Effective simulation techniques for Biological Systems

Kevin Burrage[a] and Tianhai Tian[a]

[a]Advanced Computational Modelling Centre, The University of Queensland
Brisbane, Queensland 4072, Australia

ABSTRACT

In this paper we give an overview of some very recent work on the stochastic simulation of systems involving chemical reactions. In many biological systems (such as genetic regulation and cellular dynamics) there is a mix between small numbers of key regulatory proteins, and medium and large numbers of molecules. In addition, it is important to be able to follow the trajectories of individual molecules by taking proper account of the randomness inherent in such a system. We describe different types of simulation techniques (including the stochastic simulation algorithm, Poisson Runge-Kutta methods and the Balanced Euler method) for treating simulations in the three different reaction regimes: slow, medium and fast. We then review some recent techniques on the treatment of coupled slow and fast reactions for stochastic chemical kinetics and discuss how novel computing implementations can enhance the performance of these simulations.

Keywords: Stochastic simulation methods, chemical reaction systems, multi-scaled approaches, parallel computing, biological applications

1. INTRODUCTION

There is now considerable evidence from both theoretical and experimental perspectives of the role of noise in genetic regulation. Federoff and Fontana[13] remark that "stochasticity is evident in all biological processes. The proliferation of both noise and noise reduction is a hallmark of organismal evolution." However, a natural question to ask is what is the nature of this stochasticity? Hume[22] notes that "transcription in higher eukaryotes occurs with a relatively low frequency in biologic time and is regulated in a probabilistic manner." The comment about "low frequency" is significant here and we will return to this later.

Gene expression within a cell is a complex process involving such factors as chromatin remodelling, transcription, the export of RNA and the translation of mRNA into proteins. Physiological activity and cell differentiaton within a mammalian cell is controlled by perhaps more than 10000 protein coding genes and thousands of genes are expressed at very low copy numbers. This means that new gene profiling techniques such as microarrays may not be able to reliably detect these numbers. Thus there is a great need for good models and effective simulations to guide the experimentalist and to provide additional insights into the nature of genetic regulation.

Furthermore, Sano et al.[32] remark that "initiation of gene transcription is a discrete process in which individual protein-coding genes in an off state can be stochastically switched on, resulting in sporadic pulses of mRNA production." This is the dichotomy that we must resolve - proteins are discrete objects, yet their effects are often modelled (as ordinary differential equations) in terms of concentrations.

We can, therefore, consider three different types of modelling regimes for understanding genetic regulation. These include the discrete and stochastic, the continuous and stochastic and the continuous and deterministic. An additional complexity arises when we consider both temporal and spatial effects.

Essentially the characterisations of these regimes depend on the nature of the reactions and the number of molecules in the system being studied. In this paper we will review various simulation techniques that are relevant to each of these regimes. In particular, we will also consider what happens for mixed systems with small numbers of key regulatory proteins and a mix of medium and large numbers of other types of molecules. The basis of this

Further author information: (Send correspondence to Kevin Burrage)
Kevin Burrage: E-mail: kb@maths.uq.edu.au, Telephone: 61-7-33653487
Tianhai Tian: E-mail: tian@maths.uq.edu.au, Telephone: 61-7-33656137

Fluctuations and Noise in Biological, Biophysical, and Biomedical Systems II, edited by
Derek Abbott, Sergey M. Bezrukov, Andras Der, Angel Sánchez, Proc. of SPIE Vol. 5467
(SPIE, Bellingham, WA, 2004) · 0277-786X/04/$15 · doi: 10.1117/12.548672

paper is the stochastic simulation approach to biochemical reactions which was developed by Gillespie[16] through the stochastic simulation algorithm (SSA). This is an essentially exact procedure for numerically simulating the time evolution of a well-stirred chemically reacting system by taking proper account of the randomness inherent in such a system.

Thus this paper is organised in the following manner. In section 2 we give a brief overview of the SSA approach and discuss some new discrete simulation techniques that have been developed to overcome the inherent limitations of the SSA. In section 3 we consider different ways of treating the other regimes (stochastic continuous and deterministic), while in section 4 we discuss some new multi-scaled techniques for simulating mixed systems such as stochastic partitioning. In section 5 we will investigate some novel implementations of these models based on grid computing and parallel computing while in section 6 we give some numerical results in a parallel computing environment and the paper will conclude with some general remarks and discussion for future work.

2. DISCRETE SIMULATION METHODS FOR CHEMICAL REACTION SYSTEMS

The basis of our discussion in this paper is the stochastic simulation approach to biochemical reactions that was developed by Gillespie[16] through the stochastic simulation algorithm (SSA). This is an essentially exact procedure for numerically simulating the time evolution of a well-stirred chemically reacting system by taking proper account of the randomness inherent in such a system. It is rigorously based on the same microphysical premise that underlies the chemical master equation (Gillespie[18]) and gives a more realistic representation of a system's evolution than the deterministic reaction rate equation (RRE). In particular, the RRE is entirely inappropriate if the molecular population of some critical reactant species is so small that microscopic fluctuations can produce macroscopic effects. This is especially true for the genetic/enzymatic reactions in living cells. As with the Chemical Master Equation, the SSA converges, in the limit of large numbers of reactants, to the same solution as the Law of Mass Action.

Despite continued refinements to the numerical methods used in the SSA, it remains a computationally demanding approach limiting its applicability, especially for large reaction networks required for modelling most realistic gene networks. The algorithm takes time steps of variable length, based on the rate constants and population size of each chemical species. The probability of one reaction occurring relative to another is obtained by multiplying the rate constant of each reaction with the numbers of its substrate molecules. According to the correct probability distribution derived from the statistical thermodynamics theory, a random variable is then used to choose which reaction will occur, and another random variable determines how long the step will last. The chemical populations are altered according to the stoichiometry of the reaction and the process is repeated. The cost of this detailed stochastic simulation algorithm is the large amount of computing time. The key issue is that the time step for the next reaction can be very small indeed if we are to guarantee that only one reaction can take place in that time interval.

In recent years, the SSA has been successfully applied for simulating genetic/enzymatic reactions in which the molecular population of some critical reactant species is relatively small, for example, lambda phage[2] and circadian rhythms[11].[20] It has also been applied to much larger systems than originally designed for. For example, Arkin et al.[2] used the SSA to simulate a model of lambda phage containing 75 equations in 57 chemical species.

