Parallel Implementation of Stochastic Simulation for Large-scale Cellular Processes

Tianhai Tian and Kevin Burrage
Advanced Computational Modelling Centre
University of Queensland, Brisbane, Qld 4072, Australia
tian@maths.uq.edu.au, kb@maths.uq.edu.au

Abstract

Experimental and theoretical studies have shown the importance of stochastic processes in genetic regulatory networks and cellular processes. Cellular networks and genetic circuits often involve small numbers of key proteins such as transcriptional factors and signaling proteins. In recent years stochastic models have been used successfully for studying noise in biological pathways, and stochastic modelling of biological systems has become a very important research field in computational biology. One of the challenge problems in this field is the reduction of the huge computing time in stochastic simulations. Based on the system of the mitogen-activated protein kinase cascade that is activated by epidermal growth factor, this work give a parallel implementation by using OpenMP and parallelism across the simulation. Special attention is paid to the independence of the generated random numbers in parallel computing, that is a key criterion for the success of stochastic simulations. Numerical results indicate that parallel computers can be used as an efficient tool for simulating the dynamics of large-scale genetic regulatory networks and cellular processes.

1. Introduction

Stochastic modelling of biological systems has become a very important research field in computational biology in recent years. Experimental and theoretical studies have shown the importance of stochastic processes in genetic regulatory networks and cellular processes [7]. Cellular networks and genetic circuits often involve small numbers of key proteins such as transcriptional factors and signaling proteins. It is not appropriate to use deterministic models such as ordinary differential equations to describe the dynamics of the systems with small molecular numbers. Instead of studying the variation of concentrations in deterministic models based on the population of a large number of cells, stochastic models concentrate on the system dynamics in each cell by tracking the molecular number of each species in the system. The stochastic simulation algorithm (SSA) is an essentially exact procedure for studying noise in biochemical reaction systems [9]. The SSA numerically simulates the time evolution of a well-stirred chemically reacting system by taking proper account of the randomness inherent in such a system. This method takes time steps of variable length based on the rate constants and population size of each chemical species. The tremendous success of the SSA in recent years has been encouraging scientists to study more and more complicated biological systems. However, the bottleneck in the application of the SSA is the huge computing time because of the possibility of having very small stepsizes.

There are two major approaches for reducing the computational time of the SSA. The first approach is the tau-leap methods [10, 18]. Instead of considering only one reaction in a very small step in the SSA, a number of reactions are allowed to fire in a relative larger time interval. The binomial tau-leap method can restrict the possible reaction number in a time interval, and significant improvement in the efficiency has been reported by using this method [18, 5]. More work is needed to design general-purposed sampling techniques if reactant species undergo a number of reaction channels in order to simulate complicated biological systems. The second approach is through the use of multi-scale methods [4, 12, 15]. These methods partition a chemical reaction system into subsets of slow and fast reactions and then apply different simulation methods to each subset. The complicated partition processes erode part of the efficiency gain and more sophisticated partition processes are needed in order to improve the efficiency.

In this work we use parallel computer as an efficient tool to improve the efficiency of the SSA. Because a large number of independent simulations are needed in stochastic simulations in order to obtain statistic properties of the system, parallel computing can be used straightforwardly in stochastic simulation by letting each processor simulate a trajectory independently. However, the challenge is to guarantee the property of independence for the generated random numbers in different processes.

0-7695-2486-9/05 $20.00 © 2005 **IEEE**

Here we study the parallel implementation of the SSA by using OpenMP in order to keep the independence of the generated random numbers, and report the efficiency of the parallel implementation by simulating a system of the mitogen-activated protein (MAP) kinase cascade that is activated by epidermal growth factor (EGF).

2. Simulation Method

We first give a brief description of the SSA for biochemical reaction systems. It is assumed that we have a well-stirred mixture at constant temperature in a fixed volume Ω. This mixture consists of $N \geq 1$ molecular species $\{S_1, \quad , S_N\}$ that chemically interact through $M \geq 1$ reaction channels $\{R_1, \quad , R_M\}$. The restriction that Ω is fixed can be relaxed by using reaction rates that are dependent on the volume Ω, but we do not discuss it here.

