Formal validation of asynchronous interaction-agents algorithms for reaction-diffusion problems

Pascal Redou, Kerdélo Sébastien, Gireg Desmeulles, Jean-François Abgrall, Vincent Rodin, Jacques Tisseau

▶ **To cite this version:**

Pascal Redou, Kerdélo Sébastien, Gireg Desmeulles, Jean-François Abgrall, Vincent Rodin, et al.. Formal validation of asynchronous interaction-agents algorithms for reaction-diffusion problems. PADS'07, 21st International Workshop on Principles of Advanced and Distributed Simulation, Jun 2007, San Diego, United States. pp.26-34. hal-00861413

HAL Id: hal-00861413
https://hal.science/hal-00861413

Submitted on 17 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Formal validation of asynchronous interaction-agents algorithms for reaction-diffusion problems

Pascal Redou, Sébastien Kerdélo[α], Gireg Desmeulles,
Jean-François Abgrall [β], Vincent Rodin, Jacques Tisseau
European Center for Virtual Reality, LISyC EA3883,
[α] Diagnostica STAGO
BP 38 , F-29280 Brest , France
[β] Hematology laboratory, Brest University Hospital, France
Pascal.Redou@enib.fr

Abstract

In the context of biological complex systems multi-agent simulation, we present an interaction-agent model for reaction-diffusion problems that enables interaction with the simulation during the execution, and we establish a mathematical validation for our model. We use two types of interaction-agents: on one hand, in a chemical reactor with no spatial dimension -e.g. a cell-, a reaction-agent represents an autonomous chemical reaction between several reactants, and modifies the concentration of reaction products. On the other hand, we use interface-agents in order to take into account the spatial dimension that appears with diffusion : interface-agents achieve the matching transfer of reactants between cells. This approach, where the simulation engine makes agents intervene in a chaotic and asynchronous way, is an alternative to the classical model - which is not relevant when the limits conditions are frequently modified- based on partial derivative equations. We enounciate convergence results for our interaction-agent methods, and illustrate our model with an example about coagulation inside a blood vessel.

1. Introduction

Algorithms for the numerical resolution of differential systems, though they give precise results, do not fit well with the study of complex systems [1]. which are *a priori* open (dynamical appearance/disappearance of components), heterogenous (various morphology and behaviours) and made of entities that are composite, mobile and distributed in space ; their number changes during time, and they interact with each other. Describing the evolution of such systems by means of deterministic methods like differential systems is uneasy, for limits conditions and number of processus fluctuate. As an alternative, the multi–agent approach [6, 20], already used in several biochemical models [11, 12, 19], provides a conceptual, methodological and experimental framework well-fitted for imagination, modelisation and experimentation of complexity. In this context, our work applies to the simulation of biological chemical kinetics phenomenons taking into account the variability of the number of implied reactants.

We present an interaction-agent model, dedicated to reaction-diffusion processes and based on two kinds of agents:

- In a dimensionless chemical reactor, a reaction-agent [15] represents a chemical reaction which loops into a perception/decision/action cycle : it reads the concentration of reactants, adapts its reaction speed, and modifies consequently the concentration of reaction products. Each agent independently executes a classical ordinary differential system algorithm [4]. For each of these classical methods, we build the matching reaction-agent method.

- In order to take into account spatial diffusion processes, we use an interface-agent between each pair of neighbor meshes for the transfer of chemical reactants according to the diffusion coefficient. We do not solve any partial derivative equation.

The simulation engine evolves interaction-agents asynchronously and chaotically (see section 2), in order to avoid the typical inflexibility of synchronous systems, as well as bias in numerical results.

Biochemical kinetics is a natural application context for our model: a classical example is given by cancer, since

chromosomic instability [9] implies on a regular basis modifications or creations of new reactions [2]. We have also used our reaction-agent model for simulation of MAPK pathway [14], and simulation of the extrinsic pathway of blood coagulation [13].

From a more general point of view, we set up agents autonomy as a basic principle [18] : this principle gives us the ability to interact with a running simulation, opening the path to a new way of experimenting : the *in virtuo* experimentation [17]. *In virtuo* experimentation makes it possible to interfere with a model by adding or removing reactants, as well as interaction between reactants. The main interest of such an experimentation is that these alterations are possible without having to stop the progress of the simulation : experimental conditions of the *in virtuo* way are therefore very close to the *in vivo* and *in vitro* ones, and fundamentally different from the *in silico* one.

In section 2 of this paper, we present what we call chaotic and asynchronous iterations, that is, the way we make our interaction-agents intervene. In section 3 we describe our reaction-agent model for numerical computation of differential systems for chemical kinetics inside a cell, so as our formal results of convergence. In section 4 we describe our interface-agent model, dedicated to diffusion process. We also formalize our model and state the main results about convergence. Section 5 shows an illustrating example of our approach for a blood circulation simulation inside a blood vessel. For the sake of concision, we will not expose demonstrations of mathematical results (which are not conjectures but proved results). Please contact first author to obtain proofs.

