New resolution strategy for multi-scale reaction waves using time operator splitting, space adaptive multiresolution and dedicated high order implicit/explicit time integrators

Max Duarte, Marc Massot, Stéphane Descombes, Christian Tenaud, Thierry Dumont, Violaine Louvet, Frédérique Laurent

New resolution strategy for multi-scale reaction waves using time operator splitting, space adaptive multiresolution and dedicated high order implicit/explicit time integrators

Max Duarte[a,*], Marc Massot[a], Stéphane Descombes[b], Christian Tenaud[c], Thierry Dumont[d], Violaine Louvet[d], Frédérique Laurent[a]

[a] *Laboratoire EM2C - UPR CNRS 288, Ecole Centrale Paris, Grande Voie des Vignes, 92295 Chatenay-Malabry Cedex, France*

[b] *Laboratoire J. A. Dieudonné - UMR CNRS 6621, Université de Nice - Sophia Antipolis, Parc Valrose, 06108 Nice Cedex 02, France*

[c] *LIMSI - CNRS, B.P. 133, Campus d'Orsay, 91403 Orsay Cedex, France*

[d] *Université de Lyon - Université Lyon 1 - INSA de Lyon, 69621 - Ecole Centrale de Lyon - Institut Camille Jordan, UMR CNRS 5208 - 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex, France*

Abstract

In this paper, we tackle the numerical simulation of reaction-diffusion equations modeling multi-scale reaction waves. This type of problems induces peculiar difficulties and 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 large spatial gradients in the reactive fronts which are spatially very localized. In a series of previous studies, the numerical analysis of operator splitting techniques has been conducted and such an approach has shown a great potential in the framework of reaction-diffusion and convection-diffusion-reaction systems. However, even if a firm theoretical background is available, an optimal strategy for high performance numerical simulation is still needed. In this paper, we introduce a new strategy for reaction-diffusion systems based on time operator splitting in the context of very localized and very stiff reaction fronts. It provides an optimal combination of adaptive spatial multiresolution, implicit resolution of reaction and explicit resolution of diffusion. The optimality is reached in terms of the choice of the operator splitting time step which, in the framework of self-similar reaction waves, allows a very good combination of the various dedicated solvers used in the proposed strategy. The computational efficiency is then evaluated through the numerical simulation of configurations which were so far out of reach of standard methods in the field of nonlinear chemical dynamics for spiral waves and scroll waves as an illustration.

Keywords: Reaction-diffusion, adaptive multiresolution, operator splitting, multi-scale waves
2000 MSC: 65M50, 35K57, 65M08

*Corresponding author
Email addresses: max.duarte@em2c.ecp.fr (Max Duarte), marc.massot@em2c.ecp.fr (Marc Massot), sdescomb@unice.fr (Stéphane Descombes), tenaud@limsi.fr (Christian Tenaud), tdumont@math.univ-lyon1.fr (Thierry Dumont), louvet@math.univ-lyon1.fr (Violaine Louvet), frederique.laurent@em2c.ecp.fr (Frédérique Laurent)

1. Introduction

Numerical simulations of multicomponent reactive flows are commonly used for modeling purposes in many applications such as combustion [36, 21, 55, 49, 54], chemical vapor deposition [37], and air pollution modeling [68, 48, 52]. The important development of the numerical strategies in these and in other fields such as nonlinear chemical dynamics for excitable media [5, 31, 11] or biomedical engineering [40, 23, 32] is mainly due to the constant increase of the computer power (see for instance [33] for a recent review of methods applied to turbulent combustion). Most of the time, the multicomponent character featured in these applications yields a large set of unknowns which satisfy a system of partial differential equations involving a very nonlinear "chemical" source term. Thus, in this study, we focus on reaction-diffusion systems which is the subsystem that normally involves the strongest difficulties in terms of stiffness in multi-scale phenomena, even if convection plays also a crucial role. In fact, the developments provided in this contribution are meant in order to be extended to convecting cases.

In general, these models raise several difficulties created by the large number of unknowns and the wide range of temporal scales due to large and detailed chemical kinetic mechanisms, as well as by steep spatial gradients or large higher order derivatives associated with very localized fronts of high chemical activity. Another stumbling block for 3D simulations is the unreasonable memory requirements for such multi-scale problems. Therefore, there are several strategies in order to treat the induced stiffness for time dependent problems. The most natural idea is to use dedicated numerical methods and to solve the complete models where diffusion, reaction and eventually convection are coupled together. In this context, one approach consists in the resolution of strongly coupled nonlinear systems with a fully implicit method, using eventually modified Newton methods for ill conditioned problems [29, 30, 60]. However, the computational cost as well as the huge memory requirements, even if adaptive grid are used, make this strategy difficult to handle. Yet another possibility is to use semi-implicit or linearized implicit methods instead (see [19] and references therein). Nevertheless, these methods have shown to suffer of strong stability restrictions, especially in the context of exothermic reactions [20, 19] and are not adapted to reactive flows with very fast temporal scales.

Another numerical strategy first introduced in [4] is to combine implicit and explicit schemes to discretize nonlinear evolution problems in time. In fact, further studies into [61, 73] settled the appropriate numerical background for these methods called IMEX, which in particular might be conceived to solve stiff nonlinear problems as presented in [74, 59]. These methods are usually very efficient. Nevertheless, the feasibility of utilizing dedicated implicit solvers over a discretized domain become soon critical when treating large computational domains. In fact, complementary numerical strategies must be developed in order to overcome these computational restrictions.

Finally, the only case where we can use the method of lines in conjunction with a dedicated stiff ODE solver (so that the time discretization and the spatial discretization are disconnected) is when we need a reference solution for validation and study purposes necessarily restricted to low dimensional configurations with not many unknowns.

From another point of view, we might also consider the possibility to use reduced models where the fastest chemical scales have been previously relaxed [43]. These simplified models provide reasonable predictions when the fastest characteristic chemical times are small in comparison with the flow time, and the associated computational costs are significantly reduced in comparison with comprehensive chemical models. A large literature has been devoted to this subject. There has been a tremendous effort in creating efficient and predictive numerical me-

thods in order to define and solve reduced systems of equations for combustion and air pollution modeling applications. In particular, the derivation of the reduced model is usually accessible when the system is well-partitioned and the fast scales have been isolated [63, 65]. In this case, a rigorous singular perturbation analysis can be conducted even in the context of nonlinear source terms for numerical analysis purposes [51, 27]. Nevertheless, the identification of these fast scales in terms of reaction rates or species, which can change with time, relies on sensitivity analysis which is most of the time difficult to conduct and justify in realistic configurations. Hence, it reveals the need for other strategies which do not rely on the knowledge of the fast scales.

It is then natural to envision a compromise, since the fully coupled problem is most of the time out of reach and the reduced model does not always imply straightforward implementations. In this context, splitting methods [50] also called fractional step methods [76, 69, 70] have been well known for a long time and there exists a large literature showing the efficiency of such methods for evolution problems. Yet from a theoretical point of view, they represent a suitable framework to design even higher order methods for the integration in time of such problems [22, 58]. In the practice, it is firstly necessary to decouple numerically the reaction part which can bring in some very fast time scales from the rest of the physical phenomena like convection, diffusion or both for which there also exist dedicated numerical methods. Nevertheless it is fundamental to keep some information about the good description of the global physical coupling and that is why a rigorous numerical analysis is required.

For instance, in the combustion domain, several numerical studies on multi-scale phenomena were conducted that already coupled splitting techniques with dedicated time integration methods such as Runge-Kutta-Chebyshev (RKC) [72] methods in [54] or local refined mesh strategies in [21]. Nevertheless, in order to perform such numerical simulations very sophisticated algorithms were constructed with splitting time steps defined by the fastest time scales of the problem in order to guarantee the classical numerical behavior of the splitting schemes. In fact, several works [43, 75, 64, 19] proved that the classical analysis of such methods fails in presence of scales much faster than the splitting time step and motivated more rigorous studies for these stiff configurations [27]. Moreover, complementary works described also the numerical behavior of these methods when spatial multi-scale phenomena arise mainly as a consequence of large spatial gradients [24], so that the influence of both spatial and time related stiffness has been analyzed in detail for not arbitrarily small splitting time steps.

We therefore aim at conceiving a very compact and simple numerical strategy based on these theoretical results for splitting schemes in order to guarantee a more accurate knowledge and control of the time and spatial errors in the context of splitting time steps always much larger than the fastest time scales. Keeping this in mind, on the one hand, a high order method like Radau5 [43], based on implicit Runge-Kutta schemes for stiff ODEs, solves the reaction term using adaptive time integration tools and highly optimized linear systems solvers. And on the other hand, another high order method like Rock4 [2], based on explicit stabilized Runge-Kutta schemes, solves the diffusion problem.

With the choice of the time integration solvers properly justified, the last ingredient of the algorithm is a mesh refinement technique, being aware of the interest of adaptive mesh techniques for problems exhibiting spatial multi-scale phenomena such as locally steep gradients or shock-like structures in general. In the same context, another valid numerical strategy to treat this kind of problems consists in adjusting artificially the spatial scales of interest present in the phenomenon. For example, in the combustion domain, the thickened flame model [17, 13] proposes to artificially thicken the flame front in order to solve it on a feasible LES mesh without

altering the flame speed. Nevertheless, in this work we adopt a different approach with the main goal of solving all the spatial scales without "scaling" the problem.

Historically, adaptive methods like Multi-Level Adaptive Techniques (MLAT) [10] 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 of high truncation errors are found. Then, adaptive multiresolution methods, based on Harten's pioneering work [45], have been developed for 1D and 2D hyperbolic conservation laws [16, 38] and then extended to 3D parabolic problems [57]. A high data compression might be achieved with all these methods, nevertheless, one particular advantage of the adaptive multiresolution techniques is that the numerical analysis of the errors has already been conducted [45, 16], so that they can be previously estimated.

Considering all the features of the selected numerical methods, an optimal combination of the adaptive spatial multiresolution, the implicit resolution of the reaction and the explicit resolution of the diffusion is achieved taking into account the theoretical background of the splitting techniques. In this work, we are mainly focused on time and space multi-scale reaction waves for which the choice of a suitable splitting time step remains a delicate task. Thus, considering the propagating nature of the studied phenomenon and based on theoretical considerations, a simple and efficient heuristic is developed in order to overcome this difficulty. We thus prove that we have designed a very efficient numerical strategy which is able to capture 3D large simulations with feasible computational cost and memory requirements. Nevertheless, we search to utterly identify the different aspects that need to be improved, expecting this work to become a starting point towards optimal implementations. In particular, the extension to complete convection-reaction-diffusion systems is foreseen based on the proposed numerical method.

The paper is organized as follows : in a first part, we present the various building blocks which will be used in the numerical strategy. The latter is introduced in the second part, as well as the corresponding algorithm and associated key issues. In the last part of this work, we first present the three species reaction diffusion system, which is a model of the Belousov-Zhabotinsky reaction. We then conduct a series of detailed and careful numerical simulations in a one-dimensional case in order to illustrate the key issues highlighted in the second part and prove the good behavior of the proposed strategy. Finally, the potential of the method is illustrated in the framework of two- and three-dimensional simulations which allows a detailed discussion of the capability and performance of the method.

2. Operator splitting, time integrators, adaptive multiresolution and numerical strategy

This first section contains a brief introduction to each and every numerical method considered in this work. Therein, general splitting techniques as well as the selected reaction and diffusion time integrators are presented in this order and in an independent way. Then, the principal features of the adaptive multiresolution method is described to finally introduce the proposed numerical strategy.

2.1. Operator splitting

As it was mentioned in the introduction, splitting methods represent an easy and convenient way to simulate multi-scale phenomena. Moreover, these techniques are gaining a lot of interest because of the constant development of sophisticated time integrator tools that allow even more

dedicated and efficient solvers for each split subproblem. In this context, we recall that multiscale phenomena can be usually modeled by general reaction-diffusion systems of type

$$\left.\begin{array}{ll} \partial_t \mathbf{u} - \partial_{\mathbf{x}} \left(\mathbf{D}(\mathbf{u}) \partial_{\mathbf{x}} \mathbf{u}\right) = \mathbf{f}\left(\mathbf{u}\right), & \mathbf{x} \in \mathbb{R}^d,\ t > 0, \\ \mathbf{u}(0, \mathbf{x}) = \mathbf{u}_0(\mathbf{x}), & \mathbf{x} \in \mathbb{R}^d, \end{array}\right\} \tag{1}$$

where $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^m$ and $\mathbf{u} : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^m$, with the diffusion matrix $\mathbf{D}(\mathbf{u})$, which is a tensor of order $d \times d \times m$.

In order to simplify the presentation, we consider problem (1) with linear diagonal diffusion, in which case the elements of the diffusion matrix are written as $D_{ijk}(\mathbf{u}) = D_k \delta_{ij}$, so that the diffusion operator reduces to the heat operator with scalar diffusion coefficient D_k for component u_k of $\mathbf{u}$, $k = 1, \ldots, m$. In any case, the proposed numerical strategy normally deals with general problem (1). However, performing a fine spatial discretization, we obtain the semi-discretized initial value problem

$$\left.\begin{array}{l} \dfrac{d\mathbf{U}}{dt} - \mathbf{B}\,\mathbf{U} = \mathbf{F}\left(\mathbf{U}\right), \quad t > 0, \\ \mathbf{U}(0) = \mathbf{U}^0, \end{array}\right\} \tag{2}$$

where $\mathbf{B}$ corresponds to the discretization of the Laplacian operator with the coefficients D_k within; $\mathbf{U}$ and $\mathbf{F}\left(\mathbf{U}\right)$ are arranged component-wise all over the discretized spatial domain. Performing the classical decoupling of the diffusion and reaction parts of (2), we denote $\mathcal{X}^{\Delta t}(\mathbf{U}^0)$ as the numerical solution of the discretized diffusion equation

$$\frac{d\mathbf{U}_D}{dt} - \mathbf{B}\,\mathbf{U}_D = 0, \quad t > 0, \tag{3}$$

with initial data $\mathbf{U}_D(0) = \mathbf{U}^0$ after an integration time step Δt. We also denote by $\mathcal{Y}^{\Delta t}(\mathbf{U}^0)$ the numerical solution of the reaction part,

$$\frac{d\mathbf{U}_R}{dt} = \mathbf{F}\left(\mathbf{U}_R\right), \quad t > 0, \tag{4}$$

with initial data $\mathbf{U}_R(0) = \mathbf{U}^0$.

The two Lie approximation formulae of the solution of system (2) are then defined by

$$\mathcal{L}_1^{\Delta t}(\mathbf{U}^0) = \mathcal{X}^{\Delta t}\mathcal{Y}^{\Delta t}(\mathbf{U}^0), \quad \mathcal{L}_2^{\Delta t}(\mathbf{U}^0) = \mathcal{Y}^{\Delta t}\mathcal{X}^{\Delta t}(\mathbf{U}^0), \tag{5}$$

while the two Strang approximation formulae [66, 67] of the solution of system (2) are given by

$$\mathcal{S}_1^{\Delta t}(\mathbf{U}^0) = \mathcal{X}^{\Delta t/2}\mathcal{Y}^{\Delta t}\mathcal{X}^{\Delta t/2}(\mathbf{U}^0), \quad \mathcal{S}_2^{\Delta t}(\mathbf{U}^0) = \mathcal{Y}^{\Delta t/2}\mathcal{X}^{\Delta t}\mathcal{Y}^{\Delta t/2}(\mathbf{U}^0), \tag{6}$$

where Δt is the splitting time step.

