

HSIM: A Hybrid Stochastic Simulation System for Systems Biology

Patrick Amar¹

*LRI, UMR CNRS 8623 & INRIA AMIB
Université Paris Sud, 91405 Orsay cedex, France*

Loïc Paulevé²

*LIX, École Polytechnique, 91128 Palaiseau Cedex, France
ETH Zurich, Physikstrasse 3, 8092 Zurich, Switzerland*

Abstract

This paper describes a method to obtain in a certain sense the best of two well known simulation approaches: Stochastic simulation of chemical reactions and Entity-centered (*multi-agents*) systems. The stochastic simulation algorithm and its enhanced versions are adapted to simulate the behaviour of well stirred mixture of reactants when spatial localisation is not relevant; Conversely, in entity-centered methods, the causes and consequences due to spatial localisation are well handled. In terms of computational efficiency, the SSA methods are limited by the number of reactions and not by the amount of reactants, whereas it is the opposite for the entity-centered method.

We will show how we can benefit from each approach, and also propose a new optimised algorithm for approximate stochastic simulation.

Keywords: Stochastic simulations, Entity-centered systems, Hybrid systems.

1 Introduction

Many of the biological objects (genetic regulatory networks, metabolic networks, signalling pathways, etc.) biologists and modellers are interested in, do not need to take into account the spatial location of the biomolecules that are involved in the studied processes. In the cases, where there are large populations of molecules, even a stochastic approach is not absolutely necessary, and a deterministic method, such as ordinary differential equations, gives useful results. But conversely, many biological processes involve a small number of some molecular species. In some

¹ Email: pa@lri.fr

² Email: loic.pauleve@ens-cachan.org

cases, the spatial localisation of these molecules, or of some more abundant species, is the key cause of the behaviour of the whole system.

The tetrameric repressor of the *lac* operon in *E. coli* is a good example of this [8]. There are around ten copies only of this repressor in each bacterium, but due to its high local concentration near its binding site on DNA, its effects are largely noticeable. The cause of this high local concentration is inherent to the mechanism of operation of this repressor, and to the specific distance between the locations of its binding sites on the DNA. This is a co-operative process which tends to perpetuate itself.

To study these kind of biological processes we need a well founded framework that can both account for the stochasticity and the low numbers of participants, but also for the spatial localisation of molecules. Finally, its implementation must be efficient in terms of computation time.

We present here the HSIM simulator which implements an hybrid simulation mixing entity-centered model with a new approximated Stochastic Simulation Algorithm (SSA). HSIM has been successfully applied to study large assemblies of proteins involving spatial features [1,2].

Related work. [7] recently introduced an hybrid spatial simulator involving exact SSA and individual particle tracking. Contrary to HSIM, their method allows to dynamically change the treatment of a species by switching to either individual particles or homogeneously distributed molecules.

2 Entity-centered model

The entity-centered methods are similar to multi-agent systems: each molecule is a kind of agent placed in an environment; The computer calculates the trajectory and evolution in time and space according to its characteristics (chemical species), its interactions with other agents (biochemical reactions) and the environment (mainly the membrane which delimits the compartment).

The main difference with the multi-agent systems is that all the types of entities have the same behaviour, of course parameterised by their type, but basically identical. We will benefit from this simplification to get an efficient implementation of the system. This common behaviour has two components:

- (i) **diffusion**: each molecule diffuses into the compartment according to Brownian motion, its velocity dependent on the temperature, the viscosity of the medium and the molecular crowding.
- (ii) **reaction**: due to its diffusion and the occurrence of a collision of sufficient energy with another molecule of the appropriate species, a reaction can occur, leading to the formation or the dissociation of a complex, the disappearance, or a change of type for one, or for both molecules.

2.1 Diffusion and Brownian motion

If we observe small particles in water, we will see that they move erratically. This behaviour, the Brownian motion, is due to collisions with water molecules, which makes the particles undergo random-walk motion.

If we observe such a particle for a time interval t , we will see it displaced by a position vector r . If we repeat this over and over again, we can compute the average displacement during time interval t , and we should find that $\langle r \rangle = 0$. That means that there is no preferred direction for the random forces exerted by the water molecules on the particle.

The Brownian motion is characterised by the *average displacement-squared*:

$$\langle |r|^2 \rangle = 6Dt$$

where D is the diffusion constant. The diffusion constant D depends both on the size and shape of the particle, and on the viscosity and temperature of the fluid that it is moving through. The factor 6 is related to the *dimensionality*, 6 is for 3D motion (4 is for 2D, and 2 is for linear motions). Basically, diffusion is due to the independent Brownian motion of many molecules.

2.2 Random walk

Here is a simple model that show that a random walking particle in a 3D space obeys to the same average displacement-squared law than Brownian motion.

Let's assume a particle making a step of length l every time interval τ , with each step in a random direction. Then the displacement after N steps is:

$$r = l\vec{n}_1 + l\vec{n}_2 + \dots + l\vec{n}_N$$

where \vec{n}_i are unit vectors in random directions. So the square of the distance travelled by the particle after N steps is: $|r|^2 = l^2 (\vec{n}_1 + \vec{n}_2 + \dots + \vec{n}_N)^2$

$$|r|^2 = l^2 \left(|\vec{n}_1|^2 + |\vec{n}_2|^2 + \dots + |\vec{n}_N|^2 + \sum_{i \neq j} \vec{n}_i \cdot \vec{n}_j \right) = Nl^2 + l^2 \sum_{i \neq j} \vec{n}_i \cdot \vec{n}_j$$

Now, if we repeat many times this random walk, we would find that the average dot product of two different steps is zero, $\langle \vec{n}_i \cdot \vec{n}_j \rangle = 0$, since the steps are in random directions. So the average displacement-squared of the particle after N steps is $\langle |r|^2 \rangle = Nl^2$. The time after N steps is just $t = N\tau$ so the displacement-squared grows linearly with time:

$$\langle |r|^2 \rangle = \frac{l^2}{\tau} t$$

This means that this random-walk is a model of Brownian motion with diffusion constant $D = l^2/6\tau$.

So, in our entity-centered simulator, the diffusion part will be made using a random walk for the molecules. The parameters will be chosen to account the diffusion speed of proteins and small molecules observed *in vivo* in real cells.

2.3 Reactions

Reactions can be classified in two main categories: unimolecular and bimolecular. Reactions involving more than two reactants can be reduced to a combination of bimolecular reactions.