2. Chaotic and Asynchronous iterations

In this short section, we describe precisely the way our interaction-agents intervene during the simulation, what we name chaotic and asynchronous iterations.

Each agent has to perform a perception/decision/action cycle (see section 3 for the case of reaction-agent and section 4 for the case of interface-agent). At each step, the scheduler [10] makes one interaction-agent carry out its perception/decision/action cycle. Interaction-agents act one after the other following the scheduler cycle whose length equals the number of agents. Interaction-agents each act once and only once in a sheduler cycle, but the order in which they do so is randomly chosen. Let's precise these notions :

- **Asynchronous iterations** : a fundamental statement is that in the classical approach, time discretisation induces the hypothesis that all reactions occur simultaneously during the same time-step. Indeed, classically used differencial systems numerical resolution algorithms *a priori* do this hypothesis based upon the choice of infinitesimal time-step. *A contrario*, interaction-agent model does the asynchronic hypothesis for chemical reactions. We claim that this hypothesis is not only more realistic, but moreover allows the user to interfere at runtime with the reactions by adding or removing a reaction-agent, at any time of the simulation. Time is then divided into scheduler cycles inside of which each interaction-agent acts once and only once, considering the state of the system at the moment it acts. From a physical point of view, each scheduler cycle corresponds to one time-step of the classical approach.

- **Chaotic iterations** : an unalterable arrangement for interaction-agents operations at each cycle might introduce a bias -we proved mathematical results that confirm it- in the simulation. In order to avoid this bias the scheduler makes each reaction-agent operate in a random order, which changes for each iteration step. This is what we call chaotic iterations.

Figure 1 illustrates this scheduling strategy.

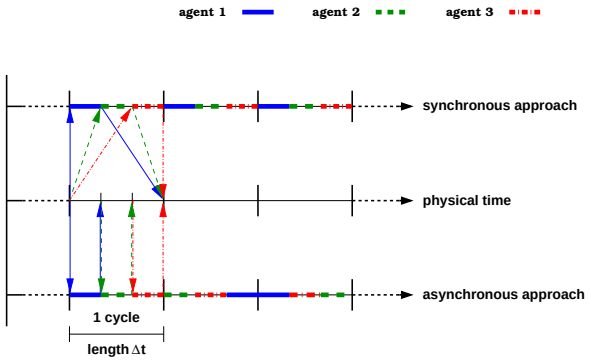

Figure 1. Classical and reaction-agent points of view for reactions scheduling. Case of 3 **interaction-agents** $r_i,\ 1 \leq i \leq 3$**.**

We now describe our reaction-agent model dedicated to biochemical reactions inside a cell.

3. Reaction-agent

3.1. Principle

The reaction-agents based methods are numerical methods for computation of differential systems which permit to take into account, at runtime, the evolvingness of these systems Each reaction-agent matches a reaction of the system

we want to modelize. Each agent behaviour loops in the following cycle:

- **Perception** : sensing of concentration of all reactions components (*i.e.* reactants and products),
- **Decision** : computation of the amount of consumed reactants (and thus of the amount of formed products),
- **Action** : writing the new concentrations of the reaction components.

Reaction-agents act by the way of chaotic and asynchronous iterations, as described in section 2.

3.2. Illustration

Consider a medium with no spatial dimension containing several reactants. Let $[\vec{C}(t)]$ be the concentrations vector at instant t. In this medium m chemical reactions occur. Their respective speeds are given by vectorial functions $f_i,\ 1 \leq i \leq m$, whose arguments are time and concentrations vector. The evolution in time of reactants concentrations are classically described by the differential system

$$\frac{d}{dt}[\vec{C}(t)] = (f_1 + f_2 + \cdots + f_m)(t, \vec{C}[t]), \tag{1}$$

under conditions $\vec{C}[t_0]$ for concentrations at initial instant. Such systems are numericaly solved by the mean of very precises algorithms [4, 7], which allows computation of all concentrations at each instant of the discretised time : for one step methods, the concentrations vector $\vec{C}_{n+1}$ at instant t_{n+1} is computed from the same vector at instant t_n, named $\vec{C}_n$. This leads to a computation algorithm such as below :

$$\begin{array}{rcl} \vec{C}_0 & = & \vec{C}[t_0] \\ \vec{C}_{n+1} & = & \vec{C}_n + h_n \Phi_{f_1+\cdots+f_m}(t_n, \vec{C}_n, h_n) \end{array} \tag{2}$$

where $h_n = t_{n+1} - t_n$, $\Phi_{f_1+\cdots+f_m}$ is a function dependent on the sum of f_i speeds, and which characterizes the chosen algorithm. As we stated, here reactions are supposed to be simultaneous and the main drawback of this modelisation is its staticness : adding or removing a reaction at runtime implies rewriting the system and reruning the program, which is unsuitable for complex system simulation and runtime modification of these systems. Our method also uses a classical resolution algorithm but applies it for each reaction during the same time-step. Let's consider an elementary example with two reactions, whose speeds are f_1 and f_2. As an alternative to the numerical computation of the system (1) (when $m = 2$) using algorithm (2), that is,