An alternative approach to the SSA is via the StochSim package developed initially by Carl Firth[28] as part of a study of bacterial chemotaxis. The aim was to develop a realistic way of representing the stochastic features of this signalling pathway and to handle the large numbers of individual reactions encountered.[14] Molecules or molecular complexes are represented as individual software objects. Reactions between molecules occur stochastically, according to probabilities derived from known rate constants.

StochSim works by quantising time into a series of discrete, independent time intervals, the sizes of which are determined by the most rapid reaction in the system. In each time interval, a molecule is selected at random and then another object (either a molecule or a pseudo-molecule) is again selected at random. If two molecules are selected, any reaction that occurs is bimolecular, whereas if one molecule and a pseudo-molecule are selected, it is unimolecular. Another random number is then generated to determine if a reaction will occur. The probability of a reaction is retrieved from a look-up table and if this exceeds the random number, the particles do not react.

On the other hand, if the probability is less than the random number, the particles react, and the system is updated.

StochSim is likely to be slower than the Gillespie algorithm in calculating the eventual outcome of a small set of simple biochemical reactions, especially when the number of molecules is large. However, if the system contains molecules that can exist in multiple states, then StochSim may not only be faster but also closer to physical reality.

One of the great challenges in the efficient simulations of chemical kinetic systems is how we deal with mixed systems in which some key species have low abundances (as is the case of some molecules in genetic regulation) while other molecules have large abundances and can be modelled via continuous SDEs. Thus a vital question to address is how we can link discrete and continuous models and simulation algorithms in a sensible and efficient manner when treating chemical kinetic systems? A number of authors have addressed this issue recently including Rao and Arkin,[29] Haseltine and Rawlings[21] and Burrage et al.[8] based on partitioning of the system. These issues become even more important when spatial effects are considered.

More recently, Schnell and Turner[33] and Berry[3] have addressed the issue of some of these stochastic approaches failing to incorporate non-homogeneities typical of in-vivo conditions into models. Berry[3] considers Monte-Carlo simulations on a two dimensional square lattice. Each molecule is mobile on the lattice through random walks and chemical interactions. The fundamental difference between this approach and SSA is that the SSA uses the spatial homogeneity to derive a probability distribution for the time between elementary reactions and then samples randomly from this distribution to simulate the dynamics of the reaction. The Monte Carlo approach assumes only that the molecular motion is Brownian. Schnell and Turner[33] have used this approach in order to try to understand how the conventional chemical kinetic equations based on rate constants fail to describe the reactions in vivo conditions.

When minimal obstructions to diffusion are present, the rate constant approach is reasonable but in the presence of significant obstructions to diffusion, simulations and experiments show that $log(k)$ decays linearly on a logarithmic time scale and so k is time-dependent. Schnell and Turner propose replacing reaction constants with time dependent reaction coefficients of the form

$$k(t) = k_1 t^{-h}, \quad h \in (0,1], \quad t \geq 1,$$

where h is a constant corresponding to inhomogeneous and/or dimensionality restricted environments. See also Kopelman[25] and Tian and Burrage.[37]

We now give a brief review of the stochastic simulation algorithm for chemical reaction systems. We will, for the meantime, assume that we have a well-stirred mixture at constant temperature in a fixed volume Ω. This mixture consists of $N \geq 1$ molecular species $\{S_1, \ldots, S_N\}$ that chemically interact through $M \geq 1$ reaction channels $\{R_1, \ldots, R_M\}$. The restriction that Ω is fixed can be relaxed but we will not do that here.

The dynamical state of this system is denoted as $X(t) \equiv (X_1(t), \ldots, X_N(t))^\top$, where $X_i(t)$ is the number of S_i molecules in the system at time t. The initial state is given by $X(t_0) = X_0$. For each j, $j = 1, \cdots, M$, we will define the propensity function $a_j(X)$ such that $a_j(X(t))dt$ 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_{ji}, which represents the change in the number of S_i molecules produced by one R_j reaction. The $N \times M$ matrix ν with elements ν_{ji} is called the stoichiometric matrix. In particular, if just the jth reaction occurs in the time interval $[t, t + dt)$, the jth vector ν_j of the stoichiometric matrix is used to update the state of the system by

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

We see that the propensity functions and state-change vectors completely characterize the chemical reaction system.

In the discrete and stochastic case the $X_i(t)$ represent the number of S_i molecules at time t and thus $X(t)$ takes on integer values in a non-negative integer lattice of dimension N. In fact $X(t)$ is a discrete (jump) Markov process. As such it has a time evolution equation associated with it which describes the probability $P(x, t|x_0, t_0)$

that $X(t) = x$ given $X(t_0) = x_0$. This equation is called the Chemical Master Equation (CME) and it can be written as

$$\frac{\partial}{\partial t}P(x,t|x_0,t_0) = \sum_{j=1}^{M} \left(a_j(x-\nu_j)P(x-\nu_j,t|x_0,t_0) - a_j(x)P(x,t|x_0,t_0)\right).$$

In general this discrete parabolic partial differential equation is too difficult to solve (either analytically or numerically) and other techniques are needed to simulate the $X(t)$.

As a particular example we consider a simple chemical reacting system with 3 molecular species and 4 reaction channels, namely

$$\begin{array}{rcl} S_1 & \overset{k_1}{\to} & 0 \\ S_1 + S_1 & \overset{k_2}{\to} & S_2 \\ S_2 & \overset{k_3}{\to} & S_1 + S_1 \\ S_2 & \overset{k_4}{\to} & S_3. \end{array}$$

This system contains a reversible dimerization of the monomer S_1 into an unstable S_2, which can convert to a stable form S_3 by reaction R_4. In this case the propensity functions are given by

$$\begin{array}{rcl} a_1 & = & k_1 S_1 \\ a_2 & = & k_2 S_1^2/2 \\ a_3 & = & k_3 S_2 \\ a_4 & = & k_4 S_2, \end{array}$$

and the stoichiometric matrix ν is given by

$$\begin{bmatrix} -1 & -2 & 2 & 0 \\ 0 & 1 & -1 & -1 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

A method for simulating such systems is the so-called Stochastic Simulation Algorithm (SSA) of Gillespie,[16] which is an exact and direct representation of the evolution of $X(t)$. There are several forms of this algorithm. The direct method works in the following manner.

