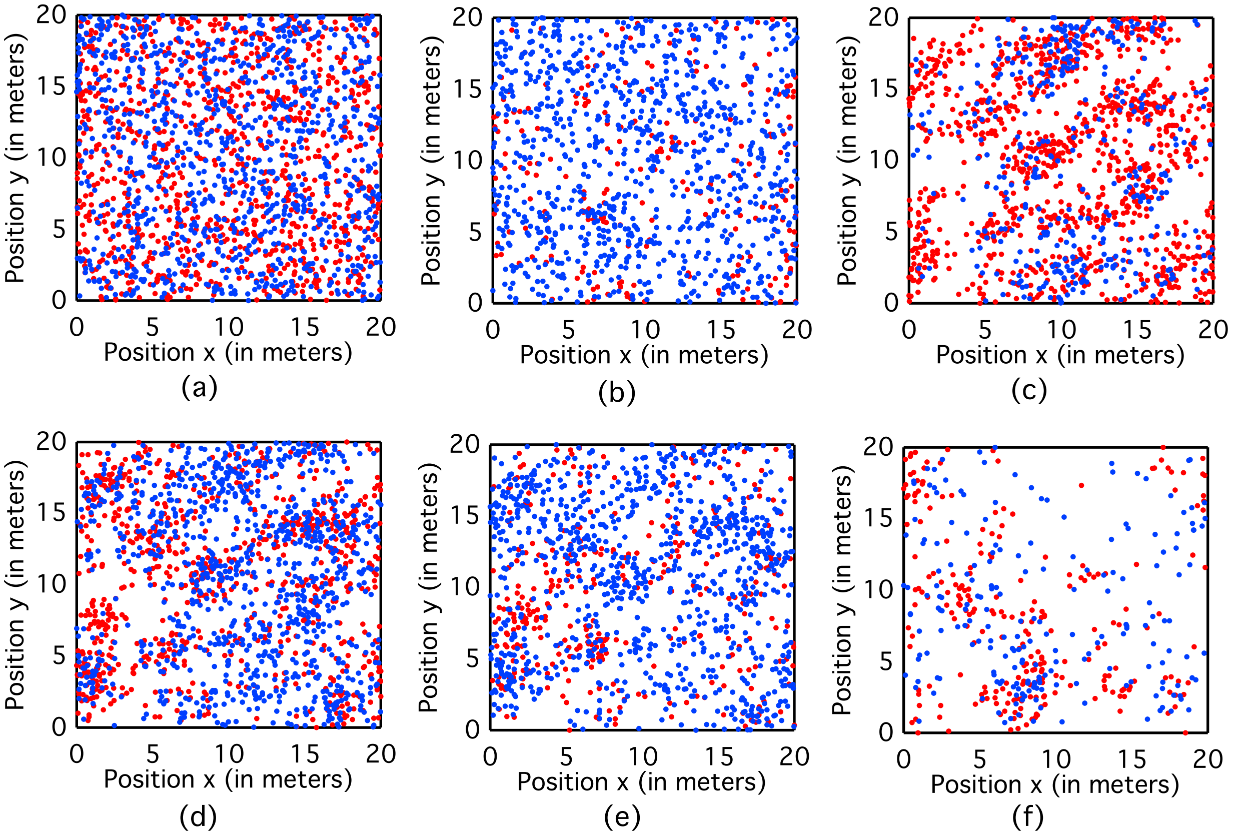

Figure 7: Snapshots of the Lotka-Volterra system at six different times (corresponding to 0, 0.1, 0.5, 0.7, 0.8 and 1.2 seconds from a-f, respectively) calculated using COPS . Blue and red dots show the locations of predators and prey, respectively.

once again. A cluster involving both predators and prey is formed, as depicted in Figure 7(d). As time progresses, all the prey clusters are converted into predator clusters, as seen in Figure 7(e). Breaking all the prey clusters results in a decrease in prey, which eventually decreases the number of predators because of fewer available predator-prey interactions. As shown in Figure 7(f), large voids can be created in the domain. The fewer number of predators results once more in the formation of prey clusters, thus repeating the cycle. Hence, the formation of predator-prey clusters and the creation of voids at different times correspond to oscillations with different amplitudes.