$$\vec{C}_{n+1} = \vec{C}_n + h_n \Phi_{f_1+f_2}(t_n, \vec{C}_n, h_n), \tag{3}$$

we propose a *reaction-agent version* of this algorithm :

$$\begin{array}{rcl} \vec{C}_\star & = & \vec{C}_n + h_n \Phi_{f_1}(t_n, \vec{C}_n, h_n) \\ \vec{C}_{n+1} & = & \vec{C}_\star + h_n \Phi_{f_2}(t_n, \vec{C}_\star, h_n) \end{array} \tag{4}$$

or, equiprobably,

$$\begin{array}{rcl} \vec{C}_\star & = & C_n + h_n \Phi_{f_2}(t_n, \vec{C}_n, h_n) \\ \vec{C}_{n+1} & = & C_\star + h_n \Phi_{f_1}(t_n, \vec{C}_\star, h_n) \end{array} \tag{5}$$

Thus, in a single time-step, the algorithm is here applied two times : once for each reaction. Each application takes into account the state of the system at the current time. In order to avoid bias, at each time step a random arrangement of reaction-agents operations is performed.

3.3. Formalization and principal results

We now give the mathematical formalization of our reaction-agent model, and the validating results we have obtained. The natural integers ring is called $\mathbb{N}$, $\mathbb{R}$ is the reals field, and S_m the permutations of order m group [3]. For the sake of simplicity we only consider differential systems of a single equation; however definitions and results are easily generalizable. More details about numerical resolution of ordinary differential equations can be found in [7].

Remark 3.1. *We have also adapted this autonomous agents point of view for classical multiple steps methods, or for implicits methods [8]. Convergence and stability features are better for these methods than for single step methods. However these methods not only conflict with principles of multi agents systems whose behaviour is markovian; but moreover they rule out the ability to modify the number of agents at runtime.*

General definition

Definition 3.2. *Let*

$$y_{n+1} = y_n + h_n \Phi_f(t_n, y_n, h_n) \tag{6}$$

be a one step method for Cauchy problem resolution

$$\left\{ \begin{array}{l} y(t_0) = y_0 \\ y'(t) = f(t, y(t)). \end{array} \right. \tag{7}$$

Let $m \in \mathbb{N}^*$. *We call* reaction-agent version of method (6), *for resolution of problem*

$$\left\{ \begin{array}{l} y'(t) = (f_1 + f_2 + \cdots + f_m)(t, y(t)) \\ y(t_0) = y_0 \end{array} \right. \tag{8}$$

the method given by

$$y_{n+1} = y_n + h_n \Phi_{\sigma_n}(t_n, y_n, h_n) \tag{9}$$

defined by an equiprobable choice, at each time step $n \to n+1$, of $\sigma_n \in S_m$, and by relations

$$\begin{array}{rcl} y_{\star 1} & = & y_n + h_n \Phi_{f_{\sigma_n(1)}}(t_n, y_n, h_n) \\ \forall i,\ 1 \leq i \leq m-1, & & \\ y_{\star i+1} & = & y_{\star i} + h_n \Phi_{f_{\sigma_n(i+1)}}(t_n, y_{\star i}, h_n) \\ y_{n+1} & = & y_{\star m} \end{array} \tag{10}$$

Example 3.3. *We remind the reader that for Cauchy problem resolution (7), order* 2 *Runge-Kutta method is given by*

$$y_{n+1} = y_n + h_n \Phi_f(t_n, y_n, h_n)$$

where

$$\Phi_f(t,y,h) = f(t + \frac{h}{2}, y + \frac{h}{2} f(t,y)).$$

The matching reaction-agent version for resolution of problem (8) is given by definition 3.2, where $\forall i$, $1 \leq i \leq m$,

$$\Phi_{f_i}(t,y,h) = f_i(t + \frac{h}{2}, y + \frac{h}{2} f(t,y)).$$

For instance, two reaction-agents case leads to

$$y_{n+1} = y_n + h_n \Phi_{\sigma_n}(t_n, y_n, h_n)$$

with, equiprobably,

$$\begin{array}{l} \Phi_{\sigma_n}(t,y,h) \\ = f_1(t + \frac{h}{2}, y + \frac{h}{2} f_1(t,y)) \\ + f_2\Big(t + \frac{h}{2}, y + h f_1(t + \frac{h}{2}, y + \frac{h}{2} f_1(t,y)) \\ \qquad + \frac{h}{2} f_2(t, y + h f_1(t + \frac{h}{2}, y + \frac{h}{2} f_1(t,y)))\Big) \\ \text{if } \sigma_n = \mathrm{Id} \text{ (identitymap)} \end{array}$$

or

$$\begin{array}{l} \Phi_{\sigma_n}(t,y,h) \\ = f_2(t + \frac{h}{2}, y + \frac{h}{2} f_2(t,y)) \\ + f_1\Big(t + \frac{h}{2}, y + h f_2(t + \frac{h}{2}, y + \frac{h}{2} f_2(t,y)) \\ \qquad + \frac{h}{2} f_1(t, y + h f_2(t + \frac{h}{2}, y + \frac{h}{2} f_2(t,y)))\Big) \\ \text{if } \sigma_n(1) = 2. \end{array}$$