It is well known that Lie formulae (5) (resp. Strang formulae (6)) are an approximation of order 1 (resp. 2) of the exact solution of (2) in the case where $\mathcal{X}^{\Delta t}$ and $\mathcal{Y}^{\Delta t}$ are the exact solutions $X^{\Delta t}$ and $Y^{\Delta t}$ of problems (3) and (4). Then, appropriate numerical approximations of $X^{\Delta t}$ and $Y^{\Delta t}$ are required in order to compute Lie and Strang formulae with the prescribed order.

Higher order splitting configurations are also possible. Nevertheless, the order conditions for such composition methods state that either negative time substeps or complex coefficients are necessary (see [42]). These normally imply important stability restrictions and more sophisticated numerical implementations. In the particular case of negative time steps, they are completely undesirable for PDEs that are ill-posed for negative time progression.

However, the key point of a splitting strategy is that dedicated and often different time integration methods might be considered, taking advantage of the particular features of each subproblem (3) and (4). Keeping this in mind, we will first present Radau5 and then Rock4 as respectively the selected numerical solvers of the reaction and the diffusion time evolution problem.

2.2. Time integration of the reaction problem

For $t > t_0$, let us consider the initial value problem

$$\frac{d\mathbf{u}}{dt} = \mathbf{f}(t, \mathbf{u}), \quad \mathbf{u}(t_0) = \mathbf{u}_0, \tag{7}$$

with some integer $m > 0$, $\mathbf{u} : \mathbb{R} \to \mathbb{R}^m$ and a general nonlinear function $\mathbf{f} : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$, which is equivalent to the reaction problem for general system (1) at a fixed point $\mathbf{x}$. Taking into account a general s-stage Runge-Kutta method, after an integration time step Δt, the solution $\mathbf{u}(t_0 + \Delta t)$ of problem (7) might be approximated by $\mathbf{u}_1$, which is given by

$$\begin{aligned} \mathbf{k}_i &= \mathbf{u}_0 + \Delta t \sum_{j=1}^{s} a_{ij} \mathbf{f}\left(t_0 + c_j \Delta t, \mathbf{k}_j\right), \quad i = 1, \ldots, s, \\ \mathbf{u}_1 &= \mathbf{u}_0 + \Delta t \sum_{j=1}^{s} b_j \mathbf{f}\left(t_0 + c_j \Delta t, \mathbf{k}_j\right), \end{aligned} \tag{8}$$

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

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

Moreover, a Runge-Kutta method is of order p if it satisfies

$$\sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q}, \quad q = 1, \ldots, p \tag{9}$$

and for $p \le \eta + \zeta + 1$ and $p \le 2\eta + 2$,

$$\begin{aligned} \sum_{j=1}^{s} a_{ij} c_j^{q-1} &= \frac{c_i^q}{q}, && i = 1, \ldots, s,\ q = 1, \ldots, \eta, \\ \sum_{i=1}^{s} b_i c_i^{q-1} a_{ij} &= \frac{b_j}{q}(1 - c_j^q), && j = 1, \ldots, s,\ q = 1, \ldots, \zeta, \end{aligned} \tag{10}$$

as it was proven by Butcher [12].

If we now consider the stability features, a classical analysis based on the *Dahlquist test equation* [18]

$$y' = \lambda y, \quad y(0) = 1, \tag{11}$$

allows to define the *stability function* $R : \mathbb{C} \to \mathbb{C}$ of a method as the numerical solution of (11) given by the method itself after one time step Δt. In the case of a Runge-Kutta method, solving (11) with $y(0) = y_0$ with the scheme given by (8) leads to

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

where R is a rational function that becomes a polynomial for explicit Runge-Kutta schemes. In the same context, the set

$$S := \{z \in \mathbb{C} \ : \ |R(z)| \leq 1\} \tag{12}$$

is called the *stability domain* of the method and a particular method will remain stable as long as $z \in S$. Notice that in a general case, problem (7) can be linearized and supposing a diagonalizable Jacobian matrix $\mathbf{J} = \partial \mathbf{f}/\partial \mathbf{u}$, an analogous problem to that of Dalquist can be obtained with the complex eigenvalues $\lambda_i, i = 1, \cdots, m$ of $\mathbf{J}$. As a consequence, if (7) is a stiff system of ODEs (see [43] 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 transient phase of the solution, whenever the initial solution does not belong to a partial equilibrium manifold where the fast scales are already relaxed. Considering (12), this implies that either 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

$$S \subset \{z \in \mathbb{C} \ : \ \Re z \leq 0\}, \tag{13}$$

are usually preferred, so that prohibitive small 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 [56], Ehle has constructed a family of formulae named 'Radau IIA' in which implicit Runge-Kutta schemes are generated by collocation at the nodes of these quadrature formulae [34]. 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 (9) and (10). For instance, it gives the backward Euler method for $s = 1$, but even higher order methods are possible. Actually, the case $s = 3$ for which the Butcher array and the corresponding stability function are given by

$$\begin{array}{c|ccc}
\frac{4-\sqrt{6}}{10} & \frac{88-7\sqrt{6}}{360} & \frac{296-169\sqrt{6}}{1800} & \frac{-2+3\sqrt{6}}{225} \\
\frac{4+\sqrt{6}}{10} & \frac{296+169\sqrt{6}}{1800} & \frac{88+7\sqrt{6}}{360} & \frac{-2-3\sqrt{6}}{225} \\
1 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9} \\
\hline
 & \frac{16-\sqrt{6}}{36} & \frac{16+\sqrt{6}}{36} & \frac{1}{9}
\end{array}, \quad R(z) = \frac{1 + \frac{2z}{5} + \frac{z^2}{20}}{1 - \frac{3z}{5} + \frac{3z^2}{20} - \frac{z^3}{60}}, \tag{14}$$

is the Radau IIA method on which Radau5 is based [43, 44]. 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 [41]. 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 by (14). 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 (see [43] for further details).

2.3. Time integration of the diffusion problem

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 stiff problems with a Jacobian matrix involving (possibly large) eigenvalues close to the real negative axis. This is the particular case of the discretized diffusion problem previously considered (3) for which the real negative eigenvalues increase with finer spatial discretizations. Therefore, instead of A-stable but time consuming implicit procedures, *stabilized* explicit Runge-Kutta methods should be preferred. As a matter of fact, such methods feature extended stability domain along the negative real axis and thus, are dedicated to these type of problems [72].

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

$$\begin{aligned} &R_s(z) = 1 + z + \cdots + \frac{z^p}{p!} + \sum_{p+1}^{s} \alpha_{i,s} z^i, \\ &|R_s(z)| \leq 1 \quad \text{for} \quad z \in [-\ell_s, 0], \end{aligned} \tag{15}$$

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

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

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 [46] and references therein). For example, RKC methods proposed by Sommeijer, Shampine and Verwer in [61] 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 [71] and [62].

Nevertheless, with this background and based on the study of optimal stability polynomials satisfying (15) in [1], Medovikov and Abdulle proposed in [3] to approximate R_s by

$$\tilde{R}_s(z) = \tilde{w}_p(z)\tilde{P}_{s-p}(z) \tag{16}$$

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 (15) with an approximated $\tilde{\ell}_s$ as close as possible to optimal ℓ_s. Based on the same ideas, the fourth order case of (16) was proposed by Abdulle in [2] and gave birth to the 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 WP. The first method, denoted by 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 the explicit Runge-Kutta scheme

$$\begin{aligned} \mathbf{k}_0 &:= \mathbf{U}^0, \\ \mathbf{k}_1 &:= \mathbf{U}^0 + \Delta t \mu_1 \mathbf{B}\mathbf{k}_0, \\ \mathbf{k}_i &:= \Delta t \mu_i \mathbf{B}\mathbf{k}_{i-1} - \upsilon_i \mathbf{k}_{i-1} - \kappa_i \mathbf{k}_{i-2}, \quad i = 2, \ldots, s-4, \\ \mathbf{U}_1 &:= \mathbf{k}_{s-4}, \end{aligned} \tag{17}$$

applied for example to the diffusion problem (3). The corresponding coefficients in the Butcher's array of P (see Table 1) are then defined considering (17) and the general scheme (8) as showed in [2]. Notice that necessarily $s \geq 5$ and that scheme (17) applied to the Dahlquist test equation (11) 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 fourth order conditions established for method WP and the previously calculated coefficients of P. Finally, construction of the Rock4 method is based on method WP with stability function given by (16) with $p = 4$.

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

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

With the choice of the time integration methods of each subproblem, the reaction (4) and the diffusion (3), we can now consider the last element of the algorithm which is the chosen method of grid adaptation, the space adaptive multiresolution technique.

2.4. Adaptive multiresolution

In this subsection, only the basis of the adaptive multiresolution strategy will be presented. Further details can be found in the literature and in the given references. In fact, for an overview

on adaptive multiresolution techniques, we refer to the books of Cohen [14] and Müller [53]. Nevertheless the following concepts are fundamental for our numerical strategy.

2.4.1. Basis of a multiresolution representation

Let us consider that we have a set of nested spatial grids, from the coarsest to the finest one. A multiresolution transformation allows to represent a discretized function as values on a coarser grid plus a series of local estimates at different levels of such nested grids. These estimates are related to the difference of values obtained by inter-level transformations : which are the projection and prediction operators. The theoretical background of such configuration (see [16]) states that wavelet coefficients can be then defined as prediction errors, and they will retain these estimates when going from a coarse to a finer grid. Hence, the main idea is to use the decay of the wavelet coefficients to obtain information on local regularity of the solution : lower wavelet coefficients are associated to local regular spatial configurations and *vice-versa*.

To clarify these ideas, we briefly introduce some fundamental concepts. Let us consider nested finite volume discretizations of general problem (1) with only one component, $m = 1$: that is, for $j = 0, 1, \cdots, J$ from the coarsest to the finest grid, we consider regular disjoint partitions (cells) $(\Omega_\gamma)_{\gamma \in S_j}$ of an open subset $\Omega \subset \mathbb{R}^d$, such that each Ω_γ, $\gamma \in S_j$, is the union of a finite number of cells Ω_μ, $\mu \in S_{j+1}$, and thus, S_j and S_{j+1} are consecutive embedded grids. We are considering here only Cartesian grids, however the method has already been developed on both unstructured grids [14] and general coordinate systems [53].

Thus, we denote $\mathbf{U}_j := (u_\gamma)_{\gamma \in S_j}$ as the representation of $\mathbf{U}$ on the grid S_j where u_γ represents the cell-average of $u : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$ in Ω_γ. For instance, in the 1D configuration ($d = 1$) :

$$u_\gamma := |\Omega_\gamma|^{-1} \int_{\Omega_\gamma} u(x)dx. \tag{18}$$

The inter-level transformations might be defined as follows,

- the projection operator P_{j-1}^{j}, which maps $\mathbf{U}_j$ to $\mathbf{U}_{j-1}$, can be obtained through exact averages computed at the coarser level by

$$u_\gamma = |\Omega_\gamma|^{-1} \sum_{|\mu|=|\gamma|+1, \Omega_\mu \subset \Omega_\gamma} |\Omega_\mu| u_\mu \tag{19}$$

  where $|\gamma| := j$ if $\gamma \in S_j$. As far as grids are nested, this projection operator is *exact* and *unique* [14].

- the prediction operator P_j^{j-1}, which maps $\mathbf{U}_{j-1}$ to an approximation $\hat{\mathbf{U}}_j$ of $\mathbf{U}_j$. Let us notice that there is an infinite number of choices to define P_j^{j-1}, but we impose at least two basic constraints :

  a) The prediction is local, *i.e.*, $\hat{u}_\mu$ depends on the values u_γ on a finite stencil R_μ surrounding Ω_μ, where $|\mu| = |\gamma| + 1$.
  b) The prediction is consistent with the projection in the sense that

$$|\Omega_\gamma| u_\gamma = \sum_{|\mu|=|\gamma|+1, \Omega_\mu \subset \Omega_\gamma} |\Omega_\mu| \hat{u}_\mu \tag{20}$$

  i.e., $P_{j-1}^{j} \circ P_j^{j-1} = Id$.

In the case where P_j^{j-1} is linear, we have

$$\hat{u}_\mu := \sum_\gamma c_{\mu,\gamma} u_\gamma, \tag{21}$$

and if the prediction has some prescribed order $r > 0$ of accuracy, then it is exact for polynomials of degree $r-1$, i.e., if $u \in \prod_{r-1}$, then $u_\gamma = \hat{u}_\gamma$ for all γ [16].

With these operators defined, we define for each cell Ω_μ the prediction error or *detail* associated as the difference between the exact and predicted values,

$$d_\mu := u_\mu - \hat{u}_\mu. \tag{22}$$

Moreover, from the consistency assumption (20), the projection operator (19) and the definition of the *detail* (22) yield :

$$\sum_{|\mu|=|\gamma|+1, \Omega_\mu \subset \Omega_\gamma} |\Omega_\mu| d_\mu = 0. \tag{23}$$

Thus, we define the *detail vector* as $\mathbf{D}_j = (d_\mu)_{\mu \in \nabla_j}$, where the set $\nabla_j \subset S_j$ is obtained by removing for each $\gamma \in S_{j-1}$ one $\mu \in S_j$ such that $\Omega_\mu \subset \Omega_\gamma$ in order to avoid redundancy from expressions (22) and (20). In this way, there is a one-to-one correspondence

$$\mathbf{U}_j \longleftrightarrow (\mathbf{U}_{j-1}, \mathbf{D}_j), \tag{24}$$

which can be implemented using the operators P_{j-1}^j and P_j^{j-1}.

By iteration of this decomposition, we obtain a multi-scale representation of $\mathbf{U}_J$ in terms of $\mathbf{M}_J = (\mathbf{U}_0, \mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_J)$. Therefore, using the local structure of the projection and prediction operators, we can implement the multi-scale transformation

$$\mathcal{M} : \mathbf{U}_J \longmapsto \mathbf{M}_J. \tag{25}$$

2.4.2. Compression and graded tree-structured data

One of the main interests of carrying on such a multi-scale decomposition is that this new representation leads us to define a whole set of regularity estimators all over the spatial domain and thus, a data compression might be performed in order to retain only a minimal quantity of nodes where it is strictly necessary.

In fact, given a set of index $\Lambda \subset \nabla^J$ where $\nabla^J := \bigcup_{j=0}^J \nabla_j$, we define a truncation operator $\mathcal{T}_\Lambda$, that leaves unchanged the component d_λ if $\lambda \in \Lambda$ and replaces it by 0, otherwise. In practice, we are typically interested in sets Λ obtained by thresholding : given a set of level-dependent threshold $(\varepsilon_0, \varepsilon_1, \cdots, \varepsilon_J)$, we set

$$\Lambda = \Lambda(\varepsilon_0, \varepsilon_1, \cdots, \varepsilon_J) := \left\{ \lambda \text{ s.t. } |d_\lambda| \geq \varepsilon_j \,;\, j = |\lambda| \right\}. \tag{26}$$

Applying $\mathcal{T}_\Lambda$ on the multi-scale decomposition of $\mathbf{U}_J$ amounts to building an approximation $\mathcal{A}_\Lambda \mathbf{U}_J$, where the operator $\mathcal{A}_\Lambda$ is given by

$$\mathcal{A}_\Lambda := \mathcal{M}^{-1} \mathcal{T}_\Lambda \mathcal{M}. \tag{27}$$

Moreover, fixing the level-dependent threshold values to

$$\varepsilon_j = 2^{\frac{d}{2}(j-J)} \varepsilon, \quad j = |\lambda|, \quad j \in [0, J], \tag{28}$$

where ε is the threshold value for the finest level J, defines $\mathcal{A}_{\Lambda_\varepsilon} := \mathcal{A}_\Lambda$ with truncation operator $\mathcal{T}_{\Lambda_\varepsilon}$ in (27), so that the set Λ_ε is given by (26) with all the level threshold values related to ε through (28).

