HAL
open science

SIMULATION OF HUMAN ISCHEMIC STROKE IN REALISTIC 3D GEOMETRY: A NUMERICAL STRATEGY

Thierry Dumont, Max Duarte, Stéphane Descombes, Marie-Aimée Dronne, Marc Massot, Violaine Louvet

To cite this version:

Thierry Dumont, Max Duarte, Stéphane Descombes, Marie-Aimée Dronne, Marc Massot, et al.. SIMULATION OF HUMAN ISCHEMIC STROKE IN REALISTIC 3D GEOMETRY: A NUMERICAL STRATEGY. 2010. hal-00546223v1

HAL Id: hal-00546223

https://hal.science/hal-00546223v1

Preprint submitted on 14 Dec 2010 (v1), last revised 16 May 2012 (v3)

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.

SIMULATION OF HUMAN ISCHEMIC STROKE IN REALISTIC 3D GEOMETRY: A NUMERICAL STRATEGY

THIERRY DUMONT, MAX DUARTE, STÉPHANE DESCOMBES, MARIE-AIMÉE DRONNE, MARC MASSOT, AND VIOLAINE LOUVET

Abstract. In silico research in medicine is thought to reduce the need for expensive clinical trials under the condition of reliable mathematical models and accurate and efficient numerical methods. In the present work, we tackle the numerical simulation of reaction-diffusion equations modeling human ischemic stroke. This problem induces peculiar difficulties like potentially large stiffness which stem from the broad spectrum of temporal scales in the nonlinear chemical source term as well as from the presence of steep spatial gradients in the reaction fronts, spatially very localized. Furthermore, simulations on realistic 3D geometries are mandatory in order to describe correctly this type of phenomenon. We introduce a new resolution strategy based mainly on time operator splitting and a well conceived parallelization strategy, that takes into account complex geometries. We consider then a high order implicit time integration of the reaction and an explicit one for the diffusion term in order to build a time operator splitting scheme that exploits efficiently the special features of each problem. Thus, we aim at solving complete and realistic models including all time and space scales with conventional computing resources, that is on a reasonably powerful workstation. Consequently, 2D and also fully 3D numerical simulations of ischemic strokes for a realistic brain geometry, are conducted for the first time and shown to reproduce the dynamics observed on MR images in stroke patients. Beyond this major step, in order to improve accuracy and computational efficiency of the simulations, we indicate how the present numerical strategy can be coupled to spatial adaptive multiresolution schemes. Preliminary results in the framework of simple geometries allow to assess the proposed strategy for further developments.

1. Introduction

Stroke is a major public health problem since it represents the second leading cause of death worldwide and the first cause of acquired disability in adults. In the United States, this disease strikes once every 40 seconds and causes death every 4 minutes, with an estimated 41.6% death rate in 2007 [43]. Most frequently (80%) strokes result from the occlusion of one or several brain vessels and are thus called ischemic strokes (in the other cases, strokes are hemorrhagic strokes). Ischemic stroke involves many pathophysiological mechanisms causing devastating neurological damage (see for review [19, 47]). Understanding these mechanisms is of the most importance to develop new therapeutic strategies since no treatments are currently available for most stroke patients. Currently, the only FDA-approved treatment for stroke patients is a thrombolytic agent (tPA) which can only be given to less than 10% of patients because of its narrow time-window and its hemorrhagic risks [31]. Many neuroprotective agents (aimed at blocking the ischemic cascade) have also been developed but, although they had given very promising results in preclinical studies in rodent models, they appeared ineffective or even noxious during the clinical trials in stroke patients (see for review [5, 14, 27, 66]). This

Key words and phrases. Ischemic stroke, reaction-diffusion equations, operator splitting, parallel computing.

discrepancy between the results in rodents and in humans is partly due to the anatomic and histological differences between rodent and human brains. In this case, results in rodents are thus difficult to extrapolate to stroke patients. As a consequence, a mathematical model and its numerical simulations can help both to test some biological hypotheses concerning the involved mechanisms and to give new insights concerning the effects of these neuroprotective agents.

Previous works have been conducted on stroke modeling. One of these models is focused on the main mechanisms leading to cell death during the first hour of an ischemic stroke (such as ionic movements, glutamate excitotoxicity and cytotoxic edema)[21]. This model is based on a system of ordinary differential equations (ODE) and is mainly an electrophysiological model. It describes the dynamics of membrane potentials, cell volumes and ionic concentrations (K^+, Na^+, Cl^-, Ca^{2+} and Glu^-) in brain cells and in the extracellular space during a stroke. This model was used to study the role of various cell types during ischemia [24] and to explore the effects of various neuroprotective agents in stroke patients [22]. Other models have been developed to simulate and study spreading depressions during a stroke. This phenomenon is characterized by a slowly propagating depolarization of brain cells along with drastic disruption of ionic gradients [57]. These spreading depressions have recently been observed in stroke patients [20] and are supposed to extent the ischemic damage [56]. Some models reproduce and study the behavior of spreading depressions in neuronal cells [40, 50]. Others describe these depolarization waves though neuronal and glial cells [23]. Other models study the influence of the human brain cortex geometry on the propagation of these spreading depressions [23] [32]. These later models are all based on scalar reaction-diffusion systems. Our aim is to model the precocious mechanisms of stroke (i.e. ionic movements, glutamate excitotoxicity and cytotoxic edema) including the spreading depressions, on a realistic brain geometry. A first description of the algorithms used for the numerical solution of this stroke model on 1D and 2D geometries was presented in a previous article [15]. However, since we need to take into account the anatomic and histological specificities of human brain, this model must be simulated on a 3D realistic geometry, which implies to develop powerful numerical methods able to deal with a broad spectrum of spatial and temporal scales. This paper focuses on the methods developed for the numerical solution of this model, with much more insights on the mathematical and numerical methods than in [15]. The numerical method is based on operator splitting and explicit/implicit Runge-Kutta methods. We then show, for the first time, numerical simulations in 3D obtained thanks to a particular implementation of parallelism in the framework of shared memory machines. Moreover, these 3D simulations are computed on realistic geometries, obtained from MRI of the human brain, on conventional computational resources, that is on nowadays reasonably powerful workstations; and they are shown to match the observed dynamics from MR images in stroke patient. Since accuracy in 3D simulations is not yet optimal, the ability of extending the proposed numerical strategy to adaptive multi-resolution is presented in the framework of preliminary computations in simple geometries. The idea is to increase the level of accuracy in order to match all the spatial scales, with a better computational efficiency.

The paper is organized as follows: in a first part, we present the reaction-diffusion model of the precocious mechanisms of stroke with a stress on the properties of the system coefficients. In a second part, we focus on numerical methods: we first mention the different approaches which can be used to discretize the system in time and explain why in the context of such a stiff and large system only very few are relevant. We then present the various building blocks which will be used in the numerical methods: splitting methods, implicit Runge-Kutta methods and

explicit Runge-Kutta methods with extended stability domains; a grid adaptation technique is proposed as a possible improvement of the numerical strategy, considering particular features of the phenomena. In a third part, we present the parallel implementation on shared memory machines of the numerical strategy, and discuss the numerical validation of the results. 2D and 3D numerical results of simulations with complex geometry are presented, compared to real observations and discussed in the next section. Biomarkers are used in order to validate these computations. Finally, a brief and prospective study based on coupling the proposed strategy to adaptive multi-resolution in space is conducted, whereas conclusion and future works are presented in the last section.

2. Stroke modeling through stiff Reaction–Diffusion systems

In this section, we describe the model on which our study is based. This model includes ionic movements, glutamate excitotoxicity, cytotoxic edema and spreading depressions [21, 24]. It thus focuses on the first hour of a stroke, when the ionic exchanges are the main mechanisms leading to cell death. This model is based on a reaction-diffusion system.

In this model, brain tissue is composed of two cell types, namely neurons and glial cells, and of extracellular space. Two domains are considered: the white and the grey matter which differ in their glial cell composition (astrocytes in grey matter and oligodendrocytes in white matter) and in their "neuronal area" composition (neuronal somas in grey matter and neuronal axons in white matter). Human brain cortex is exclusively composed of grey matter whereas human brain medium is mainly composed of white matter (except the grey kernels). For simplicity reasons, we consider in the model that brain cortex contains only grey matter and brain medium contains only white matter. The ionic species considered in this model are K^+, Na^+, Cl^-, Ca^{2+} and the Glutamate (glu). They pass through neuronal and glial membranes via ionic channels (such as voltage-gated channels, receptor-channels, stretch-channels) and via ionic pumps and transporters (which are energy-dependent) (see figure 1). The ionic exchanges through voltage-gated channels have been first modeled by Hodgkin and Huxley [38].

The main precocious mechanisms of ischemic stroke can be described as follows (see for review [19, 47]): after the stroke onset, the cells in the ischemic area do not receive enough oxygen to maintain their production of energy. As a consequence, the activity of the ionic pumps decreases, which results in variations in ionic concentrations in the cells and in the extracellular space. These ionic variations have several consequences:

- the alteration of membrane potentials, resulting in membrane depolarization and in the opening of the voltage-gated channels;
- the cell swelling due to water influx;
- the increase of the neuronal concentration of Ca^{2+}, resulting in enzyme activation and leading cells towards necrosis;
- the increase of glutamate in the extracellular space, reinforcing the excitotoxic process;
- the increase of the concentration of K^+ propagating in the extracellular space and the increase of calcium in the astrocytic synticium, creating waves of cortical spreading depressions, opening further ionic channels and thus expanding the ischemic damage far from the ischemic core.

From this, we can understand the importance of studying these propagation phenomena and of exploring the potential effects of some neuroprotectors which modulate or block specific voltage-gated channels. Consequently, the model considers the following variables:

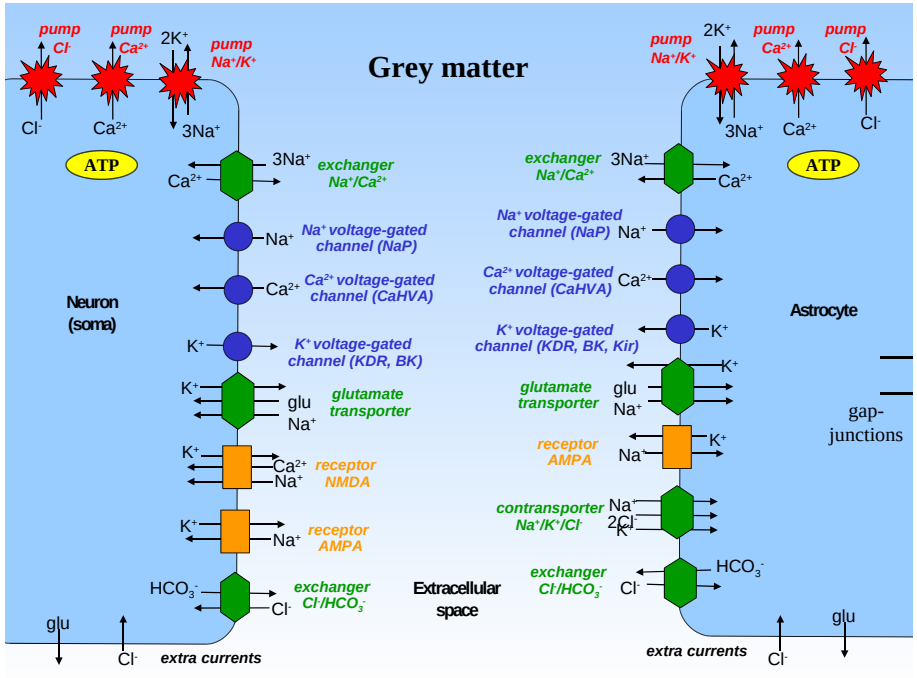


FIGURE 1. Ionic exchanges in gray matter between neurons, astrocytes and the extracellular space through voltage-gated channels, ionic transporters, receptor-channels and ionic pumps (from [21]).

- the volume fractions f_n and f_a (by brain volume unit) of neurons and glial cells. The fraction of volume remaining for the extracellular medium is thus $1 - f_n - f_a$. f_n and f_a are dependent from both coordinates and time;
- the membrane potentials V_n and V_a of neurons and glial cells (taking zero as reference potential in the extracellular medium);
- the concentrations of K^+, Na^+, Cl^-, Ca^{2+} and Glutamate in the 3 mediums (neurons, glial cells and extracellular space).