Average order of a reaction-agent method Acoording to definition 3.2, the computation of y_{n+1} in function of y_n depends upon the choice of the permutation σ_n. Thus we have to keep this in mind to characterize the convergence. With the same notations as above, the average evolution on one step is given by

$$\begin{array}{rcl} y_{n+1} & = & y_n + \bar{\Phi}(t_n, y_n, h_n), \\ \bar{\Phi} & = & \dfrac{1}{m!} \displaystyle\sum_{\sigma_n \in S_m} \Phi_{\sigma_n} \end{array} \tag{11}$$

Definition 3.4. *The order (in the usual sense) of the method given by (11) is called the* average order *of the method given by definition 3.2.*

Main results We enounciate here our main results about convergence of reaction-agent methods.

Theorem 3.5. *1. Reaction-agent version of Euler's method is convergent of average order* 1.

2. *Reaction-agent version of order* 2 *Runge-Kutta method is convergent of average order* 2.

3. *Consider a one step method, convergent of order $p \geq 3$. Thus its reaction-agent version is convergent of average order* 2.

Theorem 3.5 claims in substance that there is no point in using reaction-agent's version of a Runge-Kutta method of order ≥ 3.

Of course, because of asynchronism, the efficiency of our reaction-agent model is much weaker than the one of classical integration methods . However, we stress again the point that it is the only model -to our knowledge- that enables *in virtuo* experimentation for biochemical reactions.

4. Interface-agent

4.1. Diffusion equations and classical approach

The diffusion equation is a partial differential equation, which describes the density fluctuations in a material undergoing diffusion.

The equation is usually written as:

$$\frac{\partial X}{\partial t} = \nabla \cdot D(X) \nabla X(\vec{r}, t) \tag{12}$$

where X is the density of the diffusing material, t is time, D is the collective diffusion coefficient, $\vec{r}$ is the spatial coordinate and the nabla symbol ∇ represents the vector differential operator. If the diffusion coefficient depends on the density then the equation is nonlinear; if D is a constant, however, then the equation reduces to the following linear equation:

$$\frac{\partial X}{\partial t} = D \Delta X(\vec{r}, t), \tag{13}$$

where Δ is the Laplace operator. This equation is also called the heat equation.

In the following, we first describe the finite difference method for the resolution of equation (13), since we will compare our interface-agent model to this method to provide proofs of convergence.

For the sake of simplicity, we place ourselves in the one dimensional case (though our results can be extended to the multi-dimensional case), and equation (13) can be written

$$\frac{\partial X}{\partial t} = D \frac{\partial^2 X}{\partial x^2}. \tag{14}$$

where x is the space parameter. In order to be exhaustive, we consider the finite case $0 < x < L$, with the bounding limits :

$$X(0) = X_0, \quad \frac{\partial X}{\partial x} \mid_{x=L} = 0. \tag{15}$$

Space-time discretization The finite difference method uses a space discretization

$$[0,L] = \bigcup_{m=0}^{M-1} [x_m, x_m + h],\ x_m = mh,\ h = L/M, \tag{16}$$

so as a time discretization :

$$[0,t_{max}] = \bigcup_{n=0}^{N-1} [t_n, t_n + \delta t],\ t_n = n\delta t,\ \delta t = t_{max}/N.$$

Values of $X(x,t)$ in points (x_m,t_n) will be denoted

$$X_m^n = X(x_m,t_n)$$

Numerical scheme Approaching derivation operators with first order Taylor formula, we get the discretization of (14) :

$$\frac{X_m^{n+1} - X_m^n}{\delta t} - D\frac{X_{m+1}^n - 2X_m^n + X_{m-1}^n}{h^2} = 0,\ m = 1,\ldots,M-1.$$

Since X_m^0 is known, one can compute X_m^{n+1} by means of relation

$$X_m^{n+1} = X_m^n + D\frac{\delta t}{h^2}(X_{m+1}^n - 2X_m^n + X_{m-1}^n). \tag{17}$$

The vector form of relation (17), which gives the state of the system at time $n+1$ in function of its state at time n, is

$$X^{n+1} = \mathcal{H} \cdot X^n,$$

where $\mathcal{H}$ is the matrix

$$\mathcal{H} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \omega & 1-2\omega & \omega & 0 & \cdots & 0 & 0 \\ 0 & \omega & 1-2\omega & \omega & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & & \\ \vdots & & & \ddots & \ddots & \ddots & \\ 0 & 0 & \cdots & 0 & \omega & 1-2\omega & \omega \\ 0 & 0 & \cdots & 0 & \omega & 1-2\omega & \omega \end{bmatrix} \tag{18}$$