Then, a very important result is that operator $\mathcal{A}_{\Lambda_\varepsilon}$ guarantees a thresholding error of prescribed order ε, *i.e.*

$$\|\mathbf{U}_J - \mathcal{A}_{\Lambda_\varepsilon}\mathbf{U}_J\|_{L^2} \propto \varepsilon, \tag{29}$$

(see [45, 16] for further details), even though we have literally discarded some *details* by means of the thresholding process, and thus, considered $u_\mu = \hat{u}_\mu$ following (22).

However, we can not deliberately delete allegedly useless *details* because a certain data structure must be respected in order to perform the different computations associated to the multi-scale transformation itself. In fact, during these transformations, the availability of cell values within the local prediction stencil must be always guaranteed. And thus, the set Λ_ε must exhibit a *graded tree* structure (see [16, 57] for more details on the definition and construction of such structures). In the practice, this means that among the indices $\lambda \in \nabla_J$, we shall retain only those for which $|d_\lambda| \geq \varepsilon_{|\lambda|}$ and literally eliminate the rest of them verifying always the necessary conditions prescribed by a *graded tree* structured data. Hence, an effective data compression is accomplished because $\mathbf{U}$ is not represented on the finest grid S_J as $\mathbf{U}_J$ anymore, but on a *graded tree* Λ_ε.

In this paper, we will not conduct the analysis of such data structures which can be easily found in the given references. However, we present the following terminology associated to a tree representation that we will adopt throughout this paper:

- If $\Omega_\mu \subset \Omega_\lambda$ with $|\lambda| = |\mu| - 1$, we say that Ω_μ is a *child* of Ω_γ and that Ω_γ is the *parent* of Ω_μ.
- Moreover, we define the *leaves* $\mathrm{L}(\Lambda)$ of a *graded tree* Λ as the set of Ω_λ with $\lambda \in \mathrm{L}(\Lambda)$ such that Ω_λ has no children in Λ.
- Finally, we define Ω_λ as a *root* when it belongs to the coarsest grid, that is $\lambda \in S_0$ or $|\lambda| = 0$, in which case, we denote λ as λ_0.

2.4.3. Fully adaptive multiresolution scheme

With the previous definitions, we now present the fully adaptive multiresolution scheme as presented in [16]. The key point of such method is that the numerical solution $\mathbf{U}^n$ at time $n\Delta t$ can be represented on an adapted grid by the set $(u^n_\lambda)_{\lambda\in\mathrm{L}(\Lambda^n_\varepsilon)}$. And thus, the time evolution is performed only on the *leaves*. Hence, given the current *graded tree* Λ^n_ε, we obtain $\Lambda^{n+1}_\varepsilon$ and $\mathbf{U}^{n+1}$ through the following steps :

- *Refinement.* A new set $\tilde{\Lambda}^{n+1}_\varepsilon$ containing Λ^n_ε is constructed in order to predict the evolution of the solution. The u^n_λ with $\lambda \in \tilde{\Lambda}^{n+1}_\varepsilon \backslash \Lambda^n_\varepsilon$ can be constructed applying $\mathcal{M}^{-1}$.
- *Computation.* A first numerical solution $\tilde{\mathbf{U}}^{n+1}$ at time $(n+1)\Delta t$ is computed on $\mathrm{L}(\tilde{\Lambda}^{n+1}_\varepsilon)$ by $\tilde{\mathbf{U}}^{n+1} = \mathcal{S}^{\Delta t}(\mathbf{U}^n)$ or $\tilde{\mathbf{U}}^{n+1} = \mathcal{L}^{\Delta t}(\mathbf{U}^n)$, using one of the Lie or Strang formulae (5) or (6).
- *Thresholding.* We obtain $\mathbf{U}^{n+1}$ by thresholding $\tilde{\mathbf{U}}^{n+1}$ according to $\mathbf{U}^{n+1} = \mathcal{A}_{\Lambda^{n+1}_\varepsilon}\tilde{\mathbf{U}}^{n+1}$ where $\Lambda^{n+1}_\varepsilon \subset \tilde{\Lambda}^{n+1}_\varepsilon$.

Notice that $\tilde{\Lambda}^{n+1}_\varepsilon$ should be adapted for describing the solution at both $n\Delta t$ and $(n+1)\Delta t$. Therefore, the definition of the refinement process accomplished by the refinement operator $\mathcal{R}$, is a

crucial aspect for the success of the method. Several heuristics have been developed and analyzed in order to accomplish this goal, see for example [45, 16]. In particular, there are the classical Harten's heuristics in the univariate dyadic case.

We now denote by $\mathbf{V}_J^n := (v_\lambda^n)_{\lambda \in S_J}$, the solution obtained by a classical finite volume scheme performed on the finest grid. Moreover, denoting by $\mathbf{U}_J^n$, the solution produced by the fully adaptive multiresolution scheme reconstructed on the finest mesh S_J, keeping in mind that the adaptive scheme really operates with a compressed representation of $\mathbf{U}^n$ on the $\mathrm{L}(\Lambda_\varepsilon^{n+1})$. Then, for a fixed time $T = n\Delta t$, it can be shown that

$$\|\mathbf{U}_J^n - \mathbf{V}_J^n\|_{L^2} \propto \frac{T}{\Delta t}\varepsilon. \tag{30}$$

Hence, the error is directly related to the prefixed threshold error chosen ε for the finest grid J and thus, the level of accuracy expected from such adaptive strategy can be investigated with this theoretical background. Even if this result comes from a classical error analysis developed by Harten (see [45]), the validity of this estimate has been extended to a fully multiresolution scheme in [16] and turns out to be one of the major advantages of such methods.

2.5. Summary of the Numerical Strategy

Once we have defined all the elements of the numerical strategy, the implementation of the fully adaptive multiresolution technique can be summarized as follows :

$$(u_\lambda^n)_{\lambda \in \mathrm{L}(\tilde{\Lambda}_\varepsilon^n)} \quad \xrightarrow{\mathcal{M}} \quad (u_{\lambda_0}^n, d_\lambda^n)_{\lambda \in \tilde{\Lambda}_\varepsilon^n} \tag{31}$$

$$(u_{\lambda_0}^n, d_\lambda^n)_{\lambda \in \tilde{\Lambda}_\varepsilon^n} \quad \xrightarrow{\mathcal{T}_{\Lambda_\varepsilon^n}} \quad (u_{\lambda_0}^n, d_\lambda^n)_{\lambda \in \Lambda_\varepsilon^n} \tag{32}$$

$$(u_{\lambda_0}^n, d_\lambda^n)_{\lambda \in \Lambda_\varepsilon^n} \quad \xrightarrow{\mathcal{R}} \quad (u_{\lambda_0}^n, d_\lambda^n)_{\lambda \in \tilde{\Lambda}_\varepsilon^{n+1}} \tag{33}$$

$$(u_{\lambda_0}^n, d_\lambda^n)_{\lambda \in \tilde{\Lambda}_\varepsilon^{n+1}} \quad \xrightarrow{\mathcal{M}^{-1}} \quad (u_\lambda^n)_{\lambda \in \mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})} \tag{34}$$

$$(u_\lambda^n)_{\lambda \in \mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})} \quad \xrightarrow{\mathcal{S}^{\Delta t}} \quad (u_\lambda^{n+1})_{\lambda \in \mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})} \tag{35}$$

The set $(u_{\lambda_0}^n)$ is defined as the set of roots of $\tilde{\Lambda}_\varepsilon^n$, that is all $\lambda \in \tilde{\Lambda}_\varepsilon^n$ such that $|\lambda| = 0$ or $\lambda \in S_0$. For $n = 0$, the initial condition should be represented on $\mathrm{L}(\tilde{\Lambda}_\varepsilon^0)$ in step (31), which in a general case can be the finest grid, that is all $\lambda \in \mathrm{L}(\tilde{\Lambda}_\varepsilon^0)$ is such that $|\lambda| = J$ or $\lambda \in S_J$, and as a consequence all the nested grids $S := \bigcup_{j=0}^J S_j$ are present, $\tilde{\Lambda}_\varepsilon^0 = S$.

Nevertheless, this is not always possible due to excessive high memory requirements, in which case, the initial condition might be computed on an intermediate level of grid j, that is all $\lambda \in \mathrm{L}(\tilde{\Lambda}_\varepsilon^0)$ is such that $|\lambda| = j$ or $\lambda \in S_j$. Then, we must first threshold in order to project/predict the values over the set of grids S.

The thresholded tree Λ_ε^n into (32) is obtained applying over $\tilde{\Lambda}_\varepsilon^n$ the operator $\mathcal{T}_{\Lambda_\varepsilon^n}$ with ε_j given by (28). Then, the refinement operator $\mathcal{R}$ is applied over the thresholded tree to locally refine in order to foresee the time evolution of the solution. Finally, as an illustration, the time integration operator is the Strang operator splitting $\mathcal{S}^{\Delta t}$ given by one of the formulae (6) and it is computed only on $\mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})$. In the same way, the algorithm can consider the Lie operator splitting (5) as the time integration method.

Thus, the algorithm can schematically be summarized by

$$\mathbf{U}^{n+1} = \mathcal{S}^{\Delta t}(\mathcal{M}^{-1}\mathcal{R}\mathcal{T}_{\Lambda_\varepsilon^n}\mathcal{M}\mathbf{U}^n), \tag{36}$$

with the compressed representations of $\mathbf{U}^{n+1}$ and $\mathbf{U}^n$ by $(u_\lambda^{n+1})_{\lambda\in \mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})}$ and $(u_\lambda^n)_{\lambda\in \mathrm{L}(\tilde{\Lambda}_\varepsilon^n)}$ respectively. Hence, the three basic steps established before: *Refinement*, *Computation* and *Thresholding* are performed through steps (33), (35) and (32) by means of multi-scale transformations (31) and (34).

3. Articulation of the building blocks and algorithm implementation

In the first part of this section, we first present the building process of our numerical strategy in the context of multi-scale reaction waves. We highlight the interest of a splitting technique and the different advantages associated to the time integration methods. Once established the time integration strategy, the use of adaptive multiresolution techniques are discussed along with the global and final considerations in order to achieve a self-sufficient and optimized algorithm. In the second part, some aspects of the practical implementation of the algorithm are described.

3.1. Construction of the numerical strategy

The classical orders achieved with a Lie or Strang scheme are no longer valid since we consider very stiff reactive terms (see [27]). Order reductions may then appear due to time multi-scale phenomena, unless we consider very small integration time steps. In this context, it has been proved in [27] that better performances are expected while ending the splitting scheme by the time integration of the reaction part (4) or in a more general case, the part involving the fastest time scales of the phenomenon (see the numerical application in [25]). In particular, in the case of linear diagonal diffusion problems, no order loss is expected for the $\mathcal{L}_1^{\Delta t}$ and $\mathcal{S}_2^{\Delta t}$ schemes when fast scales are present in the reactive term. Even more, as it was presented and analyzed in [24], the presence of high spatial gradients may also degrade the performance of these methods leading to order reductions coming from space multi-scale phenomena.

In this context, a multi-scale reaction wave implies both space and time multi-scale phenomena. The theoretical studies conducted in [27, 24] then show how splitting methods should be implemented in order to either maintain the classical orders or estimate correctly the associated order loss in the case when the splitting time step largely exceeds the fastest time scale of the problem. Actually, we are placed in a context in which the fastest time scales have always an important impact on the numerical performance of the time integration methods, but do not have a leading role in the physics of the phenomenon. Thus, larger splitting time steps might be considered as long as a correct representation of the physics of the phenomenon is guaranteed; in our case, the propagation of a multi-scale reaction wave. This splitting strategy based on these theoretical studies has already been validated and confronted to other methods in [26] for these applications.

In order to remain coherent with the numerical analysis, we have to take into account a natural assumption in all splitting order estimates which is that the solutions associated to each subsystem (3) and (4) are known exactly or can be known with a sufficient accuracy (see for example [42, 46, 28, 27]). This means that suitable time integration methods in terms of order and stability must be chosen in order to guarantee the accuracy of the estimates established by the corresponding numerical analysis.

As mentioned before, Radau5 is not only an A-stable method, but also L-stable, so that very stiff systems of ODEs might be solved without any stability problem. Moreover, it relies on local adapting time integration which guarantees the requested accuracy and at the same time, allows to discriminate stiff zones from regular ones, *i.e.* smaller time steps correspond to stiffer behaviors. Finally, being a high order method (formally of order 5, which at worst might be reduced to 3), the classical operator splitting orders are guaranteed and thus, all error coming from the time integration will be bounded by the one due to the splitting procedure itself.

Nevertheless, this high order method is achieved thanks to an implicit Runge-Kutta scheme, this means that in a general case, nonlinear systems must be solved throughout the time integration process. Even if the solving system tools are highly optimized (which are based on modified Newton's methods), these procedures become very expensive for large systems and important memory requirements are needed in order to carry out these computations. That is why the size of the system is terribly limited by the memory resources. However, in a splitting scheme context, we easily overcome this difficulty since the reactive term of (2) is a system of ODEs without spatial coupling. Hence, a local approach node by node is adopted where the memory requirements will be only determined by the number of local unknowns, which normally does not exceed conventional memory resources. Even more, this approach also allows a straightforward parallelization of computations where no data exchange is needed among nodes (see for example the numerical implementations achieved in [32]).

Another very important feature of this strategy is that precious computation time is saved because we adapt the time integration step only at the nodes where the reaction phenomenon takes place. In the context of multi-scale reaction waves, this happens in a very low percentage of the spatial domain, normally only in the neighborhood of the wavefront. Then, there is practically no time step adaptation at nodes with a chemistry at (partial) equilibrium. This would not be possible if we integrate the entire reaction-diffusion system (2) at once.

If we now consider Rock4, we recall that one of the most important advantages of such method is its explicit character, hence the simplicity of its implementation. In fact, no sophisticated Linear Algebra tools are needed (no resolution of linear systems required) and thus, the resolution is based on simple matrix-vector products. Nevertheless, the cost of computations relies directly on the requested quantity of such products, that is the number of internal stages s needed over one time integration step Δt. Notice also that the memory requirements are widely reduced as a consequence of this explicit character, nevertheless we must keep in mind that these memory requirements will proportionally increase with the number of nodes considered over the spatial domain.

In order to guarantee the stability of the method for a fixed time step Δt, the number of stages s needed is directly related to the spectral radius $\rho(\partial \mathbf{f}/\partial \mathbf{u})$ (considering a general problem such as (7)), since it should verify

$$\tilde{\ell}_s \simeq 0.35 \cdot s^2 \geq \Delta t \, \rho\left(\frac{\partial \mathbf{f}}{\partial \mathbf{u}}(\mathbf{u})\right).$$

Therefore, the spectral radius must be previously estimated (for example, using the Gershgoring theorem or even numerically as proposed by the Rock4 solver by the means of a nonlinear power method). The method then is very appropriate for diffusion problems because of the usual predominance of negative real eigenvalues for which the method is efficiently stable. A very suitable example is the linear diagonal diffusion problem (3) with only negative real eigenvalues and constant spectral radius. In our particular applications, the diffusive phenomenon has a

leading role of propagator of perturbations over the (partial) equilibrium nodes that result on excitation of the reactive schemes and thus, the propagation of the reaction wave. The resulting self-similar character implies that the number of stages needed will remain practically constant throughout the evolution of the phenomenon.