Altogether, the mean field model has $m = 19$ unknowns and is of reaction-diffusion type, except that there is no diffusion for 4 unknowns, namely f_n, f_a, V_n and V_a. There is also a difference in the number of reaction-diffusion equations in grey matter and in white matter. Since grey matter contains astrocytes (which are linked into an astrocytic synticium thanks to gap-junctions), ions are able to diffuse in the astrocytic space as well as in the extracellular space in grey matter. On the other side, as the main glial cells in white matter are oligodendrocytes (which do not have the same properties as astrocytes), ions are considered to be only able to diffuse in extracellular space in white matter. As a consequence, the model contains 10 reaction-diffusion equations in grey matter (for the concentrations of (K^+, Na^+, Cl^-, Ca^{2+} and Glu^- in astrocytes and in the extracellular space) and 5 reaction-diffusion equations in white matter (for the concentrations of (K^+, Na^+, Cl^-, Ca^{2+} and Glu^- in the

extracellular space). The general equation set is as follows:

$$(1)\qquad \begin{cases} \dfrac{\partial u_i}{\partial t}(x,t) - \mathrm{div}(\varepsilon_i(x)\mathrm{grad}\ u_i(x,t)) = f_i(u_1(x,t),\dots,u_m(x,t)), & 1 \le i \le m,\ x \in \Omega, \\ u_i(x,0) = u_i^0(x), & 1 \le i \le m,\ x \in \Omega. \end{cases}$$

The domain Ω corresponds to a human brain and is divided in grey and white matter. These two matters differ in several coefficients in the reaction term (corresponding to the cell composition) and in their diffusion coefficients. There are no fluxes of ions in and out of the brain and thus, the boundary conditions are of Neumann homogeneous type. For the initial conditions $u_i(x,0) = u_i^0(x)$, $1 \le i \le m$, a classical medical hypothesis is that the system is in a stable equilibrium: thus we take, and must find, a stable constant solution of system (1).

Let us mention some characteristics of the system very important in the choice of numerical schemes: the diffusion coefficients $\varepsilon_i(x)$ are low: about 10^{-3} given by a non-dimensional analysis. They take two constant values in grey and white matter. The reaction term $F = (f_1, ..., f_m)^t$ is extremely stiff; that is to say that if we consider the system of differential equations $du/dt = F(u)$, it is a stiff system according to the definition given in [34]. To see this, we have performed, by numerical differentiation, a computation of the Jacobian matrix $(\partial f_i/\partial u_j)$, $1 \le i, j \le m$, near a stable stationary value $F(u) = 0$, and we found numerically negative eigenvalues with negligible imaginary parts but with real parts in the range from -10^8 to about -1. Moreover, it is impossible to separate fast and slow variables and even if this was possible, the voltage dependent gates would make this separation very local in time and space. We have to deal with the stiffness of the reactive term F, which is the heart of the model and is a program of about 500 lines of C language, where ionic gates are modeled by sigmoid functions.

3. Numerical strategy: operator splitting and time integrators

One dimensional simulation are very useful to fit parameters such as the diffusion coefficients which are known in the literature only with limited accuracy; two dimensional ones are useful to validate numerical methods and programs, but only three dimensional simulations can be relevant from the medical point of view. From medical considerations, and also by some considerations on reaction-diffusion systems, we know that a precise description of the brain geometry is mandatory for the simulations, otherwise the plausible waves would be strongly perturbed, see for example [22]. We then have to think of a strategy dedicated to three dimensional simulations with a very fine spatial discretization allowing to resolve the broad spectrum of spatial and temporal scales of the system (1). The method developed has to be fast, robust and must take into account the properties of the model.

Concerning the spatial discretization, we have chosen a finite volume approach. Our experience is that, with uniform finite volumes, at least $\ell = 10^7$ volumes are necessary for a realistic three dimensional simulation. The continuous unknown u is then replaced by a vector U belonging to $\mathbb{R}^{m\times\ell}$ corresponding to the m unknowns at each point x_i, $1 \le i \le \ell$. We use MRI pictures and we consider pixels as center of volumes of an uniform grid. When we apply this spatial discretization to the system (1), this yields a large system of ordinary differential equations. Let us write this system under the form

$$(2)\qquad \frac{dU}{dt} = A_\varepsilon U + F(U),$$

A_ε being a matrix corresponding to the discretization of the diffusion operator; this is a classical 5 terms (resp. 7 terms) by line matrices in dimension 2 (resp. 3). We now present the different approaches which can be used to discretize this system in time and we explain why in the context of such a stiff and large system like (1), only few are efficient.

The first idea is to use directly a solver of systems of ODEs, the so called method of lines, but due to the stiffness of the nonlinear term, at each time step, a large system of algebraic equations should be solved which is too much time consuming. It is then better to use different discretizations in time for the linear and the nonlinear terms. A first method is to use an Implicit–Explicit method by treating the linear term implicitly and the nonlinear term explicitly. If we denote by Δt the time step and U_k the approximated solution at time $k\Delta t$, the simplest method is the following:

$$\frac{U_{k+1} - U_k}{\Delta t} + A_\varepsilon U_{k+1} = F(U_k).$$

One must solve a linear system at each step since diffusion is taken implicitly but the nonlinear term is taken explicitly. This method is of order 1 in time. More accurate, but not really more expensive, methods of the same type and of order at most 6 are described and analyzed in [4]. The main advantage of these methods is that only *linear* systems must be solved but the drawback is that, due to the explicit computation of the reaction terms, these methods are adapted only to systems with non stiff reaction terms. Let us recall that the system (1) is very stiff, and these methods can only work with time steps of the same order of the fastest time scale of the system which is about 10^{-8} seconds. This would result in an prohibitive computing time, about 4×10^{11} steps for simulating the first hour of the evolution of the stroke.

A better idea in the treatment of the linear and the nonlinear part in the context of a stiff nonlinear term is to "reverse" the numerical treatments: to solve explicitly the linear part and implicitly the nonlinear part. The discretization of the linear part is made using an explicit Runge-Kutta method with extended stability domain along the negative real axis. The papers [58] and [64] settled the foundation for these methods called IMEX methods and particular methods devoted to stiff non linear problems are presented in [65] and [55]. The main advantage of these methods is that they treat diffusion terms explicitly and the stiff reaction terms implicitly. Furthermore, the stiff reaction term is decoupled over space grids and yields to small sized systems. These methods are usually very efficient; nevertheless, the computational requirements associated mainly to an implicit solver over the discretized domain with the same time step become soon critical when treating large computational domains.

Finally, the only possible methods which can solve system (1) seem to be the so called splitting methods that we describe in details now.

3.1. **Splitting methods.** The idea is as old as numerical analysis and was used and analyzed by the Soviet school in the 60's (see for example [44]). In those times, the main interest was the economy of computer memory. The idea, applied to spatially discretized reaction-diffusion equations is to solve alternatively the reaction and the diffusion problems. For example, starting from some initial condition, we solve for a time step of Δt:

$$\frac{dV}{dt} = A_\varepsilon(V),$$

with an initial condition

$$V(0) = V_0.$$

Let us call $D_{\Delta t}$ this procedure. Taking $V(\Delta t)$ as initial condition, we solve for the same Δt:

$$\frac{dW}{dt} = F(W),$$

with

$$W(0) = V(\Delta t),$$

and we call $R_{\Delta t}$ this second procedure. By taking the value $W(\Delta t)$, we repeat this procedure to obtain, for $k > 0$, $W(k\Delta t)$. The previous approximation is an approximation of order 1 in time of the solution of (2). Let us recall that a method is of order 1 (or more generally of order p) if the expansion in powers of Δt of the numerical solution coincides with that of the true solution up to and including the order 1 (more generally the order p). The previous approximation is called a Lie method, but one can define different numerical methods: to obtain an approximation at time Δt, one can apply successively $R_{\Delta t}$ and $D_{\Delta t}$, or more generally apply successively $R_{\Delta t/2}$, $D_{\Delta t}$ and $R_{\Delta t/2}$, or $D_{\Delta t/2}$, $R_{\Delta t}$ and $D_{\Delta t/2}$. The last two approximations are called Strang methods [60] and are of order 2.

Let us explain the main advantages of these methods: the reaction and diffusion are decoupled, the solution of the $D_{\Delta t}$ problem is reduced to the solution of m independent diffusion equations, and thus the complexity is reduced. Concerning the $R_{\Delta t}$ problems, one immediately see that they are decoupled in ℓ systems of ODEs of size m, as many systems as nodes in the finite volume mesh, and that *all* these systems are independent.

Assuming first that $D_{\Delta t}$ and $R_{\Delta t}$ can be solved exactly without time discretization, these splitting methods can be used to solve stiff systems of reaction-diffusion. Better performances are expected by ending the splitting scheme with the integration of the reaction part or more generally with the part involving the fastest time scales of the phenomenon (see [17, 16] and references therein).

Keeping in mind these theoretical studies and considering the various numerical alternatives previously discussed, Strang's splitting scheme ending with the reaction part remains as the most appropriate resolution scheme for general multi-scale problems and so far, the best choice for our numerical study. All the numerical simulations in this article are then performed with this scheme. The concept of splitting is very natural; we still have to describe the numerical methods used in each sub-step. These methods have to be chosen carefully in order to provide an accurate approximation of system (2). It turns out that these two methods are not classical, and thus must be explained in details. This is the purpose of the next two sections which are self-contained.

3.2. **Time integration of the reaction problem: the Radau5 method.** In this section we are considering the numerical approximation of the $R_{\Delta t}$ step. Let us recall that we denote by ℓ the number of nodes in the finite volume mesh and the $R_{\Delta t}$ part corresponds to solve ℓ independent ordinary differential equations of the form, for $t > t_0$,

$$\frac{du}{dt} = F(t, u), \quad u(t_0) = u_0, \tag{3}$$

with $F : [t_0, +\infty[\times\mathbb{R}^m \to \mathbb{R}^m$. For a general s-stage Runge-Kutta method, after an integration time step Δt, the solution $u(t_0 + \Delta t)$ of problem (3) might be approximated by $\tilde{u}_1$,

which is given by the solution of

$$
\begin{aligned}
U_i &= u_0 + \Delta t \sum_{j=1}^{s} a_{ij} F\left(t_0 + c_j \Delta t, U_j\right), \quad i = 1, \ldots, s, \\
\tilde{u}_1 &= y_0 + \Delta t \sum_{j=1}^{s} b_j F\left(t_0 + c_j \Delta t, U_j\right),
\end{aligned} \tag{4}
$$

where b and c are two vectors of $\mathbb{R}^s$, $b = (b_1, \cdots, b_s)^t$ and $c = (c_1, \cdots, c_s)^t$, and A is a $s \times s$ matrix, $A = (a_{ij})_{1 \leq i,j \leq s}$. Usually, these coefficients are arranged in a mnemonic device, known as a Butcher's array

$$
\begin{array}{c|ccccc}
c_1 & a_{11} & a_{12} & \cdots & a_{1s-1} & a_{1s} \\
c_2 & a_{21} & a_{22} & \cdots & a_{2s-1} & a_{2s} \\
\vdots & \vdots & & \ddots & & \vdots \\
c_s & a_{s1} & a_{s2} & \cdots & a_{ss-1} & a_{ss} \\
\hline
 & b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}
$$

When there are nonzero coefficients a_{ij} with $i \leq j$, the method is implicit, and systems of non linear equations must be solved to compute the U_is into (4). Let us recall that by definition a Runge-Kutta method is of order p if the expansion in powers of Δt of the numerical solution coincides with that of the true solution up to and including a certain order p. An important result obtained by Butcher [10] is that a Runge-Kutta method is of order p if it satisfies

$$
\sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q}, \quad q = 1, \ldots, p \tag{5}
$$

and for η and ζ such that $p \leq \eta + \zeta + 1$ and $p \leq 2\eta + 2$,

$$
\begin{aligned}
&\sum_{j=1}^{s} a_{ij} c_j^{q-1} = \frac{c_i^q}{q}, && i = 1, \ldots, s, \ q = 1, \ldots, \eta, \\
&\sum_{i=1}^{s} b_i c_i^{q-1} a_{ij} = \frac{b_j}{q}(1 - c_j^q), && j = 1, \ldots, s, \ q = 1, \ldots, \zeta.
\end{aligned} \tag{6}
$$

If we now consider the stability features, a classical analysis based on the *Dahlquist test equation* [13]

$$
y' = \lambda y, \quad y(0) = 1, \tag{7}
$$

allows to define the *stability function* $R : \mathbb{C} \to \mathbb{C}$ of a method as the numerical solution of (7) given by the method itself after one time step Δt. In the case of a Runge-Kutta method, solving (7) with $y(0) = y_0$ with the scheme given by (4) leads to

$$
y_1 = R(z) y_0, \quad z = \Delta t \lambda.
$$

In general, R is a rational function and in the particular case of explicit Runge-Kutta method, a polynomial function.

