Mesoscopic reaction-diffusion in intracellular signaling

Johan Elf, Andreas Dončić and Måns Ehrenberg*
Dept. of cell and molecular biology, Uppsala University, Box 596, Uppsala, Sweden

ABSTRACT

Mesoscopic modeling of intracellular kinetics is usually performed on the premise that diffusion is so fast that all concentrations are homogenous in space. However, this supposition is not necessarily valid even for small prokaryotic cells, as indicated by recent experimental data on intracellular diffusion constants. When diffusion and spatial heterogeneity are taken into account, stochastic simulation of chemical reactions in single cells is computationally demanding. We present an efficient Monte Carlo algorithm for simulation of mesoscopic reaction-diffusion kinetics in single cells. The total system (e.g. a single prokaryotic cell) is divided into *N subvolumes* (*SVs*), chosen so small that the concentrations of reactants in a *SV* are near-homogeneous in space. The molecules in a *SV* can either undergo chemical reactions or diffuse to a neighboring *SV*. The expected time for the next chemical reaction or diffusion event is only recalculated for those *SVs* that were involved in the previous event. The time for the next event in each *SV* is ordered in an event queue, which makes the computation time linear in log *N,* rather then in *N*.

Keywords: Reaction-diffusion, Gillespie, spatial, master equation

1. INTRODUCTION

Recent experiments have shown that the stochastic nature of intracellular kinetics must be taken into account in the description of biologically important reactions[1-5]. Mesoscopic modeling of gene expression and other intracellular reactions has been carried out[6-13]. The need for stochastic descriptions of gene expression is in line with intuition, since the numbers of individual genes and mRNAs per cell are small. However, fluctuations can play important roles also in systems with many copies of molecules. This has been demonstrated for microtubule formation[14], ultrasensitive modification and de-modification reactions[15], plasmid copy number control[16], noise-induced oscillations[17] and metabolite concentrations[18].

Common to most mesoscopic approaches is the assumption of a spatially homogenous mixture of all reactants, so that the chemical master equation[19] can be applied. However, recent experimental results on protein mobility in the cytoplasm of *E. col* [20] suggest that this assumption might not be valid even in small bacterial cells. When diffusion of chemical reactants is comparatively slow, the standard master equation must be replaced by a state description that accounts for local concentrations[19,21].

* ehrenberg@xray.bmc.uu.se; phone +46 18 4714213; fax +46 18 4714262; www.icm.molbio.uu.se

Fluctuations and Noise in Biological, Biophysical, and Biomedical Systems, Sergey M. Bezrukov, Hans Frauenfelder, Frank Moss, Editors, Proceedings of SPIE Vol. 5110 (2003) © 2003 SPIE · 0277-786X/03/$15.00

2. THE REACTION-DIFFUSION MASTER EQUATION

To obtain a master equation formulation of chemical reactions in systems with non-uniformly distributed concentrations due to finite diffusion constants (D), we divide the total system volume Ω in N cubic *subvolumes*† (SVs) of volume Δ and side length ℓ. Diffusion is treated as a memoryless random walk, in that molecules jump between neighboring SVs with elementary rate constants given by D/ℓ^2. The SVs must be chosen sufficiently small, so that the probability distributions of their reactants can be treated as uniform. This means that the rate by which two molecules in a *SV* react should not depend on their initial locations. This condition is fulfilled when $\ell^2 \ll Dt_R$ for all reactants, where t_R as the mean time between two reactions for a single molecule[19]. The current numbers of molecules of all reactants in all individual cells define the state of the system. The set of reactions that change the state of the system is extended in relation to the master equation for homogenous systems by the inclusion of all diffusion jumps that bring a molecule from a *SV* to a neighboring one. The following example with five elementary chemical reactions will be used to illustrate the approach:

$$\begin{array}{ll} \varnothing \xrightarrow{k_1} \mathrm{A} & \mathrm{A} \xrightarrow{\mu} \varnothing \\ \varnothing \xrightarrow{k_1} \mathrm{B} & \mathrm{B} \xrightarrow{\mu} \varnothing \\ \mathrm{A+B} \xrightarrow{k_2} \varnothing & \end{array} \qquad (1)$$

Molecules (A and B) are synthesized randomly (rate constants k_1), and disappear when they react with each other (second order rate constant k_2) or when they are degraded (first order rate constant μ). This common kinetic motif has interesting mesoscopic properties in parameter regions with near-critical system behavior[18]. When diffusion is very fast, the concentrations of the reactants in Eq. (1) will be uniform in space. In this limit, the state of the system is fully described by the numbers of A and B molecules. Furthermore, the transitions between states $\{A,B\}$ are confined to the following elementary reactions

$$\begin{array}{ll} \{A,B\} \xrightarrow{\Omega k_1} \{A+1,B\} & \{A,B\} \xrightarrow{\mu A} \{A-1,B\} \\ \{A,B\} \xrightarrow{\Omega k_1} \{A,B+1\} & \{A,B\} \xrightarrow{\mu B} \{A,B-1\} \\ \{A,B\} \xrightarrow{\Omega^{-1}\cdot A\cdot B\cdot k_2} \{A-1,B-1\} & \end{array} \qquad (2)$$