These facts make of Rock4 a very suitable method for our simulation purposes and highlight the particular advantages of the operator splitting. In the same way that we have seen for the reaction solver, the implementation of this diffusion solver over the entire reaction-diffusion system (2) will not be appropriate under neither theoretical nor practical considerations. Moreover, taking into account that also Rock4 is a high order method (order 4), the classical operator splitting orders are guaranteed and thus, all error coming either from the diffusion or reaction time integration process will be bounded by the one due to the splitting procedure itself.

Further considerations lead to other potential improvements in order to yet optimize our numerical strategy. In fact, as stated before, we are concerned with the propagation of reacting wavefronts, hence important reactive activity as well as steep spatial gradients are localized phenomena. This implies that if we consider the resolution of the reactive problem (4), a considerable amount of the time computation is spent on nodes that are practically at (partial) equilibrium (see for example a precise computational time evaluation in [32]). Moreover, there is no need to represent these quasi-stationary regions with the same spatial discretization needed to describe the reacting wavefront, *i.e.* the multi-scale phenomenon itself, so that the diffusion problem (3) might also be solved over a smaller number of nodes. An adapted mesh obtained by a multiresolution process then turn out to be a very convenient solution to overcome these difficulties.

These methods have the supplementary advantage that the numerical analysis on which they are based introduces error estimates that can be analyzed and confronted to those established by the splitting schemes. Hence, a more precise control of error can be drawn out of this optimal combination of methods. Moreover, while considering the multiresolution implementation coupled with the constructed time integration solver, we have to recall that no global or coupled stability constraints are imposed on the integration time steps since the different numerical methods were chosen in order to guarantee independently very stable features.

However, a natural constraint that might arise when considering an evolving adapted mesh is that the dynamical configuration requiring refinement should not move off the fine grid region before the next remeshing event takes place, as we have seen that the time evolution is performed on a fixed *graded tree*. Nevertheless, this is not a problem as long as we guarantee that the wavefront does not propagate and cross completely the adjacent cells on the finest grid during a splitting time step. This can be easily achieved considering the wavefront speed (previously estimated either theoretically or numerically), and verifying that this speed is solved with enough accuracy by the chosen splitting time step. As a complement, the chosen refinement criteria imply a fine grid region over a sufficiently large extent to predict the wavefront translation taking into account the propagating nature of the phenomenon. In fact, this particularity makes of the adaptive multiresolution technique a very suitable method for the simulation of very localized reaction waves.

3.2. Description of the global algorithm

Let us first summarize the algorithm as follows : (i) We consider an initial *graded tree* created by multi-scale transformation from an initial condition. Then, (ii) a time evolution based on the operator splitting scheme is computed on the *leaves*. (iii) A new multi-scale transformation is performed after each splitting time step in order to compute the *details* and adapt the tree, always

respecting a *graded tree* structure. Step (iii) is repeated until the end of the time domain. In what follows, the key aspects of the algorithm are detailed.

A dynamic *graded tree* structure is used in this implementation to represent data in the computer memory. The adapted grid corresponds to a set of nested dyadic grids generated by refining recursively a given cell depending on the local regularity of the solution. The chosen data structure can handle 1D, 2D and 3D Cartesian geometries. In the practice, the tree is represented by a arranged set of cells (defined as *nodes*) so that they can be easily retrieved by means of a searching process from the basis of the tree or from a coarser grid in a general case. Then, the basic element of the structure is the cell itself, which consists of a set of geometric and physical values, plus pointers to its *parent*, their *children* and the contiguous cells in each dimension, the *neighbors*. Figure 1 shows an example of a *graded tree* structure in 1D.

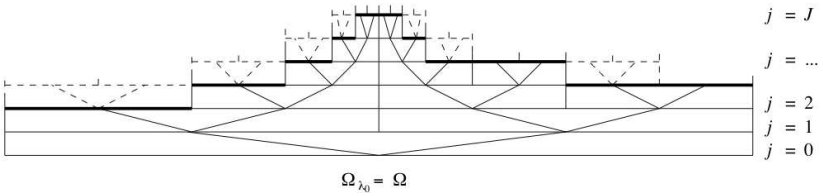

Figure 1: Example of a *graded tree* structure in 1D. *Nodes* and links to their corresponding *children* are indicated (solid lines) as well as the *leaves* (solid bold lines) and the *phantoms* (dashed lines).

We have previously defined the *roots* as the first cells of the tree structure which correspond to the basis of the tree, Ω_{λ_0} in Figure 1. The *nodes* are elements of the tree and the *leaves*, the upper elements with no *children* in the tree (see Figure 1). In d dimensions, a *parent*-cell at a level j has at most 2^d *children* cells at the level $j+1$. When there is only one *root* in the tree structure, the maximal number of *leaves* N on which the solution might be represented is given by $N = 2^{dJ}$, which is exactly the number of cells on the finest grid. The maximal number of cells M in the tree is given by $M = (2^{d(J+1)} - 1)/(2^d - 1)$.

To ensure conservativity, numerical fluxes on a cell face must have the same value for any cell or set of cell contiguous to the face. Concerning a *leaf* of a locally refined grid, the adjacent cell might however not exist at the same grid level. In such a situation, the numerical fluxes should always be computed at the highest grid level by using virtual cells called *phantoms* that must be added. Figure 2 depicts this situation in a 2D configuration where the ingoing and outgoing flux are not balanced for two different grid levels. In particular, we follow the procedure introduced in [57] and the flux are computed at the highest grid level by means of the *phantoms* as shown in Figure 2. Considering the explicit algorithm for the diffusion term, let us note that the solution is not integrated on these virtual cells but just evaluated by using the prediction operator from the previous solution at time $n\Delta t$. Thus, in the case of a multi-stage method such as Rock4, these evaluations must be performed at each stage which implies an updating of the whole tree through projection/prediction operations. Hence, the interest of limiting the number of staged performed by Rock4 is enhanced.

Following (21), in order to predict the approximated values $\hat{u}_\mu$ by P_j^{j-1}, we use a centered linear polynomial interpolation of order $2l$, *i.e.* accuracy order $r = 2l + 1$, where we consider the l nearest *neighbors* in each direction. Raising the accuracy order imposes larger stencils but more accurate interpolations. In a 1D configuration for the case $l = 1$, a third order accurate

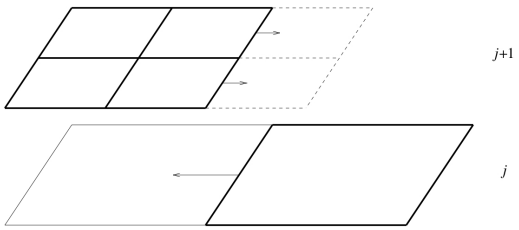

Figure 2: Example of flux computations in 2D. *Nodes* (solid lines), *leaves* (solid bold lines) and *phantoms* (dashed lines) are indicated along with the corresponding level flux.

multiresolution, the prediction is explicitly given by

$$\hat{u}_{j+1,2k} = u_{j,k} + \frac{1}{8}(u_{j,k-1} - u_{j,k+1}) \quad \text{and} \quad \hat{u}_{j+1,2k+1} = u_{j,k} + \frac{1}{8}(u_{j,k+1} - u_{j,k-1}), \tag{37}$$

where the first index denotes the grid level and the second, the indexation of the cell into the tree. Notice that depending on the accuracy order ($2l$) of the centered polynomial interpolation, *details* recover null values for smooth solutions with locally bounded $2l$-th order derivatives [15]. As Cartesian mesh is used, extension to multidimensional polynomial interpolations is easily obtained by a tensorial product of the 1D operator [9, 57]. In the numerical illustrations presented in this paper we will restrict the different analysis to the case $l = 1$.

Considering the propagating character of reaction waves, we adopt a refinement strategy based on the increase of the *details* in the neighborhood of the wavefront. Since we are studying very localized configurations, the *details* are naturally arranged alike. Thus, considering a *leaf* with index k at a grid level j, if the following criterion:

$$\frac{\left\|d_{j,k}\right\|_{L^2}}{\max_k \left|d_{j,k}\right|} \geq 2^{(2l-1)}\varepsilon_j, \tag{38}$$

is satisfied, *children* at a grid level $j + 1$ are added to the tree, where we consider a measure of *details* scaled by the global maximum. This criterion is based on the classical Harten's heuristics, slightly modified, and prove to be very efficient taking advantage of the self similarity character of the phenomena under consideration.

Finally, the choice of the splitting time step $\Delta t_{splitting}$ is mainly based on two considerations. On the one hand, the idea is to achieve a certain level of accuracy in the physical resolution of the phenomenon. And thus, we take into account the error estimates provided by the theoretical studies previously conducted on splitting errors. On the other hand, a heuristic based criterion is developed in order to guarantee the coherence between the propagating phenomenon solved by the splitting technique and the dynamical refined mesh. The global process can be then summarized by the following heuristic :

a) Computation of the wavefront speed v either numerically using a coupled solver (if possible) or based on theoretical/analytical estimates.
b) Computation of the wavefront speed v_{split} using the splitting solver with initial splitting time step Δt.
c) We then consider $\Delta t'_{splitting} = (\eta/E_v)^{1/2}\Delta t$ with $E_v = |v - v_{split}|/v$ and previously chosen η.

d) We also consider $\Delta t''_{splitting} = n\Delta x^J/(1 \pm \eta)v$ where $\pm\,\eta$ depends on $v \lessgtr v_{split}$ and with previously chosen $n \geq 1$.

e) Finally, $\Delta t_{splitting} = \min\left(\Delta t'_{splitting}, \Delta t''_{splitting}\right)$.

The estimate given in c) considers the approximation error of the wavefront speed due to the splitting technique. As mentioned before, the splitting time step is meant to solve as accurate as possible physical features of the phenomenon. Then, η is a fixed value that limits this loss of accuracy of the wavefront speed. In particular, we have limited the maximum relative error to $\eta = 0.05 \geq E_v$.

Furthermore, the estimate given in d) verifies a classical CFL-like condition which relates spatial discretization to the splitting time step. Notice that Δx^J corresponds to the finest grid at level J and n takes into account the refined region around the front, which normally implies more than one cell. In particular, we have found a suitable value of $n = 2.5$ after some numerical tests for the problems we have studied.

In the case of self-similar progression of wavefronts, the problem of the time step selection is simplified by the fact that usually it does not need to be computed more than one time. However, in an extended general case, the heuristic established before is a continuous process during the execution of the algorithm. In fact, this heuristic mimics a self-adapted time step procedure based on estimated splitting errors in which case the computed splitting time step should in addition verify the CFL-like condition. However, further studies and error characterizations are required in order to establish an extended self-adaptation process of the splitting time step, valid for more general configurations; in particular, this is a current topic of research.

The global algorithm can be summarized by the following procedure :

1. **INITIALIZATION**
   - *Initialization of parameters : e.g.* maximum and minimum grid level, domain size, number of *roots*
   - *Initialization of the mesh structure :*
     - creation of the different grids
     - initialization of parameters of each cell from the *roots*, *e.g.* position, coordinates, level threshold value ε_j
     - assignation of *children* and *neighbors* from the *roots*
   - *Computation of initial solution*
   - *Computation of the splitting time step*
2. **LOOP IN TIME**
   - *Computation of cell averages :* projection operator P^j_{j-1} from *leaves* towards *roots*
   - *Computation of details :* operator $\mathcal{M}$ from *roots* towards *leaves*
   - *Thresholding of the tree :* operator $\mathcal{T}_{\Lambda_\varepsilon}$ from *roots* towards *leaves*
   - *Refinement of the tree :* creation of the *graded tree* by operator $\mathcal{R}$
   - *Computation of cell averages from details :* operator $\mathcal{M}^{-1}$ from *roots* towards *leaves*
   - *Creation of phantom cells :* needed to compute the diffusion time step
   - *Time integration :* operator splitting $\mathcal{S}^{\Delta t}$ or $\mathcal{L}^{\Delta t}$ applied only on *leaves*
     - reaction time step, time integration performed by Radau5 cell by cell

– diffusion time step, time integration performed by Rock4. At each time internal stage we must update virtual cells through projection/prediction operations

3. **OUTPUT**
   Save the adapted grid with the corresponding cell values represented on it

4. Numerical Simulations

In this last section, we present some numerical illustrations of the proposed numerical strategy. An academical problem coming from the nonlinear chemical dynamics is then described and treated. The different features of the method as well as the numerical results are then presented and discussed for the one- and two-dimensional cases : for instance error estimates, data compression values and time consumption. Finally, the potential of the method is highlighted in the context of two- and three-dimensional simulations of important size.

4.1. Mathematical model of study

We are concerned with the numerical approximation of some models coming from nonlinear chemical dynamics : the Belousov-Zhabotinski reaction, a catalyzed oxidation of an organic species by acid bromated ion (see [35] for more details and illustrations). We can first consider the two-variable Oregonator model, studied in [47]; it has solutions that represent propagation of a steep wavefront by interplay of $HBrO_2$ (hypobromous acid) diffusion with an autocatalytic reaction that quickly generates $HBrO_2$ (using bromide ions Br^- as an intermediary species that remains always in equilibrium with local instantaneous $HBrO_2$). Denoting by $b = [HBrO_2]$ and $c = [Br^-]$, we might consider the following model :

$$\left.\begin{array}{rcl} \dfrac{\partial b}{\partial \tau} - D_b \Delta b & = & \dfrac{1}{\epsilon}\left(b(1-b) + \dfrac{f(q-b)c}{q+b}\right), \\ \dfrac{\partial c}{\partial \tau} - D_c \Delta c & = & b - c, \end{array}\right\} \tag{39}$$

with diffusion coefficients D_b and D_c and some real positive parameters f, small q and small ϵ.

Nevertheless, a more refined model, introduced in [39] and coming from the classic work of Field, Koros and Noyes (FKN), takes into account not only the two species $HBrO_2$ and Br^- but also the cerium(IV). Denoting by $a = [Ce(IV)]$, we obtain a very stiff system of three partial differential equations :

$$\left.\begin{array}{rcl} \dfrac{\partial a}{\partial \tau} - D_a \Delta a & = & \dfrac{1}{\mu}(-qa - ab + fc), \\ \dfrac{\partial b}{\partial \tau} - D_b \Delta b & = & \dfrac{1}{\epsilon}\left(qa - ab + b(1-b)\right), \\ \dfrac{\partial c}{\partial \tau} - D_c \Delta c & = & b - c, \end{array}\right\} \tag{40}$$

with additional diffusion coefficient D_a and real positive parameter $\mu \ll \epsilon$.

An interesting study on these reactions scheme can be found in [39]. In fact, the dynamical systems associated to systems (39) and (40) model reactive excitable media with an important time scale spectrum. Moreover, the addition of diffusive phenomenon generates propagating wavefronts, and thus, we have a reactive medium spatially propagated under induction of the

diffusive effect. Hence, multi-scale reaction waves are developed. The spatial multi-scale aspect is associated to the high spatial gradients propagating in time.

This model presents all the difficulties associated to a multi-scale and moreover stiff configuration, thus the main goal of the following simulations is to show the interest of applying such numerical strategies on this kind of problem. A preliminary description of the advantages of applying a splitting strategy to these models has already been presented in [26]. In what follows we consider all 1D, 2D and 3D configurations of problem (40). The same conclusions might be then extended to problem (39) since it comes from a simplification of model (40).

4.2. 1D BZ equation

Let us first consider the 1D case of problem (40) with homogeneous Neumann boundary conditions and the following parameters based on the theoretical studies conducted in [39] : $\epsilon = 10^{-2}$, $\mu = 10^{-5}$, $f = 3$ and $q = 2.10^{-4}$, with diffusion coefficients $D_a = 1$, $D_b = 1$ and $D_c = 0.6$. We study the phenomenon over a time domain of $[0, 2]$ and for the space region of $[0, 80]$.