Method 1 (The direct method). With two independent samples r_1 and r_2 of the uniformly distributed random variable $\mathbf{U}(0,1)$, the length of the time interval $[t, t+dt)$ is determined by

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

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 occuring in $[t, t+dt)$ is given by the index j satisfying

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

The update of the system is then given by

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

The point about the SSA is that the time step τ is taken small enough to guarantee that only one reaction occurs in that time interval. Clearly the SSA can be very computationally inefficient especially when there are large numbers of molecules or the propensity functions are large.

Recently, considerable attention has been paid to reducing the computational time of simulation algorithms for stochastic chemical kinetics. Gibson and Bruck[15] refined the first reaction SSA of Gillespie by reducing the number of random variables that need to be simulated. This can be effective for systems in which some reactions occur much more frequently than others. A different approach is adopted by Rao and Arkin[29] who simulate systems that have been simplified by quasi-steady state assumptions. Resat et al.[31] treat systems which have widely varying rate constants by applying a weighted Monte Carlo approach.

Gillespie[19] proposed two new methods, namely the τ-leap method and the midpoint τ-leap method in order to improve the efficiency of the SSA while maintaining acceptable losses in accuracy. The key idea here is to take a larger time step and allow for more reactions to take place in that step, but under the proviso that the propensity functions do not change too much in that interval. Thus in the time interval $[t, t+\tau)$ and with the present state $X(t)$ at time t, then the number of times that the reaction channel R_j will fire is a Poisson random variable

$$K_j(\tau; X, t) = P(a_j(X), \tau), \quad j = 1, \ldots, M.$$

Here the notation $P(\lambda, t)$ denotes a stochastic Poisson process with mean λt and variance λt and where

$$\Pr(P(\lambda, t) = k) = \frac{e^{-\lambda t}(\lambda t)^k}{k!}.$$

These considerations lead to the τ-leap method.

Method 2 (The τ-leap method). Choose a value for τ that satisfies the Leap Condition: i.e., a temporal leap by τ will result in a state change λ such that for every reaction channel R_j, $|a_j(X+\lambda) - a_j(X)|$ is "effectively infinitesimal." Generate for each $j = 1, \ldots, M$ a sample value k_j of the Poisson random variable $P(a_j(X), \tau)$, and compute $\lambda = \sum_{j=1}^{M} k_j \nu_j$. Finally, perform the updates by replacing t by $t + \tau$ and X by $X + \lambda$.

Since the τ-leap method uses the initial state X to approximate the states in the time interval $[t, t+\tau)$, its efficiency can be improved by computing a better approximation to the states in the given time interval - for example, by an estimation at the midpoint $t + \tau/2$. This leads to the midpoint τ-leap method.

Method 3 (The midpoint τ-leap method). For the selected leaping time τ (which satisfies the Leap Condition), compute the expected state change $\overline{\lambda} = \frac{\tau}{2} \sum_{j=1}^{M} a_j(X)\nu_j$ during the time period $[t, t+\frac{\tau}{2})$. Then use the estimated state $X' \equiv X + [\overline{\lambda}]$ to generate for each $j = 1, \ldots, M$ a sample value k_j of the Poisson random variable $P(a_j(X'), \tau)$. Compute the actual state change, $\lambda = \sum_{j=1}^{M} k_j \nu_j$, and perform the updates by replacing t by $t + \tau$ and X by $X + \lambda$. Here $[\,]$ denotes the integer part.

Burrage and Tian[7] introduced the framework of Poission Runge-Kutta (PRK) methods for simulating chemical reaction systems. These PRK methods are related to the class of stochastic Runge-Kutta (SRK) methods for solving stochastic differential equations driven by Wiener noise.

3. SIMULATION METHODS FOR DIFFERENT REGIMES

Now if a chemical reaction system possesses a macroscopically infinitesimal time scale so that during any dt all of the reaction channels can fire many times, yet none of the propensity functions change appreciably, then the jump Markov process can be approximated by a continuous Markov process. This Markov process is described by the Chemical Langevin Equation (CLE), which is a stochastic ordinary differential equation (SDE) - see Gillespie.[17] It takes the Itô form

$$dX = \sum_{j=1}^{M} \nu_j a_j(X) dt + \sum_{j=1}^{M} \nu_j \sqrt{a_j(X)} dW_j(t), \qquad (1)$$

where the $W_j(t)$ are independent Wiener processes.

The CLE represents processes in the intermediate regime, that is those processes that are stochastic and continuous. A Wiener process is a stochastic process satisfying

$$E(W(t)) = 0, \quad E(W(t)W(s)) = \min\{t, s\}.$$

It is known that the Wiener increments are independent Gaussian processes with mean 0 and variance $|t-s|$ (that is, $N(0, |t-s|)$). Thus the Wiener increment $\Delta W(t) \equiv W(t+\Delta t) - W(t)$ is a Gaussian random variable $N(0, \Delta t) = \sqrt{\Delta t}N(0,1)$.

The Chemical Langevin Equation is an example of the more general class of Itô Stochastic Differential Equations given by

$$dy(t) = g_0(y(t))dt + \sum_{j=1}^{d} g_j(y(t))\, dW_j(t), \qquad y(t_0) = y_0, \quad y \in \mathbb{R}^m. \tag{2}$$

Thus general classes of methods that can be used to solve (2) can also be used to simulate solutions of (1), (see Kloeden and Platen,[24] for example).

In the case that the deterministic component dominates the noise terms then this leads to the standard chemical kinetic approach that is described by the reaction rate equations

$$X'(t) = \sum_{j=1}^{M} \nu_j a_j(X(t)). \tag{3}$$

Equation (3) represents the third regime for modeling chemical reaction systems and there are standard techniques for computing numerical approximations to this ODE system.

Just as there is a natural relationship between the modeling of the discrete, continuous stochastic and deterministic regimes so there is a relationship between the simulation techniques.

