Adaptive numerical methods in time and space for the simulation of multi-scale reaction fronts

Max Duarte

École Centrale Paris

THÈSE

présentée par

Max DUARTE

pour l'obtention du

GRADE de DOCTEUR

Formation doctorale : Mathématiques Appliquées - Energétique

Laboratoire d'accueil : Laboratoire d'Énergétique Moléculaire et Macroscopique, Combustion (EM2C) du CNRS et de l'ECP

Méthodes numériques adaptatives pour la simulation de la dynamique de fronts de réaction multi-échelles en temps et en espace

Soutenue le 9 Décembre 2011

Jury :	MM.	Giovangigli	V.	Président
		Gander	M.	Rapporteur
		Hundsdorfer	W.	Rapporteur
		Poinsot	T.	Rapporteur
		Guermond	J.-L.	Examinateur
		Massot	M.	Directeur de thèse
		Descombes	S.	Co-directeur de thèse
		Candel	S.	Invité
		Larroya-Huguet	J.-C	Invité
		Rolón	J.-C.	Invité
		Tenaud	C.	Invité

Ecole Centrale des Arts et Manufactures
Grand Établissement sous tutelle
du Ministère de l'Éducation Nationale
Grande Voie des Vignes
92295 Châtenay-Malabry Cedex
Tél : 33 (1) 41 13 10 00
Télex : 634 991 F EC PARIS

Laboratoire d'Énergétique Moléculaire et Macroscopique, Combustion (E.M2.C.)
UPR 288, CNRS et Ecole Centrale Paris
Tél : 33 (1) 41 13 10 31
Fax : 33 (1) 47 02 80 35

2011ECAP0057

Remerciements

Je tiens d'abord à remercier les membres du jury: Vincent Giovangigli, Martin Gander, Willem Hundsdorfer, Thierry Poinsot, Jean-Luc Guermond, Stéphane Descombes, Marc Massot, Sébastien Candel, Juan-Carlos Larroya-Huguet, Juan-Carlos Rolon et Christian Tenaud, de leur présence et de l'intérêt qu'ils ont porté à l'égard de ce travail. J'adresse tous mes remerciements en particulier à Martin Gander, Willem Hundsdorfer et Thierry Poinsot pour avoir accepté d'être rapporteurs de cette thèse ainsi que pour leurs remarques et leurs questions. Je remercie enfin Vincent Giovangigli de m'avoir fait l'honneur de présider le jury.

Concernant ce travail de thèse, je tiens tout d'abord à exprimer ma gratitude à Marc Massot pour la confiance qu'il m'a accordée ainsi que pour ses qualités humaines et scientifiques qui m'ont toujours rempli de motivation et enthousiasme. J'adresse également ma reconnaissance la plus sincère à Stéphane Descombes, Thierry Dumont, Violaine Louvet et Frédérique Laurent pour leurs précieuses aides et leur disponibilité en tout temps. Cela a été un vrai plaisir de travailler avec eux mais surtout une chance d'avoir fait partie de cette équipe merveilleuse de personnes. Je remercie aussi Christian Tenaud qui a collaboré énormément à ce projet qui commence à peine et qui promet beaucoup. De la même façon, je remercie Anne Bourdon et Zdeněk Bonaventura pour tous les jolis travaux que nous avons mené ensemble et qui surement continueront. Je salue la disponibilité de Sébastien Candel pour toutes ses remarques et tout le temps passé avec moi. Je suis très reconnaissant aussi à Stéphane de Chaisemartin, Lucie Fréret et Christophe Audouze, qui m'ont beaucoup aidé pendant mes débuts en thèse. Finalement, je remercie très particulièrement Juan-Carlos Rolon qui est à l'origine de cette aventure.

Je remercie ensuite les instituts de mathématiques et de sciences de l'ingénieur du CNRS, l'INSMI et l'INSIS, qui ont cofinancé ma bourse de docteur ingénieur. Je remercie également le laboratoire EM2C dans la personne de son ancien directeur, Nasser Darabiha et de sa directrice actuelle, Estelle Iacona de m'avoir fourni tout les moyens dont j'avais besoin pour réaliser mon travail de thèse dans de bonnes conditions. Je remercie particulièrement l'assistance de Stéphanie et Anne-Cécile. De manière plus générale, je remercie toutes les personnes avec qui j'ai eu le plaisir de discuter et partager des moments très agréables au sein du laboratoire. De façon non exhaustive, je salue la gentillesse de Thierry Schuller, Sébastien Ducruix, Laurent Zimmer ainsi que de nombreuses personnes que j'ai eu l'occasion et la chance de connaitre: Patrick, Ammar, Hector, Gabi, Itaru, Yoann, Farah, Diane, Jean C., Jean L., Vincent, Alexis, Layal, Adrien, Jingxuan, Fabien, Yamada, Sorour, Diana, Jaroslav, Juan Ignacio, Andrea,... Je salue spécialement mes camarades de bureau: Paul, Sébastien, Gaofeng, Tapish, Miloud, Frédéric, Pierre-Edouard, Raphaël, Benedetta, Alessandro, Théodore, et mes coéquipiers: Damien, François, Aymeric et Adam.

Je tiens à exprimer ma grande gratitude à tous ceux et celles avec qui j'ai partagé toutes ses années et qui ont aussi contribué à justifier ma venue. Pour citer quelques-uns, je commence

par mes camarades à l'ECP qui m'ont très bien accueilli depuis mon arrivée en France, en particulier Aude, Emilien et Charles. Je voudrais également adresser mes plus chaleureux remerciements à Marc, Marie-Laure, Ysé et Claire-Line ainsi qu'à Stéphane, Karine, Ambre et Anaëlle pour leur aimable hospitalité et soutien. Je remercie Zdeněk, Katka et Ema de leur très belle compagnie. Je remercie aussi ma chère amie Konstantina et sa mère pour tous les moments que nous avons partagé ensemble, ainsi que mon ami Roberto et sa famille pour leur générosité et amitié. Je remercie spécialement mon camarade d'aventure Beto pour sa compagnie depuis longtemps, et Christian pour sa patience et la confiance qu'il m'a toujours accordée. Une immense gratitude est décernée à mes frères Hans, Nicolás et Daniel avec qui j'ai vécu cette expérience inoubliable. Ce séjour ne serait pas complètement justifié sans l'existence de Maria, ma source d'inspiration, à qui je dédie ce travail. Je remercie également Stefano, Amabile et Valeria pour m'avoir accueilli dans leurs vies. Je tiens finalement à remercier du plus profond de mon cœur toute ma famille et mes amis au Paraguay, qui malgré la distance, m'ont toujours apporté leur soutien et envoyé leurs pensées, en particulier mes parents, Elsa et Luis, et mon frère Luis Maria.

Résumé

Nous abordons le développement d'une nouvelle génération de méthodes numériques pour la résolution des EDP évolutives qui modélisent des phénomènes multi-échelles en temps et en espace issus de divers domaines applicatifs. La raideur associée à ce type de problème, que ce soit via le terme source chimique qui présente un large spectre d'échelles de temps caractéristiques ou encore via la présence de fort gradients très localisés associés aux fronts de réaction, implique en général de sévères difficultés numériques. En conséquence, il s'agit de développer des méthodes qui garantissent la précision des résultats en présence de forte raideur en s'appuyant sur des outils théoriques solides, tout en permettant une implémentation aussi efficace. Même si nous étendons ces idées à des systèmes plus généraux par la suite, ce travail se focalise sur les systèmes de réaction-diffusion raides. La base de la stratégie numérique s'appuie sur une décomposition d'opérateur spécifique, dont le pas de temps est choisi de manière à respecter un niveau de précision donné par la physique du problème, et pour laquelle chaque sous-pas utilise un intégrateur temporel d'ordre élevé dédié. Ce schéma numérique est ensuite couplé à une approche de multirésolution spatiale adaptative permettant une représentation de la solution sur un maillage dynamique adapté. L'ensemble de cette stratégie a conduit au développement du code de simulation générique 1D/2D/3D académique MBARETE de manière à évaluer les développements théoriques et numériques dans le contexte de configurations pratiques raides issue de plusieurs domaines d'application. L'efficacité algorithmique de la méthode est démontrée par la simulation d'ondes de réaction raides dans le domaine de la dynamique chimique non-linéaire et dans celui de l'ingénierie biomédicale pour la simulation des accidents vasculaires cérébraux caractérisée par un terme source "chimique complexe". Pour étendre l'approche à des applications plus complexes et plus fortement instationnaires, nous introduisons pour la première fois une technique de séparation d'opérateur avec pas de temps adaptatif qui permet d'atteindre une précision donnée garantie malgré la raideur des EDP. La méthode de résolution adaptative en temps et en espace qui en résulte, étendue au cas convectif, permet une description consistante de problèmes impliquant une très large palette d'échelles de temps et d'espace et des scénarios physiques très différents, que ce soit la propagation des décharges répétitives pulsées nanoseconde dans le domaine des plasmas ou bien l'allumage et la propagation de flammes dans celui de la combustion. L'objectif de la thèse est l'obtention d'un solveur numérique qui permet la résolution des EDP raides avec contrôle de la précision du calcul en se basant sur des outils d'analyse numérique rigoureux, et en utilisant des moyens de calculs standard. Quelques études complémentaires sont aussi présentées comme la parallélisation temporelle, des techniques de parallélisation à mémoire partagée et des outils de caractérisation mathématique des schémas de type séparation d'opérateur.

Mots-clés. *Problèmes multi-échelles; Réaction-diffusion-convection; Séparation d'opérateur; Multirésolution adaptative; Intégration temporelle adaptative; Contrôle d'erreur; Parallélisation à mémoire partagée; Algorithm Pararéel; Ondes chimiques non linéaires; Accidents vasculaires cérébraux; Flammes laminaires; Décharges plasma.*

Abstract

We tackle the development of a new generation of numerical methods for the solution of time dependent PDEs modeling general time/space multi-scale phenomena issued from various application fields. This type of problem induces well-known numerical restrictions and potentially large stiffness, which stem from the broad spectrum of time scales in the nonlinear chemical terms as well as from steep, spatially very localized, spatial gradients in the reaction fronts. Therefore, dedicated numerical strategies are needed to ensure the accuracy of the numerical approximations from a theoretical point of view, taking also into account adequate practical implementations to reduce computational costs. In order to cope with these problems, this study introduces a few mathematical and numerical elements for the solution of stiff reaction-diffusion systems, extensible in practice to more general configurations. The core of the numerical strategy is thus based on a specially conceived operator splitting method with dedicated high order time integration schemes for each subproblem. An appropriate choice of splitting time steps allows us the simulation of the solution within a prescribed accuracy, according to the overall physics of the problem. The resulting numerical scheme is properly coupled with an adaptive multiresolution technique for dynamic spatial mesh representations of the solution. Such an approach has led to the conception of the academic, generic 1D/2D/3D MBARETE code in order to evaluate the proposed theoretical and numerical developments in practical stiff configurations arising in several research fields. The algorithmic efficiency of the method is assessed by the simulation of propagating stiff reaction waves issued from nonlinear chemical dynamics and from biomedical engineering applications for a brain stroke model with "detailed chemical mechanisms". Moreover, in order to extend the applicability of the method to more complex and unsteady problems, we consider for the first time a time adaptive splitting scheme for stiff PDEs, that yields dynamic time stepping within the prescribed accuracy. The fully time/space adaptive method allows us then a consistent description of reaction-diffusion-convection problems disclosing a broad spectrum of time/space scales as well as different physical scenarios, such as highly nanosecond repetitively pulsed discharges or self-ignition and propagation of flames for, respectively, plasma and combustion applications. The main goal of this work is hence to numerically solve stiff PDEs with reasonable, standard computational resources and based on a mathematical background that ensures robust, general and accurate numerical schemes. Further studies are also presented that include time parallelization strategies, parallel computing techniques for shared memory architectures and complementary mathematical characterization of splitting schemes.

Keywords. *Multi-scale problems; Reaction-diffusion-convection; Time operator splitting; Adaptive multiresolution; Time adaptive integration; Error control; Shared-memory parallel computing; Parareal algorithm; Nonlinear chemical waves; Brain stroke simulation; Laminar flames; Plasma discharges.*

Contents

List of Figures

List of Tables

General Introduction

A large variety of physical phenomena is characterized by a highly multi-scale nature given by the interaction of several sub-processes evolving simultaneously at different time rates and space scales. The study and physical comprehension of the behavior of these mechanisms give rise to several models with a degree of complexity directly related to the desired level of detail of the description. These models result often from extensive research over various scientific domains ranging from fundamental theoretical physics to experimental observations and measurements. This is actually the case for different applications such as combustion [Pet09, Ech09], chemical vapor deposition [KDK+07, Mar09], nanomaterials [KC07], air pollution modeling [TGHB00, DGOZ04], or plasma discharges [EMB+06, EBD+11], in which multi-scale phenomena represent a common feature. In this context, the numerical simulations of these problems through the solution of the respective mathematical models, constitute an important tool for a better understanding of these phenomena for both industrial and scientific purposes. Nevertheless, the resulting models that aim at mimicking the original problems, raise several difficulties this time in terms of the computational effort needed to properly handle a high number of physical variables and parameters, as well as an often broad range of temporal and spatial scales due, for instance, to large and detailed chemical kinetic mechanisms and usually inhomogeneous spatial distributions[1].

In principle, the comprehensive numerical simulation of detailed mechanisms that result from a fine modeling, will certainly reproduce more accurately many physical patterns disclosed by real and complex phenomena. Such an approach is usually known as the direct numerical simulation (DNS) of a model, where the solution of the governing equations is carried out with a sufficiently fine resolution such that all temporal and spatial scales are described. Nevertheless, such a detailed numerical description is necessarily related to important computational resources, depending on the extent of the numerical simulation needed to represent the leading features of a particular application. One approach to successfully overcome potential computational restrictions and carry out very accurate numerical simulations, is thus founded on the development of performing techniques to take full advantage of parallel and massively parallel computing architectures, taking into account the strong growth of the computer power in the very last decades. In this way, the research effort is focused on the efficient exploitation of the computational resources to accomplish numerical simulations of problems with an increas-

[1] A classical example of multi-scale problem is given by the description of the dynamics of chemical waves related to nonlinear chemical reaction processes, in which the reaction rates are several times faster than the speed of propagation of the wavefronts, also very localized in space. For instance, as we will illustrate in the following chapters, the well-known Belousov-Zhabotinski reaction that models excitable media (see, *e.g.*, [EP98]), generates propagating waves with a speed of approximatively 0.6 with reaction rates for the fastest variable of the order of 10^{-5}, whereas the thickness of the moving fronts corresponds to less than 1 % of the corresponding dimension of the spatial domain occupied by the reactive medium. In particular, these nonlinear chemical waves are often used to study and evaluate the performance of numerical methods because they mimic multi-scale features of more complex models as the ones previously cited, like combustion fronts or highly nonlinear ionizing waves issued from plasma discharges.

ing degree of detail, whereas the choice of standard and rather simple but robust numerical methods, eases the construction of the numerical implementations[2].
Another alternative which is largely used for industrial and scientific applications, concentrates this time on modeling issues to reduce the complexity associated with a highly detailed description of the problem. The idea seems natural since either not all time and space scales take a leading part in the global phenomenon of a particular process, or some features of the latter can be neglected or simplified under particular physical circumstances or consistent hypotheses. This kind of approach has been extensively used to overcome computing limitations in the very beginning of numerical simulation, but has been also developed ever since to achieve large scale simulations in modern computer architectures with consistent physical models that retain a sufficiently accurate description of the main leading processes of the problem under study, out of the original exhaustive formulation. Hence, the continuous research and important investment in the development of reliable models and alternative formulations, combined with modern computational resources, yield potentially large scale numerical simulations with predictive capabilities of real complex phenomena, at often much lower expenses than direct numerical solution of comprehensive models[3].
In general for the referred applications, the resulting governing equations coming from either a fully or partially detailed, or simplified or reduced model formulation, inherit some or most of the behavioral patterns associated with the general physics of the problem, that need to be correctly represented by the numerical solutions. As a consequence, a main concern in the domain of applied mathematics is given by the construction and development of efficient numerical methods for the solution of these modeling equations. In particular, a large variety of modeling configurations and their related particularities, impose often an equivalently large number of dedicated numerical methods and successive ramifications. The effectiveness and performance of a given method can be then assessed by

- Its capabilities in terms of accuracy and thus of supplying consistently good numerical solutions. This issue is directly related to the mathematical theoretical background upon which the numerical methods are conceived, and also to numerical analysis elements which provide an evaluation of their numerical behavior under the particular conditions imposed by the description of a given process; and

- Its capabilities in terms of practical implementation and thus of the degree of algorithmic complexity and computational requirements. This second aspect accounts rather for the effective realization of the methods and the related expenses in terms of computational resources, at the interface of applied mathematics and scientific computing.

These two points stand for the theoretical and practical features of a given numerical method developed for some specific application, and generally speaking one may consider

- Solid and highly dedicated numerical schemes from a theoretical point of view but with also more complex or expensive numerical implementations; or

- More basic but still robust methods that yield an easier coupling with highly optimized techniques in terms of computer representation and execution for single or multi-processor configurations.

[2] See, *e.g.*, in [CCdS+09, Che11] some numerical developments and accomplishments of a high performance DNS code for combustion applications, and [SJWO11] for the use of Graphics Processing Units (GPUs) to accelerate the evaluation of detailed chemical kinetics.

[3] Look, *e.g.*, in [SGP07, SSGP08] for some reviews and achievements on massively parallel computations for combustion problems in industrial configurations, with an approach called Large Eddy Simulations (LES) [Pit06, Vey09] based on spatial filtering of the detailed formulation to focus on the contribution of rather large scales.

Both a compromise and consistency between theoretical and practical aspects should be then sought in practice, to achieve at the same time affordable and reasonably accurate simulations by means of numerical methods that aim simultaneously at computational efficiency with a robust mathematical basis that ensures accuracy of computations. It is quite clear that an interaction and combination of different elements coming from basic mathematical findings up to sophisticated code structures or programming techniques, are thus necessary in the global conception of these numerical schemes. These are some of the main concerns regarding the numerical methods, that along with the ones previously mentioned on physical modeling and exploitation of the computer power, constitute part of the fundamental background of the vast multi-disciplinary scientific research for numerical simulations.
In this general context, this work is rather inscribed in the conception and development of numerical strategies for the numerical simulation of multi-scale phenomena. In particular, our attention is focused on the numerical solution of problems mathematically modeled by time dependent partial differential equations (PDEs), ultimately involving complex and detailed mechanisms. The multi-scale character of these equations is numerically represented by their inherent time stiffness and a wide range of space scales, such that highly dedicated time integration schemes are often needed at least for numerical simulations employing standard computational resources, as well as sufficiently large computational domains such that the main spatial structures of the problem are properly represented. The effort is thus concentrated on the construction of suitable numerical techniques to properly handle these issues, based on a firm mathematical background with the final objective of accuracy control of the numerical simulations, but searching also efficient practical implementations which will pave the way towards more realistic applications that may fully exploit the computational resources and hence the available computer power, to treat a wide range of physical models with different degrees of detail.

In order to numerically simulate multi-scale physical problems modeled by time dependent PDEs, a key aspect is given by the definition of the numerical schemes that will be used to describe the evolution in time of these equations. Several numerical strategies were therefore introduced over the past years for the numerical time integration of the equations, for a given spatial discretization of the problem. Assuming that the latter representation is sufficiently fine to reproduce the space multi-scale features of the physical phenomenon under study, dedicated numerical methods should be considered to properly handle the time scale spectrum of the problem resulting from the various phenomena like reaction, diffusion, convection and other processes included in the model. The numerical description of the time multi-scale features of the problem and the subproblems within, may then consider the numerical solution of the complete and coupled governing equations for which generally speaking, there are two extreme configurations:

- An explicit time discretization of the equations which involves important numerical stability restrictions and consequently very long computations for small time evolution steps, ruled by the fastest physical or numerical time scale arising in the problem.

- A fully implicit treatment that allows us to cope with the numerical stiffness associated with the modeling equations, with time steps settled rather by the desired accuracy of the simulations.

Although explicit schemes are often very simple to implement and require minimal computational resources for non stiff problems, they become completely inefficient for the time integration of stiff equations and their use will be justified only if other techniques like the extensive

use of computer power are considered to reduce the computational costs, as we have previously mentioned. Nevertheless, one way to reduce these computational requirements by means of numerical techniques is based on adapted mesh representations, introduced and widely developed over the last decades. The computing effort is thus concentrated on some regions of fine spatial representation within the entire computational domain, where the phenomena under study require a more precise description[4]. By this kind of procedure, important memory space can be saved and large computational domains can be thus simulated with reduced computational resources. Although such a procedure is quite natural, new challenges are represented by the development of efficient computational implementations, considerably more complex than for standard meshing techniques (see, *e.g.,* [BMV09]). Other critical issues are given by the definition of appropriate criteria for local refinement and by a reliable evaluation of the accuracy of computations, considering the approximation errors introduced by the compressed spatial representations. These techniques are easily coupled with standard explicit time integration, even if very fine discretizations imply higher stability constraints. Special techniques were therefore developed to overcome the latter restriction, usually based on a local time stepping approach that takes into account the local grid size with adequate synchronization stages [BC89], and coupled, for instance, with hybrid implicit/explicit configurations (see, *e.g.,* [CNPT08, CNPT10]) or with time operator splitting (see, *e.g.,* [DB00, BDA+06]).
Another alternative to favor the use of less expensive explicit schemes considers this time, the reduction of the stiffness of the equations relying on a dynamic analysis of the time scales involved in the problem. These techniques are mainly based on the automatic identification of slow and fast variables in a general stiff system, and the consequent numerical solution of the problem on a reduced lower dimensional equilibrium manifold characterized by the slow time scales of the system, in which the fastest variables are assumed to be in a steady state[5]. Since broad time scale spectra are usually associated with detailed complex chemical kinetics and hence with the source term in the PDEs, different methods were developed to perform these time scale and variable reductions by numerically evaluating the Jacobians related to the resulting dynamical systems, *i.e.,* the original stiff systems of ordinary differential equations (ODEs) coming from the modeling PDEs in the absence of transport phenomena. A large literature has been devoted to this subject and consequently, there has been a tremendous effort in defining efficient and predictive numerical methods in order to reduce and to solve in a less expensive manner complex systems of equations arising for example in combustion or air pollution modeling applications, among others[6]. See, *e.g.,* [LL09, PR09] for recent reviews on these issues. Nevertheless, a critical aspect from a theoretical point of view (see, *e.g.,* [Mas02]) and a delicate task in practice, is to conceive numerical techniques that allow us to measure the accuracy of the approximate solutions issued from the reduced dynamics with respect to the corresponding original equations.
Alternatively, the numerical solution of the governing equations with a fully implicit time discretization allows us to properly handle the entire time scale spectrum of the problem and furthermore, ensures accurate numerical approximations in terms of physical couplings of the various underlying processes within the global phenomenon. Taking into consideration the strong development of dedicated time integration solvers for stiff ODEs systems over

[4] We will further detail in this work a few key aspects of classical AMR (Adaptive Mesh Refinement) methods introduced by Berger & Oliger [Ber82, BO84], as well as adaptive multiresolution techniques based mainly on the work of Harten [Har94a, Har95].

[5] In particular, the derivation of reduced models is usually straightforward when the modeling system is well-partitioned and the fast scales are easily isolated [Spo99, SD00].

[6] Some classical examples of such techniques that have been further developed, are given by the ILDM (Intrinsic Low-Dimensional Manifold) approach of Pope and Maas [MP92] and the CSP (Computational Singular Perturbation) method of Lam & Goussis [LG94].

the last decades, the implementation of these solvers showed to deliver satisfactory results in many numerical strategies for stiff detailed formulations. Historically, the first, extensively used implicit solvers for stiff ODEs were based on variable-order (up to fifth) multi-step BDF formulae [Gea71] like LSODE [Hin80, Hin83], or VODE [BBH89], among others. Other dedicated schemes were also introduced based this time on one-step Runge-Kutta schemes like diagonally or fully implicit methods (see, *e.g.,* [HW96] and references therein), or even explicit Runge-Kutta schemes with extended stability regions (see, *e.g.,* [Ver96, HV03]). Furthermore, all these solvers consider carefully conceived implementations with adaptive time-stepping tools and hence, the chosen time steps ensure a prescribed accuracy tolerance that enhance the numerical performance of the methods. The use of these solvers was often extended to PDEs by considering ODEs systems issued from a standard semi-discretization with the method of lines. Nevertheless, all these implicit schemes require important memory resources which restrict their direct implementation for large computational domains usually required for PDEs modeling multi-scale phenomena, whereas stabilized explicit schemes that are well suited and very performing for parabolic problems, fail to properly handle very stiff problems or operators that arise in convection dominated flows.

Other dedicated and even high order implicit schemes have been then conceived over the past years for the numerical integration of time dependent PDEs, based on either one-step Runge-Kutta or multi-step schemes (see, *e.g.,* [BCVK02, IZ04, CKB+05]) for the spatially discretized equations. The level of accuracy of the simulation can be directly related to the value of the integration time step, as we have previously noted. One of the key points for the success of implicit techniques is nevertheless related to the development of efficient solvers, usually based on Newton iterative methods (see, *e.g.,* [Deu04]), for the solution of large, strongly coupled and highly nonlinear systems, taking into account the important computational requirements in memory and CPU time to perform these computations[7]. In the same way, special high order space discretization schemes have also been implemented which reduce the computational stencils and hence the size of the nonlinear systems, while also contributing to better capture particular physical features (see, *e.g.,* [NS05, NBS07, DS10]). Other critical concerns in practical implementations are given by the development of complementary techniques to ensure the efficiency and feasibility of computations such as preconditioning and correct initialization of the Newton solvers, proper distributed meshing, data storage and representation, among others[8]. Taking into account the computational costs of detailed numerical simulations with these highly accurate but also expensive schemes, the conception of efficient parallelization techniques that exploit the current computer power offer another way out to perform these computations (see, *e.g.,* [TBS11] for recent investigations in the field), as well as adaptive meshing tools (see, *e.g.,* [ABS05]). All these topics constitute an important part of current research that will eventually yield accurate and fully detailed 3D simulations of complex multi-scale phenomena, not yet feasible in our days.

An intermediate numerical strategy may nevertheless exploit the advantages of dedicated implicit methods, combined with less demanding explicit schemes in a hybrid implicit/explicit time discretization structure for a spatially discretized stiff problem. In this way, one may consider many types of the so-called partitioning or IMEX methods, by associating implicit/explicit

[7] There is always an important research on these matters to achieve more performing solvers of nonlinear systems at much less computational expenses. See, *e.g.,* [KK04, ZC09] for some reviews on Newton-Krylov methods, recently considered in [DS10] for detailed chemistry simulations.

[8] See, *e.g.,* [DCS+07, BMP+09] and references therein for some efficient implementations of implicit solvers for the numerical simulations of flames with detailed kinetics, furthermore compared with experimental measurements, based on a numerical technique originally introduced by Smooke in [Smo83]. Some further developments with parallel computations on distributed memory machines can be found in [DCB+09].

schemes equation-wise to stiff/non stiff variables of the PDEs, or term-wise to different phenomena like reaction, diffusion or convection, or alternatively by an intelligent combination of both alternatives. One of the first IMEX schemes introduced by Crouzeix in [Cro80] considered multi-step time discretizations where the implicit and explicit parts corresponded to the linear and nonlinear terms in the equations, based on the efficiency of implicit solvers for linear problems (see, *e.g.,* [ACM98]). These ideas motivated later on the conception of either multi-step or one-step IMEX methods, this time specifically for the numerical solution of stiff PDEs[9]. Generally speaking the main idea is that the detailed stiff dynamics associated mainly with the reaction term is solved by an implicit method, whereas the remaining non stiff phenomena usually related to spatial transport operators like convection or diffusion are solved by a standard or possibly dedicated explicit method (a detailed review can be found in [HV03]). Alternatively, for well-partitioned systems for which fast and slow variables can be easily identified, Hofer introduced in [Hof76] another numerical alternative that considers an implicit or explicit treatment, respectively, for each subsystem (see some recent developments in [CD11][10] and references therein for ODEs, and [WR08] for PDEs), whereas the same approach can be also successfully implemented for more general PDEs after a dynamic decoupling technique to identify the various time scales of the problem (see, *e.g.,* [HW06][11]), as we have previously discussed for the explicit treatment of reduced systems[12]. Even though an IMEX approach allows us in general important performance enhancements, it supposes also a few coupling considerations among the different evolving subproblems as well as combined stability and order conditions for all inner implicit/explicit schemes[13]. Otherwise, a direct implementation without considering the latter mathematical issues would not be sufficient to define and settle the accuracy of the numerical computations, specially for stiff PDEs, while it will just reduce the numerical constraints imposed by the stiff terms down to the next consecutive most demanding subproblem in terms of stability or accuracy[14]. In practice, the time steps globally imposed over partial regions or over the entire computational domain are often limited by either the stability restrictions of the explicit solver or by the required accuracy of the implicit scheme to solve the fastest dynamics in the problem. Furthermore, another delicate issue deals with the feasibility of considering implicit solvers over a discretized domain for large computational

[9] Some classical works on multi-step or Runge-Kutta IMEX methods were introduced by Ascher *et al.*, respectively, in [ARW95] and in [ARS97]. In particular, the stability of linear multi-step IMEX methods was analyzed by Frank *et al.* in [FHV97], which proved that stable implicit and explicit schemes do not necessarily yield stable IMEX methods.

[10] In particular, these authors studied a combination of IMEX and multirate techniques that consider variable-dependent time steps with error control, and analyzed the corresponding stability domain of the method.

[11] These authors define the slow and fast variables based on the stability domain of the explicit scheme in order to ensure the stability of computations for a time step computed based on accuracy criteria.

[12] We remark than in the previous case only the resulting slow variables were actually solved in time after the dynamic reduction of the general PDEs, even though the fast scales might be reconstructed afterwards if needed. See some illustrations in [DS02b, DS03] for air pollution modeling applications, and in [VG01] for combustion problems with detailed chemical kinetics.

[13] See, *e.g.,* [KC03] and references therein for different IMEX implementations, and in particular for the theoretical issues related to the construction of general high order time integration IMEX schemes for reaction-diffusion-convection systems. This kind of study can be also found, for instance, in [LZ06] for stiff ODEs systems or in [HR07] for dedicated high order schemes that feature adequate properties to handle steep gradients or shock-like solutions for hyperbolic problems including stiff source terms. Strong stability properties like A-stability were studied, for instance, in [Gje07] for second order multi-step IMEX schemes.

[14] An illustrating example was presented and discussed in [NWK98] for the numerical simulation of flames with detailed kinetics, in which a dedicated implicit solver succeeded to overcome highly constraining stability conditions imposed by the reaction terms, but the global time steps remained restricted to ensure the numerical stability of the explicit scheme used for the diffusion problem, quite limited given the fine spatial discretization needed for these problems.

domains with a high number of unknowns[15].
A further step in the previous direction of hybrid configurations built upon different schemes, is given by time operator splitting methods [Mar68, Str68, Mar75, Mar90], also called fractional steps methods [Tém69a, Tém69b, Yan71]. These schemes consider also dedicated implicit/explicit treatment for the numerical solution of the different subproblems included in the modeling equations, but this time the latter subsystems are completely decoupled and solved independently during a prescribed splitting time step. Alternatively, dimensional splitting schemes consider the independent numerical solution of 1D subproblems for multi-dimensional configurations (see, *e.g.*, [HV03] for several term-wise or dimensional splitting techniques). The main advantage of these methods is given by an important reduction of the numerical complexity since they do not require any coupling whatsoever at the intermediate stages of the various schemes and thus dedicated numerical schemes of even completely different nature can be easily implemented for each subproblem. Nevertheless, a critical aspect for these methods is the appropriate definition of decoupling time steps such that the numerical solutions reproduce with sufficient accuracy the coupled dynamics[16]. Several studies performed a numerical evaluation of splitting techniques for PDEs and confronted them to fully implicit schemes or more generally, to numerical solutions that take simultaneously into account the coupled governing equations (see, *e.g.*, [SBP00, RSO04]). These and other studies showed in particular that the resulting splitting errors might become important if large splitting time steps are considered for the numerical solution of stiff PDEs. Moreover, the standard numerical description based on asymptotic analysis (see, *e.g.*, [HLW06]) failed to explain these behaviors and motivated new and more specific theoretical studies on these schemes for stiff problems (see, *e.g.*, [DM04, DDLM07]). A standard solution in practical implementations is thus to consider splitting time steps of the order of the fastest physical or numerical scales, or just sufficiently short to ensure negligible splitting errors, and hence hopefully accurate results for complex problems, often unfeasible with fully coupled implicit numerical strategies (see, *e.g.*, [BDS+05, BDG+07]). Time operator splitting techniques and a few theoretical and numerical related issues will be thoroughly discussed in this work.

Based on the previous discussions we can summarize some key issues to be considered during the conception of numerical strategies to efficiently carry out numerical simulations of multiscale phenomena. Generally speaking, this efficiency might be measured in terms of numerical accuracy and thus related to the mathematical background of the methods, and of their practical realization according to the resulting computational complexity and required resources. In this context, the following observations can be drawn forth:

- Standard explicit time integrations would not be performing in terms of CPU time because of the stiffness disclosed by the equations. Nevertheless, they offer straightforward implementations with reasonable memory requirements, easily coupled with other numerical techniques like adaptive mesh refinement or chemical reduction, as well as highly

[15] These are standards limitation for general IMEX schemes as seen, for instance, in [KC03] or [CD11]. Nevertheless, to ease these limitations, Verwer *et al.* considered in [VS04] an implicit Euler scheme to deal with the stiff ODEs obtained at each spatial node after decoupling the numerical solution of the reaction problems. The latter was achieved by embedding this implicit discretization in a stabilized explicit Runge-Kutta scheme, such that the global stability domain of the method, dictated by the explicit scheme, can be enlarged for the global time step computed within a prescribed accuracy for both the reaction and diffusion problems. This scheme led to the IRKC code [SSV06] for stiff reaction-diffusion problems. Extensions to reaction-diffusion-convection systems were proposed in [VSH04] by adding the corresponding stability conditions for the global time steps.

[16] The resulting splitting errors can be mathematically characterized, for instance, for general reaction-diffusion-convection problems, as described by Lanser & Verwer in [LV99].

performing parallel computing tools. The accuracy of the simulations is often guaranteed by the reduced time steps issued from very constraining stability restrictions.

- Fully implicit schemes overcome important stability restrictions for stiff problems and furthermore, ensure accurate numerical approximations of the detailed dynamics of the problem. Nevertheless, more sophisticated tools and thus an important increase of the algorithmic complexity are required to build numerical techniques that remain competitive in terms of computational resources, which become naturally much more constraining.

- An important research has been conducted to yield alternative techniques that exploit the advantages of both explicit and implicit schemes. The main goal is to reduce computational resources for the numerical solution of stiff problems by an adequate choice and combination of dedicated numerical methods. Some general examples are given by IMEX or splitting techniques. The key issue is nevertheless related to the accuracy of the numerical simulations mainly related to efficient coupling conditions among the different schemes or with an appropriate choice of splitting time steps for either IMEX or splitting methods, respectively. A solid complementary mathematical basis is hence mandatory.

In all cases, the simultaneous development of sophisticated parallel techniques is certainly desirable to better exploit current computational capacities, whereas the numerical methods should be also sufficiently malleable to properly adapt themselves to the exigencies of progressively more detailed modeling criteria of physical phenomena.

In this framework, this work addresses the development of efficient numerical methods and hence the introduction of dedicated numerical tools for the simulation of stiff reacting fronts, which represent a vast class of multi-scale phenomena. The main goal is to numerically solve the modeling stiff PDEs with reasonable standard computational resources and based on a mathematical background that ensures robust, general and accurate numerical schemes. The following study introduces then a few mathematical and numerical elements for the numerical solution of stiff reaction-diffusion systems, extensible in practice to more general configurations. The developed numerical strategy is mainly based on a specially conceived operator splitting method that exploits recent theoretical studies such that the accuracy of the numerical approximation is set by a splitting time steps which is restricted by neither the fastest scales in the source term nor by stability constraints for the explicit schemes, but only by the physics of the phenomenon. This technique is properly coupled in the current strategy with a multiresolution technique, previously introduced in the literature, to dynamically build space adaptive grids that allow us further reductions on computational requirements with a more precise knowledge of the numerical errors introduced by the compressed data representations. The novelty of the approach is given by an intelligent and careful conjunction of various numerical schemes based on theoretical and practical criteria, such that the numerical simulations of these stiff problems can be performed within a prescribed accuracy. Furthermore, a new time adaptive splitting technique has been conceived, a fundamental and missing element in the field over the past years, to extend the applicability of the method to more complex unsteady phenomena by means of a time/space adaptive method. All these techniques has resulted into an academic numerical code called MBARETE. The resulting algorithmic efficiency and the capabilities of the numerical strategy has been then evaluated through the numerical simulation of models arising from several research domains, and hence a key aspect in this work is that theoretical findings and numerical criteria have been transposed onto practical configurations given by stiff problems with different and numerically tough particularities, often out of reach of standard methods. Moreover, theoretical and practical considerations associated with the addition of

a quite recent numerical technique to enhance the numerical performance of the simulations, such as the parallelization of the time domain, have been also analyzed and tested throughout this study, as well as complementary issues going from mathematical descriptions of splitting techniques to parallel implementations on shared memory structures, from a more practical point of view. In this way, this work considers and combines theoretical aspects of numerical analysis and applied mathematics with practical implementation issues of scientific computing for current scientific applications within an academic framework, in order to settle a solid basis for more detailed and complex numerical simulations.

This study is organized in four parts. Part I introduces various crucial aspects briefly stated in this introduction concerning the numerical solution of multi-scale PDEs, and settles the theoretical background of the numerical methods considered throughout this work. On these bases all the new theoretical and numerical tools developed in this study are gathered in Part II for the numerical solution of stiff reaction-diffusion systems. Numerical simulations of problems coming from the field of nonlinear chemical dynamics are included throughout this part as numerical illustrations of the methods. Part III accounts for the algorithmic description of the numerical strategy and some other practical implementation issues in the academic MBARETE code, whereas further analyses, evaluations and extensions of the numerical strategy are conducted in the last Part IV for different applications in several domains, namely biomedical engineering, combustion and plasma applications.

Part I is composed of three chapters. Chapter 1 is about time operator splitting techniques to numerically integrate stiff PDEs. In the first part, we present briefly the main numerical context of these methods to better characterize their advantages and limitations, and we consider the construction of general time operator splitting methods of first and second order as well as their classical mathematical description. In the second part, we establish a more precise mathematical analysis of such techniques in the context of time/space stiff PDEs, illustrated by stiff reaction-diffusion systems. In particular, we recall some of the main results in the literature regarding these issues.
Chapter 2 deals with the time integration of stiff ODEs by one-step Runge-Kutta methods. We thus detail the main aspects regarding Runge-Kutta schemes and give some insights into implicit and stabilized explicit Runge-Kutta methods, two families of dedicated time integration methods extensively used in the literature. Particular attention is given to Radau5, an implicit Runge-Kutta solver developed by Hairer & Wanner [HW96], and ROCK4, a stabilized Runge-Kutta scheme introduced by Abdulle [Abd02]. This stiff characterization of ODEs and their numerical treatment allows us to complete the numerical characterization of PDEs depicted in the previous chapter.
Finally, Chapter 3 considers space adaptive multiresolution techniques for dynamic adaptive meshing for time dependent PDEs. A short introduction on general adaptive mesh refinement is first presented based on the pioneering works of Berger & Oliger [Ber82, BO84] for AMR techniques, to then consider multiresolution techniques for PDEs introduced by Harten [Har94a, Har95]. The next part is thus devoted to the mathematical background of wavelet decomposition on which these multiresolution schemes are based, to end with the description of fully adaptive multiresolution techniques introduced by Cohen *et al.* [CKMP03], for the spatial representation of time evolutionary PDEs.

There are three chapters in Part II. Chapter 4 develops a new numerical strategy for the solution of multi-scale propagating waves modeled by reaction-diffusion systems. A time operator splitting approach is introduced that considers the high order methods Radau5 [HW96] and

ROCK4 [Abd02], to solve the reaction and diffusion problems and to handle the numerical difficulties associated with each one of them, in a separate manner. The global accuracy of the time integration scheme is thus set by the splitting scheme by means of an appropriate choice of the splitting time step dictated by the global physical coupling, possibly much larger than the fastest time scales, for multi-scale propagating waves. The numerical strategy is then complemented by a space adaptive multiresolution technique, that allows us to better control the accuracy of the adapted and compressed spatial representation. In this way, both space and time errors introduced by the numerical methods can be regulated for a given semi-discretized problem. This study has been published in SIAM Journal on Scientific Computing [DMD+12], and has been presented during the CLEI 36th Latin American Informatics Conference, Asunción, Paraguay (2010) [DML+11].
Chapter 5 is devoted to time adaptive splitting schemes for PDEs. It introduces thus a new adaptive splitting technique for stiff PDEs which allows us to dynamically compute the splitting time steps of integration within a prescribed accuracy. This method is based on local error estimates computed through the incorporation of a lower order embedded splitting scheme. The numerical analysis of this technique is provided in a general nonlinear context as well as for self-similar propagating waves. A complementary theoretical and numerical study on non-asymptotic regimes allows us to extend the applicability of the method to more realistic situations for large splitting time steps for which the asymptotic theoretical estimates might fail. In particular, this adaptive method allows us to recast the previous numerical strategy of Chapter 4 for the solution of more general and highly unsteady phenomena, and yields thus a time-space adaptive numerical method with dynamic error control. This study has been published in Confluentes Mathematici [DDD+11].
Chapter 6 is a theoretical and numerical study on time operator splitting techniques combined with a time parallelization method for the solution of stiff reaction waves. The main goal is to exploit parallel computations and an efficient splitting technique to reduce numerical costs. The well known *parareal* algorithm proposed by Lions *et al.* in [LMT01] is considered, and a numerical analysis of these techniques is conducted in the context of stiff PDEs characterized by solutions with high spatial gradients. Numerical illustrations allow us to validate the theoretical estimates and to conduct a numerical evaluation of the performance of the method. This study was recently published in ESAIM: Mathematical Modelling and Numerical Analysis journal [DDM11], and it is entirely reproduced in this chapter as a self-contained part of this work.

Part III contains two chapters. Chapter 7 focuses on the description of the adaptive multiresolution technique in the MBARETE code, with the particularities defined in Chapter 4 and based on the theoretical framework of Chapter 3. This part includes thus the algorithmic representation of the several multiresolution procedures as well as the global space adaptive scheme. Some practical issues concerning data representation and code structure are also discussed. The multiresolution technique implemented in the MBARETE code is largely based on of the multiresolution kernel of MR CHORUS[17], a code developed by Christian Tenaud[18] for compressible Navier-Stokes equations. Further details on this multiresolution implementation are available in a tutorial that have been elaborated for a Summer School of CNRS *GDR Groupe Calcul* on Multiresolution and Adaptive Mesh Refinement Methods, Fréjus, France (2010) [TD11].
Chapter 8 details the splitting techniques introduced in the MBARETE code, based on the numerical strategy introduced in chapters 4 and 5. The data structure associated with the

[17] Déclaration d'Invention DI 03760-01.
[18] LIMSI - CNRS, B.P. 133, Campus d'Orsay, 91403 Orsay Cedex, France (`tenaud@limsi.fr`).

PDEs time integration is first presented, as well as the coupling with the previous multiresolution structure. The algorithms considered for the reaction and diffusion problems are then described as well as the splitting method with both constant and adaptive splitting time steps. The complementary numerical procedure introduced in Chapter 5 to correct the computed splitting time steps is also detailed.

The last Part IV includes three chapters. Chapter 9 considers the numerical simulation of human brain stroke modeled by a stiff reaction-diffusion system. It is divided into two parts. In the first part we present an article submitted for publication [DDD+12], which constitutes a complete study on the numerical simulations of such problems. Numerical simulations on complex brain geometry are thus presented, that were performed with a code called ZEBRE [Dum07], developed by Thierry Dumont[19]. This solver considers the same time operator splitting technique for stiff reaction-diffusion systems detailed in Chapter 4, with uniform grid discretizations. A cross partial validation is also conducted by performing numerical comparisons between the results of both the ZEBRE and MBARETE codes for a simplified brain geometry. In this study, the problem is modeled by a set of 19 variables describing detailed chemical mechanism in the source term. A parallel implementation of both codes is carried out for shared memory architectures. This work counts on the special collaboration of Marie-Aimée Dronne[20]. In the second part, we conduct a detailed numerical evaluation of the results obtained with the MBARETE code for simplified 2D and 3D geometries. In this way, we extend the numerical applicability of the numerical strategy of Chapter 4 to more complex models with parallel computing tools. The latter study has been also presented during the Summer School of CNRS *GDR Groupe Calcul* on Multiresolution and Adaptive Mesh Refinement Methods, Fréjus, France (2010) [DMD+11b].
Chapter 10 deals with the numerical simulation of combustion fronts. This study is performed in the context of laminar flames interacting with vortex fields. A thermo-diffusive approach is considered to decouple hydrodynamics from the transport equations and to solve only the latter set of equations with the numerical schemes developed in chapters 4 and 5. The introduction of the convection problem is taken into account for these reaction-diffusion-convection models, and the time evolution of the convective operator is performed by the OSMP scheme developed by Daru & Tenaud in [DT04]. Two kind of configuration are studied given by the propagation of premixed flames and the self-ignition of reactive mixtures. The capabilities and performance of the method are assessed for both situations featuring different physical behaviors. Preliminary results of numerical tests with a complex chemistry formulation are also reported. This work counts on the special collaboration of Christian Tenaud and Sébastien Candel[21]. Some of these results were presented during the Finite Volumes for Complex Applications VI International Symposium, Prague, Czech Republic (2011) [DMDD11], and published in the Annual Research Briefs 2011 of the Center for Turbulence Research, Stanford University, USA [DMD+11a].
Chapter 11 considers the numerical simulations of positive streamers, in the context of propagation of highly nonlinear ionizing waves originated from plasma discharges, as well as highly multi-scale nanosecond repetitively pulsed discharges. In the first part, the time adaptive scheme introduced in Chapter 5 is evaluated for a simplified reaction-diffusion model. Sec-

[19] Institut Camille Jordan - UMR CNRS 5208, Université de Lyon, Université Lyon 1, INSA de Lyon 69621, Ecole Centrale de Lyon, 43 Boulevard du 11 novembre 1918, 69622 Villeurbanne Cedex, France (tdumont@math.univ-lyon1.fr).

[20] Université de Lyon, Université Lyon 1, ISPB - Faculté de Pharmacie de Lyon 69003 Lyon, France (marie-aimee.dronne@recherche.univ-lyon1.fr).

[21] Laboratoire EM2C - UPR CNRS 288, Ecole Centrale Paris, Grande Voie des Vignes, 92295 Chatenay-Malabry Cedex, France (candel@ecp.fr).

ondly, a new numerical strategy was developed for plasma models given by a set of drift-diffusion equations coupled with the electric field computation. The resulting second order scheme features also dynamic time adaptation within a prescribed accuracy, whereas the same strategy presented in Chapter 5 and extended in Chapter 10 for reaction-diffusion-convection systems is implemented for the solution of the drift-diffusion equations. A detailed numerical evaluation of the numerical strategy is finally conducted for problems including a broad spectrum of space and time scales as well as different physical scenarios. This study has been recently published in Journal of Computational Physics [DBM+12], and has counted on the special collaboration of Zdeněk Bonaventura[22] and Anne Bourdon[23]. This article is entirely reproduced in the second part of the chapter since it constitutes a self-contained study and a further extension to this work.

This Ph.D. was mainly directed by Marc Massot[24] (Main Advisor) and Stéphane Descombes[25], and counted on the close collaboration of Thierry Dumont, Violaine Louvet[26], and Frédérique Laurent[27].

This work was supported by a Ph.D. grant from the Mathematics (INSMI) and Engineering (INSIS) Institutes of CNRS and by INCA project (National Initiative for Advanced Combustion) led by CNRS/ONERA/SAFRAN. Furthermore, it was carried out thanks to:

- *an ANR Blancs project (French National Research Agency): Séchelles (project leader S. Descombes – 2009-2013),*
- *an ANR CIS project: PITAC (project leader Y. Maday – 2006-2010),*
- *a DIGITEO RTRA project: MUSE (project leader M. Massot – 2010-2014),*
- *a CNRS PEPS Maths-ST2I project: MIPAC (project leader V. Louvet – 2009-2010),*
- *a CNRS PEPS (project leaders A. Bourdon & F. Laurent – 2007-2008),*
- *a France-Stanford project (project leaders P. Moin & M. Massot – 2011-2012).*

[22]Department of Physical Electronics, Faculty of Science, Masaryk University, Kotlářská 2, 611 37 Brno, Czech Republic (zbona@physics.muni.cz).

[23]Laboratoire EM2C - UPR CNRS 288, Ecole Centrale Paris, Grande Voie des Vignes, 92295 Chatenay-Malabry Cedex, France (anne.bourdon@ecp.fr).

[24]Laboratoire EM2C - UPR CNRS 288, Ecole Centrale Paris, Grande Voie des Vignes, 92295 Chatenay-Malabry Cedex, France (marc.massot@ecp.fr).

[25]Laboratoire J. A. Dieudonné - UMR CNRS 6621, Université de Nice - Sophia Antipolis, Parc Valrose, 06108 Nice Cedex 02, France (sdescomb@unice.fr).

[26]Institut Camille Jordan - UMR CNRS 5208, Université de Lyon, Université Lyon 1, INSA de Lyon 69621, Ecole Centrale de Lyon, 43 Boulevard du 11 novembre 1918, 69622 Villeurbanne Cedex, France (louvet@math.univ-lyon1.fr).

[27]Laboratoire EM2C - UPR CNRS 288, Ecole Centrale Paris, Grande Voie des Vignes, 92295 Chatenay-Malabry Cedex, France (frederique.laurent@ecp.fr).

Part I

Numerical Solution of Multi-Scale Evolutionary PDEs

Introduction

The numerical solution of time dependent, stiff PDEs is a delicate task. The causes of the numerical difficulties usually range from the multiple time scales related to different phenomena within the governing equations, to computational domains of important size to achieve an appropriate physical resolution of the problem. It is quite clear that the numerical strategies must simultaneously consider many theoretical and practical aspects to yield mathematically solid schemes at reasonable computational expenses. The objective of this part is to introduce some numerical methods in the current literature that were developed to cope with particular difficulties in the solution of stiff PDEs, and to settle the mathematical background of this work.

Chapter 1 refers to time operator splitting techniques to numerically integrate time dependent PDEs. The review on these schemes is not exhaustive but aims at giving sufficient information on the theoretical characterization of splitting methods and some important issues often encountered in the numerical solution of stiff problems. The reader may refer to the book of Hundsdorfer & Verwer [HV03] for further details on different types of splitting technique.

Chapter 2 deals with the time integration of stiff ODEs by one-step Runge-Kutta schemes. This description complements the previous chapter and gives a more detailed insight into the numerical solution of stiff problems. In particular, we focus on Runge-Kutta methods given by implicit and stabilized explicit techniques. A complete information can be found in the book of Hairer & Wanner [HW96].

Finally, Chapter 3 introduces some adaptive mesh refinement techniques for time dependent PDEs by means of space adaptive multiresolution. Most of the chapter is thus given by the mathematical description of wavelet decomposition and multiresolution techniques for grid adaptation. The books of Cohen [Coh00] and Müller [Mül03] constitute very good references in the domain.

Chapter 1

Time Operator Splitting for Multi-Scale Evolutionary PDEs

In this work we are concerned with the numerical solution of time dependent PDEs involving reactive terms and transport operators like diffusion, convection or both, issued from the mathematical modeling of general multi-scale phenomena. As discussed in the General Introduction, this kind of problem is rather common in many applications so that efficient solution schemes are of the utmost importance. In this chapter, our attention will be focused on the so-called time operator splitting methods for the numerical solution of such problems. A time operator splitting procedure allows us to consider dedicated solvers for the reaction part which is numerically decoupled from the other physical phenomena like convection, diffusion or others, for which there also exist dedicated numerical methods. A completely independent optimization of the solution of each subsystem might be hence pursued in practice. These methods have been used for a long time and there exists a large literature showing their efficiency for time dependent problems, as we will briefly detail in the following. We will then describe the general configuration of such methods and the classical first and second order, Lie and Strang, splitting schemes. A mathematical characterization of the splitting approximation errors will be also provided for both linear and nonlinear operators. In the second part of this chapter, we will introduce some mathematical tools and previous theoretical results concerning the numerical behavior of such methods for the solution of time and space multi-scale PDEs, illustrated in the context of reaction-diffusion systems. All of these descriptions constitute a fundamental part of the theoretical background of this work. A detailed survey and mathematical characterization of different types of splitting method can be found in the book of Hundsdorfer & Verwer [HV03]. Let us remark that throughout this chapter we will describe the numerical solutions issued from splitting techniques and the resulting splitting errors, considering neither time nor space discretization issues in the time integration of the inner subproblems. The latter matters will be discussed in the forthcoming chapters.