On the one hand, in order to conduct an analysis of the numerical results, we define as a reference solution a *quasi-exact* resolution of the coupled problem (40). This *quasi-exact* solution is obtained after the time integration of the coupled system performed by Radau5 using very fine tolerances. For this one-dimensional configuration and with the considered discretization of 4096 points in space, the memory requirements is acceptable and the simulation is totally feasible.

And on the other hand, the chosen time integration method for the proposed numerical strategy is the Strang splitting technique $\mathcal{S}_2^{\Delta t}$ in (6), based on the previous considerations and the theoretical background conducted in [27]. Then, Radau5 integrates the reaction problem (4) for (40) during the first and last half-steps $\Delta t/2$, and Rock4 solves the diffusion problem (3) during the intermediary step Δt. For these simulations, we choose a splitting time step $\Delta t \approx 3.9 \times 10^{-3}$, based on the heuristic previously presented.

With this operator splitting strategy, the space adaptive multiresolution technique is performed on 12 nested dyadic grids with $N = 2^{1\times 12} = 4096$ cells on the finest grid $j = J = 12$. In this application as well as in the following, we always compute the local *details* under a discretized L^2-norm of vector $(a, b, c)^t$ in order to perform the thresholding and refining processes. The resulting traveling waves can be seen in Figure 3 for three different times considering $\varepsilon = 1.10^{-4}$ as the thresholding value into (28). Figure 4 shows the evolution of the corresponding adapted grid where the different level grids are indicated. We notice that the adapted grid is tightened around the stiff regions and clearly propagates along the traveling wave.

Let us now define the *data compression* as one minus the ratio between the number of real *leaves* and the number of cells on the finest uniform grid, expressing the whole as a percentage. This definition comes from the fact that the solutions are represented and computed only on the *leaves* throughout the time domain. Table 2 shows the number of *leaves* considered and the different *data compression* rates for different thresholding values. Smaller thresholding values of ε imply more refinement in order to guarantee (29) and thus, the *data compression* is less important.

Considering now the L^2-error of these results, u_{MR}, with respect to the reference *quasi-exact* solution, u^J_{qe}, we decompose it into two parts, the error coming from the splitting process and that of the multiresolution decomposition :

$$\|u^J_{qe}(t) - u_{MR}(t)\|_{L^2} \leq \|u^J_{qe}(t) - u^J_{split}(t)\|_{L^2} + \|u^J_{split}(t) - u_{MR}(t)\|_{L^2}. \tag{41}$$

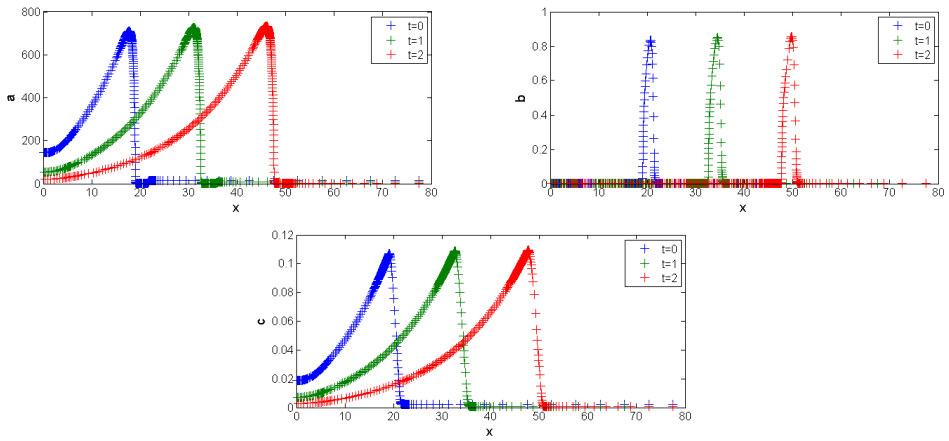

Figure 3: 1D BZ traveling waves. MR solutions for times $t = 0$, $t = 1$ and $t = 2$. Variable a (top left), b (top right) and c (bottom), $\varepsilon = 1.10^{-4}$. Finest grid : 4096.

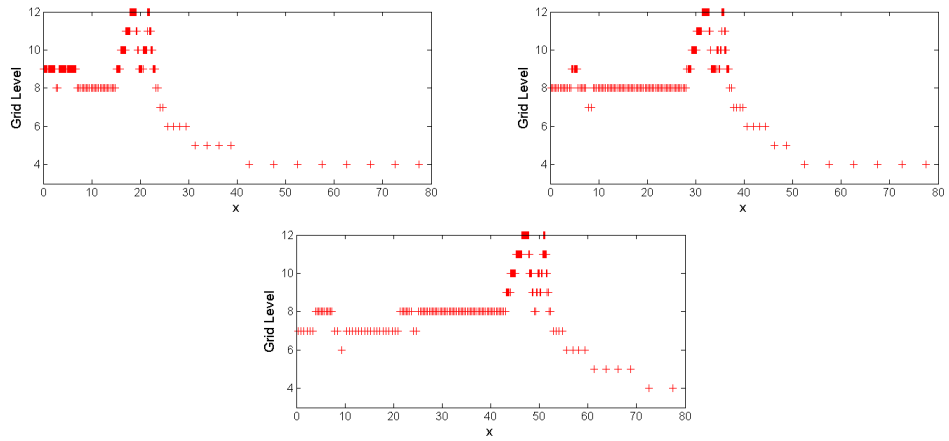

Figure 4: 1D BZ. Grid levels at $t = 0$ (top left), $t = 1$ (top right) and $t = 2$ (bottom), $\varepsilon = 1.10^{-4}$. Finest grid : 4096.

Then, we must also solve problem (40) with the proposed splitting scheme on the uniform finest grid $J = 12$ in order to obtain u^J_{split} and perform the comparisons.

Figure 5 shows these errors for variables a, b and c, and for different thresholding values of ε. The splitting error $\|u^J_{qe}(t) - u^J_{split}(t)\|_{L^2}$ is computed on the finest grid and does not depend on the grid adaptation process, and thus, it is fixed by the splitting time step. The error coming from the multiresolution process is given by $\|u^J_{split}(t) - u_{MR}(t)\|_{L^2}$ considering that both solutions use the same time integration strategy. However, in order to compare solutions u^J_{split} and u_{MR} we must consider the same spatial discretization which is easily achieved by projection/prediction operations.

The multiresolution error is estimated by expression (30) as we have presented before and

ε	t = 1		t = 2	
	Leaves number	Comp %	Leaves number	Comp %
$\mathbf{1.10^{-1}}$	84	97.95	94	97.71
$\mathbf{1.10^{-2}}$	137	96.66	142	96.53
$\mathbf{1.10^{-3}}$	215	94.75	242	94.09
$\mathbf{1.10^{-4}}$	342	91.65	410	89.99
$\mathbf{1.10^{-5}}$	518	87.35	632	84.57
$\mathbf{1.10^{-6}}$	789	80.74	881	78.49
$\mathbf{1.10^{-7}}$	1154	71.83	1318	67.82
$\mathbf{1.10^{-8}}$	1771	56.76	2156	47.36
$\mathbf{1.10^{-9}}$	1893	53.78	2660	35.06
$\mathbf{1.10^{-10}}$	1906	53.47	2688	34.37

Table 2: 1D BZ. *Data compression* for different thresholding values ε. Finest grid : 4096.

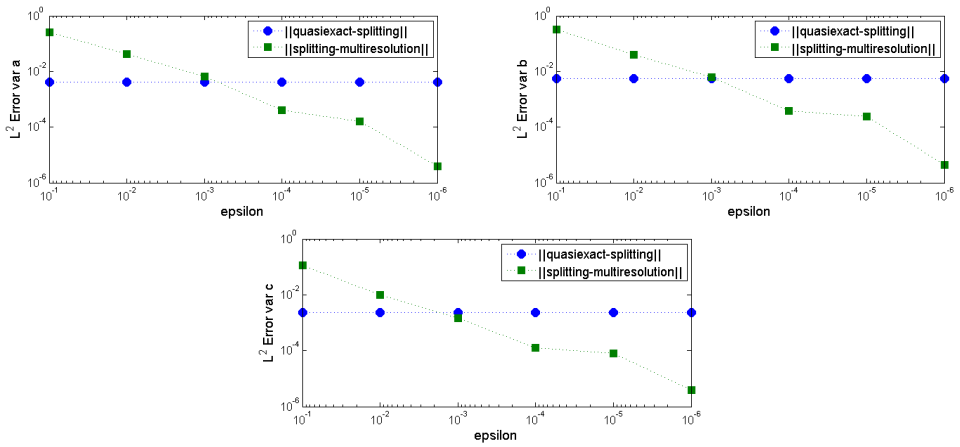

Figure 5: 1D BZ. Splitting and multiresolution L^2 errors at $t = 2$. Variable a (top left), b (top right) and c (bottom). Finest grid : 4096.

as it can also be seen in Figure 5. For $\varepsilon < 1.10^{-3}$ the global error $\|u^J_{qe}(t) - u_{MR}(t)\|_{L^2}$ is given mostly by the process of splitting, hence an important data compression is obtained with the same accuracy of the time integration solver. These estimates are computed at final time $t = 2$ where a worst performance of all the methods is expected. These conclusions are also valid for $t < 2$, even though they are not presented.

We now consider the time consumption of the numerical strategy proposed, and of those calculated on an uniform fixed grid, the *quasi-exact* and the splitting solutions. Table 3 shows these results where we consider as the reference consumption time the one utilized by the *quasi-exact* computation. However, the consumption of the multiresolution/grid adaptation process can be estimated from the time consumption of the splitting strategy.

As expected, a splitting computation is less expensive in computation time than a coupled one. This is a straightforward result of the local resolution of the reactive term in the splitting

Method	Time	Ratio	Comp %
Quasi-exact	62.71	**1**	0
Splitting	44.36	**1.41**	0
MR 1.10^{-1}	7.22	**8.69**	98
MR 1.10^{-2}	10.38	**6.04**	96
MR 1.10^{-3}	15.23	**4.12**	94
MR 1.10^{-4}	21.66	**2.90**	91
MR 1.10^{-5}	31.05	**2.02**	86
MR 1.10^{-6}	51.36	**1.22**	80
MR 1.10^{-7}	68.66	**0.91**	70
MR 1.10^{-8}	104.43	**0.60**	52
MR 1.10^{-9}	113.22	**0.55**	44
MR 1.10^{-10}	113.86	**0.55**	44

Table 3: 1D BZ. Computation time in seconds, ratio computed with respect to the *quasi-exact* computation time and average *data compression*. Finest grid : 4096.

strategy, which allows to save precious time on zones where no important reactive behavior is present. This local approach also implies very important savings concerning memory requirements for the implicit solver. Considering the splitting/multiresolution strategy, it turns out to be very efficient when the *data compression* is high enough to compensate extra computations related to the method. As an illustration, for $\varepsilon = 1.10^{-6}$ practically no gain in the time consumption is obtained while the *data compression* is still very high, $\sim 80\%$. However, for a suitable value of $\varepsilon = 1.10^{-3}$, a *data compression* of $\sim 94\%$ is achieved with a very acceptable level of accuracy as seen in Figure (5) and this configuration is ~ 3 times faster than the split solver without grid adaptation.

An analysis of the distribution of the total computation time over the different processes in the splitting/multiresolution strategy reveals that : in general, $\sim 70\%$ of the time is used by the diffusion integration against $\sim 25\%$ used by the reaction. Nevertheless, a splitting implementation without multiresolution shows the opposite behavior : $\sim 87\%$ for the reaction against $\sim 12\%$ for the diffusion integration. Taking into account these values, a more precise analysis for the first case, reveals that out of the 70% previously mentioned, $\sim 68\%$ of the time is used to update the *phantom* values at each internal stage of Rock4, leaving only a $\sim 2\%$ utilized by the diffusion integration itself. These results clearly indicate the potential and very high gains that might be achieved if the updating procedure is properly optimized. As a matter of fact, these issues are related to more technical programming aspects of the method.

These numerical illustrations show the feasibility and the potential advantages of the numerical strategy, keeping in mind that important improvements related to programming issues are still possible and even necessary. However, the main advantage of the proposed numerical strategy is the very high *data compression* achieved. This is a very important issue when we aim at performing more precise simulations of very discretized domains or whenever the smallest space scales might impose prohibitive uniform meshes.

4.3. 2D BZ equation

We now consider the 2D application of problem (40) with homogeneous Neumann boundary conditions and the following parameters, taken from a preliminary study [26] : $\epsilon = 10^{-2}$, $\mu = 10^{-5}$, $f = 1.6$ and $q = 2.10^{-3}$, with diffusion coefficients $D_a = 2.5 \times 10^{-3}$, $D_b = 2.5 \times 10^{-3}$ and $D_c = 1.5 \times 10^{-3}$. The phenomenon is studied over a time domain of $[0, 4]$ and a space region of $[0, 1] \times [0, 1]$.

We are interested in conducting the same previous analysis of performance of the numerical strategy proposed. Therefore, we consider the *quasi-exact* resolution of problem (40) on an uniform mesh of 256×256 performed by Radau5 with fine tolerances. The main limitation to perform such computation on finer grids comes from the important memory requirements of Radau5. We can easily overcome this problem with a splitting strategy, while further advantages are obtained through a combined multiresolution-splitting strategy.

Let us consider an application of the proposed numerical strategy with 8 nested dyadic grids with $N = 2^{2\times 8} = 65536$ cells on the finest grid $j = J = 8$. For the time integration method, the same Strang splitting technique presented in the previous case is implemented with splitting time step $\Delta t \approx 3.9\times 10^{-3}$. The corresponding solutions from the initial condition until the development of the spiral waves can be seen through Figures 6, 7 and 8, where the colors represent the grid levels. Figure 9 shows the evolution of the corresponding adapted grid with $\varepsilon = 1.10^{-2}$ as the thresholding value into (28).

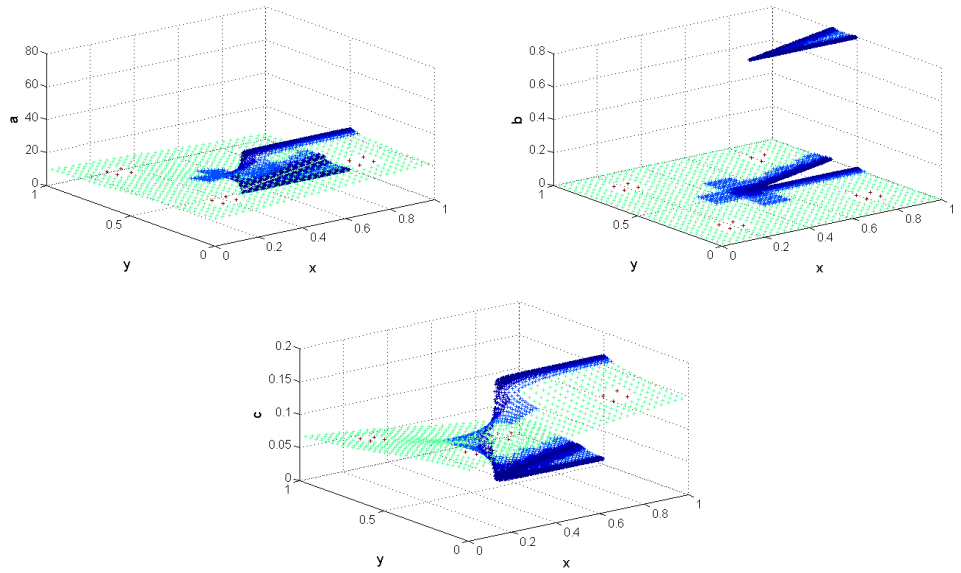

Figure 6: 2D BZ spiral waves. Initial conditions ($t = 0$) on the initial adapted grid. Variable a (top left), b (top right) and c (bottom) where the colors represent the grid levels. Finest grid : 256×256.