In the same context, the set

$$
S := \{z \in \mathbb{C} \ : \ |R(z)| \leq 1\} \tag{8}
$$

is called the *stability domain* of the method and a particular method will remain stable as long as z belongs to S. Notice that in a general case, problem (3) can be linearized and supposing a diagonalizable Jacobian matrix $J = \partial F/\partial u$, an analogous problem to that of Dalquist can be obtained with the complex eigenvalues $\lambda_i, i = 1, \cdots, m$ of J. As a consequence, if (3) is a stiff system of ODEs (see [34] for characterization of a stiff ODE), then it is very likely that large λ_i with $\Re\lambda_i \leq 0$ will take a leading role in the solution, and either a larger *stability domain* S must be required for a fixed time step Δt or smaller Δt for a fixed S. In order to overcome this difficulty, A-stable methods for which

$$S \supset \{z \in \mathbb{C} \ : \ \Re z \leq 0\}, \tag{9}$$

are usually preferred, so that prohibitive time steps Δt are dismissed.

However, it is not possible to construct such A-stable methods from explicit Runge-Kutta methods with $a_{ij} = 0$ for $i \leq j$ in the Butcher's array, because in those cases, R will always be a polynomial. Therefore, appropriate implicit Runge-Kutta schemes have been studied. Hence, based on the works of Butcher on Radau quadrature formulae [49], Ehle has constructed a family of formulae named 'Radau IIA' in which Runge-Kutta schemes are generated by collocation at the nodes of these quadrature formulae [28]. These RadauIIA methods have been then conceived in order to be A-stables and of order $p = 2s - 1$ for a given number of stages s given conditions (5) and (6). For instance, it yields the backward Euler method for $s = 1$, but higher order methods are possible. Actually, the case $s = 3$ for which the Butcher array is given by

$$\begin{array}{c|ccc}
\frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\
\frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\
1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\
\hline
 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9}
\end{array}$$

is the Radau IIA method on which the routine Radau5 is based [34, 35]. Hence, Radau5 is formally a fifth order implicit Runge-Kutta method and it has been proven that this order might be reduced at worst to a third order, for example, in the case of singular perturbation problems [33]. The corresponding stability function is given by

$$R(z) = \frac{1 + \frac{2z}{5} + \frac{z^2}{20}}{1 - \frac{3z}{5} + \frac{3z^2}{20} - \frac{z^3}{60}}. \tag{10}$$

Even though there are other implicit Runge-Kutta methods based this time on Gaussian quadrature formulae that can normally yield higher order than $p = 2s - 1$, Radau based formulae allow the construction of L-stable methods, that is

$$\lim_{|z|\to\infty} R(z) = 0,$$

as it can be verified in (10). From a theoretical point of view, this implies that these methods are more efficient in damping out fast transient phases, a very common situation when dealing with very stiff problems as in our case (see [34] for further details). In [34], a Fortran program of Radau5 is proposed, written with very careful optimizations. The most time consuming part is the computation of the Jacobian of F. As F is a complicated procedure, the only way to compute the Jacobian is by numerical differentiation, as it is proposed in the Fortran routine; this result in computing 20 times F at each evaluation in our problem. To end this paragraph, let us make further comments on the choice of this integration method:

- Concerning Runge-Kutta methods, we have tried not only the Radau5 method, but also the Linearized Euler extrapolated method and Rosenbrock methods (see [34]): none of them is as fast and as robust as Radau5.
- Multistep methods like those based on backward differentiation formulae (see [34] for details) are not adapted to splitting methods since they need more than one initial condition at each time step to perform the time integration of the reaction. These initial conditions are often approximated by a less accurate procedure.
- Finally recalling that the order might be reduced at worst to a third order, the error coming from the discretization of the reaction part will be small comparing to the error coming from the splitting strategy and this property is realized only if $s \geq 3$.

3.3. **Time integration of the diffusion problem: The ROCK4 method.** In many cases, there are stiff problems for which A-stable methods are not necessarily required. Some remarkable examples come from the discretization of parabolic PDEs which lead to problems with a Jacobian matrix involving large eigenvalues close to the real negative axis. Therefore, instead of A-stable but time consuming implicit procedures, *stabilized* explicit Runge-Kutta methods should be preferred [63].

The main idea of these methods is to construct methods of order p with a family of stability polynomial of degree s, R_s, such that

$$\begin{aligned} &R_s(z) = 1 + z + \cdots + \frac{z^p}{p!} + \sum_{i=p+1}^{s} \alpha_{i,s} z^i, \\ &|R_s(z)| \leq 1 \quad \text{for} \quad z \in [-\ell_s, 0], \end{aligned} \tag{11}$$

with $s \geq p+1$, $\alpha_{i,s} \in \mathbb{C}$ and ℓ_s as large as possible. If we consider Chebyshev polynomials, they yield ℓ_s proportional to s^2, *i.e.* $\ell_s = \beta_s s^2$. For instance, for $p = 1$, the optimal polynomials are

$$R_s(z) = T_s\left(1 + \frac{z}{s^2}\right),$$

the shifted Chebyshev polynomials which yield optimal $\ell_s = 2s^2$.

Great efforts were then made in order to achieve second order stabilized explicit Runge-Kutta methods based on Chebyshev-type polynomials (for further details, see [39] and references therein). For example, RKC methods proposed by Sommeijer, Shampine and Verwer in [58] have gained notorious reputation over the last years and are based on the use of a three-term recurrence relation of Chebyshev polynomials proposed in [61] and [59].

Nevertheless, with this background and based on the study of optimal stability polynomials satisfying (11) in [1], Medovikov and Abdulle proposed in [3] to approximate $R(z)$ by

$$\tilde{R}_s(z) = \tilde{w}_p(z)\tilde{P}_{s-p}(z) \tag{12}$$

in order to achieved $p = 2$, where $\tilde{w}_p$ is a polynomial with p complex roots and $\tilde{P}_{s-p}$ is an orthogonal polynomial associated with the weight function $\tilde{w}_p(z)^2/\sqrt{1-z^2}$. Moreover, they searched to satisfy (11) with an approximated $\tilde{\ell}_s$ as close as possible to the optimal ℓ_s. Based on the same ideas, the forth order case of (12) was proposed by Abdulle in [2] and gave birth to the so called ROCK4 method which is a stabilized explicit Runge-Kutta with stability domain limited by $\tilde{\ell}_s \simeq 0.35 \cdot s^2$ along the negative real axis.

From a theoretical point of view, ROCK4 uses the theory of composition of methods (the "Butcher group") in order to achieve a fourth order method denoted by WP. The first method, P, is built upon the three-term recurrence relation of the orthogonal polynomials $(\tilde{P}_j)_{j=1}^{s-4}$ previously mentioned,

$$\tilde{P}_j(z) = (\mu_j z - \upsilon_j)\tilde{P}_{j-1}(z) - \kappa_j \tilde{P}_{j-2}(z),$$

which is used to define an explicit Runge-Kutta scheme

$$\begin{array}{rcl} U_0 & := & u_0, \\ U_1 & := & u_0 + \Delta t\ \mu_1\ A_\varepsilon U_0, \\ U_i & := & \Delta t\ \mu_i\ A_\varepsilon U_{i-1} - \upsilon_i\ U_{i-1} - \kappa_i\ U_{i-2}, \quad i = 2, \ldots, s-4, \\ \tilde{u}_1 & := & U_{s-4}. \end{array} \tag{13}$$

The corresponding coefficients in the Butcher's array of P (see Table 1) are then defined considering (13) and the general scheme (4) as showed in [2]. Notice that necessarily the number of stages s satisfies $s \geq 5$ and that scheme (13) applied to the Dahlquist test equation (7) with $y(0) = y_0$ yields

$$y_1 = Y_{s-4} = \tilde{P}_{s-4}\, y_0,$$

and thus defines $\tilde{P}_{s-4}$ as the stability function of method P.

On the other hand, the coefficients in the Butcher's array of the method W (see Table 1), which possesses $\tilde{w}_4$ as stability function, are then derived out of the forth order conditions established for method WP and the previously calculated coefficients of P.

$$\begin{array}{c|cccc} 0 & & & & \\ \widetilde{c}_2 & \widetilde{a}_{21} & & & \\ \vdots & \vdots & \ddots & & \\ \widetilde{c}_{s-4} & \widetilde{a}_{s-4,1} & & \widetilde{a}_{s-4,s-5} & \\ \hline & \widetilde{b}_1 & \cdots & \widetilde{b}_{s-5} & \widetilde{b}_{s-4} \end{array} \qquad \begin{array}{c|cccc} 0 & & & & \\ \widehat{c}_2 & \widehat{a}_{21} & & & \\ \widehat{c}_3 & \widehat{a}_{31} & \widehat{a}_{32} & & \\ \widehat{c}_4 & \widehat{a}_{41} & \widehat{a}_{42} & \widehat{a}_{43} & \\ \hline & \widehat{b}_1 & \widehat{b}_2 & \widehat{b}_3 & \widehat{b}_4 \end{array}$$

TABLE 1. Buthcher's array of the method P (left) and W (right).

Finally, construction of the ROCK4 method is based on method WP with stability function given by (12) with $p = 4$. In [2] A. Abdulle has proposed an implementation of ROCK4. Actually, when applied to linear problems, ROCK4 relies on three elementary operation, namely 1) matrix vector products, 2) linear combination of vectors and 3) scalar products. Thus the method is easy to implement in any computational framework, sequential or parallel. Finally as mentioned for the Radau5 method, the error coming from the discretization of the

diffusion part with the ROCK4 method will be small comparing to the error coming from the splitting strategy.

3.4. **An adaptive mesh strategy: the adaptive multiresolution approach.** Most software for the solution of partial differential equations use constant discretization in time and even in space. But stroke is a phenomenon which is localized in a small part of the brain, at least at the onset, and which exhibits propagating waves and large gradients. Thus, an adaptive mesh is certainly well fitted for such simulations, resulting in a better resolution of thin spatial structures and hopefully, in better numerical performances. The mesh adaptation must be automatically managed by the software. A numerical strategy has been proposed and evaluated in [25], where the combination of adaptive spatial multiresolution with the time integration solvers previously detailed, is described.

Historically, adaptive methods like Multi-Level Adaptive Techniques (MLAT) [9] or Adaptive Mesh Refinement (AMR) [6, 8, 7] were among the first to achieve this goal, using a set of locally refined grids where steep gradients are found. Furthermore, adaptive multiresolution methods, based on Harten's pioneering work [37], have been developed for 1D and 2D hyperbolic conservation laws [12, 30] and then extended to 3D parabolic problems [53]. Consequently, high data compression might be achieved with all these methods. However, one major advantage of the adaptive multiresolution techniques is that the numerical analysis of the errors has already been conducted [37, 12] and thus, a solid theoretical background was already settled.

Considering that we have a set of nested spatial grids, from the coarsest to the finest one, a multiresolution transformation allows to represent a set of function data as values on a coarser grid plus a series of differences (error estimators) at different levels of such nested grids. The information at consecutive levels are then related by inter-level transformations: which are the *projection* and *prediction* operators. The theoretical background of such configuration (see [12]) states that wavelet coefficients can be then defined as prediction errors, and they will retain the *detail* (local regularity estimator) information when going from a coarse to a finer grid (see figure 2 for an example of nested grids). Finally, one of the main interests of

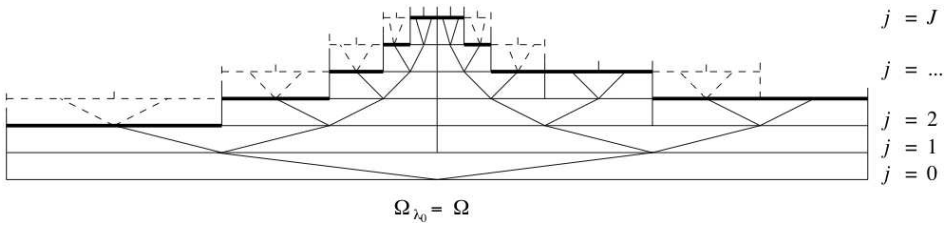


FIGURE 2. Example of nested grids for multiresolution adaptive technique.

carrying on such multiscale decomposition is that this new representation leads us to define a whole set of regularity estimators all over the spatial domain. Hence, an effective data compression might be performed in order to retain only a minimal quantity of nodes where it is strictly necessary. The evolution problem is then simulated on a dynamical adapted grid. For an overview on adaptive multiresolution techniques, we refer to the books of Cohen [11] and Müller [48].