1.1 Time Operator Splitting

Operator splitting techniques [Mar68, Str68, Mar75, Mar90], also called fractional steps methods [Tém69a, Tém69b, Yan71], were first introduced in the late sixties with the main objective of reducing computational resources. In this context, a complex and potentially large problem can be split into smaller parts with an important reduction of the algorithmic complexity as well as the computational requirements. The latter characteristics were largely exploited over the past years to carry out numerical simulations in several domains, going, for instance,

from electrocardiology simulations [BWZ+02, TK04], to combustion [OB01, SLGS03] or air pollution modeling [OOZ01, Spo07] applications. These methods can be thus considered as a standard approach in numerical applications and continue to be widely used mainly because of their simplicity of implementation and their high degree of liberty in terms of choice of dedicated numerical solvers for the split subproblems. Other advantages of these methods are given by the possibility of time stepping for the various subproblems since each one of them is independently evolved in time. Additionally, the global numerical stability of the splitting scheme is guaranteed as long as each of the inner numerical solvers ensures stability stable, and the mathematical formulation remains valid. In the context of stiff problems, a particular care must be addressed to choose adequate methods that properly handle and damp out fast transients introduced by the splitting procedure in the split subproblems, for instance, in the reaction [VBvLS96, SVdZ+98, VSBH99] or diffusion [RS05a, RS09] terms.
In most applications, first and second order splitting schemes are implemented, for which a general mathematical background is available (see, *e.g.,* [HLW06] for ODEs and [HV03] for PDEs). Even though higher order schemes are theoretically feasible, they are usually not suitable for the solution of PDEs and moreover stiff PDEs [HV03], which constitutes a natural drawback to these schemes. On the other hand, the separate time evolution of each subproblem during a given splitting time step introduces naturally the so-called *splitting errors* into the numerical solutions. In the context of PDEs, Lanser & Verwer conducted in [LV99] a fine analysis on the splitting errors in the solution of reaction-diffusion-convection systems, and defined the particular configurations for which splitting errors arising from the numerical separation of convection, diffusion and reaction subproblems, can be avoided. In a similar way, the latter kind of analysis allowed us to develop splitting techniques for some particular PDEs configurations which resulted in no splitting errors (see, *e.g.,* [HV95] for convection-reaction problems). This type of study gave new insights into the use of splitting techniques for PDE problems and furthermore, complemented the classical theoretical basis.
Nevertheless, for general problems that do not display the particular characteristics defined in [LV99], the splitting errors will likely remain throughout the numerical time integration. On the other hand, it was shown that for more complex problems involving multi-scale features, the classical mathematical characterization based on asymptotic analysis, *i.e.,* sufficiently small time steps, fails often in front of time scales much faster than the considered splitting time step. Actually, the same kind of order reduction that appears in the context of time integration of stiff ODEs (see, *e.g.,* [HLR88, HW96]), arise similarly when considering splitting techniques for stiff problems. For PDEs, this stiffness is usually induced by highly time/space multi-scale features which furthermore are very common in the mentioned applications. All these numerical observations motivated more rigorous studies on the splitting errors, specially for the solution of stiff problems, as we will present in the second part of this chapter.
Another cause of possible order reduction that should be also taken into account, comes this time from the boundary conditions considered in the time integration of PDEs with splitting methods. The key aspect is to define appropriate boundary conditions during the independent time integration of the space transport operators, such that the numerical results and furthermore the boundary values, are consistent with the global coupled problem. Although there has been some numerical studies for particular configurations (see, *e.g.,* [HV95]) that described this kind of problem and suggested numerical procedures to avoid the resulting order reduction, a general theory is still missing and thus remains an open problem in the domain [HV03]. Such issues have not been studied in this work, and we refer to [HV03] for further discussions and illustrations. Let us also remark that for some kind of problem characterized by transient phases yielding convergence towards a steady state solution, a splitting technique will naturally introduce complementary splitting errors with respect to a coupled solution of

the entire problem (see, *e.g.*, [VS04, VSH04]). Although these errors are naturally controlled by the splitting time steps, in some applications for which the correct description of the steady state becomes critical, either a combination of coupled and splitting schemes, or modified splitting techniques might be required to relieve the necessity of using small splitting time steps[1]. These issues were not explicitly analyzed in this work where we have mainly focused on time dependent PDEs modeling unsteady problems. However, these topics represent further and complementary studies in the future.

1.1.1 General Setting

Let us first consider a general linear initial value problem:

$$\left.\begin{aligned} &\mathrm{d}_t U = AU + BU, \quad t > 0, \\ &U(0) = U_0, \end{aligned}\right\} \tag{1.1}$$

with linear operators A, $B \in \mathcal{M}_m(\mathbb{R})$, where $\mathcal{M}_m(\mathbb{R})$ is the set of real square matrices of size m, $U_0 \in \mathbb{R}^m$ and $U : \mathbb{R} \to \mathbb{R}^m$, for which the exact solution is given by

$$U(t) = \mathrm{e}^{t(A+B)} U_0, \qquad t \geq 0. \tag{1.2}$$

A time operator splitting technique consists in successively solving the evolutionary problems associated with each time operator in an independent way. For system (1.1) this amounts to separately solve problems:

$$\mathrm{d}_t U = AU, \quad t > 0, \tag{1.3}$$

and

$$\mathrm{d}_t U = BU, \quad t > 0, \tag{1.4}$$

with appropriate initial conditions for each subproblem. Then, for a time discretization given by $t_0 = 0 < t_1 < \ldots < t_N$, the associated time steps or *splitting time steps* are defined as $\Delta t_n = t_{n+1} - t_n$ for $n = 0, 1, \ldots, N-1$.
Starting from the initial condition of (1.1): $U_0 = U(0)$, the splitting numerical approximation U_{n+1} of the exact values $U(t_{n+1})$ is computed from the previous U_n for $n = 0, 1, \ldots, N-1$, by means of a composition of $s \geq 1$ independent solutions of (1.3) and (1.4) with the recurrence relation:

$$U_{n+1} = \mathrm{e}^{\beta_s \Delta t_n B} \mathrm{e}^{\alpha_s \Delta t_n A} \ldots \mathrm{e}^{\beta_2 \Delta t_n B} \mathrm{e}^{\alpha_2 \Delta t_n A} \mathrm{e}^{\beta_1 \Delta t_n B} \mathrm{e}^{\alpha_1 \Delta t_n A} U_n, \tag{1.5}$$

where $\mathrm{e}^{tA} U_0$ and $\mathrm{e}^{tB} U_0$ are, respectively, the exact solutions of (1.3) and (1.4) for $t \geq 0$ from initial condition U_0. The values of the real or complex coefficients of the scheme: $(\alpha_i, \beta_i)_{i=1}^s$ such that $\sum_i \alpha_i = \sum_i \beta_i = 1$, will then define the order of approximation of the method. These splitting schemes can be seen as composition methods for which the general order conditions are well known (see [HLW06]).

1.1.2 First and Second Order Splitting Schemes

Taking into account the Taylor series expansion of the exact solution $U(\Delta t)$ after time Δt, if the corresponding numerical approximation U_1 is of order p, then the *local error* is given by

$$U(\Delta t) - U_1 = \mathcal{O}(\Delta t^{p+1}). \tag{1.6}$$

[1] As an illustration, a modified splitting approach was conceived in [SLGS03] to compute steady states of reacting flows featuring flames with detailed chemistry.

For system (1.1), the exact solution is given by $U(\Delta t) = \mathrm{e}^{\Delta t(A+B)}U_0$, whereas U_1 is the numerical solution at Δt, both computed from the initial value U_0.
Keeping this in mind for the splitting schemes, we introduce the first order *Lie* (or *Lie-Trotter* [Tro59]) splitting formulae, for which $p = 1$ and

$$s = 1, \quad \alpha_1 = \beta_1 = 1, \tag{1.7}$$

or alternatively,

$$s = 2, \quad \alpha_1 = \beta_2 = 0, \quad \alpha_2 = \beta_1 = 1, \tag{1.8}$$

into (1.5). From a practical point of view and considering problem (1.1), the first scheme (1.7) is performed by first considering the initial value problem:

$$\left.\begin{aligned} \mathrm{d}_t U &= AU, \\ U(0) &= U_0, \end{aligned}\right\} \tag{1.9}$$

during a splitting time step Δt, which yields $U(\Delta t) = \mathrm{e}^{\Delta t A}U_0$. And then, problem:

$$\left.\begin{aligned} \mathrm{d}_t U &= BU, \\ U(0) &= \mathrm{e}^{\Delta t A}U_0, \end{aligned}\right\} \tag{1.10}$$

also during Δt, that yields finally the numerical solution:

$$U_1 = \mathcal{L}_1^{\Delta t}U_0 = \mathrm{e}^{\Delta t B}\mathrm{e}^{\Delta t A}U_0, \tag{1.11}$$

according to (1.5) with coefficients given by (1.7). Alternatively, the second Lie scheme (1.8) considers first problem (1.10), and then (1.9), so that

$$U_1 = \mathcal{L}_2^{\Delta t}U_0 = \mathrm{e}^{\Delta t A}\mathrm{e}^{\Delta t B}U_0. \tag{1.12}$$

Considering both Lie approximations, we can see that one corresponds to the *adjoint method* of the other. That is, $\mathcal{L}_1^{\Delta t}$ (resp., $\mathcal{L}_2^{\Delta t}$) is the inverse map of $\mathcal{L}_2^{\Delta t}$ (resp., $\mathcal{L}_1^{\Delta t}$) with reversed time step Δt:

$$\mathcal{L}_1^{-\Delta t}\mathcal{L}_2^{\Delta t}U_0 = \mathrm{e}^{-\Delta t B}\mathrm{e}^{-\Delta t A}\mathrm{e}^{\Delta t A}\mathrm{e}^{\Delta t B}U_0 = U_0. \tag{1.13}$$

In general it can be shown that composing one-step methods of order p yields a composition method of at least order $p+1$ [HLW06]. In particular, composing with half-sized steps one method of odd order p with its adjoint, yields a symmetric $p+1$ method. In this way, we can obtain a symmetric second order splitting scheme known as the *Strang* (or *Marchuk* [Mar68]) splitting formulae [Str63, Str68] by composing $\mathcal{L}_1^{\Delta t/2}$ (resp., $\mathcal{L}_2^{\Delta t/2}$) with its adjoint method $\mathcal{L}_2^{\Delta t/2}$ (resp., $\mathcal{L}_1^{\Delta t/2}$):

$$\mathcal{S}_1^{\Delta t} = \mathcal{L}_1^{\Delta t/2}\mathcal{L}_2^{\Delta t/2}, \tag{1.14}$$

or alternatively,

$$\mathcal{S}_2^{\Delta t} = \mathcal{L}_2^{\Delta t/2}\mathcal{L}_1^{\Delta t/2}. \tag{1.15}$$

Symmetry is guaranteed because $\mathcal{S}_1^{\Delta t}$ is equal to its adjoint (the same follows for $\mathcal{S}_2^{\Delta t}$), *i.e.*,

$$\mathcal{S}_1^{-\Delta t}\mathcal{S}_1^{\Delta t} = \mathcal{L}_2^{-\Delta t/2}\mathcal{L}_1^{-\Delta t/2}\mathcal{L}_1^{\Delta t/2}\mathcal{L}_2^{\Delta t/2} = \mathrm{Id}. \tag{1.16}$$

Coming back to problem (1.1), we have thus the numerical solutions:

$$U_1 = \mathcal{S}_1^{\Delta t}U_0 = \mathrm{e}^{\Delta t B/2}\mathrm{e}^{\Delta t A}\mathrm{e}^{\Delta t B/2}U_0, \tag{1.17}$$

or

$$U_1 = \mathcal{S}_2^{\Delta t} U_0 = \mathrm{e}^{\Delta t A/2} \mathrm{e}^{\Delta t B} \mathrm{e}^{\Delta t A/2} U_0, \tag{1.18}$$

for which $p = 2$, and, respectively,

$$s = 2, \quad \alpha_1 = 0, \quad \alpha_2 = 1, \quad \beta_1 = \beta_2 = \frac{1}{2}, \tag{1.19}$$

or

$$s = 2, \quad \alpha_1 = \alpha_2 = \frac{1}{2}, \quad \beta_1 = 1, \quad \beta_2 = 0, \tag{1.20}$$

into (1.5).
Higher order splitting schemes are also possible. Nevertheless, the order conditions for such composition methods state that either negative or complex coefficients $(\alpha_i, \beta_i)_{i=1}^s$ in (1.5) are necessary (see, *e.g.*, [HLW06]). Several higher order schemes of this type were already proposed (see, *e.g.*, [Yos90, Des01, MQ02, Sch02, Tha08, CCDV09, HO09, DT10]). The former implies usually important stability restrictions and more sophisticated numerical implementations in terms of algorithmic complexity with respect to less accurate but much simpler first and second order splitting schemes. In the particular case of negative time steps, they are completely undesirable for PDEs that are ill-posed for negative time progression like parabolic equations or very stiff terms issued, for instance, from detailed chemical kinetics [HV03].

1.1.3 Classical Numerical Analysis for Splitting Schemes

In this section, we will introduce some classical mathematical tools used for the numerical analysis of splitting schemes that are going to be used throughout this work. In a first step, we will describe the *Baker-Campbell-Hausdorff* (BCH) formula on composition of exponentials. For the linear operators A and B, for which their exponentials e^{tA} and e^{tB} can be understood as a formal series expansion[2], we define the commutator:

$$[A, B] = AB - BA, \tag{1.21}$$

that we will also denote as[3]

$$\partial_A B = [A, B]. \tag{1.22}$$

The main idea is then to find $C(t)$ such that we can write

$$\mathrm{e}^{tA} \mathrm{e}^{tB} = \mathrm{e}^{C(t)}. \tag{1.23}$$

This exponential representation is known as the BCH formula for which it was demonstrated that $C(t)$ is the solution of the differential equation:

$$\mathrm{d}_t C = A + B + \frac{1}{2}[A - B, C] + \sum_{i \geq 2} \frac{B_i}{i!} \partial_C^i (A + B), \tag{1.24}$$

with initial value $C(0) = 0$ [Var74], where B_i are the *Bernoulli numbers* given by[4]

$$\sum_{i \geq 0} \frac{B_i}{i!} x^i = \frac{x}{\mathrm{e}^x - 1}. \tag{1.25}$$

[2] That is, $\mathrm{e}^{tA} = \left(\sum_{n=0}^{+\infty} \frac{t^n}{n!} A^n \right)$.

[3] Notice that for fixed A, the operator $\partial_A \cdot$ defines also a linear operator $B \mapsto [A, B]$ which is also called the *adjoint operator* [Var74].

[4] See [HLW06] for more details.

Taking into account the series expansions performed in the left-hand side of (1.23), we can infer that for sufficiently small t, $C(t)$ can be also written as

$$C(t) = tC_1 + t^2C_2 + t^3C_3 + t^4C_4 + \dots \tag{1.26}$$

which should naturally satisfy (1.23):

$$\mathrm{e}^{tA}\mathrm{e}^{tB} = \mathrm{e}^{tC_1+t^2C_2+t^3C_3+t^4C_4+\cdots}. \tag{1.27}$$

Therefore, in order to explicitly determine the coefficients of the series of $C(t)$, we insert the expansion (1.26) into (1.24), and compare like powers of t which yields

$$\left.\begin{aligned} C_1 &= A + B, \\ C_2 &= \frac{1}{4}[A-B, C_1] = \frac{1}{4}[A-B, A+B] = \frac{1}{2}[A, B], \\ C_3 &= \frac{1}{6}[A-B, C_2] + \frac{B_2}{6}\partial^2_{C_1}(A+B) = \frac{1}{12}\Big[A-B, [A, B]\Big] \\ &= \frac{1}{12}\Big[A, [A, B]\Big] + \frac{1}{12}\Big[B, [B, A]\Big], \\ C_4 &= \dots = \frac{1}{24}\Big[A, \Big[B, [B, A]\Big]\Big]. \end{aligned}\right\} \tag{1.28}$$

Using the BCH formula (1.23) and the coefficients (1.28) for $C(t)$, it is straightforward to see that the first order Lie formulae (1.11) and (1.12) verify, respectively,

$$U(\Delta t) - \mathcal{L}_1^{\Delta t}U_0 = \mathrm{e}^{\Delta t(A+B)}U_0 - \mathrm{e}^{\Delta tB}\mathrm{e}^{\Delta tA}U_0 = -\frac{\Delta t^2}{2}[B, A]U_0 + \mathcal{O}(\Delta t^3), \tag{1.29}$$

and

$$U(\Delta t) - \mathcal{L}_2^{\Delta t}U_0 = \mathrm{e}^{\Delta t(A+B)}U_0 - \mathrm{e}^{\Delta tA}\mathrm{e}^{\Delta tB}U_0 = -\frac{\Delta t^2}{2}[A, B]U_0 + \mathcal{O}(\Delta t^3). \tag{1.30}$$

It is important to notice that if the linear operators commute: $[A, B] = 0$, all the coefficients in the series of $C(t)$ are zero in (1.28) except for $C_1 = A + B$, and both Lie operators $\mathcal{L}_1^{\Delta t}$ and $\mathcal{L}_2^{\Delta t}$ act as the flow $\mathrm{e}^{\Delta t(A+B)}$ of the coupled system (1.1), according to (1.27).
Applying this time the BCH formula (1.23) to

$$\mathrm{e}^{tA/2}\mathrm{e}^{tB/2} = \mathrm{e}^{C(t)}, \tag{1.31}$$

and taking into account that

$$\mathrm{e}^{tB/2}\mathrm{e}^{tA/2} = \mathrm{e}^{-C(-t)}, \tag{1.32}$$

we can apply a second time the BCH formula (1.23) to

$$\mathrm{e}^{C(t)}\mathrm{e}^{-C(-t)} = \mathrm{e}^{tA/2}\mathrm{e}^{tB}\mathrm{e}^{tA/2} = \mathrm{e}^{S(t)}, \tag{1.33}$$

in order to obtain $S(t)$:

$$S(t) = tS_1 + t^3S_3 + t^5S_5 + \dots, \tag{1.34}$$

with

$$\left.\begin{aligned} S_1 &= A + B, \\ S_3 &= -\frac{1}{24}\Big[A, [A, B]\Big] + \frac{1}{12}\Big[B, [B, A]\Big]. \end{aligned}\right\} \tag{1.35}$$

Notice that only odd powers of t are present in (1.34) since the adjoint method of the symmetric scheme $\mathrm{e}^{tA/2}\mathrm{e}^{tB}\mathrm{e}^{tA/2}$ is obtained by just changing the sign of t and therefore of $\mathrm{e}^{S(t)}$, according to (1.33). In this case, $\mathrm{e}^{S(t)}$ is not other than the Strang scheme $\mathcal{S}_2^t$ according to (1.18), and we see that the local errors can be written as

$$\begin{aligned} U(\Delta t) - \mathcal{S}_1^{\Delta t} U_0 &= \mathrm{e}^{\Delta t(A+B)} U_0 - \mathrm{e}^{\Delta t B/2}\mathrm{e}^{\Delta t A}\mathrm{e}^{\Delta t B/2} U_0 \\ &= \frac{\Delta t^3}{24}\Big[B,[B,A]\Big]U_0 - \frac{\Delta t^3}{12}\Big[A,[A,B]\Big]U_0 + \mathcal{O}(\Delta t^4), \end{aligned} \tag{1.36}$$

and

$$\begin{aligned} U(\Delta t) - \mathcal{S}_2^{\Delta t} U_0 &= \mathrm{e}^{\Delta t(A+B)} U_0 - \mathrm{e}^{\Delta t A/2}\mathrm{e}^{\Delta t B}\mathrm{e}^{\Delta t A/2} U_0 \\ &= \frac{\Delta t^3}{24}\Big[A,[A,B]\Big]U_0 - \frac{\Delta t^3}{12}\Big[B,[B,A]\Big]U_0 + \mathcal{O}(\Delta t^4). \end{aligned} \tag{1.37}$$

In this way, we can formally represent the local errors of both Lie and Strang schemes. We remark that for both cases no splitting error is introduced for commuting operators. Furthermore, the latter error expressions can be easily extended to an arbitrary number of linear operators. However, it is important to notice that these estimates are asymptotically verified for sufficiently small splitting time steps Δt, since they are based on Taylor series expansions. Extension to general nonlinear configurations is straightforward using a Lie operator formalism [SSC94], in which case the same previous estimates remain valid with linear operators defined by the Lie derivatives associated with the various nonlinear operators, as we will show in what follows.

1.1.4 The Lie Operator Formalism

We introduce the Lie operator formalism in order to generalize the use of exponentials of linear operators in the context of nonlinear operators. Let X be a Banach space, $T > 0$, and an unbounded nonlinear operator F from $D(F) \subset X$ to X, we consider the general autonomous equation:

$$\left.\begin{aligned} &\mathrm{d}_t U = F(U(t)), && 0 < t < T, \\ &U(0) = U_0, && t = 0. \end{aligned}\right\} \tag{1.38}$$

The exact solution of this evolutionary equation is formally given by

$$U(t) = T^t U_0, \quad 0 \le t \le T, \tag{1.39}$$

where T^t is the semiflow associated with (1.38). The Lie operator D_F associated with F is then a linear operator acting on the space of operators defined in X (see, *e.g.*, [SSC94, HLW06, DT11]). More precisely, for any unbounded nonlinear operator G from $D(G) \subset X$ to X with Fréchet derivative G', D_F maps G into a new operator $D_F G$, such that for any v in X:

$$(D_F G)(v) = G'(v) F(v). \tag{1.40}$$

Using the chain rule for the solution $U(t)$ of (1.38), we have that

$$\partial_t G(U(t)) = (D_F G)(U(t)), \tag{1.41}$$

and hence applying the Lie operator iteratively, we obtain

$$\partial_t^n G(U(t)) = (D_F^n G)(U(t)). \tag{1.42}$$

A formal Taylor expansion yields[5]

$$G(U(t)) = \sum_{n=0}^{+\infty} \frac{t^n}{n!} \left(\partial_t^n G(U(t))\right)\Big|_{t=0} = \left(\sum_{n=0}^{+\infty} \frac{t^n}{n!} D_F^n G\right) U_0 = \left(\mathrm{e}^{tD_F} G\right) U_0. \tag{1.43}$$

If we now assume that G is the identity operator Id, we finally get

$$U(t) = T^t U_0 = \left(\mathrm{e}^{tD_F} \mathrm{Id}\right) U_0. \tag{1.44}$$

Therefore, the Lie operator is indeed a way of writing the solution of a nonlinear ODE in terms of a linear but differential operator.
Following (1.43), an important result obtained by Gröbner in [Grö67] considers the composition of two semiflows T_1^t and T_2^s associated with F_1 and F_2 for any v in X:

$$T_1^t T_2^s v = \left(\mathrm{e}^{sD_{F_2}} T_1^t\right) v = \left(\mathrm{e}^{sD_{F_2}} \mathrm{e}^{tD_{F_1}} \mathrm{Id}\right) v. \tag{1.45}$$

Notice that the order of the operators to the left and right are permuted for the equivalent representations in (1.45). The latter result can naturally be extended to more than two semiflows $T_1^t, T_2^s, \ldots, T_m^r$ associated with $F_1, F_2, \ldots, F_m$:

$$T_1^t T_2^s \cdots T_m^r v = \left(\mathrm{e}^{rD_{F_m}} \cdots \mathrm{e}^{sD_{F_2}} \mathrm{e}^{tD_{F_1}} \mathrm{Id}\right) v. \tag{1.46}$$

The same analysis previously detailed to estimate the splitting errors can be analogously performed by applying the Baker-Campbell-Hausdorff formula (1.23) to (1.45):

$$\mathrm{e}^{sD_{F_2}} \mathrm{e}^{tD_{F_1}} = \mathrm{e}^{D(s,t)}, \tag{1.47}$$

where the differential operator $D(s,t)$ is given by

$$\begin{aligned} D(s,t) \;=\; & sD_{F_2} + tD_{F_1} + \frac{st}{2}[D_{F_2}, D_{F_1}] + \frac{s^2 t}{12}\Big[D_{F_2}, [D_{F_2}, D_{F_1}]\Big] \\ & + \frac{st^2}{12}\Big[D_{F_1}, [D_{F_1}, D_{F_2}]\Big] + \frac{s^2 t^2}{24}\Big[D_{F_2}, \Big[D_{F_1}, [D_{F_1}, D_{F_2}]\Big]\Big] + \ldots \end{aligned} \tag{1.48}$$

according to (1.28). The *Lie bracket* for differential operators is defined exactly as for linear operators (1.21):

$$[D_{F_1}, D_{F_2}] = D_{F_1} D_{F_2} - D_{F_2} D_{F_1}, \tag{1.49}$$

and acts again as a linear differential operator:

$$[D_{F_1}, D_{F_2}] = \left(F_2' F_1 - F_1' F_2\right) \partial_v, \tag{1.50}$$

for any v in X according to (1.40).
In this way, considering a general system of nonlinear ODEs

$$\left.\begin{aligned} & \mathrm{d}_t U = F_1(U(t)) + F_2(U(t)), \quad t > 0, \\ & U(0) = U_0, \end{aligned}\right\} \tag{1.51}$$

with $U_0 \in \mathbb{R}^m$, $U : \mathbb{R} \to \mathbb{R}^m$, and $F_1, F_2 : \mathbb{R}^m \to \mathbb{R}^m$, the same asymptotic expressions for the local error estimates for the Lie and Strang formulae (1.29) and (1.30), and (1.36) and (1.37), can be recast with the linear operators A and B replaced by the Lie operators D_{F_1}

[5] We remark that if $F(U(t))$ is not an analytic function in (1.43), but $F \in \mathcal{C}^N(\mathbb{R})$, then the series has to be truncated and a $\mathcal{O}(t^N)$ remainder must be included.

and D_{F_2}. The same follows for an arbitrary number of operators. Furthermore, splitting order conditions can be then deduced by using this Lie formalism for general nonlinear operators [Yos90, HLW06]. In particular, it was with this representation that the commuting conditions for nonlinear or linear operators, yielding no splitting errors, were introduced in [LV99] for the splitting solution of reaction-convection-diffusion systems (see [HV03] for more details). Exact splitting error representations introduced in [DS02a] can be also analyzed in this framework for general nonlinear PDEs [DDL+12].

1.2 Splitting Errors for Time/Space Multi-Scale PDEs

In this second part, we will present some theoretical results previously introduced in the literature, to characterize the numerical behavior of splitting techniques for the solution of multi-scale PDEs. These multi-scale features might arise in time because of the presence of different numerical or physical evolution rates within a rather broad range, or in space because of the presence of steep gradients or large higher order spatial derivatives within the computational domain. More likely, they are coupled both in time and space throughout the numerical integration. As a consequence, there might be some perturbing effects in the accuracy of the numerical approximations of the governing equations, traduced usually by an order reduction of the splitting method. This kind of numerical difficulty might be theoretically characterized as a direct result of the stiffness of the time dependent equations as we will discuss in the next chapter, and generally speaking we can say that we are dealing with the numerical solution of stiff PDEs.
In what follows we detail some elements to describe the numerical behavior of splitting schemes faced with the mentioned stiffness, in the case of reaction-diffusion systems. The study of this kind of problem allows us to illustrate the numerical difficulties encountered in general, and the resulting conclusions might be partially extended to more complex configurations. Nevertheless, there is a continuous research in this field and more detailed mathematical descriptions are always needed to further understand these issues.

1.2.1 Mathematical Framework: Reaction-Diffusion Systems

We focus on a class of multi-scale phenomena that can be modeled by general reaction-diffusion systems of type:

$$\left.\begin{aligned} &\partial_t U - \partial_{\mathrm{x}} \cdot (D(U)\partial_{\mathrm{x}} U) = F(U), \quad \mathrm{x} \in \mathbb{R}^d,\ t > 0, \\ &U(0,\mathrm{x}) = U_0(\mathrm{x}) \qquad\qquad\qquad\quad \mathrm{x} \in \mathbb{R}^d, \end{aligned}\right\} \tag{1.52}$$

where $F : \mathbb{R}^m \to \mathbb{R}^m$, $U_0 : \mathbb{R}^d \to \mathbb{R}^m$ and $U : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^m$, with the diffusion matrix $D(U)$, which is a tensor of order $d \times d \times m$. In case we are only considering linear diagonal diffusion, the elements of the diffusion matrix are written as $D_{i_1 i_2 i_3}(U) = D_{i_3}\delta_{i_1 i_2}$ with indices i_1, i_2, $i_3 = 1, \ldots, m$, so that the diffusion operator reduces to the heat operator with scalar diffusion coefficient D_{i_3} for component $u^{(i_3)}$ of U, and the system (1.52) becomes

$$\left.\begin{aligned} &\partial_t U - D\,\partial_{\mathrm{x}}^2 U = F(U), \quad \mathrm{x} \in \mathbb{R}^d,\ t > 0, \\ &U(0,\mathrm{x}) = U_0(\mathrm{x}) \qquad\quad \mathrm{x} \in \mathbb{R}^d. \end{aligned}\right\} \tag{1.53}$$

In general, the source term F into (1.52) and (1.53) models reactive chemical mechanisms with a broad time scale spectrum. On the other hand, complementary stiffness results from the potentially fast scales introduced in the numerical solution when applying the diffusion operator to localized steep spatial gradients or highly inhomogeneous distributions, as it is

usually the case in physical phenomena characterized by the presence of fronts or irregular space multi-scale configurations. In this way, the associated stiffness will surely have an effect on the numerical behavior of the splitting schemes as we will briefly describe in the following.

1.2.2 Splitting Order Reduction for Time Multi-Scale Systems

Even though splitting schemes are usually quite efficient for the solution of time dependent equations, several works showed that the standard numerical analysis of splitting schemes fails in presence of scales much faster than the splitting time step [GPMD88, D'A94, DL95a, YP98, VS98, SBP00], and that an order reduction of the methods is numerically observed. In particular, a first major step towards a rigorous study of such cases was conducted by Sportisse in [Spo00] in the framework of a linear system of ODEs, issued from a reaction-diffusion system with a linear source term and diagonal diffusion. In this work, a fast characteristic time was associated with the source term by means of a multiplying factor ϵ^{-1}, with small ϵ, to split the original system into a stiff and a non stiff subproblem. In this context, a local order reduction of the splitting schemes was mathematically described based on singular perturbation theory, whereas splitting methods ending with the stiffest operator were also shown to be more accurate than the others. Similar conclusions were obtained by Kozlov *et al.* in [KrO04] for nonlinear systems of ODEs, split also into stiff and non stiff parts, using singular perturbation elements as well. In this framework, Descombes & Massot introduced in [DM04] a general theoretical approach for nonlinear reaction-diffusion systems with time multi-scale features issued from more realistic physical configurations. We will briefly describe in the following a few results coming from [DM04].

Supposing that the system (1.53) shows a well partitioned structure such that $U = (u^\epsilon, v^\epsilon)^T$ and thus $F(U) = (f(u^\epsilon, v^\epsilon), g(u^\epsilon, v^\epsilon)/\epsilon)^T$, where $u^\epsilon \in \mathbb{R}^{m^{\text{slow}}}$ and $v^\epsilon \in \mathbb{R}^{m^{\text{fast}}}$ stand, respectively, for the slow and fast variables of the dynamical system associated with (1.53), and $m = m^{\text{slow}} + m^{\text{fast}}$; we consider the following reaction-diffusion system:

$$\left.\begin{aligned} &\partial_t u^\epsilon - \partial_{\mathrm{x}}^2 u^\epsilon = f(u^\epsilon, v^\epsilon), && \mathrm{x} \in \mathbb{R}^d,\ t > 0, \\ &\partial_t v^\epsilon - \partial_{\mathrm{x}}^2 v^\epsilon = \frac{g(u^\epsilon, v^\epsilon)}{\epsilon}, && \mathrm{x} \in \mathbb{R}^d,\ t > 0, \\ &u^\epsilon(0, \mathrm{x}) = u_0(\mathrm{x}), && \mathrm{x} \in \mathbb{R}^d, \\ &v^\epsilon(0, \mathrm{x}) = v_0(\mathrm{x}), && \mathrm{x} \in \mathbb{R}^d, \end{aligned}\right\} \tag{1.54}$$

for a small parameter ϵ and the identity in $\mathcal{M}_m(\mathbb{R})$, as diffusion matrix. For the sake of brevity, we will only consider this diagonal case, even though a quasi-linear non-diagonal diffusion was also analyzed in [DM04]. We denote by $(u^\epsilon(t), v^\epsilon(t)) = T_\epsilon^t(u_0, v_0)$ the solution of (1.54) at some time t.

In order to settle an appropriate mathematical framework, we assume that this system admits an entropic structure [Mas02] so that the source term admits a well partitioned Tikhonov normal form [TVS85]. Therefore, there is a partial equilibrium manifold where the fast time scales have been relaxed, which is globally stable. In particular, the entropy is a global Lyapounov function and we can thus perform a singular perturbation analysis with asymptotic expansions [Mas02]. In this context, we can consider the singular perturbation analysis for the

finite dimensional dynamical system:

$$\left.\begin{aligned} &\mathrm{d}_t\bar{u}^\epsilon = f(\bar{u}^\epsilon,\bar{v}^\epsilon), \quad t>0,\\ &\mathrm{d}_t\bar{v}^\epsilon = \frac{g(\bar{u}^\epsilon,\bar{v}^\epsilon)}{\epsilon}, \quad t>0,\\ &\bar{u}^\epsilon(0)=\bar{u}_0,\\ &\bar{v}^\epsilon(0)=\bar{v}_0, \end{aligned}\right\} \tag{1.55}$$

which corresponds to a homogeneous system without diffusion. The corresponding reduced system can thus be written as

$$\left.\begin{aligned} &\mathrm{d}_t\bar{u} = G(\bar{u}), \qquad t>0,\\ &\bar{u}(0)=\bar{u}_0,\\ &\bar{v}(t)=h(\bar{u}(t)), \quad t\geq 0, \end{aligned}\right\} \tag{1.56}$$

where $G(\bar{u}) = f(\bar{u},h(\bar{u}))$, and $g(\bar{u},\bar{v}) = \bar{v} - h(\bar{u}) = 0$. The inner boundary layer, because of the well-partitioned structure of the dynamical system, can be considered as a projection step in an affine manifold onto the partial equilibrium $h(\bar{u}_0)$ in the $\bar{v}$ variable. Denoting by $\Pi_0\bar{v}$ the associated variable centered at $h(\bar{u}_0)$, the boundary layer, parametrized by the spatial coordinate x, can be described by the following differential equation:

$$\left.\begin{aligned} &\mathrm{d}_\tau\Pi_0\bar{v} = g(u_0,h(u_0)+\Pi_0\bar{v}), \quad \tau>0,\\ &\Pi_0\bar{v}(0) = v_0 - h(u_0), \end{aligned}\right\} \tag{1.57}$$

for a time scale defined by $\tau = t/\epsilon$.
Assuming that there exists a convex compact set K which contains the initial condition $(\bar{u}_0,\bar{v}_0)\in K$, and which is invariant by (1.54), (1.56) and (1.57), it has been proved in [Mas02] that for ϵ sufficiently small, we have for $t\in[0,+\infty)$:

$$\bar{v}^\epsilon(t,\epsilon) = \Pi_0\bar{v}(t/\epsilon) + \bar{v}(t) + \mathcal{O}(\epsilon), \tag{1.58}$$

$$\bar{u}^\epsilon(t,\epsilon) = \bar{u}(t) + \mathcal{O}(\epsilon), \tag{1.59}$$

and for some $\kappa>0$, we obtain an estimate for the inner boundary layer

$$\Pi_0\bar{v}(t/\epsilon) = \mathcal{O}\left(\mathrm{e}^{\left(-\kappa t/\epsilon\right)}\right). \tag{1.60}$$

Considering now the reduced problem associated with the complete system (1.54):

$$\left.\begin{aligned} &\partial_t u - \partial_{\mathrm{x}}^2 u = G(u), \quad &\mathrm{x}\in\mathbb{R}^d,\ t>0,\\ &u(0,\mathrm{x}) = u_0(\mathrm{x}), \quad &\mathrm{x}\in\mathbb{R}^d,\\ &v(t,\mathrm{x}) = h(u(t,\mathrm{x})), \quad &\mathrm{x}\in\mathbb{R}^d, t\geq 0, \end{aligned}\right\} \tag{1.61}$$

and based on the previous singular perturbation analysis as detailed in [DM04], if we assume that $(\bar{u}_0(\mathrm{x}),\bar{v}_0(\mathrm{x}))\in K$ for $\mathrm{x}\in\mathbb{R}^d$ and that the solution $T^t u_0 = (u(t),h(u(t))$ of (1.61) leaves also K invariant, for ϵ sufficiently small, we have for $t\in[0,+\infty)$:

$$\|u^\epsilon(t,\epsilon) - u(t)\|_{L^2} = \mathcal{O}(\epsilon), \tag{1.62}$$

$$\|v^\epsilon(t,\epsilon) - \Pi_0\bar{v}(t/\epsilon) - h(u(t))\|_{L^2} = \mathcal{O}(\epsilon), \tag{1.63}$$

and the corresponding estimate for the inner boundary layer:

$$\|\Pi_0\bar{v}(t/\epsilon)\|_{L^2} = \mathcal{O}\left(\mathrm{e}^{\left(-\kappa t/\epsilon\right)}\right). \tag{1.64}$$

With this framework, we introduce the standard decoupling of the diffusion and reaction problems for system (1.54). Let us then denote by $X^t(u_0, v_0)$ the solution of the diffusion problem:

$$\left.\begin{aligned} \partial_t u_D - \partial^2_{\mathrm{x}} u_D = 0, \quad \mathrm{x} \in \mathbb{R}^d,\ t > 0, \\ \partial_t v_D - \partial^2_{\mathrm{x}} v_D = 0, \quad \mathrm{x} \in \mathbb{R}^d,\ t > 0, \end{aligned}\right\} \tag{1.65}$$

for some initial data $u_D(0,\cdot) = u_0(\cdot)$ and $v_D(0,\cdot) = v_0(\cdot)$; and by $Y^t_\epsilon(u_0, v_0)$ the solution of the reaction problem:

$$\left.\begin{aligned} \partial_t u^\epsilon_R = f(u^\epsilon_R, v^\epsilon_R), \quad \mathrm{x} \in \mathbb{R}^d, t > 0, \\ \partial_t v^\epsilon_R = \frac{g(u^\epsilon_R, v^\epsilon_R)}{\epsilon}, \quad \mathrm{x} \in \mathbb{R}^d, t > 0, \end{aligned}\right\} \tag{1.66}$$

with initial data $u^\epsilon_R(0,\cdot) = u_0(\cdot)$ and $v^\epsilon_R(0,\cdot) = v_0(\cdot)$, where the spatial coordinate x can be considered as a parameter. The Lie and Strang splitting formulae associated with (1.54) are given by:

$$\mathcal{L}^t_{1,\epsilon}(u_0, v_0) = X^t Y^t_\epsilon(u_0, v_0), \tag{1.67}$$

$$\mathcal{L}^t_{2,\epsilon}(u_0, v_0) = Y^t_\epsilon X^t(u_0, v_0), \tag{1.68}$$

$$\mathcal{S}^t_{1,\epsilon}(u_0, v_0) = X^{t/2} Y^t_\epsilon X^{t/2}(u_0, v_0), \tag{1.69}$$

$$\mathcal{S}^t_{2,\epsilon}(u_0, v_0) = Y^{t/2}_\epsilon X^t Y^{t/2}_\epsilon(u_0, v_0). \tag{1.70}$$

If we consider now the reduced problem of (1.66) when ϵ tends to zero:

$$\left.\begin{aligned} \partial_t u_R = f(u_R, h(u_R)) = G(u_R), \quad \mathrm{x} \in \mathbb{R}^d, t > 0, \\ u_R(0, \mathrm{x}) = u_0(\mathrm{x}), \qquad\qquad \mathrm{x} \in \mathbb{R}^d, \\ v_R(t, \mathrm{x}) = h(u_R(t, \mathrm{x})), \qquad\quad \mathrm{x} \in \mathbb{R}^d, t \geq 0, \end{aligned}\right\} \tag{1.71}$$

with solution given by $(u_R(t), h(u_R(t)) = Y^t u_0$ as for (1.56), we define the corresponding reduced splitting schemes:

$$\mathcal{L}^t_1 u_0 = X^t Y^t u_0, \tag{1.72}$$

$$\mathcal{L}^t_2(u_0, v_0) = Y^t X^t(u_0, v_0), \tag{1.73}$$

$$\mathcal{S}^t_1(u_0, v_0) = X^{t/2} Y^t X^{t/2}(u_0, v_0), \tag{1.74}$$

$$\mathcal{S}^t_2 u_0 = Y^{t/2} X^t Y^{t/2} u_0, \tag{1.75}$$

where the fast scales have been previously relaxed in the reaction part by considering the reduced problem (1.71).
To study the order of approximation of the exact solution T^t_ϵ of the coupled problem (1.54) by the splitting schemes (1.67)-(1.70), we investigate the order of approximation of T^t associated

with the reduced problem (1.61) by the reduced splitting schemes (1.72)-(1.75). Defining the corresponding local errors:

$$\left.\begin{array}{rcl}(u_{\rm err1}, v_{\rm err1}) &=& T^t u_0 - \mathcal{L}_1^t u_0,\\ (u_{\rm err2}, v_{\rm err2}) &=& T^t u_0 - \mathcal{L}_2^t (u_0, v_0),\\ (u_{\rm err3}, v_{\rm err3}) &=& T^t u_0 - \mathcal{S}_1^t (u_0, v_0),\\ (u_{\rm err4}, v_{\rm err4}) &=& T^t u_0 - \mathcal{S}_2^t u_0,\end{array}\right\} \tag{1.76}$$

it was demonstrated in [DM04] that the local error for the slow and fast variables of the various splitting schemes satisfies

$$\|u_{\rm err1}\|_{L^2} = \mathcal{O}(t^2), \qquad \|v_{\rm err1}\|_{L^2} = \mathcal{O}(t), \tag{1.77}$$

$$\|u_{\rm err2}\|_{L^2} = \mathcal{O}(t^2), \qquad \|v_{\rm err2}\|_{L^2} = \mathcal{O}(t^2), \tag{1.78}$$

$$\|u_{\rm err3}\|_{L^2} = \mathcal{O}(t^3), \qquad \|v_{\rm err3}\|_{L^2} = \mathcal{O}(t), \tag{1.79}$$

$$\|u_{\rm err4}\|_{L^2} = \mathcal{O}(t^3), \qquad \|v_{\rm err4}\|_{L^2} = \mathcal{O}(t^3). \tag{1.80}$$

Taking into account that, for instance, for $\mathcal{L}_{1,\epsilon}^t(u_0, v_0)$ the error of approximation with respect to $T_\epsilon^t(u_0, v_0)$ is given by

$$\begin{array}{rcl} T_\epsilon^t(u_0, v_0) - \mathcal{L}_{1,\epsilon}^t(u_0, v_0) &=& T_\epsilon^t(u_0, v_0) - T^t u_0 + T^t u_0 - \mathcal{L}_1^t u_0 \\ && + \mathcal{L}_1^t u_0 - \mathcal{L}_{1,\epsilon}^t(u_0, v_0), \end{array} \tag{1.81}$$

and that

$$\begin{array}{rcl} \|T_\epsilon^t(u_0, v_0) - \mathcal{L}_{1,\epsilon}^t(u_0, v_0)\|_{L^2} &\leq& \|T_\epsilon^t(u_0, v_0) - T^t u_0\|_{L^2} + \|T^t u_0 - \mathcal{L}_1^t u_0\|_{L^2} \\ && + \|\mathcal{L}_1^t u_0 - \mathcal{L}_{1,\epsilon}^t(u_0, v_0)\|_{L^2}, \end{array} \tag{1.82}$$

for ϵ sufficiently small and for $t \geq 0$ sufficiently small, the local errors admit the following asymptotic expansions [DM04]:

$$\|T_\epsilon^t(u_0, v_0) - \mathcal{L}_{1,\epsilon}^t(u_0, v_0)\|_{L^2} = \mathcal{O}(t) + \mathcal{O}\left(\mathrm{e}^{\left(-\kappa t/\epsilon\right)}\right) + \mathcal{O}(\epsilon), \tag{1.83}$$

$$\|T_\epsilon^t(u_0, v_0) - \mathcal{S}_{1,\epsilon}^t(u_0, v_0)\|_{L^2} = \mathcal{O}(t) + \mathcal{O}\left(\mathrm{e}^{\left(-\kappa t/\epsilon\right)}\right) + \mathcal{O}(\epsilon), \tag{1.84}$$

and

$$\|T_\epsilon^t(u_0, v_0) - \mathcal{L}_{2,\epsilon}^t(u_0, v_0)\|_{L^2} = \mathcal{O}(t^2) + \mathcal{O}\left(\mathrm{e}^{\left(-\kappa t/\epsilon\right)}\right) + \mathcal{O}(\epsilon), \tag{1.85}$$

$$\|T_\epsilon^t(u_0, v_0) - \mathcal{S}_{2,\epsilon}^t(u_0, v_0)\|_{L^2} = \mathcal{O}(t^3) + \mathcal{O}\left(\mathrm{e}^{\left(-\kappa t/\epsilon\right)}\right) + \mathcal{O}(\epsilon), \tag{1.86}$$

considering estimates (1.77)-(1.80) for the second term of the right hand side of (1.82), and (1.62)-(1.64) for the other two terms.
Through this mathematical model and the corresponding numerical analysis, we can conclude that no order reduction of the splitting schemes is expected for the slow variables whenever we

consider splitting time steps much larger than the fastest scales present in the problem: $t > \epsilon$, following [DM04]. On the other hand, for a linear diagonal diffusion, if we use splitting schemes ending with the reaction operator which includes the fastest scales, then there is no reason to expect order reductions not even for the fast variables. In particular, in the configuration of a partial equilibrium manifold with non zero curvature, a situation which can only be obtained with a nonlinear reaction source term, the splitting schemes ending with the diffusion operator encounter an order reduction related to the Lie bracket between the Laplacian operator and the h function defining the partial equilibrium manifold [DM04]. Finally, let us recall that in practical implementations of splitting techniques, dedicated solvers must be considered to properly handle the fast transients associated with the inner boundary layers given by (1.60), as previously remarked [VBvLS96, SVdZ+98, VSBH99][6], and also to ensure the mathematical framework detailed in this section in which the split reaction and diffusion subproblems were exactly solved for estimates (1.83)-(1.86).

1.2.3 Splitting Errors with High Spatial Gradients

We have seen in the previous study that the classical error representations of splitting schemes are not always enough to describe more precisely some important features related to the modeling equations. Therefore, more rigorous studies were performed and in particular an exact representation of the local errors of splitting schemes was achieved by Descombes & Schatzman in [DS02a] for general linear problems like (1.1). Once again, extension to nonlinear operators is straightforward using a Lie operator formalism as shown in [DDL+12]. These results led to many further mathematical studies on splitting errors (see, *e.g.*, [DT10, DT11]), and such a precise error representation showed to be mandatory to better analyze some particular issues like the influence of high spatial gradients on the solution of reaction-diffusion systems solved by splitting techniques [DDLM07, DDM11, DDL+12]. In this way, it is possible to better depict some potential numerical difficulties issued this time from the space multi-scale character of some physical phenomena modeled by the governing equations, *e.g.*, (1.52), as previously remarked and as analyzed, for instance, in [RSO04, RS05a].
Let us recall the initial value problem (1.1), for some linear operators A, $B \in \mathcal{M}_m(\mathbb{R})$, $U_0 \in \mathbb{R}^m$, $U : \mathbb{R} \to \mathbb{R}^m$:

$$\left.\begin{array}{l} \mathrm{d}_t U + AU + BU = 0, \quad t > 0, \\ U(0) = U_0, \end{array}\right\} \tag{1.87}$$

for which the exact solution is given by

$$U(t) = \mathrm{e}^{-t(A+B)} U_0, \qquad t \geq 0. \tag{1.88}$$

The first order Lie and the second order Strang splitting formulae are given, for instance, by

$$\mathcal{L}_2^t U_0 = \mathrm{e}^{-tA}\mathrm{e}^{-tB} U_0, \tag{1.89}$$

and

$$\mathcal{S}_2^t U_0 = \mathrm{e}^{-tA/2}\mathrm{e}^{-tB}\mathrm{e}^{-tA/2} U_0. \tag{1.90}$$

In this context, it was proved in [DS02a] that the following identities hold:

$$\mathcal{L}_2^t \;=\; \mathrm{e}^{-t(A+B)} + \int_0^t \int_0^s \mathrm{e}^{-(t-s)(A+B)}\mathrm{e}^{-(s-r)A}\big(\partial_A B\big)\mathrm{e}^{-rA}\mathrm{e}^{-sB}\,\mathrm{d}r\,\mathrm{d}s, \tag{1.91}$$

[6] The same remark is valid for the numerical integration of stiff ODEs [HLR88, HW96].

$$\begin{aligned}\mathcal{S}_2^t &= \mathrm{e}^{-t(A+B)}+ \\ &\frac{1}{4}\int_0^t\int_0^s (s-r)\mathrm{e}^{-(t-s)(A+B)}\mathrm{e}^{-(s-r)A/2}\big(\partial_A^2 B\big)\mathrm{e}^{-rA/2}\mathrm{e}^{-sB}\mathrm{e}^{-sA/2}\,\mathrm{d}r\,\mathrm{d}s \\ &-\frac{1}{2}\int_0^t\int_0^s (s-r)\mathrm{e}^{-(t-s)(A+B)}\mathrm{e}^{-sA/2}\mathrm{e}^{-rB}\big(\partial_B^2 A\big)\mathrm{e}^{-(s-r)B}\mathrm{e}^{-sA/2}\,\mathrm{d}r\,\mathrm{d}s.\end{aligned} \tag{1.92}$$

These new estimates provide then an exact representation of the local errors, comparing with previous estimates for $\mathcal{L}_2^t$ (1.30) and $\mathcal{S}_2^t$ (1.37). It follows the same for $\mathcal{L}_1^t$ and $\mathcal{S}_1^t$.
In order to illustrate the influence of space multi-space phenomena given, for instance, by high spatial gradients in the solutions of the PDEs, we will consider a simplified scalar reaction-diffusion system coming from (1.53), with $m = 1$ and $d = 1$:

$$\left.\begin{aligned}&\partial_t u - \partial_x^2 u + V(x)u = 0 \quad x \in \mathbb{R}, t > 0,\\ &u(x,0) = u_0(x) \qquad\qquad x \in \mathbb{R},\end{aligned}\right\} \tag{1.93}$$

where $V : \mathbb{R} \to \mathbb{R}$ is supposed to be a positive and bounded function of class $\mathcal{C}^\infty(\mathbb{R})$ with all bounded derivatives, and the L^2-norm of the derivative of the smooth initial condition u_0 is assumed to be very high. Similar systems were considered in [DDLM07, DDM11, DDL+12] where in particular V can be seen as coming from the linearizion of the corresponding scalar reaction term $f(u)$ in (1.53). Considering that the linear operator A in (1.87) corresponds to the multiplication by V and that $B = -\partial_x^2$ (minus the second partial derivative with respect to x in one dimension), their commutator (1.21) is given by

$$\partial_A B = [A, B] = (\partial_x^2 V) + 2(\partial_x V)\partial_x. \tag{1.94}$$

If we now define

$$\mathrm{E}_{\mathcal{L}_2}^t = \mathrm{e}^{t(\partial_x^2 - V)} - \mathrm{e}^{-tV}\mathrm{e}^{t\partial_x^2}, \tag{1.95}$$

and consider (1.91), we can write the local error associated with the $\mathcal{L}_2^t$ scheme for system (1.93) as

$$\mathrm{E}_{\mathcal{L}_2}^t u_0 = -\int_0^t\int_0^s \mathrm{e}^{-(t-s)(\partial_x^2 - V)}\mathrm{e}^{-(s-r)V}\big(\partial_A B\big)\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2}u_0\,\mathrm{d}r\,\mathrm{d}s, \tag{1.96}$$

with commutator $\partial_A B$ given by (1.94). Taking norms, we have that in $L^2(\mathbb{R})$:

$$\begin{aligned}\big\|\mathrm{E}_{\mathcal{L}_2}^t u_0\big\|_{L^2} &\le \int_0^t\int_0^s \Big\|\mathrm{e}^{-(t-s)(\partial_x^2 - V)}\mathrm{e}^{-(s-r)V}\big(\partial_A B\big)\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2}u_0\Big\|_{L^2}\,\mathrm{d}r\,\mathrm{d}s \\ &\le \int_0^t\int_0^s \Big\|\big(\partial_A B\big)\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2}u_0\Big\|_{L^2}\,\mathrm{d}r\,\mathrm{d}s.\end{aligned} \tag{1.97}$$

Since

$$\begin{aligned}\big(\partial_A B\big)\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2}u_0 &= (\partial_x^2 V)\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2}u_0 + 2(\partial_x V)\partial_x\left(\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2}u_0\right) \\ &= (\partial_x^2 V)\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2}u_0 - 2(\partial_x V)r(\partial_x V)\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2}u_0 \\ &\quad +2(\partial_x V)\mathrm{e}^{-rV}\partial_x\left(\mathrm{e}^{s\partial_x^2}u_0\right) \\ &= (\partial_x^2 V)\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2}u_0 - 2(\partial_x V)r(\partial_x V)\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2}u_0 \\ &\quad +2(\partial_x V)\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2}\partial_x u_0,\end{aligned} \tag{1.98}$$

the integration of (1.97) yields

$$\left\|\mathrm{E}^t_{\mathcal{L}_2} u_0\right\|_{L^2} \leq \left(\frac{t^2}{2}\|\partial_x^2 V\|_\infty + \frac{t^3}{3}\|\partial_x V\|_\infty^2\right)\|u_0\|_{L^2} + t^2\|\partial_x V\|_\infty \|\partial_x u_0\|_{L^2}. \tag{1.99}$$

Nevertheless, we have supposed that the L^2-norm of $\partial_x u_0$ is very high, therefore the latter error bound is only interesting if the splitting time step t is sufficiently small. It is then specially relevant in this stiff configuration to obtain alternative error estimates which do not involve the derivative of the initial condition [DDLM07]. Thanks to the regularizing effect of the Laplacian, we can demonstrate through a Fourier transform of the diffusion operator, that for all $u_0 \in L^2$ and for $t > 0$:

$$\|\partial_x \mathrm{e}^{t\partial_x^2} u_0\|_{L^2} \leq \frac{1}{\sqrt{2\mathrm{e}t}}\|u_0\|_{L^2}. \tag{1.100}$$

Therefore, taking into account that

$$\begin{aligned}\left(\partial_A B\right)\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2} u_0 \;&=\; (\partial_x^2 V)\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2} u_0 - 2(\partial_x V) r (\partial_x V)\mathrm{e}^{-rV}\mathrm{e}^{s\partial_x^2} u_0 \\ &\quad + 2(\partial_x V)\mathrm{e}^{-rV}\partial_x\left(\mathrm{e}^{s\partial_x^2} u_0\right),\end{aligned} \tag{1.101}$$

into (1.97), its integration now yields

$$\left\|\mathrm{E}^t_{\mathcal{L}_2} u_0\right\|_{L^2} \leq \left(\frac{4}{3}t\sqrt{t}\frac{\|\partial_x V\|_\infty}{\sqrt{2\mathrm{e}}} + \frac{t^2}{2}\|\partial_x^2 V\|_\infty + \frac{t^3}{3}\|\partial_x V\|_\infty^2\right)\|u_0\|_{L^2}. \tag{1.102}$$

An order reduction is thus shown to appear in the local error estimate [DDLM07]. Similar conclusions are drawn considering the $\mathcal{L}_1^t$-Lie scheme, explicit computations of the estimates can be found in [DDM11]. Estimates (1.99) and (1.102) describe then the behavior of the local errors, and we see that for $t > 0$:

$$\left\|\mathrm{E}^t_{\mathcal{L}_2} u_0\right\|_{L^2} \propto \left(\|\partial_x u_0\|_{L^2} t^2, \|u_0\|_{L^2} t^{1.5}\right). \tag{1.103}$$

The first term is more relevant when t is sufficiently small, whereas the second one when t is not small enough and $\|\partial_x u_0\|_{L^2}$ is very high. More precisely, there exists some constant $\theta > 0$ such that for $t \leq \theta$, $\|\mathrm{E}^t_{\mathcal{L}_2} u_0\|_{L^2}$ behaves like t^2 and for $t \geq \theta$, $\|\mathrm{E}^t_{\mathcal{L}_2} u_0\|_{L^2}$ behaves like $t^{1.5}$ [DDLM07, DDM11, DDL+12].
In the same way, defining for the $\mathcal{S}_2^t$-Strang scheme

$$\mathrm{E}^t_{\mathcal{S}_2} = \mathrm{e}^{t(\partial_x^2 - V)} - \mathrm{e}^{-tV/2}\mathrm{e}^{t\partial_x^2}\mathrm{e}^{-tV/2}, \tag{1.104}$$

and considering (1.92), we can also write the local error associated with the $\mathcal{S}_2^t$ scheme for system (1.93). An order reduction can be once again detected and estimated for these stiff configurations. The explicit computations are shown in [DDM11], that finally yield

$$\left\|\mathrm{E}^t_{\mathcal{S}_2} u_0\right\|_{L^2} \propto \left(\|\partial_x u_0\|_{L^2} t^3, \|u_0\|_{L^2} t^2\right), \tag{1.105}$$

so that the local error $\|\mathrm{E}^t_{\mathcal{S}_2} u_0\|_{L^2}$ behaves either like t^3 for small splitting time steps or like t^2 with a consequent order reduction of the scheme.
It can thus be seen through these theoretical illustrations that an order reduction may arise for both Lie and Strang schemes whenever the solution features high spatial gradients. On the other hand, the hypothesis of a linear source term in (1.93) have just allowed us to simplify the computations and to better target the analysis on the effects of the diffusion operator on the

solution. These theoretical estimates were validated through some numerical tests presented in [DDLM07, DDM11, DDL+12] for stiff problems coming from nonlinear chemical dynamics. Taking into account that in the numerical applications envisioned in this work some of them are characterized by propagating fronts with potentially steep spatial gradients, an influence of the formers may be observed in the accuracy order of the splitting schemes. More precisely, an order reduction will likely arise for both Lie and Strang formulae for sufficiently large splitting time steps Δt. Nevertheless, the mathematical description introduced in these studies confirms that from a practical point of view the splitting errors are still set by the splitting time step even for this type of stiff configuration, whereas on the other hand a more precise theoretical understanding of the splitting errors for non asymptotic regimes was achieved. Finally, as in the previous mathematical descriptions, the numerical solvers implemented in practice should solve correctly the time evolution associated with each operator. For instance, Ropp & Shadid showed in [RS05a, RS09] that better results are obtained when using an L-stable method for the numerical solution of the diffusion in, respectively, reaction-diffusion and reaction-diffusion-convection problems[7].