Unimolecular reactions concerns a single molecule which can exist in a number of different states. The spontaneous fluctuation of the structure of a protein from one conformation to another is an example of a unimolecular reaction. Another example is the dissociation of a macromolecular complex. In entity-centered stochastic simulators, these unimolecular reactions will be modelled with a probability of being triggered.

To trigger a bimolecular reaction two events must occur: firstly a copy of each reactant molecules must collide, and secondly this collision has to have a sufficient energy. The collisions arise due to the diffusion of the molecules; it is not very difficult to see that the average number of collisions is proportional to the product of the concentrations of the reactants. This is known as *the mass action law*, set out by Guldberg and Waage in 1864.

This will be implemented in the simulator by testing, in the diffusion phase, if the random move of a molecule leads to a collision with another one. If so, the reaction is triggered according to a probability representing its kinetics.

Since the molecules are entities, the simulator can take into account the assemblies using *physical* links between these entities, slowing down their diffusion. Furthermore, large aggregates can clutter up the space, possibly leading to significant changes of the diffusion of other molecules and thus to *abnormal* local concentrations (either higher or lower). In turn, these singularities can be determinant for the behaviour of the whole system. None of these effects can be taken into account with more global methods which suppose a permanently homogeneous mixture of molecules.

This feature come with a cost in terms of computing time, even in the cases where no topological effects seems to appear where thus global methods would be more efficient.

3 The hybrid system HSIM

HSIM, for *Hyperstructure SIMulator*, is a simulation system initially developed to take into account the aggregation and dissociation of large molecular assemblies called Hyperstructures. These hyperstructures allow the structuration of the bacterial cell, they may play a key role in the metabolism regulation, cell division, cell differentiation, etc.

The simulation system has to account for the dynamics of complexes formation

/ dissociation, their localisation in the cell, their actions on the metabolic and signalling pathways and also on the regulatory gene networks.

3.1 Shape, size and time scale

HSIM is mainly designed for the modelling of bacteria, but can also be used for small eukaryotic cells such as yeast or fungus. The main compartment is a cylinder surrounded by an hemisphere at each end. It can be set to be rod shaped, like many bacteria, or to show a spherical conformation, like yeast. Subcompartments can be nested into the main compartment.

The cells are roughly microns in size. *E. coli* cells are small rods usually around 1μ by 3μ . Yeast cells are little balls around 2μ in diameter. Most human cells are roughly 20μ across, with roughly nuclei about 10μ in diameter.

The basic length scale used to describe molecules is a nanometer. Many biomolecules are made out of chemical units (e.g. nucleotides, amino acids) which are around 1nm in size. Other *small* molecules, such as lipids and sugars, are also roughly on this length scale. Most folded up proteins are a few nm in diameter.

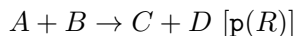
In the bacterial cytosol, due to the molecular crowding, the diffusion constant is considerably lower than in water. We adjusted the time step ($100\mu\text{s}$) and the jump distance (10nm) for the random walk used in HSIM to model the Brownian motion, according to the average velocity of molecules observed in real cells.

3.2 Inside HSIM

The simulator, HSIM, is a stochastic automaton driven by reaction rules between molecules.

In essence, each molecule is represented by a record that includes its type, its position, its size and a list of links to certain other molecules. HSIM keeps track of each molecule in real time from the computer point of view. The basic principle is that time is sliced into consecutive steps (iterations) of same duration, and in each iteration the rules are applied to every molecule. These rules mimic the chemical reactions between molecules in a real system.

The model to be simulated is specified as a set of probabilistic rewriting rules. These rules have the following generic form:



This rule means that when M_A and M_B (two molecules of species A and B) collide, they trigger the reaction that transforms M_A into M_C and M_B into M_D (of species C and D respectively) with the probability $p(R)$. The probability reflects the fact that the collision has enough energy to initiate the reaction, and correlate with the macroscopic interpretation of the reaction kinetics. The second molecule on the left side of the rule can be omitted to specify a unimolecular reaction. Similarly, the omission of the second molecule on the right side of the rule, specifies a reaction giving only one product.

3.3 Calculation algorithm

In Hsim, space is continuous since the molecules coordinates are coded with real numbers. The time is discrete, it is cut in slices of equal duration, during which one step of diffusion is made, and the resulting reactions triggered. Assuming only entities are involved, here is an overview of the algorithm used to compute one step of the evolution of the system.

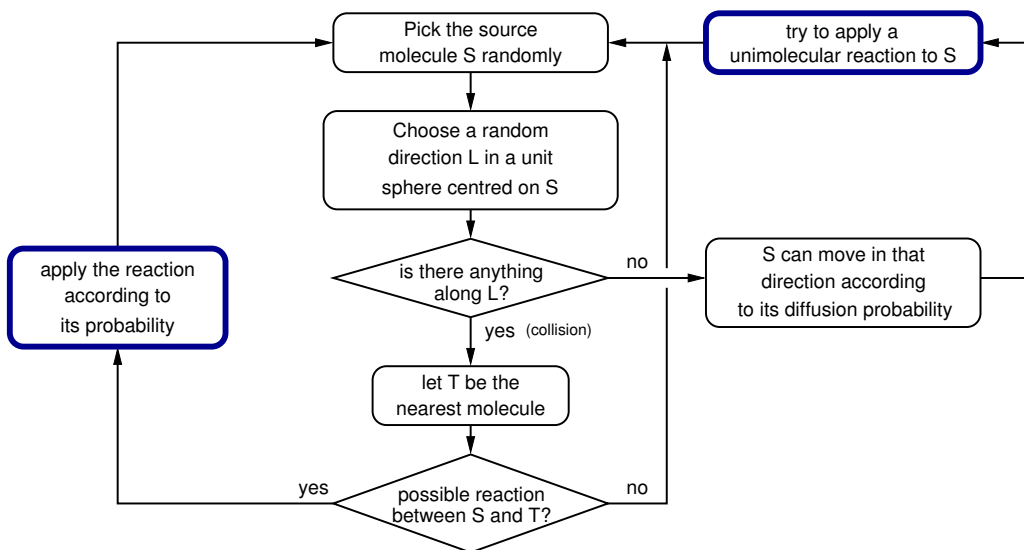


Fig. 1. Algorithm performed at each time slice for each molecule.

At each step, all the molecules are considered one and only once, in random order, to avoid statistical bias, and so correctly simulate one process for each entity. The examination of the molecule S consists in firstly verify it has not been previously processed during the same iteration, then to choose a randomly oriented vector of diffusion. If there is no obstacle along this vector (another molecule or the membrane), the molecule will be moved at its end with a probability related to its diffusion velocity, and the system will try to apply a unimolecular reaction. Conversely, if there is a collision with another molecule, T , which has not been

already examined during this iteration, the system will try to apply a bimolecular reaction rule between S and T . Then T is marked as “already processed”, and the next molecule is randomly picked to be processed, and so on until exhaustion of all the molecules. Finally the time is advanced of one time step and the next iteration can begin (fig. 1).