4. Numerical software

We have developed two different softwares for the solution of the system (1). Both implement the time integration strategy defined above in 3.1, 3.2 and 3.3:

FM: (Fixed Mesh) a code using a fixed spatial discretization, with finite volumes of constant size [26]. This code takes into account complex geometries in the following way: starting from MRI images, we take each pixel as the center of a finite volume; it aims to be a framework for testing and exploiting numerical methods for 1, 2 and 3D reaction-diffusion systems. It will be used in order to obtain the main results of the present contribution, that is numerical simulations of the complex and stiff stroke model in complex 3D geometries.

MR: (Multi Resolution) a code using an adaptive multiresolution method as defined above in 3.4. In the framework of multiresolution, an important amount of work is still required in order to optimally combine all the numerical methods described here, the most difficult aspects are related to programming features such as data and code structures, as indicated in [25]. Nowadays, this program can only solve problems in simple domains like squares and cubes; simulations with an adaptive multiresolution approach on a complex geometry are not yet available, and we will only present here 2D and 3D simulations in simplified geometries for the sake of assessing our results and perspectives in the field.

Let us remark that the (FM) code is a highly optimized and complete code for the simulation of reaction-diffusion equations. In particular, stroke simulations in complex geometry can be performed for the first time, with standard computing resources, and constitutes the major advance of our contribution. On the other hand, the second code (MR) allows to validate to some extents the previous numerical results, and it is meant to be a potential extension to (FM) in future developments.

5. Implementation and performances of the numerical methods on shared memory machines

We describe now the implementation of the 3D simulations which are performed on a uniform grid and a complex geometry with the code (FM). Let us emphasize the particular parallelism implementation that we have conceived in the framework of shared memory machines. All the computations have been performed on a 8 core (2x4) 64 bits machine (AMD Shanghai processors).

Using an implicit procedure, the reaction steps are by far the most time consuming parts of the computation. This step is naturally parallel, as we have to solve a large number ℓ of independent systems of differential equations, each of them corresponding to a single cell. But solvers like Radau5 use a time adaptive strategy, together with Newton method: that is, the computing time is varying from one point to another, and a fixed domain partition strategy, with an affectation of subdomains to processors, is not optimal for load balancing. Therefore, our implementation uses threads, implemented in the C++ boost-thread library [67]. Let us describe it shortly in the next part.

5.1. Implementation with threads.

We divide the set of finite volumes into small subsets $S_i,\ i = 1, ..., k$ where k should be much larger than the number of computing units. We build a stack of all the S_i before each time step. By calling the procedure GetSubset, each thread gets one S_i, while the stack is not empty. Threads join when the stack is empty.

Algorithm 5.1: GET A SUBSET( )

```
procedure GetSubset( )
  Lock l;
  if stack not empty
  finished ← false;
  s ← stack.pop();
    else
  finished ← true;
  s ← null;
  return (finished, s)
```

The "Lock" object (a boost thread library object), is created when a thread enters the procedure, forbidding entry to other threads (this is a basic and mandatory feature of all threading libraries). The stack is popped, giving the thread a set of differential equations to be solved. When the thread exits the procedure, the lock object is automatically destroyed and an other thread can enter. To minimize lock competition effects, the size of each S_i is taken constant (about 100 nodes) plus a random size between zero and 10% of the constant.

Each thread is actually a procedure:

Algorithm 5.2: THREAD( )

```
procedure ThreadProcedure( )
  repeat
   (finished, s) = GetSubset();
   if not finished
   for each x ∈ s
     do solve ODEsystem at x;
  until finished;
```

And eventually, the solution of the reaction step with n threads is very simple, using the BOOST library:

Algorithm 5.3: REACTION( )

```
procedure Reaction(n)
  thread_group thrds(n);
  for i ← 1 to n
    do thrds.create_thread(ThreadProcedure);
   thrds.join_all()
```

thread_group is a container, member of the boost-thread library, which manages a set of threads (n threads here). The threads are spawned with the method *create_thread* and run in parallel, each thread launching the *ThreadProcedure* routine. The *join_all* method acts

as a meeting point for all the threads, and the execution waits until all threads have finished their computations.

5.2. **Implementation checkout and accuracy evaluation of the code.** In order to check out the implementation of the method in the codes (FM) and (MR), which use two different spatial discretizations and data structures, we have conducted a detailed comparison on a numerical test-case. Nevertheless, we do not pretend it as a full validation step since the numerical methods used for the time integration are exactly the same. We have considered a 2D case in a regular geometry of $[0,5] \times [0,5]$ (cm), using two resolutions, one on a fixed grid computed with the code (FM) and the other on the adapted grid obtained by multiresolution with the code (MR). Both methods, based on Strang's splitting, use Radau5 and ROCK4 as time integrators for the reaction and for the diffusion problem.For the model parameters, we have considered only one domain, the gray matter. The time of integration was restricted to one hour, $t \in [0, 3600]$ seconds.

We thus compare both solutions with a L^2-norm at final time $t = 3600$, and define an error estimator e given by $\|u(.,t) - u_{MR}(.,t)\|_{L^2}$, where index MR denotes the use of multiresolution techniques. The spatial discretization consists of an uniform mesh of 256^2 points. In the case of the spatial adaptive method, there is a set of nested grids arranged in 8 different levels from the coarsest to the finest discretization. The latter corresponds to the uniform mesh previously considered of 256^2 points. In order to compare these results, we must consider the same spatial discretization for both solutions: this is easily achieved with the mentioned projection/prediction operations on the adapted grid. Tables 2, 3 and 4 show the minimum and maximum values of variables in the neurons, the astrocytes and in the extracellular space respectively, as well as the normalized L^2 difference of the numerical results e obtained by the adaptive multiresolution strategy and our proposed numerical strategy.

Variable	**Min value**	**Max value**	e
K^+	68.9338	141.6940	3.4059×10^{-3}
Na^+	7.0834	75.2787	6.0126×10^{-3}
Ca^{2+}	1.0558×10^{-4}	9.3376×10^{-4}	2.0559×10^{-2}
Cl^-	11.5492	22.3907	3.1682×10^{-3}
glu	0.0808	9.3415	7.0681×10^{-3}
V_n	-57.6666	-3.7338	7.0782×10^{-3}
f_n	0.0799	0.0878	6.8508×10^{-4}

TABLE 2. Minimum and maximum values of variables in the neurons, computed with the (MR) code, and normalized L^2 difference e of numerical results between uniform mesh and (MR). t= 3600 s.

Figure 3 shows the evolution of the propagating phenomenon on an adapted grid for variable K^+ in the neurons. The refined regions clearly correspond to the wavefront area where the steep spatial gradients are present. Finally, figure 4 reveals in a qualitative way the different representations of the numerical solution on an uniform mesh and on the adapted one.

We first notice a great accordance between the solutions of the two different codes: (MR) and (FM), in the 2D simulations. Let us recall that both codes rely on two well tested, robust and publicly available numerical routines: the Radau5 and ROCK4 methods; therefore, one

Variable	**Min value**	**Max value**	e
K^+	124.2309	132.6962	1.2265×10^{-3}
Na^+	15.0751	24.3063	6.3289×10^{-3}
Ca^{2+}	1.5921×10^{-4}	0.4149	3.1653×10^{-2}
Cl^-	6.7503	10.8147	1.0143×10^{-2}
glu	2.5460	2.9870	1.0133×10^{-3}
V_a	-75.2476	-19.6358	1.0817×10^{-2}
f_a	0.7128	0.8088	1.1877×10^{-3}

TABLE 3. Minimum and maximum values of variables in the astrocytes, computed with the (MR) code, and normalized L^2 difference e of numerical results between uniform mesh and (MR). t= 3600 s.

Variable	**Min value**	**Max value**	e
K^+	4.8682	59.4336	6.1640×10^{-3}
Na^+	82.7306	141.0174	2.3939×10^{-3}
Ca^{2+}	0.0740	2.0027	2.4298×10^{-2}
Cl^-	142.3254	150.2269	4.7433×10^{-4}
glu	7.2590×10^{-4}	0.0791	2.3966×10^{-3}

TABLE 4. Minimum and maximum values of variables in the extracellular space, computed with the (MR) code, and L^2 normalized difference e of numerical results between uniform mesh and (MR). t= 3600 s.

can consider that they only differ by the different spatial discretizations and data structures they use: the comparison can thus be considered as a (partial) cross validation.

Besides such a level of comparisons, (MR) will be shown to pave the way towards higher levels of refinement for a better resolution of the details of the dynamics in 3D at a reasonable cost. In fact, considering this 2D numerical test-case, finer spatial discretizations yield naturally better resolution of both the wave velocity and the dynamics of the wavefront, as seen into figure 5. See the corresponding (MR) adapted grids into figure 6. Even if it is clear that the wave is better resolved on the finest grid (1024^2), in particular it is somewhat faster, the qualitative value of the wave velocity is correctly captured even on the coarsest grid (256^2), which corresponds roughly to the 3D simulations with (FM). However, we can not yet simulate with (MR) all the complex geometries of the brain we are investigating in this paper; and thus, all the results in complex geometries will be performed with (FM).

Let us make now some comments on the performance of the numerical method on shared memory machines for both diffusion and reaction equation solvers in (FM) used in the next Results section.

5.3. **Performances of the diffusion equation solver.** Only a poor man's parallelism is implemented for this step, each diffusion equation being solved by one thread. But actually, as we will see later, the computing time of this step is less than 10% of the total computing time (in dimension 2 or 3); each step, for one diffusion equation, needs only 6 matrix vector products (5 being the minimum for the ROCK4 method, plus one for the error estimate).

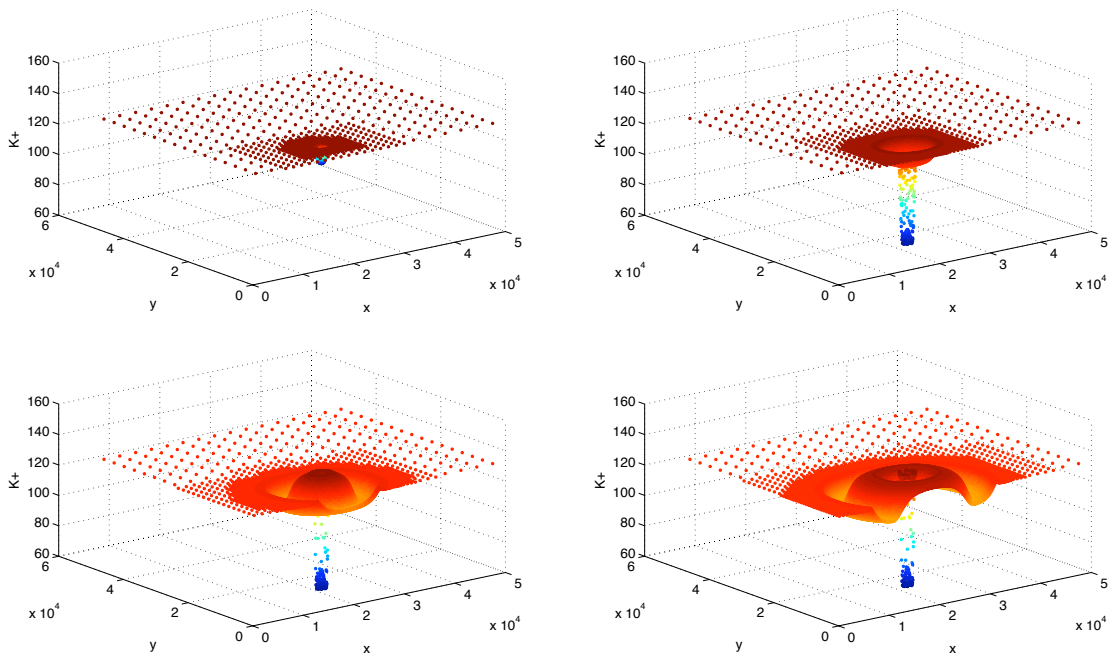


FIGURE 3. Evolution of $K+$ in the neurons at $t = 100$ (top left), $t = 1000$ (top right), $t = 2000$ (bottom left) and $t = 3000$ (bottom right).