A Poisson random variable $P(a_j(X), \tau)$ with a large mean $a_j(X)\tau$ can be approximated by a Gaussian random variable $N(a_j(X)\tau, a_j(X)\tau)$, since

$$P(a_j(X), \tau) \approx N(a_j(X)\tau, a_j(X)\tau) = a_j(X)\tau + \sqrt{a_j(X)\tau}N(0,1),$$

where $N(\mu, \sigma^2)$ is a Gaussian random variable with mean μ and variance σ^2. This can be viewed as

$$P(a_j(X), \tau) \approx a_j(X)\tau + \sqrt{a_j(X)}\Delta W(t). \tag{4}$$

Now the simplest numerical method for solving (2) is the Euler-Maruyama method. It takes the form

$$y_{n+1} = y_n + hg_0(y_n) + \sum_{j=1}^{d} \Delta W_j^{(n)} g_j(y_n), \quad t_{n+1} = t_n + h,$$

where $\Delta W_j^{(n)} \equiv W_j(t_n + h) - W_j(t_n)$ is a Gaussian random variable $N(0, h)$.

The Euler-Maruyama method converges with strong order 0.5 and weak order 1 to the Itô form of the SDE. If it is applied to (1) it takes the form

$$X_{n+1} = X_n + \tau \sum_{j=1}^{M} \nu_j a_j(X_n) + \sum_{j=1}^{M} \Delta W_j^{(n)} \nu_j \sqrt{a_j(X_n)}.$$

Now using the approximation in (4) we can write this as

$$X_{n+1} = X_n + \sum_{j=1}^{M} \nu_j P_j(a_j(X_n), \tau).$$

This method is nothing but the $\tau-$leap method of Gillespie. Thus the $\tau-$leap method is the Euler-Maruyama method applied in the discrete setting when there are small numbers of molecules.

This has led Burrage and Tian[7] to consider a general class of explicit Poisson Runge-Kutta (PRK) methods in which s intermediate approximations are simulated within a given step. This class of method takes the form

$$\begin{aligned} Y_i &= X_n + \sum_{k=1}^{M} \nu_k P_k(\sum_{j=1}^{s} W_{ij} a_k(Y_j), \tau), \quad i = 1, \cdots, s \\ X_{n+1} &= X_n + \sum_{k=1}^{M} \nu_k P_k(\sum_{j=1}^{s} \beta_j a_k(Y_j), \tau). \end{aligned}$$

In general it is sufficient to consider simulation methods in which s is 1 or 2, and this gives rise to a general class of two stage methods of the form

$$\begin{aligned} Y &= X_n + \sum_{k=1}^{M} \nu_k P_k(\theta a_k(X_n), \tau) \\ X_{n+1} &= X_n + \sum_{k=1}^{M} \nu_k P_k((1-\beta) a_k(X_n) + \beta a_k(Y), \tau). \end{aligned}$$

This method can be viewed as the application of a two stage Runge-Kutta method to (3) and takes the form

$$\begin{aligned} Y &= X_n + \tau\theta \sum_{k=1}^{M} \nu_k a_k(X_n) \\ X_{n+1} &= X_n + \tau \sum_{k=1}^{M} \nu_k ((1-\beta) a_k(X_n) + \beta a_k(Y)) \end{aligned}$$

and can be characterized in tableau form by

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \theta & 0 & 0 \\ \hline & 1-\beta & \beta \end{array}. \qquad (5)$$

Runge-Kutta methods represent a very important class of methods for solving Ordinary Differential Equations (see Butcher[10]). Note that if $\beta = \frac{1}{2\theta}$, (5) is of order two when applied to Ordinary Differential Equations of initial value type.

Burrage and Tian[7] consider two new stochastic simulation methods methods with $\beta = \frac{1}{2\theta}$: the Heun PRK method ($\theta = 1$) and the R2 PRK method ($\theta = \frac{2}{3}$). The latter is so-called because it is directly related to the R2 method for solving Stratonovich SDEs (see P.M.Burrage[9]).

An important issue here is that of stiffness. For ODE systems a problem is stiff if it has widely varying eigenvalues. In this case explicit methods cannot be used and implicit methods such as the implicit Euler or the trapezoidal method have to be used. When applied to (3) these, respectively, take the form

$$\begin{aligned} X_{n+1} &= X_n + \tau \sum_{k=1}^{M} \nu_k a_k(X_{n+1}) \\ X_{n+1} &= X_n + \tau/2 \sum_{k=1}^{M} \nu_k (a_k(X_n) + a_k(X_{n+1})). \end{aligned}$$

Rathinam et al.[30] consider how stiffness manifests itself at both the continuous deterministic and discrete stochastic levels. In this case explicit methods become impractical. The authors construct two implicit versions of the explicit τ-leap method known as the rounded and unrounded implicit τ-leap method that have better stability properties than the explicit τ-leap method and are suitable for solving stiff chemical systems. The unrounded method has the form

$$X_{n+1} = X_n + \tau \sum_{j=1}^{M} \nu_j \left(a_j(X_{n+1}) - a_j(X_n)\right) + \sum_{j=1}^{M} \nu_j P_j \left(a_j(X_n), \tau\right)$$

but suffers from the drawback that $X_{n+1} - X_n$ is typically not an integer vector. Rathinam et al. overcome this difficulty by a two-stage process which is similar to a prediction-correction process given by

$$\begin{aligned} X &= X_n + \tau \sum_{j=1}^{M} \nu_j \left(a_j(X) - a_j(X_n)\right) + \sum_{j=1}^{M} \nu_j P_j \left(a_j(X_n), \tau\right) \\ X_{n+1} &= X_n + \sum_{j=1}^{M} \nu_j \left[\tau \left(a_j(X) - a_j(X_n)\right)\right] + \sum_{j=1}^{M} \nu_j P_j \left(a_j(X_n), \tau\right) \end{aligned}$$

where again [ ] denotes the nearest nonnegative integer.