As previously mentioned, HSIM is an hybrid simulator: besides molecules treated by individual entities, it includes other chemical species treated in a global way. This is frequently small molecules with high diffusion velocity, that are present in high copy number, and are therefore statistically homogeneously distributed inside the compartment. These classes of molecules, which we call *non-entity*, will be represented only by their type and number of copies in the compartment. It would be drastically more efficient to process them using a global approach similar to the Gillespie stochastic method. This new feature requires a slight modification of the previous algorithm in two points:

- (i) During the diffusion phase, if an entity may react with *non-entity* species, we have to compute the average number of collisions between them during the time slice, then apply the resulting reactions.
- (ii) At each iteration, before or after the processing of the *molecule-entities*, we must process the specific interactions between non-entity molecules: compute the average number of their collisions during the time slice, then apply the resulting reactions.

One can notice that with this method, it is possible to have reactions between entities, between entity and non-entity, and finally between non-entities. Furthermore, the syntax used to describe the reactions remains the same regardless the class of the molecules involved.

3.4 Entity and non-entity

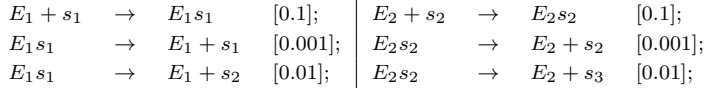
To compute the average number of collisions during a time slice between an entity and a non-entity, it is sufficient to compute the volume displaced by the entity molecule, and then, knowing the concentration of the non-entity molecules, to infer their average number of copies inside that volume. Since by definition, the non-entity species are homogeneously distributed inside the compartment, all we need to do is to multiply the total number of copies of this particular molecular species by the ratio of the volume displaced by the entity to the total volume of the compartment.

Then, for each of these collisions, the corresponding reaction rules are applied according to their probabilities. We will see more precisely how this is implemented in HSIM in the next section.

3.5 A simple example

To demonstrate the three simulations methods offered by HSIM, we have used a simple example of a cascade of two enzymes, E_1 and E_2 , where the product of the first enzyme, s_2 , is the substrate of the second one. To keep this example simple, the kinetics of the two enzymes are the same. To emphasise the stochastic effects

of the simulations, the concentration of the metabolites is very low: 100 copies for a compartment volume of one micron-cube ($0.166\mu M$).



The first method is illustrated by specifying that all the molecules are modelled with entities; The curves in figure 2A show a trajectory of the evolution of the quantities of metabolites s_1 , s_2 and s_3 .

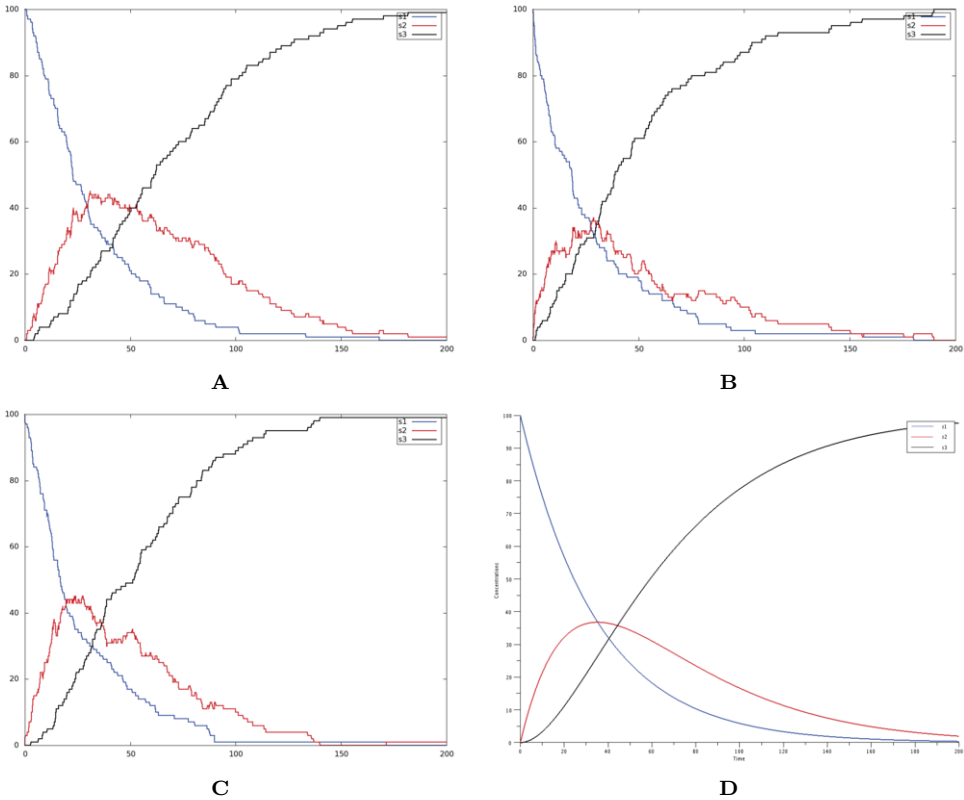


Fig. 2. **A**: simulation trace of the two enzymes cascade model using HSIM entity method, **B**: using HSIM entities for enzymes and non-entity for metabolites, **C**: using HSIM SSA method, **D**: using SCILAB ODE resolution,

To illustrate the second method, the enzymes are modelled as entities, while the substrates are treated globally; The curves in figure 2B show a trajectory of the evolution of the quantities of metabolites s_1 , s_2 and s_3 .

Finally, the third method is illustrated by specifying all the molecules are to be treated globally, using HSIM SSA; The corresponding curves are shown in figure 2C.

To have an idea of the limit curves, when there are large concentrations of reactants and when the stochastic effects are neglected, an ODE integration of the corresponding model is shown in figure 2D.

4 Approximate Stochastic Simulation

In order to accurately simulate the evolution of a large mixture of various species, we generally use of a Stochastic Simulation Algorithm (SSA), such as the exact Direct Method of Gillespie [5].

One of the major drawbacks of exact SSA algorithms is their high computational cost when a large number of fast reactions are to be simulated. To overcome this limitation, various approximate SSA have been drawn, the majority of them being based on the *tau-leap* method [6]. Instead of computing the next reaction to occur and update the time accordingly, tau-leap methods estimate the number of reactions occurring in a fixed time interval (τ) and apply them at once. Such algorithms ignore the interferences between applied reactions, hence providing an approximation of the evolution of the system.