[7] We will see in the following chapter that L-stability allows us to rapidly damp out fast numerical transients associated in this particular case with high frequencies or wave numbers arising when the discretized Laplacian operator is applied to a given solution (see, *e.g.*, [HW96, HV03]).

Chapter 2

Runge-Kutta Methods for Time Integration of Stiff ODEs

In the last chapter, we have first considered splitting techniques for the solution of linear systems of ODEs of type (1.1), with a general mathematical description on the numerical errors of such methods. A formal extension to general nonlinear systems was also detailed by means of the Lie operator formalism. We have then discussed the numerical solution by splitting methods of stiff PDEs for reaction-diffusion systems like (1.52), modeling potentially multi-scale phenomena. A theoretical characterization of the splitting errors was thus presented in the context of time and space stiff reaction-diffusion problems, which has introduced a few criteria to take into account, even for more complex PDEs. Even though the latter studies have led to the description of some numerical difficulties issued from the modeling PDEs, we have not given any detail on the solution of the split subproblems. Actually, throughout all these analyses we have assumed that the subsystems of equations were exactly solved in order to characterize only numerical errors coming from the splitting scheme. In this way, we have not considered yet either the time or space discretizations, or the numerical time integration of the associated subproblems. Nevertheless, it is quite natural to expect that the same numerical features of these modeling equations that influence the splitting accuracy, will also be present during the numerical solution of each split subproblem.

We have seen that in the context of splitting techniques we aim at solving independently and successively different time dependent systems of equations, starting from the immediately previous numerical solution. Hence, several initial value problems or Cauchy problems for PDEs are to be considered within each splitting time step. Therefore, in this chapter we will focus on the so-called *one-step integration methods* which contrarily to *multi-step methods*, do not require initial lower order approximations to build the numerical solution of each initial value problem. In this way, in this chapter we will first characterize some numerical difficulties associated with the solution of the ODEs issued from the previous problems to then describe some one-step Runge-Kutta methods that were developed in the past years to efficiently cope with these matters. In particular, we will concentrate on implicit and stabilized explicit Runge-Kutta schemes that have shown to be very efficient for the numerical solution of, respectively, reaction and diffusion problems, as an illustration of proper selection criteria of time integration solvers for the split subproblems issued from a splitting technique. For further details, an exhaustive mathematical description and analysis on the numerical solution of stiff systems of ODEs can be found in the book of Hairer & Wanner [HW96].

2.1 Characterization of Stiffness

Let us consider for $t > 0$, the scalar initial value problem:

$$\left.\begin{aligned} \mathrm{d}_t u &= f(t, u(t)), \\ u(0) &= u_0, \end{aligned}\right\} \tag{2.1}$$

with some $u_0 \in \mathbb{R}$ and $u : \mathbb{R} \to \mathbb{R}$, $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$. We aim at obtaining a numerical approximation u_n of the exact solution $u(t_n)$ of (2.1) for a time discretization given by $t_0 = 0 < t_1 < \ldots < t_n < \ldots$, and $n = 0, 1, \ldots$.
Nevertheless, we assume, and therefore we must take into account, that (2.1) is a stiff problem for which a precise and simple notion of stiffness is given in [HW96]:

> *"Stiff equations are problems for which explicit methods don't work."*

In order to illustrate this, we will first approximate the solution of (2.1) at some $t_1 = t_0 + \delta t$

$$u(\delta t) = u_0 + \int_{t_0}^{t_0+\delta t} f(t, u(t))\, \mathrm{d}t, \tag{2.2}$$

by

$$u_1 = u_0 + \delta t f(t_0, u_0), \tag{2.3}$$

which implies an explicit time discretization solution of (2.2) and it is known as the *explicit Euler* method, where δt is defined as the integration time step. It is straightforward to see that this is a first order method according to (1.6).
Taking a very simple case for (2.1), given by

$$\left.\begin{aligned} \mathrm{d}_t u &= -100\, u, \\ u(0) &= u_0, \end{aligned}\right\} \tag{2.4}$$

with exact solution $u(\delta t) = \mathrm{e}^{-100\,\delta t} u_0$ at $t_1 = \delta t$. We have that u_1 computed by (2.3) is given by

$$u_1 = u_0 - 100\, \delta t u_0. \tag{2.5}$$

If we set, for instance, an initial condition $u_0 = 1$, and a relatively small time step of $\delta t = 0.5$ compared with 100, the exact and numerical solutions give, respectively, $u(0.5) = \mathrm{e}^{-50} \approx 1.9 \times 10^{-22}$ and $u_1 = -49$. And integrating over another time step δt: $u(1) = \mathrm{e}^{-100} \approx 3.7 \times 10^{-44}$ and $u_2 = 2401$. It follows then that the explicit time discretization given by (2.3) is not capable of reproducing the right dynamics given by the exact solution. However, since this solution models a rapid transition from u_0 towards a final equilibrium value, we can easily identify the associated time scale $\tau = 1/100 = 0.01$ of the transient phase and therefore, we can expect that integration time steps δt of the order or smaller than τ will be capable to track the right dynamics. For instance, for $\delta t = 0.001$, we have $u(0.001) = \mathrm{e}^{-0.1} \approx 0.904837418$ and $u_1 = 0.9$, and $u(0.002) = \mathrm{e}^{-0.2} \approx 0.818730753$ and $u_2 = 0.81$. These rapid variations or transients associated with fast scales are typical of stiff equations, but they are neither sufficient nor necessary to qualify them as stiff. Actually, an initial condition u_0 close enough to the equilibrium manifold of the solution will not develop such fast transients, and thus stiff features may not be observed.
As a first conclusion, we can deduce that an explicit time discretization scheme to solve (2.4) will generally fail to approach the right dynamics, unless we consider integration time steps

smaller than the time scales disclosed by the equations. This may seem natural. Nevertheless, if we consider the counter-part of (2.3), *i.e.,* an *implicit Euler* method, also of order 1:

$$u_1 = u_0 + \delta t f(t_1, u_1), \tag{2.6}$$

and the previous $\delta t = 0.5$, we obtain the numerical approximations $u_1 = 0.019607843$ and $u_2 = 0.000384468$. Therefore, although solutions are not quite accurate, they show convergence towards the right solution with a time step several times the associated time scale. As a second conclusion, we can then add that both explicit and implicit schemes are of the same order, and would therefore yield results of the same accuracy for sufficiently small time steps. From a time step larger than a given value, the explicit method will not deliver any valid result.

2.1.1 Some Typical Stiff Configurations

If we now consider a general nonlinear system

$$\mathrm{d}_t U = F(U) \tag{2.7}$$

with $U : \mathbb{R} \to \mathbb{R}^m$, $F : \mathbb{R}^m \to \mathbb{R}^m$ and define a solution $\varphi(t) \in \mathbb{R}^m$ such that $\mathrm{d}_t \varphi(t) = F(\varphi(t))$, we can linearize F in its neighborhood:

$$\mathrm{d}_t U = F(\varphi(t)) + \partial_U F(\varphi(t))\,(U(t) - \varphi(t)) + \mathcal{O}\left((U(t) - \varphi(t))^2\right), \tag{2.8}$$

to obtain

$$\mathrm{d}_t \overline{U} = J\overline{U}, \tag{2.9}$$

where higher order terms in $\overline{U}(t) := U(t) - \varphi(t)$ are neglected, and with the Jacobian: $J(U) = \partial_U F(U)$. Supposing a constant Jacobian that is moreover diagonalizable, we can write the i-th component $\overline{u}^{(i)}(t)$ of $\overline{U}(t)$, solution of (2.9), as

$$\overline{u}^{(i)}(t) = \sum_{i=1}^{m} c_i \mathrm{e}^{\lambda_i t} \overline{u}_0^{(i)}, \tag{2.10}$$

for some initial condition $\overline{U}_0 \in \mathbb{R}^m$ and constants c_i, where the λ_i are the corresponding eigenvalues associated with J. Therefore, we can see that the solutions $\overline{u}^{(i)}(t)$ of (2.9) are clearly reproduced by a linear combination of

$$\left(\mathrm{e}^{\lambda_i t} \overline{u}_0^{(i)}\right)_{i=1,2,\ldots,m}, \tag{2.11}$$

that is, solutions of the same type as for the previous linear problem (2.4), and thus the latter simpler case mimics somehow the dynamics of more general nonlinear problems. We can then expect the same behavior previously described for explicit and implicit schemes, depending in this case on the spectrum of the Jacobian J and the set of initial conditions $\overline{u}_0^{(i)}$, $i = 1, 2, \ldots, m$. As a consequence, if (2.7) is a stiff system of ODEs, then it is very likely that some λ_i with large negative real part $\mathrm{Re}\,\lambda_i \leq 0$, 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. In particular, not only large eigenvalues will generate the fast variations previously discussed, but also an important dispersion of the eigenvalues in the spectrum of J will certainly induce multi-scale dynamics issued from the composition of the various time scales (or eigenvalues) included in (2.10). This is a typical situation for example in the context of chemical reaction systems modeling a set of reactions with different reaction scales,

and hence time scales for which fast projection of some species onto equilibrium manifolds are usually developed (see, *e.g.*, [MP92]). These systems are usually very stiff and moreover, the stiffness increases with the precision and the detail of description of the mathematical model. Another classical example of a stiff problem, where stiffness is not necessarily related to the presence of fast variables, is given by the system:

$$\mathrm{d}_t U = AU, \tag{2.12}$$

with $U : \mathbb{R} \to \mathbb{R}^m$, $A \in \mathcal{M}_m(\mathbb{R})$:

$$A = \frac{1}{\Delta x^2} \begin{pmatrix} -1 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -1 \end{pmatrix} \tag{2.13}$$

and $\Delta x = 1/(N_x + 1)$, issued from the spatial discretization on a grid of $N_x = m$ points with second order centered finite differences for the heat equation:

$$\partial_t u - \partial_x^2 u = 0, \tag{2.14}$$

for $x \in [0, 1]$ and homogeneous Neumann conditions.
As previously seen, the solution of (2.14) in infinite dimension is given by

$$u(x, t) = \mathrm{e}^{t\partial_x^2} u_0(x), \tag{2.15}$$

for some initial condition $u_0(x)$, where the associated spectrum of the differential operator is given by the whole set of numbers in the negative real axis. Furthermore, performing a Fourier transform in the x direction

$$\hat{f}(k) = \mathcal{F}(f) := \int_{-\infty}^{\infty} \mathrm{e}^{-ikx} f(x)\, \mathrm{d}x, \tag{2.16}$$

of the heat equation (2.14) yields as solution:

$$\hat{u}(k, t) = \mathrm{e}^{-k^2 t} \hat{u}_0(k). \tag{2.17}$$

Hence, a strong damping of the highest frequencies given by the frequency parameter[1] k will arise and will smoothen the initial condition. This is typical of diffusion problems. The analogy with the previous linear case (2.4) can be then established by this analysis for which in particular, we see that the frequency composition of the initial condition u_0, will or will not activate these fast decays, similar to (2.4). As a consequence, we can directly relate the stiffness associated with equation (2.14) to the presence of high gradients or discontinuities in x in $u_0(x)$. For instance, if we consider an extreme case for which $u_0(x) = \delta(x)$, *i.e.*, the *Dirac delta function*, all the frequency spectrum will appear on (2.17) with fast decays, since $\hat{u}_0(k) = 1$.
Coming back to the discretized problem (2.12) which is the one that will be numerically integrated, we can infer that the discretized counter-part mimics the previous theoretical analysis. This is reflected, for instance, by the spectrum of the matrix A:

$$\lambda_j = -\frac{4}{\Delta x^2} \sin^2\left(\frac{\pi j \Delta x}{2}\right), \quad j = 1, \ldots, N_x, \tag{2.18}$$

[1] Also referred to as the wave number.

or alternatively,

$$\lambda_j = -4(N_x+1)^2 \sin^2\left(\frac{\pi j}{2(N_x+1)}\right), \quad j = 1, \ldots, N_x, \tag{2.19}$$

for which we can identify potentially large eigenvalues increasing quadratically with the number of discretization points N_x with a maximum dispersion between $-4(N_x+1)^2$ and 0, which explains the spurious patterns found in some numerical approximations for this kind of problem (see some illustrations in[HW96]). In particular, we see that finer discretizations that introduce naturally more resolution scales, result also in broader spectra to represent them. Once again, these large eigenvalues will arise in the global solution depending on the distribution of the initial conditions. Stiffer behavior will then take place for discontinuous or large variations within the initial distributions.
With this brief introduction and illustrations, we introduce in the following the so-called Runge-Kutta (RK) time integrations method, as well as some dedicated RK schemes conceived to handle stiff systems of ODEs.

2.2 Runge-Kutta Time Integration Methods

We have previously considered the explicit Euler method given by (2.3). This kind of method is called *one-step integration method* because we aim at recursively approximating the exact solution (2.2) after one time step, based on the previous one. The Euler scheme is of first order but by approximating the integral in (2.2) by a higher order quadrature formula, we can build higher order one-step methods. A second order scheme can be constructed, for instance, by using the mid-point approximation:

$$u_1 = u_0 + \delta t f\left(t_0 + \frac{\delta t}{2}, u\left(t_0 + \frac{\delta t}{2}\right)\right), \tag{2.20}$$

and the Euler method, which leads to the *Runge method*:

$$u_1 = u_0 + \delta t f\left(t_0 + \frac{\delta t}{2}, u_0 + \frac{\delta t}{2} f(u_0)\right). \tag{2.21}$$

Generalizing this idea with higher order quadrature formulae leads to define the so-called s-stage *Runge-Kutta methods*:

$$\left.\begin{aligned} g_i &= u_0 + \delta t \sum_{j=1}^{s} a_{ij} f\left(t_0 + c_j \delta t, g_j\right), \quad i = 1, \ldots, s; \\ u_1 &= u_0 + \delta t \sum_{j=1}^{s} b_j f\left(t_0 + c_j \delta t, g_j\right), \end{aligned}\right\} \tag{2.22}$$

for which the arrays b, $c \in \mathbb{R}^s$ gather the various coefficients $b = (b_1, \ldots, b_s)^T$ and $c = (c_1, \ldots, c_s)^T$, and $A \in \mathcal{M}_s(\mathbb{R})$ such that $A = (a_{ij})_{1 \le i,j \le s}$. These coefficients are usually arranged in a mnemonic device, known as a Butcher tableau:

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

For instance, for the Runge method (2.21), we have

$$\begin{array}{c|cc} 0 & & \\ \frac{1}{2} & \frac{1}{2} & \\ \hline & 0 & 1 \end{array}$$

When $a_{ij} = 0$ for $j \geq i$, the scheme is explicit in time (*Explicit RK methods*, ERK) with

$$g_i = u_0 + \delta t \sum_{j=1}^{i-1} a_{ij} f\left(t_0 + c_j \delta t, g_j\right), \quad i = 1, \ldots, s, \tag{2.23}$$

in (2.22), whereas the case for which $a_{ij} = 0$ for $j > i$ and at least one of the diagonal coefficients is non-zero, $a_{ii} \neq 0$, is defined as a *Diagonal Implicit RK method* (DIRK). Otherwise, we are considering *Implicit RK methods* (IRK). We will further describe these schemes in the following, but first, we will introduce some basic theoretical and numerical properties for general RK methods: the order and stability features, as well as the choice of the time steps of integration.

2.2.1 Order and Stability of Runge-Kutta Schemes

We now consider the *Dahlquist test equation* [Dah63]:

$$\left.\begin{array}{l} \mathrm{d}_t u = \lambda u, \\ u(0) = 1, \end{array}\right\} \tag{2.24}$$

with $\lambda \in \mathbb{C}$ (a particular case was given by (2.4)), and we can successively compute the g_j of the explicit RK method (2.23) for problem (2.24). We obtain

$$u_1 = R(z) u_0, \quad z = \delta t \lambda, \tag{2.25}$$

where

$$R(z) = 1 + z \sum_j b_j + z^2 \sum_{j,k} b_j a_{j,k} + \ldots, \tag{2.26}$$

is a polynomial of degree $\leq s$. If the RK method is of order p we know that $u_1 = R(z) u_0$ must satisfy

$$\mathrm{e}^z - R(z) = \mathcal{O}(\delta t^{p+1}) = \mathcal{O}(z^{p+1}), \tag{2.27}$$

where $\mathrm{e}^z u_0$ is the exact solution of (2.24), and thus $R(z)$ is given by

$$R(z) = 1 + z + \frac{z^2}{2!} + \ldots + \frac{z^p}{p!} + \mathcal{O}(z^{p+1}). \tag{2.28}$$

In particular, for all explicit RK methods of order p with $s = p$ intermediate stages, we have

$$R(z) = 1 + z + \frac{z^2}{2!} + \ldots + \frac{z^s}{s!}. \tag{2.29}$$

A classical analysis based on the Dahlquist test equation (2.24) allows us to define $R : \mathbb{C} \to \mathbb{C}$ given in general by (2.25), as the *stability function* of a given method. That is, $R(z)$ is the numerical solution of (2.24) given by the method itself after one time step δt. Furthermore, the numerical solution recursively computed can be written as

$$u_n = \left(R(z)\right)^n u_0 \tag{2.30}$$

which allows us to define the *stability domain* of the method given by the set of z for which u_n remains bounded for $n \to \infty$, *i.e.*,

$$S := \{z \in \mathbb{C} \text{ s.t. } |R(z)| \leq 1\}. \tag{2.31}$$

For instance, considering the explicit Euler method (2.3) for which

$$R(z) = 1 + z, \tag{2.32}$$

according to (2.28), its stability domain S is given by all $z \in \mathbb{C}$ such that

$$|1 + z| = |z - (-1)| \leq 1, \tag{2.33}$$

which is the circle of radius 1 and center -1 in the complex plane. Coming back to the previous example (2.4) with $\lambda = -100$, we can see that an explicit Euler method will remain stable as long as $z = \delta t \lambda \in S$, *i.e.*, $0 \leq \delta t \leq 2/100$, which explains the previous bad results for $\delta t = 0.5$. Alternatively, considering the implicit Euler method (2.6) yields

$$R(z) = \frac{1}{1 - z}, \tag{2.34}$$

as stability function, with stability domain given by all $z \in \mathbb{C}$ such that

$$\left|\frac{1}{1 - z}\right| \leq 1 \;\Rightarrow\; |z - 1| \geq 1, \tag{2.35}$$

that is, the exterior of the circle with radius 1 and center $+1$ in the complex plane. For problem (2.4), we can then see that $R(z)$ will remain bounded for any time step $\delta t > 0$, as it is shown by $(R(z = -100\,\delta t))^n = (1 + 100\,\delta t)^{-n}$ into (2.30). This better performance of an implicit discretization for large negative λ into (2.24), characteristic of stiff ODEs, leads us to give more details on these schemes in a forthcoming section. In particular, it was demonstrated that for $p \geq 5$ there is no explicit RK method of order p with $s = p$ stages [But64c, But64d]. This and other order constraints for explicit RK schemes are known as the *Butcher Barriers* (see more details in [HNW87]). Finally, it is important to recall that in a general case, we can perform the same analysis on the linearized problem (2.9), similar to the Dahlquist test equation, taking into account the complex eigenvalues $\lambda_i, i = 1, \cdots, m$, of the associated Jacobian J.

2.2.2 Time Step Selection

Whether the time discretization schemes are explicit or implicit, or if the orders of approximations are high or low, a key question for a numerical time integration method is the choice of the time step of integration. We have seen, for instance, that for stiff problems, explicit methods should consider rather small time steps to guarantee the stability of computations. However, for a given problem if we suppose that we are only considering time steps contained in the stability domain, the former ones must be chosen such that the numerical solutions yield approximations within a desired accuracy. In this case, a constant time step might be sufficient for some kind of problem to efficiently solve the corresponding dynamics. In a more general context, more sophisticated techniques must be consider to dynamically select these time steps in order to render computations efficient or even possible in practice. In any of both cases, the main goal is to choose a time step δt such that the local error verifies

$$\|u(\delta t) - u_1\| = C \delta t^{p+1} \leq Tol, \tag{2.36}$$

where Tol is the desired accuracy requested to the numerical computations. It is straightforward to see that higher order methods would satisfy (2.36) with larger time steps. Furthermore, for a given scheme the expression (2.36) might be satisfied with time steps evolving in time. For problems describing different dynamics, having an adaptive time step strategy would then involve important savings of numerical work. In this context, a lot of research has been conducted to develop time step control or adaptive time stepping techniques. A review of some explicit solvers with automatic time step selection can be found in [HNW87] for non stiff problems. A complementary idea developed for explicit schemes was to use these control techniques to automatically detect stiffness (see, *e.g.,* [Sha77, SH77, HW96]) in order to automatically switch to a more suitable method.
One of the most standard ways of time stepping is based on computing a numerical approximation: err, of the exact local error in (2.36), by considering a solution $\hat{u}_1$ computed by a lower order method of order $\hat{p} < p$ [HNW87], such that

$$\|u(\delta t) - u_1\| \lessapprox err = \|u_1 - \hat{u}_1\|. \tag{2.37}$$

Since

$$u_1 - \hat{u}_1 = (u_1 - u(\delta t)) - (\hat{u}_1 - u(\delta t)) = \mathcal{O}(\delta t^{p+1}) + \mathcal{O}(\delta t^{\hat{p}+1}) \approx \mathcal{O}(\delta t^{\hat{p}+1}), \tag{2.38}$$

and thus,

$$err \approx \tilde{C}\delta t^{\hat{p}+1}, \tag{2.39}$$

we can suppose that the optimal time step δt_{opt} such that $err \approx Tol$:

$$Tol \approx \tilde{C}\delta t_{\text{opt}}^{\hat{p}+1}, \tag{2.40}$$

is given by

$$\delta t_{\text{opt}} = fac \cdot \delta t \left(\frac{Tol}{err}\right)^{1/\hat{p}+1}, \tag{2.41}$$

where fac is a safety factor usually close to 1.
In this way, we can compute the time step needed to integrate problem (2.1) with a local accuracy given by Tol, where the $\hat{p}$-order method should be embedded into the p-order method in order to minimize the required number of operations. Additionally, we can use the expression (2.41) to dynamically compute the time steps in time. In this case, we use the computations at the n-th step to predict the error at the next step:

$$err_{n+1} = \|u_n - \hat{u}_n\| \approx \tilde{C}_n \delta t_n^{\hat{p}+1}, \tag{2.42}$$

which yields as new time step:

$$\delta t_{\text{new}} = fac \cdot \delta t_n \left(\frac{Tol}{err_{n+1}}\right)^{1/\hat{p}+1}, \tag{2.43}$$

by assuming $\tilde{C}_{n+1} \approx \tilde{C}_n$ into

$$Tol \approx \tilde{C}_{n+1}\delta t_{\text{new}}^{\hat{p}+1}. \tag{2.44}$$

The next step δt_{n+1} will be then given by δt_{new} if $err_{n+1} \leq Tol$. Alternatively, the current n-th time step will be rejected if $err_{n+1} > Tol$, and in this case the procedure works as an *a posteriori* verification where the same n-th step will be integrated again with the new time step δt_{new}.

Based on the same ideas and on more rigorous theoretical studies carried out by Gustafsson [Gus94], a better procedure assumes that $\log C_n$ is a linear function of n, and thus $\log C_{n+1} - \log C_n$ is constant or, equivalently [HW96]

$$\frac{C_{n+1}}{C_n} \approx \frac{C_n}{C_{n-1}}, \tag{2.45}$$

which finally yields

$$\delta t_{\text{new}} = fac \cdot \delta t_n \left(\frac{Tol}{err_{n+1}}\right)^{1/\hat{p}+1} \frac{\delta t_n}{\delta t_{n-1}} \left(\frac{err_n}{err_{n+1}}\right)^{1/\hat{p}+1}. \tag{2.46}$$

This technique is also known as the *step size strategy with memory* of Watts [Wat84] and Gustafsson [Gus94], and usually shows better performances than the standard technique (2.43). In particular, it allows us fast reduction of time steps without rejection in the context of stiff problems [HW96]. There are other step size control techniques to numerically estimate or predict local errors and therefore, to guarantee a given accuracy of computations according to (2.36). We mention, for instance, time step computations using extrapolation techniques [Deu83, Sha87], or theoretical or numerical estimates of the leading term of the local error expansion [Hin80, SSV97].

2.3 Implicit Runge-Kutta Methods

Let us consider now the implicit RK scheme (2.22). We apply it to the Dahlquist test equation (2.24), and we obtain

$$\left.\begin{aligned} g &= u_0 \mathbb{1} + \delta t \lambda A g, \\ u_1 &= u_0 + \delta t \lambda b^T g, \end{aligned}\right\} \tag{2.47}$$

with $g = (g_1, \ldots, g_s)^T$ and $\mathbb{1} = (1, \ldots, 1)^T$. The linear system for $g_1, \ldots, g_s$ gives

$$g = (\mathrm{Id} - \lambda \delta t A)^{-1} u_0 \mathbb{1}, \tag{2.48}$$

and the corresponding stability function may be written as

$$R(z) = 1 + z b^T (\mathrm{Id} - zA)^{-1} \mathbb{1}. \tag{2.49}$$

However, a better representation might be obtained by considering the solution of (2.47):

$$\begin{pmatrix} \mathrm{Id} - zA & 0 \\ -zb^T & 1 \end{pmatrix} \begin{pmatrix} g \\ u_1 \end{pmatrix} = u_0 \begin{pmatrix} \mathbb{1} \\ 1 \end{pmatrix}, \tag{2.50}$$

using the Cramer's rule:

$$u_1 = \frac{\det \begin{pmatrix} \mathrm{Id} - zA & u_0 \mathbb{1} \\ -zb^T & u_0 \end{pmatrix}}{\det \begin{pmatrix} \mathrm{Id} - zA & 0 \\ -zb^T & 1 \end{pmatrix}}, \tag{2.51}$$

and taking into account that

$$\det \begin{pmatrix} \mathrm{Id} - zA & \mathbb{1} \\ -zb^T & 1 \end{pmatrix} = \det \begin{pmatrix} \mathrm{Id} - zA + z\mathbb{1}b^T & 0 \\ -zb^T & 1 \end{pmatrix} = \det \left(\mathrm{Id} - zA + z\mathbb{1}b^T\right). \tag{2.52}$$

This yields

$$R(z) = \frac{P(z)}{Q(z)} = \frac{\det\left(\mathrm{Id} - zA + z\mathbb{1}b^T\right)}{\det(\mathrm{Id} - zA)}, \tag{2.53}$$

so we can see that for implicit RK schemes, the stability function $R(z)$ becomes a rational function with polynomial numerator $P(z)$ and denominator $Q(z)$ of degree less than or equal to s.

A direct consequence of this rational stability function as seen for the implicit Euler method (2.6), is that the associated schemes can be stable on the entire left-half plane $\mathbb{C}^-$. This set of z corresponds precisely to eigenvalues of negative real part for which the exact solutions are bounded in time $|e^z| \leq 1$ and for which we have seen before, the numerical method should preserve this stability property.

A method is then called *A-stable* if its stability domain satisfies [Dah63]

$$S \supset \{z \in \mathbb{C} \text{ s.t. } \mathrm{Re}\, z \leq 0\}. \tag{2.54}$$

For instance, the implicit Euler method (2.6) is A-stable. Even though this is a desirable and necessary stability property to properly handle stiff problems, it is not sufficient for very stiff problems. For eigenvalues with very large real part, the stability function $R(z)$ of an A-stable method will surely keep the numerical approximations bounded during the fast transients. Nevertheless, only a $R(z)$ much smaller that 1, can guarantee that the numerical solutions will rapidly approach the exact solution, damping out the numerical transients phases. Numerical methods with such a property are known as *L-stable* [Ehl69].

Taking into account that for rational functions

$$\lim_{z\to\infty} R(z) = \lim_{z\to-\infty} R(z), \tag{2.55}$$

a method is called L-stable if it is A-stable and if in addition

$$\lim_{z\to\infty} R(z) = 0. \tag{2.56}$$

Considering that for an implicit RK method we have that

$$R(\infty) = 1 - b^T A^{-1}\mathbb{1}, \tag{2.57}$$

according to (2.49), it follows that if an A-stable implicit RK method with nonsingular A satisfies one of the following conditions:

$$a_{sj} = b_j, \quad j = 1, \ldots, s; \tag{2.58}$$

$$a_{i1} = b_1, \quad i = 1, \ldots, s, \tag{2.59}$$

then $R(\infty) = 0$ in (2.57), and the method is also L-stable. In particular, methods satisfying (2.58) are called *stiffly accurate* [PR74] and are particularly important for the solution of singular perturbation problems and for differential-algebraic equations [HW96].

Finally, there are some implicit schemes with large stability domains that are not A-stable. In order to characterize these methods, $A(\alpha)$-*stability* constitutes another stability property for which a method is said to be $A(\alpha)$-stable if a sector α is contained in the stability region [Wid67]:

$$S_\alpha = \{z \in \mathbb{C} \text{ s.t. } |\arg(-z)| < \alpha,\ z \neq 0\}. \tag{2.60}$$

In this work, we consider only one-step integration methods. Nevertheless, dedicated *multi-step integration methods* for the resolution of stiff problems were also developed. These schemes

consider several time steps in order to reconstruct the numerical solution that satisfies the differential equations at each considered time step. Moreover, the *Second Dahlquist Barrier* states that an A-stable multi-step method must be of order $p \leq 2$ [Dah63]. Nevertheless, there are many multi-step schemes performing good $A(\alpha)$-stability properties for high orders, and L-stability for lower ones, which can be efficiently used to solve stiff problems. Some examples are the *LSODE* [Hin80, Hin83] (Livermore Solver for ODEs) or the *VODE* solver [BBH89] (Variable-coefficient ODE solver), both based on a variable-order (up to fifth) *Backward Differentiation Formulae* developed by Gear [Gea71] (see [HW96] for more details on dedicated multi-step methods for stiff problems).

2.3.1 Construction of Implicit Runge-Kutta Methods

As previously detailed for the explicit case, an implicit RK method is of order p if condition (2.27) is satisfied, in which case we see that $R(z)$ is this time a rational approximation to e^z according to (2.53). In this context, the construction of fully implicit RK methods relies heavily on the following conditions [HW96]:

$$\left.\begin{array}{lll} B(p): & \displaystyle\sum_{i=1}^{s} b_i c_i^{q-1} = \frac{1}{q}, & q = 1, \ldots, p; \\ C(\eta): & \displaystyle\sum_{j=1}^{s} a_{ij} c_j^{q-1} = \frac{c_i^q}{q}, & i = 1, \ldots, s, \quad q = 1, \ldots, \eta; \\ D(\zeta): & \displaystyle\sum_{i=1}^{s} b_i c_i^{q-1} a_{ij} = \frac{b_j}{q}(1 - c_j^q), & j = 1, \ldots, s, \quad q = 1, \ldots, \zeta. \end{array}\right\} \tag{2.61}$$

The first condition $B(p)$ states that the quadrature formula $(b_i, c_i)_{i=1}^s$ is of order p, whereas it was proved by Butcher [But64a] that if the coefficients b_i, c_i, a_{ij} of a RK method satisfy $B(p)$, $C(\eta)$, $D(\zeta)$ with $p \leq \eta + \zeta + 1$ and $p \leq 2\eta + 2$, then the method is of order p.
With these tools, one way of building these RK schemes considers *collocation methods* based on quadrature formulae. The main goal is to find a polynomial $p(t)$ of degree s such that $p(t_n) = u_n$, and that for a set of *collocation points* $0 \leq c_1 < \ldots < c_s \leq 1$, it verifies

$$\mathrm{d}_t p(t_n + c_i \delta t) = f\left(p(t_n + c_i \delta t)\right), \quad i = 1, \ldots, s; \tag{2.62}$$

such that $u(t_{n+1}) = u(t_n + \delta t)$ will be approximated by $u_{n+1} = p(t_n + \delta t)$ [GS69, Wri71]. We can then determine the collocation points based on the quadrature formulae used to numerically approximate

$$\int_{t_0}^{t_0 + \delta t} f(t)\, \mathrm{d}t \approx \delta t \sum_{i=1}^{s} b_i f(t_0 + c_i \delta t). \tag{2.63}$$

If the quadrature method yields approximations of order p, an important mathematical result is that the collocation method will also yield approximations of order p for the differential problem (2.62) [GS69].
In this way, Butcher [But64b] introduced RK methods based on Radau quadrature formulae [Rad80], for which the collocation points $c_1, \ldots, c_s$, are the zeros of the polynomials

$$\mathrm{I}: \quad \mathrm{d}_x^{s-1}\left(x^s (x-1)^{s-1}\right), \tag{2.64}$$

$$\mathrm{II}: \quad \mathrm{d}_x^{s-1}\left(x^{s-1} (x-1)^{s}\right), \tag{2.65}$$

and the weights $b_1, \ldots, b_s$, are computed in order to verify $B(s)$ for the quadrature formula $(b_i, c_i)_{i=1}^s$ into (2.61). Finally, we have that $B(2s-1)$ since $p = 2s - 1$ for a Radau quadrature

formula. Both polynomials have positive zeros with $c_1 = 0$ and $c_i < 1$, $i = 2, \ldots, s$ for (2.64), and $c_i > 0$, $i = 1, \ldots, s-1$ and $c_s = 1$ for (2.65), whereas the remaining coefficients are computed based on the order conditions (2.61). These first schemes were not A-stable but based on these ideas, Ehle [Ehl69] constructed some A- and L-stable schemes which gave birth to the families of formulae called RadauIA and RadauIIA, depending on the used quadrature formula (2.64) or (2.65). Tables 2.1 and 2.2 show, respectively, the corresponding coefficients for RadauIA and RadauIIA of order $p = 5$ with $s = 3$ stages. L-stability can be retrieved in this case for $p = 5$, by verifying, respectively, conditions (2.59) and (2.58).

Table 2.1: *RadauIA method of order 5.*

0	$\frac{1}{9}$	$\frac{-1-\sqrt{6}}{18}$	$\frac{-1+\sqrt{6}}{18}$
$\frac{6-\sqrt{6}}{10}$	$\frac{1}{9}$	$\frac{88+7\sqrt{6}}{360}$	$\frac{88-43\sqrt{6}}{360}$
$\frac{6+\sqrt{6}}{10}$	$\frac{1}{9}$	$\frac{88+43\sqrt{6}}{360}$	$\frac{88-7\sqrt{6}}{360}$
	$\frac{1}{9}$	$\frac{16+\sqrt{6}}{36}$	$\frac{16-\sqrt{6}}{36}$

Table 2.2: *RadauIIA method of order 5.*

$\frac{4-\sqrt{6}}{10}$	$\frac{88-7\sqrt{6}}{360}$	$\frac{296-169\sqrt{6}}{1800}$	$\frac{-2+3\sqrt{6}}{225}$
$\frac{4+\sqrt{6}}{10}$	$\frac{296+169\sqrt{6}}{1800}$	$\frac{88+7\sqrt{6}}{360}$	$\frac{-2-3\sqrt{6}}{225}$
1	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$
	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$

Alternatively, other schemes were derived based on other quadrature formulae. For instance, a family of s-stage *Gauss methods* were constructed this time from Gaussian quadrature formulae, and perform A-stability properties with the maximum possible order: $p = 2s$ [But64a, Ehl68]. Nevertheless, these schemes are usually not L-stable. Another large group considers Lobatto quadrature formulae which yields some A- and L-stable schemes of order $p = 2s - 2$ [But64a, Ehl68, Chi71, Axe72]. In what follows, we will recall some of the previous concepts and give some insights into the practical implementation of these implicit RK methods by considering the *Radau5* solver developed by Hairer & Wanner [HW96].

2.3.2 The Radau5 Solver

Let us recall the general nonlinear problem (2.1), this time of dimension m, that is, $u_0 \in \mathbb{R}^m$, $u : \mathbb{R} \to \mathbb{R}^m$, and $f : \mathbb{R} \times \mathbb{R}^m \to \mathbb{R}^m$, to keep the previous notations:

$$\left.\begin{aligned} \mathrm{d}_t u &= f(t, u(t)), \\ u(0) &= u_0. \end{aligned}\right\} \tag{2.66}$$

The solution of this problem by a s-stage fully implicit RK method (2.22) will lead to the solution of a nonlinear system of equations of size $m \times s$ in order to determine the unknowns $g_1, \ldots, g_s$. In order to avoid solving these large systems, a family of diagonally implicit RK schemes called *SDIRK* (Singly Diagonally Implicit RK) were developed, that considers a less expensive alternative by solving s successive stages with only m-dimensional systems to be solved at each stage. Nevertheless, more stages than the previously seen for fully implicit RK schemes are usually needed to build A- or stiffly accurate L-stable methods, for instance, $p = s + 1$ or $p = s$. A further simplification considered the linearization of DIRK schemes in order to replace the nonlinear systems by a sequence of linear problems. These methods are usually called *linearly implicit RK methods* or simply *Rosenbrock methods*, and show good $A(\alpha)$-stability properties. A survey and analysis of these and other methods can be found in [HW96].

As a consequence, we can infer that an efficient solution of large nonlinear systems is mandatory for practical purposes and constitutes the main difficulty in the implementation of a fully implicit RK method [HW96]. In this context, Hairer & Wanner developed the Radau5 solver for which they had to introduce a few performing tools to overcome the many numerical difficulties associated with the practical implementation of implicit RK schemes. All of these issues are discussed in details in their book [HW96], but we will present here some of them that are usually common to various implicit RK solvers, for the sake of completeness of this work.

Radau5 implements the fifth order, 3-stage Ehle's method RadauIIA, given in Table 2.2. This is a high order, A- and L-stable scheme, very suitable for highly stiff problems. The solver considers RadauIIA because among other reasons, this is a stiffly accurate scheme given by condition (2.58). From a practical point of view and for very stiff problems such as singularly perturbed problems, condition (2.58) implies that the numerical solution becomes also an internal stage in the solution of the $g_1, \ldots, g_s$ ($c_3 = 1$ in Table 2.2). Therefore, we can expect that fast transients in the exact solution will be better reproduced by numerically considering the relaxed fast variables after one time step δt [HW96].

Considering the general implicit RK scheme (2.22), we define a new set o variables $z_1, \ldots, z_s$, for the computation of the $g_1, \ldots, g_s$:

$$z_i = g_i - u_0, \tag{2.67}$$

in order to reduce the influence of round-off errors [HW96]. This yields

$$\left.\begin{aligned} z_i &= \delta t \sum_{j=1}^{s} a_{ij} f(t_0 + c_j \delta t, u_0 + z_j), \qquad i = 1, \ldots, s; \\ u_1 &= u_0 + \delta t \sum_{j=1}^{s} b_j f(t_0 + c_j \delta t, u_0 + z_j). \end{aligned}\right\} \tag{2.68}$$

Therefore, knowing the solution $z_1, \ldots, z_s$ implies an explicit formula for u_1, for which s additional function evaluations are required. These extra computation can nevertheless be avoided

if the matrix $A = (a_{ij})$ is nonsingular, which is the case for RadauIIA. Actually, considering that

$$\begin{pmatrix} z_1 \\ \vdots \\ z_s \end{pmatrix} = A \begin{pmatrix} \delta t f(t_0 + c_1 \delta t, u_0 + z_1) \\ \vdots \\ \delta t f(t_0 + c_s \delta t, u_0 + z_s) \end{pmatrix}, \tag{2.69}$$

the computation of u_1 is equivalent to

$$u_1 = u_0 + \sum_{i=1}^{s} d_i z_i, \tag{2.70}$$

where

$$(d_1, \ldots, d_s) = (b_1, \ldots, b_s) A^{-1}. \tag{2.71}$$

Taking into account the coefficients in Table 2.2, we see that for RadauIIA: $d = (0, 0, 1)$, since $b_i = a_{si}$ for all i according to (2.58).
To solve the nonlinear system (2.69), Radau5 considers an iterative Newton's method. This amounts to solve at each iteration a linear system with the matrix:

$$\begin{pmatrix} \mathrm{Id} - \delta t a_{11} \partial_u f(t_0 + c_1 \delta t, u_0 + z_1) & \ldots & -\delta t a_{1s} \partial_u f(t_0 + c_s \delta t, u_0 + z_s) \\ \vdots & \ddots & \vdots \\ -\delta t a_{s1} \partial_u f(t_0 + c_1 \delta t, u_0 + z_1) & \ldots & \mathrm{Id} - \delta t a_{ss} \partial_u f(t_0 + c_s \delta t, u_0 + z_s) \end{pmatrix}. \tag{2.72}$$

If we approximate all Jacobians $\partial_u f(t_0 + c_i \delta t, u_0 + z_i)$ by

$$J \approx \partial_u f(t_0, u_0), \tag{2.73}$$

we consider a simplified Newton's method for

$$G(Z) = Z - (\mathrm{Id} - \delta t A \otimes J) F(Z) = 0, \tag{2.74}$$

where $Z = (z_1, \ldots, z_s)^T$, and $F(Z) = (f(u_0 + c_1 \delta t, u_0 + z_1), \ldots, f(t_0 + c_s \delta t, u_0 + z_s))^T$, so that the $(k+1)$-th approximation of the solution Z is recursively computed by

$$\left.\begin{aligned} (\mathrm{Id} - \delta t A \otimes J) \Delta Z^k &= -Z^k + \delta t (A \otimes \mathrm{Id}) F(Z^k), \\ Z^{k+1} &= Z^k + \Delta Z^k. \end{aligned}\right\} \tag{2.75}$$

Each iteration requires then s evaluations of f to compute $F(Z^k)$, and the solution of a $m \times s$ linear system to compute the increments $\Delta Z^k = (\Delta z_1^k, \ldots, \Delta z_s^k)^T$. Fortunately, the matrix $(\mathrm{Id} - \delta t A \otimes J)$ is the same for all iterations with the approximated Jacobians (2.73), and its inversion by an LU-decomposition, usually quite expensive, is done only once. Furthermore, exploiting the special structure of the matrix $(\mathrm{Id} - \delta t A \otimes J)$, a decomposition of the linear system into two subsystems following a procedure introduced by Butcher [But76], leads to an important reduction of the number of operations, which is also implemented in the Radau5 solver [HW96]. If no analytical expression is available, the Jacobians can always be numerically approximated by

$$J_{ij} \approx \frac{f^{(i)}(t_0, u^{(j)} + \delta u^{(j)}) - f^{(i)}(t_0, u^{(j)})}{\delta u^{(j)}}, \qquad i, j = 1, \ldots, m, \tag{2.76}$$

for relatively small, positive perturbations: $\delta u = (\delta u^{(1)}, \ldots, \delta u^{(m)})$. Finally, Hairer & Wanner defined also dedicated stopping criteria for the iterative method as well as appropriate starting values Z^0 for the Newton iterations [HW96].

In order to select the time step and guarantee a prescribed accuracy, Radau5 uses a lower order embedded method to numerically estimate the local error in the same spirit of section § 2.2.2. We illustrate this procedure for this particular case. A lower order approximation of the solution $\hat{u}_1$ according to (2.37) is computed by

$$\hat{u}_1 = u_0 + \delta t \hat{b}_0 f(t_0, y_0) + \delta t \sum_{i=1}^{3} \hat{b}_i f(t_0 + c_i \delta t, g_i), \tag{2.77}$$

using the same collocation points c_1, c_2, c_3 of RadauIIA (see Table 2.2), and thus the same evaluations of f. An extra evaluation of f is needed at t_0, whereas $\hat{b}_0 = \hat{\gamma}_0$, where $\hat{\gamma}_0^{-1}$ is a real eigenvalue of A^{-1} previously computed. In order to set the new weights $\hat{b}_1$, $\hat{b}_2$, $\hat{b}_3$ we consider the difference:

$$\hat{u}_1 - u_1 = \delta t \hat{\gamma}_0 f(t_0, y_0) + \delta t \sum_{i=1}^{3} (\hat{b}_i - b_i) f(t_0 + c_i \delta t, g_i), \tag{2.78}$$

into (2.61) for $B(3)$ such that $\hat{u}_1 - u_1 = \mathcal{O}(\delta t^4)$. Considering the representation (2.70), this yields finally

$$\hat{u}_1 - u_1 = \delta t \hat{\gamma}_0 f(t_0, y_0) + \sum_{i=1}^{3} \hat{d}_i z_3, \tag{2.79}$$

where

$$(\hat{d}_1, \hat{d}_2, \hat{d}_3) = \frac{\hat{\gamma}_0}{3}(-13 - 7\sqrt{6}, -13 + 7\sqrt{6}, -1). \tag{2.80}$$

With these solutions, Radau5 computes the approximation:

$$err = (\mathrm{Id} - \delta t \hat{\gamma}_0 J)^{-1} (\hat{u}_1 - u_1), \tag{2.81}$$

as error estimate in order to simultaneously guarantee that the difference (2.79) is bounded for $\delta t \to 0$ and $\delta t \lambda \to \infty$ (if $f(u) = \lambda u$ and $J = \lambda$), for stiff problems [HW96].
The time steps are then computed by taking the minimum of

$$\delta t_{\text{new}} = fac \cdot \delta t_n \left(\frac{1}{\|err_{n+1}\|} \right)^{1/4}, \tag{2.82}$$

and

$$\delta t_{\text{new}} = fac \cdot \delta t_n \left(\frac{1}{\|err_{n+1}\|} \right)^{1/4} \frac{\delta t_n}{\delta t_{n-1}} \left(\frac{\|err_n\|}{\|err_{n+1}\|} \right)^{1/4}, \tag{2.83}$$

based, respectively, on (2.43) and (2.46) with

$$\|err\| = \sqrt{\frac{1}{m} \sum_{i=1}^{m} \left(\frac{err^{(i)}}{sc_i} \right)^2}, \tag{2.84}$$

with $err^{(i)} = (\mathrm{Id} - \delta t \hat{\gamma}_0 J)^{-1} \left(\hat{u}_1^{(i)} - u_1^{(i)} \right)$, and $sc_i = Atol_i + \max(|u_0^{(i)}|, |u_1^{(i)}|) \cdot Rtol_i$, where $Atol$ and $Rtol$ are defined as absolute and relative accuracy tolerances [HW96]. With the definition of the error estimate given by (2.84), the current time step is accepted if $\|err\| \leq 1$, otherwise it is rejected. In this case as well as for the first step, Radau5 uses a second error estimate instead of (2.81):

$$\widetilde{err} = (\mathrm{Id} - \delta t \hat{\gamma}_0 J)^{-1} \left(\delta t \hat{\gamma}_0 f(t_0, y_0 + err) + \sum_{i=1}^{3} \hat{d}_i z_3 \right), \tag{2.85}$$

which implies an additional evaluation of f, but we have that $\widetilde{err} \to 0$ is satisfied for $\delta t \lambda \to \infty$, in the same way as the numerical solution u_1 does.

2.4 Stabilized Explicit Runge-Kutta Methods

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 heat equation (2.14) in § 2.1.1, for which the real negative eigenvalues (2.18) increase with finer spatial discretizations. Therefore, instead of A-stable but time consuming implicit procedures, *stabilized explicit RK methods* should be preferred. These explicit methods avoid the solution of algebraic systems, while featuring an extended stability domain along the negative real axis, very appropriate for this type of problem. A detailed survey on these schemes can be found in [Ver96], and in the book of Hundsdorfer & Verwer [HV03].
The main goal is to construct methods of order p with a family of stability polynomial R_s of degree s:

$$R_s(z) = 1 + z + \cdots + \frac{z^p}{p!} + \sum_{p+1}^{s} \alpha_{i,s} z^i, \tag{2.86}$$

with $s \geq p+1$, and $\alpha_{i,s} \in \mathbb{C}$, such that $R_s(z)$ remains bounded as much as possible along the real negative axis, *i.e.*,

$$|R_s(z)| \leq 1, \qquad z \in [-\ell_s, 0], \tag{2.87}$$

with ℓ_s as large as possible. One way of building such stability polynomials considers the family of Chebyshev polynomials:

$$T_s(\cos(z)) = \cos(s\, z), \tag{2.88}$$

defined also by the recurrence relation:

$$T_0(z) = 1, \quad T_1(z) = z, \quad T_s(z) = 2zT_{s-1}(z) - T_{s-2}(z), \tag{2.89}$$

which remain bounded between 1 and -1 for $z \in [-1, 1]$, and in particular yield boundaries ℓ_s proportional to s^2.
These schemes are usually called *Runge-Kutta-Chebyshev* methods, and feature extended real stability intervals proportional to s^2, a good property inherited from Chebyshev-type polynomials. For instance, for $p = 1$, the optimal polynomials that satisfies (2.86) are directly the shifted Chebyshev polynomials:

$$R_s(z) = T_s\left(1 + \frac{z}{s^2}\right), \tag{2.90}$$

which are shown to yield the optimal $\ell_s = 2s^2$. However, in the points where $R_s(z) = \pm 1$ for $z \in \mathbb{R}^-$, the stability domain has zero width and therefore, there is no damping at all of high frequencies. The standard way to overcome this difficulty considers a small parameter $\varepsilon > 0$ in order to build *damped Chebyshev stability functions* [GL61]:

$$R_s(z) = \frac{1}{T_s(w)} T_s(w_0 + w_1 z), \quad w_0 = 1 + \frac{\varepsilon}{s^2}, \quad w_1 = \frac{T_s(w_0)}{T_s'(w_0)}. \tag{2.91}$$

As a consequence, the stability domains are reduced by approximatively ε: $|R_s(z)| \leq 1 - \varepsilon$, while the stability length is shortened by approximatively $(4\varepsilon/3)s^2$; nevertheless, the order of the scheme is preserved and a safe distance from the real axis is guaranteed [HW96].
Based on these ideas, a first family of method called *Lebedev-type methods* [Leb89, Leb94], aims at building RK schemes based on the optimal stability polynomials that satisfy (2.86) for a

given p. For $p = 1$ we have seen that these polynomials are the shifted Chebyshev polynomials (2.90), so the idea is to write them as [Sau60, GL61]:

$$R_s(z) = \prod_{i=1}^{s}(1 + \delta_i z), \quad \delta_i = -\frac{1}{z_i}, \tag{2.92}$$

where z_i are the roots of $R_s(z)$, and to represent the RK scheme as a composition of explicit Euler steps:

$$\left.\begin{aligned} &g_0 = u_0, \\ &g_i = g_{i-1} + \delta t \delta_i f(g_{i-1}), \quad i = 1, \ldots, s, \\ &u_1 = g_s. \end{aligned}\right\} \tag{2.93}$$

The main difficulty constitutes finding the best sequence of integration of the Euler steps to ensure stability properties of the scheme [Leb93a, Leb93b]. Formulae of order up to four were also achieved even though there is no analytical expression for the optimal stability polynomials of order $p \geq 2$ [LM98, Med98]. The computations of these polynomials are therefore performed numerically and yield, for instance, second order schemes with practically optimal $\ell_s \approx 0.82 \cdot s^2$ for $s \gg 1$. These results have been implemented in the *DUMKA* code [Leb94, Leb00].
Based on numerical approximations of the optimal boundaries ℓ_s [vdH77], and knowing that among all polynomials of order p and degree s satisfying (2.86), the optimal one satisfies the so-called *equal ripple property* which states that there exist $s - p + 1$ points $z_0 < z_1 < \cdots < z_{s-p} < 0$, with $z_0 = -\ell_s$, such that

$$\left.\begin{aligned} &R(z_i) = -R(z_{i+1}), \quad i = 0, \ldots, s - p - 1, \\ &|R(z_i)| = 1, \qquad\quad i = 0, \ldots, s - p; \end{aligned}\right\} \tag{2.94}$$

another approach known as the *Van der Houwen-Sommeijer methods* [vdHS80], constructs the RK schemes based on a linear combination of scaled and shifted Chebyshev polynomials that aim at approximating the optimal polynomial by verifying (2.94), and generates about 80 % of the optimal interval ℓ_s. First and second order schemes known as *RKC* methods were built with these approximated optimal polynomials using the three-term recurrence formula (2.89):

$$\left.\begin{aligned} &g_0 = u_0, \\ &g_1 = g_0 + \tilde{\mu}_1 \delta t f(g_0), \\ &g_i = (1 - \mu_i - \nu_i) g_0 + \mu_i g_{i-1} + \nu_i g_{i-2} \\ &\qquad + \tilde{\mu}_i \delta t f(g_{i-1}) + \tilde{\gamma}_i \delta t f(g_0), \qquad i = 2, \ldots, s, \\ &u_1 = g_s, \end{aligned}\right\} \tag{2.95}$$

where all the coefficients $(\tilde{\mu}_i, \mu_i, \nu_i, \tilde{\gamma}_i)$ are available in analytical form for arbitrary $s \geq 2$ [SV80]. In this way, an efficient second order solver known simply as *RKC* proposed by Sommeijer *et al.* in [SSV97], gained notorious reputation over the last years. The RKC solver also features local error control, with variable step sizes, computed on an approximation of the leading term of the local error expansion, theoretically derived from a detailed stability and convergence analysis presented in [VHS90]. The stability bound is given by $\ell_s \approx 0.653 \cdot s^2$ for the second order RKC scheme, and hence for a given time step computed according to a prescribed accuracy tolerance, an adequate number of stages s is chosen in order to ensure the stability of the method.