Now an SDE of the form (2) is said to be stiff if it has widely varying Lyapunov exponents (these are the stochastic counterparts of eigenvalues). In this case there are three possible simulation approaches: explicit, semi-implicit and fully-implicit methods. In the first case, explicit methods can be suitable for stiff problems only if the stepsize is not too small or if the additional computation associated with implicit methods is prohibitive. Perhaps the simplest method in the second class is the semi-implicit Euler method which takes the form

$$y_{n+1} = y_n + hg_0(y_{n+1}) + \sum_{j=1}^{d} \Delta W_j^{(n)} g_j(y_n).$$

This method works well if[27] is stiff only in the deterministic component but less well if there is also stiffness in the stochastic components. Milstein et al.[27] introduced the Balanced Euler method to overcome this limitation; it takes the form

$$y_{n+1} = y_n + (I + C_n)^{-1} \left(hg_0(y_n) + \sum_{j=1}^{d} \Delta W_j^{(n)} g_j(y_n) \right).$$

The matrix C_n is chosen to be of the form

$$c_0(y_n)h + \sum_{j=1}^{d} c_j(y_n) \left| \Delta W_j^{(n)} \right|,$$

where the $c_j(y_n)$ are matrix functions chosen to give appropriate damping and guarantee existence of solutions. Note that the fully-implicit Euler method

$$y_{n+1} = y_n + hg_0(y_{n+1}) + \sum_{j=1}^{d} \Delta W_j^{(n)} g_j(y_{n+1})$$

cannot guarantee convergence at any particular time step since the Wiener increments can take on positive or negative values with equal probability and in any case does not converge to the Itô solution if convergence does take place - see Burrage and Tian,[6] for example. Alcock and Burrage[1] have considered improvements over the Balanced Euler method in terms of better order and stability properties while Tian and Burrage[36] have constructed high order implicit Taylor methods for stiff SDEs. Both the semi-implicit Euler method and the Balanced Euler method have strong order 0.5 and weak order 1.

What we see from the above discussion is an attempt to construct families of simulation methods that can move between the discrete, continuous stochastic and deterministic regimes in a natural manner. This is very important when dealing with mixed chemical systems, such systems can be viewed as consisting of three different regimes and can be solved by coupling together three different simulation approaches applied to each of these regimes. For example, for mixed systems Burrage et al.[8] use the SSA when there are only a very few molecules; the explicit PRK approach (as typified by the τ-leap method) is used for components of the system with moderate numbers of molecules and a simple SDE method for solving the CLE (1) is used when there are very large numbers of molecules.

4. MULTI-SCALED APPROACHES TO CHEMICAL REACTION SYSTEMS

Recently two new approaches by Rao and Arkin[29] and Haseltine and Rawlings[21] have been considered in an attempt to speed up the performance of the SSA. Both of these ideas are based on partitioning of the system. In the case of Rao and Arkin, they consider a time scale separation in which a subset of the system is asymptotically at steady state. This is called the quasi-steady-state assumption (QSSA) and eliminates the fast dynamics that is responsible for the poor performance of the SSA. If the QSSA is applied in deterministic kinetics, the ODEs describing the intermediate species are set to 0. In the stochastic setting the system is split into primary (y) and ephemeral (z) subsystems.

Let $P(y, z; t)$ be the probability density function of the entire system so that

$$P(y, z; t) = P(z|y; t)\, P(y; t).$$

Then Rao and Arkin assume that z conditional on y is Markovian, so that for fixed y the conditional probability distribution $P(z|y; t)$ approximately satisfies a master equation. If, in addition,

$$\frac{dP(z|y;t)}{dt} \approx 0$$

so that

$$P(z|y; t) \approx P(z|y),$$

then a chemical master equation for describing the evolution of the probability density function can be obtained solely in terms of the primary species y. The SSA can then be applied to this subsystem in a transparent manner. As a particular case Rao and Arkin[29] show how a simple enzymatic reaction involving an enzyme, substrate and enzyme-substrate complex in which the substrate concentration is much larger than the enzyme concentration leads, via QSSA arguments, to applying the SSA with a propensity function of the form $a(s) = \frac{\alpha s}{\beta + s}$ - which is of course the Michaelis-Menten approximation. Finally, Rao and Arkin[29] consider, as a specific example, the behaviour of the P_R promoter in conjunction with the Cro protein in λ bacteriophage. The P_R promoter plays an important regulatory component for determining the lysis or lysogenic pathways in the lambda infection of *E. coli*; see, for example, Shea and Ackers,[34] Arkin et al.,[2] Tian and Burrage.[37]

Using the ideas of Rao and Arkin,[29] Haseltine and Rawlings[21] attempt to speed up the performance of the SSA by partitioning a chemical reaction system into slow and fast reaction subsets. The slow subsystem corresponds to extents with small propensity functions and few numbers of reactions, while the latter corresponds to large propensity functions and large numbers of reactions. This partitioning is achieved by exploiting the structure of the CME and deriving master equations that describe the evolution of the probability density function for both the slow and fast subsystems. The slow system is treated by the SSA, while the fast system is treated either deterministically or by applying the explicit Euler-Maruyama method to the CLE. Thus at each time point t_n

the CLE is repeatedly solved until $t_{n+1} = t_n + \tau$ is reached and then the SSA is applied to the slow subsystem with a stepsize of τ.

Some remarks can be made about this approach.

- In order to move from the continuous to the discrete stochastic regime a rounding process must be adopted. This causes negligible errors as the values for the molecular species in the continuous regime are large.
- In the Haseltine and Rawlings approach it is not clear what the specific details for partitioning into slow and fast reactions are but they recommend maintaining at least two orders of magnitude difference between the partitioned reaction probabilities. However it is important for the partitioning to be adaptive and to change throughout the interval of integration.
- Haseltine and Rawlings use an explicit method, namely the Euler-Maruyama method, for simulating the CLE. However, since the propensity functions in the CLE are large, the SDE is stiff (in the sense of widely varying Lyapunov exponents) and thus some consideration could be given to using semi-implicit or fully-implicit methods for this component. This could come at some cost if the dimension of the fast subsystem is at least moderately large.

In spite of these remarks, the papers by Rao and Arkin[29] and Haseltine and Rawlings[21] represent a significant attempt for developing simulation techniques that interface between microscopic and macroscopic regimes.

Burrage et al.[8] extended this approach to classifying reactions into slow, intermediate and fast regimes. These regimes are characterised by the presence of one or more slow, intermediate and fast reacting species. The classification is in terms of the size of the propensity functions but also in terms of the number of molecules in the system. Thus at every time step they classify the system as slow, intermediate or fast and then form three vectors corresponding to the slow, intermediate and moderate regimes and place in those vectors the corresponding reaction number. If there are no reactions in say the intermediate vector for a given time step then that means there are no intermediate reactions for that step and the simulation regime changes accordingly.