Table 4 shows the *data compression* achieved for different thresholding values. Taking into account the previous results in the 1D case, the current results show that the *data compression* might not be large enough to achieve the operator splitting accuracy, important *data compression* and savings in time consumption at the same time.

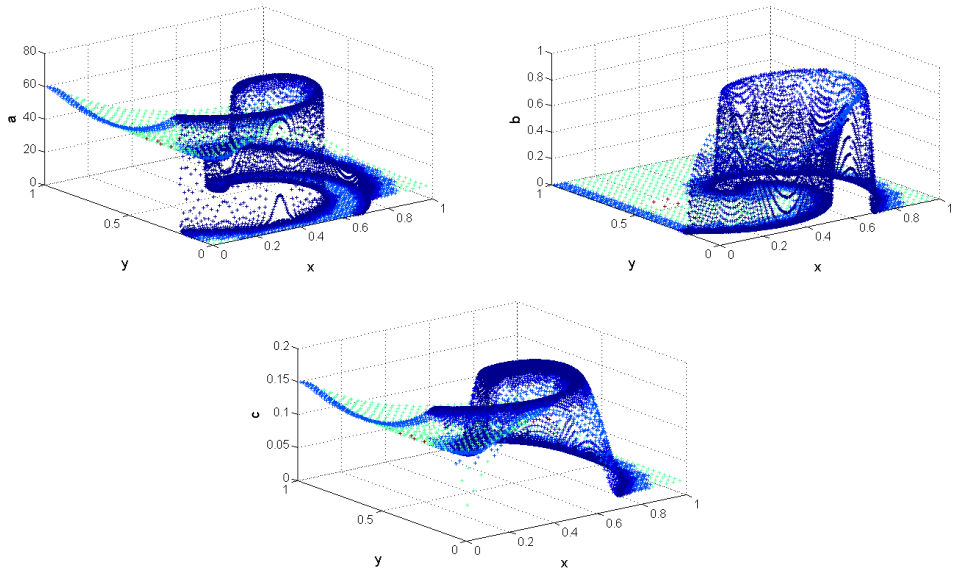

Figure 7: 2D BZ spiral waves. MR solutions on an adapted grid with $\varepsilon = 1.10^{-2}$ at $t = 2$. Variable a (top left), b (top right) and c (bottom) where the colors represent the grid levels. Finest grid : 256×256.

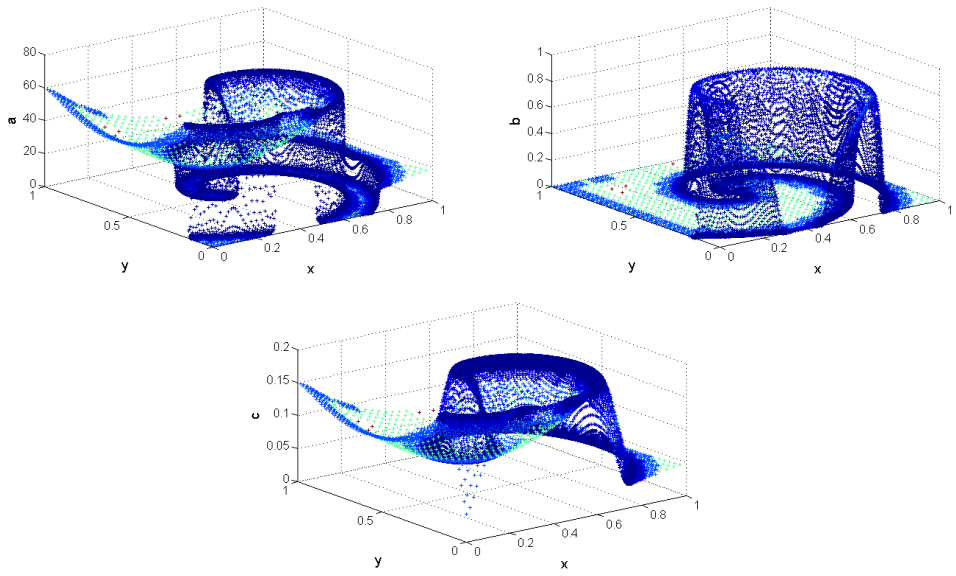

Figure 8: 2D BZ spiral waves. MR solutions on an adapted grid with $\varepsilon = 1.10^{-2}$ at $t = 4$. Variable a (top left), b (top right) and c (bottom) where the colors represent the grid levels. Finest grid : 256×256.

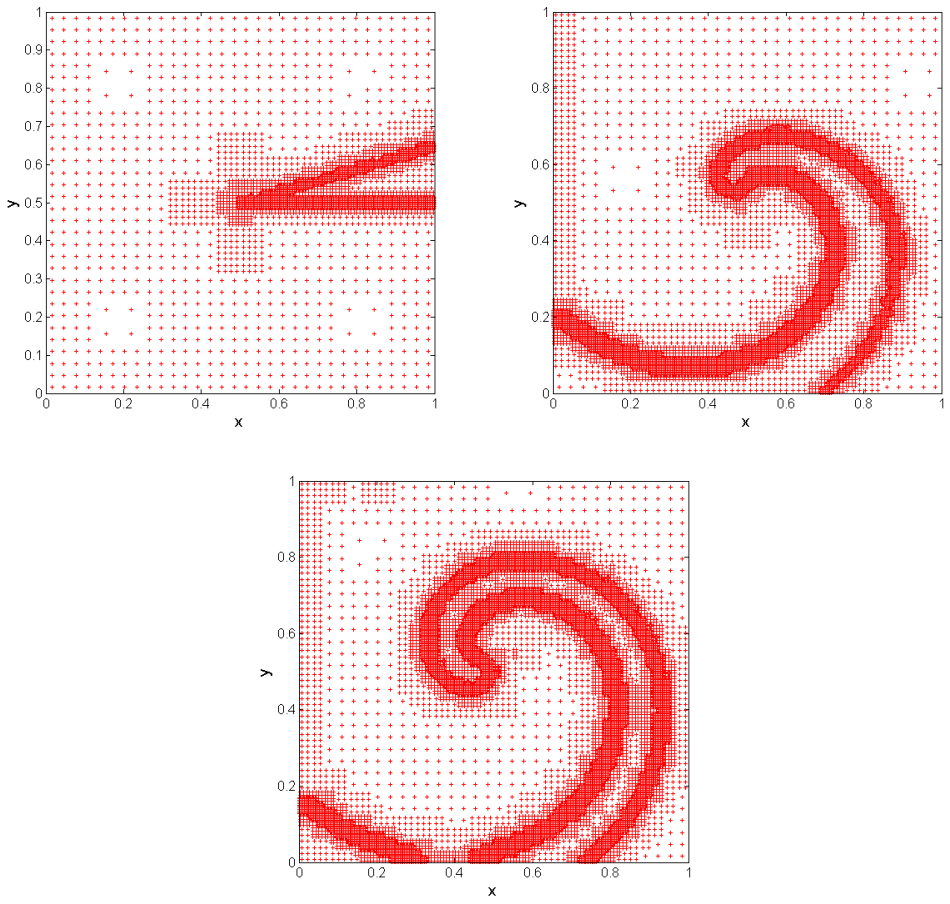

Figure 9: 2D BZ. Adapted grids at $t = 0$ (top left), $t = 2$ (top right) and $t = 4$ (bottom), $\varepsilon = 1.10^{-2}$. Finest grid : 256×256.

ε	**t = 2**		**t = 4**	
	Leaves number	Comp %	Leaves number	Comp %
$\mathbf{1.10^{-1}}$	10594	83.97	14035	78.58
$\mathbf{1.10^{-2}}$	15244	76.74	18874	71.20
$\mathbf{1.10^{-3}}$	27544	57.97	29182	55.47
$\mathbf{1.10^{-4}}$	55207	15.76	51934	20.76

Table 4: 2D BZ. *Data compression* for different thresholding values ε. Finest grid : 256×256.

The L^2-error estimates based also on the expression (41) and computed in the same way as before, show that for $\varepsilon \leq 1.10^{-2}$, the multiresolution errors become negligible compared to the splitting ones. Figure 10 shows these error estimates. The proportionality given by (30) is once

again verified while the splitting error depends only on the splitting time step.

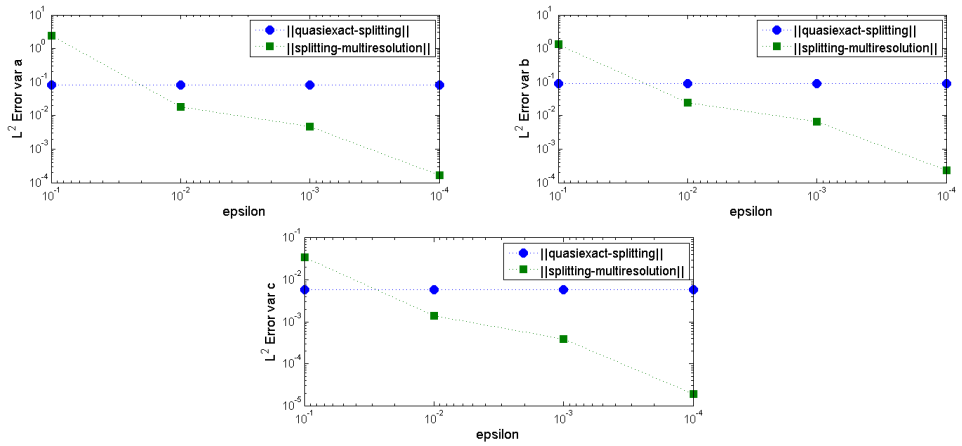

Figure 10: 2D BZ. L^2 errors at $t = 4$. Variable a (top left), b (top right) and c (bottom). Finest grid : 256×256.

Considering the time consumption of the different alternatives summarized in Table 5, we can note that for this configuration, a splitting strategy is always faster for the same final accuracy, which is the one fixed by the splitting error. As a matter of fact, Table 4 shows that for *data compressions* $\sim$ 80% or less, the extra computations previously mentioned are not fully compensated.

Method	**Time**	**Ratio**	**Comp %**
Quasi exact	16833	**1**	0
Splitting	2057	**8.18**	0
MR 1.10^{-1}	3095	**5.44**	81
MR 1.10^{-2}	4176	**4.03**	74
MR 1.10^{-3}	6699	**2.51**	57
MR 1.10^{-4}	12221	**1.38**	37

Table 5: 2D BZ. Computation time in seconds, ratio computed with respect to the *quasi-exact* computation time and average *data compression*. Finest grid : 256×256.

A similar analysis of the time elapsed on each process gives the same proportions previously seen in the 1D case for the reaction and the diffusion integration, and in particular, for the updating process for the diffusion. Nevertheless, an adapted grid implies very important savings in terms of memory requirements and moreover, makes possible simulations with a still higher number of points. Hence, this numerical strategy becomes really interesting on behalf of more accurate simulations of phenomena requiring very fine spatial resolution, for which classical strategies are simply unfeasible with conventional computational resources.

Therefore, we now consider the same problem but with finer spatial discretizations on the finest grid at $j = J$. Table 6 summarizes these results for the thresholding values ε of 1.10^{-1} and 1.10^{-2}. As expected, the *data compression* increases with the number of levels which is defined

by the spatial discretization on the finest grid. In fact, looking at expression (28), for a fixed thresholding value ε on the finest grid J, more intermediary thresholding values are possible in order to utterly discriminate the space scales present.

Grid	**Levels**	ε	**t = 2**		**t = 4**	
$j = J$	J		Leaves	Comp %	Leaves	Comp %
128×128	7	1.10^{-1}	4507	72.49	5884	64.09
128×128	7	1.10^{-2}	6718	59.00	8008	51.12
256×256	8	1.10^{-1}	10594	83.97	14035	78.58
256×256	8	1.10^{-2}	15244	76.74	18874	71.20
512×512	9	1.10^{-1}	21205	91.91	31315	88.05
512×512	9	1.10^{-2}	33712	87.14	44917	82.87
1024×1024	10	1.10^{-1}	40921	96.10	60511	94.23
1024×1024	10	1.10^{-2}	80425	92.33	110428	89.47

Table 6: 2D BZ. *Data compression* for different thresholding values ε and finest grids.

Considering the time consumption of the case 512×512, for which a splitting resolution is still feasible, the latter is more expensive than the splitting/multiresolution strategy for $\varepsilon = 1.10^{-1}$ and of the same order for $\varepsilon = 1.10^{-2}$. Hence, for the case 1024×1024 with $\varepsilon = 1.10^{-2}$ we can expect a better global performance, considering the higher *data compression* achieved. The multiresolution errors for the case 512×512 give similar results to those previously obtained for the 256×256 case, however, we can not compute the *quasi-exact* solution with the same resources in order to estimate the splitting error.

As an illustrating example, let us consider the case 1024×1024 and the corresponding memory requirements for each solver. Moreover, as a measure of this requirement, let us consider the array size of the working space needed by Radau5 and Rock4 in a general case :

- *Radau5 :* $L_1 = 4 \times W_1 \times W_1 + 12 \times W_1 + 20$ (from [43])
- *Rock4 :* $L_2 = 8 \times W_2$ (from [2])

where W_1 and W_2 are the number of unknowns solved by Radau5 and Rock4. In the case of an uniform mesh, the total number of unknowns is $W = 3 \times 1024 \times 1024 \approx 3.15 \times 10^6$ and thus, the global size L required for each solver is :

- *Quasi-exact :* $W_1 = W \approx 3.15 \times 10^6$ and $L = L_1 \approx 4 \times 10^{13}$
- *Splitting :* $W_1 = 3$, $W_2 = W \approx 3.15 \times 10^6$ and $L = L_1 + L_2 \approx 2.5 \times 10^7$

In the case of the proposed numerical strategy considering an average *data compression*, we have

- *MR* 1.10^{-1} *:* $W_1 = 3$, $W_2 = 0.05 \times W \approx 1.5 \times 10^5$ and $L = L_1 + L_2 \approx 1.2 \times 10^6$
- *MR* 1.10^{-2} *:* $W_1 = 3$, $W_2 = 0.09 \times W \approx 2.9 \times 10^5$ and $L = L_1 + L_2 \approx 2.3 \times 10^6$

Therefore, on the one hand, it is hopeless trying to solve problem (40) with the *quasi-exact* scheme for these very fine discretizations, at least with standard computational resources. For instance, for the one- and two-dimensional simulations we have used so far, an Intel(R) Core(TM)2

processor of 2 GHz with memory capacity of 2 Gb. And on the other hand, also a splitting strategy becomes more difficult to implement since the diffusion term is solved considering the entire spatial domain at once. Finally, a major advantage of the proposed numerical strategy is the possibility of representing the results in a highly compressed way. As an example, Figure 11 shows the adapted grid obtained for the 1024×1024 configuration with a thresholding value of $\varepsilon = 1.10^{-2}$.