2.4.1 The ROCK Method

A third approach that combined the previous ones by searching practically optimal stability bounds ℓ_s, and by using a three-term recurrence relation, gave birth to the *ROCK* methods (for Orthogonal-Runge-Kutta-Chebyshev) [AM01, Abd02]. A preliminary important result of Abdulle [Abd00] was that the optimal stability polynomials satisfying (2.86) for a given p and the equal ripple property (2.94), possess exactly p complex roots if p is even and exactly $p-1$ complex roots if p is odd. Therefore, if p is even, we can then split the stability function in the following form:

$$R_s(z) = w_p(z)P_{s-p}(z), \tag{2.96}$$

where w_p retains the p complex roots and P_{s-p}, the remaining $(s-p)$ real roots. The idea developed by Medovikov & Abdulle in [AM01] for $p=2$, and then extended to $p=4$ by Abdulle in [Abd02], was to approximate $R_s(z)$ by

$$\tilde{R}_s(z) = \tilde{w}_p(z)\tilde{P}_{s-p}(z), \tag{2.97}$$

with the orthogonal polynomials $\tilde{P}_{s-p}$, associated with the weight function $\tilde{w}_p^2(z)/\sqrt{1-z^2}$, such that $\tilde{R}_s(z)$ results in a p-order stability polynomial which remains bounded as much as possible along the negative real axis, taking also into account some damping. The techniques to compute the orthogonal polynomials and the weight function are given in [AM01] and [Abd02]. Once the stability functions have been computed, a three-term recurrence relation:

$$\tilde{P}_0(z) = 1, \quad \tilde{P}_1(z) = 1 + \mu_1 z, \quad \tilde{P}_i(z) = (\mu_i z - \nu_i)\tilde{P}_{i-1}(z) - \kappa_i \tilde{P}_{i-2}(z), \tag{2.98}$$

with $i = 2, \ldots, s-p$, satisfied by the orthogonal polynomials, is used to define the internal stages of the RK method following the idea of [vdHS80]:

$$\left.\begin{aligned} g_0 &= u_0, \\ g_1 &= g_0 + \tilde{\mu}_1 \delta t f(g_0), \\ g_i &= \tilde{\mu}_i \delta t f(g_{i-1}) - \nu_i g_{i-1} - \kappa_i g_{i-2}, \quad i = 2, \ldots, s-p. \end{aligned}\right\} \tag{2.99}$$

Considering $\mathrm{d}_t u = \lambda u$ and $z = \lambda \delta t$, the resulting $\tilde{P}_{s-p}(z)$ is the stability function associated with (2.99): $g_{s-p} = \tilde{P}_{s-p}(z)u_0$. The coefficients (μ_i, ν_i, κ_i) are computed by a procedure introduced in [AM01].
The case $p=2$ yields thus the second order ROCK2 method [AM01] for which $\tilde{w}_2(z)$ is a two-stage finishing procedure applied to $g_{s-2} = \tilde{P}_{s-2}(z)u_0$. For $\mathrm{d}_t u = \lambda u$ and $z = \lambda \delta t$, this implies

$$u_1 = \tilde{w}_2(z)g_{s-2} = \tilde{w}_2(z)\tilde{P}_{s-2}(z)u_0 = \tilde{R}_s(z)u_0. \tag{2.100}$$

The order conditions for $p=2$ are classical to explicit RK schemes and allow us to compute the coefficients of the final stages. In particular for second order, the order conditions are the same for both linear and nonlinear problems. A solution $\hat{u}_1$ of order $\hat{p}=1$, is computed embedded at the final step $\tilde{w}_2(z)$, and an estimate of the local error $err = (\hat{u}_1 - u_1)$, is computed for the step size selection, according to the same criteria used by Radau5 [HW96] with expressions (2.82) and (2.83). The nearly optimal stability interval is given by $\tilde{\ell}_s \approx 0.81 \cdot s^2$ (the optimal ratio is about 0.82 [vdH77]). Therefore, with the time step fixed by the prescribed accuracy (*Atol* and *Rtol*), the number of stages needed to guarantee stability is computed by

$$\delta t \rho\left(\partial_u f(u)\right) \leq 0.81 \cdot s^2, \tag{2.101}$$

where ρ is the spectral radius of the Jacobian of the system of ODEs. A dynamic computation of this spectral radius is provided by ROCK2 using a non-linear power method which is a slight modification of the algorithm proposed in [SSV97] for the RKC code.
Just like before, for the fourth order ROCK4 ($p = 4$) the coefficients of the weight function $\tilde{w}_4(z)$ must be computed such that the order conditions of order 4 are satisfied. As in [Med98], a theory of composition of methods (the "Butcher group") is applied to achieve a fourth order method denoted WP, where the first method, denoted by P is given by the three-term recurrence relation in (2.99) this time with $p = 4$, whereas the coefficients of the four stages method W associated with $\tilde{w}_4(z)$ are computed such that the "composite" method WP is of order 4 as shown in [Abd02]. As in the previous second order case, an embedded method $\hat{W}$ is built embedded into W in order to keep the same recurrence formulae (2.98) for both the fourth order and embedded methods. A third order embedded RK scheme is thus constructed by adding a new stage to $\tilde{w}_4(z)$, and the coefficients are computed with the same composition technique such that the "composite" method $\hat{W}P$ is of order 3, and that the stability polynomials of the embedded methods are bounded in the same interval as the ones of the ROCK4 scheme. The latter feature is indispensable to guarantee stability of the lower order method, and to obtain thus reliable error estimates.
The practically optimal stability interval is this time given by $\tilde{\ell}_s \approx 0.35 \cdot s^2$ (the optimal ratio for fourth order is about 0.34 in [vdH77] and 0.35 in [Med98]). The ROCK4 solver implements the same tools as ROCK2 for time step selection in terms of estimates (2.82) and (2.83), as well as the numerical computation of the spectral radius. For a given time step δt, computed based on the prescribed accuracy (*Atol* and *Rtol*), the number of stages that ensures stability of computations is now given by

$$\delta t \rho \left(\partial_u f(u) \right) \leq 0.35 \cdot s^2. \tag{2.102}$$

A notorious advantage of the three-term recurrence formulae used by the RKC (2.95) and ROCK (2.99) methods, is that even though an arbitrary number of stages s might be required to guarantee stability, only the current three arrays in the recurrence relations need to be saved. Considering the two-stage $\tilde{w}_2(z)$ for the second order ROCK2, five solution arrays need thus to be saved to perform all the computations. The same follows for ROCK4 for which seven arrays shall be required. Notice that the construction of the ROCK schemes through (2.96) involves at least $s = 3$ and $s = 5$ internal stages, respectively, for ROCK2 and ROCK4 schemes. The main advantage of the ROCK schemes compared with previous stabilized RK schemes is that it combines the best features of both Lebedev- and Van der Houwen-Sommeijer-type methods by using the three-term recurrence formulae with practically optimal stability polynomials. The latter implies larger stability domains in the practical implementations considering that ℓ_s is approximated by $0.81 \cdot s^2$ for ROCK2 compared with $0.65 \cdot s^2$ for the also second order RKC solver [SSV97]. In particular, a higher order, stabilized explicit scheme of easy implementation with an optimal stability interval, was achieved with the ROCK4 solver. In this way, the stability domains of explicit RK methods are extended without altering the orders of the numerical approximations, and furthermore without requiring excessive supplementary memory space with respect to a standard explicit RK scheme.

Chapter 3

Space Adaptive Multiresolution for Multi-Scale Evolutionary PDEs

The previous chapters were mainly dedicated to the numerical integration of time dependent stiff problems. In Chapter 1 we have considered time operator splitting techniques for PDEs, in particular for reaction-diffusion systems for which the time and space discretization errors in the solutions of the split subproblems were neglected. On the other hand, we have considered in Chapter 2 the time integration of stiff systems of ODEs which in our particular case, are issued from a space discretization of the original PDEs, *i.e.,* from the semi-discretized PDEs, and hence the space discretization errors were not discussed. We have seen that a key aspect to efficiently solve all of these problems is related to the accuracy of computations. Nevertheless in many applications the spatial representation of the phenomena, *i.e.,* the spatial discretization of the PDEs, plays a crucial role to properly describe the physical dynamics of the problem. Moreover, the mathematical models and therefore the modeling equations might be neither adequate nor valid if the spatial discretization skips some fundamental scales of the problem. Consequently, the dedicated and possibly high order time integration methods we have previously discussed in chapters 1 and 2, might also give wrong approximations to the physical problems even though they are solving accurately the corresponding spatially discretized PDEs. A natural and necessary condition to guarantee the efficiency of the previous schemes relies then on a sufficiently accurate spatial mesh representation.
Since for realistic applications, a uniform fine mesh is often expensive in computational resources, many high order and dedicated space discretization schemes were developed and studied over the past years (see, *e.g.,* [LeV92, LeV02, HV03]). Another alternative to reduce these computing requirements is to consider rather *adaptive mesh refinement* techniques, with possibly high order space discretizations. In this context, a sufficiently fine mesh to accurately represent the physical phenomena might be built and successfully implemented, by considering a set of spatial grids that are adapted to the local space scales of the problem. The resulting grid is thus sufficiently fine only in partial regions of the computational domain. The degree of refinement in the spatial representations can be considerably enhanced with respect to a uniform grid case, and remains limited by the available computational resources. In this chapter we will focus on these adaptive grid techniques, and in particular on *space adaptive multiresolution* techniques for the spatial representation of PDEs. A general description of adaptive mesh refinement schemes will be first introduced in order to settle the corresponding framework. The general multiresolution analysis on which space adaptive multiresolution is founded, will then be described as well as its mathematical background based on wavelet decomposition, to then end up with the main aspects of these techniques. For further details

on adaptive multiresolution techniques, we refer to the books of Cohen [Coh00] and Müller [Mül03].

3.1 Adaptive Mesh Refinement Techniques

Mesh adaptation techniques constitute a powerful tool to solve PDEs more efficiently by avoiding expensive uniform grids, according to the nature of the problem. For instance, many physical phenomena are characterized by the presence of localized fronts of steep spatial gradients or even discontinuities into a rather smooth larger domain. For these smoother regions the level of spatial resolution may be then loosened, whereas the major computational effort given by a much finer discretization will be concentrated on the localized singularities. The dynamic broad spectrum of active spatial scales in realistic configurations motivated the development of adaptive mesh techniques for both unstructured and structured meshes. The reduced computational complexity of the latter ones contrast with the higher geometry flexibility of the former ones to represent more complex computational domains. In this work, we will discuss only adaptive mesh techniques on structured meshes.

Historically, adaptive discretization techniques were developed from the late seventies, such as the Multi-Level Adaptive Techniques *MLAT* introduced by Brandt [Bra77], and yielded over the past years a large family of method called *AMR* or *SAMR* (respectively, for Adaptive Mesh Refinement or Structured Adaptive Mesh Refinement), conceived especially for CFD (Computational Fluid Dynamics) applications. In this context, the first AMR approaches introduced by Berger & Oliger [Ber82, BO84] considered rotated refined regions known usually as *patches*, over a coarser underlying mesh in order to discriminate smooth and irregular regions within the computational domain. A simplified and thus much more efficient technique developed afterward by Berger & Collela [BC89], considered refinement patches always aligned with the underlying coarse mesh. The latter strategy was shown to feature very high efficiencies as demonstrated by Bell *et al.* [BBSW94], and settled the general framework for AMR methods. In this way, several AMR variants and extensions to these initial works were developed for single processor computing machines (see, *e.g.*, [CW93, FGM97, BL98]) and parallel architectures (see, *e.g.*, [BBSW94, KB95, RBL+00]).
In general, AMR methods follow a local refinement approach by means of refined patches. Therefore, considering a uniform structured mesh over the whole computational domain, finer partitions or cells are considered on the regions exhibiting locally steep gradients or shock-like structures. These finer cells are then clustered in non-overlapping rectangular subgrids of appropriate size aligned with the coarser mesh, and the set of finer subgrids or patches generates a new level of spatial discretization. This procedure is then applied successively to generate new levels of refinement wherever this is required, and an entire hierarchy of embedded grids is constructed. In this way, the refined patches in each discretization level overlay the coarser subgrids from which they have been created, avoiding in particular data fragmentation. The simplicity of this approach implies an important flexibility taking into account that the numerical schemes need to be implemented on a single rectangular subgrid, and thus patches and consequently levels, can be successively solved using the same procedure and hopefully in an independent way.
The variables of the new cells contained in an added patch are initialized, for instance, by interpolation from the underlying coarser mesh, whereas the values of cells covered by refined patches are overwritten by averaged fine-grid values after each update of the solution. The superfluous work on the coarse grid is negligible compared with the computational costs for

integrating the overlying fine grids. Special care must be taken when computing numerical fluxes at the boundaries of the patches in order to avoid loss of conservation. A standard solution replaces simply the numerical fluxes at coarser meshes by the sum of the fine grid fluxes along the corresponding coarse boundary [BC89]. This can be seen as an *a posteriori* flux correction in order to keep the recursion and simplicity of the scheme, since levels are usually updated from the coarsest to the finest one [Dei05]. With the same spirit, *ghost* cells are added around each patch to computationally decouple the subgrids. These ghost cells account also for the physical boundary conditions and otherwise, they are computed either by interpolation from the underlying mesh or by copying the data value from neighboring patches at the same level. Since the advancing time steps are usually defined level-wise according to the corresponding space discretization [BC89], the latter procedure implies either a synchronization among patches of the same level or a time-space interpolation from the underlying mesh to define the ghost cells at intermediate times. The parallelization of the previous general technique is straightforward on shared memory architectures. It suffices to parallelize the loop that updates the levels, or the patches, one by one (see, *e.g.*, [BBSW94]). The communication costs on distributed memory machines cannot be neglected, and the parallelization strategies with appropriate load balancing become significantly more complex (see [Dei05] and references therein).
In order to define whether a cell must be refined, appropriate indicators must identify the presence of steep gradients or discontinuities in the solution. Taking into account that rigorous error estimates are only available for scalar equations in the context of conservation laws [KO00], and that they are usually difficult to implement, heuristic or physical parameters are preferred instead. A standard procedure considers, for instance, a scaled gradient criteria which evaluate the local gradients multiplied by the corresponding spatial step in all directions, and refine cells beyond a prescribed tolerance (see, *e.g.*, [Dei05]). Another indicator is based on different ways to estimate the local truncation errors of the numerical scheme (see, *e.g.*, [SHH94, HMSW99]), which can be heuristically computed in practice by means of a Richardson extrapolation performed using a fine and the corresponding coarser local solution [Ber82, BO84, BC89]. Finally, a security layer of cells is often added in order to guarantee that the fronts remain into the corresponding refined regions during the integration time step. In general, AMR methods are largely used and show to be very performing in many applications. Some examples of freely available AMR software libraries, developed for multidimensional simulations with parallel computing features, are Chombo[1] by Collela *et al.*, PARAMESH[2] [MOM+00], RAMSES[3] [Tey02], AMROC[4] [Dei05], SAMRAI[5] [HWK06], among others. Although the theoretical framework and ideas are rather common for AMR techniques, there are several differences in terms of data structure, programming language, routines optimization, and other practical implementation issues. In this context, the strong heuristic component of such strategies as briefly discussed in this part, constitutes an important drawback to establish an appropriate mathematical framework that might allow us to describe the approximation errors introduced in the numerical solutions.

Adaptive multiresolution methods, based on the pioneering work of Harten [Har94a, Har95], were then developed as a new adaptive mesh refinement technique. In particular, a family of *fully adaptive multiresolution scheme* was introduced by Cohen *et al.* in [CKMP03]. Considering

[1] `https://seesar.lbl.gov/anag/chombo`
[2] `http://sourceforge.net/projects/paramesh/`
[3] `http://irfu.cea.fr/Projets/Site_ramses`
[4] `http://www.cacr.caltech.edu/asc`
[5] `https://computation.llnl.gov/casc/SAMRAI`

a set of nested spatial grids from the coarsest to the finest one, a multiresolution analysis allows us 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 correspond to the wavelet coefficients of a wavelet decomposition obtained by inter-level operations, and retain the information on local regularity when going from a coarse to a finer grid. Lower wavelet coefficients are associated with local regular spatial configurations and vice-versa. A thresholding process applied to this representation allows us to dynamically build the corresponding adapted grid on which the solutions are represented [CKMP03, Mül03]. The main advantage of such procedure is that, based on the mathematical basis of the multiresolution analysis, an error of the spatial representation of the solution represented on the adapted grid can be inferred, with respect to the solution discretized on the finest grid.
In what follows, the theoretical framework of multiresolution techniques will be detailed. We will then describe the space adaptive multiresolution scheme conceived as an adaptive mesh refinement method for time dependent PDEs. Some reviews on such topics can be found in [Har94b, CDD04, Pos05], whereas a first comparison of performances between general AMR and multiresolution methods was conducted in [DDG+09].

3.2 Basis of Multiresolution Analysis

For positive integers $j \geq j_0$, let us consider partitions of the interval $[0,1]$, given by the set of dyadic intervals:

$$I_{j,k} := [2^{-j}k, 2^{-j}(k+1)[, \qquad k = 0, \ldots, 2^j - 1. \tag{3.1}$$

We define for a given $f(x) \in L^2([0,1])$, a constant approximation $P_j f|_{I_{j,k}}$ of f on each interval $I_{j,k}$, given by

$$P_j f|_{I_{j,k}} := 2^j \int_{I_{j,k}} f(x)\,\mathrm{d}x = a_{I_{j,k}}(f), \tag{3.2}$$

such that

$$P_j f = \sum_{k=0}^{2^j-1} a_{I_{j,k}}(f)\chi_{I_{j,k}}(x), \tag{3.3}$$

is the piecewise constant approximation of f on a dyadic mesh of size 2^{-j}. In (3.3), $\chi_\Omega(x)$ is the standard characteristic function which takes values of 1 in Ω, and zero outside. Therefore, for each[6] $j \in \mathbb{N}_0$, $P_j f$ is indeed an L^2-orthogonal projection onto the space V_j of piecewise constant functions:

$$V_j := \left\{ f \in L^2 \text{ s.t. } f|_{I_{j,k}} = \text{constant}, \, k = 0, \ldots, 2^j - 1 \right\}. \tag{3.4}$$

3.2.1 Wavelet Decomposition

Introducing the *box function* or *B-spline of order* 1: $\phi(x) = \chi_{[0,1)}(x)$, an orthonormal basis of V_j is given by the set of normalized $\phi_{j,k}$ that dilates and shifts

$$\phi_{j,k} = 2^{j/2}\chi_{I_{j,k}}(\cdot) = 2^{j/2}\phi(2^j \cdot - k), \qquad k = 0, \ldots, 2^j - 1, \tag{3.5}$$

of dimension $\dim V_j = 2^j$, for which

$$\langle \phi_{j,k}, \phi_{j,l} \rangle_{[0,1)} = \int_0^1 \phi_{j,k}(x)\phi_{j,l}(x)\,\mathrm{d}x = \delta_{k,l}. \tag{3.6}$$

[6] We denote the set $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$.

A proper definition of the piecewise constant approximation operator (3.3) is then given by the orthogonal projection:

$$P_j f := \sum_{k=0}^{2^j-1} \langle f, \phi_{j,k} \rangle \phi_{j,k} \tag{3.7}$$

where

$$c_{j,k} := \langle f, \phi_{j,k} \rangle = 2^{-j/2} a_{I_{j,k}}(f), \tag{3.8}$$

is the *approximation coefficient* at scale 2^{-j} and position $2^{-j}k$. This projection resolves the function f up to the scale 2^j, while finer details are discarded.
Taking into account that the spaces V_j, $j \geq j_0$, are nested: $V_j \subset V_{j+1} \subset V_{j+2} \subset \ldots$, and that $\overline{\cup V_j} = L^p([0,1])$, we have that if $f \in L^p([0,1])$, then $P_j f$ is also convergent in $L^p([0,1])$, *i.e.*,

$$\lim_{j \to +\infty} \|f - P_j f\|_{L^p} = 0, \tag{3.9}$$

for $1 \leq p < \infty$. The same follows for the uniform norm as long as $f \in \mathcal{C}^0$:

$$\lim_{j \to +\infty} \|f - P_j f\|_{\infty} = 0. \tag{3.10}$$

It is straightforward to see that if from an initial coarsest j_0 scale, we add the finer details successively skipped at each scale j, we obtain

$$f = P_{j_0} f + \sum_{j=j_0}^{\infty} (P_{j+1} - P_j) f, \tag{3.11}$$

that is, a *multi-scale representation* of f, where each term $Q_j f := (P_{j+1} - P_j) f$ represents the *details* in f at scale 2^{-j}. It is then natural to decompose the projection of f onto V_j: $P_j f$, as the sum of the projection onto V_{j-1}: $P_{j-1} f$, plus the associated detail: $(P_j - P_{j-1}) f$. Iterating from a fixed $J > j_0$, leads to the following multi-scale representation:

$$P_J f = P_{J-1} f + [P_J f - P_{J-1} f] = \ldots = P_{j_0} f + \sum_{j=j_0}^{J-1} Q_j f. \tag{3.12}$$

Moreover, we can see from (3.2) that constant approximations at successive scales are related by

$$a_{I_{j,k}}(f) = \frac{1}{2} (a_{I_{j+1,2k}}(f) + a_{I_{j+1,2k+1}}(f)), \tag{3.13}$$

which means that $Q_j f$ should oscillate within each $I_{j,k}$ in order to be able to reconstruct $P_{j+1} f$ from $P_j f$. In this way, the fine scale values can be recovered from a coarser scale by using the box function $\phi(x)$, and an oscillatory profile $\psi(x)$. For instance, considering the *Haar wavelet* given by $\psi(x) := \chi_{[0,1/2)}(x) - \chi_{[1/2,1)}(x)$, or

$$\psi(x) := \phi(2x) - \phi(2x-1), \tag{3.14}$$

and that for the box function

$$\phi(x) = \phi(2x) + \phi(2x+1), \tag{3.15}$$

we have that

$$\phi(2x) = \frac{1}{2}(\phi(x) + \psi(x)), \qquad \phi(2x-1) = \frac{1}{2}(\phi(x) - \psi(x)). \tag{3.16}$$

Thus defining the $\psi_{j,k} = 2^{j/2}\psi(2^j \cdot -k)$ that dilates and shifts on the partitions of $[0,1]$ yields

$$Q_j f = \sum_{k=0}^{2^j-1} \langle f, \psi_{j,k}\rangle \psi_{j,k}, \tag{3.17}$$

where

$$d_{j,k} := \langle f, \psi_{j,k}\rangle, \tag{3.18}$$

is the *wavelet coefficient* at scale 2^{-j} and position $2^{-j}k$. Following (3.12), Q_j is then the orthogonal projection of f on the orthogonal complement space W_j of V_j into V_{j+1}, $W_j \oplus^\perp V_j = V_{j+1}$. We see thus that $(\psi_{j,k})_{k=0,\dots,2^j-1}$ is an orthonormal basis of W_j, and therefore

$$(\psi_{j,k})_{k=0,\dots,2^j-1} \cup (\phi_{j,k})_{k=0,\dots,2^j-1}\,, \tag{3.19}$$

is an orthonormal basis of V_{j+1}. In the same way, V_j can be written as

$$V_j = W_{j-1} \oplus^\perp V_{j-1} = W_{j-1} \oplus^\perp W_{j-2} \oplus^\perp V_{j-2} = \dots, \tag{3.20}$$

as we can infer from the multi-scale representation (3.12). In particular, when $J \to +\infty$ with $j_0 = 0$, $P_J f$ tends to f in L^2 by (3.11), and it follows that

$$\{\phi\} \cup \{\psi_{j,k}\}_{j\geq 0, k=0,\dots,2^j-1}\,, \tag{3.21}$$

is an orthonormal basis of $L^2[0,1]$, where (3.21) is usually known as the *Haar system*.
With this representation, the following *two-scale relations* are verified according to (3.16):

$$\left.\begin{aligned} \phi_{j,k} &= \frac{1}{\sqrt{2}}(\phi_{j+1,2k} + \phi_{j+1,2k+1}), & \psi_{j,k} &= \frac{1}{\sqrt{2}}(\phi_{j+1,2k} - \phi_{j+1,2k+1}),\\ \phi_{j+1,2k} &= \frac{1}{\sqrt{2}}(\phi_{j,k} + \psi_{j,k}), & \phi_{j+1,2k+1} &= \frac{1}{\sqrt{2}}(\phi_{j,k} - \psi_{j,k}), \end{aligned}\right\} \tag{3.22}$$

which leads to a *change of basis*:

$$\sum_{k=0}^{2^{j+1}-1} \langle f, \phi_{j+1,k}\rangle \phi_{j+1,k} = \sum_{k=0}^{2^j-1} \langle f, \phi_{j,k}\rangle \phi_{j,k} + \sum_{k=0}^{2^j-1} \langle u, \psi_{j,k}\rangle \psi_{j,k}, \tag{3.23}$$

or equivalently,

$$\sum_{k=0}^{2^{j+1}-1} c_{j+1,k}\phi_{j+1,k} = \sum_{k=0}^{2^j-1} c_{j,k}\phi_{j,k} + \sum_{k=0}^{2^j-1} d_{j,k}\psi_{j,k}, \tag{3.24}$$

where

$$\left.\begin{aligned} c_{j,k} &= \frac{1}{\sqrt{2}}(c_{j+1,2k} + c_{j+1,2k+1}), & d_{j,k} &= \frac{1}{\sqrt{2}}(c_{j+1,2k} - c_{j+1,2k+1}),\\ c_{j+1,2k} &= \frac{1}{\sqrt{2}}(c_{j,k} + d_{j,k}), & c_{j+1,2k+1} &= \frac{1}{\sqrt{2}}(c_{j,k} - d_{j,k}). \end{aligned}\right\} \tag{3.25}$$

The representation in term of the fine scales can be retrieved from the coarse scale averages by adding the detail, lost through the coarse projection. A recursive change of basis based on these two-scale coefficients (3.25) yields a telescopic transform known as the *fast wavelet transform* $\mathcal{W}$.
As a consequence, for a given $J > j_0$, a function $f_J \in V_J$ can be written either on the standard canonical basis:

$$f_J = \sum_{k=0}^{2^J-1} c_{J,k}\phi_{J,k}, \tag{3.26}$$

or on a *wavelet* or *multi-scale basis*:

$$f_J = \sum_{k=0}^{2^{j_0}-1} c_{j_0,k}\phi_{j_0,k} + \sum_{j=j_0}^{J-1}\sum_{k=0}^{2^j-1} d_{j,k}\psi_{j,k}, \tag{3.27}$$

according to (3.7), (3.12), and (3.17). The change of representation from (3.26) to (3.27) is performed by the wavelet decomposition, where $\mathcal{W}$ transforms a linear combination of fine scale box functions with an array of coefficients $\mathbf{c}_J$, into a linear combination of coarse scale box functions with coefficient array $\mathbf{c}_0$ and Haar wavelets with array of detail coefficients $\mathbf{d}_j$ for each dyadic level $j < J$:

$$\mathcal{W} : \mathbf{c}_J \to \mathbf{d}^J := (\mathbf{c}_0, \mathbf{d}_0, \mathbf{d}_1, \ldots, \mathbf{d}_{J-1})\,. \tag{3.28}$$

In the same way and based on the same relations (3.25), the inverse transform $\mathcal{W}^{-1} : \mathbf{d}^J \to \mathbf{c}_J$, turns the wavelets coefficients into the single scale J. Due to the telescopic structure of these computations and because the relations (3.25) involve only finite coefficients ($\pm 1/\sqrt{2}$) called usually *masks*, the number of operations required by both transforms is $\mathcal{O}(2^J)$.
Taking into account the convergence of the orthogonal projections P_j in L^2, and the wavelet decomposition (3.27) with $j_0 = 0$, we can write (3.11) as

$$f = P_0 f + \sum_{j=0}^{\infty}(P_{j+1} - P_j)f = \sum_{j=-1}^{\infty}\sum_{k=0}^{2^j-1} d_{j,k}\psi_{j,k} := \mathbf{d}(f)^T\boldsymbol{\Psi}, \tag{3.29}$$

where $P_{-1} := 0$ and $\psi_{-1,k} := \phi_{0,k}$. One of the main advantages of such a wavelet representation is that there is a tight relation between the function and the coefficient norms [CDD04]:

$$\|f\|_{L^2} = \left(\sum_{j=-1}^{\infty}\|(P_{j+1} - P_j)f\|_{L^2}^2\right)^{1/2} = \|\mathbf{d}(f)\|_{\ell^2}, \tag{3.30}$$

due to the orthonormality of $\psi_{j,k}$ and using the Parseval's identity. This means that if in the wavelet representation some small coefficients are discarded, the norm of the function will be modified only by the same small amount. In particular, the size of these coefficients can be inferred from local properties of f. It can be demonstrated that within the support $I_{j,k}$ of $\psi_{j,k}$:

$$|d_{j,k}| = \inf_{c\in\mathbb{R}}|\langle f - c, \psi_{j,k}\rangle| \le \inf_{c\in\mathbb{R}}\|f - c\|_{L^2(I_{j,k})} \le 2^{-j}\|f'\|_{L^2(I_{j,k})}, \tag{3.31}$$

by using a formal Taylor series expansion, and noticing that $\|\psi_{j,k}\|_{L^2(I_{j,k})} = 1$, and that the Haar wavelets $\psi_{j,k}$ are orthogonal to any constant $c \in \mathbb{R}$, *i.e.*, they have *first order vanishing moments*:

$$\langle c, \psi_{j,k}\rangle = 0. \tag{3.32}$$

The decay of the wavelet coefficients is directly influenced by the local smoothness of f. Consequently, the coefficients $d_{j,k}$ get small at fine scales when $f|_{I_{j,k}}$ is sufficiently smooth, whereas high gradients involve more significant values.
This kind of constant piecewise approximation shows to be only *first order accurate*. The Haar's set of wavelets is therefore only suitable to efficiently represent sufficiently smooth f. From a practical point of view, this seldom happens and motivated further studies based on the potential advantages inherited from this simple Haar configuration. In particular, a family of compactly supported orthonormal wavelets in $L^2(\mathbb{R})$ introduced by Daubechies [Dau88, Dau92], offered the possibility of arbitrary high regularity of the multi-scale approximations.

3.2.2 Orthonormal Wavelets

The general mathematical framework as presented before for the Haar case, was actually settled by the *multiresolution approximation* introduced by Mallat [Mal89] and Meyer [Mey90], that considers the nested spaces $V_j \subset V_{j+1} \subset \ldots$ of $L^2(\mathbb{R})$ such that $\overline{\cup V_j} = L^2(\mathbb{R})$, and thus $P_j f$ converges in $L^2(\mathbb{R})$ for all $f \in L^2(\mathbb{R})$, where P_j is the L^2-orthogonal projector. In this context, there exists a *scaling function* $\phi \in V_0$ such that

$$\phi_{j,k} = 2^{j/2}\phi(2^j \cdot -k), \qquad k \in \mathbb{Z}, \tag{3.33}$$

is a *Riesz basis* of V_j, *i.e.*, there exist positive constants, $0 < c < C < \infty$, such that

$$c\|x_{j,k}\|_{\ell^2}^2 \leq \left\| \sum_{k\in\mathbb{Z}} x_{j,k}\phi_{j,k} \right\|_{L^2}^2 \leq C\|x_{j,k}\|_{\ell^2}^2, \tag{3.34}$$

and hence there is a unique representation of $f_j \in V_j$ in this basis: $f_j = \sum_{k\in\mathbb{Z}} x_{j,k}\phi_{j,k}$. Furthermore, if the Riesz property (3.34) is satisfied for any $j \in \mathbb{N}_0$, for c and C independent of j, then the set $(\phi_{j,k})_{j\in\mathbb{N}_0}$ is said to be *uniformly stable*.
Considering now $\phi \in V_0 \subset V_1$, we define a two-scale relation:

$$\phi(x) = \sum_{k\in\mathbb{Z}} a_k \phi(2x - k), \tag{3.35}$$

with finitely supported *masks*: $(a_k)_{k\in\mathbb{Z}}$. For instance, for the box function, $a_0 = a_1 = 1$, and $a_k = 0$ otherwise, following (3.15). Considering (3.33) for $V_j \subset V_{j+1}$ yields

$$\phi_{j,k} = \frac{1}{\sqrt{2}} \sum_{n\in\mathbb{Z}} a_n \phi_{j+1,2k+n}. \tag{3.36}$$

Assuming that the set $(\phi_{j,k})_{k\in\mathbb{Z}}$ is an orthonormal basis of V_j such that

$$P_j f := \sum_{k\in\mathbb{Z}} \langle f, \phi_{j,k}\rangle \phi_{j,k}, \tag{3.37}$$

the wavelet ψ is constructed by

$$\psi(x) = \sum_{k\in\mathbb{Z}} b_k \phi(2x - k), \tag{3.38}$$

with $b_k = (-1)^k a_{1-k}$. This $(\psi_{j,k})_{k\in\mathbb{Z}}$ is then conceived as an orthonormal basis of the orthogonal complement $W_j = V_{j+1} \cap V_j^{\perp}$, defining the orthogonal projection:

$$Q_j f = (P_{j+1} - P_j) f = \sum_{k\in\mathbb{Z}} \langle f, \psi_{j,k}\rangle \psi_{j,k}, \tag{3.39}$$

and thus f can be written on an orthonormal basis of $L^2(\mathbb{R})$:

$$\left.\begin{aligned} f &= P_0 f + \sum_{j\in\mathbb{N}_0} Q_j f, \\ f &= \sum_{k\in\mathbb{Z}} \langle f, \phi_{0,k}\rangle \phi_{0,k} + \sum_{j\in\mathbb{N}_0} \sum_{k\in\mathbb{Z}} \langle f, \psi_{j,k}\rangle \psi_{j,k}. \end{aligned}\right\} \tag{3.40}$$

With this representation, the equivalence of norms of the function and the wavelet decomposition (3.30) previously estimated, holds naturally as a consequence of the Riesz property

(3.34) for orthonormal basis: $c = C = 1$, *i.e.*, the Parseval's identity. Nevertheless, in order to obtain more accurate representations, the main idea is to build scaling functions (3.35) with a composition of piecewise polynomials, *i.e.*, *B-splines*, such that the set of wavelet (3.38) has N vanishing polynomials moments:

$$\langle P, \psi_{j,k} \rangle_{\Sigma_{j,k}} = 0, \tag{3.41}$$

for any polynomial $P \in \mathbb{P}_{N-1}$, where $\Sigma_{j,k} := \operatorname{supp} \psi_{j,k}$. And consequently we obtain approximations of order N [CDD04],

$$\begin{aligned} |\langle f, \psi_{j,k} \rangle| &= \inf_{P \in \mathbb{P}_{N-1}} |\langle f - P, \psi_{j,k} \rangle| \\ &\leq \inf_{P \in \mathbb{P}_{N-1}} \|f - P\|_{L^p(\Sigma_{j,k})} \|\psi_{j,k}\|_{L^q(\Sigma_{j,k})} \\ &\leq C 2^{-j\left(N + \frac{1}{2} - \frac{1}{p}\right)} |f|_{W_p^N(\Sigma_{j,k})}, \end{aligned} \tag{3.42}$$

using *Hölder's inequality* and thus $p + q = 1$, with the assumption that

$$\|\psi_{j,k}\|_{L^q(\Sigma_{j,k})} \leq C 2^{-j\left(\frac{1}{q} - \frac{1}{2}\right)} = C 2^{-j\left(\frac{1}{2} - \frac{1}{p}\right)}, \tag{3.43}$$

when $\|\psi_{j,k}\|_{L^2(\Sigma_{j,k})} \approx 1$, and based on a standard estimate on *local polynomial approximation* (see, *e.g.*, [DS84]):

$$\inf_{P \in \mathbb{P}_n} \|f - P\|_{L^p(\Omega)} \leq C (\operatorname{diam} \Omega)^n |f|_{W_p^n(\Omega)}. \tag{3.44}$$

The semi-norm associated with the *Sobolev space* $W_p^n(\Omega)$:

$$W_p^n(\Omega) := \{ f \text{ s.t. } \partial^\alpha f \in L^p(\Omega),\ |\alpha| \leq n \}, \tag{3.45}$$

is given by $|f|_{W_p^n(\Omega)} := \left(\sum_{|\alpha|=n} \|\partial^\alpha f\|_{L^p(\Omega)}^p \right)^{1/p}$.
Finally, in order to construct the scaling function $\phi(x)$, one has to compute the masks into (3.35) such that

$$\sum_{n \in \mathbb{Z}} a_n a_{n+2k} = \begin{cases} 2, & k = 0, \\ 0, & k \neq 0, \end{cases} \tag{3.46}$$

according to (3.36), to guarantee the orthonormality of $(\phi_{j,k})_{k \in \mathbb{Z}}$ and $(\psi_{j,k})_{k \in \mathbb{Z}}$; and

$$\sum_{n \in \mathbb{Z}} a_n = 2, \qquad \sum_{n \in \mathbb{Z}} (-1)^n n^m a_n = 0, \qquad m = 0, \ldots, N, \tag{3.47}$$

to obtain order N, *i.e.*, N vanishing moments for the wavelet defined by (3.38). The construction of orthonormal wavelets conceived by Daubechies [Dau88, Dau92] considers then for $N > 0$, a set of (a_n) supported on $\{0, \ldots, 2N-1\}$, such that the resulting $\phi^{(N)}$ compactly supported on $[0, 2N-1]$, generates orthonormal shifts and yields approximations of order N. However, except for $\phi^{(1)}$ which is the first order Haar wavelet decomposition, no explicit expressions exist.
The construction of this orthonormal family of wavelets of arbitrary order approximation constituted a breakthrough in the domain and led, for instance, to the construction of *biorthogonal wavelets* introduced by Cohen *et al.* [CDF92], which have succeeded to somehow improve and establish a more appropriate framework for these techniques.

3.2.3 Biorthogonal Wavelets

The main idea of a biorthogonal wavelet approach is to replace the orthogonality assumption and to introduce a *dual scaling function* $\tilde{\phi}$ along with ϕ, so that we have

$$\phi(x) = \sum_{k\in\mathbb{Z}} a_k \phi(2x-k), \quad \tilde{\phi}(x) = \sum_{k\in\mathbb{Z}} \tilde{a}_k \tilde{\phi}(2x-k), \tag{3.48}$$

with finitely supported masks $(a_k)_{k\in\mathbb{Z}}$, $(\tilde{a}_k)_{k\in\mathbb{Z}}$, and with a biorthogonal property:

$$\langle \phi, \tilde{\phi}(\cdot - k)\rangle = \delta_{0,k}, \quad k \in \mathbb{Z}. \tag{3.49}$$

With this *dual pair* of scaling functions, we can define a non-orthogonal projection

$$P_j f := \sum_{k\in\mathbb{Z}} \langle f, \tilde{\phi}_{j,k}\rangle \phi_{j,k}, \tag{3.50}$$

onto V_j. In the same way, a dual pair of wavelets are defined

$$\psi(x) = \sum_{k\in\mathbb{Z}} b_k \phi(2x-k), \quad \tilde{\psi}(x) = \sum_{k\in\mathbb{Z}} \tilde{b}_k \tilde{\phi}(2x-k), \tag{3.51}$$

with $b_k = (-1)^k \tilde{a}_{1-k}$ and $\tilde{b}_k = (-1)^k a_{1-k}$; and the pairwise biorthogonality property:

$$\langle \psi_{j,k}, \tilde{\psi}_{l,m}\rangle = \delta_{(j,k),(l,m)}, \quad k,m \in \mathbb{Z},\ j,l \in \mathbb{N}_0. \tag{3.52}$$

In particular,

$$\langle \phi_{j,k}, \tilde{\psi}_{j,l}\rangle = \langle \tilde{\phi}_{j,k}, \psi_{j,l}\rangle = 0, \tag{3.53}$$

and we have

$$Q_j f = (P_{j+1} - P_j) f = \sum_{k\in\mathbb{Z}} \langle f, \tilde{\psi}_{j,k}\rangle \psi_{j,k}, \tag{3.54}$$

onto the non-orthogonal complement $W_j = V_{j+1} \cap \tilde{V}_j^{\perp}$. The pairs $(\phi_{j,k}, \psi_{j,k})$ and $(\tilde{\phi}_{j,k}, \tilde{\psi}_{j,k})$ are usually called the *primal* scaling function and wavelet, and the *dual* scaling function and wavelet, respectively.

Finally, if the set $(\phi_{j,k})_{j\in\mathbb{N}_0}$ is uniformly stable and the set $\phi_{0,k} \cup (\psi_{j,k})_{j\in\mathbb{N}_0}$ is a Riesz basis of $L^2(\mathbb{R})$, there exists another Riesz basis given by $\tilde{\phi}_{0,k} \cup (\tilde{\psi}_{j,k})_{j\in\mathbb{N}_0}$, which satisfies the biorthogonality property (3.52), and such that $f \in L^2(\mathbb{R})$ has the unique expansions in the biorthogonal bases of $L^2(\mathbb{R})$:

$$\left.\begin{aligned} f &= \sum_{k\in\mathbb{Z}} \langle f, \tilde{\phi}_{0,k}\rangle \phi_{0,k} + \sum_{j\in\mathbb{N}_0}\sum_{k\in\mathbb{Z}} \langle f, \tilde{\psi}_{j,k}\rangle \psi_{j,k}, \\ f &= \sum_{k\in\mathbb{Z}} \langle f, \phi_{0,k}\rangle \tilde{\phi}_{0,k} + \sum_{j\in\mathbb{N}_0}\sum_{k\in\mathbb{Z}} \langle f, \psi_{j,k}\rangle \tilde{\psi}_{j,k}. \end{aligned}\right\} \tag{3.55}$$

Taking into account the Riesz property (3.34), it can be shown that

$$\left.\begin{aligned} c \sum_{j=-1}^{\infty}\sum_{k\in\mathbb{Z}} |\langle f, \tilde{\psi}_{j,k}\rangle|^2 \le \|f\|_{L^2}^2 \le C \sum_{j=-1}^{\infty}\sum_{k\in\mathbb{Z}} |\langle f, \tilde{\psi}_{j,k}\rangle|^2, \\ c \sum_{j=-1}^{\infty}\sum_{k\in\mathbb{Z}} |\langle f, \psi_{j,k}\rangle|^2 \le \|f\|_{L^2}^2 \le C \sum_{j=-1}^{\infty}\sum_{k\in\mathbb{Z}} |\langle f, \psi_{j,k}\rangle|^2, \end{aligned}\right\} \tag{3.56}$$

where $\psi_{-1,k} := \phi_{0,k}$ and $\tilde{\psi}_{-1,k} := \tilde{\phi}_{0,k}$, and thus a tight relation between the function and the coefficient norm is preserved.

The masks are determined in a similar way as in the previous orthonormal case, accounting this time for the duality relations:

$$\sum_{n\in\mathbb{Z}} \tilde{a}_n a_{n+2k} = \begin{cases} 2, & k=0, \\ 0, & k\neq 0. \end{cases} \tag{3.57}$$

The main advantage of such biorthogonal construction is that it allows us a higher degree of liberty with respect to orthogonal wavelets. For instance, computing the primal scaling function ϕ as any B-spline, the primal wavelet generator ψ becomes directly a spline function with an explicit analytical expression given by the piecewise polynomials according to (3.51). One may also require that the primal or dual scaling functions, which can be seen as a composition of piecewise functions as in the previous case, satisfies some order conditions as given by (3.47). In this way, the biorthogonality relations imply that the generated primal or dual wavelets inherit arbitrary N or $\tilde{N}$ vanishing moments, and thus a decay of 2^{-jN} or $2^{-j\tilde{N}}$ when f has bounded derivatives, following (3.42). From a practical point of view, this flexibility yields simple implementations while the theoretical framework guarantees good mathematical properties.

As in the first Haar's case, we can define a fast wavelet transform taking into account both the standard and wavelet representations for a given $J > j_0 > 0$:

$$f_J = \sum_{k\in\mathbb{Z}} c_{J,k}\phi_{J,k} = \sum_{k\in\mathbb{Z}} c_{j_0,k}\phi_{j_0,k} + \sum_{j=j_0}^{J-1}\sum_{k\in\mathbb{Z}} d_{j,k}\psi_{j,k}, \tag{3.58}$$

where , for instance, and similar to the two-scale relations (3.25), we can deduce more general expressions based on biorthogonal wavelets. We can thus deduce

$$\begin{aligned} c_{j,k} &= \langle f, \tilde{\phi}_{j,k}\rangle = \left\langle f, \frac{1}{\sqrt{2}}\sum_{n\in\mathbb{Z}} \tilde{a}_n \tilde{\phi}_{j+1,2k+n}\right\rangle \\ &= \frac{1}{\sqrt{2}}\sum_{n\in\mathbb{Z}} \tilde{a}_n c_{j+1,2k+n} = \frac{1}{\sqrt{2}}\sum_{n\in\mathbb{Z}} \tilde{a}_{n-2k} c_{j+1,n}. \end{aligned} \tag{3.59}$$

Proceeding in the same way for the coefficients $d_{j,k}$, we finally have

$$c_{j,k} = \frac{1}{\sqrt{2}}\sum_{n\in\mathbb{Z}} \tilde{a}_{n-2k} c_{j+1,n}, \quad d_{j,k} = \frac{1}{\sqrt{2}}\sum_{n\in\mathbb{Z}} \tilde{b}_{n-2k} c_{j+1,n}. \tag{3.60}$$

Computing in the opposite direction

$$\begin{aligned} f_{j+1} &= \sum_{k\in\mathbb{Z}} c_{j+1,k}\phi_{j+1,k} = \sum_{n\in\mathbb{Z}} c_{j,n}\phi_{j,n} + \sum_{n\in\mathbb{Z}} d_{j,n}\psi_{j,n} \\ &= \sum_{n\in\mathbb{Z}} c_{j,n}\left(\frac{1}{\sqrt{2}}\sum_{k\in\mathbb{Z}} a_{k-2n}\phi_{j+1,k}\right) + \sum_{n\in\mathbb{Z}} d_{j,n}\left(\frac{1}{\sqrt{2}}\sum_{k\in\mathbb{Z}} b_{k-2n}\phi_{j+1,k}\right) \\ &= \sum_{k\in\mathbb{Z}} \frac{1}{\sqrt{2}}\left(\sum_{n\in\mathbb{Z}} c_{j,n}a_{k-2n} + \sum_{n\in\mathbb{Z}} d_{j,n}b_{k-2n}\right)\phi_{j+1,k}, \end{aligned} \tag{3.61}$$

yields

$$c_{j+1,k} = \frac{1}{\sqrt{2}}\sum_{n\in\mathbb{Z}} c_{j,n}a_{k-2n} + \frac{1}{\sqrt{2}}\sum_{n\in\mathbb{Z}} d_{j,n}b_{k-2n}. \tag{3.62}$$

The recursive operations (3.60) and (3.62), define the direct and inverse wavelet transforms $\mathcal{W} : \mathbf{c}_J \to \mathbf{d}^J$ and $\mathcal{W}^{-1} : \mathbf{d}^J \to \mathbf{c}_J$, where all computations involve only the set of constant coefficients $(a_n, \tilde{a}_n, b_n, \tilde{b}_n)$.

3.3 Adaptive Multiresolution Finite Volume Scheme

Let us consider a general scalar PDE:

$$\left.\begin{array}{ll} \partial_t u = \Phi(u), & \mathrm{x} \in \mathbb{R}^d,\ t > 0, \\ u(0,\mathrm{x}) = u_0(\mathrm{x}) & \mathrm{x} \in \mathbb{R}^d, \end{array}\right\} \tag{3.63}$$

with $u : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$, $u_0 \in \mathbb{R}$, and

$$\Phi(u) = \partial_{\mathrm{x}} \cdot (D(u)\partial_{\mathrm{x}} u) - \partial_{\mathrm{x}} \cdot g(u) + f(u), \tag{3.64}$$

for $g(u) = (g_i(u))_{i=1,\dots,d}$ and $g_i : \mathbb{R} \to \mathbb{R}$.
Based on the wavelet decomposition theoretical framework previously described, Beylkin *et al.* [BCR91] used wavelet transforms to design fast multiresolution algorithms for matrix-vector multiplication which were later used in [EOZ94] to solve linear hyperbolic and parabolic initial value problems. Alternatively, Liandrat & Tchamitchian [LT90] and Maday & Ravel [MR92] introduced some multiresolution schemes with a Galerkin-type approach with respect to a wavelet basis for the Burgers' equation, whereas Bacry *et al.* [BMP92] developed a wavelet based space/time adaptive method for parabolic and hyperbolic problems. Further studies developed collocation-based adaptive wavelet methods to solve PDEs with general boundary conditions (see, *e.g.,* the work of Vasilyev *et al.* in [VPS95, VP96, VP97]). Another wavelet-based family of adaptive method for PDEs [VB00, Vas03], was built upon the so-called *second generation wavelets* developed by Sweldens [Swe95, Swe98], which utterly led to time/space adaptive wavelet methods (see [AKV06] and references therein). In general, the main idea in these methods was to entirely solve the PDEs in the wavelet basis by first computing all nonlinear and differential operators on an appropriate basis, discretized over a set of different grids. As a consequence, a sequence of algebraic problems issued from the adaptive discretized representations, needed to be solved to advance the solution in time[7]. A recent review on wavelet methods for CFD can be found in [SV10].
Nevertheless, a different approach that we will present in the following, was introduced by Harten [Har94a, Har95], and settled a more appropriate and general framework for multiresolution schemes for the solution of hyperbolic conservation laws, *i.e.,* $\Phi(u) = -\partial_{\mathrm{x}} \cdot g(u)$. These developments utterly led to the introduction of *fully adaptive multiresolution finite volume schemes*, introduced by Cohen *et al.* [CKMP03]. Extensions to reaction-diffusion or reaction-convection-diffusion ware later performed (see, *e.g.,* [RS02, RSTB03, BRBSS08, BBRBS09]). The notion of multiresolution analysis was also extended to nested unstructured grids in [Abg95, AH98], curvilinear grid patches in [DGMM01, BGMH+03], and to cell-centered triangles for more complex geometries in [CDKP00]. On the other hand, another multiresolution approach considers point-valued algorithms instead of the cell-averaged finite volume scheme [Har95, CD01]. These issues will not be discussed in this work and we refer to the given bibliography for more detailed information.
Without loss of generality, we perform then a nested finite volume discretizations of the general problem (3.63). For $j = 0, 1, \cdots, J$, from the coarsest to the finest grid, we build regular

[7] In [AKV06], a single algebraic system is built since both time and space operators are simultaneously discretized.

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. The index j refers thus to the scale level and we denote

$$|\gamma| := j \quad \text{if} \quad \gamma \in S_j, \tag{3.65}$$

with the abbreviated notation $\Omega_\gamma := \Omega_{j,k}$, where $k \in \mathbb{Z}^d$. For instance, we can consider the univariate dyadic intervals in 1D, $d = 1$:

$$\Omega_\gamma = \Omega_{j,k} := [2^{-j}k, 2^{-j}(k+1)),\ \gamma \in S_j := \{(j,k) \text{ s.t. } j \in \mathbb{N}_0,\ k \in \mathbb{Z}\}. \tag{3.66}$$

The same follows for higher dimensions.

3.3.1 Multiresolution Analysis

We denote $\mathbf{U}_j := (u_\gamma)_{\gamma\in S_j}$ as the spatial representation of 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:

$$u_\gamma := |\Omega_\gamma|^{-1} \int_{\Omega_\gamma} u(t, \mathrm{x})\, \mathrm{dx}, \quad \mathrm{x} \in \mathbb{R}^d. \tag{3.67}$$

Data at different levels of discretization are related by two inter-level transformations which are defined as follows:

1. The *projection operator* P_{j-1}^j, which maps $\mathbf{U}_j$ to $\mathbf{U}_{j-1}$. It is obtained through exact averages computed at the finer level by

$$u_\gamma = |\Omega_\gamma|^{-1} \sum_{|\mu|=|\gamma|+1, \Omega_\mu \subset \Omega_\gamma} |\Omega_\mu| u_\mu. \tag{3.68}$$

   As far as grids are nested, this projection operator is *exact* and *unique* [Coh00].

2. The *prediction operator* P_j^{j-1}, which maps $\mathbf{U}_{j-1}$ to an approximation $\hat{\mathbf{U}}_j$ of $\mathbf{U}_j$. There is an infinite number of choices to define P_j^{j-1}, but at least two basic constraints are usually imposed [CKMP03]:

   (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

$$u_\gamma = |\Omega_\gamma|^{-1} \sum_{|\mu|=|\gamma|+1, \Omega_\mu \subset \Omega_\gamma} |\Omega_\mu| \hat{u}_\mu; \tag{3.69}$$

   i.e., one can retrieve the coarse cell averages from the predicted values:

$$P_{j-1}^j \circ P_j^{j-1} = \mathrm{Id}. \tag{3.70}$$

   In particular, this property implies that the stencil R_μ must contain the unique index γ such that $|\mu| = |\gamma| + 1$ and $\Omega_\mu \subset \Omega_\gamma$.

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

$$d_\mu := u_\mu - \hat{u}_\mu, \tag{3.71}$$

or in terms of inter-level operations:

$$d_\mu = u_\mu - P_{|\mu|}^{|\mu|-1} \circ P_{|\mu|-1}^{|\mu|} u_\mu. \tag{3.72}$$

The consistency assumption (3.69) and the definitions of the projection operator (3.68) and of the *detail* (3.71), imply

$$\sum_{|\mu|=|\gamma|+1, \Omega_\mu \subset \Omega_\gamma} |\Omega_\mu| d_\mu = 0. \tag{3.73}$$

We can then construct as shown in [CKMP03], a *detail vector* defined 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 (3.71) and (3.69), and to get a one-to-one correspondence:

$$\mathbf{U}_j \longleftrightarrow (\mathbf{U}_{j-1}, \mathbf{D}_j), \tag{3.74}$$

issued by operators P_{j-1}^j and P_j^{j-1}. For instance, in the univariate dyadic case (3.66) the detail vector is given by $\mathbf{D}_j = (d_{j,k})_{k \in \mathbb{Z}}$ with $d_{j,k} = u_{j,k} - \hat{u}_{j,k}$. By iteration of this decomposition, we finally obtain a *multi-scale representation* of $\mathbf{U}_J$ in terms of $\mathbf{M}_J = (\mathbf{U}_0, \mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_J)$:

$$\mathcal{M} : \mathbf{U}_J \longmapsto \mathbf{M}_J, \tag{3.75}$$

and similarly, its inverse $\mathcal{M}^{-1}$.