The state dependent intensity of each transition is written over each arrow. The constants k_1 and k_2 have the units Ms^{-1} and M^1s^{-1} and the corresponding intensities are proportional to Ω or Ω^{-1}, respectively. The master equation for the homogeneous system is

$$\frac{dP}{dt} = \Omega k_1\left(\mathbb{E}^{-1,0}-1\right)P + \Omega k_1\left(\mathbb{E}^{0,-1}-1\right)P + \Omega^{-1}k_2\left(\mathbb{E}^{1,1}-1\right)ABP - \mu\left(\mathbb{E}^{1,0}-1\right)AP + \mu\left(\mathbb{E}^{0,1}-1\right)BP \qquad (3)$$

Here, $P = P\left(\{A,B\},t\right)$ and the step operator[19] $\mathbb{E}$ is defined from $\mathbb{E}^{i,j} f\left(A,B\right) = f\left(A+i,B+j\right)$, where f is a function of the state variables A and B.

When diffusion is slow, the spatial distribution of A and B molecules will not be uniform and local concentrations must be accounted for. To do this, we use a state space consisting of the numbers (A_i, B_i) of A and B molecules in each of the N subvolumes (*SVs*) of the system. The state of the system changes when a chemical reaction takes place in a *SV* or when a molecule diffuses from one *SV* to a neighboring one. The diffusion of an A molecule from *SV* γ to *SV* λ is described as

$$\left\{..,A_\lambda,..,A_\gamma..\right\} \xrightarrow{A_\gamma d^A_{\lambda\gamma}} \left\{..,A_\lambda+1,..,A_\gamma-1,..\right\}$$

† In the physics literature subvolumes are often called cells, which can be misleading in a biological context.

and the same type of scheme describes the diffusion of B molecules.

The rate constants for jumps between neighboring *SVs* are $d^A_{\lambda\mu} = d^A_{\mu\lambda} = D/\ell^2$ and otherwise zero. With this choice of $d_{\lambda\mu}$, the random walk approximates the diffusion equation,

$$\frac{\partial p(\mathbf{r},t)}{\partial t} = D\nabla^2 p(\mathbf{r},t),$$

for the probability density $p(\mathbf{r},t)$ that a particle is at position **r** at time *t*[22]. The discrete random-walk treatment of diffusion is exact in the limit of infinitely small *SVs*, and it approximates correct reaction-diffusion behavior when jumps between *SVs* occur much more frequently than the chemical reactions.

In this model, the reaction-diffusion master equation for the time dependent probability distribution $P = P(\{A_1, B_1, .., A_\lambda, B_\lambda, .., A_N, B_N\}, t)$ for the number of A and B molecules in the system's subvolumes is given by

$$\begin{aligned}\frac{dP}{dt} = &\sum_\lambda \Big[\Delta k_1 \left(\mathbb{E}_\lambda^{-1,0} - 1\right) P + \Delta k_1 \left(\mathbb{E}_\lambda^{0,-1} - 1\right) P + \Delta^{-1} k_2 \left(\mathbb{E}_\lambda^{1,1} - 1\right) A_\lambda B_\lambda P - \mu\left(\mathbb{E}_\lambda^{1,0} - 1\right) A_\lambda P + \mu\left(\mathbb{E}_\lambda^{0,1} - 1\right) B_\lambda P\Big] + \\ &\sum_\lambda \sum_{\gamma \neq \lambda} \Big[\left(\mathbb{E}_\lambda^{-1,0}\mathbb{E}_\gamma^{1,0} - 1\right) d^A_{\lambda\gamma} A_\gamma P + \left(\mathbb{E}_\lambda^{0,-1}\mathbb{E}_\gamma^{0,1} - 1\right) d^B_{\lambda\gamma} B_\gamma P\Big]\end{aligned} \tag{4}$$

The step operator $\mathbb{E}_\lambda^{i,j}$ is defined as $\mathbb{E}^{i,j}$ above for changes confined to the state of the subvolume λ. The reaction-diffusion master equation and its range of validity have previously been discussed [19,21,23].

The reaction-diffusion master equation has a much larger state space than its homogeneous counterpart. It keeps track of the *M* different molecular species in the *N* different subvolumes that participate in *R* different types of chemical reactions. Since one subvolume has six neighbors, the number of elementary reactions that can change the state of the system is *N(6M+R)* instead of *R* as in the homogeneous case. Since *N*-values larger than 10^5 often are required for realistic system descriptions, the number of elementary reactions is dramatically increased when reaction-diffusion is taken into account.