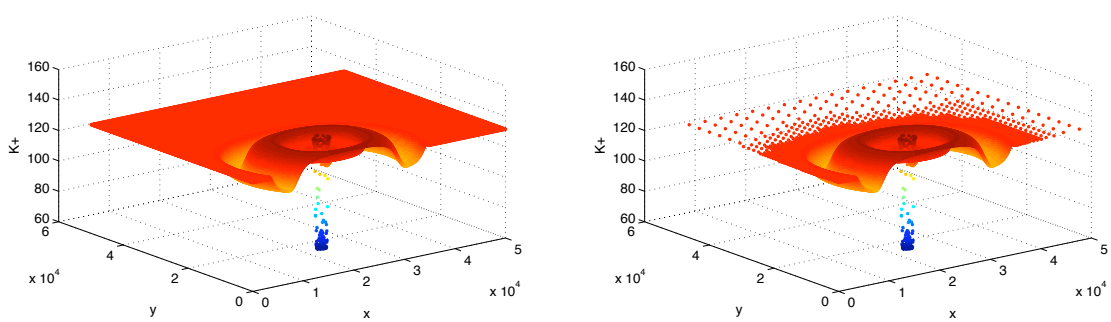


FIGURE 4. $K+$ in the neurons at $t = 3600$ on a 256^2 uniform mesh (left) and the corresponding adapted grid (right).

5.4. **Performances of the reaction solver.** The main question concerns the efficiency of the multithreaded parallelism. Figure 7 shows the computing time with 1, 2, 4 and 8 threads, across 12 steps. The wall clock computing time is multiplied by the number of threads (unit is in number of CPU clock ticks); in case of perfect scalability, all the points with the same number of threads should be at the same ordinate. This is not exactly the case, but the loss of scalability, comparing computations with 1 thread and computation with 8 threads, is about 6%. We conclude that this implementation is very efficient. This is a consequence of the complexity of the right hand side (the reaction term): even one single numerical evaluation of F is much more time consuming than the overhead introduced by the thread mechanism.

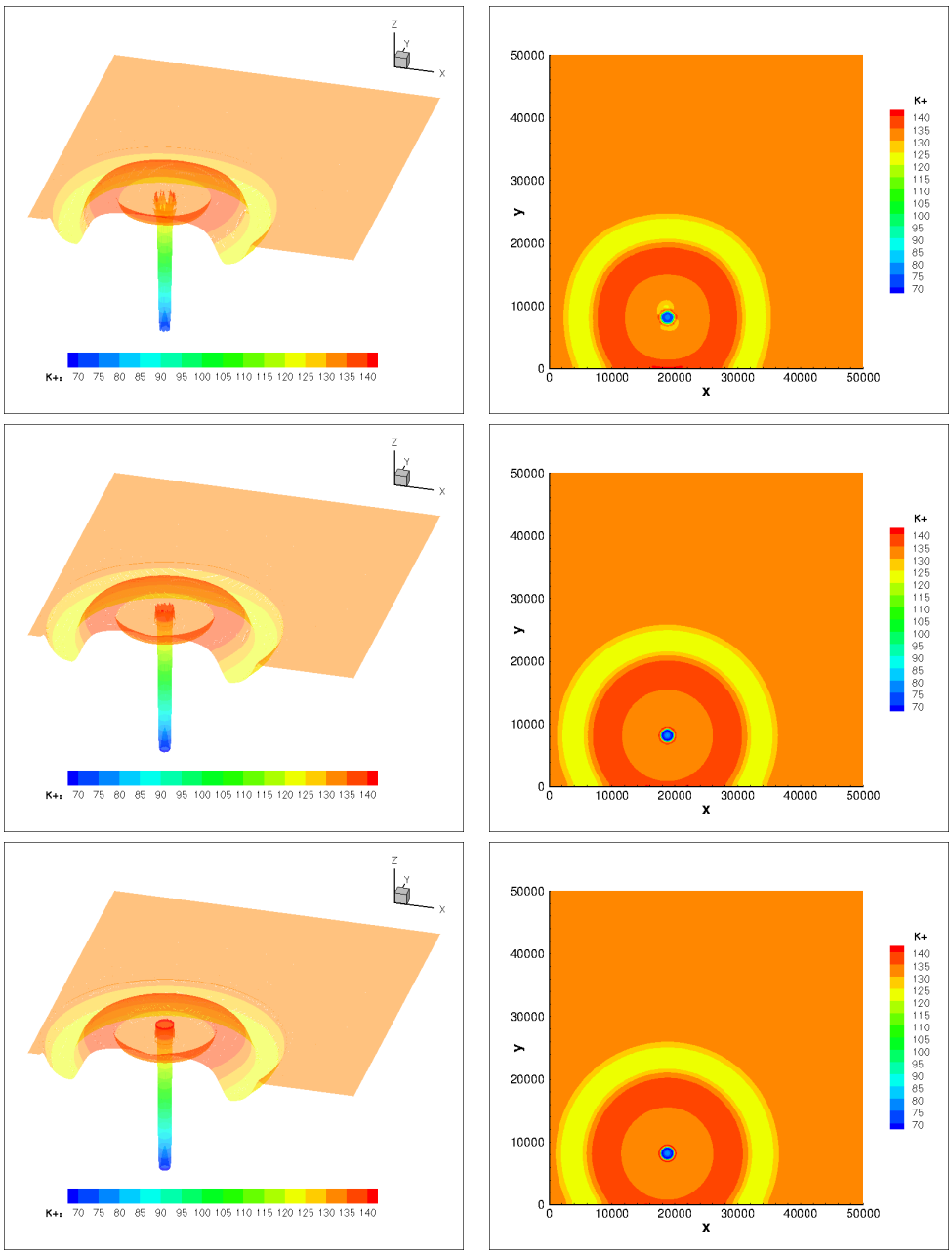


FIGURE 5. $K+$ in the neurons at 3600 s. for a 2D mesh of 256^2 (top), 512^2 (center) and 1024^2 (bottom).

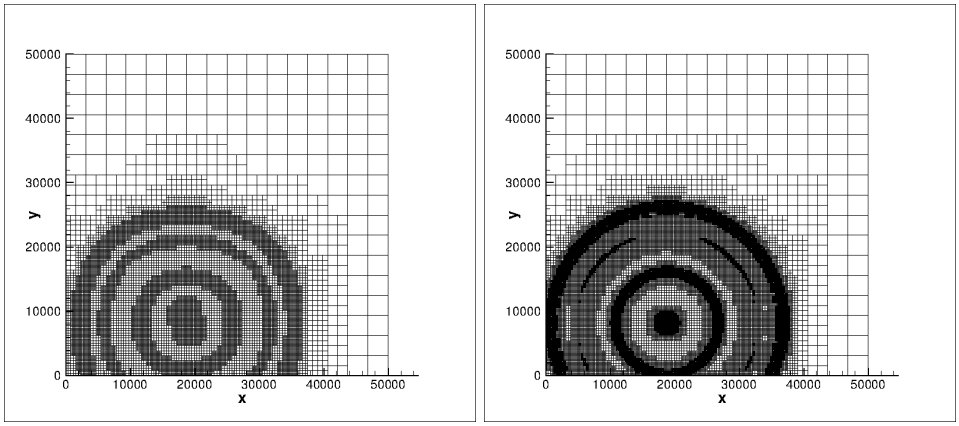


FIGURE 6. 2D adapted meshes equivalent to 256^2 (left) and 1024^2 (right) spatial discretizations at the finest grid.

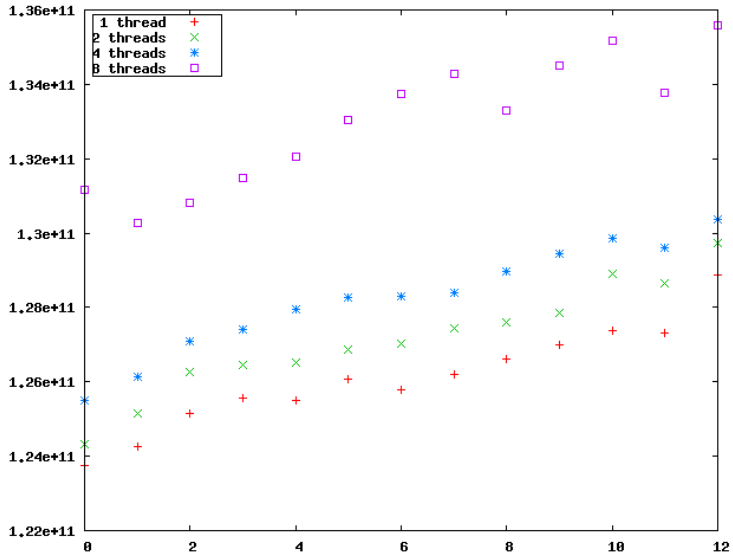


FIGURE 7. Performances of the multithreaded reaction solver.

6. 2D and 3D numerical results in complex geometry with fixed mesh simulations and validation

We present here some results of 2D and 3D simulations obtained with the code (FM) in complex geometries of the brain. Two variables have been chosen for the model validation: the potassium concentration in the extracellular space ([K+]e) and the ratio of apparent diffusion coefficient of water (rADCw).

- The potassium concentration cannot be measured in vivo in the brain of stroke patients but it can be measured ex vivo or in vitro on brain tissues. These concentration values give some indications on the severity of the damage. The physiological value of [K+]e is about 5 mM. It was observed to be able to increase up to 35 mM in areas of moderate ischemia where depolarization waves can spread [42] and up to 75-90 mM in areas of severe ischemia where most cells are dead [36]. The first step of the model validation is thus to compare the values of the [K+]e obtained in the simulations to those values.
- The rADCw is a biomarker which can be estimated in the brain of stroke patients thanks to diffusion-weighted (DW) magnetic resonance (MRI) imaging. It reflects the severity of the cytotoxic edema and could thus be used to predict the ischemic damage and its extension [62, 52]. The value of this ratio is supposed to be 1 in physiological conditions and is known to decrease in ischemic areas. In several studies, this value in stroke patients was shown to be between 0.75 and 0.9 in areas of moderate ischemia and between 0.5 and 0.75 in areas of severe ischemia ([18, 29, 51, 54]). This biomarker can be related to the proportions of the intracellular volumes. It was shown to be proportional to the volume of the extracellular space [62] Moreover, since the extracellular proportion was displayed to have a value of 0.2 in physiological conditions (i.e. when rADCw=1) [46], rADCw can be expressed as follows: $rADCw = 5\ (1 - f_n - f_a)$. Since f_n and f_a are two variables of the model, this ratio can be calculated for each time and for each coordinate. Another step of the model validation is thus to compare the calculated values of rADCw obtained in the simulations to the experimental values.

We present in figures 8, 9, 10, 11, 12 and 13 some results of 2D and 3D simulations, showing the values of rADCw and K^+ concentration in the extracellular space in different areas.

Let us make some biological comments about these results:

- First of all, we obtained depolarization waves after the simulation of a vessel occlusion in brain cortex and this depolarization waves spread in grey matter (i.e. in brain cortex) and not in white matter, which is consistent with MR images obtained in human brain [20].
- Concerning the potassium concentration in the extracellular space, figures 4, 6 and 8 show that this concentration reaches values such as 77 mM in the areas where the vessel was occluded, which is in accordance with the results obtained in the infarct core in some experimental studies ([36]). These figures also show that [K+]e is about 20 mM in the areas where depolarization waves are spreading, which is consistent with the values obtained in penumbra (i.e. areas of moderate ischemia, able to recover) during spreading depressions in several studies [42]. We can also notice that, in the sane areas, [K+]e remains at its physiological value (5 mM).

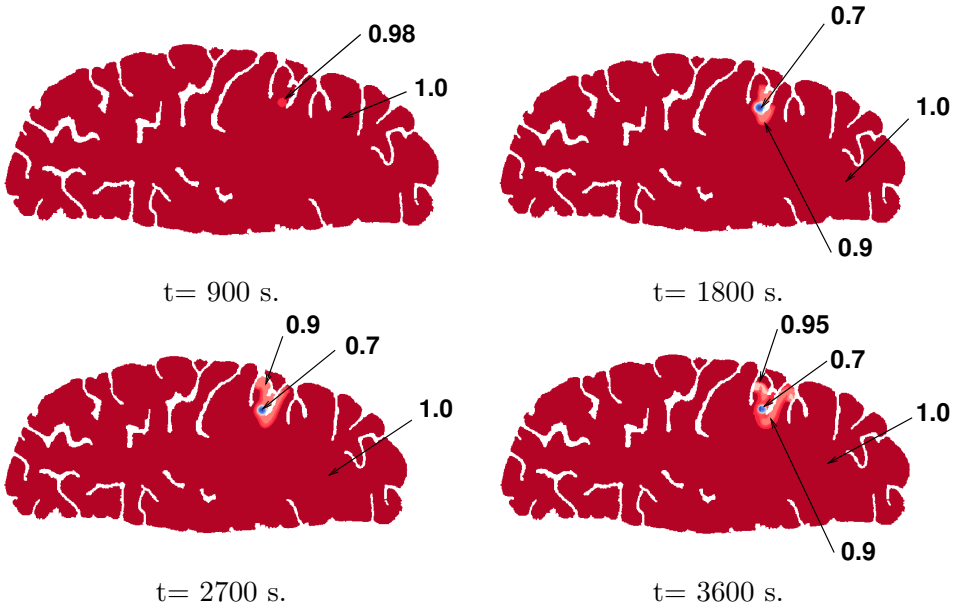


FIGURE 8. Evolution of rADCw in the extra cellular medium over one hour, 2D simulation, from left to right, top to bottom, every 15 minutes.