3.3.2 Wavelet Representation

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

$$\hat{u}_\mu := \sum_\gamma c_{\mu,\gamma} u_\gamma, \tag{3.76}$$

and $\mathcal{M}$ and $\mathcal{M}^{-1}$ are simply *changes of basis*. Based on the previous theoretical studies, we can then identify a wavelet representation [CKMP03] by defining for $\mathbf{U}_j$ the *dual scaling wavelet* $\tilde{\phi}_\gamma$ in (3.67):

$$u_\gamma := \langle u, \tilde{\phi}_\gamma \rangle, \tag{3.77}$$

such that

$$\tilde{\phi}_\gamma := |\Omega_\gamma|^{-1} \chi_{\Omega_\gamma}, \tag{3.78}$$

and where according to (3.76)

$$d_\mu := u_\mu - \hat{u}_\mu = \langle u, \tilde{\phi}_\mu \rangle - \sum_\gamma c_{\mu,\gamma} \langle u, \tilde{\phi}_\gamma \rangle = \langle u, \tilde{\psi}_\mu \rangle, \tag{3.79}$$

defines the *dual wavelet* $\tilde{\psi}_\mu$:

$$\tilde{\psi}_\mu := \tilde{\phi}_\mu - \sum_\gamma c_{\mu,\gamma} \tilde{\phi}_\gamma. \tag{3.80}$$

The *multiresolution representation* $\mathbf{M}_J$ can be then written as

$$\mathbf{M}_J = (d_\lambda)_{\lambda \in \nabla^J} = (\langle u, \tilde{\psi}_\lambda \rangle)_{\lambda \in \nabla^J}, \tag{3.81}$$

which corresponds exactly to $\mathbf{d}^J$ in the definition of the wavelet transform (3.28), where we have defined $\nabla^J := \bigcup_{j=0}^{J} \nabla_j$ with $\nabla_0 := S_0$, and where $d_\lambda = u_\lambda$ and $\tilde{\psi}_\lambda = \tilde{\phi}_\lambda$ if $\lambda \in \nabla_0$. With this representation, the multiresolution representation will be of order N if for all $u \in \mathbb{P}_{N-1}$ and for all $\lambda \in \nabla^J$, we have

$$\langle u, \tilde{\psi}_\lambda \rangle = d_\lambda = 0, \tag{3.82}$$

that is, if the wavelet $\tilde{\psi}_\lambda$ has N vanishing moments.
We can then build a prediction operator such that the associated dual wavelet is of order N. A standard procedure defines P_j^{j-1} based on polynomial interpolations of order N. For instance, for the univariate dyadic case (3.66), considering a centered stencil $(u_{j,k-M}, \ldots, u_{j,k+M})$ and the unique polynomial of degree $2M$ such that

$$2^j \int_{\Omega_{j,l}} p_{j,k}(x)\,\mathrm{d}x = u_{j,l}, \quad l = k-M, \ldots, k+M, \tag{3.83}$$

we can define the prediction approximation taking into account the consistency property (3.69) [CKMP03]:

$$\hat{u}_{j+1,2k} = 2^{j+1} \int_{\Omega_{j+1,2k}} p_{j,k}(x)\,\mathrm{d}x, \qquad \hat{u}_{j+1,2k+1} = 2^{j+1} \int_{\Omega_{j+1,2k+1}} p_{j,k}(x)\,\mathrm{d}x. \tag{3.84}$$

This procedure is exact for polynomials of degree $2M$, *i.e.*, it has accuracy order $N = 2M+1$. As an illustration, for $M = 1$, the prediction operator is explicitly given by:

$$\hat{u}_{j+1,2k} = u_{j,k} + \frac{1}{8}(u_{j,k-1} - u_{j,k+1}), \qquad \hat{u}_{j+1,2k+1} = u_{j,k} + \frac{1}{8}(u_{j,k+1} - u_{j,k-1}). \tag{3.85}$$

Higher order formula can be found in [Mül03], and will be introduced in Chapter 7. For Cartesian grids, extension to multidimensional polynomial interpolations is easily obtained by a tensorial product of the 1D operator [BH97, RSTB03]. Taking into account the estimate (3.42), local smoothness results in a stronger size reduction of the corresponding wavelet coefficients for higher approximation orders. Nevertheless, more accurate approximation formulae require also larger stencil, *e.g.*, for the dyadic 1D case (3.66) and according to (3.83):

$$R_{j,k} = \{(j-1, \lfloor k/2 \rfloor + l) \text{ s.t. } |l| \leq M\}. \tag{3.86}$$

3.3.3 Data Compression and Tree-Structured Data

One of the main interests of carrying out such a multi-scale decomposition is that this new representation (3.81), defines a whole set of regularity estimates all over the spatial domain, and thus a data compression might be achieved. Given a set of index $\Lambda \subset \nabla^J$, 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 interested in sets Λ obtained by *thresholding*:

$$\lambda \in \Lambda \text{ if } |d_\lambda| \geq \varepsilon_{|\lambda|}, \tag{3.87}$$

with the level-dependent threshold values $(\varepsilon_0, \varepsilon_1, \ldots, \varepsilon_J)$. Data compression is then achieved by discarding the cells whose *details* are not into Λ according to (3.87). Applying $\mathcal{T}_\Lambda$ on the multi-scale decomposition $\mathbf{M}_J$ of $\mathbf{U}_J$ amounts to building an approximation $\mathcal{A}_\Lambda \mathbf{U}_J$, where the operator $\mathcal{A}_\Lambda$ is given by

$$\mathcal{A}_\Lambda := \mathcal{M}^{-1} \mathcal{T}_\Lambda \mathcal{M}. \tag{3.88}$$

Taking into account that

$$u = \sum_{j \in \mathbb{N}_0} \sum_{|\lambda|=j} \langle u, \tilde{\psi}_\lambda \rangle \psi_\lambda, \tag{3.89}$$

it can be seen that for a given J, the array $\mathbf{\Psi}_{J,\lambda}$ with $|\lambda| \leq J$, corresponds to the cell averages of the primal wavelet ψ_λ at level J, *i.e.*, $\mathbf{\Psi}_{J,\lambda} = (\langle \psi_\lambda, \tilde{\phi}_\gamma \rangle)_{\gamma \in S_J}$. We can thus define the normalized norm ℓ^1 by

$$\|\mathbf{U}_J\| := 2^{-dJ} \sum_{\lambda \in S_J} |u_\lambda|, \tag{3.90}$$

which corresponds to the L^1-norm of a piecewise constant function. For $\mathbf{\Psi}_{J,\lambda}$, this yields

$$\|\mathbf{\Psi}_{J,\lambda}\| \leq C \|\psi_\lambda\|_{L^1} \leq C 2^{-d|\lambda|}. \tag{3.91}$$

And for the thresholded representation of $\mathbf{U}_J$ after applying $\mathcal{A}_\Lambda$ [CKMP03]:

$$\|\mathbf{U}_J - \mathcal{A}_\Lambda \mathbf{U}_J\| = \left\| \sum_{\lambda \neq \Lambda} d_\lambda \mathbf{\Psi}_{J,\lambda} \right\| \leq C \sum_{\lambda \neq \Lambda} |d_\lambda| 2^{-d|\lambda|} = C \sum_{|d_\lambda| \leq \varepsilon_{|\lambda|}} |d_\lambda| 2^{-d|\lambda|}, \tag{3.92}$$

where we see that the approximation error is bounded by the sum of the discarded details. Taking into account that $|d_\lambda| 2^{-d|\lambda|} \leq \varepsilon_{|\lambda|} 2^{-d|\lambda|}$, and considering a level-wise threshold parameter:

$$\varepsilon_j := 2^{dj} \eta, \tag{3.93}$$

the next bound follows[8]

$$\|\mathbf{U}_J - \mathcal{A}_\Lambda \mathbf{U}_J\| \leq C \#(\nabla^J) \eta = C \#(S_J) \eta \leq C 2^{dJ} \eta, \tag{3.94}$$

with the cautious assumption that all the d_λ such that $\lambda \notin \Lambda$, are equal to $\varepsilon_{|\lambda|}$, although many of them might be much smaller. The latter estimate (3.94) justifies the choice $\eta = 2^{-dJ} \varepsilon$ in order to have

$$\|\mathbf{U}_J - \mathcal{A}_\Lambda \mathbf{U}_J\| \leq C \varepsilon, \tag{3.95}$$

with the level-dependent threshold values proposed by Harten [Har94a, Har95]:

$$\varepsilon_j = 2^{d(j-J)} \varepsilon, \quad j \in [0, J], \tag{3.96}$$

where ε becomes the threshold value for the finest level J.
Nevertheless, allegedly useless details cannot be deliberately deleted because a certain data structure must be respected in order to perform the different computations associated with the multi-scale transformation itself, mainly the prediction operator. The set Λ must then exhibit a *graded tree* structure in order to guarantee the availability of cell values within the local prediction stencil. In order to define such a structure, we first introduce the following terminology:

- If $\Omega_\mu \subset \Omega_\lambda$ with $|\mu| = |\lambda| + 1$, we say that Ω_μ is a *child* of Ω_γ, and that Ω_γ is the *parent* of Ω_μ.
- By the definition of ∇_j, if Ω_λ has $N(\Omega_\lambda)$ children, $N(\Omega_\lambda) - 1$ of them are in $\nabla := \bigcup_{j \geq 0} \nabla_j$. We call these cells the *detail children* of Ω_λ.
- Moreover, we define the *leaves* $\mathrm{L}(\Lambda)$ of a *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.

[8] Symbol $\#(\,)$ denotes the cardinality of a set.

A set of indices $\Lambda \in \nabla$ is a tree if the following holds [CKMP03]:

- The fundamental level $\nabla_0 = S_0$ is contained in Λ.
- If Ω_μ and Ω_υ are detail children of the same Ω_λ, then $\mu \in \Lambda$ if $\upsilon \in \Lambda$.
- If Ω_λ is such that its detail children are in Λ, then the parent of Ω_λ has the same property.

For the 1D dyadic configuration (3.66), Λ is a tree if $\nabla_0 \in \Lambda$ and[9]

$$(j,k) \in \Lambda \Rightarrow (j-1, \lfloor k/2 \rfloor) \in \Lambda. \tag{3.97}$$

The set $R(\Lambda)$ contains the tree Λ plus the missing cells Ω_λ in the construction of ∇_j. A tree Λ is thus graded if for all $\mu \in R(\Lambda)$, the prediction stencil R_μ is contained in $R(\Lambda)$. Coming back to the dyadic example, Λ is a graded tree if

$$(j,k) \in R(\Lambda) \Rightarrow (j-1, \lfloor k/2 + l \rfloor) \in R(\Lambda), \quad |l| \leq M. \tag{3.98}$$

Defining Λ_ε as the smallest graded tree containing Λ given by (3.87), we introduce the corresponding tree approximation operator $\mathcal{A}_\varepsilon := \mathcal{A}_{\Lambda_\varepsilon} = \mathcal{M}^{-1}\mathcal{T}_{\Lambda_\varepsilon}\mathcal{M}$, following (3.88). Since $\Lambda \subset \Lambda_\varepsilon$, it follows directly that

$$\|\mathbf{U}_J - \mathcal{A}_\varepsilon \mathbf{U}_J\| \leq C\varepsilon. \tag{3.99}$$

3.3.4 Fully Adaptive Multiresolution Scheme

The previous multiresolution analysis for the solution of PDEs was first introduced by Harten [Har94a, Har95] in the context of hyperbolic conservation laws. Considering problem (3.63), we represent the numerical approximation of its solution at $(n+1)\Delta t$ by a finite volume scheme given by

$$\mathbf{V}_J^{n+1} = \mathbf{V}_J^n - \mathbf{B}_J^n, \tag{3.100}$$

considering a fine spatial discretization with grid S_J. The set $\mathbf{V}_J^n := (v_\lambda^n)_{\lambda \in S_J}$ is the array containing the numerical solution at time $n\Delta t$, whereas $\mathbf{B}_J^n := (b_\lambda^n)_{\lambda \in S_J}$ accounts for the numerical computation between $n\Delta t$ and $(n+1)\Delta t$ of fluxes and source terms on the finite volume discretization. In the original framework [Har94a, Har95, CKMP03, Mül03], $\mathbf{B}_J^n$ considers only the numerical fluxes of the hyperbolic operator.
The main idea is then to define a solution $\mathbf{U}_J^n$ that approximates $\mathbf{V}_J^n$, computed this time on the adaptive and therefore, compressed grid generated by the multiresolution transform. From a practical point of view, this data compression implies important savings in computational resources compared with the reference solution (3.100), performed on a uniform grid. However, it will naturally introduce an additional error:

$$a_n := \|\mathbf{U}_J^n - \mathbf{V}_J^n\|, \tag{3.101}$$

which should be controlled. To limit these approximation errors introduced by a multiresolution technique, Harten [Har94a, Har95] considered originally a compressed representation and computation only for the fluxes $\mathbf{B}_J^n$, taking into account that these ones represent the highest computational effort for the solution of conservation laws. Based on these ideas, Cohen *et al.* [CKMP03] introduced then the appropriate mathematical background for a *fully adaptive multiresolution scheme* which performs all computations on the adapted grid.

[9] Symbol $\lfloor\ \rfloor$ denotes the floor function, *i.e.*, it maps a real number to the largest integer, smaller than the given real number.

We have previously seen that the multi-scale approach is based on a solid mathematical framework that guarantees compressed representation of discretized functions within a prescribed tolerance. Nevertheless, for the solution of time evolution PDEs we have to consider moving meshes. The main difficulty of such a technique is thus to define appropriate criteria that guarantee the previous approximation estimates, even for time varying configurations. More precisely, given the graded tree Λ_ε^n issued from the application of $\mathcal{A}_\varepsilon$ to some $\mathbf{U}_J^n$, Harten proposed to enlarge Λ_ε^n into a larger graded tree $\tilde{\Lambda}_\varepsilon^{n+1}$ which should contain both Λ_ε^n and $\Lambda_\varepsilon^{n+1}$ such that

$$\left\|\mathbf{U}_J^n - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}}\mathbf{U}_J^n\right\| \leq C\varepsilon, \qquad \left\|\mathbf{U}_J^{n+1} - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}}\mathbf{U}_J^{n+1}\right\| \leq C\varepsilon, \tag{3.102}$$

according to (3.99), where $\mathbf{U}_J^{n+1} = \mathcal{E}_J\mathbf{U}_J^n$, with the time evolution operator $\mathcal{E}_J$ applied on grid S_J. The relations given in (3.102) are often referred as the *Harten's heuristics* and define a graded tree $\tilde{\Lambda}_\varepsilon^{n+1}$ such that the solution at both $n\Delta t$ and $(n+1)\Delta t$ are correctly represented. In practice, $\tilde{\Lambda}_\varepsilon^{n+1}$ should not be much larger than Λ_ε^n and it is usually derived from the size of the current detail coefficients. A *refinement* operator $\mathcal{R}$ is therefore introduced which allows us to construct $\tilde{\Lambda}_\varepsilon^{n+1}$ such that $\mathcal{T}_{\tilde{\Lambda}_\varepsilon^{n+1}} := \mathcal{R}\mathcal{T}_{\Lambda_\varepsilon^n}$.
For the univariate dyadic case (3.66), Harten proposed the following refinement criteria [Har94a, Har95]:

1. If (j,k) is in Λ_ε^n, then (j,k) and its $2\bar{k}$ neighbors at the same scale j are included in $\tilde{\Lambda}_\varepsilon^{n+1}$:

$$(j,k) \in R(\Lambda_\varepsilon^n) \Rightarrow (j,k+l) \in R(\tilde{\Lambda}_\varepsilon^{n+1}), \quad |l| \leq \bar{k}. \tag{3.103}$$

2. Current values of details are used to predict details at the next scale and thus new levels are locally added in $\tilde{\Lambda}_\varepsilon^{n+1}$:

$$|d_{j,k}^n|_{\Lambda_\varepsilon^n} \geq 2^{\bar{N}+1}\varepsilon_j \Rightarrow (j+1,2k+l) \in R(\tilde{\Lambda}_\varepsilon^{n+1}), \quad l = 0,1. \tag{3.104}$$

The first criterion takes into account a possibly moving solution where $\bar{k}$ can be chosen , for instance, as the support of the numerical flux evaluation, or can be based on the speed of propagation of the solutions. In particular for explicit time integration of hyperbolic problems, $\bar{k} = 1$ is often sufficient, issued from a standard CFL condition. On the other hand, the second criterion accounts for possible loss of smoothness during Δt and therefore, the need of adding scales to the tree. It is based on the estimate (3.42), for which a function u with locally $\mathcal{C}^s$ smoothness in the support $\Sigma_{j,k}$ of the dual wavelet, we have:

$$d_{j,k} \sim \begin{cases} 2^{-js}\,\partial_x^s u|_{\Sigma_{j,k}}, & 0 \leq s \leq N, \\ 2^{-jN}\,\partial_x^N u\big|_{\Sigma_{j,k}}, & s > N, \end{cases} \tag{3.105}$$

and hence

$$|d_{j+1,2k}| \approx 2^{-\bar{N}}|d_{j,k}|, \qquad \bar{N} = \min(s,N). \tag{3.106}$$

Harten proposed $1 \leq \bar{N} \leq N-1$; nevertheless, $\bar{N} = N$ is often considered [Mül03], where we recall that N stands for the vanishing moments of the dual wavelet and consequently for the accuracy order of the polynomial interpolations (3.84).
The Harten's multiresolution scheme computes then the cell averages by

$$\mathbf{U}_J^{n+1} = \mathbf{U}_J^n - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}}\mathbf{B}_J^n, \tag{3.107}$$

where $\mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}}\mathbf{B}_J^n$ is reconstructed on the finest grid J by the prediction operator, from the $(b_\lambda^n)_{\lambda\in \mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})}$ computed at the adapted grid, *i.e.*, at the leaves of the tree $\tilde{\Lambda}_\varepsilon^{n+1}$: $\mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})$.

For the flux evaluation and since the grids are nested, the computation on the adapted grid supposes then considering the cell averages at the finest level $(u_\lambda^n)_{\lambda\in S_J}$ but only those at the edges of the corresponding adapted grid. For hyperbolic problems, the reduction of the number of cells needed for flux computations involves an important gain in CPU time. On the other hand, the time evolution (3.107) takes place on the finest discretization S_J, and thus implies a complexity of $\mathcal{O}(N_J)$ operations for $N_J = \#(S_J)$ cells.
In this context, Harten's heuristics (3.102) implies naturally that

$$\left\|\mathbf{B}_J^n - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}}\mathbf{B}_J^n\right\| \leq C\varepsilon. \tag{3.108}$$

In particular, for $\varepsilon \to 0$, Harten's and reference solutions become closer and thus $a_n \to 0$ into (3.101). Therefore, the error a_n is usually referred as the *perturbation error*. Furthermore, supposing that the reference scheme satisfies for some fixed $C \geq 0$ and all $\mathbf{U}$, $\mathbf{V}$:

$$\|\mathcal{E}_J\mathbf{U} - \mathcal{E}_J\mathbf{V}\| \leq (1 + C\Delta t)\|\mathbf{U} - \mathbf{V}\|, \tag{3.109}$$

we can bound the error with respect to the reference finite volume scheme (3.100) by considering

$$\begin{aligned} a_n &\leq \|\mathcal{E}_J\mathbf{U}_J^{n-1} - \mathcal{E}_J\mathbf{V}_J^{n-1}\| + \|\mathcal{E}_J\mathbf{U}_J^{n-1} - \mathbf{U}_J^n\| \\ &\leq (1 + C\Delta t)a_{n-1} + c_n, \end{aligned} \tag{3.110}$$

where

$$c_n := \left\|\mathbf{B}_J^{n-1} - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^n}\mathbf{B}_J^{n-1}\right\|, \tag{3.111}$$

represents the *refinement error* that measures the error approximation of $\mathbf{B}_J^{n-1}$ on the adaptive set $\tilde{\Lambda}_\varepsilon^n$, refined from $\Lambda_\varepsilon^{n-1}$. We thus have

$$a_n \leq \sum_{i=0}^{n-1} C\varepsilon(1 + C\Delta t)^i = C\varepsilon\frac{(1 + C\Delta t)^n - 1}{\Delta t} \leq C\frac{\varepsilon}{\Delta t}\left(\mathrm{e}^{CT} - 1\right), \tag{3.112}$$

for some fixed time $T = n\Delta t$. In particular, if $C = 0$ into (3.109), we obtain [Har95]:

$$a_n \leq \sum_{i=1}^{n} c_i \leq Cn\varepsilon = C\frac{T}{\Delta t}\varepsilon, \tag{3.113}$$

which corresponds also to estimate (3.112) whenever T is small enough. In this way, the threshold parameter ε of the multiresolution decomposition becomes an accuracy tolerance of the corresponding adapted and compressed spatial representation.
In order to reduce the complexity of the Harten's scheme (3.107) and to further exploit the adapted multiresolution representations, Cohen *et al.* [CKMP03] introduced a *fully adaptive multiresolution scheme*, for which $\mathbf{U}_J^n$ is represented this time on the adapted grid, *i.e.*, on the leaves of the corresponding graded tree $\mathrm{L}(\Lambda_\varepsilon^n)$. As a consequence, the time evolution $\mathcal{E}_J$ is performed only on the adapted representation. Hence, considering the graded tree Λ_ε^n and $\mathbf{U}_J^n$ represented by the set $(u_\lambda^n)_{\lambda\in\mathrm{L}(\Lambda_\varepsilon^n)}$, or equivalently by $(d_\lambda^n)_{\lambda\in\Lambda_\varepsilon^n}$, we compute $\mathbf{U}_J^{n+1}$ applying the following procedure [CKMP03]:

- *Refinement.* The set $\tilde{\Lambda}_\varepsilon^{n+1}$ containing Λ_ε^n is constructed in order to predict the evolution of the solution and guarantee (3.102). The added set of u_λ^n with $\lambda \in \tilde{\Lambda}_\varepsilon^{n+1}\backslash\Lambda_\varepsilon^n$ are then computed by applying the inverse transform $\mathcal{M}^{-1}$, *i.e.*, by the prediction operator.

- *Time Evolution.* A numerical approximation $\tilde{\mathbf{U}}_J^{n+1}$ at time $(n+1)\Delta t$ is computed on $\mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})$ by
$$\tilde{u}_\lambda^{n+1} = u_\lambda^n - \tilde{b}_\lambda^n, \qquad \lambda \in \mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1}), \tag{3.114}$$
where the set $(\tilde{b}_\lambda^n)_{\lambda\in\mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})}$ is directly computed from the set $(u_\lambda^n)_{\lambda\in\mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})}$, on the adapted grid.

- *Thresholding.* The numerical solution $\mathbf{U}_J^{n+1}$ is finally obtained by thresholding $\tilde{\mathbf{U}}_J^{n+1}$ according to
$$\mathbf{U}_J^{n+1} = \mathcal{A}_\varepsilon \tilde{\mathbf{U}}_J^{n+1}, \tag{3.115}$$
which generates the new graded tree $\Lambda_\varepsilon^{n+1} \subset \tilde{\Lambda}_\varepsilon^{n+1}$.

In this way, the time evolution considers only $\mathcal{O}\left(\#\left(\mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})\right)\right)$ operations, performed on the adapted representation, and an important reduction of both CPU time and memory requirement is achieved.
While performing a fully adaptive computation, two crucial aspects must be taken into account. The first one is common to the standard Harten's scheme and deals with the proper definition of a refinement operator $\mathcal{R}$ such that the Harten's heuristics (3.102) is satisfied. One possibility considers , for instance, Harten's refinement criteria (3.103) and (3.104), as previously detailed. The second difficulty is to define the computation of the $(\tilde{b}_\lambda^n)_{\lambda\in\mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})}$ into (3.114), now that the u_λ^n are not always available at the finest level J, in opposition to the standard Harten's technique. For the flux evaluations $(\tilde{b}_\lambda^n)_{\lambda\in\mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})}$, there are mainly three alternatives [CKMP03, Mül03]:

- *Exact flux evaluation.* It reconstructs locally by prediction operator the cell averages u_λ^n at the finest level J along the edges of the adapted grids. The fluxes are then computed on the adapted grid using data at the finest level. This is the procedure initially proposed in [CKMP03] as *exact local reconstruction.* The reconstructed cells are usually called *phantom cells* and are temporarily created in order to perform the flux evaluations. The required number of phantoms is given by the number of cells in the flux evaluation stencils. The time evolution (3.114) is nevertheless computed only on the leaves. It is important to notice that the graduation of the tree must take also into account the cells in the prediction stencils needed to build the phantoms.

- *Locally unstructured flux evaluation.* It computes the fluxes along the edges of the adapted grids using the available data. Phantom cells are therefore not needed. This much more economic and simple strategy denoted as *direct evaluation* in [CKMP03], might nevertheless become quite inaccurate in the coarsest regions for low order flux computations.

- *Locally structured flux evaluation.* A third hybrid strategy which combines the previous ones, computes locally by prediction operator the phantom cells such that the fluxes can be evaluated using data on the same refinement level. This amounts to locally consider uniform grids and it is thus suited to numerical fluxes corresponding to structured grids (see, *e.g.*, [RSTB03]). As in the first case, these phantoms are temporarily kept and the graded tree must contain their corresponding prediction stencils. This strategy is naturally less expensive than an exact local reconstruction, and in practice no significant loss of accuracy is expected for phantoms computed based on high order reconstruction formulae and high order flux evaluation schemes [CKMP03, Mül03].

As for the Harten's scheme, we are interested in quantifying the additional error introduced in the general numerical solution (3.100) by considering this time the fully adaptive multiresolution scheme. That is, the perturbation error defined by (3.101), where $\mathbf{U}_J^n$ corresponds this time to the solution fully computed on the adapted grid and finally reconstructed on the finest grid S_J. Using the same cumulative error analysis as before, we then have

$$\begin{aligned} a_n &\leq \|\mathcal{E}_J\mathbf{U}_J^{n-1} - \mathcal{E}_J\mathbf{V}_J^{n-1}\| + \|\mathcal{E}_J\mathbf{U}_J^{n-1} - \mathbf{U}_J^n\| \\ &\leq (1 + C\Delta t)a_{n-1} + d_n, \end{aligned} \tag{3.116}$$

considering the Lipschitz condition (3.109). For a hyperbolic problem as considered in [CKMP03] with exact local reconstruction of the fluxes, we have that (3.114) can be written as

$$\tilde{\mathbf{U}}_J^{n+1} = \mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}}\mathcal{E}_J\mathbf{U}_J^n, \tag{3.117}$$

since $\mathcal{A}_{\tilde{\Lambda}_\varepsilon^{n+1}}\mathbf{U}_J^n = \mathbf{U}_J^n$, and the cumulative error $d_n := \|\mathcal{E}_J\mathbf{U}_J^{n-1} - \mathbf{U}_J^n\|$ is thus bounded by

$$d_n \leq \|\mathcal{E}_J\mathbf{U}_J^{n-1} - \tilde{\mathbf{U}}_J^n\| + \|\tilde{\mathbf{U}}_J^n - \mathbf{U}_J^n\| = c_n + t_n, \tag{3.118}$$

where

$$c_n := \|\mathcal{E}_J\mathbf{U}_J^{n-1} - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^n}\mathcal{E}_J\mathbf{U}_J^{n-1}\| = \|\mathbf{B}_J^{n-1} - \mathcal{A}_{\tilde{\Lambda}_\varepsilon^n}\mathbf{B}_J^{n-1}\|, \tag{3.119}$$

corresponds exactly to the same refinement error as in the previous Harten's scheme and it will be bounded by $C\varepsilon$ as long as the Harten's heuristics (3.102) is satisfied. On the other hand, according to (3.115)

$$t_n := \|\tilde{\mathbf{U}}_J^n - \mathbf{U}_J^n\| = \|\tilde{\mathbf{U}}_J^n - \mathcal{A}_\varepsilon\tilde{\mathbf{U}}_J^n\|, \tag{3.120}$$

corresponds to a *thresholding error* which is always bounded by $C\varepsilon$, based on the multiresolution decomposition, following (3.99). Under these considerations the previous error estimates (3.112) and (3.113) for the standard Harten's scheme hold also for the fully adaptive multiresolution scheme, and the perturbation error associated with the spatially compressed representation can be consequently tuned. In particular, a rigorous setting for the Harten's heuristics, that is, an appropriate refinement criteria that ensures (3.102) was introduced in [CKMP03] in order to mathematically prove that the refinement error c_n is indeed bounded by $C\varepsilon$ for both Harten's and the fully adaptive multiresolution schemes.

Nevertheless, although there is no rigorous mathematical proof on the reliability of the Harten's refinement criteria (3.103) and (3.104), they were shown to be sufficient in practice [CKMP03, Mül03, BLM04, HM10], and hence they are often used instead of the more sophisticated criteria introduced in [CKMP03]. More recently, Hovhannisyan & Müller extended the validity of estimates (3.112) and (3.113) to fully adaptive multiresolution for inhomogeneous conservation laws [HM10], by using a polynomial reconstruction of the fluxes (instead of the exact local reconstruction in [CKMP03]), and source approximations based on a quadrature rule. Similarly, the convergence of multiresolution approximations for nonlinear conservation laws towards the unique entropy solution was also demonstrated by Coquel *et al.* in [CPT11], for a multiresolution technique that employs local time stepping tools, introduced by Berger & Colella in [BC89] for classical AMR applications.

Let us remark that in all cases, the perturbation errors (3.112) and (3.113) measure the numerical errors associated with the multiresolution representation with respect to a reference discretized solution $\mathbf{V}_J^n$, at the finest grid. The numerical approximations related to the space discretization of problem (3.63) are therefore not contemplated, and hence the perturbation errors account only for the supplementary error introduced by the spatially compressed representation. The latter information constitutes nevertheless an indispensable tool to properly

evaluate the efficiency of the numerical method and to monitor the accuracy of the numerical simulations. Moreover, it represents a major advantage of multiresolution schemes with respect to other adaptive grid techniques. On the other hand, the effective space discretization error in the numerical solutions is settled in this case by the finite volume discretization, more precisely by its order, and will proportionally decrease with finer mesh representations for both uniform or adapted grids.

Part II

New Mathematical and Numerical Elements for the Solution of Multi-Scale Evolutionary PDEs

Introduction

In this part, we introduce a few mathematical and numerical tools for the numerical solution of stiff PDEs. Chapter 4 deals with the numerical simulation of multi-scale reacting fronts. It considers a new numerical strategy built upon the theoretical background previously presented in Part I, and mainly based on a dedicated splitting scheme for stiff reaction-diffusion problems coupled with a multiresolution decomposition. The performance and capabilities of the method are evaluated through 1D, 2D, and 3D numerical simulations of, respectively, traveling, spiral, and scroll chemical waves arising in the study of nonlinear chemical dynamics.

Chapter 5 introduces a new time stepping technique for the numerical integration of PDEs, based on an embedded splitting method. The numerical analysis is conducted as well as complementary theoretical and numerical studies. This new tool is implemented in the numerical strategy of Chapter 4, and extends its domain of application to more complex problems.

Finally, Chapter 6 explores a new numerical strategy based on operator splitting and the parareal algorithm, a numerical technique for parallelization of the time domain. A complete numerical analysis of the scheme is performed in the context of multi-scale waves with high spatial gradients. Numerical illustrations are also considered for chemical waves, and allow us to complete the theoretical study by a practical implementation and some numerical results.

Chapter 4

New Resolution Strategy for Multi-Scale Reactions Waves

This chapter describes a new numerical strategy that has been developed in this work for the solution of multi-scale reaction waves, modeled by stiff reaction-diffusion systems. The numerical tools introduced in this chapter are based on the previous mathematical and theoretical background presented in Part I. In this way, we consider the mathematical description of the splitting errors discussed in Chapter 1 in the context of stiff PDEs, as well as the time integration of the split subsystems, the reaction and diffusion problems, taking into account the numerical methods described in Chapter 2. The space adaptive multiresolution techniques detailed in Chapter 3 are also taken into consideration. As a consequence, this new numerical strategy aim at merging theoretical and numerical aspects mentioned in the previous chapters and in the referred literature, with a few new elements into a novel and robust solver for the numerical simulation of reaction waves. The resulting method constitutes the main core of the global numerical strategy implemented in the MBARETE code. In particular, this study was published in SIAM Journal on Scientific Computing [DMD+12], and has been presented as well during the CLEI 36th Latin American Informatics Conference, Asunción, Paraguay (2010) [DML+11].

4.1 Context and Motivation

In this part, we tackle the numerical simulation of reaction-diffusion equations modeling multi-scale reaction waves. This type of problem 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 steep gradients in the reaction fronts, spatially very localized. We have previously seen in the General Introduction that many numerical strategies have been developed in the past years to cope with these problems. Since the numerical solution of the fully coupled problem is most of the time out of reach and involves important computational investments, and on the other hand the appropriate definition of reduced models is usually difficult to establish and furthermore is intrinsically problem dependent, alternative numerical strategies has been also developed. In this context, IMEX methods allow us to overcome some important numerical restrictions and yield very efficient implementation as briefly discussed in the General Introduction. Nevertheless, in this work we are particularly interested in time operator splitting methods described with more details in Chapter 1. Let us remark that the main advantage of IMEX schemes with respect to splitting techniques is that no splitting errors are introduced. On the other hand, both IMEX and split-

ting methods are rather low order schemes since the order conditions for higher order IMEX schemes are often cumbersome for practical implementations (see remarks in [VS04, VSH04]).

A splitting procedure allows us to consider dedicated solvers for each subproblem and thus reduces strongly the computational complexity of the numerical implementations. From a general theoretical point of view, to guarantee the accuracy of the solution obtained by a splitting scheme, the splitting time steps used for the independent solution of each subproblem should be either taken of the order of the fastest scales included in the problem, or related to some particular stability constraint, or simply chosen sufficiently small in order to yield negligible splitting errors. Many splitting strategies that aim at reducing the computational resources for the numerical simulation of stiff problems, are often built such that the inner and performing solvers for the subproblems ensure the global accuracy order of the splitting scheme. The order is experimentally evaluated in simple configurations, and the splitting time steps are then chosen in practice such that the numerical simulations deliver qualitatively good results for more complex phenomena (see, *e.g.*, [OB01, NK05, SPN06b, RP08]). Unfortunately, in the context of multi-scale stiff problems for which stability constraints or the accurate resolution of fast scales might become critical, the previous alternatives to choose the splitting time step imply an important reduction of the potential performance and accuracy of the numerical strategy.
We have seen in Chapter 1 that more rigorous studies for stiff configurations (mainly [DM04, DDLM07]), succeeded to better characterize the behavior of splitting schemes with splitting time steps much larger than the fastest scales of the problem, a feature that is common in certain applications, and hence allowed us to complement the classical theoretical background of these methods already available for asymptotic regimes. Based on these recent mathematical studies, a new time operator splitting approach has been introduced in this work, which exploits these theoretical results and enlarges their practical extent with an important gain of efficiency. Contrary to classical splitting strategies that consider arbitrarily small or stability related splitting time steps, or in which the attention is rather concentrated on the inner solvers, in the present strategy the choice of the splitting time steps assumes the leading role in terms of accuracy of the time integration process. This choice coupled with dedicated solvers aims at enhancing the performance of the method but within a prescribed accuracy. In this way, the splitting time steps are explicitly defined to settle the overall accuracy of the numerical solution, independent of the time scale spectrum of the problem, and on the sole basis of the physics of the global phenomenon and its decoupling capabilities. Its choice is therefore not related to any stability requirement of the numerical methods used to integrate each subsystem, even if strong stiffness is present.

In order to guarantee the validity of the mathematical background and to exploit the specificities of each split subsystem, dedicated time integration methods must be chosen to deal with the stiffness associated with each one of them, in a separate manner. In Chapter 2, we have briefly characterized this stiffness and introduced some performing numerical methods that have been recently developed. Based on these previous studies, the present strategy considers a high order method like Radau5 [HW96], based on implicit Runge-Kutta schemes for stiff ODEs, to solve the reaction term; and on the other hand, another high order method like ROCK4 [Abd02], based on explicit stabilized Runge-Kutta schemes, to solve the diffusion problem. The entire spectrum of temporal scales associated with each subproblem can be efficiently solved in an independent way, while the global accuracy of the time integration scheme is mainly set by the splitting scheme through the choice of the splitting time step. An effective decoupling of the time scale spectrum is achieved with an important improvement of efficiency

whenever a broad decoupling is physically possible. The different physical or numerical time scales associated with each problem can be then isolated and treated independently by each time integration method and the splitting composition.
This novel approach to choose the time integration solvers allows us also to decouple the numerical errors associated with each problem and with the splitting scheme. Contrarily, most splitting applications consider solvers of the same order than the splitting scheme (see, *e.g.,* [KNW99, DB00, NK05]), for which an interaction occurs between the several numerical errors associated with each subproblem and the splitting technique. On the other hand, in many splitting strategies (like the previously cited or [SLGS03]) dedicated implicit multi-step methods such as VODE [BBH89] or LSODE [Hin80, Hin83] have been widely implemented with sufficiently fine tolerances for the numerical treatment of stiff reactive terms, also in opposition to the current approach that considers only one-step integration schemes[1]. In this way, the time integration strategy conceived in this work under the mentioned criteria is quite general and can be implemented in principle to any stiff reaction-diffusion system, whereas the same ideas can be naturally extended to consider other phenomena, for instance, convection, in the modeling equations. In particular, the time-stepping features of both Radau5 and ROCK4 with accuracy control allow us to properly solve the initial fast transients previously discussed. The same follows for other splitting applications either with the less efficient multi-step VODE or LSODE solvers for the reaction problem[2], or with a much more expensive implicit L-stable solver as suggested by [RS05a, RS09] for the diffusion term[3].

Particular advantages can be drawn in the context of self-similar propagating waves for which constant splitting time steps are enough to capture the global dynamics of the phenomenon, and a strong decoupling of the time spectrum can be achieved. To the best of our knowledge, almost all the splitting implementations in the literature developed for the numerical simulation of realistic stiff problems, were restricted to constant splitting time steps. Considering the adequate choice of higher order numerical methods with adaptive time stepping based on accuracy criteria, the main error of the time integration in this implementation is piloted by the splitting scheme, and it is thus settled by the splitting time step even for stiff cases as recently proved [DM04, DDLM07]. The latter can be computed out of numerical error estimates or based on a physical feature such as the profile of the wavefront or its propagation speed, which usually feature scales much slower than the inner reactive or diffusive characteristic times. In practice, the estimated splitting errors are always evaluated for the semi-discretized problem, *i.e.,* the reference coupled and split solutions are considered discretized in space, so that the space discretization error is not taken into account and consequently there is a decoupling of time and space numerical errors. Such an approach allows us to perform the latter numerical evaluations on sufficiently representative and feasible computational domains, to then extrapolate the results to larger domains out of reach of standard methods, mainly because the splitting errors will no longer depend on the spatial discretization. The global numerical error is then given by both the splitting and the space discretization errors, separately evaluated, whereas from a sufficiently fine spatial discretization it will be indeed set by the splitting errors, controlled by the splitting time step.

[1] For instance, a detailed study was presented in [VG01] that demonstrated the important loss of efficiency of LSODE to treat stiff PDEs in splitting configurations, mainly because of the expensive starting procedure of the multi-step scheme at each splitting time step.

[2] We recall that a multi-step method cannot be L-stable with an order higher than two [Dah63].

[3] Actually, Ropp & Shadid considered in [RS05a, RS09] a splitting technique in which the splitting time step is equal to the diffusion one, without error control. Therefore, the A-stability of the numerical method chosen to treat the diffusion problem ensures only stability but not the necessary damping for stiff configurations.

Being aware of the interest of adaptive mesh techniques for problems exhibiting locally steep spatial gradients and that consequently finer spatial discretizations become feasible, the performance of the time integration strategy can be notably improved by a spatial multiresolution technique based on Harten's pioneering work [Har94a, Har95] and the fully multiresolution schemes [CKMP03, Mül03], previously described in Chapter 3. One of the main advantages of such a method is that for a given semi-discretized problem, the error introduced by the corresponding compressed spatial representation can be better controlled. Even though a rigorous mathematical proof of multiresolution errors for parabolic problems is not yet available, the mathematical background of wavelet decomposition allows us to justify and validate such a choice (see, *e.g.*, [RS02, RSTB03, BRBSS08, BBRBS09]). We have thus defined in this work a few simple criteria to properly couple for the first time a space adaptive multiresolution approach with dedicated time operator splitting for stiff problems.
As a consequence, this work introduces mainly two new elements in the context of multiresolution applications. First, an implicit time integration apt to handle stiff problems by means of an operator splitting approach, considering that an implicit integration over the whole computational domain involves a high degree of algorithmic complexity for these spatially adapted representations and more sophisticated data structures. Second, a new approach in terms of decoupling of numerical errors is considered in which we have the splitting errors defined on the adapted grid, the spatial multiresolution representation errors settled by the thresholding parameter, and the spatial discretization errors. In this way, for a problem represented with a spatial discretization limited mainly by the computational resources, the proposed MR/splitting strategy allows us to track the corresponding numerical errors of the simulation, introduced by the numerical methods of solution, taking into account that the space discretization errors are difficult to evaluate in practice unless an analytical solution is available. Furthermore, the global error of the numerical simulation will be successfully monitored if a sufficiently fine spatial discretization is achieved.

The main goal of the proposed numerical strategy is thus to perform computationally performing simulations of multi-scale reaction waves within a prescribed accuracy and with standard computational resources.

4.2 Construction of the Numerical Strategy

The proposed numerical strategy handles general reaction-diffusion systems of type (1.52). Nevertheless, in order to simplify the presentation we shall consider linear diagonal diffusion like for system (1.53). From a practical point of view and complementing the theoretical framework of chapters 1 and 2, we perform a spatial discretization of problem (1.53), sufficiently fine to guarantee a good description of the physical problem and such that the spatial discretization errors are negligible with respect to the ones coming from the numerical time integration. We obtain thus the semi-discretized initial value problem:

$$\left.\begin{aligned} &\mathrm{d}_t \mathbf{U} - \mathbf{L}\,\mathbf{U} = \mathbf{F}\left(\mathbf{U}\right), \quad t > 0, \\ &\mathbf{U}(0) = \mathbf{U}^0, \end{aligned}\right\} \tag{4.1}$$

where $\mathbf{L}$ corresponds to the discretization of the Laplacian operator with the diffusion coefficients D within. $\mathbf{U}$ and $\mathbf{F}\left(\mathbf{U}\right)$ are arranged component-wise all over the discretized spatial domain, and stand, respectively, for the discretization of $U : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^m$, and $F : \mathbb{R}^m \to \mathbb{R}^m$.

4.2.1 Time Operator Splitting

Considering a standard decoupling of the diffusion and reaction parts of (4.1), we denote $\mathcal{X}^{\Delta t}\mathbf{U}^0$ as the numerical solution of the diffusion equation[4]:

$$\mathrm{d}_t\mathbf{U}_D - \mathbf{L}\,\mathbf{U}_D = 0, \quad t > 0, \tag{4.2}$$

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:

$$\mathrm{d}_t\mathbf{U}_R = \mathbf{F}\left(\mathbf{U}_R\right), \quad t > 0, \tag{4.3}$$

with initial data $\mathbf{U}_R(0) = \mathbf{U}^0$. According to the definitions introduced in Chapter 1, the two Lie approximation formulae of the solution of the semi-discretized system (4.1) are defined by

$$\mathcal{L}_1^{\Delta t}\mathbf{U}^0 = \mathcal{X}^{\Delta t}\mathcal{Y}^{\Delta t}\mathbf{U}^0, \quad \mathcal{L}_2^{\Delta t}\mathbf{U}^0 = \mathcal{Y}^{\Delta t}\mathcal{X}^{\Delta t}\mathbf{U}^0, \tag{4.4}$$

whereas the two Strang approximation formulae [Str63, Str68] are given by

$$\mathcal{S}_1^{\Delta t}\mathbf{U}^0 = \mathcal{X}^{\Delta t/2}\mathcal{Y}^{\Delta t}\mathcal{X}^{\Delta t/2}\mathbf{U}^0, \quad \mathcal{S}_2^{\Delta t}\mathbf{U}^0 = \mathcal{Y}^{\Delta t/2}\mathcal{X}^{\Delta t}\mathcal{Y}^{\Delta t/2}\mathbf{U}^0, \tag{4.5}$$

where Δt stands for the splitting time step.
We recall that the Lie formulae (4.4) (resp., Strang formulae (4.5)) are approximations of order 1 (resp., 2) of the exact solution of (4.1) in the case where $\mathcal{X}^{\Delta t}$ and $\mathcal{Y}^{\Delta t}$ correspond to the exact solutions $\mathbf{X}^{\Delta t}$ and $\mathbf{Y}^{\Delta t}$ of problems (4.2) and (4.3). We have seen in Chapter 1 that the standard orders achieved with a Lie or Strang scheme are no longer valid when we consider very stiff reactive or diffusive terms with large splitting time steps. Furthermore, if the fastest time scales play a leading role in the global physics of the phenomenon, then the solution obtained by means of a splitting composition scheme will surely fail to capture the global dynamics of the phenomenon, unless we consider splitting time steps small enough to resolve such scales. In the opposite case where these fast scales are not directly related to the physical evolution of the phenomenon, larger splitting time steps might be considered, but order reductions may then appear due to short-life transients associated with the fastest variables. This is usually the case for propagating reaction waves where, for instance, the speed of propagation is much slower than the chemical scales. In this context, we recall that better performances are expected while ending the splitting scheme by the time integration of the reaction part (4.3) [DM04]:

$$\left\|T^{\Delta t}U_0 - \mathcal{L}_1^{\Delta t}U_0\right\|_{L^2} = \mathcal{O}(\Delta t), \tag{4.6}$$

$$\left\|T^{\Delta t}U_0 - \mathcal{S}_1^{\Delta t}U_0\right\|_{L^2} = \mathcal{O}(\Delta t), \tag{4.7}$$

$$\left\|T^{\Delta t}U_0 - \mathcal{L}_2^{\Delta t}U_0\right\|_{L^2} = \mathcal{O}(\Delta t^2), \tag{4.8}$$

$$\left\|T^{\Delta t}U_0 - \mathcal{S}_2^{\Delta t}U_0\right\|_{L^2} = \mathcal{O}(\Delta t^3), \tag{4.9}$$

where $T^{\Delta t}U_0$ stands for the exact solution of (1.53) with linear diagonal diffusion, and the fastest scales are present in the reactive term. In a general case, the splitting scheme should always end with the part involving the fastest time scales of the phenomenon (see a numerical case in [DDL+12]).

[4] Throughout this work, a standard centered second order space discretization is considered for the diffusive terms.

On the other hand, we have seen that order reductions might also arise from space multi-scale phenomena due to steep spatial gradients whenever large splitting time steps are considered [DDLM07, DDM11]:

$$\left\|T^{\Delta t}U_0 - \mathcal{L}^{\Delta t}U_0\right\|_{L^2} \propto \left(\|\partial_x U_0\|_{L^2}\Delta t^2, \|U_0\|_{L^2}\Delta t^{1.5}\right), \tag{4.10}$$

$$\left\|T^{\Delta t}U_0 - \mathcal{S}^{\Delta t}U_0\right\|_{L^2} \propto \left(\|\partial_x U_0\|_{L^2}\Delta t^3, \|U_0\|_{L^2}\Delta t^2\right), \tag{4.11}$$

for which the first terms are more relevant when Δt is small and the second ones when Δt is not small enough and $\|\partial_x U_0\|_{L^2}$ is very high. We notice that in both cases the Lie (4.4) and Strang (4.5) formulae are built with the exact solutions of each subproblem, as in the theoretical framework of Chapter 1. These theoretical studies allow us to depict more precisely the numerical behavior of the splitting techniques for a broader range of splitting time steps and strong stiffness, and thus help us to select among the various splitting alternatives, depending on the nature of the problem.

4.2.2 High Order Dedicated Time Integration Methods

The choice of suitable time integration methods for subsystems (4.2) and (4.3) is mandatory not only to guarantee the previous theoretical estimates, but also to take advantage of the particular features of each independent subproblem and to solve them with reasonable resources, as accurately as possible. In particular, the proposed splitting technique considers high order dedicated integration methods for each subproblem in order to properly solve the fastest time scales associated with each one and in such a way that the main source of error is led by the operator splitting error. Then, the control of the accuracy of the time integration is ruled by the splitting scheme by means of a splitting time step that is chosen to describe the global physical phenomenon within a required level of accuracy, even for stiff configurations.

Time Integration of the Reaction: Radau5

We have seen in Chapter 2 that Radau5 [HW96] is a fifth order implicit Runge-Kutta method for which order conditions proved by Butcher [But64a] are satisfied up to the order 5. Its stability function is generated by a collocation procedure with the Radau quadrature formulae [Ehl73] that guarantees A- and L-stability properties, so that very stiff systems of ODEs might be solved without any stability problem.
Nevertheless, nonlinear systems must be solved throughout the time integration process because of this implicit character. Even if the Newton's method to handle such computations is highly optimized, these procedures become very expensive for large systems and important memory requirements are needed in order to carry them out. As a consequence, the size of the system of equations to be solved is strongly limited by the computing resources. In a splitting scheme context, we easily overcome this difficulty because the reactive term (4.3) is a system of ODEs without spatial coupling. Therefore, a local approach node by node is adopted where the memory requirements are only set by the number of local unknowns, which usually does not exceed conventional memory resources. This approach is also posed as an embarrassingly parallel problem where no data exchange is needed among nodes, that therefore yields optimal load balancing on shared memory architectures as we shall see in forthcoming chapters [DMD+11b, DDD+12].
A very important feature of the Radau5 solver is that precious computing time is saved because of its adaptive time stepping strategy. The latter guarantees a requested accuracy and at the same time allows us to discriminate stiff zones from regular ones so that small time steps are

only considered for stiff behaviors, related either to the modeling equations or to the spurious transients introduced by the splitting technique [VBvLS96, SVdZ+98, VSBH99]. In a splitting context, the reaction time integration step Δt_R will be then adapted only at nodes where the reaction phenomenon takes place, yielding local reaction time steps potentially much smaller than the splitting time step, whereas the global time advancement of the solution given by the splitting time step Δt will not be limited by these physical or numerical fast time scales. For multi-scale reaction waves, this adaptation happens in a very low percentage of the spatial domain, usually only in the neighborhood of the wavefront. Contrarily, larger time steps are considered at nodes with a chemistry at (partial) equilibrium in which the splitting should not introduce artificial transients[5]. This local time stepping with no data exchange and no reconstruction of intermediate values would not be possible if we integrated the entire reaction-diffusion system (4.1) at once, as in a fully implicit or IMEX coupled solution scheme.

Time Integration of the Diffusion: ROCK4

One of the most important advantages of ROCK4 [Abd02] is its explicit character and hence the simplicity of its implementation. No sophisticated linear algebra tools are needed since no solution of linear systems is required The numerical integration is thus based on simple matrix-vector products. Nevertheless, the computation cost relies directly on the requested quantity of such products, *i.e.,* the number of internal stages s needed within one time integration step of the diffusion problem, Δt_D, inside each splitting time step Δt. The memory requirements are also reduced as a consequence of its explicit scheme. Nevertheless we must keep in mind that these requirements increase proportionally with the number of nodes considered over the spatial domain.

The ROCK solver [Abd02] features also dynamic time step adaptation so that Δt_D is chosen in order to guarantee a prescribed accuracy of computations. This is also a suitable characteristic for splitting configurations to properly handle high frequency modes in the solution [RSO04]. As detailed in Chapter 2, ROCK4 is formally a fourth order *stabilized* explicit Runge-Kutta method and such methods feature extended stability domain along the negative real axis [Ver96]. For a general diffusion problem such as $\mathbf{v}' = \mathbf{g}(\mathbf{v})$, the number of stages s needed to guarantee the stability of computations for a given time step Δt_D, is directly related to the spectral radius $\rho(\partial \mathbf{g}/\partial \mathbf{v})$ as long as the latter is dominated by real negative eigenvalues. For a given Δt_D needed to guarantee the accuracy of the integration, the minimum number of stages s required for stability is computed by the ROCK4 solver through (2.102), written now as

$$0.35 \cdot s^2 \geq \Delta t_D \, \rho \left(\frac{\partial \mathbf{g}}{\partial \mathbf{v}}(\mathbf{v}) \right), \tag{4.12}$$

which extends quadratically on s, the stability domain of the method along the negative axis. According to the construction of the ROCK4 scheme [Abd02], at least $s = 5$ internal stages are required.

The method 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 (4.2) with only negative real eigenvalues and constant spectral radius $\rho(\mathbf{L})$, as analyzed in Chapter 2 for the general heat equation (2.14). An important gain of efficiency is obtained in this case because the discretized diffusion operator has a sparse matrix structure that yields more performing matrix-vector products. In our particular applications, the diffusive phenomenon has a leading role of propagator of perturbations over

[5] Let us notice that chemical equilibrium regions coincide usually with spatially homogeneous distributions so that spatial operators like diffusion or convection should preserve constancy.

the (partial) equilibrium nodes that result in excitation of the reactive schemes, and thus the propagation of the reaction wave. The resulting self-similar character implies that the needed number of stages will remain practically constant throughout the evolution of the phenomenon. Finally, the spectral radius must be previously estimated (for example, using the Gershgoring theorem or even numerically, as proposed by the ROCK4 solver by means of a nonlinear power method).
Notice that for a general and more complex diffusion problem like the one in system (1.52), such a stabilized explicit method will remain suitable as long as the spectral radius is dominated by real negative eigenvalues. This may be the case in some applications[6] but remains a problem dependent feature. Otherwise, a less efficient standard high order explicit scheme should be considered if an implicit solver imposes excessive computational resources[7]. The same follows for stiffer diffusion problems that require a high number of inner stages s to ensure stability. Nevertheless, in this latter case the explicit formulation of ROCK4 might justify its use in terms of memory requirements, and on the other hand, a standard explicit scheme would certainly be more expensive in CPU time. In this work, we will mainly consider linear diffusion problems with constant or time dependent coefficients[8]. More complex configuration like multi-species diffusion will be certainly studied in the future.
Once again, the implementation of this diffusion solver for the entire reaction-diffusion system (4.1) will not be appropriate under either theoretical or practical considerations, and highlights the inherited advantages of the time operator splitting. In particular, solving within a prescribed tolerance the diffusion problem may also yield Δt_D much smaller than the splitting time step Δt, and thus the global time advancement of the solution given by Δt will not be necessarily limited by the diffusive time scales but by the global features of the coupled problem.