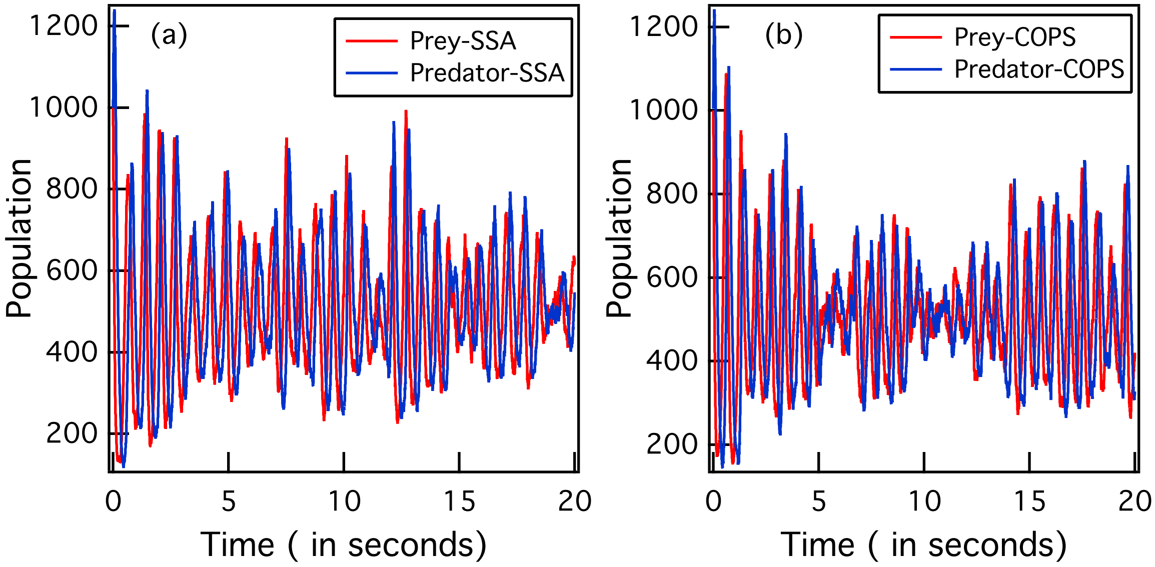

Figure 8: Variation of the populations of predator and prey as a function of time obtained using (a) SSA and (b) COPS. Both profiles were obtained by running the simulations for a k_1 value of 0.2 s^{-1}.

As explained above, oscillations are seen in the simulated profiles shown in Figure 8; the system does not remain at one stationary point. The oscillations that develop are centered around the stable solution which is the mean of the oscillations. The concentration profiles obtained from SSA and COPS are shown in Figure 8. COPS captures the same oscillating trends as SSA, although an exact match between the concentration profiles of SSA and COPS at a given time point cannot be made due to the inherently stochastic behavior of the system.

All the above described results are shown for one set of parameters. However, given that the predator-prey interaction is one of the most important events in the system, we have explored the accuracy and efficiency of COPS by running simulations for different values of the rate constant of predator-prey interactions. It is expected that, as the value of the rate constant of an important event decreases, the efficiency of SSA will decrease as the system evolves into the category of rare events.