HSIM implements an approximate SSA algorithm similar to the tau-leap method. Its main originality resides in the evaluation of the number of reactions occurring in the fixed time interval, which requires the generation of very few (uniform) random variables. Hence its total complexity is very low, with the consequence of reducing the time variance of abundant reactions (but preserving their mean duration).

4.1 Algorithm

As introduced in the previous section, each reaction rule in HSIM (either unary or binary) is associated to a probability of being triggered within a fixed time step. In the remaining of this section, we denote by $p(R)$ such a probability for the reaction R .

The main simulation loop of HSIM SSA is as follows: at each time step τ , (i) compute the number of possible reactions, (ii) evaluate the number of effective reactions (according to $p(R)$), (iii) applies at once these reactions.

If we compute only the mean number of effective reactions, the results will be close to an ODE integration. In order to introduce some stochasticity, the (integer) counts of reactions are randomly rounded: given a real number x , its fractional part is compared to a uniformly distributed random number in the interval $[0; 1]$ (noted $\mathcal{U}_{[0;1]}$) to decide whether x is rounded to the lower or upper integer. $\lfloor x \rfloor$ denotes the integer part of x .

```

randround( $x$ ):
  (i)  $i \leftarrow \lfloor x \rfloor$ 
  (ii) if  $\mathcal{U}_{[0;1]} \leq x - i$  then  $i \leftarrow i + 1$ 
  (iii) return  $i$ .

```

During a time interval τ and for each reaction R , HSIM evaluates **collisions**(R) as the number of time R may occur. In the case of unary reactions (1), it is simply the population of the transforming species ($[A]$ is the population of species A). In the case of binary reactions (2), it is the total number of possible collisions, scaled

by the cell volume V and a constant α .

$$\text{collisions}(A \rightarrow C) \triangleq [A] \quad (1)$$

$$\text{collisions}(A + B \rightarrow C + D) \triangleq \frac{\alpha}{V} \begin{cases} [A][B] & \text{if } A \neq B \\ [A]([A] - 1)/2 & \text{otherwise.} \end{cases} \quad (2)$$

The counting of the effective reaction R is then done by randomly rounding the number of collisions multiplied by the probability for these collisions to trigger the reaction.

```
count( $R$ ):
  (i)  $s \leftarrow \text{collisions}(R)$ 
  (ii)  $n \leftarrow \text{randround}(s \cdot p(R))$ 
  (iii) if ( $R$  is binary)  $n \leftarrow \min(n, [R_A], [R_B])$ 
  (iv) return  $n$ 
```

Where $[R_A]$ (resp. $[R_B]$) stands for the population of the first (resp. second) reactant of the reaction R (see subsection 4.3). Finally, at each constant time step τ , the main simulation loop applies, in a random order, each reaction as many time as it is effective (returned by `count(R)` function). It can be summarised by the following steps, where t is the total simulated time:

```
(i)  $t \leftarrow 0$ 
(ii) Repeat forever:
  (a) For each reaction  $R$  sorted randomly:
     $n \leftarrow \text{count}(R)$ .
    apply the reaction  $R$   $n$  times.
  (b)  $t \leftarrow t + \tau$ 
```

Overall, the computational complexity of one time step is linear with the number of reactions and includes the generation of one uniform random variable for each reaction. In addition, the list of reaction has to be randomly shuffled, which can be done in linear time with R . The total complexity is then in $\mathcal{O}(\#R \cdot t)$ where $\#R$ is the number of reactions.

In the current implementation of HSIM, τ is fixed to $100\mu\text{s}$ (see 3.1); α has then been evaluated to $7.4 \cdot 10^{-7}$ in order to match the number of collisions observed in the entity-centered model with the same species concentrations.

4.2 Relationship between HSIM parameters and standard reactions rates

Given a reaction R , HSIM assumes a parameter $p(R)$ assigning the probability for the reaction to be effective within a time step τ . This parameter should depend only on the nature of involved molecules and the kind of the chemical reaction. On the other hand, standard SSAs *à la* Gillespie assume that each reaction R is parameterised by a *rate* (noted $r(R)$), giving the mean number of time the reaction can be triggered in one unit of time. This rate embeds both the volume of the cell

and the probabilities of triggering the reaction after one collision.

In that way, HSIM allows to specify parameters independently of the cell volume, providing a transparent scaling of rates when the volume changes.

The relationship between $\mathbf{p}(R)$ and $\mathbf{r}(R)$ can be straightforwardly derived from the HSIM SSA, where the time interval is composed of τ unit of time:

$$\text{unary reaction: } \mathbf{r}(R) = \frac{1}{\tau} \mathbf{p}(R); \quad \text{binary reaction: } \mathbf{r}(R) = \frac{1}{\tau} \mathbf{p}(R) \frac{\alpha}{V}.$$

Because $\mathbf{p}(R)$ has to be in $[0; 1]$, this relationship imposes a maximum boundary on the rates of reactions, depending on τ , α and the cell volume V . Assuming these parameters fixed, this uncovers a trade-off between an exact SSA having its costs following the number of occurring reactions (but being independent from the absolute simulation time), and the HSIM SSA having its costs following the total simulation time, which may have to be rescaled in order to satisfy the above relationships.

4.3 Discussion

As described in subsection 4.1, the proposed algorithm approximates the number of firing of reaction R by the random variable $n = \text{randround}(s \cdot \mathbf{p}(R))$ where $s = \text{collisions}(R)$ is the count of possible applications. This random variable n is actually the sum of the integer part of $s \cdot \mathbf{p}(R)$, noted i , and a Bernoulli random variable with success probability being the non-integer part of $s \cdot \mathbf{p}(R)$, noted f . Hence, the expected value of n is $i + f = s \cdot \mathbf{p}(R)$, and its variance is $f - f^2$.

Classical τ -leaping methods typically approximate the number of firing of reaction R with a Poisson random variable $\mathcal{P}(a \cdot \mathbf{r}(R) \cdot \tau)$, where a is the so-called activity of reaction R : for unary reactions $A \rightarrow B$, $a = [A]$ and for binary reactions $A + B \rightarrow C$, $a = [A][B]$ if $a \neq B$, otherwise $a = [A]([A] - 1)/2$. From the relationship between $\mathbf{p}(R)$ and $\mathbf{r}(R)$ established in the previous subsection, it is clear that $a \cdot \mathbf{r}(R) \cdot \tau = s \cdot \mathbf{p}(R)$. Hence, the HSIM approximated SSA has the same expected value than classical τ -leaping methods, but has a much lower variance, especially when i (the integer part of $s \cdot \mathbf{p}(R)$) is large.