Furthermore, the number of states in the reaction-diffusion formulation of the master equation increases exponentially with *N*. This is because all combinations of copy numbers in all subvolumes have to be considered. For a system with *M* species in *N* subvolumes with a maximum of A_{max} molecules of each species in a subvolume the number of states is $A_{\max}^{M \cdot N}$. If, to give an example, *M=2*, *N=50* and $A_{max}=10$, there are 10^{100} states. Therefore, the state vector for most realistic systems cannot be stored in any computer memory, which rules out direct numerical solutions of the reaction-diffusion master equation.

An alternative approach is to use Monte Carlo (MC) simulations of reaction-diffusion systems. In this work, we have shown how application of an efficient version [24] of the Gillespie algorithm [25] to reaction-diffusion systems as defined by Eq. (4) can significantly reduce the computation times.

3.THE SPATIAL NEXT REACTION ALGORITHM

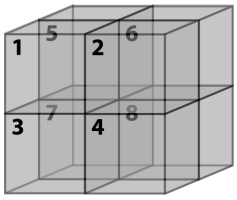
3.1 GILLESPIE'S DIRECT METHOD

Gillespie's direct method [25] is an algorithm for Monte Carlo simulations of mesoscopic kinetics in spatially homogeneous systems. Trajectories, statistically correct according to the homogeneous master equation Eq. (3), are generated by the execution of one elementary reaction at the time. The time for the next reaction is sampled from an exponential distribution with a mean time equal to the inverse of the sum of all reaction intensities. Which reaction to execute is chosen randomly with a probability determined by its normalized intensity. Iteration number n of the algorithm has the following steps

1. Calculate the intensities $a_i(\mathbf{x}_n)$ [*unit "per second"*] of all possible elementary reactions $i=1,2,...R$ for the state $\mathbf{x}_n$, a vector that specifies the copy numbers of the different molecular species in the system in iteration number n.
2. Calculate the total reaction intensity, $r(\mathbf{x}_n) = \sum_i a_i(\mathbf{x}_n)$.
3. Sample the time interval Δt between a previous event (n) and the next one ($n+1$) according to $\Delta t = -ln(rand)/r$, where *rand* is a random number uniformly distributed between 0 and 1.
4. Sample the next reaction i according to the probability $P(i)=a_i/r$.
5. Update the state vector $\mathbf{x}_n$ to account for the reaction.

This direct use of the Gillespie technique is feasible for spatially homogenous systems. However, when spatial coordinates must be taken into account, the method becomes far too slow, as noted by Gillespie[25].

One way to speed up MC simulations of trajectories in reaction-diffusion systems is to take advantage of the fact that reaction intensities only change in those particular subvolumes where a reaction or diffusion event has occurred. In each iteration it is therefore sufficient to recalculate only those intensities that belong to the subvolume, where an event has occurred. This reduces the number of recalculated intensities from $N(M+R)$ to maximally $2(M+R)$. In addition, the time to search through all subvolumes to find where the next event occurs can be significantly reduced by using an indexed event queue for the reaction times, as described by Gibson and Bruck [24]. With this method, one can store the subvolumes where an event has occurred in a time that increases logarithmically, rather than linearly, with the number of subvolumes in the system. We will explain this method, here applied to reaction-diffusion systems for the first time, by giving an outline of the iteration steps followed by more detailed descriptions of selected items.



SV	→	←	↓	↑	↗	↙
1	2	1	3	1	5	1
2	2	1	4	2	6	2
3	4	3	3	1	7	3
4	4	3	4	2	8	4
5	6	5	7	5	5	1
6	6	5	8	6	6	2
7	8	7	7	5	7	3
8	8	7	8	6	8	4

Connectivity matrix

#A	*#B*	*#C*
10	2	0
9	1	3
5	0	2
7	1	1
4	0	2
7	1	3
8	2	4
5	0	2

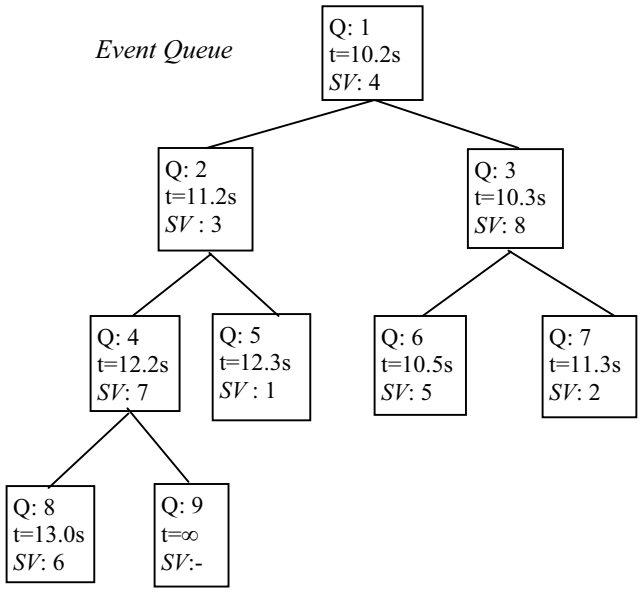
Configuration

$r\ [s^{-1}]$	$d\ [s^{-1}]$	$r+d\ [s^{-1}]$
2.2	10	12.2
4.2	11.3	15.5
2.3	5.4	7.3
1.4	6.4	7.8
0.4	4.3	4.7
0.5	10.3	10.8
1.0	13.3	14.3
5.3	5.4	10.7