The dynamical state of this system is denoted as $X(t) \equiv (X_1(t), \quad , X_N(t))^{\mathrm{T}}$, where $X_i(t)$ is the molecular number of S_i in the system at time t. For each j, $j = 1, \quad , M$, we define a propensity function $a_i(X)$ such that $a_i(X)dt$ is the probability that given $X(t) = X$, one reaction R_j will occur inside Ω in the next infinitesimal time interval $[t, t+dt)$. When that reaction occurs, $X(t)$ changes its state. The amount by which X_i changes is given by ν_{ij} that represents the change in the molecular number of S_i produced by one R_j reaction. The $N \times M$ matrix ν with elements ν_{ij} is called the stoichiometric matrix. In particular, if just the *j*th reaction occurs in the time interval $[t, t+\tau)$, the *j*th vector ν_j of the stoichiometric matrix is used to update the state of the system by $X(t+\tau) = X(t) + \nu_j$. We see that the propensity functions and state-change vectors completely characterize the chemical reaction system.

The SSA is an exact and direct representation of the evolution of $X(t)$. There are several forms of this algorithm. The direct method and the first reaction method are widely use and work in the following manner.

The direct method: With two independent samples r_1 and r_2 of the uniformly distributed random variable $U(0,1)$, the length of the time interval $[t, t+\tau)$ for the next reaction is determined by

$$\tau = \frac{1}{a_0(X)} \ln\left(\frac{1}{r_1}\right),$$

where $a_0(X(t))$ is the sum of all the propensity functions

$$a_0(X) = \sum_{k=1}^{M} a_k(X).$$

The determination of the specific reaction occurring in $[t, t+\tau)$ is given by the index j satisfying

$$\sum_{k=1}^{j-1} a_k(X) < r_2 a_0(X) \leq \sum_{k=1}^{j} a_k(X).$$

The update of the system is then given by

$$X(t+\tau) = X(t) + \nu_j.$$

The first reaction method: For each reaction channel R_j, generate a tentative reaction time by

$$\tau_j = \frac{1}{a_j(X)} \ln\left(\frac{1}{r_j}\right), \qquad j = 1, \quad , M,$$

where $r_1, \quad , r_M$ are M statistically independent samples of $U(0,1)$, and let

$\tau =$ the smallest of $\{\tau_1, \quad , \tau_M\}$,

$j =$ the index of the smallest of $\{\tau_1, \quad , \tau_M\}$.

The update of the system is then given by

$$X(t+\tau) = X(t) + \nu_j.$$

The direct method requires less computing time for calculating the reaction time and has been widely used in stochastic simulations on sequential computers. On the other hand, the first reaction method has a relatively simpler process for determining the index of the next reaction and thus is more appropriate to be used in parallel computing. In addition, the computation of the tentative reaction time of each reaction channel in the first reaction method can be implemented in parallel if the number of reactions is large.

In order to represent large-scale stochastic models in a simple and concise way, we use a general formula for representing propensity functions of different types of biochemical reactions. This general formula can significantly simplify the programming process and may lead to a general-purpose computer program for stochastic simulation. Here we are interested in biological systems modeled by three types of elementary reactions, namely the first order reaction, the second order reaction and the homodimer formation. Third and high order reactions are not studied as they can be reasonably estimated by the combination of second order reactions [13]. The propensity functions of these three types of elementary reactions are written in the following way:

(1) the first order reaction

$$X_i \xrightarrow{c_1} X_k \qquad a_1(X) = c_1 X_i X_{M+1},$$

where $X_{M+1} \equiv 1$;

(2) the second order reaction

$$X_i + X_j \xrightarrow{c_2} X_k \qquad a_2(X) = c_2 X_i X_j,$$

(3) the homodimer formation

$$X_i + X_i \xrightarrow{c_3} X_k \qquad a_3(X) = \frac{1}{2} c_3 X_i (X_i - 1).$$

The propensity functions of these three types of reactions can be written as [2]

$$a_j(X) = k_{j1} X_{j1} X_{j2} - k_{j2} X_{j1}, \qquad j = 1, \quad , M$$

that can be defined by a ($M \times 2$) rate matrix with elements k_{j1} and k_{j2} in the *j*-th row, and a ($M \times 2$) index matrix with elements $j1$ and $j2$ in the *j*-th row. For the first and second order reactions, $k_{j1} = c_i$ and $k_{j2} = 0$, while $k_{j1} = k_{j2} = c_3 / 2$ for the homodimer formation. Note that this representation is also designed for parallel simulations of biochemical reaction systems.