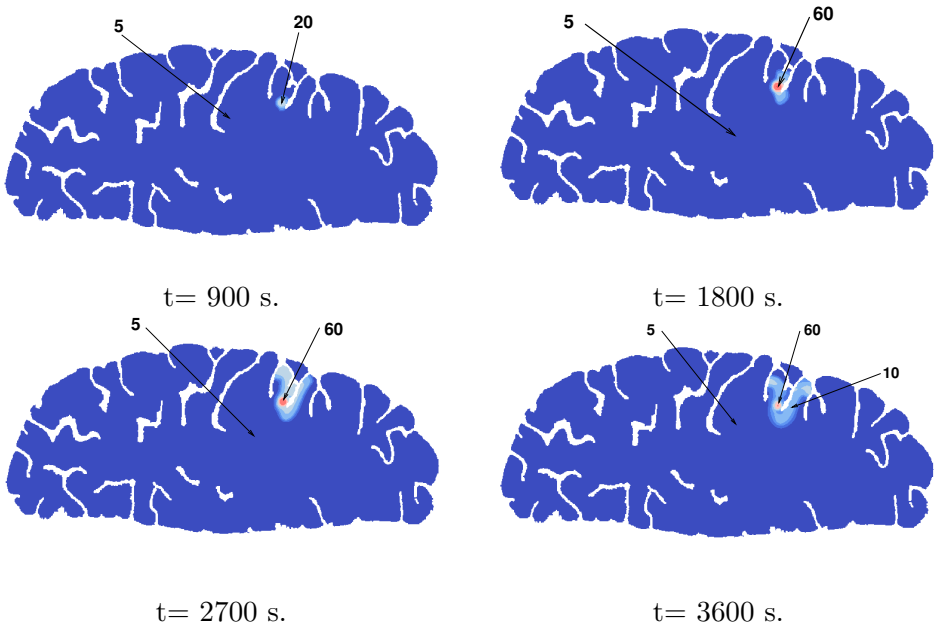


FIGURE 9. Evolution of K^+ in the extra cellular medium over one hour, 2D simulation (in millimolar (mM)), from left to right, top to bottom.

- Concerning rADCw, figures 3, 5 and 7 show that this ratio reaches values such as 0.6 in the areas where the vessel was occluded, which is in accordance with the values observed in MR images in the infarct ed core of the brain of stroke patient [29, 51, 54].

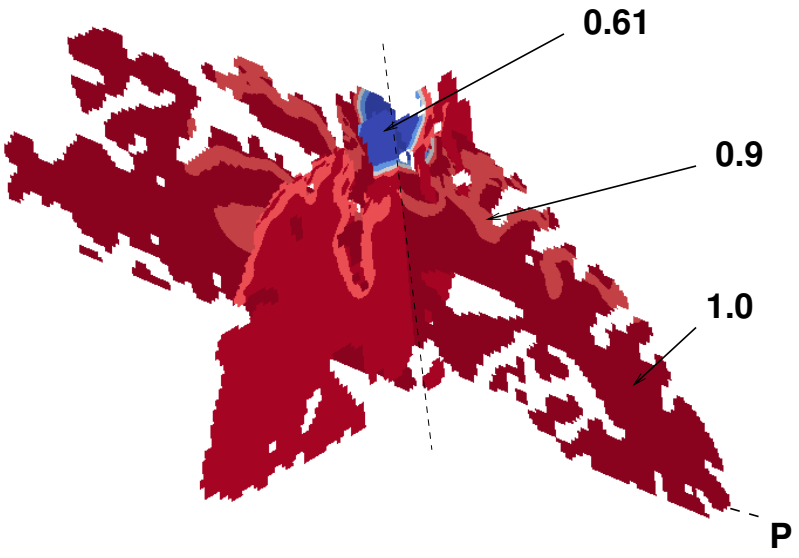


FIGURE 10. 3D simulation; cut by 2 perpendicular planes of the rADCw field at time T= 3600 seconds.

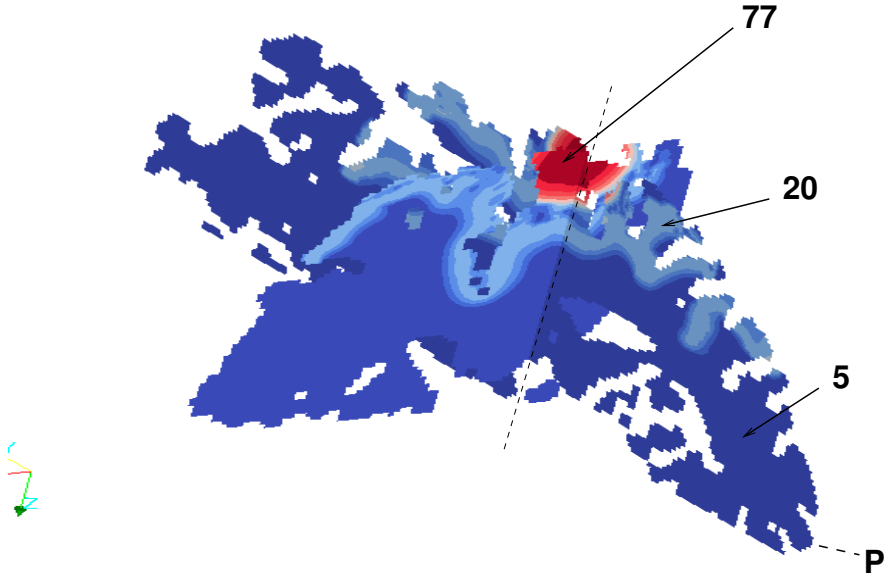
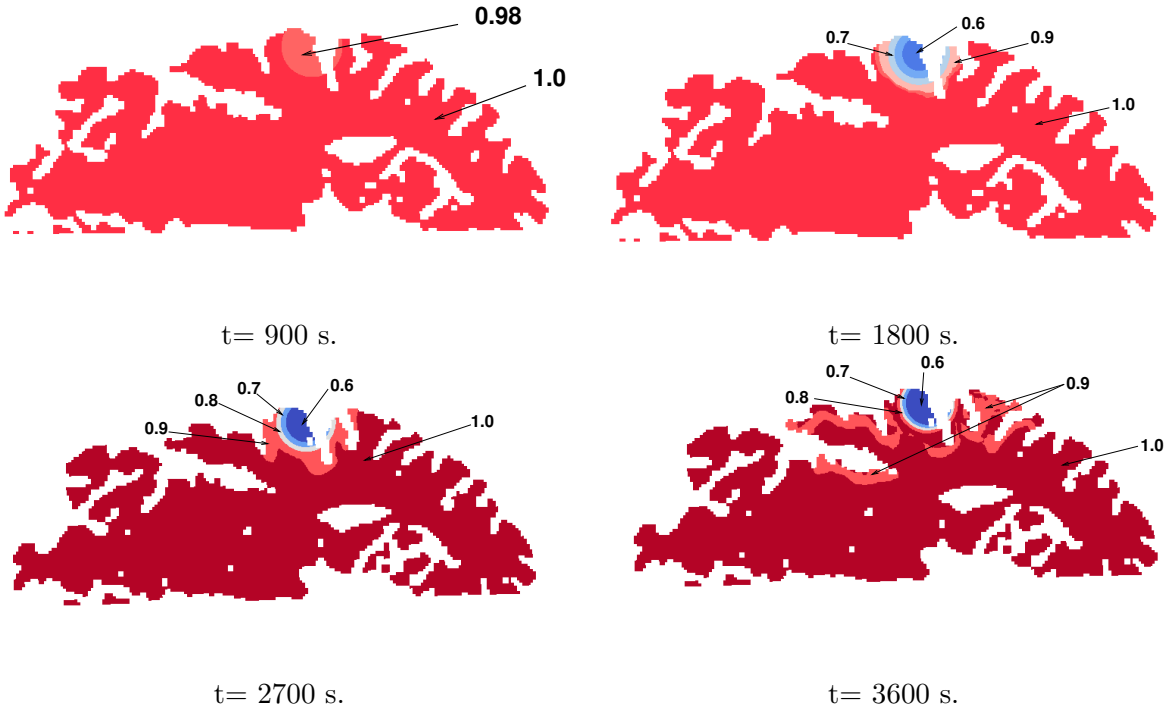


FIGURE 11. 3D simulation; cut by 2 perpendicular planes of the K^+ in the extra cellular medium field at time T= 3600 seconds (in millimolar)

These figures also show that rADCw has values between 0.75 and 0.9 in the areas where depolarization waves are spreading, which is consistent with the values obtained in



t= 900 s. t= 1800 s.

t= 2700 s. t= 3600 s.

FIGURE 12. Evolution of rADCw over one hour, 3D simulation. View in the plane **P** in figure 10.

penumbra during spreading depressions in stroke patients [29, 54]. We can also notice that, in the sane areas, rADCw remains at its physiological value (1).

To conclude, the simulation results concerning the localization of spreading depressions and the values of [K+]e and rADCw are consistent with those obtained in experimental studies or observed on MR images in stroke patients. These results give thus a first step of validation for the model and for the numerical methods used in this study.

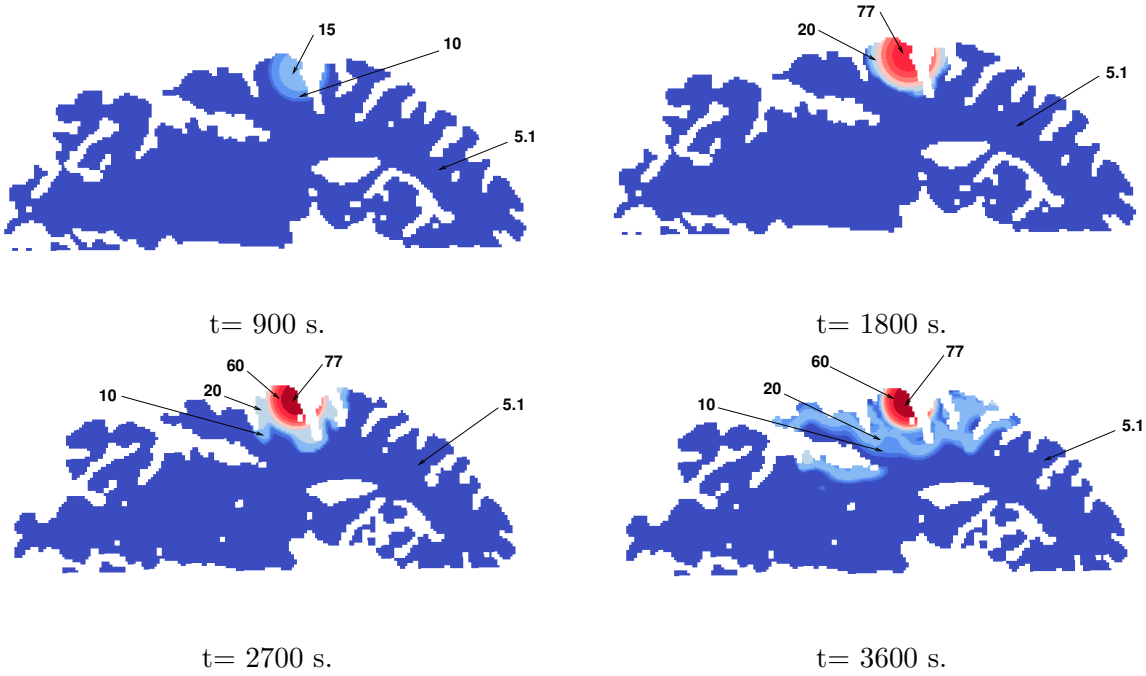


FIGURE 13. Evolution of K^+ in the extra cellular medium. View in the plane **P** in figure 10 (in mM)

7. Towards better computational efficiency and improved accuracy: Adaptive Multiresolution

In the previous simulations, we notice that the simulated waves spread at a slightly slower speed. In several studies, spreading depressions were shown to spread at a rate of several millimeters per minute [45], which is not currently the case in our simulations. In fact, it is shown in [41] that traveling waves solutions of reaction–diffusion equations can disappear in the numerical solution if the spatial discretization is too coarse; the velocity of the traveling waves is a function of the mesh size, and coarse meshes might perturb the accuracy of the computed wave velocity. In particular, in the previous 3D simulations, the mesh we can use is not fine enough to obtain a correct level of accuracy for the wave velocities. In fact, coming back to the 2D numerical test-case of section 5.2, we have seen in figure 5, that a high number of volumes is needed to reproduce accurately the phenomenon, approximately 1000 per dimension.