Rate matrix

Q
5
7
2
1
6
9
4
3

Q-array



Event Queue array

Q	*SV*	*t [s]*
1	4	10.2
2	3	11.2
3	8	10.3
4	7	12.2
5	1	13.3
6	5	10.5
7	2	11.3
8	6	13.0

Figure 1: Data structures The structures within solid borders are arrays used in the algorithm.
For each *SV* the *connectivity matrix* (N×6) stores the neighbors' indices. This defines the geometry and boundary conditions for the system, as illustrated for 2×2×2 subvolumes (*top left*).
The *configuration matrix* (N×M) stores the current number of molecules of each species in each *SV*.
The *rate matrix* (N×3) stores the sum of reaction rate constants (*r*) and the sum of diffusion rate constants (*d*).
The *Q-array* specifies the position of each *SV* in the *event queue.*

In the *event queue,* the elements are ordered such that, in each branch of the binary tree, a subvolume with an earlier event time (*t*) is higher up. The event times (*t*) depend on the time for the previous event in the same *SV*; and on the *SV*'s reaction and diffusion rates (*r*+*d*) through a random number.
The queue can be represented in an *array* (N×2), as seen to the right.

3.2 THE SPATIAL NEXT REACTION METHOD

Initialization

1. Generate a *connectivity matrix* (see Fig.1, legend)
2. Distribute the initial numbers of molecules between the subvolumes and store them in the *configuration* matrix (see Fig.1, legend).
3. Calculate the sum, $r_\alpha = \sum_{j=1}^{R} a_{j\alpha}$, of intensities ($a_{j\alpha}$) for chemical reactions (j) in the subvolume α and store it in the *rate matrix* (Fig. 1, legend*).* The reaction intensities are calculated by using the volume Δ of the *SV* and the number of molecules in the *SV* to obtain the current concentrations.
4. Calculate the sum, $s_\alpha = \sum_{j=1}^{M} d_j X_j^\alpha$, of diffusion intensities $\left(d_j X_j^\alpha\right)$ in the subvolume α and store it in the *rate matrix.* The parameter $d_j = D_j / \ell^2$ is the rate constant for jumps between neighboring subvolumes for species *j*, as defined above. X_α^j is the number of molecules of species *j* in subvolume α and *M* is the number of different molecular species in the system.
5. Calculate the sum, $r_\alpha + s_\alpha$, for each subvolume and generate a random number, *rand,* uniformly distributed in [0,1]. This number samples the time for the first reaction-diffusion event in each subvolume as $t_\alpha = -\ln(rand)/(r_\alpha + s_\alpha)$.
6. Store the t_α in the *event queue array*, in such a way that all branches of the *event queue* are sorted with increasing event time. (Fig. 1, legend)

Iterations

7. The next reaction-diffusion event will occur at time t_λ in the subvolume, $\alpha=\lambda$, that is at the top of the *event queue*. The event will be a chemical reaction if a newly generated $rand < r_\lambda/(r_\lambda + s_\lambda)$, and otherwise a jump out from the volume by diffusion.
8. Chemical reaction event ($rand < r_\lambda/(r_\lambda + s_\lambda)$)
    a. Rescale *rand* to [0,1], by dividing it with r_λ, and use the updated *rand* to sample which chemical reaction, *i*, that has occurred in subvolume λ according to the probability $P(i)=a_{i\lambda}/r_\lambda$.
    b. Update the elements in the configuration matrix that belong to the subvolume where the chemical event occurred.
    c. Recalculate the sum, $r_\lambda + s_\lambda$, in this subvolume and generate a new *rand* in [0,1] to obtain the time of the next reaction-diffusion event in this subvolume $t_\lambda^{next} = t_\lambda - \ln\left(rand\right)/\left(r_\lambda + s_\lambda\right)$.
    d. Reorder the branch of the event queue with subvolume λ according to the value of t_λ^{next} (see below).
9. Diffusion event ($rand > r_\lambda/(r_\lambda + s_\lambda)$).
    a. Rescale *rand* from paragraph 7. above according to $(rand - r_\lambda)/(1 - r_\lambda)$ and use the rescaled *rand* to sample which species, *i*, that diffused out from the subvolume according to the probability distribution $P(i) = d_i X_i^\lambda / s_\lambda$.
    b. The neighboring subvolume, γ, to which the diffusion event is targeted is sampled by randomly choosing one of the six columns in the connectivity matrix.
    c. Update the states of these *SVs* by removing a molecule of species *i* from *SV* λ and adding it to *SV* γ. Recalculate the sums, $r_\lambda + s_\lambda$ and $r_\gamma + s_\gamma$, for the SV and its neighbor where events have occurred. Generate two new random numbers, *rand*1 and *rand*2, and sample the times when the next reaction or diffusion events occurs in the subvolumes, $t_\lambda^{next} = t_\lambda - \ln\left(rand1\right)/\left(r_\lambda + s_\lambda\right)$ and $t_\gamma^{next} = t_\lambda - \ln\left(rand2\right)/\left(r_\gamma + s_\gamma\right)$.
    d. Reorder the *event queue* according to the values of t_λ^{next} and t_γ^{next} (see below).
10. Return to 7 for the next iteration.