3. Large-scale cellular signaling processes

EGF receptor belongs to the tyrosine kinase family of receptors and is expressed in virtually all organs of mammals. The EGF receptor is activated by a variety of ligands that are crucial in the formation and propagation of many tumors through their effect on cell signaling pathway, cellular proliferation, control of apoptosis, and angio-genesis. The importance of the EGF receptor in tumorigenesis and tumor progression makes it an attractive target for the development of anticancer therapies. Over the past two decades, much effort has been directed at developing anticancer agents that can interfere with the EGF receptor activity. A variety of targeting strategies to exploit the role of the EGF receptor in tumors have been employed and clinical evaluations have yielded some promising results [14].

The EGF receptor is probably the best-known receptor system that has allowed the development of mathematical models [1, 16]. Although the principal hierarchy of the EGF receptor signal pathway and its activation sequence is well known, recent experimental discoveries provide more and more information for the protein-protein interactions and positive/negative regulatory loops in this signal pathway. The kinetic network of the EGF receptor pathway is much more complicated than previously thought. In addition, we still have a poor understanding of critical signaling events that control divergent cellular responses such as cell growth, survival or differentiation. Based on the biochemical properties of cellular components, sophisticated mathematical models can provide a tool to manage, interpret and understand the complexity of large-scale cellular signaling processes [8].

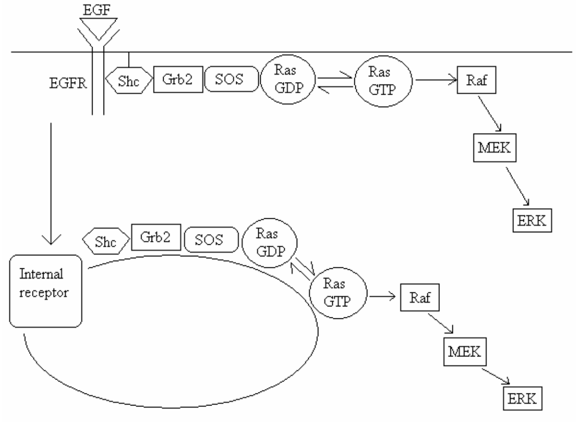

Figure 1 The MAP kinase cascade activated by surface and internalized EGF receptors.

This work studies the parallel simulation of a stochastic model that is derived from the mathematical model in the form of ordinary differential equations [16]. The network of the MAP kinase cascade that is activated by EGF can be divided into four modules (see Figure 1). The first module describes the reception of EGF and EGF receptor autophosphorylation that results from the dimerization of pairs of ligand-occupied receptors. The second module begins with the binding of a GTPase activating protein (GAP). Downstream of GAP binding is the formation of signaling complexes by the interaction of several signaling proteins such as Sos, Grb2 and Shc. There are two principal pathways in the second module, namely Shc-dependent and Shc-independent. Both principal pathways lead to the activation of Ras protein in the third module. The signaling complexes formed in the second module stimulate the conversion of RasGDP to RasGTP. Activated Ras in the form of RasGTP has a high affinity for the binding of Raf that leads to the activation of the MAP kinase cascade in the last module of this network. The MAP kinase cascade contains three types of proteins, namely Raf, MEK and ERK-1/2, and is a highly conserved module in cell signaling pathways. At each of 3 levels, a kinase must be phosphorylated at 2 sites before it can act as a catalyst for the next level. Further more, this network also includes the internalization processes, hence duplicating all the steps described above and increasing the complexity of the system. Unoccupied receptors, single ligand-receptor complexes and dimerized receptors are all subject to endocytosis but at different rates. Endocytosis may lead to receptors reforming at the cell surface. In either case, signalling downstream of the EGF receptor may continue as the internalized complex travels through the cytoplasm in a vesicle [11].

Schoeberl et al. [16] has developed a mathematical model for the signal transduction pathway of the EGF receptor. This model contains 94 compounds (variables) and is in the form of ordinary differential equations. Based on this deterministic model, Chatterjee et al. [5] has developed a corresponding stochastic model for testing the efficiency of the binomial tau-leap method. However, Gillespie has commented that this implementation of the binomial tau-leap method in [5] may generate bias because all the reactions are not treated equally in stochastic simulations. Indeed it is not easy to write a sophisticated computer program by using the tau-leap methods to simulate this complicated stochastic system. In addition, there is not any reported simulation result of the stochastic model because the published paper is an applications note [5]. Thus we use parallel computing to simulate this system in order to reduce the huge computing time of the SSA. Due to space limits, we do not list all of the 226 reactions here and readers can find the detailed information in Schoeberl et al. [16] for biochemical reactions, kinetic rates and initial conditions.