Remark 4.1. *Here, and in the following, we put* $\omega = D\frac{\delta t}{h^2}$

4.2. Interface agent model

General principle Our interface-agent approach is basically different, since we do not solve any partial derivative equation. The system evolves by means of agents acting cyclically, as described below :

1. The system is discretized as in (16).

2. In the middle of each interval $[x_i, x_{i+1}]$ is placed an interface-agent A_{i+1}^i, which, when it intervenes, has to update the values of X in x_i and x_{i+1}. Thus, if A_{i+1}^i intervenes at time j, it computes X_i^j and X_{i+1}^j, taking into account the diffusion process. *Values in other points are unchanged.*

3. Interface-agents intervene a chaotic and asynchronous way, as defined in section 2. Time is divided into scheduler cycles, each one containing M time steps, whith each interface-agent operating once and only once, in a random order.

4. In order to compare our model to the finite difference method, we will consider the state of the system at a moment j which is a multiple of M, so that values of X in each point have been updated the same number of times.

5. We will extend our results to the case, where agents can stop operating for a while, and do not not inevitably act in each scheduler cycle (see theorem 4.6).

Action of an interface-agent We now describe more precisely the action of A_{i+1}^i, which separates two meshes with the same length h, denoted C_i and C_{i+1}, in which are computed X_i and X_{i+1}.

Suppose A_{i+1}^i intervenes at time j. It updates the values X_i and X_{i+1} according to the following linear equations, which represent a discretization of Fick's Law :

$$\left| \begin{array}{lcl} X_i^{j+1} & = & X_i^j + \omega(X_{i+1}^j - X_i^j) \\ X_{i+1}^{j+1} & = & X_{i+1}^j - \omega(X_{i+1}^j - X_i^j) \\ X_k^{j+1} & = & X_k^j,\ j \notin \{i, i+1\}. \end{array} \right. \tag{19}$$

Note that A_1^0 operates the following way :

$$\left| \begin{array}{lcl} X_0^{j+1} & = & X_0^j \\ X_1^{j+1} & = & X_1^j - \omega(X_1^j - X_0^j) \\ X_k^{j+1} & = & X_k^j,\ k \geq 2 \end{array} \right. \tag{20}$$

and as regards A_M^{M-1}, we have :

$$\left|\begin{array}{lcl} X_i^{j+1} & = & X_i^j,\ 0 \le i \le M-2 \\ X_{M-1}^{j+1} & = & (1-\omega)X_{M-1}^j + \omega X_M^j \\ X_M^{j+1} & = & X_{M-1}^j. \end{array}\right. \tag{21}$$

We denote $\mathcal{M}_{i,i+1}$ the matrix which matches with the action of A_{i+1}^i. For instance, for $1 \le i \le M-2$,

$$\mathcal{M}_{i,i+1} = \begin{bmatrix} 1 & 0 & & \cdots & & 0 \\ 0 & \ddots & & & & \vdots \\ & & 1-\omega & \omega & & \\ & & \omega & 1-\omega & 0 & 0 \\ \vdots & & & & \ddots & \\ 0 & & \cdots & & 0 & 1 \end{bmatrix} \tag{22}$$

Description of a scheduler cycle Let us consider the following example : at moment nM, we suppose that a cycle begins, in which interface-agents intervene in the precise order A_1^0, A_2^1, ..., A_M^{M-1}. Applying equations (20, 19, 21) one after the other in this order, the state of the system at moment $(n+l)M$ is given by

$$X^{(n+1)M} = \mathcal{M}_{M-1,M} \cdots \mathcal{M}_{1,2} \cdot \mathcal{M}_{0,1} \cdot X^{nM},$$

For instance, if $M = 4$, with this choice of order,

$$X^{(n+1)M} = \mathcal{L} \cdot X^{nM},$$

where

$$\mathcal{L} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ (1-\omega)\omega & (1-\omega)^2 & \omega & 0 & 0 \\ (1-\omega)\omega^2 & (1-\omega)^2\omega & (1-\omega)^2 & \omega & 0 \\ (1-\omega)\omega^3 & (1-\omega)^2\omega^2 & (1-\omega)^2\omega & (1-\omega)^2 & \omega \\ (1-\omega)\omega^3 & (1-\omega)^2\omega^2 & (1-\omega)^2\omega & (1-\omega)^2 & \omega \end{bmatrix}$$

Remark 4.2. *Remark the essential fact that if we develop the coefficients of $\mathcal{L}$ and only keep terms of order ≤ 1 we get*

$$\mathcal{L} \simeq \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \omega & 1-2\omega & \omega & 0 & 0 \\ 0 & \omega & 1-2\omega & \omega & 0 \\ 0 & 0 & \omega & 1-2\omega & \omega \\ 0 & 0 & \omega & 1-2\omega & \omega \end{bmatrix},$$

which is precisely the operator $\mathcal{H}$ obtained in the finite difference numerical scheme (equation (18)). We are going to generalize this observation.

4.3. Main results

We now enounciate the mathematical results that validate our interface-agent model. Proofs of results of convergence are based upon linear algebra methods ([16]).