The event queue and its representation in an array
The event queue allows the subvolume, where the next reaction-diffusion event will occur in the system, to be found without a search through all subvolumes. The event queue is created as a binary tree (Fig.1). The element positions in the queue are denoted Q=1,..,N. Each element stores the index of one subvolume and the time for the next event in that subvolume. Each branch of the queue is ordered from early to late event times, with the earliest time at the top There are several ways to achieve an ordering according to this principle. When the time for the next event has changed for one subvolume, the corresponding element in the queue swaps positions in the branch until correct order is restored.

A convenient way to store the event queue is in an event queue array (Fig. 1). Each row in the array corresponds to one element of the queue, where the Q-number for the element is used as row index. By using this ordering, the element over element $Q=k$ in the branched queue is placed in the array row with index Q=(k/2) truncated to an integer. The elements below $Q=k$ in the branch queue have the row indices $Q=2k$ and $Q=2k+1$.

In addition to the *event* queue array there is a Q-array (Fig. 1) that stores the position, Q, in the event queue array for each SV. The Q-array is necessary to find the element in the queue that corresponds to the neighbor of the active cell.

The connectivity matrix and boundary conditions
In order to rapidly find the index of a neighboring *SV*, we generate a lookup table; the connectivity matrix. Each row in the matrix corresponds to one *SV* so that *SV* and row indices are the same. The indices for each of the six neighbors to a *SV* are stored in six different columns, which determine the geometry of the system. (Fig. 1)

Closed (reflecting) boundaries prevent diffusion of molecules out of the total volume; and is implemented by assigning the same index to *SV*s at and outside the boundary. This is illustrated for a 2×2×2 volume in Fig. 1.

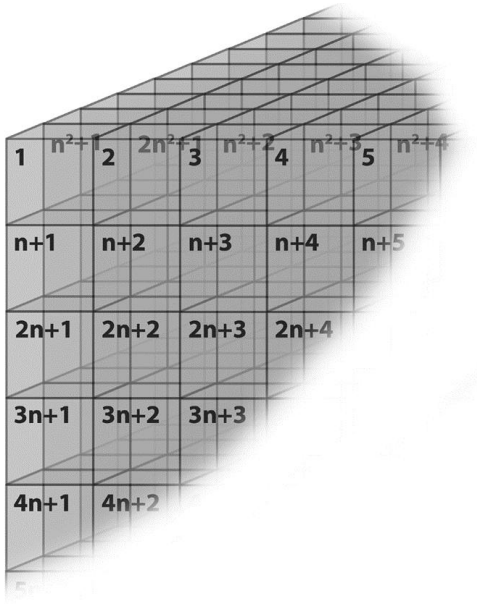


Figure 2: An example of indexing n^3 *SV*s.

3.3 PERFORMANCE ISSUES

Reaction intensities have to be recalculated for one or two subvolumes per iteration among a total of N subvolumes in the system. The event queue makes the time required per iteration proportional to *log(N),* rather than to N as with the direct Gillespie approach[25]. These improvements make mesoscopic simulations of reaction-diffusion systems feasible, since one iteration will not take much more time than for a spatially uniform system. However, diffusion can only be approximated as jumps between subvolumes of finite size, conditional on that these jumps occur much more frequently than the chemical reactions. This is the necessary and sufficient condition for the concentrations of reactants to be near uniformly distributed within a subvolume and smoothly varying between subvolumes. If the algorithm is arranged so that the frequency of diffusion events is hundred times that of chemical reaction events, there will be approximately hundred times more iterations in an MC simulation of a system with reaction-diffusion (Eq. 4) than in the corresponding homogeneous system (Eq. 2).

During the simulation of a reaction-diffusion system, at least $N(M+8)$ integers must be stored for the connectivity-, configuration-, Q-, and event queue-arrays (see fig. 1) as well as N real numbers for the times in the event queue. If the memory of the computer allows, storage also of the $N(R+3)$ real numbers for the individual reaction intensities ($a_{i\lambda}, r_\lambda, s_\lambda, r_\lambda + s_\lambda$) can be used to decrease the computation times.