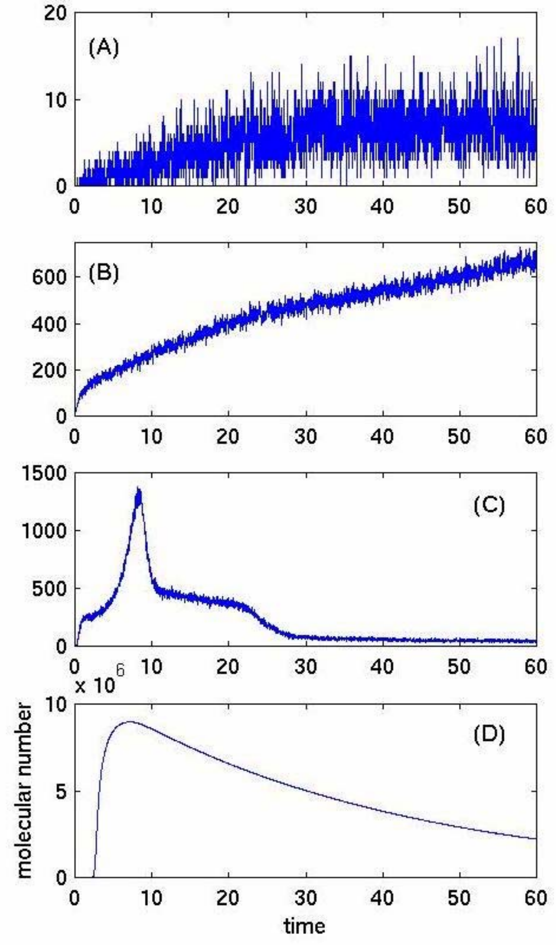

Figure 2 Simulations of four protein complexes in the MAP kinase cascade activated by the EGF receptor. (A) (EGF-EGFRi)2, internalized dimer of EGF-EGFR; (B) Shc*-Grb2, the complex of the activated Shc and Grb2; (C) Raf*, activated Raf; (D) ppERK, activated ERK.

Figure 2 gives a simulation of four protein complexes in the time interval [0, 60] minutes. This is the first reported simulation result for the stochastic properties of the MAP kinase cascade activated by the EGF receptor. The numbers of these protein complexes differ from very small numbers of the internalized dimer of the ligand-occupied receptors (EGF-EGFRi)2 to very large numbers of ppERR, the activated form of ERK that is dually phosphorylated. A very interesting result is that the signaling output of this network (ppERK) is not very noisy although the dynamics of quite a few protein complexes is very noisy. Although deterministic models in the form of differential equations are widely used in describing the kinetics of the cell signaling transduction pathways, stochastic models should be developed to study the importance of small molecular numbers in the system dynamics, especially when the concentrations (activities) of the signal input such as EGF are small (low). In this case stochastic properties of certain signal proteins may have profound impact on the study of inhibitors for the anticancer therapies.

4. Parallel implementation

There are four types of implementation techniques for stochastic simulations on parallel computers [3]. These include "parallelism across the method" and "parallelism across the simulation". The first type of parallelism involves domain decomposition, either at a functional level or at a data level. This can be programmed using, for example, OpenMP, which is an application programmer's interface for shared memory parallelism. OpenMP uses a single master thread, with a team of slaves (processors) executing code in parallel. Parallel implementation of the first reaction method can be considered in the computation of the propensity functions and in the system update, if the number of reactions in a system is very large. In addition we can parallelise the process for determining the reaction time of the next reaction.

The second approach is based on the fact that a large number of independent simulations (of the order of 1000 or more) needs to be performed in order to calculate statistics about the nature of the solution – such as the mean, variance and probability density function at a number of time points. It is a straightforward parallelization technique to run a number of different simulations on separate processes (for example, on a multi-processor supercomputer), as these simulations are independent of each other. This approach can be implemented in the OpenMP or MPI/PVM environments. In either case, communication between the processors is required only at the end of the simulations in order to generate statistics. Such an implementation is referred to as coarse-grain parallelism (in contrast to fine-grained parallelism where only small amount of computation can be carried out between processor synchronization).