In some cases it is possible to scale systems such that each term in the governing equations is composed of an expression of order of magnitude unity, multiplied by a dimensionless parameter, and this can lead to semi-autonomous simplification procedures. However, the approach adopted by Burrage et al.[8] is based on trying to get a completely general, adaptive, partitioning approach for simulating chemical reaction systems.

5. NOVEL IMPLEMENTATIONS

Recently the SSA has been applied in a number of settings involving genetic regulation. Arkin et al.[2] used the SSA to simulate a model of lambda phage containing 75 equations in 57 chemical species, while Kierzek[23] simulates the expression and activity of LacZ and LacY proteins in *E. coli* with 22 reactions and 23 molecular species. In this latter work, Kierzek presents a quite sophisticated implementation of the SSA in a software package known as STOCKS. The implementation treats both the growing volume of a cell and the simulation of cell division.

Because of the nature of SSA, even systems of moderate size such as the ones described above can take several hours to simulate on a fast PC. If hundreds or even thousands of simulations are needed inorder to calculate statistics about moments or to estimate the underlying probability density function, then it clearly makes sense to use novel forms of computing such as grid computing or parallel computing. The need for such infrastructure becomes even more apparent if we are trying to simulate cell models through the cell cycle from a single cell. Then we may be dealing with thousands of cells - see Smallwood et al.,[35] for example.

Indeed Endy and Brent[12] have observed that researchers investigating the cell doubling of relatively simple organisms such as *E. coli* require a single simulation of 10^{14} - 10^{16} reactions. In addition, in order to collate meaningful statistics, hundreds, if not thousands, of these simulations are needed. The main focus of this paper has been on a review of appropriate simulation algorithms operating at the slow, intermediate and fast reaction regimes. Irrespective of these algorithmic advances there is also a need to couple these approaches to sophisticated implementations using, for example, parallel and grid computing. A number of groups are working on this - see, for example, Kierzek,[23] Burrage et al.[5] and McCollum et al..[26]

Now we discuss how high performance computing can be used to reduce computational time. According to the features of underlying problems, it has been proposed that there are four types of parallelism for stochastic simulations. They are parallelism across the method, parallelism across the system, parallelism across the step and parallelism across the simulation (Burrage et al.[4]). Here we only address parallelism across the system and parallelism across the simulation.

We first propose a general formulae for representing propensity functions for different types of biochemical reactions. 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. These reactions can be written in a general form

$$X_i + X_j \xrightarrow{c} X_k \tag{6}$$

with propensity function $a(X) = cX_iX_j$. The third and higher order reactions are not studied here as they can be reasonably estimated by the combination of second order reactions.[23] For the three types of elementary reactions, the molecular species, reaction rates and propensity functions are defined in the following way:

(1) the first order reaction

$$X_i \xrightarrow{c_1} X_k, \quad a(X) = c_1 X_1 X_{M+1}$$

where X_{M+1} denotes a pseudo-molecular species whose molecular number is always 1;

(2) the second order reaction

$$X_i + X_j \xrightarrow{c_2} X_k, \quad a(X) = c_2 X_i X_j;$$

(3) the homodimer formation

$$X_i + X_i \xrightarrow{c_3} X_k, \quad a(X) = \frac{c_3}{2} X_i X_i.$$

Using the general form (6), propensity functions can be written as

$$a_j(X) = k_j X_{j1} X_{j2}, \quad j = 1, \ldots, n$$

which can be defined by a rate vector $k = (k_1, \ldots, k_n)^\top$ and a $(n \times 2)$ index matrix with elements $j1$ and $j2$ in the i-th row. Then the calculation of propensity functions can be implemented in parallel if the number of reactions is large.

For stochastic models we normally need hundreds or thousands simulations to get statistical properties of the underlying system. One feature of these stochastic simulations is the independence of each simulation. It is ideal to simulate the system on different processes in the MPI/PVM environment using the MASTER/SLAVE model. The MASTER process will send parameters and initial conditions to each SLAVE process, receive simulation results from each SLAVE, and then compute the final statistical values. Each SLAVE receives from the MASTER process the parameters it needs for computation, simulates the system and sends back the results. The MASTER process can either be a worker process for one simulation or just be the process for passing information between SLAVE processes.

One important issue in stochastic simulations is the independence of generated random numbers in each process. This property directly influences the statistical results of stochastic simulations. In the MPI/PVM environment we generate different random seeds for different processes and each process receives a seed from the MASTER process at the beginning of simulation. Then each process uses the generator RANDOM_NUMBER in FORTRAN 90 to generate uniformly distributed random numbers. In order to generate the Gaussian random numbers, we use the Box-Muller method, given by

$$G_1 = \sqrt{-2\ln(U_1)}\cos(2\pi U_2), \quad G_2 = \sqrt{-2\ln(U_1)}\sin(2\pi U_2), \tag{7}$$

where U_1 and U_2 are two independent $U(0,1)$ uniformly distributed random numbers, and G_1 and G_2 are two independent $N(0,1)$ standard Gaussian samples.

	Reaction	rate constant
1	PLac + RNAP → PLacRNAP	0.17
2	PLacRNAP → PLac + RNAP	10
3	PLacRNAP → TrLacZ1	1
4	TrLacZ1 → RbsLacZ + PLac + TrLacZ2	1
5	TrLacZ2 → TrLacY1	0.015
6	TrLacY1→ RbsLacY+TrLacY2	1
7	TrLacY2 → RNAP	0.36
8	Ribosome + RbsLacZ → RbsRibosomeLacZ	0.17
9	Ribosome+RbsLacY → RbsRibosomeLacY	0.17
10	RbsRibosomeLacZ → Ribosome+RbsLacZ	0.45
11	RbsRibosomeLacY → Ribosome+RbsLacY	0.45
12	RbsRibosomeLacZ → TrRbsLacZ+RbsLacZ	0.4
13	RbsRibosomeLacY → TrRbsLacY+RbsLacY	0.4
14	TrRbsLacZ → LacZ	0.015
15	TrRbsLacY → LacY	0.036
16	LacZ → dgrLacZ	6.42E-5
17	LacY → dgrLacY	6.42E-5
18	RbsLacZ → dgrRbsLacZ	0.3
19	RbsLacY → dgrRbsLacY	0.3
20	LacZ+lactose → LacZlactose	9.52E-5
21	LacZlactose → product+LacZ	431
22	LacY → lactose+LacY	14

Table 1. A full list of reactions and rates

Numerical results in this paper are obtained from parallel computing which is carried out on an SGI Origin 2000 scalable shared memory parallel computer at the University of Queensland. The command in Fortran 90 DATE_AND_TIME 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 over 3 runs.