The main result is the following : given an unspecified vector $X^0 = (X_0^0, X_1^0, \ldots, X_M^0)$, its image by a composition of scheduler cycles, each randomly ordered, admits for limit, when the number of cycles tends to infinity, the equilibrium state $\bar{X} = (X_0^0, X_0^0, \ldots, X_0^0)$. This means . that the repartition of X balances with the source value X_0^0.

Notations A scheduler cycle is characterized by an endomorphism on $\mathbb{R}^{M+1}$, compounded with M applications $f_1, f_2, \ldots, f_M$. The application f_i traduces the action of the interface-agent A_i^{i-1} and is represented, in the canonical basis of $\mathbb{R}^{M+1}$, by the matrix $\mathcal{M}_{i-1,i}$. Vectors of the canonical basis are denoted by e_i, $1 \le i \le M+1$. Recall that S_m is the permutations of order m group. The following application defines a scheduler cycle corresponding to $\sigma \in S_{M+1}$:

$$\varphi_\sigma = f_{\sigma(M+1)} \circ f_{\sigma(M)} \circ \cdots \circ f_{\sigma(1)}, \quad \sigma \in S_{M+1}.$$

Finally, we put $\bar{X} = (1, \ldots, 1)$.

The following proposition gives properties of applications f_i that will lead to results of convergence :

Proposition 4.3. *1. The application f_1 (to which is associated $\mathcal{M}_{0,1}$) is diagonalizable. Its eigenvalues are 1 with multipicity M, and $1-\omega$ with multiplicity 1. For this application :*

- *The eigenspace associated to the eigenvalue 1 is generated by $e_1 + e_2, e_3, \ldots, e_{M+1}$.*
- *The eigenspace associated to the eigenvalue $1-\omega$ is generated by e_2.*

2. For all i such that $2 \le i \le M-1$: the application f_i is diagonalizable. Its eigenvalues are 1 with multipicity M, and $1-2\omega$ with multiplicity 1. For this application f_i :

- *The eigenspace associated to the eigenvalue 1 is generated by $e_i + e_{i+1}, e_j,\ j \notin \{i, i+1\}$.*
- *The eigenspace associated to the eigenvalue $1-2\omega$ is generated by $-e_i + e_{i+1}$.*

3. The application f_M (to which is associated $\mathcal{M}_{M-1,M}$) is diagonalizable. Its eigenvalues are 1 with multipicity M, and 0 with multiplicity 1. For this application :

- *The eigenspace associated to the eigenvalue 1 is generated by $e_M + e_{M+1}, e_j,\ j \notin \{M, M+1\}$.*
- *The eigenspace associated to the eigenvalue 0 is generated by $\frac{\omega}{\omega-1} e_M + e_{M+1}$.*

We deduce from this proposition our main result of convergence :

Theorem 4.4. *For all sequence $(\sigma_k)_{k\in\mathbb{N}}$ of elements in S_{M+1}, and fo all X in $\mathbb{R}^{M+1}$, we have*

$$\lim_{p\to\infty} \varphi_{\sigma_p} \circ \varphi_{\sigma_{p-1}} \circ \cdots \circ \varphi_{\sigma_1}(X) = x_1 \bar{X},$$

where x_1 is the first component of X.

Remark 4.5. *The main task of the proof of theorem 4.4 is to show that each φ_σ is a contraction, and $\bar{X}$ is a fixed point for φ_σ.*

Proof of theorem 4.4 can be extended to the case, where a scheduler cycle (in which each interface-agent acts once and only once) is replaced by a sequence in which each interface-agent acts **at least** once. This leads to the following general result :

Theorem 4.6. *We keep the same notations. Let $(\tilde{f}_n)_{n\in\mathbb{N}^*}$ be a sequence of elements (not necessarily different) in the set $\{f_1, f_2, \ldots, f_{M+1}\}$. Let Γ_k be the space of applications of the type $\tilde{f}_k \circ \tilde{f}_{k-1} \circ \cdots \circ \tilde{f}_1$. We denote by $N_i(\Phi_k)$ the number of interventions of f_i in $\Phi_k \in \Gamma_k$. Suppose*

$$\lim_{k\to\infty} \min_{\Phi_k\in\Gamma_k} (N_i(\Phi_k)) = +\infty.$$

Thus,

$$\forall X = (x_1, x_2, \ldots, x_{M+1}) \in \mathbb{R}^{M+1}, \forall (\Phi_k)_{k\in\mathbb{N}^*},$$
$$\lim_{k\to\infty} (\Phi_k(X)) = x_1 \bar{X}.$$

This extends convergence of our interface-agent model to the case, where agents can stop operating for a while, and do not not inevitably act in each scheduler cycle.