We can also measure the computing time of the reaction at a typical step (see figure 14[1]). Clearly the most expensive nodes are about 37 times more expensive than the less ones! But on the other hand, the overwhelming part of the nodes are not expensive ones. Actually, 79% of the nodes (the less expensive ones, which cost less than 4.5 10^6 tics) takes 60% of the computing time.

Therefore, one way to improve both the performances and the accuracy of the resolution is to use an adaptive mesh: use a fine mesh in the ischemized zone, where the solution exhibits

[1]one clock tick is about 0.35 10^{-9} second.

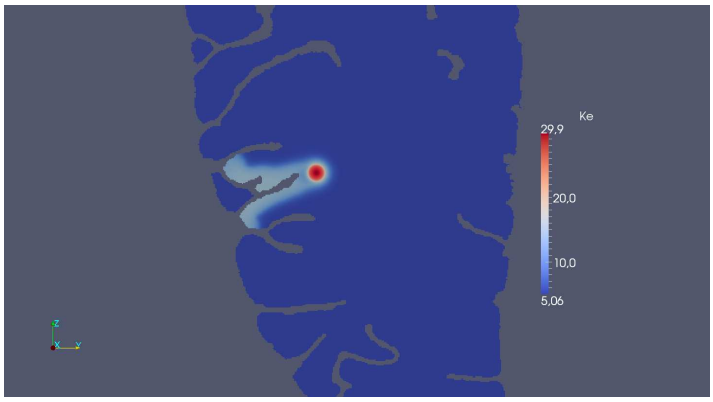


FIGURE 14. Local measure of the computational cost of the reaction (in cpu. clock tics).

large gradients and wave propagation, and a coarser mesh far from this part of the domain: the multiresolution strategy, as implemented in the code (MR) is a step towards this goal.

In order to make this more concrete than just a statement, we eventually present here a first 3D multiresolution simulation in a cube, where the finest grid available has size 256^3, since complex geometry is not yet at hand. Therefore, considering the same computing resources, the computing time is reduced by a factor of about five with respect to the fixed mesh simulation with (FM) on the finest grid, even though trying to compare two so heterogeneous codes is a very delicate task. In fact, the code (MR) considers a very low percentage ($\leq 10\%$) of 256^3, which coupled with what has been presented before, allows to explain the gain in CPU time. Nevertheless, at the current state of development of the code (MR), a lot of work remains to be done like taking into account complex geometry, improving performances of multiresolution methods by using adapted data structures and optimized routines, and finally, a high performance distributed parallelism implementation. Once this is achieved, an adaptive mesh approach shall overcome the natural limitations of accuracy and performance of even very performing strategies such as the implemented in (FM), and will allow one to solve entirely the multi-scale dynamics of this kind of phenomena.

Finally, increasing the accuracy of the discretization, of the numerical methods and even of the implementation, is not sufficient to generate more precise and predictive simulations. This should be carefully coupled with the development of a more precise modeling of coefficients and boundary conditions, and it is beyond the scope of the present paper.

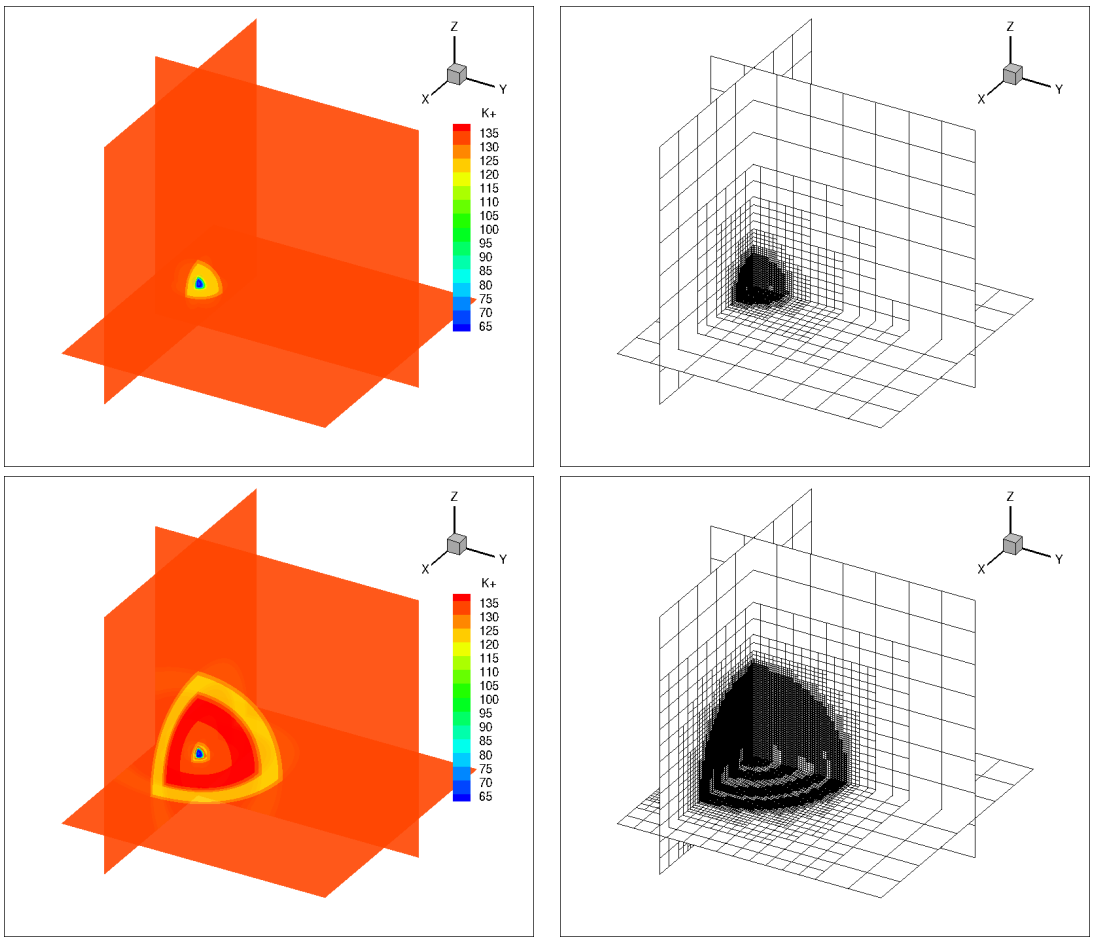


FIGURE 15. 3D simulations with (MR). $K+$ in the neurons (left) and corresponding adapted grid (right) at 1000 (top) and 3600 (bottom) seconds.

8. Conclusions and future works

We have presented for the first time numerical 3D simulations of an ischemic stroke based on the model Dronne *et* al. [21]. Results are encouraging from numerical and medical points of view. This is a first major step towards an usable tool for predicting the evolution of a stroke. The next steps are to improve both numerical performances and modeling. For this, a lot of work remains to be done from the model to practical implementations. Concerning the numerical methods, many parameters in the model are known only with a coarse approximation. Thus, numerical simulations must be conducted to explore the sensitivity of the model to these parameters. The ultimate way to improve the performances is to switch from multithreaded parallelism to distributed parallelism, on massive parallel computers.

From a medical point of view, this model is of the most importance since it could be used to simulate on a realistic human brain geometry several neuroprotective agents aimed at blocking the ischemic cascade and at reducing the ischemic damage. Since the model contains many pharmacological targets (such as ionic transporters, voltage-gated channels, channel-receptors

and stretch channels), it could be used to assess and study the effects of various therapeutic agents or associations of therapeutic agents. Moreover, since the model includes both ionic movements through the cells and their diffusion, we will be able to study the effects of these neuroprotective agents both on the severity and on the extension of the damage in each brain area. Developing powerful numerical methods are thus of the most importance to be able to simulate the time and spatial evolutions of these phenomena on a realistic human brain geometry.

Acknowledgments

This research was supported by two ANR grants (Agence Nationale pour la Recherche - French National Research Agency): **AVC in silico** (ANR-06-BYOS-0002-02 - projects leaders: M.-A. Dronne and E. Grenier - 2006-2009) and **Séchelles** (ANR-09-BLAN-0075-01 - project leader: S. Descombes - 2009-2013). We would like to thank Christian Tenaud (LIMSI - CNRS) for providing the basis of the multi-resolution kernel of MR_CHORUS, code that he developed at LIMSI for compressible Navier-Stokes equations.

References

[1] A. Abdulle. On roots and error constants of optimal stability polynomials. *BIT*, 40(1):177–182, 2000.
[2] A. Abdulle. Fourth order Chebyshev methods with recurrence relation. *SIAM J. Sci. Comput.*, 23(6):2041–2054 (electronic), 2002.
[3] A. Abdulle and A. Medovikov. Second order Chebyshev methods based on orthogonal polynomials. *Numer. Math.*, 90(1):1–18, 2001.
[4] G. Akrivis, M. Crouzeix, and C. Makridakis. Implicit-explicit multistep finite element methods for nonlinear parabolic problems. *Math. Comp.*, 67(222):457–477, 1998.
[5] P. A. Barber, R. N. Auer, A. M. Buchan, and G. R. Sutherland. Understanding and managing ischemic stroke. *Can. J. Physiol. Pharmacol.*, 79(3):283–296, 2001.
[6] J. Bell, M. J. Berger, J. Saltzman, and M. Welcome. Three-dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 15:127–138, 1994.
[7] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:67–84, 1989.
[8] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.
[9] A. Brandt. Multi-level adaptive solutions to boundary value problems. *Math. Comp.*, 31:333–390, 1977.
[10] J. C. Butcher. Implicit Runge-Kutta processes. *Math. Comp.*, 18:50–64, 1964.
[11] A. Cohen. *Wavelet methods in numerical analysis*, volume 7. Elsevier, Amsterdam, 2000.
[12] A. Cohen, S. M. Kaber, S. Müller, and M. Postel. Fully adaptive multiresolution finite volume schemes for conservation laws. *Mathematics of Computation*, 72:183–225, 2003.
[13] G. Dahlquist. A special stability problem for linear multistep methods. *Nordisk Tidskr. Informations-Behandling*, 3:27–43, 1963.
[14] J. De Keyser, G. Sulter, and P. Luiten. Clinical trials with neuroprotective drugs in acute ischaemic stroke: are we doing the right thing? *Trends in Neurosciences*, 22(12):535–40, 1999.
[15] S. Descombes and T. Dumont. Numerical simulation of a stroke: Computational problems and methodology. *Progress in Biophysics & Molecular Biology*, 97(1):40–53, 2008.
[16] S. Descombes, T. Dumont, and M. Massot. Operator splitting for stiff nonlinear reaction-diffusion systems: order reduction and application to spiral waves. In *Patterns and waves (Saint Petersburg, 2002)*, pages 386–482. AkademPrint, St. Petersburg, 2003.
[17] S. Descombes and M. Massot. Operator splitting for nonlinear reaction-diffusion systems with an entropic structure: singular perturbation and order reduction. *Numer. Math.*, 97(4):667–698, 2004.
[18] P. M. Desmond, A. C. Lovell, A. A. Rawlinson, M. W. Parsons, P. A. Barber, Qing Yang, Ting Li, D. G. Darby, R. P. Gerraty, S. M. Davis, and B. M. Tress. The value of apparent diffusion coefficient maps in early cerebral ischemia. *Am. J. Neuroradiol.*, 22(7):1260–1267, 2001.

[19] U. Dirnagl, C. Iadecola, and M.A. Moskowitz. Pathobiology of ischaemic stroke: an integrated view. *Trends in Neurosciences*, 22(9):391–397, 1999.