6. A NUMERICAL EXAMPLE

In this section we will simulate the expression and activity of LacZ and LacY proteins in *E. Coli.* A detailed description of the biological significance of the model is given in Kierzek[23] but we give the full list of reactions here in Table 1. There are 22 reactions and 23 molecular species in this model. The initial state is PLac = 1, RNAP and Ribosome are generated from ramdom pools at each step, and all other elements 0.

In a single generation it is assumed that the cell doubles its volume from 1 to 2. This is achieved by letting the volume grow as $V(t) = 1 + t/T$, where T is the cell generation time. Thus at each simulation time step, the rates of all the second order reactions are divided by the current volume. Secondly when the system reaches the generation time, all of the reactants that model the DNA elements are doubled (implemented by a separate set of reactions from that being modelled). Then the numbers of all the molecules present in the system are divided by two, the volume of the cell is reset and the behaviour of a new cell is simulated for the next generation time.

In the OpenMP environment we use the command

```
!\$OMP PARALLEL DO PRIVATE(i), shared(a,b)
       DO I = 1, N
                .......
       END DO
!\$OMP END PARALLEL DO
```

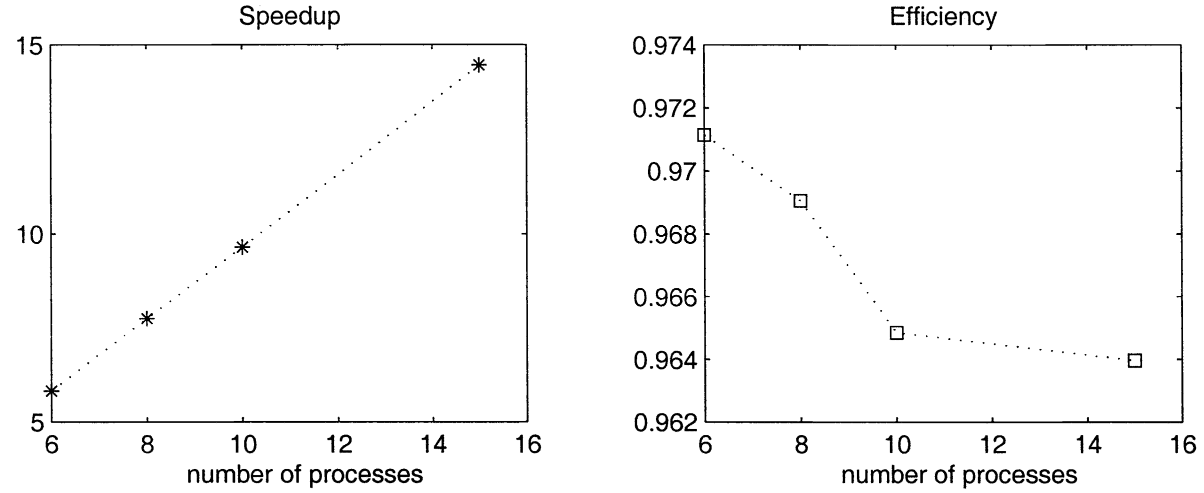

Figure 1. Speedup and efficiency of parallel computations in the MPI environment.

to perform the following parallel computations:

(1) calculate the propensity functions for 22 reactions at each step of SSA;

(2) simulate samples from random pools for RNAP and Ribosome;

(3) generate uniformly distributed random numbers.

As we record the numbers of proteins after every 10000 steps in SSA, the samples for the random pools and for the uniformly distributed random variable are generated in advance for every 10000 reactions.

As the number of reactions in this system is just 22, it is hard to see any speed-up in the OpenMP environment for a single simulation. The computing time with two processes is just a few minutes less than the sequential computing time which is one hour, thirty-six minutes and thirty-four seconds. If four processes are used, the computing time is larger than the sequential computing time. Here we should indicate that the idea in Section 5 is very useful for parallelism across the system. Currently we are investigating a biological system with spatial properties. When dividing space into a number of subspaces, we obtain a very large number of reactions in the system which is the sum of the numbers of reactions in all of the subspaces.

For parallel computing in the MPI environment, the MASTER sends the random seed and the initial condition to each SLAVE. After one simulation each SLAVE sends to the MASTER the simulation results which are then stored in a result matrix. If the number of simulations is larger than the number of available processes, we can determine in advance the number of simulations for each process. When all of the required simulation results are obtained, the MASTER will calculate certain statistical values such as the mean and variance.

MPI is an ideal environment for the parallel simulation of stochastic biological systems. From the simulation results given in Figure 1, we can see the efficiency is very close to 1. The data in Figure 1 are based on 120 simulations with 6, 8, 10 or 15 processes in use.

In conclusion, the dominating theme of the research described in this paper is the understanding of cellular dynamics in terms of interactions among the molecular components of a living cell. Of course we are a long way from this goal but new technologies such as the Functional Molecular Cinematography Unit offer a way of tracking the motion of individual molecules within a living cell. This offers a mechanism for the development and validation of more sophisticated models based on stochastic chemical reaction systems. If this is then coupled with sophisticated simulation and three dimensional visualisation techniques then we can really start to approach the holy grail of genomics, namely the ability to predict the dynamic effects on an organism of gene expression.

ACKNOWLEDGMENTS

The first named author would like to thank the Australian Research Council for funding under the Federation Fellowship scheme.