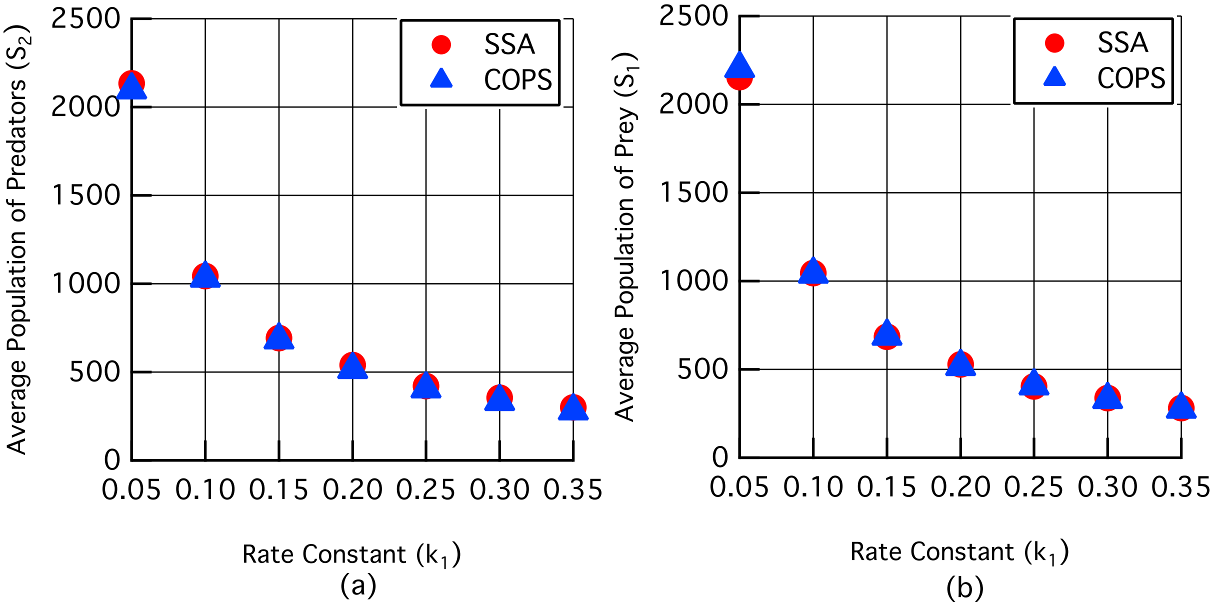

Figure 9: Average population of (a) predators and (b) prey for different values of k_1, the rate constant governing predator-prey interaction.

To check the accuracy of COPS , the mean values of oscillations for predator and prey were extracted from each k_1 value and compared with those derived from SSA. The comparison of SSA and COPS is plotted in Figure 9. Both the trend and the values show almost an exact match with SSA. The decreasing trend of mean predator and prey values with an increase in k_1 can be explained by the fact that higher values of k_1 lead to more predator-prey interactions. This results in a decrease in the size of prey clusters at a given time, which ultimately decreases the size of predator clusters. Hence the system will oscillate at smaller average values of predators and prey.

Figure 10 shows a comparison of the computational speed for different values of k_1 used. It can be seen that, at high values of the rate constant, COPS is about 10-15 times faster than SSA.

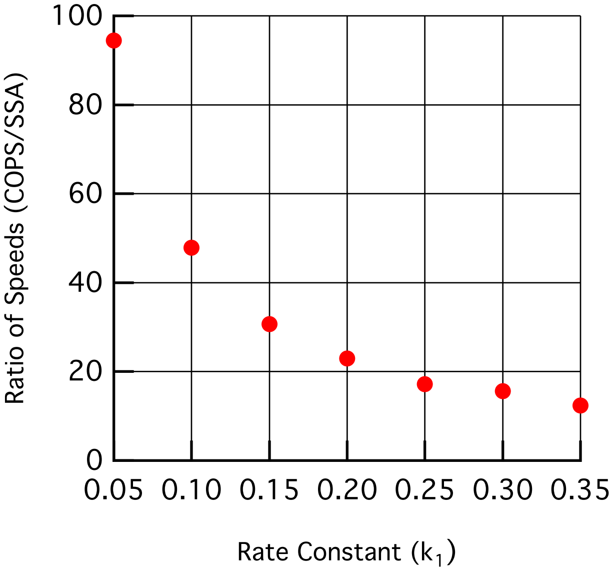

Figure 10: Ratio of simulation speeds between COPS and SSA for different values of the rate constant of predator-prey interactions, k_1.