4.2.3 Space Adaptive Multiresolution Technique

We are concerned with the propagation of reacting wavefronts for which important reactive activity as well as steep spatial gradients are localized phenomena. This implies that if we consider the solution of the reactive problem (4.3), a considerable amount of computing time is spent on nodes that are practically at (partial) equilibrium. We will see in Chapter 9 that for a numerical simulation with complex source mechanisms on a uniform grid, 60 % of the computing time is spent on nodes with very reduced chemical activity [DDD+12]. Moreover, there is no need to represent these quasi-stationary regions with the same spatial discretization needed to describe the reaction front, so that the diffusion problem (4.2) might also be solved over a smaller number of nodes. An adapted mesh obtained by the multiresolution analysis, described in Chapter 3, allows us to discriminate the various space scales of the phenomenon, and turns out to be a very convenient solution to overcome these difficulties.
A fully adaptive multiresolution technique based on [CKMP03, Mül03] is then coupled with the previous dedicated time operator splitting strategy. In this way, considering a finite volume discretization for problem (4.1) on a finest grid S_J, we define as before a set of nested meshes S_j on which problem (4.1) is represented, for $j = 0, 1, \cdots, J$, from the coarsest to the finest grid. By performing a multiresolution analysis on the solution $\mathbf{U}$, the operator $\mathcal{A}_{\Lambda_\varepsilon} = \mathcal{M}^{-1}\mathcal{T}_{\Lambda_\varepsilon}\mathcal{M}$ yields an effective data compression because $\mathbf{U}$ is no longer represented on the finest grid

[6] See, *e.g.*, [NK05, SPN06b] for diffusion problems with detailed multi-species transport coming from combustion modeling, solved with the RKC solver [SSV97] for which no stability problems were reported.
[7] In particular, the implicit scheme should be either L-stable based on the previously mentioned studies [RS05a, RS09], or sufficiently stable with error control features.
[8] The diffusion problem for plasma applications detailed in Chapter 11 present, for instance, time and space dependent coefficients.

S_J as $\mathbf{U}_J$, but on Λ_ε, where we recall that Λ_ε is the smallest graded tree containing the thresholded tree Λ, defined in (3.87) and constructed based on the values of the details. In this way, the numerical solution $\mathbf{U}^n$ at time $n\Delta t$ is represented on an adapted grid by the set $(u_\lambda^n)_{\lambda\in \mathrm{L}(\Lambda_\varepsilon^n)}$, *i.e.,* on the leaves of the adapted tree, on which the time evolution strategy is performed.
Taking into account that the time evolution is performed on a fixed adapted grid during each splitting time step Δt, a crucial aspect is to define an appropriate refinement operator $\mathcal{R}$. This one must generate a set $\tilde{\Lambda}_\varepsilon^{n+1}$ containing Λ_ε^n, on which the time integration is computed, such that $\tilde{\Lambda}_\varepsilon^{n+1}$ is adapted for describing the solution at both $n\Delta t$ and $(n+1)\Delta t$. These conditions are known as the Harten's heuristics (3.102). In the proposed numerical implementation, the operator $\mathcal{R}$ refines the adapted grid based on the values of the details as follows:

- By enlarging uniformly with $\bar{k}$ cells in each direction the refined regions of Λ_ε^n in order to predict the propagation of the solution, according to the first Harten's refinement criterion [Har94a, Har95]; and
- By refining all leaves of Λ_ε^n such that $|d_\lambda| \geq \varepsilon_{|\lambda|}$ with $\lambda \in \mathrm{L}(\Lambda_\varepsilon^n)$. This procedure adds 2^d cells everywhere and it is hence equivalent to adding one more level all over Λ_ε^n.

The latter enlarged region is also known in the literature as the *security* or *safety zone*, and it has been considered in many multiresolution applications (see, *e.g.,* [RSTB03, BRBS10]). The added u_λ^n with $\lambda \in \tilde{\Lambda}_\varepsilon^{n+1}\backslash\Lambda_\varepsilon^n$ can be constructed by the prediction operator. Comparing with the standard Harten's refinement criteria for the univariate dyadic case (3.66), the first criterion (3.103) is kept, whereas the second one given by (3.104) is written now as

$$|d_{j,k}^n|_{\mathrm{L}(\Lambda_\varepsilon^n)} \geq \varepsilon_j \Rightarrow (j+1, 2k+l) \in R(\tilde{\Lambda}_\varepsilon^{n+1}), \quad l = 0, 1. \tag{4.13}$$

Harten related the $\bar{k}$ cells in each direction to the support of the numerical flux evaluation, and $\bar{k} = 1$ was adopted in practice since the time step is limited by a standard CFL condition [Har95]. In our numerical strategy, there is not such a stability restriction on the splitting time step Δt, and the $\bar{k}$ cells are rather used to allow us larger time evolution steps while ensuring an appropriate spatial representation. These refinement criteria are rather conservative. Nevertheless, they are simple and completely avoid unrefined resolution taking into account the choice of the time evolution steps detailed in the following and the propagating nature of reaction waves at finite speed. In particular, the latter feature guarantees that a given adapted grid propagates along with the waves and no more that one grid level needs to be added at each time iteration.
An important theoretical result and one of the main advantages of a multiresolution approach as detailed in Chapter 3, is that if we denote by $\mathbf{V}_J^n := (v_\lambda^n)_{\lambda\in S_J}$ the solution fully computed on the finest grid, and by $\mathbf{U}_J^n$, the solution reconstructed on the finest grid that used adaptive multiresolution (keeping in mind that the time integration was really performed on the *leaves* $\mathrm{L}(\tilde{\Lambda}_\varepsilon^n)$ of a compressed representation of $\mathbf{U}^n$); then, for a fixed time $T = n\Delta t$, it can be shown that the error introduced by the compressed spatial representation or perturbation error is given by

$$\|\mathbf{U}_J^n - \mathbf{V}_J^n\| \propto n\varepsilon. \tag{4.14}$$

This result was first stated by Harten in [Har95] for hyperbolic problems in an L^1-norm, where the multiresolution decomposition was used to accelerate the flux evaluations for a time integration performed on the finest uniform grid. Later on, (4.14) was mathematically proved in [CKMP03] under more rigorous constraints for the refinement criteria and for the computation of the fluxes also for hyperbolic problems in an L^1-norm, this time for a fully multiresolution

scheme where all operations were performed on the adapted grid, as detailed in Chapter 3. Nevertheless, in the same spirit of these works and based on the mathematical background of wavelet decomposition, we shall consider (4.14) but with an L^2-norm, more suited for parabolic problems, as also considered, for instance, in [RS02, RSTB03, BRBSS08, BBRBS09]. Even though a fully mathematical proof is still required for this case, numerical validations were already provided. In this context, the level-dependent threshold values (3.96) proposed by Harten in [Har94a, Har95], should be written as

$$\varepsilon_j = 2^{\frac{d}{2}(j-J)}\varepsilon, \quad j \in [0, J], \tag{4.15}$$

where ε is the threshold value for the finest level J, in order to be consistent with the standard definition of the L^2-norm.
Concerning the flux computations associated with the spatial operators, we consider the locally structured flux evaluation detailed in [Mül03] and [RSTB03], and previously described in Chapter 3. The latter takes into account ghost cells called phantoms that are locally and temporarily added to the tree in order to always compute the numerical fluxes of diffusion and convection operators at the highest grid level between two neighboring cells, following the procedure introduced in [RSTB03] for finite volume discretizations. Although there is no rigorous mathematical analysis, this procedure constitutes a much more efficient alternative to the exact flux evaluations as shown in [CKMP03, Mül03] with essentially the same accuracy for high order reconstruction schemes. In this implementation as in standard structured meshes, we consider projection operators built upon polynomial interpolation of at least order $N = 3$, as for the 1D case (3.84). The case of locally unstructured flux evaluations should be even more efficient in terms of computational complexity and consequently, in computational resources, but it was not implemented in this work in order to guarantee at least locally, the accuracy order of the flux schemes in all directions at the same level.

4.3 Summary of the Numerical Strategy

The numerical solution scheme can be summarized as follows, according to the multiresolution terminology adopted in Chapter 3:

$$(u^n_\lambda)_{\lambda\in \mathrm{L}(\tilde\Lambda^n_\varepsilon)} \xrightarrow{\mathcal{M}} (u^n_{\lambda_0}, d^n_\lambda)_{\lambda\in\tilde\Lambda^n_\varepsilon} \tag{4.16}$$

$$(u^n_{\lambda_0}, d^n_\lambda)_{\lambda\in\tilde\Lambda^n_\varepsilon} \xrightarrow{\mathcal{T}_{\Lambda^n_\varepsilon}} (u^n_{\lambda_0}, d^n_\lambda)_{\lambda\in\Lambda^n_\varepsilon} \tag{4.17}$$

$$(u^n_{\lambda_0}, d^n_\lambda)_{\lambda\in\Lambda^n_\varepsilon} \xrightarrow{\mathcal{R}} (u^n_{\lambda_0}, d^n_\lambda)_{\lambda\in\tilde\Lambda^{n+1}_\varepsilon} \tag{4.18}$$

$$(u^n_{\lambda_0}, d^n_\lambda)_{\lambda\in\tilde\Lambda^{n+1}_\varepsilon} \xrightarrow{\mathcal{M}^{-1}} (u^n_\lambda)_{\lambda\in \mathrm{L}(\tilde\Lambda^{n+1}_\varepsilon)} \tag{4.19}$$

$$(u^n_\lambda)_{\lambda\in \mathrm{L}(\tilde\Lambda^{n+1}_\varepsilon)} \xrightarrow{\mathcal{S}^{\Delta t}} (u^{n+1}_\lambda)_{\lambda\in \mathrm{L}(\tilde\Lambda^{n+1}_\varepsilon)} \tag{4.20}$$

We recall that the set $(u^n_{\lambda_0})_{\lambda\in\Lambda}$ is defined as the set of roots of some Λ, *i.e.*, all $\lambda \in \Lambda$ such that $|\lambda| = 0$ or $\lambda \in S_0$. Considering $\mathbf{U}^n$ represented on the adapted grid $\mathrm{L}(\tilde\Lambda^n_\varepsilon)$, the first step (4.16) performs a multiresolution transform from the physical to the wavelet basis space. A data compression is achieved by means of the threshold operator $\mathcal{T}_{\Lambda^n_\varepsilon}$ according to (3.87) and respecting a graded structure. This amounts to build the smallest graded tree Λ^n_ε containing the thresholded tree Λ defined into (3.87). The latter tree is enlarged to create $\tilde\Lambda^{n+1}_\varepsilon$ in

(4.18) according to the refinement criteria previously introduced. The inverse multiresolution transform is finally applied in (4.19) in order to retrieve the physical variables on the adapted grid $\mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})$, on which the time integration is performed in (4.20).
The algorithm can be schematically represented as

$$\mathbf{U}^{n+1} = \mathcal{S}^{\Delta t}\mathcal{M}^{-1}\mathcal{R}\mathcal{T}_{\Lambda_\varepsilon^n}\mathcal{M}\mathbf{U}^n, \tag{4.21}$$

with the compressed representations of $\mathbf{U}^{n+1}$ and $\mathbf{U}^n$ given 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, and the Strang operator splitting $\mathcal{S}^{\Delta t}$ given by one of the formulae (4.5) as time evolution operator. The procedure (4.16)–(4.20) follows the standard fully adaptive multiresolution scheme as presented in [Mül03] with a general time evolution operator in (4.20). One might add a last thresholding step as originally proposed in [CKMP03] to perform the numerical analysis (see Chapter 3), to represent the solution on $\mathrm{L}(\Lambda_\varepsilon^{n+1})$, instead of $\mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})$, by applying

$$(u_\lambda^{n+1})_{\lambda\in\mathrm{L}(\tilde{\Lambda}_\varepsilon^{n+1})} \xrightarrow{\mathcal{A}_{\Lambda_\varepsilon^{n+1}}} (u_\lambda^{n+1})_{\lambda\in\mathrm{L}(\Lambda_\varepsilon^{n+1})} \tag{4.22}$$

where $\mathcal{A}_{\Lambda_\varepsilon^{n+1}} = \mathcal{M}^{-1}\mathcal{T}_{\Lambda_\varepsilon^{n+1}}\mathcal{M}$. This last thresholding step implies slightly higher data compression for the solution outputs, and it is implicitly performed anyway into the recursive procedure (4.16)–(4.20).
For $n = 0$, the initial condition should be represented on $\mathrm{L}(\tilde{\Lambda}_\varepsilon^0)$ in step (4.16), which can usually be the finest grid, *i.e.,* for all $\lambda \in \mathrm{L}(\tilde{\Lambda}_\varepsilon^0)$, $|\lambda| = J$ or equivalently $\lambda \in S_J$. Nevertheless, this is not possible for large domains simulations, in which case, the initial condition is computed on an intermediate grid level j^0 such that for all $\lambda \in \mathrm{L}(\tilde{\Lambda}_\varepsilon^0)$, $|\lambda| = j^0$ or $\lambda \in S_{j^0}$. The solution is then refined and recomputed over the next finer level after a thresholding process. This procedure is recursively applied until the pre-established finest level J is reached, or until no cell needs to be refined for a given threshold parameter. The general procedure is explained in details in [Mül03], and will be presented in Chapter 7 for the current implementation.

4.4 Computation of the Splitting Time Step

The efficiency of the previous time integration strategy both in terms of accuracy and computational resources, relies mainly on the selected splitting time step. In order to properly couple the space adaptive multiresolution with the splitting scheme, some criteria will be introduced in the following in the context of propagating multi-scale waves. One of the main novelties is that the splitting time step is set by the desired level of accuracy in the resolution of the wave speed, the wave profile, both, or any other parameter, depending on the problem and considering that each split subsystem is solved exactly or sufficiently accurately. It is thus only depending on the global physics of the phenomenon we want to describe and therefore, on the degree of decoupling we can achieve between the various subsystems within a prescribed error tolerance.
Considering the semi-discretized problem (4.1) with a sufficiently fine spatial discretization, if an accurate reference wave solution $\mathbf{U}$, or the corresponding wavefront speed v, can be computed either numerically or based on theoretical/analytical estimates, then the approximated solution $\mathbf{U}_{split}$ of speed v_{split}, computed with a splitting time step Δt and an operator splitting technique with exact integration of the subsystems, must verify:

$$\mathrm{E}_p = \left\|\mathbf{U} - \mathbf{U}_{split}\right\|_{L^2} \leq \eta_p, \quad \mathrm{E}_v = |v - v_{split}|/v \leq \eta_v, \tag{4.23}$$

where η_p and η_v are accuracy tolerances for the profile and velocity errors: E_p and E_v, respectively. The profile error E_p should be evaluated superposing both $\mathbf{U}$ and $\mathbf{U}_{split}$. However, a

simpler and more practical strategy would just evaluate the L^2-error at some fixed time t^*:

$$\mathrm{E} = \left\| \mathbf{U}(t^*) - \mathbf{U}_{split}(t^*) \right\|_{L^2} \leq \eta, \tag{4.24}$$

in which case, both profile and velocity errors are simultaneously considered. For problems with variables of different order of magnitude, all these L^2-norms (E_p and E) that are independently evaluated for each variable, should be normalized by an appropriate scaling factor for the corresponding evaluated variable. In the error evaluations conducted in this work, we have considered a scaling factor for each variable given by the corresponding maximum value of the variable. Sometimes, it is also useful to scale the spatial representation with another scaling factor, given this time by the size of the computational domain.
Notice that in order to remain coherent with the previous constraints and also to guarantee an accurate resolution of the reaction and diffusion problems, the corresponding accuracy tolerances η_{Radau5} and η_{ROCK4} of these solvers must verify:

$$\eta_{Radau5}, \eta_{ROCK4} < \min\{\eta_p, \eta_v, \eta\}. \tag{4.25}$$

In this way, we can isolate the various integration errors and guarantee that the overall time integration error is practically given by the splitting scheme approximation. In particular, an evaluation of the sole splitting error allows us to decouple the time scale spectrum of the problem whenever this is possible, contrary to classical applications of time operator splitting methods for which the splitting time step is directly settled by the fastest time scale of the phenomenon. We therefore extend the use of these methods to splitting time steps defined by the global coupling scales, potentially larger than the fastest physical or numerical scales. This is a direct consequence of previous mathematical studies that demonstrated that even though there will possibly be an order reduction for time or space stiff problems and large splitting time steps, the splitting schemes will still consistently approximate the coupled resolution with an error piloted by the splitting time step [DM04, DDLM07].
We have established so far the criteria to handle time integration errors given by the splitting procedure. Nevertheless, the proposed strategy combines this splitting approach with a space multiresolution adaptive technique so that the approximation error introduced by the latter must be also taken into account. According to (4.14), we consider the following error bound for a fixed time $T = n\Delta t$:

$$\left\| \mathbf{U}^{MR}_{split} - \mathbf{U}^{J}_{split} \right\|_{L^2} \leq Cn\varepsilon, \tag{4.26}$$

for some positive C, where $\mathbf{U}^{MR}_{split}$ is the MR/splitting solution at $n\Delta t$ reconstructed on the finest grid J, which corresponds to the spatial discretization of the semi-discretized problem (4.1). A basic constraint to assume the validity of (4.26), as in the hyperbolic case, is that the propagating locally refined spatial gradients remain into the corresponding fine regions during each time step evolution Δt, *i.e.*, the Harten's heuristics (3.102) are satisfied. For a given spatial discretization $(\Delta x, \Delta y, \Delta z)$ corresponding to the maximum J level, the splitting time step must be bounded by a maximum splitting time step $\Delta t_{\max}$ computed by:

$$\Delta t \leq \Delta t_{\max} = \min\left\{ \frac{\hat{k}\Delta x}{v_x}, \frac{\hat{k}\Delta y}{v_y}, \frac{\hat{k}\Delta z}{v_z} \right\}, \tag{4.27}$$

for the directional components (v_x, v_y, v_z) of the wavefront speed and where $\hat{k} = 2 + \bar{k}$ stands for the refined region in which the gradients might propagate without losing spatial resolution. The refined region is obtained with the refinement criteria previously detailed for which one finer level is added everywhere (2 cells at J in each direction since the corresponding 2^d children are simultaneously present in the safety layer) and $\bar{k}$ cells on the same level. We remark that

the refined region $\hat{k}$ might be even larger in practice because if one cell is a leaf, the remaining $2^d - 1$ "brother-cells" issued from the same parent-cell are also leaves, even if some of them might not be necessary according to the threshold criterion (4.15).
We recall that there are no stability restrictions on the splitting time Δt since the internal time integration solvers deal independently with these issues, and thus the splitting time step is uniquely associated with the numerical accuracy of the computations. In this context, the bound $\Delta t_{\max}$ is necessary only for adapted grids, and allows us simply to ensure an appropriate spatial representation of the steepest gradients and to decouple time and space approximation errors. That is, without this bound the propagating front might considerably leave the refined regions and the time integration performed on the resulting coarser cells might thus introduce complementary numerical errors into the numerical time integration process. The latter coupling of space and time features usually results into qualitatively different physical behaviors, for instance, in the velocity of propagation, which in practice allows us to identify an inappropriate choice of the splitting time step for a given adapted grid. Finally, increasing the parameter $\bar{k}$ in the refinement criteria allows us to further enlarge the refined regions if larger splitting time steps are desired.
In the case of propagating wavefronts, a constant splitting time step based on a prescribed accuracy is more than reasonable, whereas the bound (4.27) guarantees a proper coupling between the space and time numerical methods. If no theoretical hints of the wave profile or velocity exist, the computation of a reference solution is usually very expensive but still feasible for 1D or relatively smaller computational domains. These simulations might give some insights in the behavior of the numerical methods in order to extrapolate the conclusions to larger or multi-dimensional problems according to a standard numerical procedure in scientific computing. On the other hand, the speed of the wavefront needed to establish (4.27) can be always approximated by directional 1D measurements taken either from fully coupled 1D configurations or, for instance, from feasible multi-dimensional MR/splitting solutions for which the accuracy tolerances are tightened.
However, if a more precise error control is required, or if we are faced with highly unsteady problems, the previous procedure can be dynamically implemented, for instance, by estimating the local splitting error as explained for stiff systems of ODEs in Chapter 2, without any need to compute a reference solution. Based on these estimates, the splitting time steps can be dynamically computed within the prescribed accuracy tolerance η into (4.24). All these issues will be discussed in details in the next chapter. However, these procedures introduce naturally an overhead which might not be justified in the simplified case of propagating waves. A hybrid strategy that considers a constant splitting time step computed out of a local error estimate, performed on the fly, can be seen as the most convenient solution.

4.5 Basic Features of the Algorithm Implementation

A dynamic graded tree structure is used to represent the data in the computer memory. This kind of data structure has been used in many multiresolution applications (see, *e.g.*, [RSTB03]) and other dedicated data structures have also been developed [BMV09, BMMS09, BBB+10]. 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. Figure 4.1 shows an example of a graded tree structure in 1D.
This data structure is suitable for 1D, 2D and 3D Cartesian geometries, whereas the basic element of the structure is the cell itself, which consists of a set of geometric and physical values, along with pointers to its parent, their children, and the contiguous cells in each

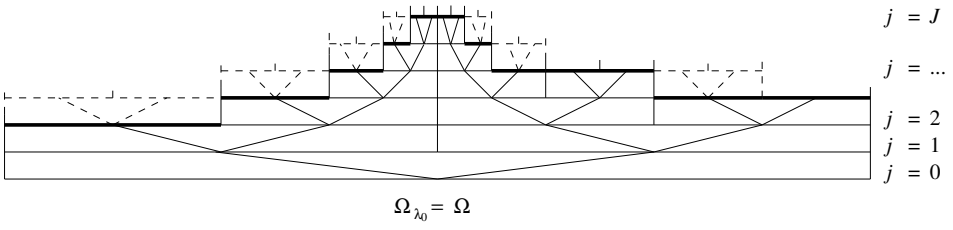

Fig. 4.1: *Example of 1D graded tree structure. Cells and links to their corresponding children are indicated (solid lines), as well as the leaves (solid bold lines) and the phantoms (dashed lines).*

dimension, the *neighbors*. The N_{R} roots correspond to the basis of the tree, Ω_{λ_0}, whereas the leaves are the upper elements with no children in the tree. In d dimensions, a parent-cell at a level j has at most 2^d children cells at level $j+1$. In this particular implementation, we impose that all 2^d children issued from the same parent-cell are simultaneously present if at least one child among them is needed. The maximum number of leaves N_{L} on which the solution might be represented is given by $N_{\mathrm{L}} = N_{\mathrm{R}} 2^{dJ}$, which must correspond to the number of cells on the finest grid N_J. Additionally, the maximum number of cells N_Λ in the tree is given by

$$N_\Lambda = N_{\mathrm{R}} \frac{2^{d(J+1)} - 1}{2^d - 1}. \tag{4.28}$$

Further details concerning the algorithm implementation will be discussed in Chapter 7.

4.5.1 Algorithm Scheme

The global algorithm can be summarized as:

1. **Initialization.**
   - **Initialization of parameters:** *e.g.,* maximum and minimum grid levels, domain size, number of roots.
   - **Initialization of the mesh structure:**
     - creation of nested grids;
     - initialization of parameters of each cell from the roots, *e.g.,* position, coordinates, level threshold value ε_j;
     - definition of children and neighbors from the roots.
   - **Computation of initial solution** at an intermediate grid level and recursive refinement and computation at most up to the maximum level.
2. **Loop in time.**
   - **Computation of cell values throughout the tree:** projection operator P_{j-1}^{j} from leaves towards roots.
   - **Computation of details:** operator $\mathcal{M}$ from roots towards leaves.
   - **Thresholding and graduation:** operator $\mathcal{T}_{\Lambda_\varepsilon}$ throughout the tree.
   - **Refinement of the tree:** operator $\mathcal{R}$ throughout the tree.
   - **Computation of cell values from details:** operator $\mathcal{M}^{-1}$ from roots towards leaves.

- **Creation of phantom cells:** needed for diffusion time step.
- **Time integration:** performed only on the leaves. For Strang operator splitting $\mathcal{S}_2^{\Delta t}$ in (4.5):
  - reaction half time step, time integration by Radau5 cell by cell;
  - diffusion time step, time integration by ROCK4 considering phantom cells at the grid level boundaries, computed by prediction operator;
  - reaction half time step, time integration by Radau5 cell by cell.

3. **Output.**

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

The general algorithm will be described with more details in Chapter 7, we note nevertheless that the computation of the cell values throughout the tree as well as the details are simultaneously evaluated by the multiresolution transform $\mathcal{M}$. The same follows for the thresholding and refinement operators $\mathcal{T}_{\Lambda_\varepsilon}$ and $\mathcal{R}$, which are also simultaneously performed. Let us remark that during the thresholding process, cells are only labeled according to (4.15), but no cell is removed nor is its detail set to zero, because a particular cell might be retained afterward under the refinement or graduation criteria. The phantom cells are created before the time integration process in the code, in order to separate the multiresolution and time integration operations. The values on the phantoms are nevertheless computed at each internal stage of ROCK4 by the prediction operator after locally updating their prediction stencils with the projection operator.

4.6 Numerical Simulations

In what follows, we present some numerical illustrations of the proposed strategy. A problem coming from nonlinear chemical dynamics is described and treated. The performance of the method is then discussed in the context of multi-dimensional simulations. All simulations were performed on an AMD-Shanghai 2.7 GHz processor with memory capacity of 32 GB.

4.6.1 Mathematical Model: The Belousov-Zhabotinski Reaction

We are concerned with the numerical approximation of a model for the Belousov-Zhabotinski reaction, a catalyzed oxidation of an organic species by acid bromated ion (see [EP98] for more details and illustrations). We thus consider the model detailed in [GS94] and coming from the classic work of [FKN72] which takes into account three species: hypobromous acid $HBrO_2$, bromide ions Br^-, and cerium (IV). Denoting by $a = [\text{Ce(IV)}]$, $b = [HBrO_2]$, and $c = [Br^-]$, we obtain a very stiff system of three PDEs:

$$\left.\begin{aligned} \partial_t a - D_a\,\partial_{\mathrm{x}}^2 a &= \frac{1}{\mu}(-qa - ab + fc), \\ \partial_t b - D_b\,\partial_{\mathrm{x}}^2 b &= \frac{1}{\epsilon}\left(qa - ab + b(1-b)\right), \\ \partial_t c - D_c\,\partial_{\mathrm{x}}^2 c &= b - c, \end{aligned}\right\} \tag{4.29}$$

$\mathrm{x} \in \mathbb{R}^d$, with diffusion coefficients D_a, D_b and D_c, and the real positive parameters: f, small q, and small ϵ and μ, such that $\mu \ll \epsilon \ll 1$.
The dynamical system associated with this system models reactive excitable media with a large time scale spectrum (see [GS94] for more details). The spatial configuration with the

addition of diffusion develops propagating wavefronts with steep spatial gradients. Hence, this model presents all the difficulties associated with a stiff multi-scale configuration. The advantages of applying a splitting strategy to this problem have already been studied and presented in [DDM03]. In what follows, we will briefly consider a 1D case of (4.29) in order to illustrate the errors of splitting schemes for stiff problems, then 2D and 3D configurations will be implemented.

4.6.2 1D BZ Equation

Let us perform a short illustrating study of the behavior of splitting schemes when dealing with stiff problems, as explained in Chapter 1. In the BZ model, stiffness is given by fast time scales as well as steep spatial gradients. We consider then a 1D configuration of problem (4.29) with homogeneous Neumann boundary conditions and the following parameters, taken from [GS94]:

$$\epsilon = 10^{-2}, \qquad \mu = 10^{-5}, \qquad f = 3, \qquad q = 2 \times 10^{-4}, \tag{4.30}$$

with diffusion coefficients:

$$D_a = 1, \qquad D_b = 1, \qquad D_c = 0.6, \tag{4.31}$$

for a space region of $[0, 80]$. A sufficiently fine uniform mesh of 4000 points is considered while the exact solution $T^{\Delta t}$ is approximated by a reference or *quasi-exact* solution $\mathcal{T}^{\Delta t}$ of the semi-discretized coupled reaction-diffusion problem (4.29), performed by Radau5 with very fine tolerances. The splitting schemes (4.4) and (4.5) consider Radau5 and ROCK4 as integration methods for the reaction and diffusion problems.

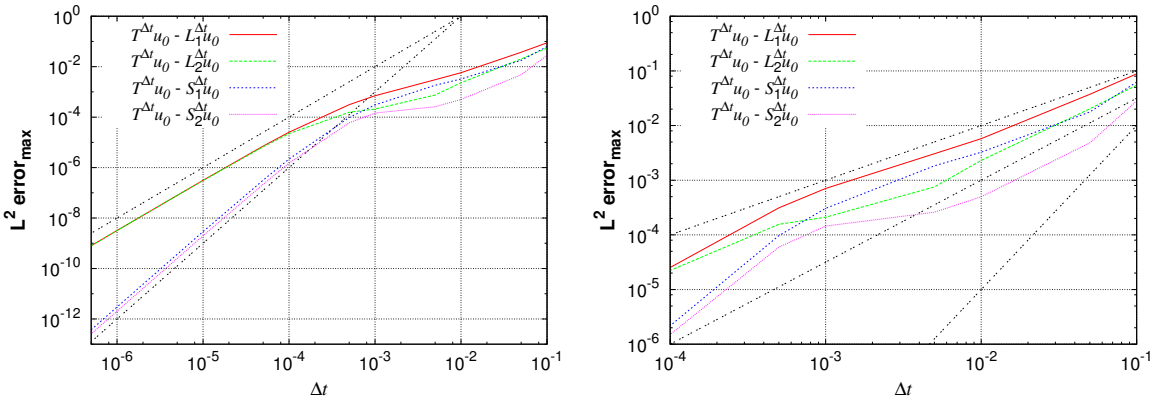

Fig. 4.2: *1D BZ equation. Local errors for Lie and Strang splitting schemes. Lines with slopes* 2 *and* 3 *are depicted (left), and slopes* 1, 1.5 *and* 3 *in the zoomed loss order region (right).*

Figure 4.2 shows the local errors after one splitting time step, where fully developed and converged waves are taken as initial condition. The maximum L^2 errors account for the maximum value between the computed and normalized local errors for a, b, and c variables. In these numerical tests, the maximum error corresponds to variable b. It can be seen that both Lie and Strang schemes have asymptoticly local order 2 and 3 for small time steps. Nevertheless, for larger time steps, the results in [DM04] and [DDLM07] describe better the numerical behavior of these schemes:

- For $\mathcal{L}_1^{\Delta t}$ in (4.4), order 2 drops to 1 as predicted by (4.6); whereas

- For $\mathcal{L}_2^{\Delta t}$, we see the influence of spatial gradients as predicted by (4.10), and thus order 1.5 is recovered after some transition phase.

The same conclusions are drawn for the Strang schemes (4.5):

- The order of $\mathcal{S}_1^{\Delta t}$ drops from 3 to 1 according to (4.7); while
- For $\mathcal{S}_2^{\Delta t}$, we see the influence of steep spatial gradients that alter the order 3 given by (4.9).

Finally, in all cases and for large splitting time steps, the reaction ending schemes show better behaviors for larger splitting time steps, according to [DM04]. In particular, $\mathcal{L}_2^{\Delta t}$ behaves even better than $\mathcal{S}_1^{\Delta t}$, whereas $\mathcal{S}_2^{\Delta t}$ is the best alternative for all time steps.

4.6.3 2D BZ Equation

Let us consider the 2D configuration of problem (4.29) with homogeneous Neumann boundary conditions, and the following parameters taken from [JSW89] and [DDM03]:

$$\epsilon = 10^{-2}, \qquad \mu = 10^{-5}, \qquad f = 1.6, \qquad q = 2 \times 10^{-3}, \tag{4.32}$$

with diffusion coefficients:

$$D_a = 2.5 \times 10^{-3}, \qquad D_b = 2.5 \times 10^{-3}, \qquad D_c = 1.5 \times 10^{-3}. \tag{4.33}$$

The phenomenon is studied over a time domain of $[0, 4]$ and a space region of $[0, 1]^2$. In the following, we will first consider a 2D computational domain with uniform mesh of 256^2, for which the coupled and split solutions of the semi-discretized problem derived from (4.29) are rather expensive but still feasible. The main idea is to compare these solutions with the ones computed by the proposed MR/splitting procedure in order to analyze the splitting and multiresolution errors regarding the corresponding accuracy tolerances previously detailed. In a second step and based on these results, we will evaluate the performance of the method by considering larger computational domains and 3D problems.
We thus consider 8 nested dyadic grids with $N_J = 2^{2\times 8} = 65536 = 256^2$ cells on the finest grid $J = 8$, and define:

- A reference or *quasi-exact* solution $\mathbf{U}_{qe}^J$ as the solution of the semi-discretized coupled reaction-diffusion problem (4.29) on the finest mesh J, performed by ROCK4 with very fine tolerance $\eta_{ROCK4} = 10^{-14}$; whereas
- Based on the previous 1D case, the split solution $\mathbf{U}_{split}^J$ uses the RDR Strang $\mathcal{S}_2^{\Delta t}$ scheme as time integration method of the semi-discretized problem (4.29) with Radau5 for the time integration of the reaction term and ROCK4 for the diffusive part, $\eta_{Radau5} = \eta_{ROCK4} = 10^{-5}$.
- Finally, with the same splitting time integration strategy $\mathcal{S}_2^{\Delta t}$, we consider the proposed MR/splitting solution $\mathbf{U}_{split}^{MR}$.

For the coupled reaction-diffusion problem and with the previous parameters, the spectral radius of the Jacobian of the reaction term into (4.29) is usually dominated by the negative real parts of the associated eigenvalues, although imaginary parts are also present. Therefore only fine tolerances that yield sufficiently small time steps allow us to fully guarantee the stability of the ROCK4 scheme whenever the imaginary part appears, considering the reduced stability domains of these methods along the imaginary axis. The main limitation to directly perform such a computation with the Radau5 solver comes from its important memory requirements.

Data Initialization

The initialization of the problem is based on [JSW89] for the two-variable model of (4.29) with b and c:

$$\left.\begin{aligned} \partial_t b - D_b\,\partial_{\mathrm{x}}^2 b &= \frac{1}{\epsilon}\left(b(1-b) + \frac{f(q-b)c}{q+b}\right), \\ \partial_t c - D_c\,\partial_{\mathrm{x}}^2 c &= b - c, \end{aligned}\right\} \tag{4.34}$$

by taking $\mu \to 0$ into the evolution equation of a, and thus

$$a = \frac{fc}{q+b} \tag{4.35}$$

into the evolution equations of b and c into (4.29). An approximation of the steady state values of the dynamical system associated with the two-variable problem (4.34) is given by

$$b_{ss} = c_{ss} = q\,\frac{f+1}{f-1}, \tag{4.36}$$

and the initial condition can be then computed with

$$b = \begin{cases} 0.8 & \text{if } 0 < \theta < \arctan(0.3), \\ b_{ss} & \text{elsewhere}, \end{cases} \tag{4.37}$$

$$c = c_{ss} + \frac{\theta}{8\pi f}, \tag{4.38}$$

and (4.35), where θ is a polar coordinate angle considering as origin $(0.5, 0.5)$ into the domain $[0,1]^2$.

Computation of the Splitting Time Step

In order to illustrate the choice of the appropriate splitting time step Δt, we set an accuracy tolerance of $\eta = 10^{-2}$, considering the normalized L^2-errors (4.24) with the quasi-exact and splitting solutions. Figure 4.3 shows these errors evaluated at final time $t^* = 4$ for all three variables.

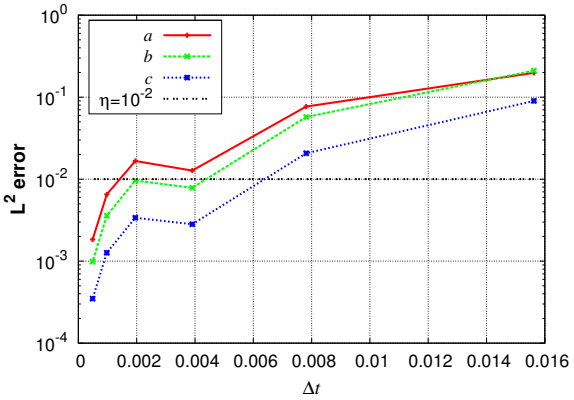

Fig. 4.3: *2D BZ spiral waves. Normalized L^2-errors for several splitting time steps Δt at final time $t^* = 4$ according to (4.24) and prefixed accuracy tolerance of $\eta = 10^{-2}$. Uniform grid of 256^2.*

A rounded value of splitting time step of $\Delta t = 4/1024 \approx 3.91 \times 10^{-3}$ is finally chosen for which L^2-errors are close to η for all three variables, and times $t^* \in [0, 4]$ into (4.24). In general all these computations over the whole time domain are not necessary, and from a practical point of view and for the simulation of propagating waves, we can consider a much less expensive procedure that evaluates the local errors after one splitting time step starting from an intermediate solution for which the waves are fully developed. A rather large accuracy tolerance η was considered in order to show the potential decoupling of time steps for reaction, diffusion, and the time operator splitting needed to solve the problem within the prescribed tolerance:

- The imposed tolerance for the reaction resolution implies time steps varying from 8.88×10^{-5} to $\Delta t/2 \approx 1.95 \times 10^{-3}$, for points located, respectively, in the neighborhood of the reactive front and the reduced chemical activity regions.
- The selected tolerance for ROCK4 yields time steps Δt_D relatively constant of about $6.5 - 8 \times 10^{-4}$, *i.e.*, 5 or 6 diffusion time steps within each splitting time step Δt.

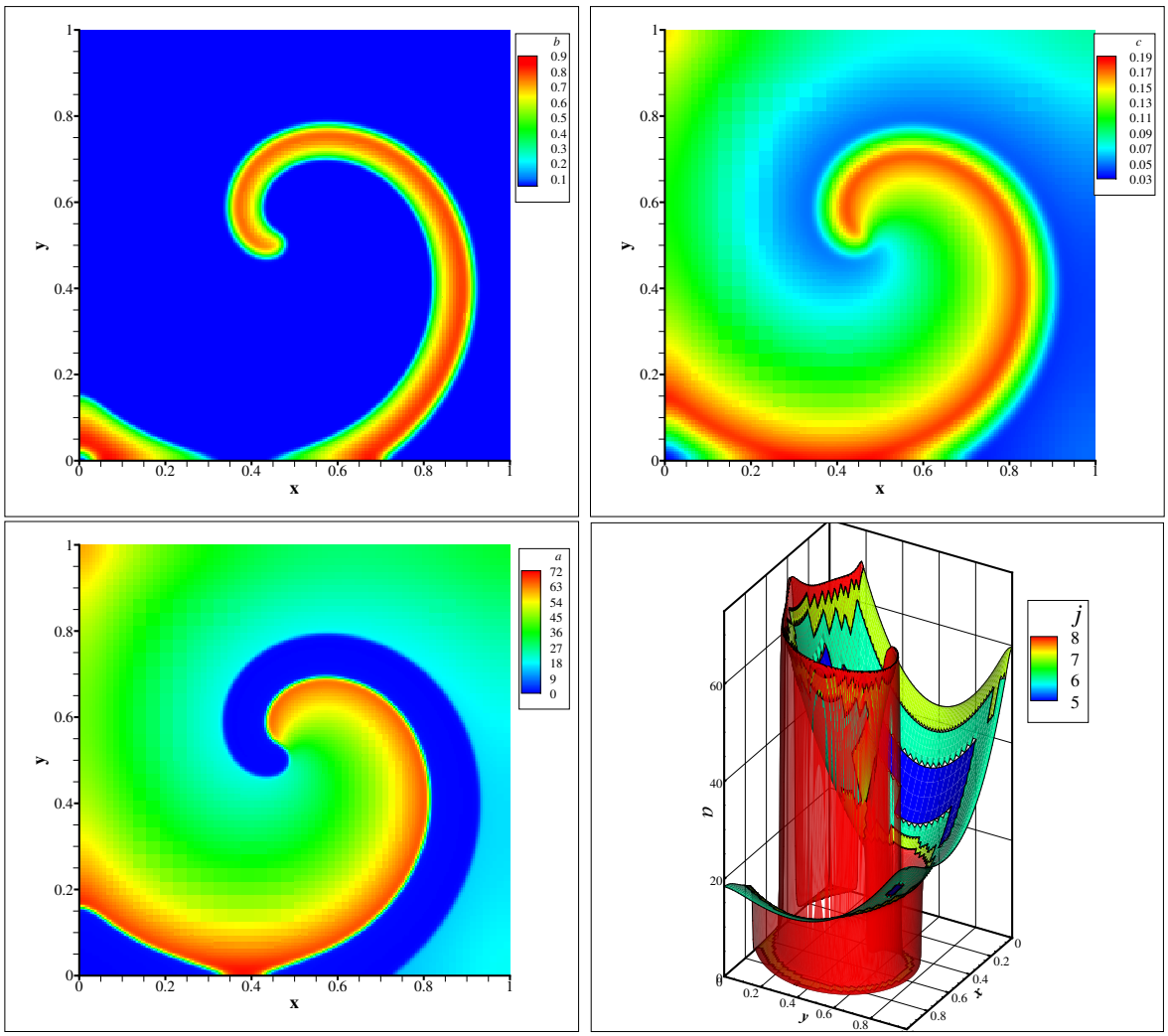

Fig. 4.4: *2D BZ spiral waves. Top: variable b (left) and c (right) at $t = 4$. Bottom: variable a (left) and its representation with four levels of mesh discretization with $\varepsilon = 10^{-2}$ (right). Finest grid:* 256^2.

For the spatial discretization of 256^2, the spectral radius $\rho(\mathbf{L})$ estimated by ROCK4[9] is about $1,400$, so that no more than the minimum number of stages $s = 5$ is required according to (4.12) for the splitting solution $\mathbf{U}^J_{split}$. As a consequence, a CPU time of 1029 s is needed compared with the coupled resolution with ROCK4 that takes 23967 s. The latter considers time steps of about 2.4×10^{-6} with the imposed fine tolerance, and 5 internal stages for a larger spectral radius of $95,000$, that includes both reaction- and diffusion-associated eigenvalues. Even though this coupled resolution should be more accurate than a splitting technique, it will be no longer feasible for larger computational domains and moreover not appropriate for more complex chemical terms. For instance, for the coupled problem (4.29), ROCK4 starts showing stability problems for η_{ROCK4} larger than 10^{-5}.

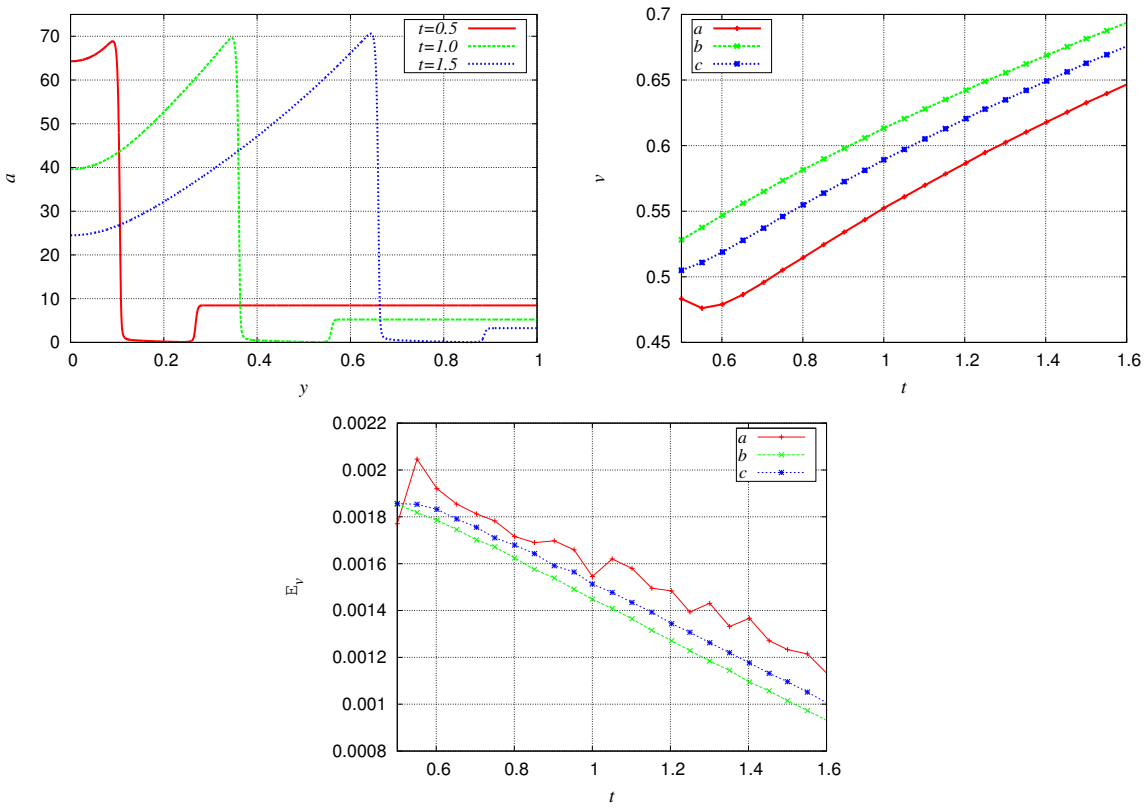

Fig. 4.5: *BZ wave speed for splitting time step $\Delta t = 4/1024$. Top left: time evolution of variable a along y-axis (see Figure 4.4); and right, estimated wavefront speed v. Bottom: speed relative errors E_v for all three variables according to (4.23).*

We consider now the proposed strategy that combines the previous splitting solver with the multiresolution adaptive technique. Figure 4.4 shows the spiral waves and the four different levels of spatial discretization on which they have been simulated with $\varepsilon = 10^{-2}$, for the stiffest variable a. Whenever we consider grid adaptation the bound (4.27) on the splitting time step Δt must be taken into account. We need then to estimate the speed of propagation v of the wavefront. This can be done by computing the propagating speed of each variable along each direction as shown in Figure 4.5 for variable a along the y-axis. For the BZ waves, we have estimated a maximum speed of $v_x = v_y \approx 0.7$, which yields a maximum splitting time step of

[9] Although here we consider a finite volume spatial discretization for the diffusion problem, the estimated value of the spectral radius is coherent with $8 \times D_a/\Delta x^2 = 1310.72$ computed with (2.18).

$\Delta t_{\max} = 1.6 \times 10^{-2}$ for $\Delta x = \Delta y = 1/256$, and $\hat{k} = 3$, considering one enlarging cell in each direction $\bar{k} = 1$ in the refinement criteria. In this particular case, we can also compute the speed relative error E_v following (4.23) between the quasi-exact and a splitting solution with $\Delta t = 4/1024$. These errors remain practically lower than 0.2% as seen in Figure 4.5, which imply an accuracy tolerance of $\eta_v = 2 \times 10^{-3}$ considering the speed resolution.

Data Compression and CPU Time

The proposed MR/splitting strategy represents and computes solutions only on adapted grids, *i.e.*, the leaves of the tree structure, throughout the time domain. Therefore, we define the data compression (DC) as 1 minus the ratio between the number of cells on the adapted grid (AG) and those on the finest uniform grid (FG), expressing the whole as a percentage:

$$\mathrm{DC} = \left(1 - \frac{\mathrm{AG}}{\mathrm{FG}}\right) \times 100. \tag{4.39}$$

Figure 4.6 shows different data compression rates for several threshold values ε. Smaller values of ε imply more refinement, and thus compressions are less important. The whole finest grid is necessary for $\varepsilon < 10^{-5}$. The corresponding CPU times for each case are included in Table (4.1), along with the quasi-exact and the splitting solutions without any grid adaptation.

Table 4.1: *2D BZ spiral waves. CPU time in seconds for quasi-exact, splitting, and MR/splitting solutions with different threshold values ε. Finest grid:* 256^2.

	MR/splitting $\varepsilon =$				splitting	quasi-exact
	10^{-1}	10^{-2}	10^{-3}	10^{-4}		
CPU time (s)	536	886	1233	2402	1029	23967

Table 4.2: *2D BZ spiral waves. CPU time in seconds for the reaction and diffusion time integrations for a splitting and a MR/splitting resolution with $\varepsilon = 10^{-2}$. Finest grid:* 256^2.

	splitting		MR/splitting $\varepsilon = 10^{-2}$	
	CPU time (s)	%	CPU time (s)	%
Reaction	963	65.4	486	44.0
Diffusion	481	32.7	348	31.5
Total	1472	100.0	1104	100.0

A more precise analysis of the CPU time consumption summarized in Table (4.2) shows that:

- An adapted grid allows us to significantly reduce the time cost of the reaction integration as a consequence of the important reduction of the number of points without any chemical activity.
- Although for $\varepsilon = 10^{-2}$ we consider only 25 % of the 256^2 points, an important overhead is introduced in the time integration of the diffusion because the introduction of phantom

cells yields denser matrix representations of the discretized diffusion operator and thus more expensive matrix-vector products.

For these stiff problems, the MR operations represent less than 15 % of the total time, whereas the construction of the diffusion matrix takes over 6 % since it has to be recomputed at each splitting time step, contrary to a uniform grid representation for which this matrix is constant. From a practical point of view, we can see that a more efficient strategy will directly consider the adaptive grid for the flux evaluations as detailed in [Mül03], without the introduction of any phantom. An overhead is introduced in the CPU times in Table (4.2) coming from the code profiling.

Numerical Accuracy of Computations

We consider now the numerical accuracy of the MR/splitting strategy $\mathbf{U}^{MR}_{split}$, with respect to the reference solution $\mathbf{U}^{J}_{qe}$, for the semi-discretized problem (4.29) on a uniform mesh J given by 256^2 points, taking into account that

- The accuracy of the splitting solution $\mathbf{U}^{J}_{split}$, on the same uniform mesh J, is given by an accuracy tolerance η, according to (4.24) through the proper choice of the splitting time step Δt (see Figure 4.3) regardless of the possible loss of order for the Strang $\mathcal{S}_2^{\Delta t}$ scheme.
- The multiresolution decomposition yields a compressed spatial representation whose accuracy to approximate the corresponding uniform mesh representation is related to the threshold value ε through (4.26).

At some fixed time t^*, the overall numerical accuracy of the MR/splitting solution is then set by the previous splitting and multiresolution errors:

$$\left\|\mathbf{U}^{J}_{qe} - \mathbf{U}^{MR}_{split}\right\|_{L^2} \leq \underbrace{\left\|\mathbf{U}^{J}_{qe} - \mathbf{U}^{J}_{split}\right\|_{L^2}}_{\mathcal{O}(\eta)} + \underbrace{\left\|\mathbf{U}^{J}_{split} - \mathbf{U}^{MR}_{split}\right\|_{L^2}}_{\mathcal{O}(\varepsilon)}. \tag{4.40}$$

The space discretization error is thus not considered since the reference solution $\mathbf{U}^{J}_{qe}$ is already discretized in space. The latter error depends on the degree of spatial refinement of the solution and might be accessible by comparing $\mathbf{U}^{J}_{qe}$ with an analytical solution, if the latter is available, or with a highly refined representation. In any case, (4.40) measures the numerical errors related to the time integration procedure and the compressed spatial representations. For sufficiently fine spatial representations, estimate (4.40) represents the global accuracy of the numerical simulation as well.
Figure 4.6 shows the corresponding normalized L^2 errors at $t^* = 4$ for several threshold values ε, and $\eta = 10^{-2}$ for a splitting time step of $\Delta t = 4/1024$. Multiresolution errors are evaluated at the finest grid J after reconstruction from the adapted mesh solution and depend proportionally on the imposed threshold value ε, according to (4.26). For this time multi-scale phenomenon, the accuracy of the MR/splitting strategy should be fixed by the time integration process in order to guarantee an appropriate resolution of the time scale spectrum of the stiff problem. The multiresolution procedure allows us to compress the spatial representation by retaining the desired level of refinement only wherever it is necessary, taking into account the space multi-scale features of the physical problem. In this case, these error estimates show that for $\varepsilon \leq 10^{-2}$, the multiresolution errors become negligible compared with the operator splitting ones, so that the overall accuracy is indeed given by η.

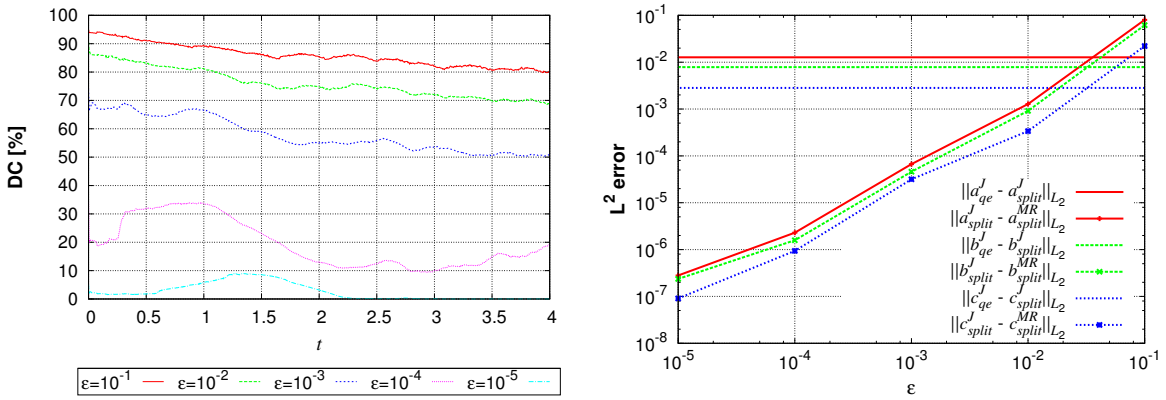

Fig. 4.6: *2D BZ spiral waves. Left: time evolution of the data compression* DC *given by (4.39), in percentage. Right: normalized* L^2 *errors at* $t^* = 4$ *given by the splitting technique on a uniform grid according to (4.24), and by the MR procedure according to (4.26), for several threshold values* ε *and a splitting time step of* $\Delta t = 4/1024$. *Finest grid:* 256^2.

Extension to Larger Computational Domains

We have analyzed so far the numerical behavior of the proposed strategy in terms of the splitting method, the computational costs, and the numerical errors for a computational domain of 256^2 points. This mesh allows us to represent accurately enough the multi-scale phenomenon, and moreover it allows us to conduct several computations with reasonable computational resources. Let us now consider a more challenging configuration with a larger computational domain in order to complete the present illustration. We therefore consider the semi-discretized problem (4.29) discretized this time over 10 nested dyadic grids with $N_J = 2^{2\times 10} = 1048576 = 1024^2$ cells on the finest grid $J = 10$.
In order to take into account the memory requirements of each numerical solution strategy, we estimate the array size of the working space needed by Radau5 and ROCK4:

1. Radau5: $L_1 = 4 \times W_1 \times W_1 + 12 \times W_1 + 20$ (from [HW96]);
2. ROCK4: $L_2 = 8 \times W_2$ (from [Abd02]);

where W_1 and W_2 are, respectively, the number of unknowns solved by Radau5 and ROCK4. In the case of a uniform mesh, the total number of unknowns is $W = 3 \times 1024^2 \approx 3.15 \times 10^6$, and thus the global size L required for each solver is:

1. Quasi-exact with Radau5: $W_1 = W \approx 3.15 \times 10^6$ and $L = L_1 \approx 4 \times 10^{13}$.
2. Splitting: $W_1 = 3$, $W_2 = W \approx 3.15 \times 10^6$ and $L = L_1 + L_2 \approx 2.5 \times 10^7$.
3. MR/splitting with $\varepsilon = 10^{-2}$: $W_1 = 3$, $W_2 = 0.09 \times W \approx 2.9 \times 10^5$, and $L = L_1 + L_2 \approx 2.3 \times 10^6$; with an average data compression of 91%.