3.4 IMPLEMENTATION AND PERFORMANCE

The algorithm has been implemented in C/C++, using the Intel C++ 7.0 compiler. Simulations were run on a Pentium 4 2.53GHz with 512Mb PC800 RAM. The chemical system used to illustrate the method is defined by Eq. (1) and the corresponding reaction-diffusion master equation Eq. (4). The parameters were k_1=1μMs^{-1}, k_2=10^{6}μM^{-1}s^{-1} and μ=0.001s^{-1} [18]. With these parameter choices there were on the average about 600 molecules of each species in a total volume Ω=1 μm^3.

How the average computation time per iteration depends on the system volume, Ω, for fixed subvolume size ($\Delta = \ell^3 = 1\mu m^3$) and diffusion constant (D=10^{-6}*cm*^{2}*s*$^{-}$1) is shown in Fig. 3A. For this choice of diffusion constant and *SV* size, the ratio between frequencies of reaction and diffusion events was 0.015. The computation time was approximately proportional to *log(N),* with a deviation that could be ascribed to an increasing memory usage as *N* increases.

The computation time to simulate one second of a stochastic reaction-diffusion system for different diffusion constants is shown in Fig. 3B. The system volume was 8 μm^3 and the ratio between the frequencies of chemical reaction and diffusion events was always kept close to 0.015 by tuning of the size of subvolumes in relation to the value of the diffusion constant. The computation time to simulate the reaction-diffusion system was about a thousand times longer than the time to simulate the homogenous system. This is explained by that: (i) There was about 67 times more diffusion events than chemical reaction events in the reaction-diffusion system and each diffusion event leads to changes in two subvolumes. This makes the reaction-diffusion simulation 134=67×2 times slower than the homogenous simulation. (ii) For each event in the reaction-diffusion simulation, one or two branches of the event queue has to be reordered. This makes every iteration a few times longer, depending on the number of subvolumes.

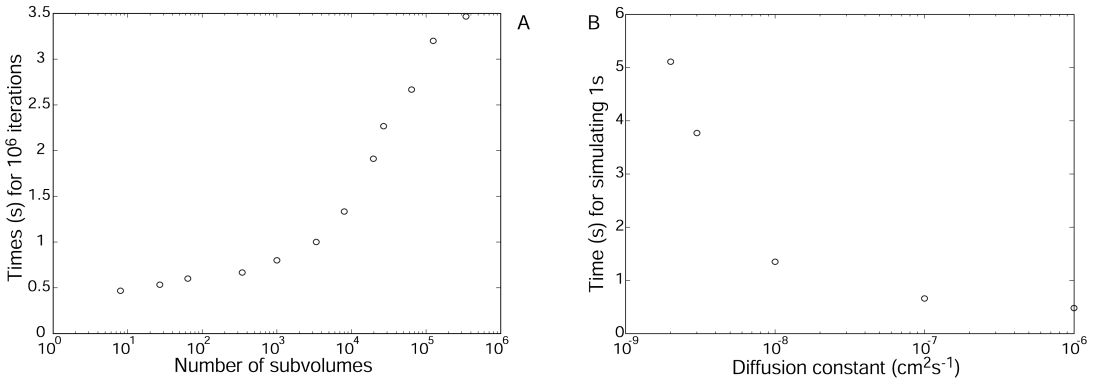


Figure 3: Scaling properties. *A*. *x-axis*: The time required for 10^6 iterations for different number of cells (*y-axis*). The reaction-diffusion ratio is kept at 0.015 (D=10^{-6}*cm*^{2}*s*$^{-1}$ and $\ell = 1\mu m$). *B. y-axis*: The time required to simulate the chemical system for 1 second at different diffusion constants (*x-axis*). The volume is constant at 8μm^3. The reaction-diffusion ratio is kept at 0.015 by varying the number of subvolumes. {8, 216, 8000, 64000, 125000} subvolumes for D={10^{-6}, 10^{-7}, 10^{-8}, 3·10^{-9}, 2·10^{-9}}c*m*^{2}*s*$^{-1}$, respectively. The time required to simulate the corresponding homogenous system without diffusion is 0.40ms.

4. DISCUSSION

Stochastic simulations of intracellular chemical reactions are typically carried out in model systems with reactants that are uniformly distributed in space. This means that a molecule is assumed to have the same probability to interact with any of its reaction partners, irrespective of how these are located in space. That is, each molecule will diffuse throughout the whole cell volume with high probability before it participates in a chemical reaction. This condition of fast (global) diffusion compared to the (local) rate of chemical reactions is, however, often violated. Spatially non-uniform systems will therefore arise in many different types of cells, and accurate descriptions of their properties will require the design of fast computer algorithms of a new type.

In this work, we have developed an efficient Monte Carlo method for simulation of reaction-diffusion systems in single cells. The algorithm is a combination of the reaction-diffusion master equation [19,21,23] and the efficient MC method suggested by Gibson and Bruck[24]. The algorithm allows for simulation of mesoscopic reaction-diffusion in realistic systems, by making the computation time logarithmic, rather than linear, in the number of subvolumes of the system.