As the value of rate constant k_1 decreases, COPS becomes much more computationally efficient than SSA. The SSA tends to fire events with high rate constants, which rules out the important event here (predator-prey interaction). The capability of firing multiple events in COPS removes that limitation and the ratio of speeds of COPS and SSA increases as the value of k_1 decreases. For example, when k_1 is as small as 0.05 s^{-1}, the COPS is almost 90 times faster than SSA.

This comparison of computational speeds for different values of k_1 is analogous to results obtained in our previous work,[15] where our Partitioned Leaping Algorithm (PLA) is used to quantify stochastic effects in biochemical reaction networks. The results showed that CPU times of the PLA and SSA converge as the binding-unbinding rate constant for one of the important reactions is increased. Similarly here, we have showed the ratio of speeds between SSA and COPS converges to 1 as the rate constant k_1 is increased.

7 Conclusions

We have developed an accelerated, τ-leap based, KMC algorithm to handle off-lattice reaction-diffusion systems at a microscopic level. The cornerstone in this algorithm is an explicit expression for τ, derived through the use of "impact factors," which relate how one event affects another. The mathematical foundation for our treatment of bimolecular reactions mirrors that for the $\lambda - \rho$ model introduced by Erban and Chapman[69] who cite their, and by extension our, algorithm's improvements over other off-lattice KMC methods to handle situations that would otherwise be incorrectly represented. The algorithm developed here has the capability to simulate both unimolecular and bimolecular reactions. Three-body collisions (trimolecular reactions) have not been simulated in the present algorithm. Indeed, it would be a significant challenge to extend the current algorithm to such reactions as this would involve the definition of three cut-off radii in order to calculate the number of triplets, since each particle would need to be within the cut-off radius of one another for reaction to occur. This identification would impose a heavy computational burden since calculating the number of triplets would need to be updated each time the reaction is executed.

Our implementation used computational techniques such as the concept of neighbor lists to calculate the propensities in an efficient way, and hash map data structures to calculate the number of reactive pairs, *etc.*. This significantly reduced the computational burden. The expression developed for the time step involves considerable calculations, but its computational cost is significantly decreased by our observation that the reactant-reactant effect, p_{ij}, is derived from a symmetric matrix. The use of binomial random numbers resolved the unwanted issues of particle overlap and negative populations.

The simulation results suggest that the τ-leap method introduced in the algorithm can be made to work with a similar level of accuracy to the SSA, but in much more computationally efficient way. The efficiency and accuracy of the algorithm was tested by taking Fisher's equation and Lotka-Volterra equations as numerical examples. The results for the Fisher's equation showed a significant gain in computational efficiency for systems with higher population. Results using the Lotka-Volterra equations shows that the algorithm is particularly efficient for "rare event" type systems. While both the Fisher and Lotka-Volterra examples are well-used examples for testing algorithmic capabilities, the Lotka-Volterra equations are nonetheless a source of rich and complex dynamics, manifested in many kinds of oscillations and chaotic patterns. In this paper, for example, our use of Lotka-Volterra equations exhibited some interesting dynamics as a result of changing the rate constant of predator-prey interactions: We showed that, at higher rate constant values of predator-prey interaction, the off-lattice system oscillates at smaller average predator and prey values. The interesting dynamics arises from the fact that the higher rate constant value of predator prey interaction results in large number of interactions between predators and prey. The larger is the number of interactions, the smaller will be the size of prey clusters. The smaller sizes of prey clusters leads to smaller sizes of predator clusters, which ultimately results in smaller mean values of predator prey clusters.