Considering a standard platform on which each double precision value is represented by 64 bits, we shall require 2.3 Pb, 1.5 Gb, and 140.4 Mb, respectively, for each solver. For standard computational resources, an implicit resolution with Radau5 is completely out of reach. These expensive memory requirements are strongly reduced with a splitting strategy but further reductions are achieved by adding a multiresolution adaptive procedure.
Figure 4.7 (top) shows the spatial representation of variable a on the finest level corresponding to a 1024^2 spatial discretization of problem (4.29), at an intermediate time $t = 2$, and after

one revolution at final time $t = 4$. The corresponding data compressions DC are, respectively, of 92.3 % and 89.9 %, while the steepest spatial gradients of the front are always solved within the finest region taking into account that the splitting time step $\Delta t = 4/1024$ remains bounded by $\Delta t_{\max} = 4.2 \times 10^{-3}$ for $\Delta x = \Delta y = 1/1024$ according to (4.27). For this case, six levels of grid were used from $j = 5$ to the finest grid $J = 10$.

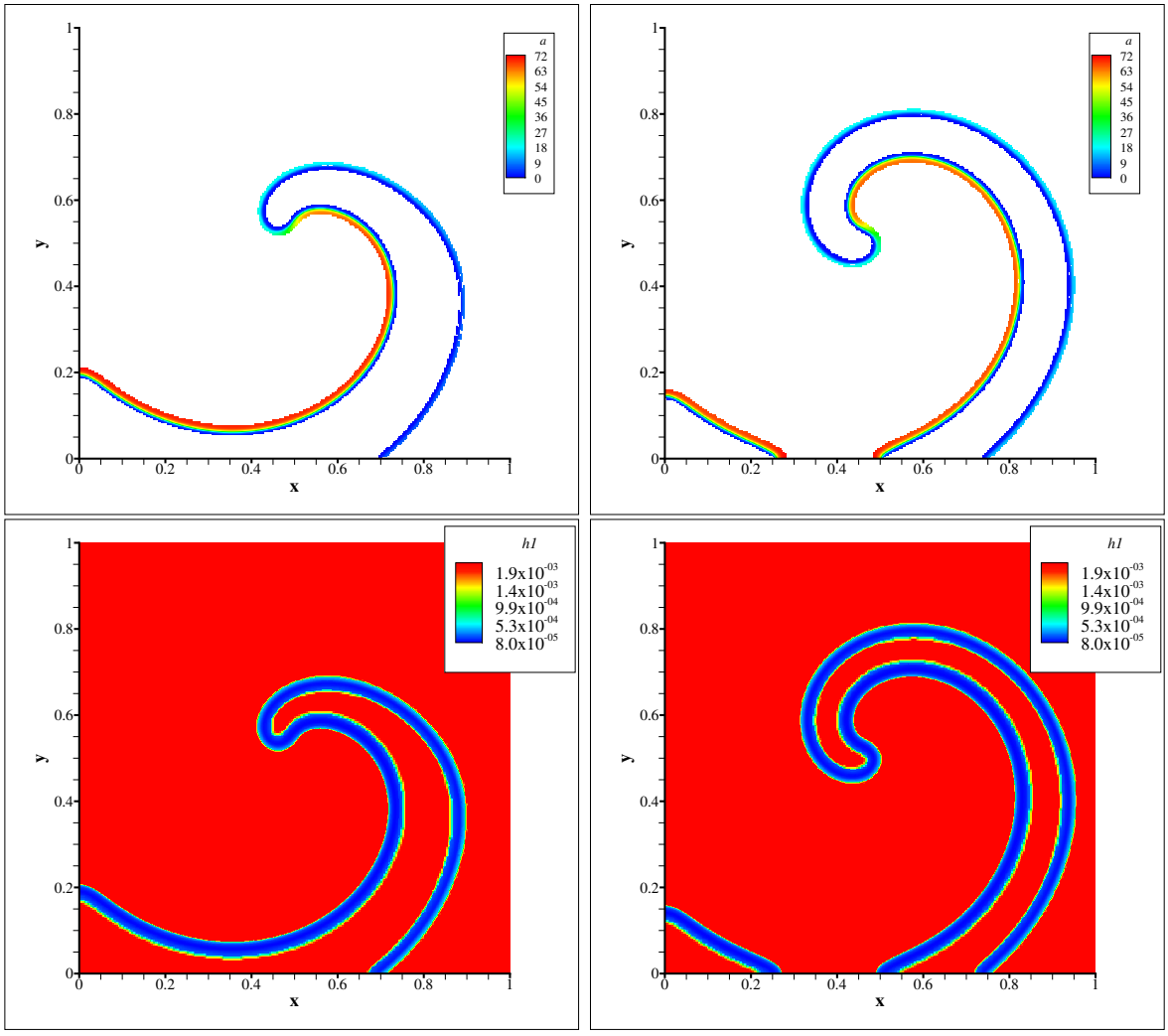

Fig. 4.7: *2D BZ spiral waves. Variable a on the finest grid (top) and local reaction time steps h*1 *(bottom) at* $t = 2$ *(left) and* $t = 4$ *(right) with* $\Delta t = 4/1024$ *and* $\varepsilon = 10^{-2}$. *Finest grid:* 1024^2.

The bottom of Figure 4.7 shows the corresponding mean reaction time steps within $\Delta t/2$ for each point. We have the same distribution as in the previous 256^2 case with reaction time steps going from $\Delta t/2$ to time steps almost 22 times smaller depending of the local chemical activity. On the other hand, the spectral radius $\rho(\mathbf{L})$ estimated by ROCK4[10] is larger because of the finer spatial discretization of the Laplacian operator and it is of the order of $23,000$. If we consider a diffusion time step Δt_D equal to the splitting time step, $s = 16$ stages will be needed according to (4.12). Nevertheless, for a given tolerance of $\eta_{ROCK4} = 10^{-5}$, an initial time step given by $\Delta t_D = \Delta t$ is rejected to finally reach a relatively constant value of $2.5 - 3.5 \times 10^{-4}$

[10] The estimate of the spectral radius (2.18) yields in this case $20,971.52$.

for which 5 stages instead of 16 are enough to guarantee the stability of the method[11]. Finally, from 11 to 15 diffusion time steps Δt_D are computed inside each splitting time step Δt.
From a practical point of view, the general idea is that the finest grid of computation is previously settled, and it is basically limited by the computational resources. The multiresolution error is then indicating the numerical approximation of the compressed spatial representation with respect to the semi-discretized problem regardless of its spatial discretization as shown in Figure 4.8 (left) for a finer spatial discretization of 1024^2 instead of 256^2 in Figure 4.6. The quasi-exact solution was computed with ROCK4 and $\eta_{ROCK4} = 10^{-10}$, and took over 65072 s compared with 13943 s and 9529 s, respectively, for the splitting solution and the MR/splitting solution with $\varepsilon = 10^{-2}$. Figure 4.8 (right) shows the dependence of the data compression on the accuracy order of the prediction operator, as discussed in Chapter 3. For higher order polynomial interpolations that yield more accurate and thus more compressed multiresolution representations, larger stencils are also needed so that the resulting data compressions take into account both opposite features. In a general case, this is a problem dependent feature that can be turned into a useful parameter to improve the performance of the multiresolution technique.

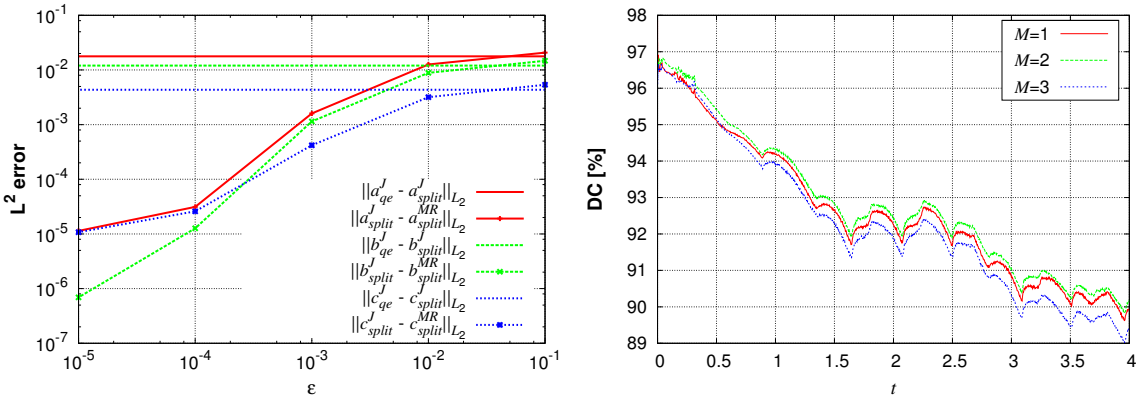

Fig. 4.8: *2D BZ spiral waves. Right: normalized L^2 errors at $t^* = 4$ given by the splitting technique on a uniform grid according to (4.24), and the MR procedure according to (4.26) for several threshold values ε. Left: time evolution of data compressions* DC *for prediction operators given by polynomial interpolations of different orders $N = 2M + 1$. The splitting time step is of $\Delta t = 4/1024$ with a finest grid of 1024^2.*

4.6.4 3D BZ Equation

In order to extend previous applications to 3D simulations, we now consider problem (4.29) in a 3D configuration with the same parameters considered for the 2D case (4.32) and (4.33), for a time domain of $[0, 2]$, and in a space region of $[0, 1]^3$. The initialization is made in the same way, but with the coordinate angle θ given this time by

$$\theta = \arctan\left(\frac{(x-0.5)\sin(\pi z) + (y-0.5)\cos(\pi z)}{(x-0.5)\cos(\pi z) - (y-0.5)\sin(\pi z)}\right). \tag{4.41}$$

We retrieve the previous 2D case with z equal to zero.

[11] We recall that ROCK4 needs to save only 8 arrays of the size of the number of unknowns regardless of the number of stages. One of these arrays contains the approximate solution used to estimate the local error in order to adapt the time step within the prescribed tolerance.

First, we take into account 8 nested dyadic grids with $N_J = 2^{3\times 8} = 16777216 = 256^3$ cells on the finest grid $J = 8$. Then, with a threshold value of $\varepsilon = 10^{-2}$, and a splitting time step $\Delta t = 4/1024$, the strategy features data compressions of 92.61 % for the initial condition, 85.64 % at $t = 1$ when the scroll waves are fully developed, and 81.42 % at final time $t = 2$. Figure 4.9 shows the scroll waves for variable a at two different times and the adapted grid at $t = 2$. The finest regions correspond to the neighborhood of the wavefront. The CPU computation time was of about 41.94 hours with one processor.

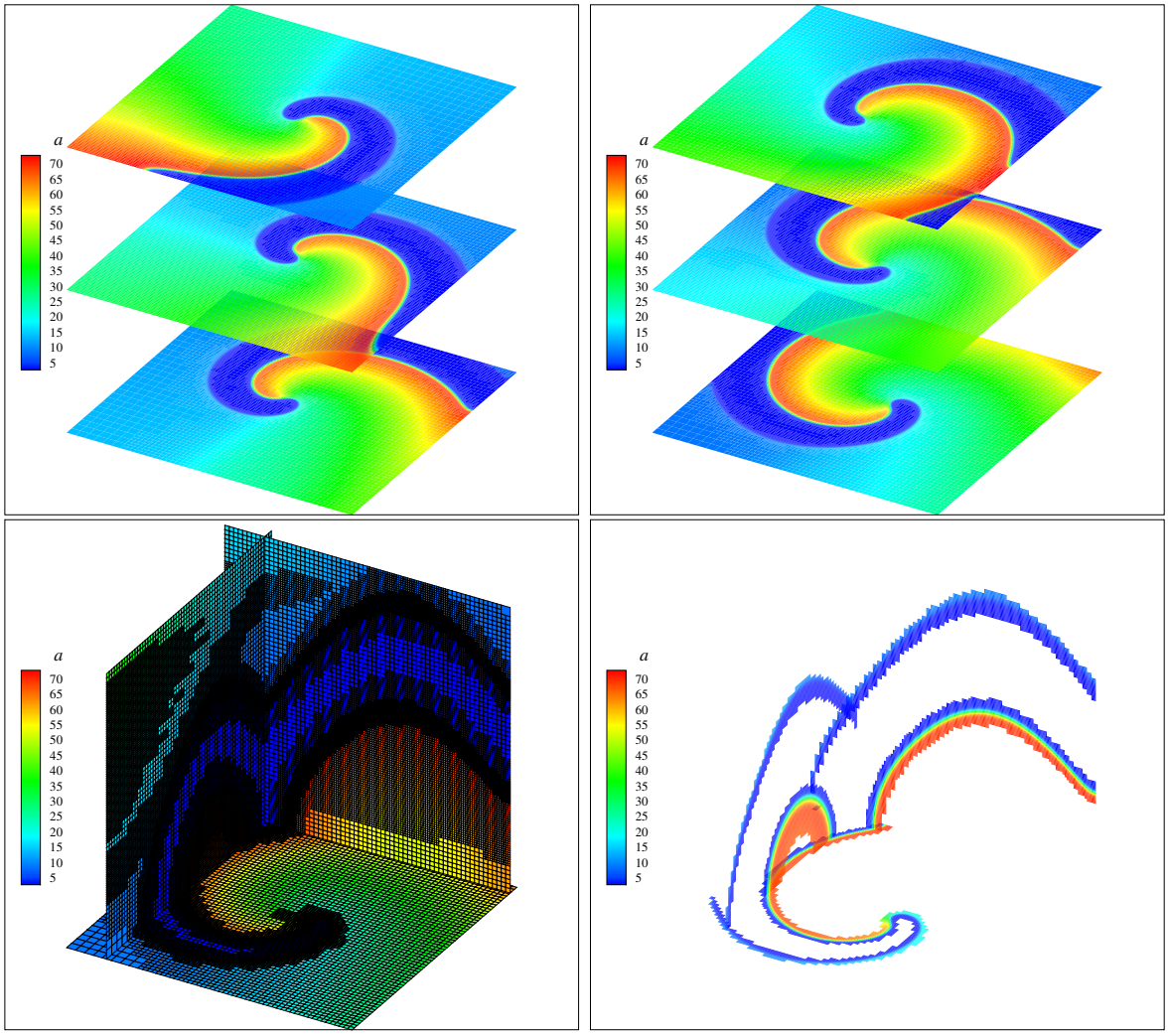

Fig. 4.9: *3D BZ scroll wave. Top: evolution of variable a at $t = 1$ (left) and $t = 2$ (right). Bottom: Adapted grid (left) and finest grid (right) at $t = 2$ for $\varepsilon = 10^{-2}$. Finest grid: 256^3.*

To explore the feasibility and potential advantages of the method, let us consider 9 nested dyadic grids with $N_J = 2^{3\times 9} = 134217728 = 512^3$ cells on the finest grid $J = 9$. The initialization must take place on a intermediate grid, $j^0 = 8$ in this example. For this configuration, a two times larger splitting time step of $\Delta t = 4/512 \approx 7.8 \times 10^{-3}$, and a threshold value of $\varepsilon = 10^{-1}$ were chosen to have splitting and multiresolution errors potentially of the same order. Smaller threshold values yield larger simulation domains which are not longer feasible with the considered computing resource and the current state of development of the code. Figure 4.10 shows the adapted grid at $t = 2$ and the corresponding finest regions. Compared with the 256^3

case, finer regions are added at the steepest spatial gradients of the front. Additionally, in order to globally guarantee (4.26), more refinement is needed at the lower levels according to (4.15) for a given threshold ε. The multiresolution representation error (4.26) is always measured with respect to the corresponding uniform semi-discretized problem at the finest level J. The latter is mainly limited by the computational resources and the desired level of accuracy of the spatial resolution. Data compressions are now of 95.79 % for the initial condition, 91.56 % at $t = 1$, and 91.20 % for final time $t = 2$, with a CPU time of 159.4 hours.

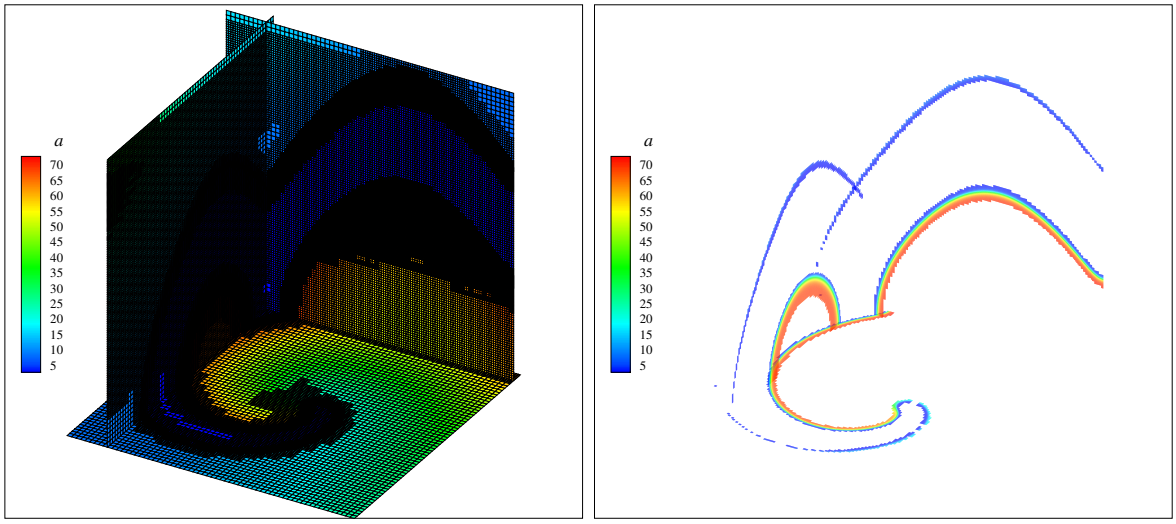

Fig. 4.10: *3D BZ scroll wave. Adapted grid (left) and finest grid (right) at* $t = 2$ *for* $\varepsilon = 10^{-1}$. *Finest grid:* 512^3.

Performing the same comparison concerning 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 the global size of L required by each solver is:

1. Quasi-exact with Radau5: $W_1 = W \approx 4.03 \times 10^8$ and $L = L_1 \approx 6.5 \times 10^{17}$.
2. Splitting: $W_1 = 3$, $W_2 = W \approx 4.03 \times 10^8$ and $L = L_1 + L_2 \approx 3.2 \times 10^9$.
3. MR/Splitting with $\varepsilon = 10^{-1}$: $W_1 = 3$, $W_2 = 0.13 \times W \approx 5.3 \times 10^7$, and $L = L_1 + L_2 \approx 4.2 \times 10^8$; with a data compression of 87%.

Therefore, we shall require at least 36.1 Eb, 190.7 Gb, and 25.0 Gb of memory capacity, respectively, for each solver.

4.7 Concluding Remarks

To finish this chapter, let us synthesize the main points discussed in details throughout the different sections. First of all, we have introduced a numerical strategy for the numerical solution of stiff reaction-diffusion equations, based mainly on the following three blocks:

- A second order Strang time operator splitting with a splitting time step defined within a prescribed accuracy, according to the global physics of the problem and the decoupling capabilities of the governing equations.

- High order one-step time integration methods for the independent numerical solution of the split subproblems with adaptive time stepping tools within a prescribed accuracy.
- A space adaptive multiresolution technique that yields compressed spatial representation within a prescribed accuracy with respect to a uniform grid space resolution.

In order to monitor the global accuracy of the numerical simulation we have decoupled the numerical approximation errors in three constituting pieces with the following criteria:

- The space discretization error associated with the semi-discretized problem. This error is difficult to evaluate in practice since in most of the cases an analytical solution is not available, and a highly refined spatial representation of the problem is not feasible.
- The splitting error associated with the decoupled numerical solution of each subproblem coming from the semi-discretized problem. This error is mainly set by the choice of the splitting time step.
- The multiresolution error associated with the spatially compressed representation of the semi-discretized problem. This error is ruled by the choice of the threshold value during the multiresolution decomposition.

Notice that in this context both the splitting and multiresolution errors become independent of the space discretization of the problem. The approximation errors associated with the numerical methods, *i.e.,* the time integration and the adaptive mesh refinement techniques, can be hence tracked, whereas for a sufficiently fine space discretization limited by the computational resources, a complete evaluation of the global accuracy of the numerical simulation is also possible. In this work we have considered second order spatial discretization for the diffusion problem although higher order schemes might be implemented with the same time integration solver. The numerical strategy is thus quite general and the same constructing and implementing criteria can be applied to consider other phenomena in the time dependent PDEs. An important gain of computational performance is shown to be achieved with the resulting method mainly because of the adaptive grid features and the implementation of a dedicated time integration strategy. Further enhancements are possible if the physical configuration of the problem allows us to perform an effective decoupling of the time scale spectrum of the governing equations.
In this chapter, we have focused our attention on the numerical simulation of propagating waves, and therefore the efficiency of the previous strategy will be ensured for more general problems as long as:

- A constant splitting time step is sufficient to describe the global physics of the simulated phenomenon.
- The simple refinement and coupling criteria between the time evolution of the waves and the adaptive grid refinement ensure a proper dynamic representation of the steepest spatial gradients in the solution.
- The reactive activity is mainly concentrated in the wavefronts which are spatially refined to avoid an important accuracy reduction in the evaluation of the source terms and thus perturbations in the global phenomenon.

The first constraint reduces considerably the applicability of the method for a large number of application featuring more general and probably highly unsteady phenomena. The natural solution to this limitation is given by the construction of an efficient adaptive splitting

scheme for general stiff problems. This time adaptive procedure must be performed within a prescribed accuracy in order to ensure the previous working framework in terms of accuracy of the numerical simulations. These key issues remained an open problem for many years, and in the next chapter of this work we propose a theoretical and numerical solution to overcome these difficulties.
Concerning the second constraint, a more sophisticated refinement criteria might be required, as the ones introduced by Cohen *et al.* in [CKMP03], that allow us to add more than one refined level before each new time step in order to properly describe suddenly high variations in the new numerical solution. Nevertheless, we shall see in the forthcoming chapters that an appropriate frequency of remeshing set by the dynamic time stepping technique developed in this work allows us also to rapidly capture the spatial dynamics of highly unsteady problems, keeping the same and much simpler previous refinement and coupling criteria. A combination of more dedicated refinement techniques and the adaptive splitting time stepping might be however considered to further improve the numerical strategy in some particular cases. The latter option has not been sufficiently analyzed for the moment and will not be described in the present work. The same follows for the last constraint which might require special techniques for the evaluation of the source terms, as the ones developed, for instance, by Hovhannisyan & Müller in [HM10] for an efficient numerical simulation of some particular applications.

Chapter 5

Adaptive Splitting Scheme for Multi-Scale Evolutionary PDEs

This chapter introduces a time adaptive operator splitting scheme for the numerical integration of stiff PDEs. Time adaptive techniques were partially mentioned in the General Introduction and in Chapter 2, where dynamical time stepping was introduced in the context of stiff ODEs systems. In this part, we will first present a brief overview of some time adaptive schemes for PDEs in the literature. Then, we will introduce an adaptive splitting scheme that complements the numerical strategy presented in Chapter 4. An article on the adaptive splitting strategy was published in Confluentes Mathematici [DDD+11], in a special issue dedicated to the memory of Michelle Schatzman.

5.1 Time Adaptive Schemes for PDEs

Time adaptation for the solution of stiff PDEs is a critical aspect for numerical simulations mainly because it allows us to track the numerical accuracy of the computations as usually done for ODEs systems, and because highly unsteady PDEs can be efficiently simulated. Either if the solution is globally or locally advanced in time, a dynamic adaptation to the numerical and physical scales of the problem is mandatory to overcome important numerical restrictions. It what follows, some time stepping techniques for PDEs will be briefly discussed.

- *Time stepping for IMEX methods.* Standard time stepping techniques for IMEX schemes consider embedded lower order methods or other kind of procedure to numerically estimate the local error (see [HNW87, HW96]), as previously seen for some stiff ODE solvers in Chapter 2[1]. In general, the order conditions of IMEX methods take simultaneously into account the explicit and implicit schemes. Therefore, in practice, it suffices to consider, for instance, an embedded method with supplementary stages that satisfies lower order conditions (see [KC03] and references therein). An estimate of the local error can be then dynamically computed and consequently the time integration steps, as performed in classical time integration of ODEs. This is also the case, for instance, for the time integration scheme employed in the partitioning schemes presented in [HW06] for reaction-diffusion systems, for which space grid adaption was also proposed. In the IRKC code [SSV06] that implements an IMEX scheme based on RKC methods proposed by

[1] Let us remark that the same procedures are valid for any coupled solution scheme like a fully implicit one. For instance, the leading term of the local error Taylor expansion is numerically evaluated in [DS10] to ensure a prescribed accuracy of computations for a first order implicit Euler scheme used to solve the coupled governing equations for combustion problems with detailed chemistry.

[VS04] for reaction-diffusion systems, and then extended to convection-diffusion-reaction problems [VSH04], the leading term in the Taylor series of the local error is numerically evaluated taking into account that the method is of order 1 (hopefully 2 in practice)[2]. In this way, the time step is chosen in order to guarantee computations within a prescribed tolerance and within the stability domain of the explicit scheme in which, by construction, the implicit scheme for the stiff terms remains stable [VS04].

- *Time stepping via adjoint error representation.* Other strategies take into account adjoint operators (see, *e.g.,* [Sül99, Ste08, SN08, SMN10]), which consider the dual problem associated with the time dependent PDEs. The main advantage of this approach relies on its rigorous mathematical background which allows us to better control the error of computations. The theoretical framework is mainly settled by the linear or nonlinear operators present in the problem, and their corresponding adjoint operators. The latter ones lead to an adjoint representation of the original problem, the dual problem, for which one can deduce theoretical error estimates of the integration scheme. In particular, adjoint operators have been also used to describe and analyze splitting errors for reaction-diffusion systems [EGR+08, ECG+08, Gin10]. The main difficulty of these methods is that more sophisticated schemes are required to handle the evolution of the adjoint time operators, which are not always well-posed. A mathematical study is therefore required and the procedures are often difficult to generalize. The same follows for multi-dimensional configurations. Nevertheless, for particular problems as Burgers' equation [SN08] or Euler equations [SMN10], such a scheme might be efficiently implemented. In general, an important overhead is introduced by the computation of the adjoint error representation since one needs to compute the forward and dual problems plus supplementary operations at each time step, which usually cost several times one single forward solution. These costs are nevertheless reduced in [SMN10] by computing the adjoint errors on a coarser mesh issued from a multiresolution analysis. In this way, a time-space adaptive was introduced for 2D Euler equations. Another alternative combines implicit and explicit time integrations to reduce computational requirements.

- *Multirate time stepping strategies.* The main idea of these schemes is to use a self-adjusting time step for a particular component of an ODE system, instead of using a single global time step for the whole system. In this way, potentially stiff components with different time scales are computed within a prescribed tolerance based on their own local temporal variations. These are thus local time stepping techniques with error control, and can be adapted to general time integration strategies. In particular, the ODE systems might come from semi-discretized PDEs. Automatic multirate schemes were first introduced by [GW84] for linear multi-step methods. Further developments considered, for instance, partitioned Runge-Kutta methods for fast and slow well-partitioned systems [GKR01]. In general, fully implicit [Log02, Log03] or linearly implicit Runge-Kutta methods of Rosenbrock-type [BG02, SHV07] have been developed with local error estimates based on lower order embedded schemes, suitable for stiff problems. Some crucial aspects are related to the internal and final synchronization steps for all components of the system, in which interpolations are usually implemented to define missing data at intermediate time steps, and the implementation complexity of fully implicit methods. For a recent review on these methods, see [SM10] and references therein. On the other hand, a combination of IMEX and multirate techniques have been recently introduced

[2] This procedure is different to the one implemented in the previous RKC code [SSV97] for which there is an estimate of the leading term of the second order RKC scheme, based on theoretical findings [VHS90].

in [CD11].

In the context of adaptive meshing schemes, many *local time stepping* techniques have been developed to locally adapt time steps according to the local mesh size. This idea dates back to the original works of Berger & Oliger on AMR techniques [BO84], and were then formalized by Berger & Collela in [BC89]. The general application framework was given by hyperbolic conservation laws for which the time steps of explicit schemes would be heavily constrained by the finest space scale of the computational domain (see, *e.g.*, [BDA+06] and references therein). The time steps are usually chosen based on local stability criteria and therefore allow us to greatly improve numerical performances. Appropriate synchronization steps among cells evolved with different time steps must be defined, usually based on interpolation approximations, in order to limit the corresponding overhead of computations. A study on the optimal choice of the local time steps and the synchronization stages can be found in [CNPT09]. All these ideas were extended to multiresolution techniques for which Müller & Stiriba introduced in [MS07] a locally varying time stepping for explicit and implicit schemes. The latter motivated many other extensions for hyperbolic (see, *e.g.*, [CPPT06, CNPT08, CNPT10]) and parabolic problems [DGRS08]. The latter authors introduced also embedded explicit Runge-Kutta schemes for error control in [DRS09], which were later coupled with local time stepping in [DGRS09]. Another multiresolution technique with local time stepping for parabolic problems was developed in [BBRBS09], and applied for excitable media problems in [BRBS10], where a special first order splitting technique was also implemented to handle stiff reaction terms with a semi-implicit scheme to improve computing performances.

In the previous chapter, we have introduced a new time splitting technique for stiff PDEs. This approach allows us to independently treat different problems and the associated time scales according to the decoupling capabilities of the physical problem. The splitting time step is chosen within a prescribed accuracy in order to guarantee error control of the time integration. In this way, the time evolution of the global solution is not limited by stability restrictions as in standard configurations, and an important numerical performance can be achieved whenever a broad decoupling of the numerical and physical time scale spectrum is possible. In particular, this splitting approach considers a dedicated implicit solver for stiff reaction problems that can be seen as a local time stepping strategy defined this time by accuracy criteria as in a multirate configuration, but without any data synchronization and interpolation errors; whereas the explicit solvers might perform several time steps within the splitting time step in order to guarantee numerical stability, in opposition to standard coupled schemes, like fully implicit or consistent IMEX methods[3]. Additionally, the numerical complexity of the splitting environment is rather minimum considering the successive and independent solution of the various subproblems, and that numerical schemes specially conceived for each one of them are implemented.
We have also seen in the previous chapter that in the context of propagating waves a constant splitting time step is more than reasonable to precisely describe the global coupling of the split phenomena. Fixed splitting time step schemes constitute the most standard splitting implementation up to our days and have been largely used in the literature (see, *e.g.*, [KNW99, DB00, OB01, SLGS03, NK05, SPN06a, RP08]). However, such a fixed time stepping strategy would surely lead to major difficulties and limitations for problems describing highly unsteady

[3]By consistent IMEX method, we mean that the numerical solution of the problem ensures the complete coupling of the different terms. This excludes, for instance, the possibility of time stepping of the implicit solver within the explicit and global time step, in which case some sort of splitting errors would be naturally introduced.

models with very different dynamics such as flame ignition and propagation, or repetitively pulsed plasmas discharges, all the more in the framework of large scale simulations. It seems thus essential to be able to dynamically adapt splitting time steps for the simulation of such multi-scale problems with strongly evolving dynamics.
In order to guarantee a precise description of the coupled multi-scale phenomenon, this adaptive splitting scheme must rely on a local error estimate which can be obtained, for instance, by considering a lower order embedded method, as seen for ODEs in Chapter 2. The time steps can therefore dynamically adapt according to a given tolerance. Nevertheless, it is well known that for stiff problems and larger accuracy tolerances, the order of the methods can degenerate, yielding non reliable error estimates and possibly, much larger global errors than expected by the given tolerance. Such a scenario will be all the more valid in the framework of PDE solutions, where fine grid and large gradients coupled with stiff source terms lead to especially stiff problems. In particular, the numerical strategy of Chapter 4 was built in such a way that the main source of error is the splitting error. Therefore, it is essential to construct a reliable splitting error estimate to guarantee an effective error control within the so claimed accuracy tolerance.

In this context, to the best of our knowledge the first and only previous attempt to dynamically adapt splitting time steps was considered by Gerisch & Verwer in [GV02] by means of a standard Richardson extrapolation[4] (see [HNW87]), for the numerical solution of PDEs issued from chemotaxis models. Nevertheless, such a general procedure will not be suitable for stiff PDEs mainly because it neglects order reductions, and because each time step has to be integrated twice. Therefore, other alternatives need to be sought. A simple and general adaptive strategy for non stiff problems is, for instance, given by the *Embedded Split-Step Formulae* presented by Koch & Thalhammer in [KT11]. This approach is based on lower order embedded methods, which combine Lie and Strang schemes, and in general allow us to build higher order techniques with negative or complex coefficients. Alternatively, an adaptive splitting scheme suitable for stiff configurations was recently introduced in [DDD+11], and will be detailed in the following.
This strategy aims also at estimating the local splitting error with two different splitting schemes. The first one is the standard second order Strang technique, whereas the second one considers a shifted Strang formula built with a δ-shift in time of the classical Strang formula. This latter method is thus of first order, and is embedded because the first substep is common to both methods to reduce computational costs. It inherits from the Strang scheme its stability properties and the same numerical behavior in the context of stiff problems, as studied in [DM04, DDLM07]. A numerical analysis of the method will be detailed in what follows to settle a solid mathematical background. Furthermore, a complementary numerical procedure based on theoretical estimates will be introduced to overcome classical restrictions of adaptive time stepping schemes whenever asymptotic estimates fail to predict the true dynamics of the problem. The main goal is to conceive and implement a fully adaptive time stepping strategy that guarantees an effective control of the errors of integration for a large range of time steps; a key issue for problems for which splitting time steps can go beyond the fastest physical scales of the problem. In this way, compared with a standard procedure for which accuracy is guaranteed by considering time steps of the order of the fastest scale, this adaptive splitting technique allows us to extend the strategy of Chapter 4 to more complex and highly unsteady problems modeling various physical scenarios, independent of the fastest physical or numerical

[4] An estimate of the local error is computed with the difference between two numerical solutions at each time step, in which one of the them (in general the retained solution) is obtained after two half steps. A higher order solution can be also extrapolated (this is not used in [GV02] where the second order Strang is considered).

time scale.

5.2 Time Adaptive Splitting Method

Throughout this part a scalar 1D case for the linear diagonal diffusion system (1.53) is considered in order to simplify the presentation, taking into account that extensions to higher dimensions of x or u are straightforward:

$$\left.\begin{array}{ll} \partial_t u - \partial_x^2 u = f(u), & x \in \mathbb{R},\ t > 0, \\ u(0,x) = u_0(x), & x \in \mathbb{R}, \end{array}\right\} \tag{5.1}$$

where f and u_0 are smooth functions. As usual, we denote by $T^t u_0$ the solution of (5.1), by $X^t u_0$ the solution of the diffusion equation:

$$\partial_t u_D - \partial_x^2 u_D = 0, \qquad x \in \mathbb{R},\ t > 0, \tag{5.2}$$

with initial data $u_D(0,\cdot) = u_0(\cdot)$ after some time t; and by $Y^t u_0$, the solution of the reaction part where the spatial coordinate x can be considered as a parameter:

$$\partial_t u_R = f(u_R), \qquad x \in \mathbb{R},\ t > 0, \tag{5.3}$$

with $u_R(0,\cdot) = u_0(\cdot)$. The Lie and Strang approximation formulae of the solution of system (5.1) are then defined by

$$\mathcal{L}_1^t u_0 = X^t Y^t u_0, \quad \mathcal{L}_2^t u_0 = Y^t X^t u_0, \tag{5.4}$$

$$\mathcal{S}_1^t u_0 = X^{t/2} Y^t X^{t/2} u_0, \quad \mathcal{S}_2^t u_0 = Y^{t/2} X^t Y^{t/2} u_0. \tag{5.5}$$

An adaptive time stepping strategy is based on a local error estimate which can be obtained by using two schemes of different order, for instance, $\mathcal{S}_1^t$ or $\mathcal{S}_2^t$, locally of order 3, and $\mathcal{L}_1^t$ or $\mathcal{L}_2^t$, locally of order 2. In this case, the Embedded Split-Step Formulae given in [KT11] consider $\mathcal{S}_1^t$ and $\mathcal{L}_2^t$, or $\mathcal{S}_2^t$ and $\mathcal{L}_1^t$, noticing that[5]

$$\mathcal{L}_1^t u_0 = X^t Y^{t/2} Y^{t/2} u_0, \tag{5.6}$$

where $Y^{t/2} u_0$ is also used to compute $\mathcal{S}_2^t u_0$. Nevertheless, we have seen in Chapter 1 that in the context of multi-scale phenomena, order reductions may appear due to short-life transients associated with the fastest variables, when one considers splitting time steps larger than the fastest scales. Furthermore, it has been proved in [DM04] that better performances are expected while ending the splitting scheme by the part involving the fastest time scales of the phenomenon. In particular, in the case of linear diagonal diffusion problems, no order reduction is expected for the $\mathcal{L}_2^t$ and $\mathcal{S}_2^t$ schemes when fast scales are present in the reactive term. Therefore, the embedding procedure must be carefully conceived taking into consideration these theoretical studies.
We introduce a shifted Strang formula

$$\mathcal{S}_{2,\delta}^t u_0 = Y^{(1/2-\delta)t} X^t Y^{(1/2+\delta)t} u_0, \tag{5.7}$$

locally of order 2, due to the lack of symmetry, for δ in $[-1/2, 0) \cup (0, 1/2]$, which likely features the same numerical behavior as $\mathcal{S}_2^t$. In this way, a local error estimate is computed based on

[5] In [KT11] the central step is actually written as the common one for both Lie and Strang schemes, we believe that this is just a printing error.

two solutions for which orders are guaranteed, and a potential loss of order is simultaneous, according to

$$\begin{pmatrix} \mathcal{S}_2^{\Delta t} u_0 \\ \mathcal{S}_{2,\delta}^{\Delta t} u_0 \end{pmatrix} = \begin{pmatrix} Y^{\Delta t/2} X^{\Delta t} Y^{\Delta t/2} u_0 \\ Y^{(1/2-\delta)\Delta t} X^{\Delta t} Y^{(1/2+\delta)\Delta t} u_0 \end{pmatrix}, \tag{5.8}$$

for some splitting time step $\Delta t > 0$. Embedding is accomplished as long as δ is different from $-1/2$,, *i.e.*, $\mathcal{S}_{2,\delta}^{\Delta t} u_0$ different from $\mathcal{L}_2^t u_0$. Contrarily, if δ is equal to $1/2$, $\mathcal{S}_{2,\delta}^{\Delta t} u_0$ is defined as $\mathcal{L}_1^t u_0$, which it is not suitable for stiff configurations as previously discussed [DM04]. Therefore, δ should be contained in $(-1/2, 0) \cup (0, 1/2)$. Shifted $\mathcal{S}_{1,\delta}^{\Delta t} u_0$ is defined in a similar way and depending on the multi-scale character of the problem, it might be the appropriate choice along with $\mathcal{S}_1^{\Delta t} u_0$.
Considering exactly the same time stepping procedure detailed in Chapter 2 for stiff ODEs, we have that

$$\begin{aligned} \mathcal{S}_2^{\Delta t} u_0 - \mathcal{S}_{2,\delta}^{\Delta t} u_0 &= \mathcal{S}_2^{\Delta t} u_0 - T^{\Delta t} u_0 + T^{\Delta t} u_0 - \mathcal{S}_{2,\delta}^{\Delta t} u_0 \\ &= \mathcal{O}(\Delta t^3) + \mathcal{O}(\Delta t^2) \approx \mathcal{O}(\Delta t^2). \end{aligned} \tag{5.9}$$

Therefore, for a given accuracy tolerance η:

$$\left\| \mathcal{S}_2^{\Delta t} u_0 - \mathcal{S}_{2,\delta}^{\Delta t} u_0 \right\| < \eta, \tag{5.10}$$

must be verified in order to accept the current computation with Δt. The new time step is then calculated by

$$\Delta t^{\text{new}} = \upsilon \, \Delta t \sqrt{\frac{\eta}{\left\| \mathcal{S}_2^{\Delta t} u_0 - \mathcal{S}_{2,\delta}^{\Delta t} u_0 \right\|}}, \tag{5.11}$$

with security factor $0 < \upsilon \leq 1$, close to 1. More sophisticated formulae than (5.11) like the step size strategy with memory (2.46), can be also considered.
The error control of these adaptive methods is fully guaranteed as long as the orders of both, the main and the embedded integration methods, remains valid and (5.9) is satisfied. This is the case for small enough time steps for which asymptotic theoretical estimates hold, but remains an open problem for larger time steps for which the validity of the former orders is assumed in any standard time-stepping scheme. This is a key point in this work because we propose not only a new splitting strategy with adaptive time steps as described in this section, but we also aim at applications for which splitting time steps may go beyond the fastest scales associated with each subproblem in order to obtain important computational savings. A technique that consistently guarantees error control for all possible splitting scales must be then pursued, but first a detailed numerical analysis of the method must be performed.

5.3 Numerical Analysis of the Adaptive Scheme

In this part, we conduct the numerical analysis of the method previously detailed. It is mainly based on the theoretical study of the approximation of the solution T^t of (5.1), by the shifted Strang formula (5.7): $\mathcal{S}_{2,\delta}^t$. General estimates for the approximation of T^t by the standard $\mathcal{S}_2^t$ can be naturally obtained by taking $\delta = 0$. To simplify the notations in the following, we will denote $\mathcal{S}_2^t$ by $\mathcal{S}^t$, and $\mathcal{S}_{2,\delta}^t$ by $\mathcal{S}_\delta^t$.

5.3.1 Linear Framework

Assume that A and B are linear bounded operators and define according to (5.7),

$$\mathcal{S}_\delta^t u_0 = \mathrm{e}^{(1/2-\delta)tA}\mathrm{e}^{tB}\mathrm{e}^{(1/2+\delta)tA}u_0, \tag{5.12}$$

as an approximation of $\mathrm{e}^{t(A+B)}u_0$. The following theorem gives the expansion in powers of t of the difference between $\mathrm{e}^{t(A+B)}u_0$ and $\mathcal{S}_\delta^t$. We recall the definition of the brackets between A and B given in (1.21): $[A,B] = AB - BA$.

Theorem 5.1. *Assume that A and B are linear bounded operators, for t and δ small enough, the following asymptotic holds*

$$\begin{aligned}\mathrm{e}^{t(A+B)}u_0 - \mathcal{S}_\delta^t u_0 &= -\delta t^2[A,B]u_0 + \frac{t^3}{24}\big([A,[A,B]] + 2[B,[A,B]]\big)u_0 \\ &\quad +\mathcal{O}(\delta t^3) + \mathcal{O}(t^4).\end{aligned} \tag{5.13}$$

Proof. The proof is straightforward by using the Taylor formula with integral remainder for a linear bounded operator A:

$$\mathrm{e}^{tA} = \mathrm{Id} + tA + \frac{t^2A^2}{2} + \frac{t^3A^3}{6} + \int_0^t \frac{(t-s)^3}{6}A^4\mathrm{e}^{sA}\,\mathrm{d}s, \tag{5.14}$$

and the Baker-Campbell-Hausdorff formula (1.23). □

5.3.2 Nonlinear Framework

We extend now the previous theorem to our nonlinear framework given by system (5.1), by using the Lie formalism introduced in Chapter 1. Furthermore, considering the spaces $\mathcal{C}^\infty(\mathbb{R})$ of functions of class $\mathcal{C}^\infty$ on $\mathbb{R}$, and $\mathcal{C}_b^\infty(\mathbb{R})$ of functions of class $\mathcal{C}^\infty$ on $\mathbb{R}$ and bounded over $\mathbb{R}$, we introduce the *Schwartz space* $\mathbb{S}(\mathbb{R})$ defined by

$$\mathbb{S}(\mathbb{R}) = \{g \in \mathcal{C}^\infty(\mathbb{R}) \text{ s.t. } \sup_{v\in\mathbb{R}} |v^{\alpha_1}\partial_v^{\alpha_2} g(v)| < \infty \quad \forall\, \alpha_1,\alpha_2 \in \mathbb{Z}\}; \tag{5.15}$$

and we define the space $\mathbb{S}_1(\mathbb{R})$, made out of functions u belonging to $\mathcal{C}_b^\infty(\mathbb{R})$ such that u' belongs to $\mathbb{S}(\mathbb{R})$:

$$\mathbb{S}_1(\mathbb{R}) = \{u \in \mathcal{C}_b^\infty(\mathbb{R}) \text{ s.t. } u' \in \mathbb{S}(\mathbb{R})\}. \tag{5.16}$$

Taking into account the system (5.1), we perform the expansion in powers of t of the difference between T^t and $\mathcal{S}_\delta^t$, given by (5.7).

Theorem 5.2. *Assume that u_0 belongs to $\mathbb{S}_1(\mathbb{R})$, and that f belongs to $\mathcal{C}^\infty(\mathbb{R})$. For t and δ small enough, the following asymptotic holds*

$$\begin{aligned}T^t u_0 - \mathcal{S}_\delta^t u_0 &= -\delta t^2 f''(u_0)\,(\partial_x u_0)^2 \\ &\quad +\frac{t^3}{24}\Big(f'(u_0)f''(u_0) + f(u_0)f^{(3)}(u_0)\Big)\,(\partial_x u_0)^2 \\ &\quad -\frac{t^3}{12}f^{(4)}(u_0)\,(\partial_x u_0)^4 - \frac{t^3}{3}f^{(3)}(u_0)\,(\partial_x u_0)^2\,\partial_x^2 u_0 \\ &\quad -\frac{t^3}{6}f''(u_0)\,\big(\partial_x^2 u_0\big)^2 + \mathcal{O}(\delta t^3) + \mathcal{O}(t^4).\end{aligned} \tag{5.17}$$

Proof. We introduce the two Lie operators D_Δ and D_f associated with ∂_x^2 and f, and write

$$T^t u_0 - \mathcal{S}_\delta^t u_0 = \left(\mathrm{e}^{t(D_\Delta + D_f)}\mathrm{Id}\right) u_0 - \left(\mathrm{e}^{(1/2+\delta)tD_f}\mathrm{e}^{tD_\Delta}\mathrm{e}^{(1/2-\delta)tD_f}\mathrm{Id}\right) u_0. \tag{5.18}$$

From Theorem 5.1, we can deduce that

$$\begin{aligned} T^t u_0 - \mathcal{S}_\delta^t u_0 &= -\delta t^2 \left([D_f, D_\Delta]\mathrm{Id}\right) u_0 + \frac{t^3}{24}\left(\left[D_f, [D_f, D_\Delta]\right]\mathrm{Id}\right) u_0 \\ &\quad + \frac{t^3}{12}\left(\left[D_\Delta, [D_f, D_\Delta]\right]\mathrm{Id}\right) u_0 + \mathcal{O}(\delta t^3) + \mathcal{O}(t^4). \end{aligned} \tag{5.19}$$

We are not interested in giving the exact form of the terms $\mathcal{O}(\delta t^3)$ and $\mathcal{O}(t^4)$, but these terms can be computed following the same technique developed in [DT11]. For the term in $\mathcal{O}(t^2)$, we have by definition (1.49) and with (1.40):

$$\begin{aligned} \left([D_f, D_\Delta]\mathrm{Id}\right) u_0 &= \left(D_f(D_\Delta \mathrm{Id}) - D_\Delta(D_f \mathrm{Id})\right) u_0 \\ &= (D_\Delta \mathrm{Id})'(u_0) f(u_0) - (D_f \mathrm{Id})'(u_0)\partial_x^2 u_0 \\ &= \partial_x^2\left(f(u_0)\right) - f'(u_0)\partial_x^2 u_0. \end{aligned} \tag{5.20}$$

The last term is by definition the Lie bracket between ∂_x^2 and f, a simple computation shows that

$$\begin{aligned} \partial_x^2 f(u_0) - f'(u_0)\partial_x^2 u_0 &= f''(u_0)\left(\partial_x u_0\right)^2 + f'(u_0)\partial_x^2 u_0 - f'(u_0)\partial_x^2 u_0 \\ &= f''(u_0)\left(\partial_x u_0\right)^2. \end{aligned} \tag{5.21}$$

Furthermore,

$$\left(\left[D_f, [D_f, D_\Delta]\right]\mathrm{Id}\right)(u_0) = \left(f'(u_0)f''(u_0) + f(u_0)f^{(3)}(u_0)\right)\left(\partial_x u_0\right)^2, \tag{5.22}$$

and

$$\begin{aligned} \left(\left[D_\Delta, [D_f, D_\Delta]\right]\mathrm{Id}\right) u_0 &= -f^{(4)}(u_0)\left(\partial_x u_0\right)^4 - 4f^{(3)}(u_0)\left(\partial_x u_0\right)^2 \partial_x^2 u_0 \\ &\quad -2f''(u_0)\left(\partial_x^2 u_0\right)^2. \end{aligned} \tag{5.23}$$

All the terms in (5.19) are now computed and this concludes the proof of the theorem. □

For $\delta = 0$, the next corollary follows directly.

Corollary 5.3. *Assume that u_0 belongs to $\mathbb{S}_1(\mathbb{R})$, and that f belongs to $\mathcal{C}^\infty(\mathbb{R})$. For t small enough, the following asymptotic holds*

$$\begin{aligned} T^t u_0 - \mathcal{S}^t u_0 &= \frac{t^3}{24}\left(f'(u_0)f''(u_0) + f(u_0)f^{(3)}(u_0)\right)\left(\partial_x u_0\right)^2 \\ &\quad -\frac{t^3}{12}f^{(4)}(u_0)\left(\partial_x u_0\right)^4 - \frac{t^3}{3}f^{(3)}(u_0)\left(\partial_x u_0\right)^2\partial_x^2 u_0 \\ &\quad -\frac{t^3}{6}f''(u_0)\left(\partial_x^2 u_0\right)^2 + \mathcal{O}(t^4). \end{aligned} \tag{5.24}$$

5.3.3 Domain of Validity of the Numerical Scheme

From (5.17) and (5.24), we can see that

$$\mathcal{S}^t u_0 - \mathcal{S}^t_\delta u_0 = \delta t^2 f''(u_0)\,(\partial_x u_0)^2 + \mathcal{O}(\delta t^3), \tag{5.25}$$

and thus

$$T^t u_0 - \mathcal{S}^t_\delta u_0 = \underbrace{T^t u_0 - \mathcal{S}^t u_0}_{\mathcal{O}(t^3)} + \underbrace{\mathcal{S}^t u_0 - \mathcal{S}^t_\delta u_0}_{\mathcal{O}(\delta t^2)}. \tag{5.26}$$

Therefore, we are sure that the real local error of the method, $T^t u_0 - \mathcal{S}^t u_0$, will be bounded by the local error estimate, $err = \mathcal{S}^t u_0 - \mathcal{S}^t_\delta u_0$, when for a given δ:

$$T^t u_0 - \mathcal{S}^t_\delta u_0 \approx \mathcal{O}(t^2), \tag{5.27}$$

is verified into (5.26), *i.e.*, when the embedded method is really of lower order as assumed in (5.9).
The latter will be always verified for sufficiently small time steps t, for which

$$T^t u_0 - \mathcal{S}^t u_0 \approx \mathcal{O}(t^3) < err = \mathcal{S}^t u_0 - \mathcal{S}^t_\delta u_0 \approx \mathcal{O}(\delta t^2), \tag{5.28}$$

is guaranteed, and thus yields (5.27) into (5.26). Nevertheless, for larger time steps err will fail to properly predict $T^t u_0 - \mathcal{S}^t u_0$, since we will eventually have

$$T^t u_0 - \mathcal{S}^t u_0 \approx \mathcal{O}(t^3) > err = \mathcal{S}^t u_0 - \mathcal{S}^t_\delta u_0 \approx \mathcal{O}(\delta t^2). \tag{5.29}$$

When this happens, (5.27) is no longer true and the previous estimates show that we will rather have

$$T^t u_0 - \mathcal{S}^t_\delta u_0 \approx \mathcal{O}(t^3), \tag{5.30}$$

and assumption (5.9) will no longer hold.
In order to overcome this difficulty, we must estimate a *critical time step* $t^\star > 0$ such that for all t in $[0, t^\star]$, (5.27) is guaranteed for a given δ. This parameter fixes a *domain of validity* of the adaptive scheme in which the Strang local error, $T^t u_0 - \mathcal{S}^t u_0$, will be indeed bounded by the local error estimate, err, and an effective error control will be achieved for err smaller than a given accuracy tolerance η into (5.10). Finally, a suitable choice of δ can be also made since $t^\star$ is related to δ following (5.26). A natural strategy to predict this critical $t^\star$ will rely on the previous theoretical estimates, and on a more precise knowledge of the structure of the solutions of the PDEs. The latter is, for instance, illustrated in the next part in the context of traveling wave solutions similar to the application background established in Chapter 4.

5.4 Application to Traveling Waves

In this part, we will confront the previous theoretical study to a simple reaction-diffusion problem that admits self-similar traveling wave solutions such as the KPP equation [KPP37]. The main advantages of considering this kind of problem are that analytic solutions exist and that the featured stiffness can be tuned using a space-time scaling. Therefore, it provides a first numerical validation of the numerical estimates of the method and an evaluation of its domain of application. Moreover, a detailed study can be conducted on the impact of the stiffness featured by propagating fronts with steep spatial gradients as performed, for instance, in [DDM11]. In what follows, we recast previous estimates in the context of these reaction traveling waves, and then deduce an estimate of the time step $t^\star$ that defines the limit of application of the method for which local error estimates yield effective error control. We end with a numerical validation of the theoretical results in the context of the numerical solution of the KPP model.

5.4.1 Numerical Estimates

We consider the propagation of self-similar waves modeled by parabolic PDEs of type:

$$\left.\begin{array}{ll} \partial_t u - D\,\partial_x^2 u = kf(u), & x \in \mathbb{R},\ t > 0, \\ u(0,x) = u_0(x), & x \in \mathbb{R}, \end{array}\right\} \tag{5.31}$$

with solution $u(x,t) = u_0(x - ct)$, where c is the steady speed of the wavefront, and D and k stand, respectively, for diffusion and reaction coefficients.
Considering Theorem 5.2, we obtain the following estimate for system (5.31).