It is worth noticing that, as stated, the HSIM SSA algorithm cannot undergo negative population. A population can become negative only in two cases:

- (i) Suppose there are two reactions, R_i and R_j , which consume the species A. When the result of the application of reaction R_i leads to deplete enough the population of reactant A so that the application of reaction R_j leads to consume more than the remaining population of reactant A.
- (ii) In a binary reaction $A + B \rightarrow C$ when $[B]/[A] > V/\alpha \cdot \mathbf{p}(R)$, with $[B] \gg [A]$, the worst case being $[A] = 1$. This can lead to a calculated number of application of the reaction greater than the population of one of its reactants.

In the HSIM SSA algorithm, the first case can never appear because the application of the reactions being done sequentially, the next call to `count` will use the already

updated populations of reactants. This is why, to avoid statistical bias, at each time step, the reactions are randomly sorted prior to their application.

The second case is avoided in the HSIM SSA algorithm by a specific test done in the case of binary reactions. One can notice that this particular case can appear only when the population of one reactant is considerably larger than the population of the other one. With the current settings of HSIM ($\alpha = 7.4 \cdot 10^{-7}$), this leads to a very high concentration rarely reached in physiological conditions (of the order of few milli molar).

4.4 Elementary examples

We have compared the HSIM SSA algorithm with two of the well known methods: the exact Gillespie SSA method implemented in BIOCHAM and the ODE resolution method implemented in SCILAB. For the latter, the differential equations system was automatically obtained from the *export* feature of HSIM.

To achieve this goal we have used three elementary examples:

- (i) the *single* model in which a unimolecular reaction transforms species A to species B at a given rate.
- (ii) the *binary* model where a bimolecular reaction degrade couples of species A and B.
- (iii) the *concur* model where some kind of *race condition* exists between species A and species B.

4.4.1 The single model

This model is shown below using the HSIM modelling language. It simply states that there are two different species A and B which can diffuse in a compartment of one micron-cube volume. The only reaction of the model transforms each copy of A into a copy of B with the probability 10^{-5} . Initially there are 1000 copies of species A inside the compartment.

```
geometry = 142:110; // 1 u^3 compartment.
metabolite A, B;
A -> B [1e-5];
init (1000, A);
```

Figures 3A and 3B show one sample trajectory of the model using HSIM SSA and the exact Gillespie method; As one can see these curves are very similar. The curves in figure 3C show the ODE resolution of the same model with the corresponding values of the parameters. These perfect curves can be approximated by plotting the mean values from many runs of HSIM (figure 3D). The error bars show the standard deviation at each point.

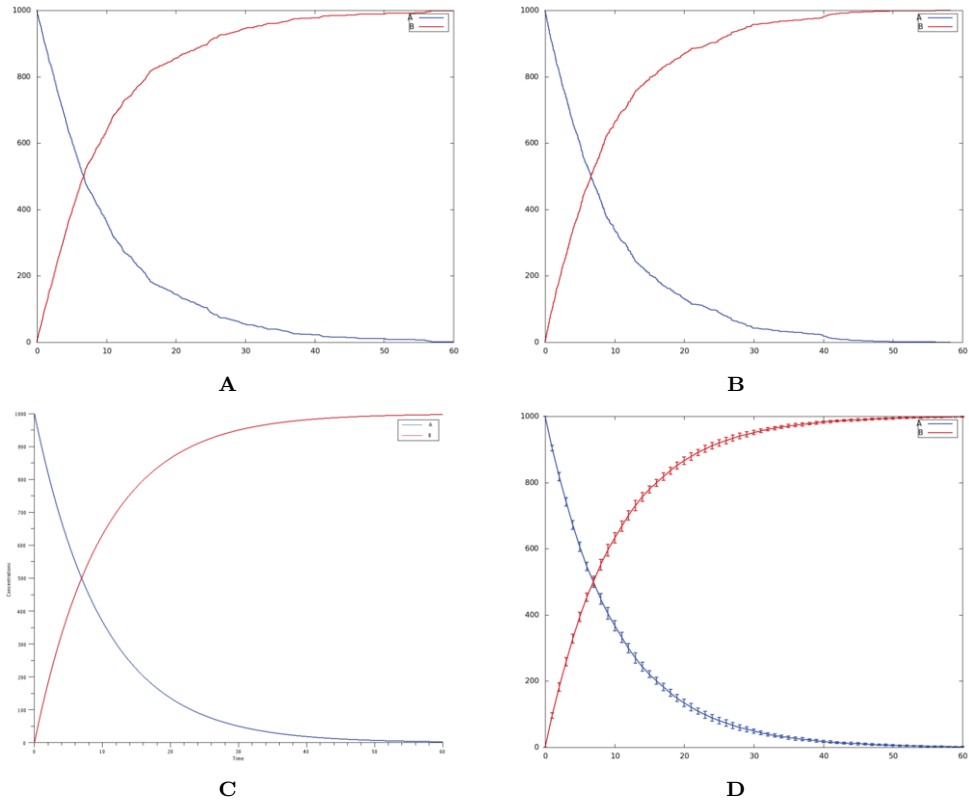


Fig. 3. **A:** simulation trace of the *single* model using HSIM SSA method, **B:** using BIOCHAM SSA exact method, **C:** using SCILAB ODE resolution, **D:** mean value and standard deviation obtained with 20 runs of HSIM.

4.4.2 The binary model

In this model there are two different species A and B, which can diffuse in a compartment of one micron-cube volume. The only reaction of this model transforms each pair of molecule of type A and B into a single molecule of type C, with the probability 0.135. The species C has a very short lifetime: as indicated in the *metabolite C[1]* statement, each copy of this species vanishes after 1 timestep, so basically this reaction degrades pairs of A and B. Initially there are 1000 copies of each of the species A and B inside the compartment.

```

geometry = 142:110; // 1 u^3 compartment.
metabolite A, B, C [1];
display (A, B);
A + B -> C [0.135];
init (1000, A);
init (1000, B);

```

Again, figures 4A and 4B show one sample trajectory of the model using HSIM SSA and the exact Gillespie method; These curves are also very similar. The curves in figure 4C show the ODE resolution of the same model with the corresponding values of the parameters. Figure 4D) shows the plotting of the mean values in each point,

obtained from many runs of HSIM.