The code will be made available upon request in open source format as a GNU General Public License. Future work will test the performance and efficiency of the algorithm in a variety of more challenging materials science-focused applications.

Acknowledgement

We sincerely thank Dr. Leonard Harris at the University of Pittsburgh for carefully reading the manuscript and making numerous changes to improve its readability. The Semiconductor Research Corporation (award 2012-VJ-2272) is thanked for financial support of VT in the generation of this code. We also thank the Cornell Engineering Learning Initiative (ELI), funded by Intel Corporation, for financial support of Philip Berard as an undergraduate researcher.

References

(1) Arkin, A. P.; McAdams, H. H. *Genetics* **1998**, *149*, 1633.

(2) McAdams, H. H.; Arkin, A. *Trends Genet.* **1999**, *15*, 65.

(3) Elowitz, M. B.; Levine, A. J.; Siggia, E. D.; Swain, P. S. *Science* **2002**, *297*, 1183.

(4) Fedoroff, N.; Fontana, W. *Science* **2002**, *297*, 1129–1131.

(5) Rao, C. V.; Wolf, D. M.; Arkin, A. P. *J. Chem. Phys.* **2003**, *118*, 4999.

(6) Raser, J. M.; O'Shea, E. K. *Science* **2004**, *304*, 1811–1814.

(7) Kærn, M.; Elston, T. C.; Blake, W. J.; Collins, J. J. *Nat. Rev. Genet.* **2005**, *6*, 451–464.

(8) Raser, J. M. *Science* **2005**, *309*, 2010–2013.

(9) Plummer, J. D.; Griffin, P. B. *Nucl. Instrum. Methods Phys. Res. B* **1995**, *102*, 160.

(10) Roy, S.; Asenov, A. *Science* **2005**, *309*, 388.

(11) Pelaz, L.; Jaraiz, M.; Gilmer, G. H.; Gossmann, H. J.; Rafferty, C. S.; Eaglesham, D. J.; Poate, J. M. *Appl. Phys. Lett.* **1997**, *70*, 2285.

(12) Jaraiz, M.; Pelaz, L.; Rubio, E.; Barbolla, J.; Gilmer, G. H.; Eaglesham, D. J.; Gossmann, H. J.; Poate, J. M. *Mater. Res. Soc. Symp. P.* **1998**, *532*, 43–54.

(13) Pelaz, L.; Marques, L.; Aboy, M.; Lopez, P.; Barbolla, J. *Comput. Mater. Sci.* **2005**, *33*, 92–105.

(14) Pelaz, L.; Aboy, M.; Lopez, P.; Marques, L. *J. Vac. Sci. Technol. B* **2006**, *24*, 2432.

(15) Harris, L.; Piccirilli, A.; Majusiak, E.; Clancy, P. *Phys. Rev. E* **2009**, *79*.

(16) Bortz, A. B.; Kalos, M. H.; Lebowitz, J. L. *J. Comp. Phys.* **1975**, *17*, 10.

(17) Gillespie, D. T. *J. Comput. Phys.* **1976**, *22*, 403–434.

(18) Chatterjee, A.; Vlachos, D. G. *J. Comput.-Aided Mater. Des.* **2007**, *14*, 253–308.

(19) Gillespie, D. T. *J. Phys. Chem.* **1977**, *81*, 2340–2361.

(20) Gillespie, D. T. *J. Chem. Phys.* **2000**, *113*, 297.

(21) Gillespie, D. T. *J. Chem. Phys.* **2001**, *115*, 1716.

(22) Gillespie, D. T.; Petzold, L. R. *J. Chem. Phys.* **2003**, *119*, 8229.

(23) Rathinam, M.; Petzold, L. R.; Cao, Y.; Gillespie, D. T. *J. Chem. Phys.* **2003**, *119*, 12784.

(24) Cao, Y.; Li, H.; Petzold, L. *J. Chem. Phys.* **2004**, *121*, 4059.