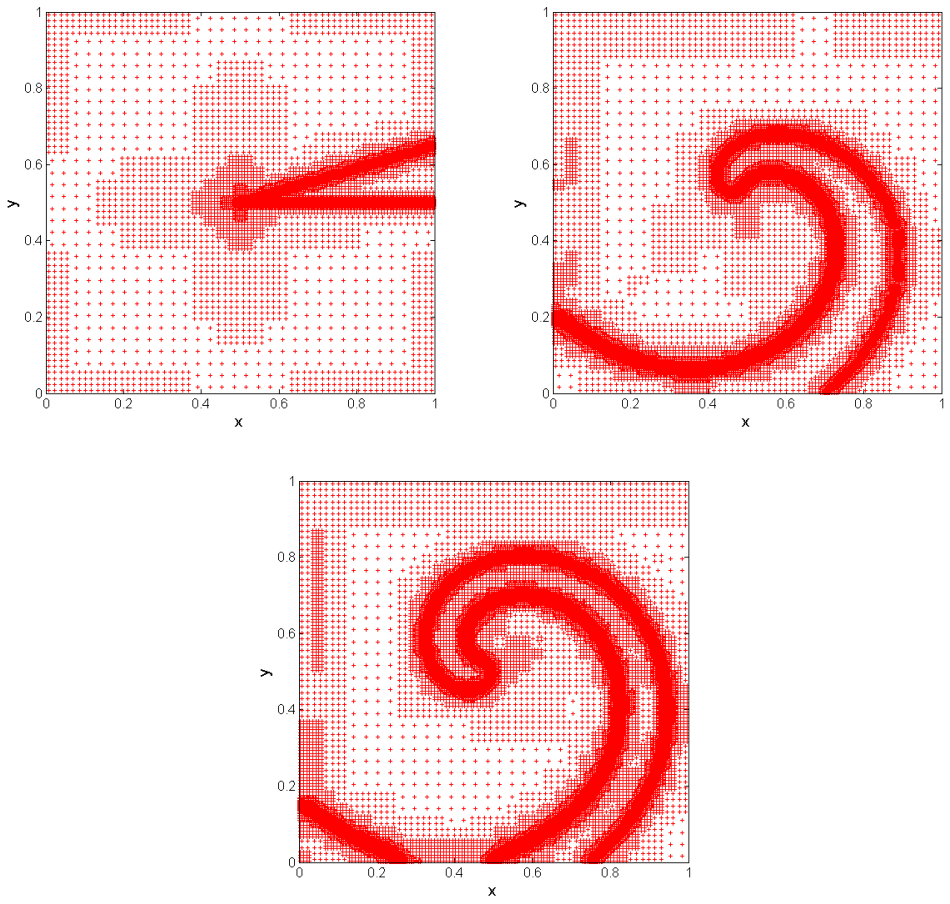

Figure 11: 2D BZ. Adapted grid at $t = 0$ (top left), $t = 2$ (top right) and $t = 4$ (bottom), $\varepsilon = 1.10^{-2}$. Finest grid : 1024×1024.

4.4. 3D BZ equation

We consider problem (40), now in a 3D configuration with the same parameters considered in the 2D case for a time domain of $[0, 2]$ and in a space region of $[0, 1] \times [0, 1] \times [0, 1]$. We first apply the proposed numerical strategy in a configuration of 8 nested dyadic grids with $N = 2^{3\times8} = 16777216 = 256 \times 256 \times 256$ cells on the finest grid $j = J = 8$.

Based on the previous results for the 2D case, we consider thresholding values of $\varepsilon = 1.10^{-1}$ and $\varepsilon = 1.10^{-2}$ and a splitting time step $\Delta t \approx 3.9 \times 10^{-3}$. For $\varepsilon = 1.10^{-1}$, the implementation of the proposed numerical strategy features *data compressions* that goes from 83.06% for the initial condition, 84.40% at $t = 1$ when the scroll waves are fully developed and 79.61% at final time $t = 2$. As a matter of fact, following (28), we see that the thresholding values for the intermediary meshes are smaller than for the 2D case and as a consequence, the *data compressions* are also smaller even though we consider the same parameters for this three-dimensional configuration (see Table 4 for the 2D results). In this way, for $\varepsilon = 1.10^{-2}$, an even smaller average *data compression* of $\sim 20.52\%$ is accomplished and, once again, the advantages of the method are enhanced for simulations requiring very fine spatial resolution.

Figures (12), (13) and (14) show the evolution of the finest grid $J = 8$ of the adapted grid from the initial condition until the final time of study for $\varepsilon = 1.10^{-1}$. We can clearly distinguish the developed scroll waves where the colors represent the values of variable a. As expected, the finest regions correspond to the neighborhood of the wavefront. For these three-dimensional implementations, we use an AMD-Shanghai processor of 2.7 GHz with memory capacity of 32 Gb.

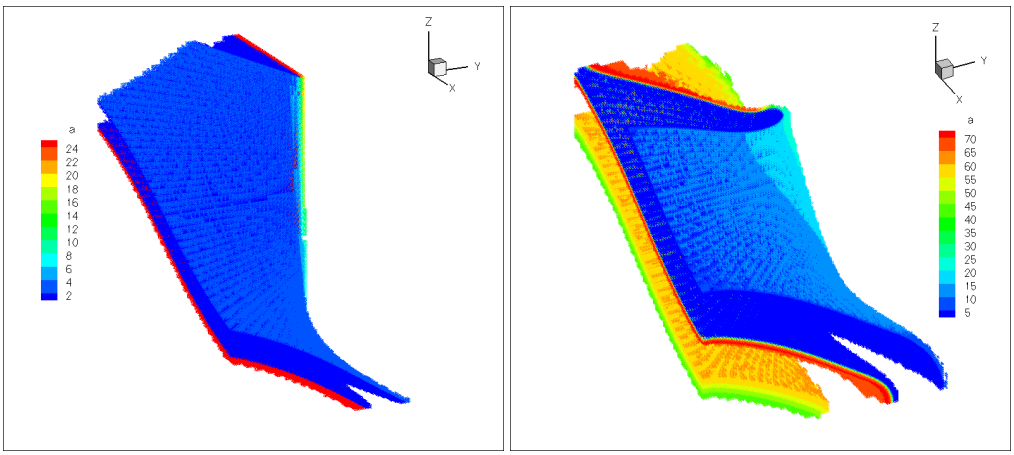

Figure 12: 3D BZ scroll wave. Finest grid of the adapted grid at $t = 0$ (left) and $t = 0.5$ (right), $\varepsilon = 1.10^{-1}$, where the colors represent values of variable a. Finest grid : $256 \times 256 \times 256$.

Finally, in order to explore the feasibility and potential advantages of the method, let us consider 9 nested dyadic grids with $N = 2^{3\times 9} = 134217728 = 512 \times 512 \times 512$ cells on the finest grid $j = J = 9$. In this case and with the mentioned resources, it is not possible to initialize the problem directly on the finest grid anymore, and thus, the initialization takes place on a intermediary grid like $j = 8$. With a thresholding value of $\varepsilon = 1.10^{-1}$ and a splitting time step $\Delta t \approx 7.8 \times 10^{-3}$, the implementation of the proposed numerical strategy features *data compressions* that goes from 95.54% for the initial condition, 89.62% at $t = 1$ when the scroll waves are fully developed and 87.05% at final time $t = 2$.

For this configuration, a two times larger splitting time step is chosen in order to have splitting and multiresolution errors potentially of the same order, considering the higher multiresolution error given by $\varepsilon = 1.10^{-1}$. In fact, computations with smaller thresholding values are not feasible with the considered computational resource and at the same time, we aim at maintaining an

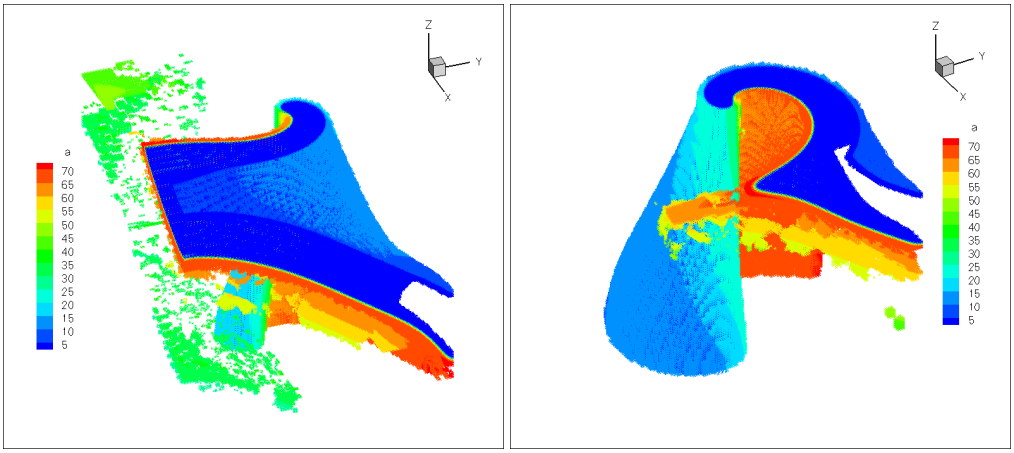

Figure 13: 3D BZ scroll wave. Finest grid of the adapted grid at $t = 1$ (left) and $t = 1.5$ (right), $\varepsilon = 1.10^{-1}$, where the colors represent values of variable a. Finest grid : $256 \times 256 \times 256$.

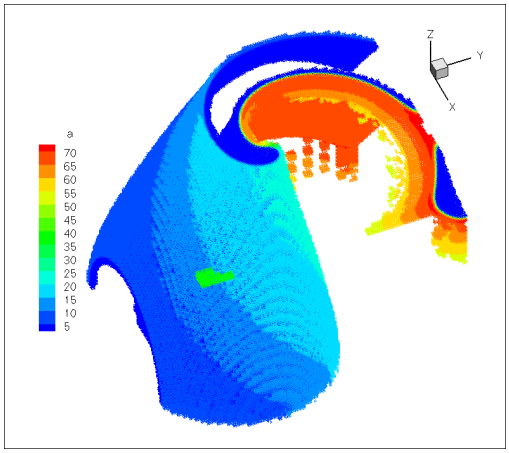

Figure 14: 3D BZ scroll wave. Finest grid of the adapted grid at $t = 2$, $\varepsilon = 1.10^{-1}$, where the colors represent values of variable a. Finest grid : $256 \times 256 \times 256$.

accurate physical resolution of the problem with the mentioned splitting time step. Figures (15), (16) and (17) show the evolution of the finest grid of the adapted grid from the initial condition until the final time of study. Once again, the finest regions correspond to the neighborhood of the wavefront and considering a qualitative comparison, both resolutions for 256^3 and 512^3 points give the same results.

Performing the same comparison concerning the memory requirements, the total number of unknowns for this case is $W = 3 \times 512 \times 512 \times 512 \approx 4.03 \times 10^8$ and thus, the global size L required for each solver is :

- *Quasi-exact :* $W_1 = W \approx 4.03 \times 10^8$ and $L = L_1 \approx 6.5 \times 10^{17}$

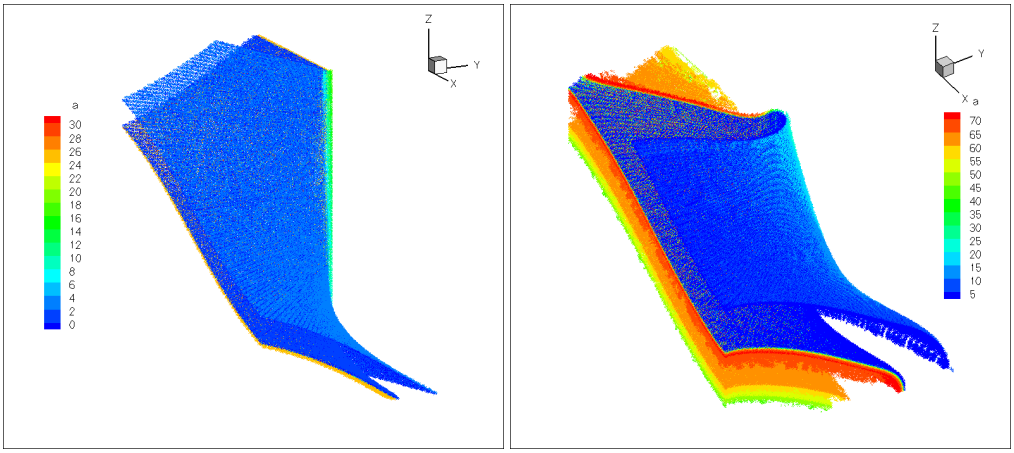

Figure 15: 3D BZ scroll wave. Finest grid of the adapted grid at $t = 0$ (left) and $t = 0.5$ (right), $\varepsilon = 1.10^{-1}$, where the colors represent values of variable a. Finest grid : $512 \times 512 \times 512$.

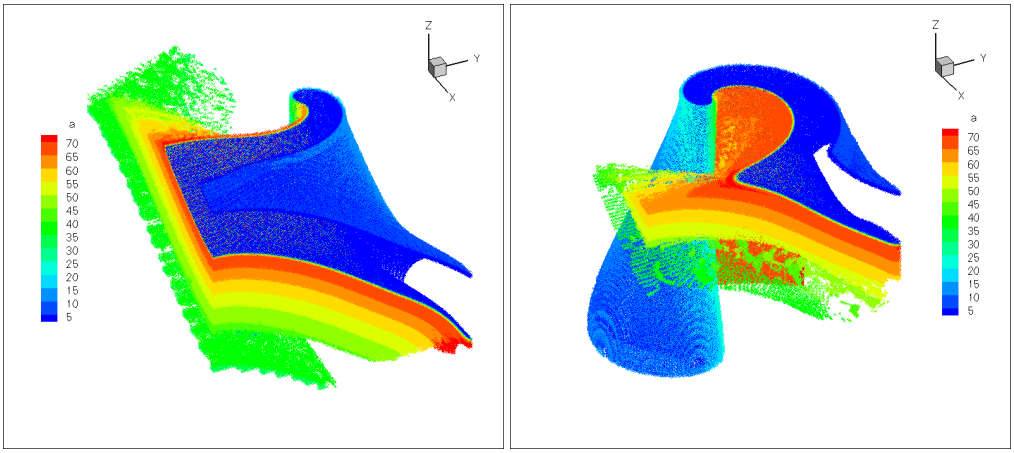

Figure 16: 3D BZ scroll wave. Finest grid of the adapted grid at $t = 1$ (left) and $t = 1.5$ (right), $\varepsilon = 1.10^{-1}$, where the colors represent values of variable a. Finest grid : $512 \times 512 \times 512$.

- *Splitting :* $W_1 = 3$, $W_2 = W \approx 4.03 \times 10^8$ and $L = L_1 + L_2 \approx 3.2 \times 10^9$
- *MR* 1.10^{-1} *:* $W_1 = 3$, $W_2 = 0.13 \times W \approx 5.3 \times 10^7$ and $L = L_1 + L_2 \approx 4.2 \times 10^8$

Therefore, important resource restrictions are necessarily associated to solvers acting on an uniform mesh whenever an accurate spatial resolution like this is required. In particular, numerical strategies without grid adaptation for this level of resolution are usually out of reach. Concerning the proposed numerical strategy, the feasibility of performing very large simulations is proven, and even greater advantages might be obtained with further and appropriate optimizations.