4.4. Convergence speed : comparing with finite difference method

We have established convergence results for our interface-agent model. We now compare its convergence order with the one of finite difference method. To this end, we compare consistency errors for both methods. Basic definitions in numerical analysis can be found in [4]. Recall that consistency error for the finite difference method is given by

$$e_m^{n+1} = \frac{X_m^{n+1} - \mathcal{H} X_m^n}{\delta t}$$

where $X^n = (X_0^n, X_1^n, \ldots, X_M^n)$ is the solution vector at time n, and $\mathcal{H}$ is given by (18). Thus, we have

$$\begin{aligned} e_m^{n+1} &= \frac{1}{\delta t}\left(X_m^{n+1} - X_m^n - \omega(X_{m+1}^n - 2X_m^n + X_{m-1}^n)\right) \\ &= \frac{\partial X}{\partial t}(x_m, t_n) - \frac{h^2}{\delta t}\omega \frac{\partial^2 X}{\partial x^2}(x_m, t_n) + O(\delta t) + O(h^2) \\ &= O(\delta t) + O(h^2), \end{aligned} \tag{23}$$

since X satisfies the heat equation (13).

Equations (23) show that the finite difference method has order 1 in time and order 2 in space.

We establish that the efficiency of our interface-agent method is exactly the same:

Theorem 4.7. *Let $\varepsilon^{(n+1)M}$ be the consistency error vector for the interface-agent method at the end of the $(n+1)$-th scheduler cycle.*

We have :

$$\left|\begin{array}{l} \varepsilon_0^{(n+1)M} = 0 \\ \forall m,\ 1 \le m \le M-1, \\ \qquad \varepsilon_m^{(n+1)M} = O(\delta t) + O(h^2) + 3\frac{K}{\delta t}\omega^2, \\ \qquad K \le \max_{0\le i\le M-1}(|X_{i+1}^{nM} - X_i^{nM}|). \end{array}\right. \tag{24}$$

Remark 4.8. Interpretation. *We first stress the point that theorem 4.4 implies that all values $|X_{i+1}^{nM} - X_i^{nM}|$ tend to 0 if $n \to \infty$. Thus, the coefficient of ω^2 decreases in time. However, in order to avoid a significant error at the first time steps, the term $\omega^2/\delta t$ has to be negligible in front of h^2, which is equivalent to the condition*

$$\delta t < h^6$$

for the choice of time and space steps.

Finally, section 5 presents an example in which we use reaction-agents and interface-agents in the context of hemostasis simulation.

5. Example

We are interested in the modelisation of hemostasis [5] and especially in the construction of a virtual 3D blood vessel in which it will be possible to simulate and study the blood coagulation process. We show here a first attempt of this work, by means of interaction-agent method.

The blood vessel is represented by a cylinder discretized into meshes, see figure 2. In each mesh, a numerical model of the blood coagulation cascade can be found ; this model presents almost all of the enzymatic reactions involved in the extrinsic and intrinsic pathways, such as the tenase or the prothrombinase reactions. The different biochemical species can also diffuse from one mesh to a neighboring one following Fick's law as illustrated on figure 2. The overall model includes 32 species that interact *via* 40 biochemical reactions and diffusion.

This model can be simulated using a PDE-based approach and our multi-agent-system-based approach. In order to validate our methodology, we have simulated the model using both approaches, and compared the curves of

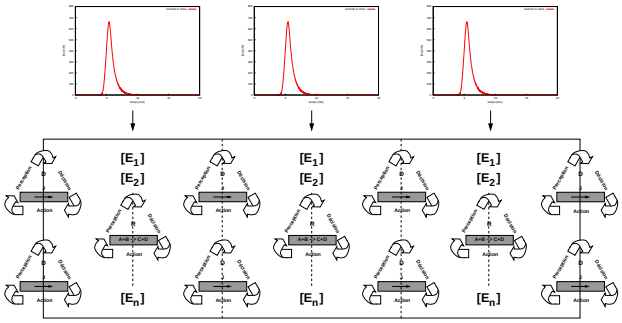

Figure 2. Numerical model of a virtual blood vessel. The blood vessel is represented by a cylinder discretized into meshes. Each mesh contains a numerical model of the blood coagulation enzymatic cascade, and each specie can diffuse from a mesh to another by means of an interface-agent.

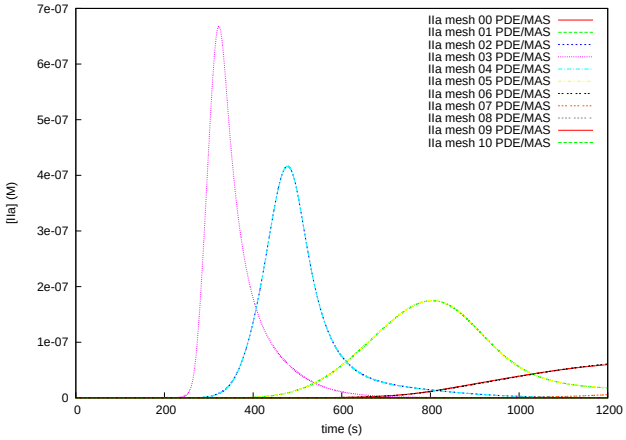

Figure 3. Comparison of numerical solutions obtained by the PDE-based approach and the interaction-agent method (MAS). The virtual blood vessel is composed of 11 meshes. In each mesh, the thrombin generation curve obtained with the interaction-agent method is compared to the thrombin generation curve obtained with PDE-based approach.

thrombin[1] generation obtained in both cases ; results are plotted in figure 3. In each mesh, we have obtained a thrombin generation curve, and for each approach these have been compared. In each case, the curves are overlapping, which confirms that the PDE-based approach and the interaction-agent method converge to the same solution.