(25) Tian, T.; Burrage, K. *J. Chem. Phys.* **2004**, *121*, 10356.

(26) Chatterjee, A.; Vlachos, D. G.; Katsoulakis, M. A. *J. Chem. Phys.* **2005**, *122*, 024112.

(27) Cao, Y.; Gillespie, D. T.; Petzold, L. R. *J. Chem. Phys.* **2005**, *123*, 054104.

(28) Harris, L. A.; Clancy, P. *J. Chem. Phys.* **2006**, *125*, 144107.

(29) Cao, Y.; Gillespie, D. T.; Petzold, L. R. *J. Chem. Phys.* **2006**, *124*, 044109.

(30) Auger, A.; Chatelain, P.; Koumoutsakos, P. **2006**, *125*, 084103.

(31) Marquez-Lago, T.; Leier, A.; Burrage, K. *BMC Syst. Bio.* **2010**, *4*, 19.

(32) Cao, Y.; Gillespie, D. T.; Petzold, L. R. *J. Chem. Phys.* **2007**, *126*, 224101.

(33) Li, T. *Multiscale Model. Simul.* **2007**, *6*, 417.

(34) Peng, X.; Zhou, W.; Wang, Y. *J. Chem. Phys.* **2007**, *126*, 224109.

(35) Pettigrew, M. F.; Resat, H. *J. Chem. Phys.* **2007**, *126*, 084101.

(36) Kuwahara, H.; Mura, I. *J. Chem. Phys.* **2008**, *129*, 165101.

(37) Anderson, D. F. *J. Chem. Phys.* **2008**, *128*, 054103.

(38) Ferm, L.; Hellander, A.; Lotstedt, P. *J. Chem. Phys.* **2010**, *229*, 343.

(39) Marquez-Lago, T.; Burrage, K. *J. Chem. Phys.* **2007**, *27*, 104101.

(40) Roh, M. K.; Daigle, B. J.; Gillespie, D. T.; Petzold, L. R. *J. Chem. Phys.* **2011**, *135*, 234108.

(41) Lampoudi, S.; Gillespie, D. T.; Petzold, L. R. *J. Chem. Phys.* **2012**, 1–17.

(42) Ramaswamy, R.; Sbalzarini, I. F. *J. Chem. Phys.* **2012**, 1–18.

(43) Slepoy, A.; Thompson, A. P.; Plimpton, S. J. *J. Chem. Phys.* **2008**, *128*, 205101.

(44) Bernstein, D. *Phys. Rev. E.* **2005**, *71*, 041103.

(45) Rossinelli, D.; Bayati, B.; Koumoutsakos, P. *Chem. Phys. Lett.* **2008**, *451*, 136–140.

(46) Iyengar, K. A.; Harris, L. A.; Clancy, P. *J. Chem. Phys.* **2010**, *132*, 094101.

(47) Koh, W.; Blackwell, K. T. *J. Chem. Phys.* **2011**, *134*, 154103.

(48) Koh, W.; Blackwell, K. T. *J. Chem. Phys.* **2012**, *137*, 154111.

(49) Eaglesham, D. J.; Stolk, P. A.; Gossmann, H. J.; Poate, J. M. *Appl. Phys. Lett.* **1994**, *65*, 2305–2307.

(50) Jaraiz, M.; Gilmer, G. H.; Poate, J. M.; de la Rubia, T. D. *Appl. Phys. Lett.* **1996**, *68*, 409.

(51) Pelaz, L.; Gilmer, G. H.; Gossmann, H. J.; Rafferty, C. S.; Jaraiz, M.; Barbolla, J. *Appl. Phys. Lett.* **1999**, *74*, 3657.

(52) Sadigh, B.; Lenosky, T. J.; Theiss, S. K.; Caturla, M. J.; de la Rubia, T. D.; Foad, M. A. *Phys. Rev. Lett.* **1999**, *83*, 4341–4344.