Many homogeneous master equations converge to the same deterministic equation in the macroscopic limit[19]. Similarly, there is a multitude of mesoscopic reaction-diffusion systems with very different properties, that all converge to the same homogenous master equation in the limit of very fast diffusion. This is illustrated by the snap-shots of stationary distributions of molecules in the "AB-system" from Eq. (1) shown in Figs. 4A and B. Molecules of type A are synthesized in one corner and B molecules in the opposite corner of the cubic volume of the system. Fig. 4A illustrates a situation with relatively slow and Fig. 4B a situation with relatively fast diffusion. In the former case, the spatial origins of synthesis of A and B molecules clearly shape their distributions. In the latter, the A and B molecules have almost uniform distributions as described by Eq. (3).

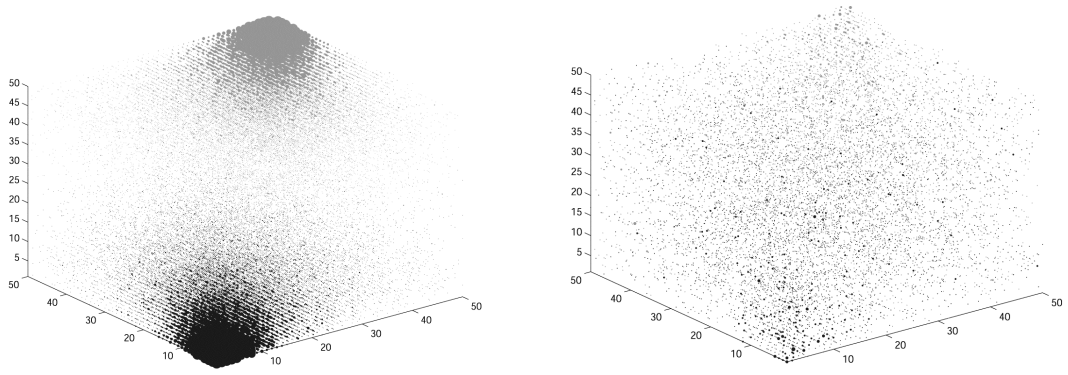


Figure 4: Snap-shot of systems with 50x50x50 subvolumes. The size of the dots is proportional to the number of molecules in the subvolumes. All A molecules (*gray*) are synthesizes in one corner and the B molecules (*black*) in the other. In *A* $D=1.5\cdot10^{-9}cm^2s^{-1}$ and in *B* $D=10^{-7}cm^2s^{-1}$. In B the system approaches the limit where the homogenous master equation applies.

Some biological systems have been designed to control events that occur at distinct locations in space. Examples from bacteria are reviewed in [26]. In all such cases, reaction-diffusion couplings must be taken into account to obtain an understanding of system behavior. However, reaction-diffusion approaches will be necessary to analyze chemical networks also of other types. One such case is provided by systems with multiple steady states. These can be in different attractors in different regions in space, and this has profound effects on their kinetics (J. Elf, manuscript in preparation). Accurate modeling of the dynamic behavior of such systems, which is beyond reach for theories based on spatial uniformity, will be crucial to understand the switching between functionally distinct states in λ-phage kinetics [27] and signal cascades [28].

ACKNOWLEDGEMENTS

This work was supported by the Swedish Research Council and the National Graduate School of Scientific Computing. We thank Martin Lovmar for help with the graphics and Paul Sjöberg for valuable comments on the manuscript.