6. Conclusion and perspectives

We have exposed the proof of efficiency of our interaction-agent model for the *in virtuo* simulation of reaction-diffusion phenomena. Due to their asynchronism, our algorithms are of course weaker than classical ones for the resolution of partial derivative systems. Nevertheless, as far as we know, interaction-agent model is the only one which enables a real dynamical interaction with the simulation, without stopping it. In the future, we plan to study the same way advection phenomena, in order to be able to make assuptions, by means of a complete model of virtual blood vessel, about unexplained phenomena of hemostasis.

References

[1] H. Atlan. The living cell as a paradigm for complex natural systems. *ComplexUs*, 1:1–3, 2003.

[2] J. L. Bos. Ras oncogene in human cancer : a review. *Cancer Research*, 50:1352–1361, 1989.

[3] P. Cameron. *Permutation Groups*. New York : Cambridge University Press, 1999.

[1]Thrombin is the key enzyme of the blood coagulation process, that's why we have focused on it.

[4] P. G. Ciarlet and J. L. Lions. *Handbook of Numerical Analysis*. North Holland, 1990.

[5] R. W. Colman, J. Hirsh, V. J. Marder, C. A. W., and J. N. George, editors. *Hemostasis and thrombosis. Basic principles and clinical practice*. Lippincott williams & wilkins, fourth edition, 2001.

[6] J. Ferber. *Multi-agent systems : An introduction to distributed artificial intelligence*. Addison Wesley, 1999.

[7] E. Hairer, S. P. Norsett, and P. Wanner. *Solving Ordinary Differential Equations I. Nonstiff Problems*. Springer Series in Comput. Mathematics, 1983.

[8] E. Hairer, S. P. Norsett, and P. Wanner. *Solving Ordinary Differential Equations II. Stiff and differential-algebraic problems*. Springer Series in Comput. Mathematics. Springer, 1996.

[9] D. Hanahan and J. A. Weinberg. The hallmarks of cancer. *Cell*, 100:57–70, 2000.

[10] F. Harrouet, J. Tisseau, P. Reignier, and P. Chevaillier. oris : un environnement de simulation interactive multi-agents. *RSTI-TSI*, 21(4):499–524, 2002.

[11] Y. Huang, X. Xiang, G. Madey, and S. E. Cabaniss. Agent-based scientific simulation. *Computing in Science and Engineering*, 07(1):22–29, 2005.

[12] C. M. Jonker, J. L. Snoep, J. Treur, H. V. Westerhoff, and W. C. A. Wijngaards. Bdi-modelling of intracellular dynamics. In *In A.B. Williams and K. Decker (eds.), Proceedings of the First International Workshop on Bioinformatics and Multi-Agent Systems,BIXMAS'02*, pages 15–23, 2002.

[13] G. Lu, G. J. Broze, and S. Krishnaswamy. Formation of factors ixa and xa by the extrinsic pathway. differential regulation by tissue factor pathway inhibitor and antithrombin iii.

The journal of biological chemistry, 279(17):17241–17249, 2004.

[14] G. Querrec, V. Rodin, J. F. Abgrall, S. Kerdelo, and J. Tisseau. Uses of multiagents systems for simulation of mapk pathway. In *In Third IEEE Symposium on BioInformatics and BioEngineering*, pages 421–425, 2003.

[15] P. Redou, S. Kerdélo, C. Le Gal, G. Querrec, V. Rodin, J.-F. Abgrall, and J. Tisseau. Reaction-agents: first mathematical validation of a multi-agent system for dynamical biochemical kinetics. In *Progress in artificial intelligence (EPIA 2005)*, volume 3808 of *Lecture notes in artificial intelligence*, pages 156–166. Springer, 2005.

[16] S. Roman. *Advanced linear algebra*. Graduate texts in mathematics. New York, *Springer*, 1992.

[17] J. Tisseau. Virtual reality -in virtuo autonomy-. accreditation to Direct Research, 2001.

[18] J. Tisseau and F. Harrouet. Autonomie des entités virtuelles. *in* Le traité de la réalité virtuelle, 2nd edition,vol.2, 2003.

[19] K. Webb and T. White. Cell modeling using agent-based formalisms. In *AAMAS'04*, pages 25–29, July 2004.

[20] M. Wooldridge and P. Ciancarini. *Agent-Oriented Software Engineering: The State of the Art*. Springer-Verlag Lecture Notes in AI. P. Ciancarini and M. Wooldridge, editors, Agent-Oriented Software Engineering, 2001.