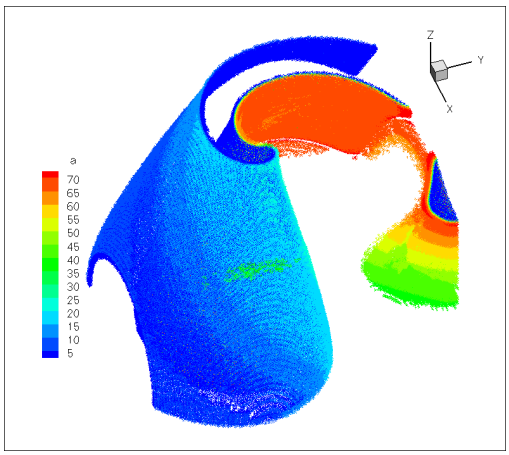

Figure 17: 3D BZ scroll wave. Finest grid of the adapted grid at $t = 2$, $\varepsilon = 1.10^{-1}$, where the colors represent values of variable a. Finest grid : $512 \times 512 \times 512$.

5. Conclusions

The numerical illustrations as well as the theoretical considerations conducted have shown that the splitting techniques are highly recommended for the simulation of this kind of multi-scale problems. This decomposition allows the use of dedicated solvers for each type of problem, but also relies on a numerical analysis already developed about the numerical behavior of these methods in this particular multi-scale context. One of the key points is that usually prohibitive small integration time steps are not required. Thus, very simple but optimal combination of heterogeneous numerical strategies can be accomplished.

As an outstanding example, the present work proposes a very simple but solid numerical strategy that couples adaptive multiresolution techniques with splitting techniques plus optimal time integration methods. The highly compressed *data representation* as well as the accurate and feasible resolution of these localized and stiff phenomena prove that large computational domains previously out of reach can be successfully simulated with conventional computational resources. For the moment, we have focused on reaction-diffusion systems, however, these numerical developments are meant in order to be extended to convecting cases.

At the same time, an important amount of work is still in progress concerning programming features such as data structures, optimized routines and parallelization strategies in order to construct yet more efficient algorithms. Finally, from a theoretical point of view, the continuous construction of a more precise numerical analysis of the different aspects of the method may surely lead to even better characterizations of the splitting errors and with them, to improve these numerical strategies; this is a particular topic of our current research.

Acknowledgements

This research was supported by a fundamental project grant from ANR (French National Research Agency - ANR Blancs) *Séchelles* (Project leader S. Descombes), by a CNRS Ph.D. grant

for M. Duarte from the Institute of Mathematics (INSMI) and Engineering Institute (INSIS) of CNRS, and through a PEPS Maths-ST2I CNRS project (V. Louvet).

References

[1] Abdulle, A., 2000. On roots and error constants of optimal stability polynomials. BIT 40 (1), 177–182.
[2] Abdulle, A., 2002. Fourth order Chebyshev methods with recurrence relation. SIAM J. Sci. Comput. 23 (6), 2041–2054 (electronic).
[3] Abdulle, A., Medovikov, A., 2001. Second order Chebyshev methods based on orthogonal polynomials. Numer. Math. 90 (1), 1–18.
[4] Akrivis, G., Crouzeix, M., Makridakis, C., 1998. Implicit-explicit multistep finite element methods for nonlinear parabolic problems. Math. Comp. 67 (222), 457–477.
[5] Barkley, D., 1991. A model for fast computer simulation of waves in excitable media. Physica D 49, 61–70.
[6] Bell, J., Berger, M. J., Saltzman, J., Welcome, M., 1994. Three-dimensional adaptive mesh refinement for hyperbolic conservation laws. SIAM J. Sci. Comput. 15, 127–138.
[7] Berger, M. J., Colella, P., 1989. Local adaptive mesh refinement for shock hydrodynamics. J. Comput. Phys. 82, 67–84.
[8] Berger, M. J., Oliger, J., 1984. Adaptive mesh refinement for hyperbolic partial differential equations. J. Comput. Phys. 53, 484–512.
[9] Bihari, B. L., Harten, A., 1997. Multiresolution schemes for the numerical solution of 2-D conservation laws I. SIAM J. Sci. Comput. 18 (2), 315–354.
[10] Brandt, A., 1977. Multi-level adaptive solutions to boundary value problems. Math. Comp. 31, 333–390.
[11] Bürger, R., Ruiz-Baier, R., Schneider, K., 2010. Adaptive multiresolution methods for the simulation of waves in excitable media.Private communication.
[12] Butcher, J. C., 1964. Implicit Runge-Kutta processes. Math. Comp. 18, 50–64.
[13] Charlette, F., Meneveau, C., Veynante, D., 2002. A power-law flame wrinkling model for LES of premixed turbulent combustion. part I: non-dynamic formulation and initial tests. Combustion and Flame 131 (1-2), 159–180.
[14] Cohen, A., 2000. Wavelet methods in numerical analysis. Vol. 7. Elsevier, Amsterdam.
[15] Cohen, A., Daubechies, I., Feauveau, J. C., 1992. Biorthogonal bases of compactly supported wavelets. Comm. Pure Appl. Math. 45, 485–560.
[16] Cohen, A., Kaber, S., Müller, S., Postel, M., 2003. Fully adaptive multiresolution finite volume schemes for conservation laws. Mathematics of Computation 72, 183–225.
[17] Colin, O., Ducros, F., Veynante, D., Poinsot, T., 2000. A thickened flame model for large eddy simulations of turbulent premixed combustion. Physics of Fluids 12 (7), 1843–1863.
[18] Dahlquist, G., 1963. A special stability problem for linear multistep methods. Nordisk Tidskr. Informations-Behandling 3, 27–43.
[19] D'Angelo, Y., 1994. Analyse et simulation numérique de phénomènes liés à la combustion supersonique. Ph.D. thesis, Ecole Nationale des Ponts et Chaussées.
[20] D'Angelo, Y., Larrouturou, B., 1995. Comparison and analysis of some numerical schemes for stiff complex chemistry problems. RAIRO Modél. Math. Anal. Numér. 29 (3), 259–301.
[21] Day, M. S., Bell, J. B., 2000. Numerical simulation of laminar reacting flows with complex chemistry. Combust. Theory Modelling 4, 535–556.
[22] Descombes, S., 2001. Convergence of a splitting method of high order for reaction-diffusion systems. Math. Comp. 70 (236), 1481–1501.
[23] Descombes, S., Dumont, T., 2008. Numerical simulation of a stroke : Computational problems and methodology. Progress in Biophysics and Molecular Biology 97, 40–53.
[24] Descombes, S., Dumont, T., Louvet, V., Massot, M., 2007. On the local and global errors of splitting approximations of reaction-diffusion equations with high spatial gradients. Int. J. of Computer Mathematics 84 (6), 749–765, special Issue on Splitting Methods for Differential Equations, Ed. Q. Sheng.
[25] Descombes, S., Dumont, T., Louvet, V., Massot, M., Laurent, F., Beaulaurier, J., 2010. Operator splitting techniques for multi-scale reacting waves and application to low mach number flames with complex chemistry : theoretical and numerical aspects. In progress.
[26] Descombes, S., Dumont, T., Massot, M., 2003. Operator splitting for stiff nonlinear reaction-diffusion systems: order reduction and application to spiral waves. In: Patterns and waves (Saint Petersburg, 2002). AkademPrint, St. Petersburg, pp. 386–482.
[27] Descombes, S., Massot, M., 2004. Operator splitting for nonlinear reaction-diffusion systems with an entropic structure: singular perturbation and order reduction. Numer. Math. 97 (4), 667–698.

[28] Descombes, S., Schatzman, M., 2002. Strang's formula for holomorphic semi-groups. J. Math. Pures Appl. 81 (1), 93–114.
[29] Deuflhard, P., 1974. A modified Newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting. Numer. Math. 22, 289–315.
[30] Deuflhard, P., 2004. Newton Methods for Nonlinear Problems. Springer-Verlag, Affine invariance and adaptive algorithms.
[31] Dowle, M., Mantel, R. M., Barkley, D., 1997. Fast simulations of waves in three-dimensional excitable media. Int. J. Bif. Chaos 7, 2529–2545.
[32] Dumont, T., Duarte, M., Descombes, S., Dronne, M., Massot, M., Louvet, V., 2010. Optimal numerical strategy for human stroke simulation in complex 3D geometry of the brain. Submitted to Bulletin of Mathematical Biology, available on HAL.
[33] Echekki, T., 2009. Multiscale methods in turbulent combustion: strategies and computational challenges. Computational Science & Discovery 2.
[34] Ehle, B., 1973. *A*-stable methods and Padé approximations to the exponential. SIAM J. Math. Anal. 4, 671–680.
[35] Epstein, I. R., Pojman, J. A., 1998. An Introduction to Nonlinear Chemical Dynamics. Oxford University Press, Oscillations, Waves, Patterns and Chaos.
[36] Giovangigli, V., 1999. Multicomponent flow modeling. Birkhäuser Boston Inc., Boston, MA.
[37] Gokoglu, S. A., 1988. Significance of vapor phase chemical reactions on cvd rates predicted by chemically frozen and local thermochemical equilibrium boundary layer theories. J. Electrochem. Soc.
[38] Gottschlich-Müller, B., Müller, S., 1999. Adaptive finite volume schemes for conservation laws based on local multiresolution techniques. In: Hyperbolic problems: Theory, numerics, applications. M. Fey and R. Jeltsch, pp. 385–394.
[39] Gray, P., Scott, S. K., 1994. Chemical oscillations and instabilites. Oxford University Press.
[40] Grenier, E., Dronne, M. A., Descombes, S., Gilquin, H., Jaillard, A., Hommel, M., Boissel, J. P., 2008. A numerical study of the blocking of migraine by Rolando sulcus. Progress in Biophysics and Molecular Biology 97, 54–59.
[41] Hairer, E., Lubich, C., Roche, M., 1988. Error of Runge-Kutta methods for stiff problems studied via differential algebraic equations. BIT 28 (3), 678–700.
[42] Hairer, E., Lubich, C., Wanner, G., 2006. Geometric Numerical Integration, 2nd Edition. Springer-Verlag, Berlin, Structure-Preserving Algorithms for Odinary Differential Equations.
[43] Hairer, E., Wanner, G., 1996. Solving ordinary differential equations II, 2nd Edition. Springer-Verlag, Berlin, Stiff and differential-algebraic problems.
[44] Hairer, E., Wanner, G., 1999. Stiff differential equations solved by Radau methods. J. Comput. Appl. Math. 111 (1-2), 93–111, numerical methods for differential equations (Coimbra, 1998).
[45] Harten, A., 1995. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. Comm. Pure and Applied Math. 48, 1305–1342.
[46] Hundsdorfer, W., Verwer, J. G., 2003. Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations. Springer-Verlag, Berlin.
[47] Jahnke, W., Skaggs, W. E., Winfree, A. T., 1989. Chemical vortex dynamics in the Belousov-Zhabotinsky reaction and in the two-variable Oregonator model. J. Phys. Chem. 93, 740–749.
[48] Kim, J., Cho, S., 1997. Computation accuracy and efficiency of the time-splitting method in solving atmosperic transport-chemistry equations. Atmos. Environ.
[49] Knio, O. M., Najm, H. N., Wyckoff, P. S., 1999. A semi-implicit numerical scheme for reacting flow. II. stiff, operator-split formulation. J. Comput. Phys. 154, 482–467.
[50] Marchuk, G. I., 1990. Splitting and alternating direction methods. In: Handbook of numerical analysis, Vol. I. North-Holland, Amsterdam, pp. 197–462.
[51] Massot, M., 2002. Singular perturbation analysis for the reduction of complex chemistry in gaseous mixtures using the entropic structure. Discrete Contin. Dyn. Syst. Ser. B 2 (3), 433–456.
[52] McRae, G. J., Goodin, W. R., Seinfeld, J. H., 1982. Numerical solution of the atmospheric diffusion equation for chemically reacting flows. J. Comput. Phys. 45 (1), 1–42.
[53] Müller, S., 2003. Adaptive multiscale schemes for conservation laws. Vol. 27. Springer-Verlag, Heidelberg.
[54] Najm, H. N., Knio, O. M., 2005. Modeling Low Mach number reacting flow with detailed chemistry and transport. Journal of Scientific Computing 25 (1/2), 263–287.
[55] Najm, H. N., Wyckoff, P. S., Knio, O. M., 1998. A semi-implicit numerical scheme for reacting flow. I. stiff chemistry. J. Comput. Phys. 143, 381–402.
[56] Radau, R., 1880. Étude sur les formules d'approximation qui servent à calculer la valeur numérique d'une intégrale definie. J. Math. Pures Appl. (6), 283–336.
[57] Roussel, O., Schneider, K., Tsigulin, A., Bockhorn, H., 2003. A conservative fully adaptive multiresolution algorithm for parabolic PDEs. J. Comput. Phys. 188 (2), 493–523.
[58] Schatzman, M., 2002. Toward non commutative numerical analysis : high order integration in time. Journal of

Scientific Computing 17 (1-3), 107–125.
[59] Shampine, L. F., Sommeijer, B. P., Verwer, J. G., 2006. IRKC: an IMEX solver for stiff diffusion-reaction PDEs. J. Comput. Appl. Math. 196 (2), 485–497.
[60] Smooke, M. D., 1983. Error estimate for the modified Newton method with applications to the solution of nonlinear, two-point boundary value problems. J. Optim. Theory Appl. 39 (4), 489–511.
[61] Sommeijer, B. P., Shampine, L. F., Verwer, J. G., 1998. RKC: an explicit solver for parabolic PDEs. J. Comput. Appl. Math. 88 (2), 315–326.
[62] Sommeijer, B. P., Verwer, J. G., 1980. 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.
[63] Sportisse, B., 1999. Contribution à la modélisation des écoulements réactifs : réduction des modèles de cinétique chimique et simulation de la pollution atmosphérique. Ph.D. thesis, Ecole Polytechnique.
[64] Sportisse, B., 2000. An analysis of operator splitting techniques in the stiff case. J. Comput. Phys. 161 (1), 140–168.
[65] Sportisse, B., Djouad, R., 2000. Reduction of chemical kinetics in air pollution modeling. J. Comput. Phys. 164 (2), 354–376.
[66] Strang, G., 1963. Accurate partial difference methods. I. Linear Cauchy problems. Arch. Rational Mech. Anal. 12, 392–402.
[67] Strang, G., 1968. On the construction and comparison of difference schemes. SIAM J. Numer. Anal. 5, 506–517.
[68] Sun, P., 1996. A pseudo non-time splitting method in air quality modeling. J. Comp. Phys.
[69] Témam, R., 1969. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires. I. Arch. Rational Mech. Anal. 32, 135–153.
[70] Témam, R., 1969. Sur l'approximation de la solution des équations de Navier-Stokes par la méthode des pas fractionnaires. II. Arch. Rational Mech. Anal. 33, 377–385.
[71] van der Houwen, P. J., Sommeijer, B. P., 1980. On the internal stability of explicit, m-stage Runge-Kutta methods for large m-values. Z. Angew. Math. Mech. 60 (10), 479–485.
[72] Verwer, J. G., 1996. Explicit Runge-Kutta methods for parabolic partial differential equations. Appl. Numer. Math. 22 (1-3), 359–379, special issue celebrating the centenary of Runge-Kutta methods.
[73] Verwer, J. G., Sommeijer, B. P., 2004. An implicit-explicit Runge-Kutta-Chebyshev scheme for diffusion-reaction equations. SIAM J. Sci. Comput. 25 (5), 1824–1835 (electronic).
[74] Verwer, J. G., Sommeijer, B. P., Hundsdorfer, W., 2004. RKC time-stepping for advection-diffusion-reaction problems. J. Comput. Phys. 201 (1), 61–79.
[75] Verwer, J. G., Sportisse, B., 1998. Note on operator splitting in a stiff linear case. Rep. MAS-R9830.
[76] Yanenko, N. N., 1971. The method of fractional steps. The solution of problems of mathematical physics in several variables. Springer-Verlag, New York.