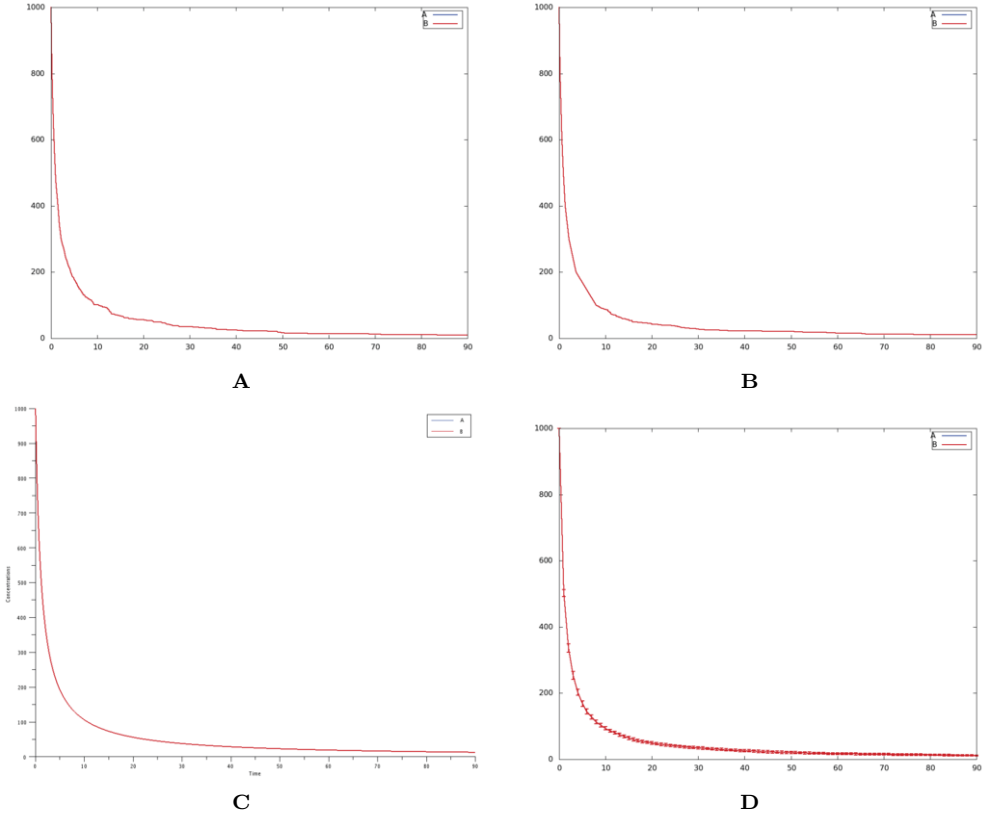


Fig. 4. **A:** simulation trace of the *binary* model using HSIM SSA method, **B:** using BIOCHAM SSA exact method, **C:** using SCILAB ODE resolution, **D:** mean value and standard deviation obtained with 20 runs of HSIM.

4.4.3 The *concur* model

In this model, species A is competing between two futures: it can vanish ($A \rightarrow D$) or by reacting with itself create species B ($A + A \rightarrow B$). Meanwhile, species B has also two possible futures: synthetise two copies of A ($B \rightarrow A + A$) or be transformed irreversibly into C ($B \rightarrow C$). As in the previous example, species D has a very short lifetime, so the reaction $A \rightarrow D$ degrades A with the probability 10^{-4} . Initially there are 10000 copies of the species A inside the compartment.

```
metabolite A, B, C, D [1];
A -> D [1e-4];
A + A -> B [0.2705];
B -> A + A [5e-5];
B -> C [4e-6];
init (10000, A);
```

The behaviour of this model is very dependent on the kinetics of the reactions. With those chosen here, after a very short period of time, about 80% of species A are

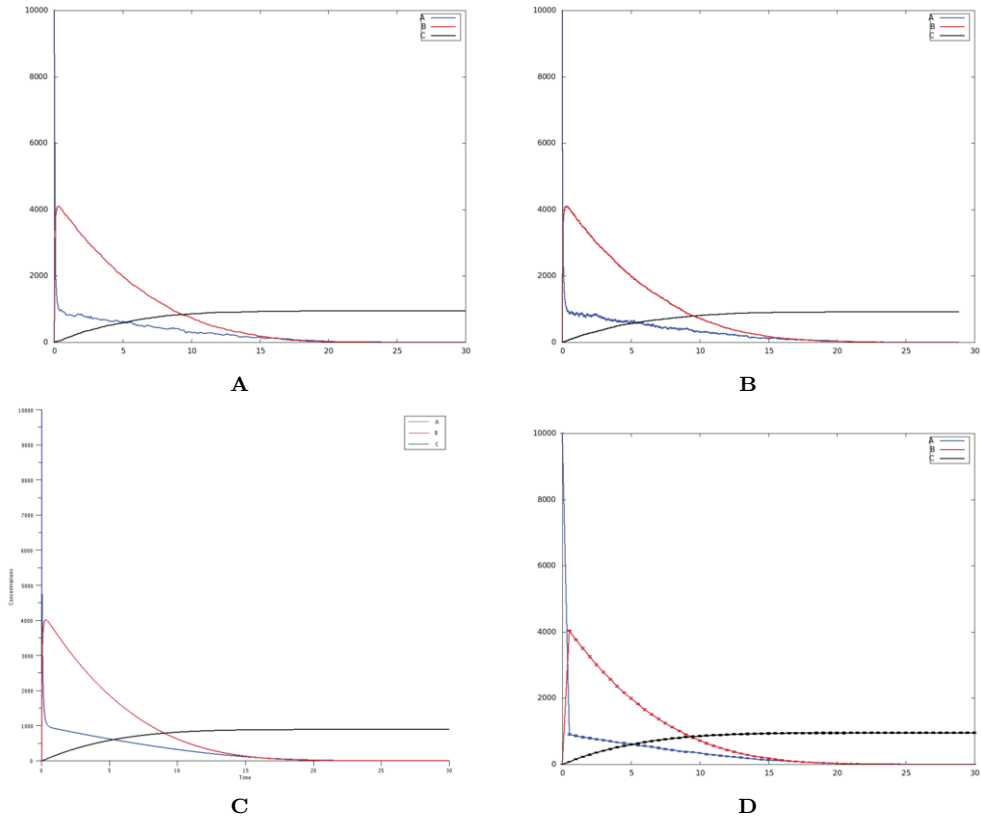


Fig. 5. **A:** simulation trace of the *concur* model using HSIM SSA method, **B:** using BIOCHAM SSA exact method, **C:** using SCILAB ODE resolution, **D:** mean value and standard deviation obtained with 20 runs of HSIM.

transformed into B, while the rest is degraded. Then most of the B are transformed into C, while the remaining B are transformed back into A.