REFERENCES

1. Becskei, A. & Serrano, L. Engineering stability in gene networks by autoregulation. *Nature* **405**, 590-3 (2000).
2. Gardner, T., Cantor, C. & Collins, J. Construction of a genetic toggle switch in Escherichia coli. *Nature* **403**, 339-342 (2000).
3. Ozbudak, E. M., Thattai, M., Kurtser, I., Grossman, A. D. & van Oudenaarden, A. Regulation of noise in the expression of a single gene. *Nat Genet* **31**, 69-73 (2002).
4. Cluzel, P., Surette, M. & Leibler, S. An ultrasensitive bacterial motor revealed by monitoring signaling proteins in single cells. *Science* **287**, 1652-5 (2000).
5. Elowitz, M. B., Levine, A. J., Siggia, E. D. & Swain, P. S. Stochastic gene expression in a single cell. *Science* **297**, 1183-6 (2002).
6. Berg, O. G. A model for the statistical fluctuations of protein numbers in a microbial population. *J Theor Biol* **71**, 587-603 (1978).
7. McAdams, H. H. & Arkin, A. Stochastic mechanisms in gene expression. *Proc Natl Acad Sci U S A* **94**, 814-9 (1997).
8. Thattai, M. & van Oudenaarden, A. Intrinsic noise in gene regulatory networks. *Proc Natl Acad Sci U S A* **98**, 8614-9 (2001).
9. Ko, M. S. Induction mechanism of a single gene molecule: stochastic or deterministic? *Bioessays* **14**, 341-6 (1992).
10. Cook, D. L., Gerber, A. N. & Tapscott, S. J. Modeling stochastic gene expression: implications for haploinsufficiency. *Proc Natl Acad Sci U S A* **95**, 15641-6 (1998).
11. Kepler, T. B. & Elston, T. C. Stochasticity in transcriptional regulation: origins, consequences, and mathematical representations. *Biophys J* **81**, 3116-36 (2001).
12. Hasty, J., Pradines, J., Dolnik, M. & Collins, J. J. Noise-based switches and amplifiers for gene expression. *Proc Natl Acad Sci U S A* **97**, 2075-80 (2000).
13. Paulsson, J., Berg, O. G. & Ehrenberg, M. Stochastic focusing: fluctuation-enhanced sensitivity of intracellular regulation. *Proc Natl Acad Sci U S A* **97**, 7148-53 (2000).
14. Dogterom, M. & Leibler, S. Physical aspects of the growth and regulation of microtubule structures. *Physical Review Letters* **70**, 1347-1350 (1993).
15. Berg, O. G., Paulsson, J. & Ehrenberg, M. Fluctuations and quality of control in biological cells: zero-order ultrasensitivity reinvestigated. *Biophys J* **79**, 1228-36 (2000).
16. Paulsson, J. & Ehrenberg, M. Noise in a minimal regulatory network: plasmid copy number control. *Q Rev Biophys* **34**, 1-59 (2001).
17. Vilar, J. M., Kueh, H. Y., Barkai, N. & Leibler, S. Mechanisms of noise-resistance in genetic oscillators. *Proc Natl Acad Sci U S A* **99**, 5988-92 (2002).
18. Elf, J., Paulsson, J., Berg, O. G. & Ehrenberg, M. Near-critical phenomena in intracellular metabolite pools. *Biophys J* **84**, 154-70 (2003).
19. van Kampen, N. *Stochastic processes in physics and chemistry, second edition* (Elsevier, Amsterdam, 1997).
20. Elowitz, M. B., Surette, M. G., Wolf, P. E., Stock, J. B. & Leibler, S. Protein mobility in the cytoplasm of Escherichia coli. *J Bacteriol* **181**, 197-203 (1999).
21. Gardiner, C. *Handbook of stochastic Methods* (Springer Verlag, Berlin, 1985).
22. Koch, C. *Biophysics of computation* (Oxford University Press, New York, 1999).
23. Nicolis, G. & Prigogine, I. *Proc Natl Acad Sci U S A* **68**, 2102 (1971).
24. Gibson, M. & Bruck, J. Efficient exact stochastic simulation of chemical systems with many species and channels. *Journal of physical chemistry A* **104**, 1876-1889 (2000).
25. Gillespie, D. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *journal of computational physics* **22**, 403-434 (1976).

26. Shapiro, L., McAdams, H. & Losick, R. Generating and exploiting polarity in bacteria. *Science* **298**, 1942-1946 (2002).
27. Arkin, A., Ross, J. & McAdams, H. H. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells. *Genetics* **149**, 1633-48 (1998).
28. Ferrell, J. E. J. Self-perpetuating states in signal transduction: positive feedback, double-negative feedback and bistability. *Current Opinion in Cell Biology* **14**, 140-148 (2002).

COMPLEMENTARY INFORMATION AFTER PRINT

It has recently come to our attention that the present work (Elf et al. SPIE, 2003) is not the first application of the Next Reaction Method (NRM) to reaction-diffusion problems, since the NRM had already been applied to such problems in the SmartCell project (1,2,3). This project was not quoted, due to a failure on our part to recognize the connection between the present work and the algorithm embedded in project work reports available to us at the time (2, 3).
There are, at the same time, considerable differences between the two approaches: While we use the NRM only to keep track of in which sub-volume the next event (reaction or diffusion) occurs, the SmartCell uses the NRM to order *all* events. We order the binary tree once or twice per event, whereas SmartCell uses a dependency-graph to keep track of which and how many events that are needed to be reordered. In SmartCell, the geometry of the system is described in the dependency-graph, whereas we use a separate geometry matrix. In addition, SmartCell, as described in (2, 3), has a large number of other features of no relevance for the comparison. The differences between the two approaches make them complementary, and we foresee that different types of problems will motivate different choices of algorithm.

COMPLEMENTARY REFERENCES

(1) Serrano et al, submitted manuscript
(2) Kaplan A, UPTEC X 01 044 ISSN 1401-2138, http://www.ibg.uu.se/upload/01044.pdf
(3) Andér M, UPTEC X 02 021 ISSN 1401-2138, http://www.ibg.uu.se/upload/Maria_A.pdf