[20] C. Dohmen, O. W. Sakowitz, M. Fabricius, B. Bosche, T. Reithmeier, R.-I. Ernestus, G. Brinker, J. P. Dreier, J. Woitzik, A. J. Strong, and R. Graf. Spreading depolarizations occur in human ischemic stroke with high incidence. *Annals of Neurology*, 63:720–728, 2008.

[21] M.-A. Dronne, J.-P. Boissel, and E. Grenier. A mathematical model of ion movements in grey matter during a stroke. *J. of Theoretical Biology*, 240(4):599–615, 2006.

[22] M.-A. Dronne, S. Descombes, E. Grenier, and H. Gilquin. Examples of the influence of the geometry on the propagation of progressive waves. *Math. Comput. Modelling*, 49(11-12):2138–2144, 2009.

[23] M.-A. Dronne, E. Grenier, G Chapuisat, M. Hommel, and J.-P. Boissel. A modelling approach to explore some hypotheses of the failure of neuroprotective trials in ischemic stroke patient. *Progress in Biophysics & Molecular Biology*, 97:60–78, 2008.

[24] M.-A. Dronne, E. Grenier, T. Dumont, M. Hommel, and J.-P. Boissel. Role of astrocytes in grey matter during a stroke: A modelling approach. *Brain Research*, 1138:231–242, 2007.

[25] M. Duarte, M. Massot, S. Descombes, C. Tenaud, T. Dumont, V. Louvet, and F. Laurent. New resolution strategy for multi-scale reaction waves using time operator splitting, space adaptive multiresolution and dedicated high order implicit/explicit time integrators. *Submitted to SIAM, available on HAL*, 2010.

[26] T. Dumont. Zebre: Numerical software for reaction–diffusion systems. Source code and documentation at: http://math.univ-lyon1.fr/ tdumont/zebre/, 2007.

[27] A. Durukan and T. Tatlisumak. Acute ischemic stroke: Overview of major experimental rodent models, pathophysiology, and therapy of focal cerebral ischemia. *Pharmacology Biochemistry and Behavior*, 87(1):179 – 197, 2007.

[28] B. L. Ehle. A-stable methods and Padé approximations to the exponential. *SIAM J. Math. Anal.*, 4:671–680, 1973.

[29] J. Fiehler, R. Knab, J. R. Reichenbach, C. Fitzek, C. Weiller, and J. Röther. Apparent diffusion coefficient decreases and magnetic resonance imaging perfusion parameters are associated in ischemic tissue of acute stroke patients. *J. Cereb. Blood Flow & Metab.*, 97(1):54–89, 2001.

[30] B. Gottschlich-Müller and S. Müller. Adaptive finite volume schemes for conservation laws based on local multiresolution techniques. In *Hyperbolic problems: Theory, numerics, applications*, pages 385–394. M. Fey and R. Jeltsch, 1999.

[31] G. D. Graham. Tissue plasminogen activator for acute ischemic stroke. *Stroke*, 34:2847–2850, 2003.

[32] E. Grenier, M.-A. Dronne, S. Descombes, H. Gilquin, A. Jaillard, M. Hommel, and J.-P. Boissel. A numerical study of the blocking of migraine by rolando sulcus. *Progress in Biophysics & Molecular Biology*, 97(1):54–59, 2008.

[33] E. Hairer, Ch. Lubich, and M. Roche. Error of Runge-Kutta methods for stiff problems studied via differential algebraic equations. *BIT*, 28(3):678–700, 1988.

[34] E. Hairer and G. Wanner. *Solving ordinary differential equations. II*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 1996. Stiff and differential-algebraic problems.

[35] E. Hairer and G. Wanner. Stiff differential equations solved by Radau methods. *J. Comput. Appl. Math.*, 111(1-2):93–111, 1999. Numerical methods for differential equations (Coimbra, 1998).

[36] A. J. Hansen. The extracellular potassium concentration in brain cortex following ischemia in hypo- and hyperglycemic rats. *Acta Physiologica Scandinavica*, 102(3):500–544, 1978.

[37] A. Harten. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Comm. Pure and Applied Math.*, 48:1305–1342, 1995.

[38] A. L. Hodgkin and A. F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, 117(4):500–544, 1952.

[39] W. Hundsdorfer and J. Verwer. *Numerical solution of time-dependent advection-diffusion-reaction equations*, volume 33 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2003.

[40] H. Kager, W. J. Wadman, and G. G. Somjen. Simulated seizures and spreading depression in a neuron model incorporating interstitial space and ion concentrations. *J. Neurophysiol.*, 84(1):495–512, 2000.

[41] J. P. Keener. Propagation and its failure in coupled systems of discrete excitable cells. *SIAM J. Appl. Math.*, 47(3):556–572, 1987.

[42] R. P. Kraig and C. Nicholson. Extracellular ionic variations during spreading depression. *Neuroscience,*, 3(11):1045–1059, 1978.

[43] D. Lloyd-Jones, R. Adams, M. Carnethon, G. De Simone, T. B. Ferguson, K. Flegal, E. Ford, K. Furie, A. Go, K. Greenlund, N. Haase, S. Hailpern, M. Ho, V. Howard, B. Kissela, S. Kittner, D. Lackland, L. Lisabeth, A. Marelli, M. McDermott, J. Meigs, D. Mozaffarian, G. Nichol, C. O'Donnell, V. Roger, W. Rosamond, R. Sacco, P. Sorlie, R. Stafford, J. Steinberger, T. Thom, S. Wasserthiel-Smoller, N. Wong, J. Wylie-Rosett, and Y. Hong. Heart disease and stroke statistics–2009 update: A report from the American Heart Association Statistics Committee and Stroke Statistics Subcommittee. *Circulation*, 119(3):e21–181, 2009.

[44] G. I. Marchuk. Splitting and alternating direction methods. In *Handbook of numerical analysis, Vol. I*, Handb. Numer. Anal., I, pages 197–462. North-Holland, Amsterdam, 1990.

[45] H. Martins-Ferreira, M. Nedergaard, and C. Nicholson. Perspectives on spreading depression,. *Brain Research Reviews*, 31(1):215–234, 2000.

[46] C. J. McBain, S. F. Traynelis, and R. Dingledine. Regional variation of extracellular space in the hippocampus. *Science*, 249 (4969), 1990.

[47] M. A. Moskowitz, E. H. Lo, and C. Iadecola. The science of stroke: mechanisms in search of treatments. *Neuron*, 67(2):181–198, 2010.

[48] S. Müller. *Adaptive multiscale schemes for conservation laws*, volume 27. Springer-Verlag, Heidelberg, 2003.

[49] R. Radau. Étude sur les formules d'approximation qui servent a calculer la valeur numérique d'une intégrale définie. *J. Math. Pures Appl. (6)*, pages 283–336, 1880.

[50] K. Revett, E. Ruppin, S. Goodall, and J. A. Reggia. Spreading depression in focal ischemia: a computational study. *J. Cereb. Blood Flow & Metab.*, 18(9):998–1007, 1998.

[51] L. Røhl, L. Østergaard, C. Z. Simonsen, P. Vestergaard-Poulsen, G. Andersen, M. Sakoh, D. Le Bihan, and C. Gyldensted. Viability thresholds of ischemic penumbra of hyperacute stroke defined by perfusion-weighted MRI and apparent diffusion coefficient. *Stroke*, 32(5):1140–1146, 2001.

[52] C. Rosso, N. Hevia-Montiel, S. Deltour, E. Bardinet, D. Dormont, S. Crozier, S. Baillet, and Y. Samson. Prediction of infarct growth based on apparent diffusion coefficients: penumbral assessment without intravenous contrast material. *Radiology*, 95(3):450–458, 2009.

[53] O. Roussel, K. Schneider, A. Tsigulin, and H. Bockhorn. A conservative fully adaptive multiresolution algorithm for parabolic PDEs. *J. Comput. Phys.*, 188(2):493–523, 2003.

[54] M. Sakoh, L. Østergaard, A. Gjedde, L. Røhl, P. Vestergaard-Poulsen, D. F. Smith, D. Le Bihan, S. Sakaki, and C. Gyldensted. Prediction of tissue survival after middle cerebral artery occlusion based on changes in the apparent diffusion of water. *J. Neurosurg.*, 95(3):450–458, 2001.

[55] L. F. Shampine, B. P. Sommeijer, and J. G. Verwer. IRKC: an IMEX solver for stiff diffusion-reaction PDEs. *J. Comput. Appl. Math.*, 196(2):485–497, 2006.

[56] H. K. Shin, A. K. Dunn, P. B. Jones, D. A. Boas, M. A. Moskowitz, and C. Ayata. Vasoconstrictive neurovascular coupling during focal ischemic depolarizations. *J. Cereb. Blood Flow & Metab.*, 26(8):1018–1030, 2005.

[57] G. G. Somjen. Mechanisms of spreading depression and hypoxic spreading depression-like depolarization. *Physiol. Rev.*, 81(3), 2001.

[58] B. P. Sommeijer, L. F. Shampine, and J. G. Verwer. RKC: an explicit solver for parabolic PDEs. *J. Comput. Appl. Math.*, 88(2):315–326, 1998.

[59] B. P. Sommeijer and J. G. Verwer. *A performance evaluation of a class of Runge-Kutta-Chebyshev methods for solving semidiscrete parabolic differential equations*. Afdeling Numerieke Wiskunde [Department of Numerical Mathematics], 91. Mathematisch Centrum, Amsterdam, 1980.

[60] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506–517, 1968.

[61] P. J. Van der Houwen and B. P. Sommeijer. On the internal stability of explicit, m-stage Runge-Kutta methods for large m-values. *Z. Angew. Math. Mech.*, 60(10):479–485, 1980.

[62] H. B. Verheul, R. Balazs, J. W. Berkelbach van der Sprenkel, C. A. F. Tulleken, K. Nicolay, K. S. Tamminga, and M. van Lookeren Campagne. Comparison of diffusion-weighted MRI with changes in cell volume in a rat model of brain injury. *NMR Biomed.*, 7(1–2):96–100, 1994.

[63] J. G. Verwer. Explicit Runge-Kutta methods for parabolic partial differential equations. *Appl. Numer. Math.*, 22(1-3):359–379, 1996. Special issue celebrating the centenary of Runge-Kutta methods.

[64] J. G. Verwer and B. P. Sommeijer. An implicit-explicit Runge-Kutta-Chebyshev scheme for diffusion-reaction equations. *SIAM J. Sci. Comput.*, 25(5):1824–1835 (electronic), 2004.

[65] J. G. Verwer, B. P. Sommeijer, and W. Hundsdorfer. RKC time-stepping for advection-diffusion-reaction problems. *J. Comput. Phys.*, 201(1):61–79, 2004.
[66] N. G. Wahlgren and N. Ahmed. Neuroprotection in cerebral ischaemia: facts and fancies–the need for new approaches. *Cerebrovasc. Dis.*, 17:155–166, 2004.
[67] A. Williams and W. Kempf. *The boost thread library. Portable C++ multi-threading.* http://www.boost.org/.

(T. Dumont, V. Louvet) Institut Camille Jordan - UMR CNRS 5208, Université de Lyon, Université Lyon 1, INSA de Lyon 69621, Ecole Centrale de Lyon, 43 Boulevard du 11 novembre 1918, 69622 Villeurbanne Cedex, France and Project-team NUMED, INRIA, Ecole Normale supérieure de Lyon, 46 allée d'Italie, 69007 Lyon Cedex 07, France ({tdumont,louvet}@math.univ-lyon1.fr).

(M. Duarte, M. Massot) Laboratoire EM2C - UPR CNRS 288, Ecole Centrale Paris, Grande Voie des Vignes, 92295 Chatenay-Malabry Cedex, France ({max.duarte,marc.massot}@em2c.ecp.fr).

(S. Descombes) Laboratoire J. A. Dieudonné - UMR CNRS 6621, Université de Nice - Sophia Antipolis, Parc Valrose, 06108 Nice Cedex 02, France and Project-team NACHOS, INRIA Sophia Antipolis - Méditerranée research center, 2004 Route des Lucioles, BP 93, 06902 Sophia Antipolis Cedex, France. (sdescomb@unice.fr).

(M.A. Dronne) Université de Lyon, Université Lyon 1, ISPB - Faculté de Pharmacie de Lyon 69003 Lyon, France and Project-team NUMED, INRIA, Ecole normale supérieure de Lyon, 46 allée d'Italie, 69007 Lyon Cedex 07, France (marie-aimee.dronne@recherche.univ-lyon1.fr).