As in the previous examples, the different methods show similar curves. One can notice in figure 5D that the fall of species A is not as quick as in the other simulations. This is due to the low level of quantisation used to make visible the error bars.

5 Case studies

5.1 The circadian clock in the cyanobacteria

The cyanobacterium *Synechococcus elongatus* is a gram-negative bacterium that uses photosynthesis for energy intake. Its metabolism is regulated by an internal circadian clock synchronised with daylight through photosensitive receptors. The clock mechanism is mainly made of three proteins KaiA, KaiB and KaiC. In the cyanobacterium these proteins are genetically regulated, but spontaneous oscillations of the phosphorylation level of KaiC hexamers, showing a period of approximately 24h, have been observed *in vitro* in a mixture of KaiA, KaiB, KaiC and ATP [9].

The model explaining the operation of this circadian clock is from the article of K. Egushi *et al.* [4]. KaiC assembles into hexamers that can be in two states, tense or relaxed (fig. 6A).

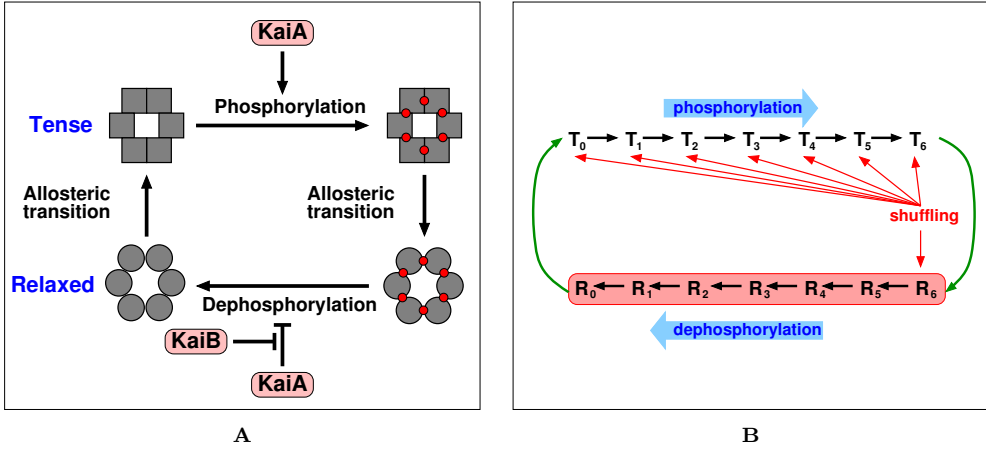


Fig. 6. **A:** KaiC hexamers state transitions and phosphorylation levels regulated by KaiA and KaiB. **B:** Model explaining the synchronised oscillations.

A very slow allosteric transition can switch the hexamers from one state to the other. The KaiA protein activates the phosphorylation of the KaiC hexamers in the tense state. In the relaxed state, KaiC hexamers tend to dephosphorylate. This dephosphorylation is counteracted by KaiA, which is itself inhibited by KaiB. The relative concentrations of the KaiA, KaiB and KaiC proteins leads to the phosphorylation in the tense state and to the dephosphorylation in the relaxed state of the KaiC hexamers.

In the *in vitro* experiments, the kinetics of phosphorylation and dephosphorylation of KaiC are constant and of the same order of magnitude (although variable *in vivo* due to the regulation of KaiA and KaiB). The proposed model shows a cycle traversed by each KaiC hexamer during which its phosphorylation level varies. The biological hypothesis proposed in the literature is that the monomer shuffling would be the main cause for the synchronisation of the phosphorylation level of all the KaiC hexamers leading to the stable periodic behaviour of the system (fig. 6B).

In the model, a molecule R_i (resp. T_i) stands for a relaxed (resp. tense) hexamers with i phosphorylations. The rules (depicted below) include the relaxed monomers shuffling, together with the shuffling of R_6 with tense monomers inducing the relaxation of the hexamer. The probabilities have been fixed to $P_s = 0.5$, $P_p = 10^{-5}$, $P_r = 10^{-8}$, $P_t = 5 \cdot 10^{-6}$.

$\forall i, j, k, l \in \{0, \dots, 6\}$ s.t. $k = i - 1, l = j + 1, R_i + R_j$	$\rightarrow R_k + R_l$	$[P_s]$
$\forall i \in \{0, \dots, 6\}, T_i + R_6$	$\rightarrow R_i + R_6$	$[P_s]$
$\forall i \in \{0, \dots, 5\}, T_i + R_6$	$\rightarrow R_{i+1} + R_5$	$[P_s]$
$\forall i \in \{0, \dots, 5\}, T_i$	$\rightarrow T_{i+1}$	$[P_p]$
$\forall i \in \{1, \dots, 6\}, R_i$	$\rightarrow R_{i-1}$	$[P_p]$
T_6	$\rightarrow R_6$	$[P_r]$
R_6	$\rightarrow T_6$	$[P_t]$

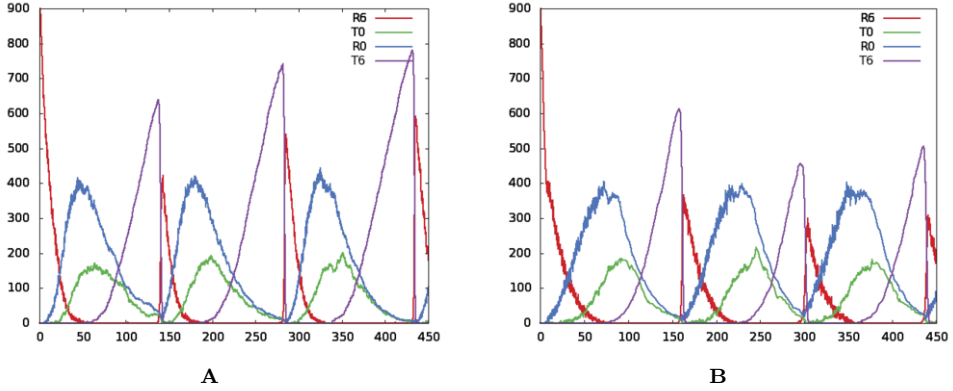


Fig. 7. **A:** simulation trace of KaiC model using HSIM SSA. **B:** using BIOCHAM SSA (tau-leap).