A challenge problem in the parallel implementation of stochastic simulations is the quality of the random number generator. Although Monte-Carlo computations are considered easy to parallelize, simulation results can be adversely affected by defects in the parallel pseudorandom number generator used [17]. The independence of the generated random numbers and the subsequent independence of simulations on different processes are the primary requirement for the success of stochastic simulations. However, finding a good parallel random number generator has proven to be a very challenge problem, and is still the subject of much research and debate [6]. Based on recent empirical tests for a number of parallel random number generators, it was still suggested to use a number of different generators to run the application in order to increase our confidence on simulation results [17]. In our previous attempts, we tried to use different random seeds in different processors in the MPI environment, and obtained very good speed-up and efficiency that is close to 1 [2]. However, theoretical analysis and empirical tests are needed to study the selection process of random seeds in order that different simulations generated from different processors are independent from each other.

In order to avoid the difficult issue of the independence properties of random number generators, we consider another parallel implementation of the SSA by using OpenMP. Although we still consider parallelism across the simulation, the current technique considers the parallel implementation at each step rather than in the whole simulation that we have studied before. We guarantee independence by using the master thread at each step to generate random numbers required in this step for all simulations. The major structure of this implementation of the SSA is given in FORTRAN as follows.

```
DO WHILE (min_t .LT. L)
    Generate random numbers
    !$OMP PARALLEL DO
    DO I = 1, Num_simu
        One step of the SSA
        (or a number of steps of the SSA)
    END DO
    !$OMP END PARALLEL DO
    min_t  = min{t(1), …, t(Num_simu)}
END DO
```

At each DO WHILE (min_t .LT. L) loop, we first generate a matrix of random numbers with dimension

(Num_simu) × (Num_perstep) × (Num_steps),

where Num_simu is the number of stochastic simulations, Num_perstep is the number of random samples required at each step of the SSA, and Num_steps is the number of steps of the SSA implemented in each PARALLEL DO loop. Then this matrix is defined as a shared variable in the following PARALLEL DO loop. As two random numbers are required in the direct method, namely Num_perstep = 2, we can generate two matrices of random numbers with dimension (Num_simu) × (Num_steps).

The advantage of this implementation is that all random numbers are generated from the master thread. Thus we can ensure the independent properties of different stochastic simulations. However, the cost is the communication time because a number of shared variables should be used at each step. We can consider a number of techniques for improving the efficiency of this parallel implementation. For example, we can use detailed formulas for the propensity functions in order to avoid communication of the index matrices for variables and reaction rates, although this is not recommended in a general-purposed computer program, or run a number of steps of the SSA in each OpenMP PARALLEL DO loop. It is expected the efficiency of the parallel computation can be improved by the employment of these techniques.

Numerical results in this paper are obtained from a parallel implementation carried out on an SGI Altix 3700 scalable-shared memory parallel computer at the University of Queensland. The command in Fortran 90 ETIME is used to measure the program's elapsed time. The timings were calculated from 5 runs, discarding the slowest and fastest and then averaging the remaining times. Based on 1000 simulations (Num_simu=1000) in the time interval [0, 5] minutes, Figure 3 gives the speed-up and efficiency of the parallel implementation of the SSA. We can achieve significant improvement on the efficiency if each PARALLEL DO loop contains 100 steps of the SSA over one step of the SSA. In addition, the parallel simulation time is more stable if a number of steps of the SSA are included in one PARALLEL DO loop than that if only one step of the SSA is considered.

5. Conclusions

In this work we have focused on the application of parallel computing to the system of the mitogen-activated protein kinase cascade that is activated by epidermal growth factor. This is currently a very important topic in computational biology – namely the use of stochastic chemistry for the understanding of genetic regulatory networks and large-scale cellular processes. Based on this system with 96 species and 224 reactions, we studied the parallel implementation of the SSA by using OpenMP through parallelism across the simulation. Special attention has been paid to the dependence of the generated random numbers. Numerical results indicate that parallel computers can be used as an efficient tool for improving the efficiency of stochastic simulations. Recent research in stochastic modelling of biological networks has provided larger and larger systems. Future work includes the development of parallel algorithms through the parallelism across the method for simulating large-scale genetic regulatory networks and cellular processes.