REFERENCES

1. J. Alcock and K. Burrage, "A note on the balanced method", to appear in *BIT*, 2004.
2. A. Arkin, J. Ross and H. H. McAdams, "Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells", *Genetics* **149**, pp. 1633–1648, 1998.
3. H. Berry, "Monte Carlo simulations of enzyme reactions in two dimensions: fractal kinetics and spatial segregation", *Biopys J.* pp. 1891–1901, 2002.
4. K. Burrage, P.M.Burrage and T. Tian, "Numerical methods for solving stochastic differential equations on parallel computers", *Proceedings of the 5th international conference on High-Performance Computing in the Asia-Pacific Region*, 2001.
5. K. Burrage, P. Burrage, S. Jeffrey, T. Pickett, R. Sidje and T. Tian, "A grid implementation of chemical kinetic simulation methods in genetic regulation", *Proceedings of APAC03 Conference on Advanced Computing, Grid Applications and eResearch*, 2003.
6. K. Burrage and T. Tian, "The composite Euler method for solving stiff stochastic differential equations", *J. Comp. App. Maths* **131**, pp. 407–426, 2001.
7. K. Burrage and T. Tian, "Poisson Runge-Kutta methods for chemical reaction systems", to appear in *Proceedings of the Hong Kong Conference on Scientific Computing*, 2004.
8. K. Burrage, T. Tian and P. Burrage, "A Multi-scaled Approach for Simulating Chemical Reaction Systems", to appear in *Prog. Biophysica and Molecular Biology*, 2004.
9. P. M. Burrage, *Runge-Kutta Methods for Stochastic Differential Equations*, PhD thesis, University of Queensland, Brisbane, Australia, 1999.
10. J. C. Butcher, *The Numerical Analysis of Ordinary Differential Equations*, Wiley, U.K., 1987.
11. M. B. Elowitz and S. Leibler, "A synthetic oscillatory network of transcriptional regulators", *Nature* **403**, pp. 335–338, 2000.
12. D. Endy and R. Brent, "Modelling cellular behaviour", *Nature* **409**, pp. 391–395, 2001.
13. N. Federoff and W. Fontana, "Small numbers of big molecules", *Science* **297**, pp. 1129–1131, 2002.
14. C. J. Firth and D. Bray, "Stochastic simulation of cell signaling pathways", in *Computational Modeling of Genetic and Biochemical Networks*, J.M. Bower and H. Bolouri, ed., MIT Press, Cambridge, MA., pp. 263–286, 2000.
15. M. A. Gibson and J. Bruck, "Efficient exact stochastic simulation of chemical systems with many species and many channels", *J. Phys. Chem. A* **104**, pp. 1876–1889, 2000.
16. D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions", *J. Phys. Chem.* **81**, pp. 2340–2361, 1977.
17. D. T. Gillespie, "A rigorous derivation of the chemical master equation", *Physica A* **188**, 404–425, 1992.
18. D. T. Gillespie, *Markov Processes: an introduction for Physical Scientists*, Academic Press, 1992.
19. D. T. Gillespie, "Approximate accelerated stochastic simulation of chemical reaction systems", *J. Chem. Phys.* **115**, pp. 1716–1733, 2001.
20. D. Gonze, J. Halloy and A. Goldbeter, "Robustness of circadian rhythms with respect to molecular noise", *Proc. Natl. Acad. Sci.* **99**, pp. 673–678, 2002.
21. E. L. Haseltine and J. B. Rawlings, "Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics", *J. Chem. Phys.* **117**, pp. 6959–6969, 2002.
22. D. A. Hume, "Probability in transcriptional regulation and its implications for leukocyte differentiation and inducible gene expression", *Blood* **96**, 2323–2328, 2000.
23. A. M. Kierzek, "STOCKS: stochastic kinetic simulations of biochemical systems with Gillespie algorithm", *Bioinformatics* **18**, pp. 470–481, 2002.
24. P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*, Springer-Verlag, Berlin, 1992.

25. R. Kopelman, "Rate-processes on fractals - theory, simulations and experiments", *J. Stat. Phys.* **42**, pp. 185-200, 1986.

26. J. M. McCollum, C. D. Cox, M. L. Simpson and G. D. Peterson, "Accelerating gene regulatory network modeling using grid-based simulation", University of Tennessee, 2002.

27. G. Milstein, E. Platen and H. Schurz, "Balanced implicit methods for stiff stochastic systems", *SIAM J. Numer. Anal.* **35**, pp. 1010–1019, 1998.

28. C. J. Morton-Firth, "Stochastic simulation of cell signalling pathways", PhD thesis, Cambridge, 1998.

29. C. Rao and A. Arkin, "Stochastic chemical kinetics and the quasi-steady-state assumption: application to the Gillespie algorithm", *J. Chem. Phys.* **118**, pp. 4999–5010, 2003.

30. M. Rathinam, L. R. Petzold and D. T. Gillespie, "Stiffness in stochastic chemically reacting systems: The implicit tau-leap method", submitted for publication 2003.

31. V. H. Resat, H. S. Wiley and D. A. Dixon, "Probability-weighted dynamic Monte Carlo method for reaction kinetics simulations", *J. Phys. Chem. B* **105**, pp. 11026-11034, 2001.

32. Y. Sano, T. Shimada, H. Nakashima, R. H. Nicholson, J. F. Eliason, T. A. Kocarek and M. S. H. Ko, "Random monoallelic expression of three genes clustered within 60Kb of Mouse t Complex Genomic DNA", *Genome Res.* **11**, pp. 1833–1845, 2001.

33. S. Schnell and T. E. Turner, "Reaction kinetics in intracellular environments with macromolecular crowding: simulations and rate laws", to appear in *Prog. Biophysica and Molecular Biology*, 2004.

34. M. A. Shea and G. K. Ackers, "The O_R control system of bacteriophage lambda: a physical-chemical model for gene regulation", *J. Mol. Biol.* **181**, pp. 211–230, 1985.

35. R. Smallwood, M. Holcombe, D. walker, J. Southgate, S. MacNeil, "Computational modeling of the social behaviour of cells", *Biosystems review*, 2003.

36. T. Tian and K. Burrage, "Implicit Taylor methods for stiff stochastic differential equations", *Appl. Numer. Math.* **38**, pp 167–185, 2001.

37. T. Tian and K. Burrage, "Bistability and switching in the lysis/lysogeny genetic regulatory network of bacteriophage λ", *J. Theor. Biol.* **227**, pp. 229–237, 2004.