The execution of this model leads to a very large amount of fast reactions principally due to R_i monomer shuffling. Fig. 7A shows an output of a simulation obtained with HSIM. The simulation has been stopped after 1000 unit of times (only 450 shown in figures) and took approximatively 6s on a standard desktop computer. To give a comparison, the simulation with BIOCHAM [3] using the approximate tau-leap method (fig. 7B) took around 55s (and the exact SSA around 20min).

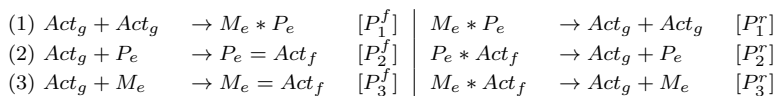
5.2 Actin strand polymerisation

In order to emphasize the importance of the spatial localisation of molecules, we present a simplified model of the polymerisation dynamics of filaments in a cell compartment. This example is derived from the actin cytoskeleton of eukaryotic cells.

The compartment is initially filled with globular actin, a protein that can self assemble to make a polarised dimer, then polymerise from each end of the initial dimer with kinetics depending on which end of the initial dimer the polymerisation process started from. The process is reversible, the polymer can shrink, releasing globular actin monomers, that can, after a certain amount of time, polymerise again. This mechanism needs energy, which is provided by the conversion of ATP to ADP; In this simplified model, we do not take it into account.

Actin polymerises into rectilinear filaments, which have a slow diffusion velocity. In the model we will assume they do not diffuse at all. So the orientation of the initial dimer defines the orientation of the resulting filament.

The model is defined with only three pairs of rules (" $M_A * M_B$ " means that the two molecules are bound, while " $M_A = M_B$ " means that the geometry of the assembly is rectilinear)



The first pair of rules is for the formation/dissociation of the dimer, the second pair is for the polymerisation/depolymerisation from the *plus* end, and the last pair is

the same for the *minus* end. To account for the favourable kinetics from the plus end the probabilities are such that: $P_2^f/P_2^r > P_3^f/P_3^r$. The diffusion velocities for all the elements of a filament (M_e , P_e and Act_f) are set to 0.

After a short period of time, the concentrations of free and bound actin reach an equilibrium and remain stationary. The surprise comes from the geometric arrangement of the filaments: they seem to become more or less aligned along the big axe of the compartment (fig. 8).

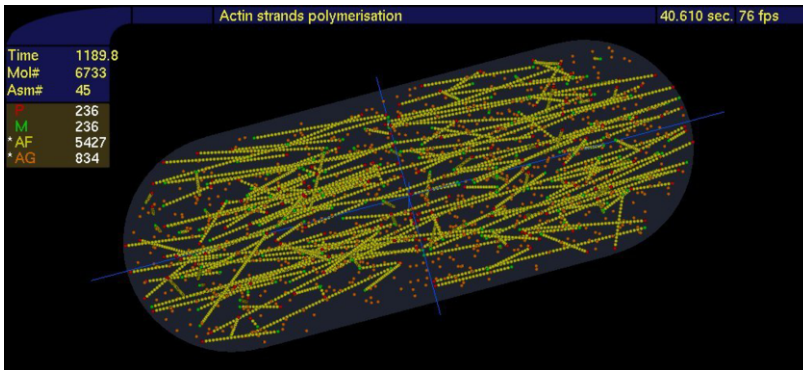


Fig. 8. Self alignment of actin filaments along the big axe of the main compartment.

This emerging phenomenon arises from the selective pressure exerted on the short filaments. The short filaments are more prone not to resist to a long period of depolymerisation, they simply disappear. Conversely the long filaments, in the same conditions may get shorter, but as they remain, their orientation also remains. Consequently, the filaments that remain are oriented where there is enough room to grow longer, along the big axe of the compartment.

Of course, without a system that takes into account the spatial localisation and the geometry of the molecular assemblies, we would not became aware of this phenomenon.

6 Conclusion

HSIM implements an hybrid simulation system integrating both an entity-centered system and an approximate Stochastic Simulation Algorithm (SSA). We have shown the advantages of an entity-centered simulation system in terms of expressiveness and notably, taking into account the spatial localisation. We have also proposed a new and very efficient approximate SSA to compute the evolution of a system of reaction rules.

We illustrated this method through the study of two very different biological models. The first one, where the spatial localisation was not needed, but involving a large number of reactions, have shown a nearly ten fold decrease in the computationnal time required by HSIM compared to standard approximate SSA. The second one, where conversely, the spatial localisation and the ability to take into account aggregates of molecules of HSIM, have been shown to be determinant for the discovering of an emergent phenomenon.

HSIM can be downloaded at <http://www.lri.fr/~pa/Hsim>.

References

- [1] Amar P. , Bernot G., Norris V., *HSIM: a simulation programme to study large assemblies of proteins*, Journal of Biological Physics and Chemistry **4** (2004), 79-84.
- [2] Amar P. , Legent G., Thellier M., Ripoll C., Bernot G., Nystrom T., Saier Jr M., Norris V., *A stochastic automaton shows how enzyme assemblies may contribute to metabolic efficiency*, BMC Systems Biology **2** (2008), 27.
- [3] Calzone L., Fages F., Soliman S., *BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge*, Bioinformatics, **22** (2006), 1805-1807
- [4] Eguchi K., Yoda M., Terada T.P., and Sasai M., *Mechanism of Robust Circadian Oscillation of KaiC Phosphorylation In Vitro*, Biophysical Journal **95**(4) (2008), 1773-1784.
- [5] Gillespie, Daniel T., *Exact Stochastic Simulation of Coupled Chemical Reactions*, The Journal of Physical Chemistry **81**(25) (1977), 2340-2361.
- [6] Gillespie, Daniel T., *Approximate accelerated stochastic simulation of chemically reacting systems*, The Journal of Physical Chemistry **115** (2001), 1716-1733.
- [7] Klann M., Ganguly A., and Koepl H., *Hybrid spatial Gillespie and particle tracking simulation*, BMC Bioinformatics **28** (2012), 18, pp. i549-i555.
- [8] Müller-Hill, Benno, *Repression of lac Promoter as a Function of Distance, Phase and Quality of an Auxiliary lac Operator*. J. Mol. Biol. **257** (1996), 21-29.
- [9] Nakajima M., Imai K., Ito H., Nishiwaki T., Murayama Y., Iwasaki H., Oyama T., and Kondo T., *Reconstitution of Circadian Oscillation of Cyanobacterial KaiC Phosphorylation in Vitro*, Science **308** (2005), 414-415.