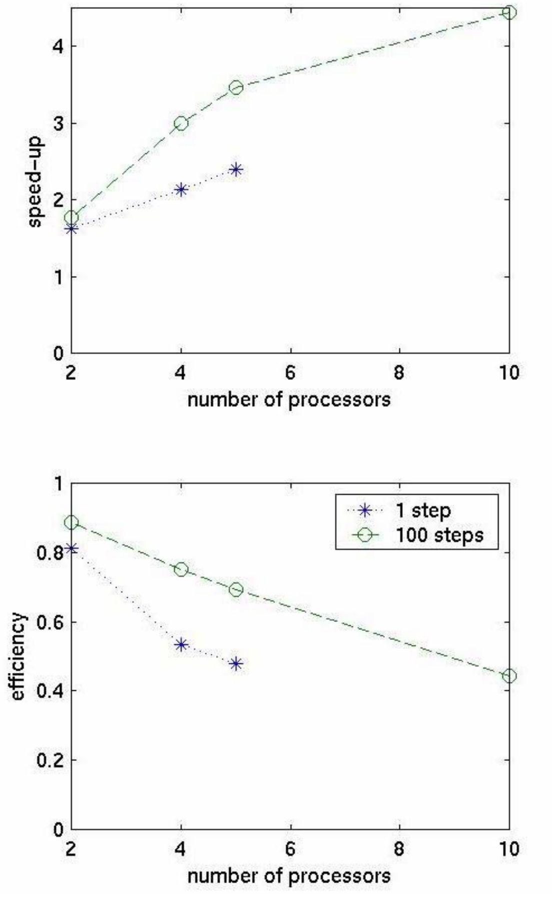

Figure 3 Speed-up and efficiency of parallel computing. (dot-line: one step of the SSA in each parallel do loop; dash-line: 100 steps of the SSA in each parallel do loop).

References

[1] Bhalla, U. S. and Iyengar, R. (1999) Emergent properties of networks of biological signaling pathways, *Science* **283**, 381-387.

[2] Burrage, K., Burrage, P. M., Hamilton, N. and Tian, T. (2005) Computer-intensive simulations for cellular models, to appear in *Parallel Computing for Bioinformatics and Computational Biology*, Wiley.

[3] Burrage, K., Burrage, P. M. and Tian (2001) Numerical methods for solving stochastic differential equations on parallel computers, *Proceedings of the 5th International Conference of HPC-Asia*.

[4] Burrage, K., Tian, T. and Burrage, P. M. (2004) A multi-scaled approach for simulating chemical reaction systems, *Prog. Biophys. Mol. Bio.* **85**, 217-234.

[5] Chatterje, A., Mayawala, K., Edwards, J. S. and Vlachos, D. G. (2005) Time accelerated Monte Carlo simulations of biological networks using the binomial tau-leap method, *Bioinformatics* **21**, 2136-2137.

[6] Coddington P. D. (1996) Random Number Generators for parallel computers, *NHSE Review*, No.2, download from http://nhse.cs.rice.edu/NHSEreview/RNG/.

[7] Elowitz, M. B., Levine, A. J., Siggia, E. D. and Swain, P. S. (2000) Stochastic gene expression in a single cell, *Science* **297**, 1183-1186.

[8] Endy, D. and Brent, R. (2001) Modelling cellular behaviour, *Nature* **409**, 391-395.

[9] Gillespie, D. T. (1977) Exact stochastic simulation of coupled chemical reactions, *J. Phys. Chem.* **81**, 2340--2361.

[10] Gillespie, D. T. (2001) Approximate accelerated stochastic simulation of chemical reaction systems, *J. Chem. Phys.* **115**, 1716--1733.

[11] Haugh, J. M. (2002) Localization of receptor-mediated signal transduction pathways: the inside story, *Molecular Interventions* **2**, 292-307.

[12] Haseltine E. L. and Rawlings, J. B. (2002) Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics, *J. Chem. Phys.* **117**, 6959-6969.

[13] Kierzek, A. M. (2002) STOCKS: stochastic kinetic simulations of biochemical systems with Gillespie algorithm, *Bioinformatics* **18**, 470--481.

[14] Lockhart, C. and Berlin, J. D. (2005) The epidermal growth factor receptor as a target to colorectal cancer therapy, *Seminars in Oncology* **32**, 52-60.

[15] Puchalka J. and Kierzek, A.M. (2004) Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks, *Biophy. J.* **86**, 1357-1372.

[16] Schoeberl, B., Eichler-Jonsson, C., Gilles, E. D. and Muller, G. (2002) Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors, *Nature Biotechnology* **20**, 370-375.

[17] Srinivasan, A., Mascagni, M. and Ceperley, D. (2003) Testing parallel random number generators, *Parallel Computing* **29**, 69-94.

[18] Tian T. and Burrage, K. (2004) Binomial leap methods for simulating stochastic chemical kinetics, *J. Chem. Phys.* **121**, 10356-10364.