(53) Jaraiz, M.; Rubio, E.; Castrillo, P.; Pelaz, L.; Bailon, L.; Barbolla, J.; Gilmer, G. H.; Rafferty, C. S. *Mat. Sci. Semicon. Proc.* **2000**, *3*, 59–63.

(54) Pelaz, L.; Marqués, L. A.; Aboy, M.; Barbolla, J.; Gilmer, G. *Appl. Phys. Lett.* **2003**, *82*, 2038.

(55) Deák, P.; Gali, A.; Sólyom, A.; Ordejón, P.; Kamarás, K.; Battistig, G. *J. Phys.: Condens. Matter* **2003**, *15*, 4967.

(56) Ortiz, C.; Cristiano, F.; Colombeau, B.; Claverie, A.; Cowern, N. *Mater. Sci. Eng. B* **2004**, *114-115*, 184–192.

(57) Martinbragado, I.; Avci, I.; Zographos, N.; Jaraiz, M.; Castrillo, P. *Solid-State Electronics* **2008**, *52*, 1430–1436.

(58) Mirabella, S.; De Salvador, D.; Bruno, E.; Napolitani, E.; Pecora, E.; Boninelli, S.; Priolo, F. *Phys. Rev. Lett.* **2008**, *100*.

(59) Pelaz, L.; Marques, L.; Lopez, P.; Santos, I.; Aboy, M. *Nucl. Instrum. Meth. B* **2007**, *255*, 95–100.

(60) Pelaz, L.; Marqués, L. A.; Aboy, M.; Lopez, P.; Santos, I. *Eur. Phys. J. B* **2009**, *72*, 323–359.

(61) Plimpton, S. J.; Slepoy, A. *J. Phys.: Conf. Ser.* **2005**, *16*, 305–309.

(62) Kerr, R. A.; Bartol, T. M.; Kaminsky, B.; Dittrich, M.; Chang, J. C. J.; Baden, S. B.; Sejnowski, T. J.; Stiles, J. R. *SIAM J. Sci. Comput.* **2008**, *30*, 3126.

(63) Andrews, S. S.; Bray, D. *Phys. Biol.* **2004**, *1*, 137–151.

(64) Andrews, S. S.; Addy, N. J.; Brent, R.; Arkin, A. P. *PLoS Comput. Biol.* **2010**, *6*, –.

(65) Burrage, K.; Burrage, P.; Leier, A.; Marquez-Lago, T.; Nicolau Jr, D. *Design and Analysis of Biomolecular Circuits: Engineering Approaches to Systems and Synthetic Biology* **2011**, 43–62.

(66) Yang, J.; Monine, M.; Faeder, J.; Hlavacek, W. *Phys. Rev. E.* **2008**, *78*.

(67) Sneddon, M. W.; Faeder, J. R.; Emonet, T. *Nat. Meth.* **2010**, *8*, 177–183.

(68) Vlachos, D. G. *Phys. Rev. E* **2008**, *78*, 046713.

(69) Erban, R.; Leier, A.; Chapman, S. J. *Phys. Biol.* **2009**, *6*, 046001.

(70) Fisher, R. A. *Ann. Eugen.* **1937**, *7*, 353–369.

(71) A. Kolmogorov, I. P.; Piscounov, N. *In Selected Works of A. N. Kolmogorov I*; Kluwer, Dordrecht, 1991; pp 248–270.

(72) Täuber, U. C. *J. Phys.: Conf. Ser.* **2011**, *319*, 012019.

(73) Mobilia, M.; Georgiev, I. T.; Täuber, U. C. *J. Stat. Phys.* **2006**, *128*, 447–483.

Supporting Information Available

Supplemental information entails the explanation of familiar concepts including step by step procedure of Stochastic Simulation Algorithm (SSA), neighbor lists and binomial leaping. This material is available free of charge via the Internet at `http://pubs.acs.org/`.