Corollary 5.4. *Assume that u_0 belongs to $\mathbb{S}_1(\mathbb{R})$, and that f belongs to $\mathcal{C}^\infty(\mathbb{R})$. For t and δ small enough, the following asymptotic holds*

$$\begin{aligned} T^t u_0 - \mathcal{S}_\delta^t u_0 \;=\; & -\delta k D t^2 f''(u_0)\,(\partial_x u_0)^2 \\ & + \frac{k^2 D t^3}{24}\left(f'(u_0)f''(u_0) + f(u_0)f^{(3)}(u_0)\right)(\partial_x u_0)^2 \\ & - \frac{k D^2 t^3}{12} f^{(4)}(u_0)\,(\partial_x u_0)^4 - \frac{k D^2 t^3}{3} f^{(3)}(u_0)\,(\partial_x u_0)^2\,\partial_x^2 u_0 \\ & - \frac{k D^2 t^3}{6} f''(u_0)\left(\partial_x^2 u_0\right)^2 + \mathcal{O}(\delta t^3) + \mathcal{O}(t^4). \end{aligned} \tag{5.32}$$

Proof. The proof follows directly the demonstration of Theorem 5.2 using (5.19) and considering that

$$[D_{kf}, D_{D\Delta}] = kD[D_f, D_\Delta], \tag{5.33}$$

$$\left[[D_{kf}, D_{D\Delta}], D_{D\Delta}\right] = kD^2\left[[D_f, D_\Delta], D_\Delta\right], \tag{5.34}$$

$$\left[[D_{kf}, D_{D\Delta}], D_{kf}\right] = k^2 D\left[[D_f, D_\Delta], D_f\right], \tag{5.35}$$

where $D_{D\Delta}$ and D_{kf} are the Lie operators associated with $D\partial_x^2$ and kf. □

If we now consider the system (5.31) with $k = 1$ and $D = 1$, *i.e.*, the original problem (5.1), the following corollary establishes $t^\star > 0$ for a given δ such that for all t in $[0, t^\star]$, the condition (5.27) is guaranteed.

Corollary 5.5. *Assume that u_0 belongs to $\mathbb{S}_1(\mathbb{R})$, and that f belongs to $\mathcal{C}^\infty(\mathbb{R})$. For a given δ small enough, define*

$$M_1 = \left\| f''(u_0)\,(\partial_x u_0)^2 \right\|_{L^2}, \tag{5.36}$$

$$\begin{aligned} M_2 \;=\; \Bigg\| & \frac{f'(u_0)f''(u_0) + f(u_0)f^{(3)}(u_0)}{24}\,(\partial_x u_0)^2 - \frac{f^{(4)}(u_0)}{12}\,(\partial_x u_0)^4 \\ & - \frac{f^{(3)}(u_0)}{3}\,(\partial_x u_0)^2\,\partial_x^2 u_0 - \frac{f''(u_0)}{6}\left(\partial_x^2 u_0\right)^2 \Bigg\|_{L^2}, \end{aligned} \tag{5.37}$$

and define $t^\star$ by

$$t^\star M_2 = \delta M_1. \tag{5.38}$$

For all t such that $0 < t \le t^\star$, the following holds

$$\|T^t u_0 - \mathcal{S}_\delta^t u_0\|_{L^2} \approx \mathcal{O}(t^2). \tag{5.39}$$

In general, if the evaluation of the derivatives of u_0 and f is feasible, it is then possible to predict the domain of application of the method $[0, t^\star]$, for a given δ, according to the previous result.
In the particular case of traveling wave solutions for (5.31), the diffusion and reaction coefficients, D and k, might be seen as scaling coefficients in time and space. A dimensionless analysis of a traveling wave, as shown in [GS94], can be then conducted considering a dimensionless time τ and a dimensionless space r with

$$\tau = kt, \qquad r = (k/D)^{1/2}x. \tag{5.40}$$

This analysis allows us to find a steady velocity of the wavefront:

$$c = \mathrm{d}_t x \propto (Dk)^{1/2}, \tag{5.41}$$

whereas the sharpness of the wave profile is measured by

$$\mathrm{d}_x u|_{\max} \propto (k/D)^{1/2}. \tag{5.42}$$

Therefore, condition $Dk = 1$ implies constant velocity for all $k = 1/D$, but greater k (or smaller D) implies higher spatial gradients, and thus stiffer configurations.
This study gives complementary information on the solution of (5.31), and in particular when condition $Dk = 1$ is satisfied, it allows us to deduce from Corollary 5.5 the following expression

$$kt^\star M_2 = \delta M_1, \tag{5.43}$$

with M_1 and M_2 given, respectively, by (5.36) and (5.37). Stiffer configurations with steeper spatial gradients and thus larger k, will then restrain the application domain of the method according to (5.43). Nevertheless, smaller time steps are also required for higher gradients for a given level of accuracy, and hence we can expect a simultaneous reduction of both critical and accurate splitting time steps such that t remains into the domain of application $[0, t^\star]$.

5.4.2 Numerical Illustration: 1D KPP Equation

Let us recall the Kolmogorov-Petrovskii-Piskunov model. In their original paper [KPP37], these authors introduced a model describing the propagation of a virus, and the first rigorous analysis of a stable traveling wave solution of a nonlinear reaction-diffusion equation [GS94]. The equation is the following:

$$\partial_t u - D\,\partial_x^2 u = k\,u^2(1-u). \tag{5.44}$$

We consider a 1D discretization with 5001 points on a $[-70, 70]$ region with homogeneous Neumann boundary conditions, for which we have negligible spatial discretization errors with respect to the ones coming from the numerical time integration.
The description of the dimensionless model and the structure of the exact solution can be found in [GS94], where a theoretical analysis shows that in the case of $D = 1$ and $k = 1$, the velocity of the self-similar traveling wave is $c = 1/\sqrt{2}$ in (5.41) and the maximum gradient value reaches $1/\sqrt{32}$ in (5.42). The key point of this illustration is that the velocity of the traveling wave is proportional to $(k\,D)^{1/2}$, whereas the maximum gradient is proportional to $(k/D)^{1/2}$. Hence, we consider the case $kD = 1$ for which one may obtain steeper gradients with the same speed of propagation.
Throughout this section, the exact solution $T^t u_0$ will be approximated by a reference or *quasi-exact* solution given by the numerical solution of the coupled reaction-diffusion problem performed by the Radau5 method [HW96] with fine tolerance, $\eta_{Radau5} = 10^{-10}$. The

Strang approximations $\mathcal{S}^t u_0$ and $\mathcal{S}^t_\delta u_0$ will be computed with the splitting technique of Chapter 4, with Radau5 for the reactive term, and the ROCK4 method [Abd02] for the diffusion problem. In order to properly discriminate the previously estimated splitting errors from those coming from the temporal integration of the substeps, we consider also fine tolerances, $\eta_{\text{Radau5}} = \eta_{\text{ROCK4}} = 10^{-10}$. Figures 5.1 and 5.2 show L^2 errors between the $T^t u_0$, $\mathcal{S}^t u_0$, and $\mathcal{S}^t_\delta u_0$ solutions for $k = 1$, $k = 10$, and $k = 100$, and several values of δ. Notice that estimates (5.17), (5.24), and (5.25) for all three errors in (5.26), are verified. In particular, for Δt larger than critical $\Delta t^\star$, the estimated error $err = \|\mathcal{S}^{\Delta t} u_0 - \mathcal{S}^{\Delta t}_\delta u_0\|_{L^2}$ is no longer predicting the real local error given by $T^t u_0 - \mathcal{S}^t u_0$.

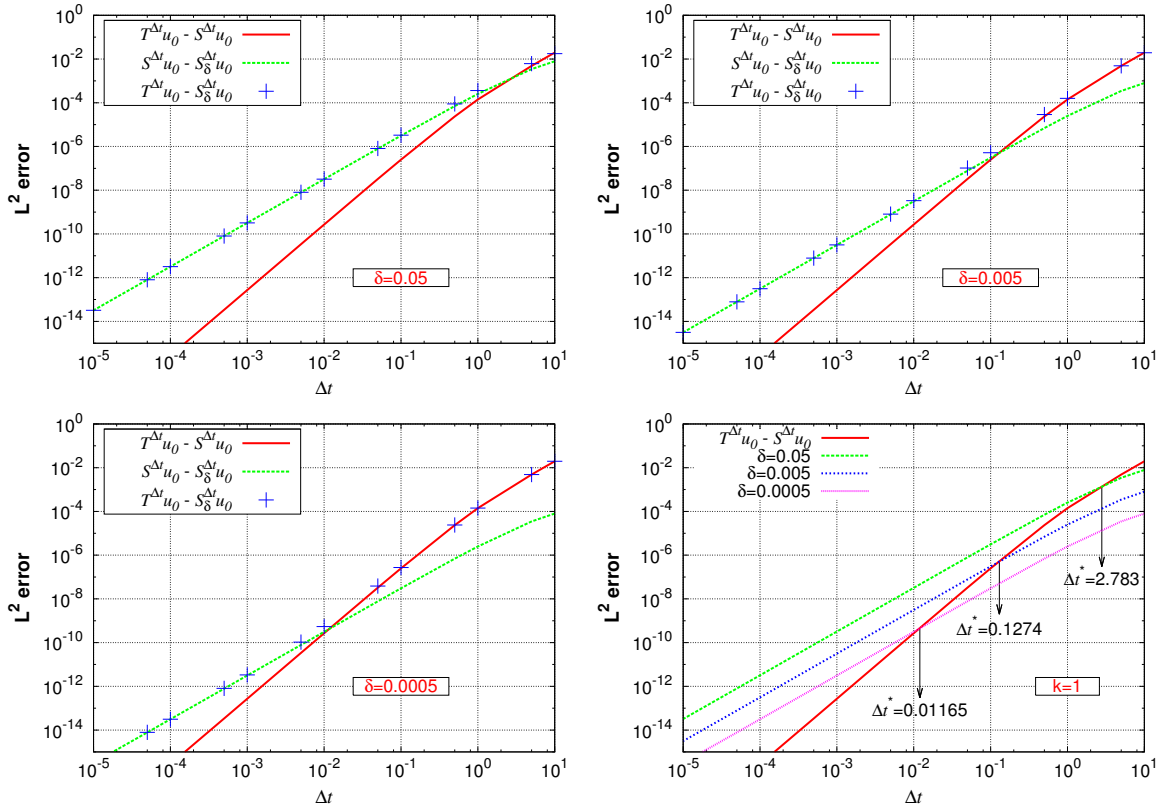

Fig. 5.1: *1D KPP equation with $k = 1$. Local L^2 errors for several splitting time steps Δt and $\delta = 0.05$ (top left),* 0.005 *(top right), and* 0.0005 *(bottom left). Bottom right: critical splitting time steps $\Delta t^\star$ obtained when $\|T^{\Delta t} u_0 - \mathcal{S}^{\Delta t} u_0\|_{L^2} \approx \|\mathcal{S}^{\Delta t} u_0 - \mathcal{S}^{\Delta t}_\delta u_0\|_{L^2}$, in the numerical tests.*

With these results, we can also compare the real $\Delta t^\star_{\text{real}}$, obtained when $\|T^{\Delta t} u_0 - \mathcal{S}^{\Delta t} u_0\|_{L^2} \approx \|\mathcal{S}^{\Delta t} u_0 - \mathcal{S}^{\Delta t}_\delta u_0\|_{L^2}$ in the numerical tests, with the theoretically estimated $\Delta t^\star_{\text{est}}$ following (5.43). Table 5.1 summarizes these results where $\Delta t^\star_{\text{est}}$ is given in (5.43) by the computation of M_1 and M_2 with Maple© according to (5.36) and (5.37). A really good agreement can be observed even though the theoretical results underestimate the real values. The loss of order depicted by the numerical results, is due to the influence of spatial gradients in the solution, as previously analyzed in Chapter 1 [DDLM07]. This explains the error of the predicted critical $\Delta t^\star_{\text{est}}$ in (5.43), whenever one gets close to the order loss region.
The numerical results also show that $\|\mathcal{S}^{\Delta t} u_0 - \mathcal{S}^{\Delta t}_\delta u_0\|_{L^2} \propto \delta$ for a given Δt according to (5.25), and consequently $\Delta t^\star \propto \delta$. Therefore, the working region or domain of application of the method, $\Delta t < \Delta t^\star$, depends directly on the choice of δ, as seen in Table 5.1. In the context

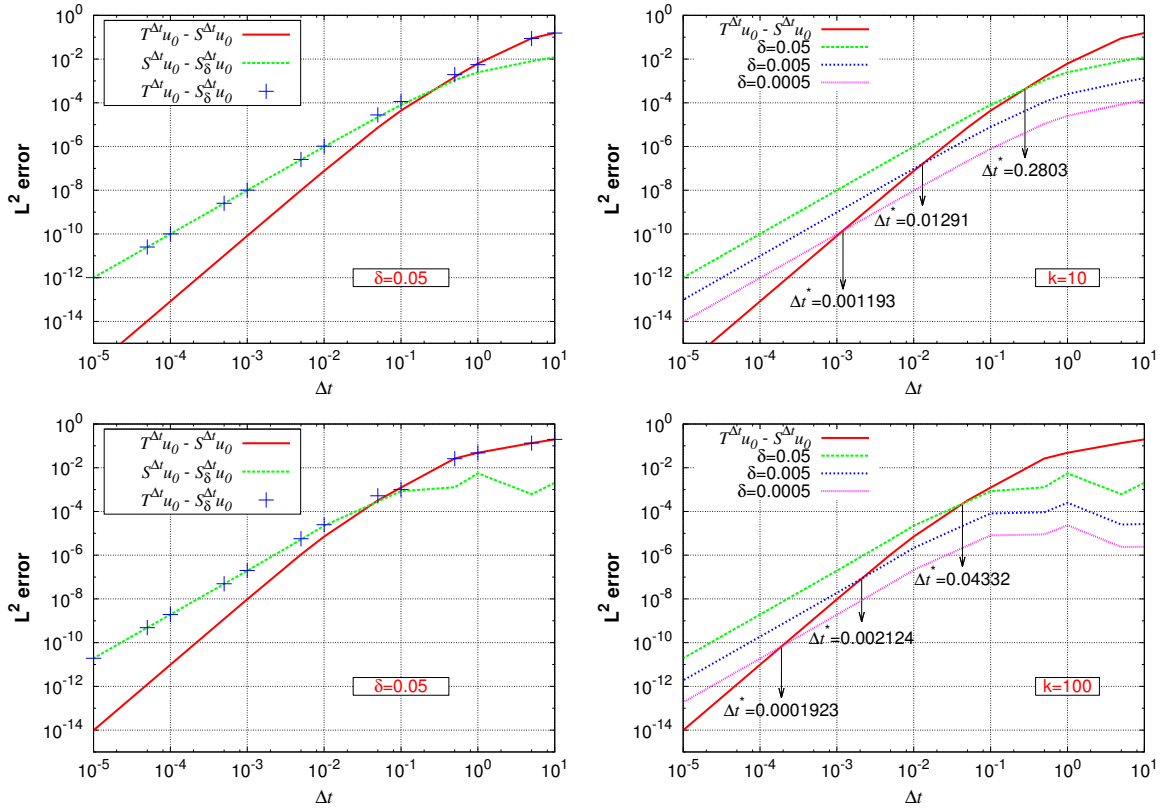

Fig. 5.2: *1D KPP equation with $k = 10$ (top) and $k = 100$ (bottom). Left: local L^2 errors for several splitting time steps Δt and $\delta = 0.05$. Right: critical splitting time steps $\Delta t^\star$ obtained when $\|T^{\Delta t}u_0 - \mathcal{S}^{\Delta t}u_0\|_{L^2} \approx \|\mathcal{S}^{\Delta t}u_0 - \mathcal{S}_\delta^{\Delta t}u_0\|_{L^2}$, in the numerical tests.*

of traveling waves, these numerical experiments show that $\Delta t^\star \propto k^{-1} \propto 1/\|\partial_x u_0\|_\infty$ according to Table 5.1. Therefore, the application domains are reduced for stiffer configurations but the numerical results also show that smaller time steps are required for the same level of accuracy. These conclusions are easily extrapolated to more general self-similar propagating waves.

Table 5.1: *1D KPP equation. Comparison between the real $\Delta t^\star_{\text{real}}$, obtained when $\|T^{\Delta t}u_0 - \mathcal{S}^{\Delta t}u_0\|_{L^2} \approx \|\mathcal{S}^{\Delta t}u_0 - \mathcal{S}_\delta^{\Delta t}u_0\|_{L^2}$ in the numerical tests, and the theoretically estimated $\Delta t^\star_{\text{est}}$ following (5.43).*

	δ	0.05	0.005	0.0005
$k = 1$	$\Delta t^\star_{\text{real}}$	2.783	0.1274	1.17×10^{-2}
	$\Delta t^\star_{\text{est}}$	1.107	0.1107	1.11×10^{-2}
$k = 10$	$\Delta t^\star_{\text{real}}$	0.2803	1.29×10^{-2}	1.19×10^{-3}
	$\Delta t^\star_{\text{est}}$	0.1107	1.11×10^{-2}	1.11×10^{-3}
$k = 100$	$\Delta t^\star_{\text{real}}$	4.33×10^{-2}	2.12×10^{-3}	1.92×10^{-4}
	$\Delta t^\star_{\text{est}}$	1.11×10^{-2}	1.11×10^{-3}	1.11×10^{-4}

5.5 Numerical Study on Non-Asymptotic Regimes

We have presented so far a time adaptive numerical scheme fully based on theoretical error estimates. Nevertheless, some general conditions must be taken into consideration in order to guarantee an effective error control. This has been shown in the case of reaction traveling waves for which theoretical studies give us some insights into the PDE solution. Nevertheless, this is not always possible and it is usually difficult to carry out such an analysis for more realistic models. Based on the theoretical analysis and previous illustrations, a general numerical procedure that complements the adaptive scheme should be introduced. The main goal of the following study is to settle the theoretical framework for non-asymptotic regimes, and thus an appropriate numerical procedure to estimate $t^\star$, and a suitable shift time δ.
Let us consider the general system (5.1), we can write based on the theoretical estimates (5.24) and (5.25):

$$\mathcal{S}^{\Delta t}u_0 - T^{\Delta t}u_0 = C_0\Delta t^3, \tag{5.45}$$

where $C_0 = C_1(u_0) + \mathcal{O}(\Delta t^4)$, and

$$\mathcal{S}^{\Delta t}u_0 - \mathcal{S}_\delta^{\Delta t}u_0 = \delta C_\delta \Delta t^2, \tag{5.46}$$

where $C_\delta = C_2(u_0) + \mathcal{O}(\delta, \Delta t^3)$; the dependence of C_δ on δ is only given in the higher order terms and it is thus neglected. For a given δ, in the same spirit of Corollary 5.5, we search for a critical $\Delta t^\star$ such that

$$\left\|\mathcal{S}^{\Delta t}u_0 - T^{\Delta t}u_0\right\| \leq \left\|\mathcal{S}^{\Delta t}u_0 - \mathcal{S}_\delta^{\Delta t}u_0\right\|, \tag{5.47}$$

for all $\Delta t \leq \Delta t^\star$. According to (5.45) and (5.46), we have then the following estimate:

$$\Delta t^\star \approx \frac{\delta C_\delta}{C_0}. \tag{5.48}$$

The latter estimate establishes for a given δ an upper bound for the time steps for which the local error estimate, $err = \|\mathcal{S}^{\Delta t}u_0 - \mathcal{S}_\delta^{\Delta t}u_0\|$, is properly estimating the real Strang local error, $\left\|\mathcal{S}^{\Delta t}u_0 - T^{\Delta t}u_0\right\|$, following (5.47). In particular, when $\Delta t \to \Delta t^\star$, we have that

$$err \approx \left\|\mathcal{S}^{\Delta t}u_0 - T^{\Delta t}u_0\right\|, \tag{5.49}$$

and the local error estimate is predicting more accurately the real error of integration. The critical time step $\Delta t^\star$ is directly related to δ through (5.48), as already concluded in the previous numerical results. Therefore, a suitable δ will define a critical $\Delta t^\star$ such that the estimated splitting time steps Δt for a given tolerance η, will be sufficiently close to the critical $\Delta t^\star$, and thus an excessive overestimate of the Strang local error is avoided. In this way, larger time steps can be chosen for a given accuracy tolerance η.
In order to compute $\Delta t^\star$ for a given δ, we must first estimate C_0 in (5.48), since C_δ is computed out of the local error estimate, err, for known Δt and δ in (5.46). Estimating C_0 amounts to directly estimate Strang local error through (5.45), and thus the accuracy of the simulation might be controlled in this way without relying on a local error estimate computed with a second, embedded method. Nevertheless, as we will see in the following, in order to estimate C_0 and the Strang local error, we must define new local estimators and a numerical procedure that becomes rapidly very expensive if we want to implement only such an error control technique. Therefore, we should rather rely on a local error estimate given by a less expensive strategy for which the computation of C_0 is only performed from time to time, to guarantee the validity of the local error estimates.
The next Lemma will be useful to define the numerical procedure to estimate C_0.

Lemma 5.6. *Let us consider system (5.1), and assume a local Lipschitz condition for f:*

$$\|f(u) - f(v)\| \leq \lambda \|u - v\| . \tag{5.50}$$

For a finite Δt, the following holds

$$\left\|T^{\Delta t}u_0 - T^{\Delta t}v_0\right\| \leq \omega \|u_0 - v_0\| , \tag{5.51}$$

with $\omega = 1 + \kappa\Delta t$ for small enough Δt.

Proof. Using Duhamel's formula for (5.1) yields

$$T^t u_0 - T^t v_0 = \mathrm{e}^{t\partial_x^2}(u_0 - v_0) + \int_0^t \mathrm{e}^{(t-s)\partial_x^2} \left(f(T^s u_0) - f(T^s v_0)\right) \mathrm{d}s. \tag{5.52}$$

Taking norms and applying recursively (5.52),

$$\begin{aligned} \left\|T^t u_0 - T^t v_0\right\| &\leq \|u_0 - v_0\| + \lambda \int_0^t \|T^s u_0 - T^s v_0\| \,\mathrm{d}s \\ &\leq \mathrm{e}^{\lambda t}\|u_0 - v_0\|, \end{aligned} \tag{5.53}$$

proves (5.51) for $t = \Delta t$ finite. □

If we define a local estimator:

$$e_1 = \mathcal{S}^{a_1\Delta t}u_0 - \mathcal{S}^{b_1\Delta t}(\mathcal{S}^{c_1\Delta t}u_0), \tag{5.54}$$

such that $a_1 = b_1 + c_1$, we obtain that

$$\begin{aligned} \mathcal{S}^{b_1\Delta t}(\mathcal{S}^{c_1\Delta t}u_0) - T^{a_1\Delta t}u_0 &= \mathcal{S}^{b_1\Delta t}(\mathcal{S}^{c_1\Delta t}u_0) - T^{b_1\Delta t}(\mathcal{S}^{c_1\Delta t}u_0) \\ &\quad + T^{b_1\Delta t}(\mathcal{S}^{c_1\Delta t}u_0) - T^{b_1\Delta t}(T^{c_1\Delta t}u_0) \\ &= C_{\mathcal{S}^{c_1\Delta t}u_0} b_1^3 \Delta t^3 + T^{b_1\Delta t}(\mathcal{S}^{c_1\Delta t}u_0) \\ &\quad - T^{b_1\Delta t}(T^{c_1\Delta t}u_0), \end{aligned} \tag{5.55}$$

where $C_{\mathcal{S}^{c_1\Delta t}u_0} = C_1(\mathcal{S}^{c_1\Delta t}u_0) + \mathcal{O}(\Delta t^4)$. Assuming that $C_{\mathcal{S}^{c_1\Delta t}u_0} \approx C_0$, and considering Lemma 5.6, it follows from the difference between (5.45) at $a_1\Delta t$ and (5.55):

$$\begin{aligned} \|e_1 - (a_1^3 - b_1^3)C_0\Delta t^3\| &\leq \omega\|T^{c_1\Delta t}u_0 - \mathcal{S}^{c_1\Delta t}u_0\| \\ &\leq \omega C_0 c_1^3 \Delta t^3. \end{aligned} \tag{5.56}$$

Defining a second local estimator:

$$e_2 = \mathcal{S}^{a_2\Delta t}u_0 - \mathcal{S}^{b_2\Delta t}(\mathcal{S}^{c_2\Delta t}u_0), \tag{5.57}$$

such that $a_2 = b_2 + c_2$, we obtain a second expression similar to (5.56), with e_2 and (a_2, b_2, c_2). We can thus estimate C_0 and ω. With this information we can either estimate the critical time step $\Delta t^\star$ for a given δ, or a suitable δ such that $\Delta t \approx \Delta t^\star$ through (5.48). In particular, we notice that

- b_1 should be close to b_2 in order to better approximate ω into (5.51) and (5.56); and

- c_1 and c_2 should be sufficiently small to guarantee $C_{\mathcal{S}^{c_1\Delta t}u_0} \approx C_0$ and $C_{\mathcal{S}^{c_2\Delta t}u_0} \approx C_0$.

From a practical point of view to optimize the number of extra computations, we can consider

- Using estimator e_2 to compute estimator e_1 by setting $a_2 = c_1$; and
- Setting $a_1 = 1$, so we can use $\mathcal{S}^{a_1\Delta t}u_0$ for the time integration of the problem.

Consequently, the extra computations needed to compute local estimators e_1 and e_2, will be given by

$$\mathcal{S}^{c_2\Delta t}u_0, \qquad \mathcal{S}^{b_2\Delta t}(\mathcal{S}^{c_2\Delta t}u_0), \qquad \mathcal{S}^{c_1\Delta t}u_0, \qquad \mathcal{S}^{b_1\Delta t}(\mathcal{S}^{c_1\Delta t}u_0), \tag{5.58}$$

within a time step Δt. We will then be able to compute ω and C_0, by solving two expressions of type (5.56).

5.5.1 Numerical Evaluation of Critical $t^\star$: 1D BZ Equation

We consider the 1D configuration of the BZ problem (4.29) with the same parameters (4.30) and (4.31), considered in Chapter 4, for which the spatial discretization of 4001 points is good enough to prevent important spatial discretization errors. The reference solution and the Strang approximations are defined in the same way as in the previous KPP application with the same tolerances for the time integration solvers. First of all, we validate for this case the theoretical order estimates (5.17), (5.24), and (5.25), and verify relation (5.26). Figure 5.3 shows L^2 errors between the $T^t u_0$, $\mathcal{S}^t u_0$, and $\mathcal{S}_\delta^t u_0$ solutions for several δ, and the real $\Delta t^\star$ obtained from the numerical results such that $\|T^{\Delta t}u_0 - \mathcal{S}^{\Delta t}u_0\|_{L^2} \approx \|\mathcal{S}^{\Delta t}u_0 - \mathcal{S}_\delta^{\Delta t}u_0\|_{L^2}$. The maximum L^2 errors consider the maximum value between the normalized local errors for variables a, b, and c. In these numerical tests, it corresponds usually to variable b.
Let us now define the two sets (a_1, b_1, c_1) and (a_2, b_2, c_2), and compute local estimators e_1 and e_2, in order to obtain C_0 according to (5.56) with $\Delta t = \Delta t_0 = 10^{-5}$, *i.e.*, a time step for which there is no order loss yet, as seen in Figure 5.3. As previously detailed:

1. We consider $a_1 = 1$ and $a_2 = c_1$, to avoid some extra computations.
2. Additionally, b_2 should be set close to b_1 with sufficiently small c_1 and c_2.

Setting b_1 larger than $1/2$ would yield a more different b_2, since $c_1 = a_2$. Alternatively, for b_1 smaller than $1/2$ we can even set $b_2 = b_1$, but in this case c_1 will be larger than $1/2$. Therefore, we reach a compromise by setting $b_1 = 1/2$ that yields $c_1 = a_2 = 1/2$, so we can choose, for instance, $b_2 = 2/5$ close to b_1, and hence,

$$a_1 = 1, \quad b_1 = 1/2, \quad c_1 = 1/2, \quad a_2 = 1/2, \quad b_2 = 2/5, \quad c_2 = 1/10. \tag{5.59}$$

With the local error estimate $err = \|\mathcal{S}^{\Delta t}u_0 - \mathcal{S}_\delta^{\Delta t}u_0\|_{L^2}$ for the various time steps and several δ shown in Figure 5.3, Figure 5.4 (left) presents the estimated critical $\Delta t^\star$ calculated with (5.48) from the estimated $C_0(\Delta t_0)$ and err. These critical time steps are in good agreement with the ones numerically measured in Figure 5.3, and depend on the value of δ. Hence, the domain of application or working region of the method, $\Delta t \leq \Delta t^\star$, might be settled depending on the desired level of accuracy by means of an appropriate choice of δ. For instance, if we consider the case $\delta = 0.05$ in Figure 5.3, for $\Delta t = 10^{-6}$, the local error estimate is given by $err \approx 10^{-10}$, whereas the real Strang local error is about 10^{-12}. This overestimate of the local error will certainly imply an underestimate of the size of the time steps, required for a given tolerance. Therefore, for a given tolerance η a more suitable configuration should consider a δ such that $\Delta t \approx \Delta t^\star$, in order to reduce excessive overestimates of local errors.

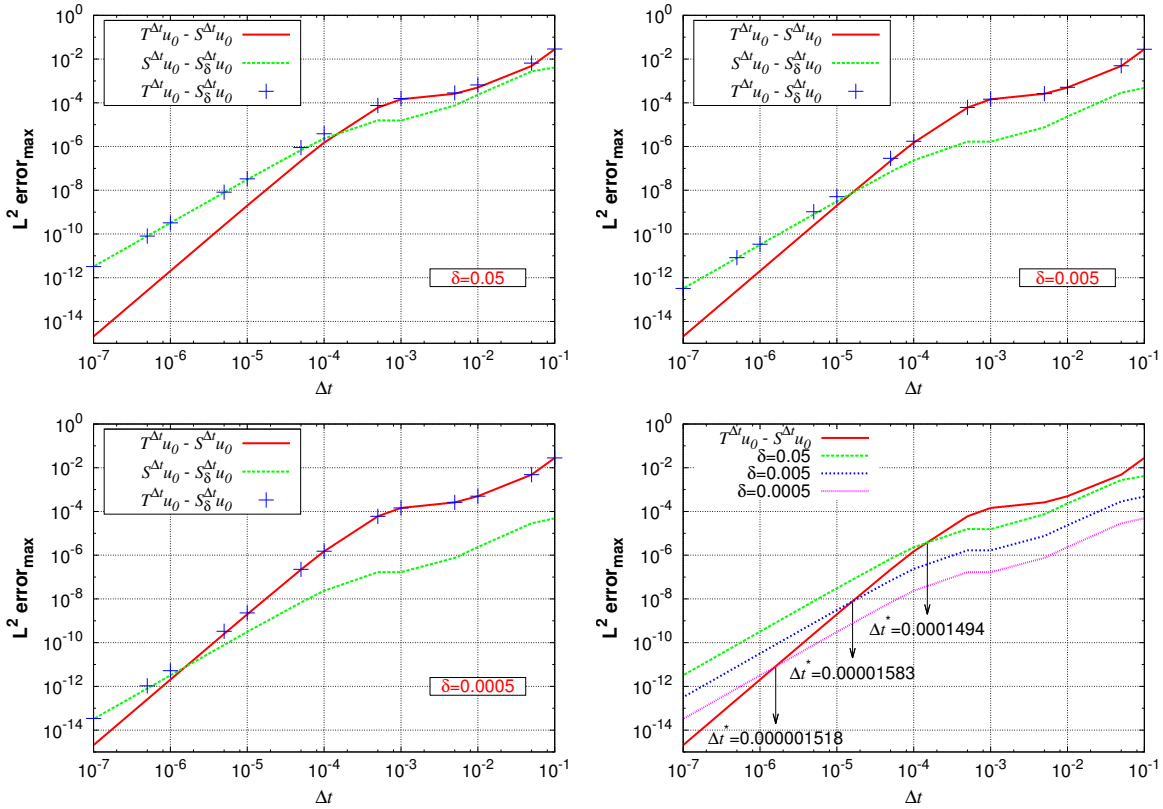

Fig. 5.3: *1D BZ equation. Maximum local L^2 errors for several splitting time steps Δt and $\delta = 0.05$ (top left), 0.005 (top right), and 0.0005 (bottom left). Bottom right: critical splitting time steps $\Delta t^\star$ obtained when $\|T^{\Delta t}u_0 - \mathcal{S}^{\Delta t}u_0\|_{L^2} \approx \|\mathcal{S}^{\Delta t}u_0 - \mathcal{S}^{\Delta t}_\delta u_0\|_{L^2}$, in the numerical tests.*

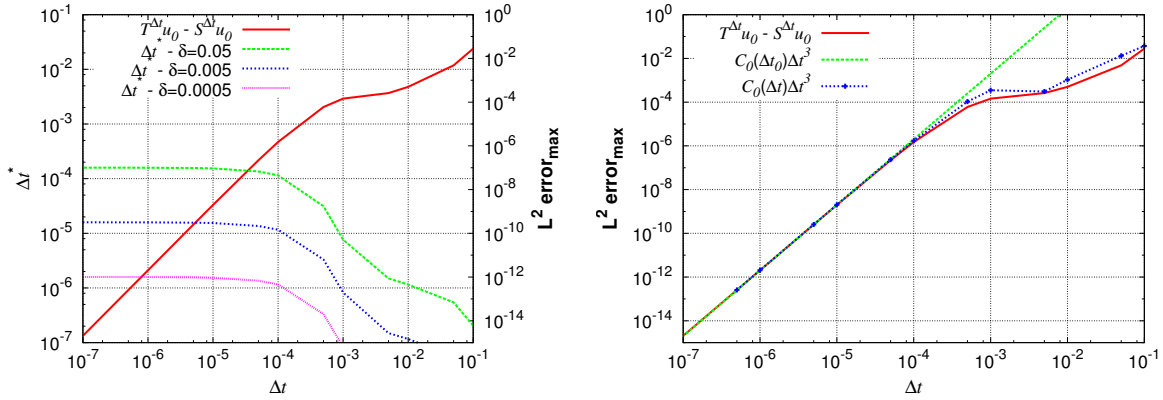

Fig. 5.4: *1D BZ equation. Left: working region of the method $\Delta t \leq \Delta t^\star$, with $\Delta t^\star$ calculated with C_0 estimated at $\Delta t_0 = 10^{-5}$ and err obtained for several splitting time steps Δt and δ. Right: predicted Strang error calculated with C_0 estimated at $\Delta t_0 = 10^{-5}$, and locally at several splitting time steps Δt.*

In the illustration shown in Figure 5.4 (left), C_0 was estimated in the third order region of the method and therefore, all values are well approximated as long as Δt remains in this region. In particular, critical $\Delta t^\star$ will be progressively underestimated for larger δ and consequently, it will impose smaller time steps for a given tolerance. This is already the case for $\delta = 0.05$, for which

$\Delta t^\star$ is in the transition zone towards the lower order region. Even though the computation of C_0 with a small time step will be less expensive, a much more accurate procedure considers the current time step, as shown in Figure 5.4 (right). In particular, by estimating locally C_0, we are estimating the real Strang error, and thus $\Delta t \leq \Delta t^\star$ guarantees the prescribed accuracy even if the asymptotic orders are no longer verified. This allows us to properly extend the domain of application of the adaptive scheme over the whole range of possible time steps for a given accuracy. The latter is an extremely important issue for real applications for which splitting time steps may go far beyond asymptotic behaviors, including the potential order reduction region associated with the stiffness of the problem, as already contemplated by the splitting strategy of Chapter 4.

5.6 General Description of the Numerical Method

The studies conducted in the context of traveling waves and non-asymptotic regimes allow us to properly complete the adaptive splitting strategy, initially introduced. We describe in this part the global description of the final adaptive splitting scheme.
Let us consider the general problem (1.52), where $U \in \mathbb{R}^m$, for which we use $\mathcal{S}_2^t$ in (5.5) as numerical solution scheme. Depending on the problem, the adaptive method will be applied considering the time evolution of $l \leq m$ variables: $\tilde{U} \in \mathbb{R}^l$. Let us denote Θ_l the set of indices of these variables. In order to consider only $l < m$ variables, the former ones must be decoupled from the remaining $m-l$ variables in the reactive term $F(\tilde{U})$ in (1.52). To simplify the presentation, we will only consider the set $\delta \in (0, \delta_{max})$, with $\delta_{max} < 1/2$.
The general scheme can be summarized by

1. We set the accuracy tolerance η, an initial time step Δt^0, and an initial shift δ^0.
2. We perform the time integration of (1.52) with the Strang scheme $\mathcal{S}_2^t$, and the embedded shifted one $\mathcal{S}_{2,\delta}^t$ given by (5.7).
3. We compute the local error estimate err and the new time step Δt^{new} according to (5.11).
4. If err is smaller than η, the current time step solution is accepted, and the simulation continues with the new Δt^{new} during next iteration.
5. Otherwise, the current solution is rejected and the time integration is recomputed with the new Δt^{new}.

In particular, it is better to choose rather small Δt^0 to avoid initial rejections. In order to guarantee an effective error control of the previous scheme, we define the working region $\Delta t \leq \Delta t^\star$ by estimating the corresponding $\Delta t^\star$ for the current δ. This is done for the first time step Δt^0 and then periodically, after N_δ accepted time steps depending on the problem, based on the numerical procedure previously detailed. The computation of the critical $\Delta t^\star$ is also performed with $\tilde{U}$, and a rather large initial δ^0 is suitable to initially guarantee $\Delta t \leq \Delta t^\star$. A suitable working region is thus defined by

$$\Delta t \in [\beta \Delta t^\star, \gamma \Delta t^\star], \tag{5.60}$$

with $0 < \beta < \gamma \leq 1$, for which splitting time steps are close to $\Delta t^\star$. A new shift δ is computed if

1. Δt is much lower than $\Delta t^\star$ ($\Delta t < \beta \Delta t^\star$) in order to avoid unnecessary small time steps; or

2. Δt is very close or possibly larger than $\Delta t^\star$ ($\Delta t > \gamma \Delta t^\star$) with γ close to 1, in order to increase the upper bound of the application domain.

The latter criteria guarantee that δ is dynamically computed and properly adapted to the dynamics of the phenomenon.

5.6.1 Algorithm Scheme

The numerical strategy is implemented as follows, where $\mathbf{U} \in \mathbb{R}^{m \times N_x}$ stands for the spatial discretization of U over N_x points: $\mathbf{U} := (u^{(i,k)})$ such that $i \in [1, m]$ and $k \in [1, N_x]$:

- **Input parameters.** Define accuracy tolerance η, time domain of study $[t_0, T]$, initial time step Δt^0, initial shift δ^0, and period of computation of $\Delta t^\star$: N_δ.
- **Initialization.** Set iteration counter ite $= 0$ and $t = t_0$, $\mathbf{U} = \mathbf{U^0}$, $\Delta t = \Delta t^0$, $\delta = \delta^0$. We define a flag *estimate* initialized as `.false.`. Throughout the whole computation, we need to save $\mathbf{U}$, an array of size $m \times N_x$.
- **Time evolution.** If $t < T$:
  1. Only if $\dfrac{\text{ite}}{N_\delta} = \left\lfloor \dfrac{\text{ite}}{N_\delta} \right\rfloor$ or *estimate* is `.true.`:
     Computation of the critical $\Delta t^\star$ I: For the sets (a_1, b_1, c_1) and (a_2, b_2, c_2) with $a_1 = 1$ and $a_2 = c_1$, we compute successively:
     - $\tilde{\mathbf{U}}^{\mathbf{1}} = \mathcal{S}^{c_2 \Delta t} \tilde{\mathbf{U}}^{\mathbf{0}}$, where $\tilde{\mathbf{U}}^{\mathbf{0}}$ is built out of $\mathbf{U}$, $\tilde{\mathbf{U}}^{\mathbf{0}} = (u^{(i,\cdot)})_{i \in \Theta_l}$;
     - $\tilde{\mathbf{U}}^{\mathbf{1}} = \mathcal{S}^{b_2 \Delta t} \tilde{\mathbf{U}}^{\mathbf{1}}$;
     - $\tilde{\mathbf{U}}^{\mathbf{2}} = \mathcal{S}^{c_1 \Delta t} \tilde{\mathbf{U}}^{\mathbf{0}}$;
     - $e_2 = \max_{i \in \Theta_l} \|\tilde{u}_2^{(i,\cdot)} - \tilde{u}_1^{(i,\cdot)}\|$;
     - $\tilde{\mathbf{U}}^{\mathbf{2}} = \mathcal{S}^{b_1 \Delta t} \tilde{\mathbf{U}}^{\mathbf{2}}$;
     - *estimate* is set to `.true.`.

     These operations need to save $\tilde{\mathbf{U}}^{\mathbf{1}}$ and $\tilde{\mathbf{U}}^{\mathbf{2}}$, two arrays of size $l \times N_x$.
  2. **Time integration over Δt:** We compute successively:
     - for each $k \in [1, N_x]$, $u_{\text{new}}^{(\cdot,k)} = Y^{\Delta t/2} u^{(\cdot,k)}$;
     - for each $k \in [1, N_x]$, $\tilde{u}_1^{(\cdot,k)} = Y^{\delta \Delta t}\, u_{\text{new}}^{(i,k)}\Big|_{i \in \Theta_l}$;
     - $\mathbf{U}^\star = X^{\Delta t} \mathbf{U}^\star$, with $\mathbf{U}^\star = (\mathbf{U}^{\text{new}}, \tilde{\mathbf{U}}^{\mathbf{1}})^T$;
     - for each $k \in [1, N_x]$, $u_\star^{(\cdot,k)} = Y^{(1/2-\delta)\Delta t} u_\star^{(\cdot,k)}$;
     - for each $k \in [1, N_x]$, $u_{\text{new}}^{(\cdot,k)} = Y^{\delta \Delta t} u_{\text{new}}^{(\cdot,k)}$;
     - $err = \max_{i \in \Theta_l} \|\tilde{u}_{\text{new}}^{(i,\cdot)} - \tilde{u}_1^{(i,\cdot)}\|$.

     We need to save $\mathbf{U}^{\text{new}}$, an array of size $m \times N_x$.
  3. Only if *estimate* is `.true.`:
     Computation of the critical $\Delta t^\star$ II: We compute successively:
     - $e_1 = \max_{i \in \Theta_l} \|\tilde{u}_{\text{new}}^{(i,\cdot)} - \tilde{u}_2^{(i,\cdot)}\|$;
     - C_0 using (5.56) with e_1 and e_2;
     - estimate $\Delta t^\star$ out of (5.48), and set $\Delta t^\star = \zeta \Delta t^\star$ with security factor $0 < \zeta \leq 1$, close to 1;
     - *estimate* is set to `.false.`.

- If $\Delta t \notin [\beta\Delta t^\star, \gamma\Delta t^\star]$ with $0 < \beta < \gamma \leq 1$: *estimate* is set to `.true.`.

4. Only if *estimate* is `.true.` and ite > 0:
   Computation of δ: According to (5.48) with *err*, C_0, and $\Delta t^\star = \Delta t$:
   - $\delta = \min\{\theta\delta, \delta_{max}\}$, with $\theta \geq 1$ as security factor;
   - computation of $\Delta t^\star$ with new δ;
   - *estimate* is set to `.false.`.
5. **Computation of Δt^{new}:** According to (5.11) with security factor $0 < \upsilon \leq 1$, close to 1.
   - If $\Delta t > \Delta t^\star$: set $err = \eta + C_\delta^0$ with $C_\delta^0 > 1$, used to potentially reject initial $\Delta t = \Delta t^0$.
   - If $\Delta t^{\text{new}} > \Delta t^\star$ and $\delta \neq \delta_{max}$: *estimate* is set to `.true.`.
   - If $err \leq \eta$: $t = t + \Delta t$, ite $=$ ite $+ 1$, and $\mathbf{U} = \mathbf{U}^{\text{new}}$.
   - $\Delta t = \min\{\Delta t^{\text{new}}, \Delta t^\star, T - t\}$.

In this splitting strategy, reaction is always integrated point by point if the reactive term is modeled by an ODEs system without spatial coupling. This integration can be performed completely in parallel as we will see in forthcoming chapters. Similarly, for linear diffusion problems as system (1.53), a better alternative considers a variable by variable solution:

$$u_\star^{(i,\cdot)} = X^{\Delta t} u_\star^{(i,\cdot)}, \tag{5.61}$$

for each $i \in [1, m] \bigcup \Theta_l$, that reduces the memory requirements and can be also performed in parallel [DDD+12, DMD+11b]. Depending on the problem, either the computation of critical $\Delta t^\star$ (steps (1), (3), and (4)), or the computation of δ (step (4)) can be potentially removed if one considers large enough δ^0 and sufficiently fine η. Finally, the whole strategy with all steps needs to save at worst two arrays of size $l \times N_x$ and other two of size $m \times N_x$, without considering the memory requirements of the diffusion and reaction solvers. All these issues will be thoroughly discussed in Chapter 8.

5.6.2 Numerical Evaluation of the Method: 1D BZ Equation Revisited

In this part, we evaluate the performance of the method in terms of accuracy of the simulation, and show that an effective control of the simulation error is performed following the previous solution scheme. Coming back to the BZ model, we perform a time integration of (4.29) with several accuracy tolerances η. First, we consider the adaptive numerical strategy without taking into account steps (1), (3), and (4), *i.e.,* without computing either critical $\Delta t^\star$ or δ. We set $\Delta t^0 = 10^{-7}$ and $\delta^0 = 0.05$ in all cases, with $t \in [0, 2]$. In this example, a rather small initial splitting time step is chosen to avoid initial rejections even though the rejection phases do not usually take many steps, as we will see in forthcoming examples. We have chosen an intermediate value of δ in order to clearly distinguish the different behaviors of the strategy in terms of prediction of the local errors, depending on the tolerance.
Figure 5.5 shows the time evolution of accepted splitting time steps Δt. As already studied in Chapter 4, the BZ equations model propagating self-similar waves, so splitting time step stabilizes after the overall phenomenon is solved approximately with the prescribed tolerance η. Local error estimates *err* are also displayed, which naturally verify the prescribed accuracy because we impose time steps for which *err* is limited by η according to (5.11).
Table 5.2 summarizes global L^2 errors between the splitting and reference solutions at the end of the time domain of study, $t = 2$. For a sufficiently fine η and consequently small enough

time steps, a precise error control is achieved by the local error control strategy, as we could have expected from previous results in Figure 5.3 for $\delta = 0.05$. Nevertheless, for $\eta = 10^{-4}$ we retrieve rather high global errors even if this configuration considers less time integration steps and thus less accumulation of local approximation errors. If we take a look at Figure 5.3, we note that for $\delta = 0.05$ and local errors of about 10^{-4}, the local error estimate *err* is not predicting properly the real Strang errors, as previously discussed, since $\Delta t > \Delta t^\star$. Therefore, a strategy that considers a more precise evaluation of errors must be considered for a larger range of time steps, whenever the required accuracy casts the method away from its asymptotic behavior. This is a standard difficulty of any time adaptive technique based on a lower order embedded method, and to our knowledge, a common problem that has not been studied much, and that this work tries to better investigate.

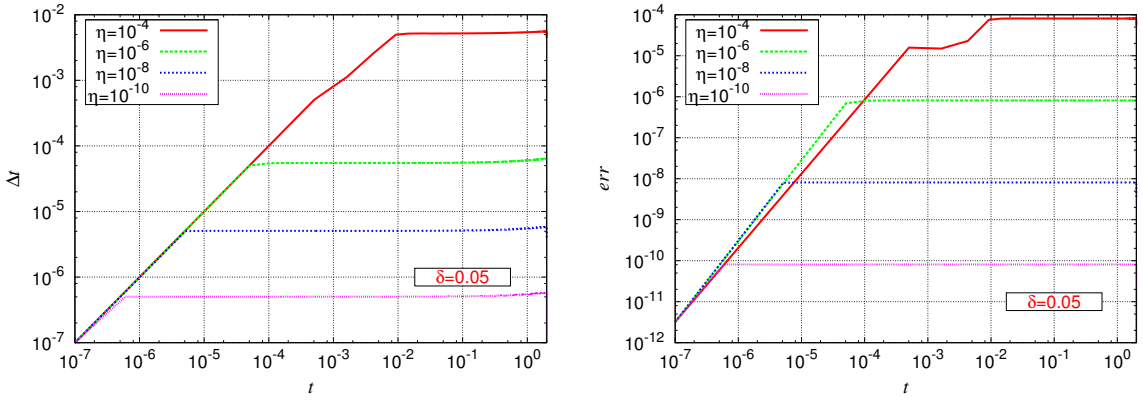

Fig. 5.5: *1D BZ equation. Time evolution of accepted splitting time steps Δt (left) and local L^2 error estimates $err = \|S^{\Delta t}u_0 - S_\delta^{\Delta t}u_0\|_{L^2}$ (right), for several tolerances η and $\delta = 0.05$.*

Table 5.2: *1D BZ equation. L^2 errors at final time $t = 2$ for (a, b, c) variables and several accuracy tolerances η.*

η	L^2 error a	L^2 error b	L^2 error c
10^{-4}	7.97×10^{-3}	1.07×10^{-2}	4.72×10^{-3}
10^{-6}	1.71×10^{-6}	1.83×10^{-6}	7.98×10^{-7}
10^{-8}	1.45×10^{-8}	1.54×10^{-8}	6.78×10^{-9}
10^{-10}	1.74×10^{-10}	1.75×10^{-10}	1.08×10^{-10}

Let us now consider the entire strategy with all steps, for several tolerances with $\Delta t^0 = 5 \times 10^{-7}$, and $\delta^0 = 0.05$. In the coming illustrations we have considered the following parameters:

- $\delta_{max} = 0.999$;
- $a_1 = 1$, $b_1 = c_1 = a_2 = 1/2$, $b_2 = 2/5$, and $c_2 = 1/10$ for the intermediate time steps, as in (5.59);
- $\zeta = 0.9$ as security factor of the critical $\Delta t^\star$ estimate;
- $\beta = 0.1$ and $\gamma = 0.95$ to define the working region (5.60);

- $\theta = 10$ as security factor of the δ estimate;
- $C_\delta^0 = 10$ to potentially reject the initial time step Δt^0; and
- $\upsilon = 0.9$ as security factor of the Δt^{new} estimate.

All local estimators, err, e_1, and e_2 are computed with normalized L^2 norms as in the previous chapter. Considering the propagating phenomenon, we set $N_\delta = 10$, but we estimate $\Delta t^\star$ only twice for ite $= 0$ and ite $= N_\delta$. Figure 5.6 shows the time evolution of the splitting time steps.

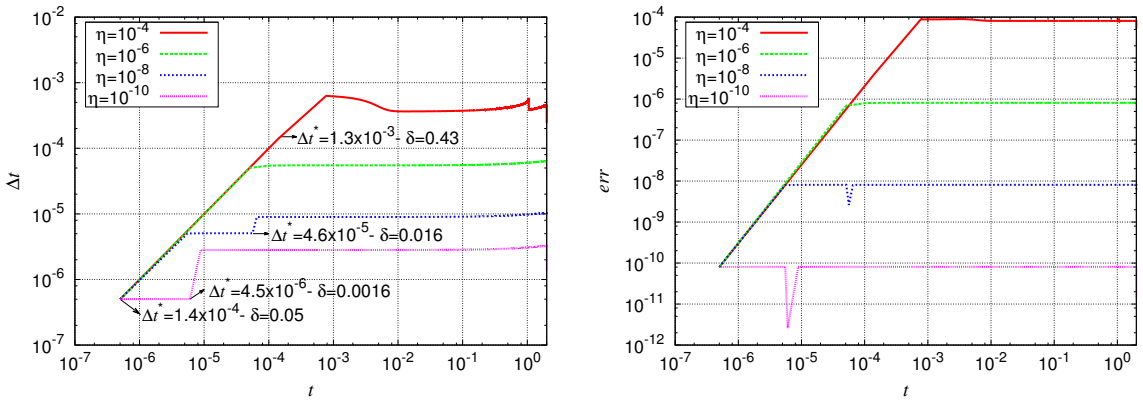

Fig. 5.6: *1D BZ equation. Time evolution of accepted splitting time steps Δt (left) and local L^2 error estimates $err = \|S^{\Delta t}u_0 - S_\delta^{\Delta t}u_0\|_{L^2}$ (right), for several tolerances η, considering critical $\Delta t^\star$ and computation of δ.*

Table 5.3: *1D BZ equation. L^2 errors at final time $t = 2$ for (a, b, c) variables and several tolerances η, considering critical $\Delta t^\star$ and computation of δ.*

η	L^2 error a	L^2 error b	L^2 error c
10^{-4}	6.85×10^{-5}	9.04×10^{-5}	4.06×10^{-5}
10^{-6}	1.71×10^{-6}	1.83×10^{-6}	7.98×10^{-7}
10^{-8}	4.53×10^{-8}	4.84×10^{-8}	2.12×10^{-8}
10^{-10}	4.48×10^{-9}	4.77×10^{-9}	2.15×10^{-9}

There are different scenarios depending on the required accuracy:

- In all cases for $\delta^0 = 0.05$, we initially estimate $\Delta t^\star \approx 1.4 \times 10^{-4}$.
- For $\eta = 10^{-4}$, the initial $\Delta t^\star$ implies smaller time steps than what is required for the prescribed tolerance. Hence, Δt increases until $\Delta t^{\text{new}} > \Delta t^\star$, and a new δ is estimated: $\delta \approx 0.43$. No substantial changes are made when ite $= N_\delta$, since $\Delta t \in [\beta \Delta t^\star, \gamma \Delta t^\star]$ for the current η.
- For $\eta = 10^{-6}$, we keep initial $\Delta t^\star$ and δ^0 since $\Delta t \in [\beta \Delta t^\star, \gamma \Delta t^\star]$, as seen in Figure 5.3.

- Finally, for $\eta = 10^{-8}$ and $\eta = 10^{-10}$, we have that $\Delta t < \beta \Delta t^{\star}$ and thus δ is recomputed, giving, respectively, $\delta \approx 0.016$ and 0.0016. In particular, we consider larger splitting time steps for which Strang local errors are better predicted.

Table 5.3 shows that error control is this time guaranteed for all values of tolerance η, and thus for a larger range of time steps. Compared with previous results in Table 5.2, we completely correct the errors in the prediction of local errors, which yields more accurate resolutions for the largest tolerances; whereas slightly less accurate results are obtained for the smallest tolerances since larger splitting time steps are considered.

5.7 Numerical Simulations: 2D BZ Equation Revisited

In the past sections of the present chapter, we have always considered sufficiently fine spatial discretizations in order to perform an evaluation of the theoretical estimates introduced for the proposed adaptive time integration scheme. For higher dimensional problems, fine spatial discretization becomes a critical issue in terms of computational costs and as previously discussed, a technique of local grid refinement becomes a good solution to overcome the latter difficulty and to guarantee the theoretical behavior of the splitting schemes in terms of spatial representation. The splitting error estimate is dynamically evaluated in practice in a semi-discretized configuration and hence retains only the splitting errors. The coupling of this adaptive technique with the MR/splitting strategy introduced in Chapter 4 becomes natural, under the same criteria in terms of construction of the solution scheme and accuracy of the numerical simulations. Moreover, the numerical implementation is straightforward. In this part, we recast the numerical simulations considered in Chapter 4 for the 2D BZ model (4.29), with the recently introduced adaptive splitting scheme. One of the objectives of this illustration is to justify and validate the chosen constant splitting time step in the previous simulations, and thus the proposed numerical strategy for stiff propagating waves. Moreover, the present study will yield some concluding remarks that will be used in further implementations of the combined time/space adaptive scheme in the forthcoming chapters.

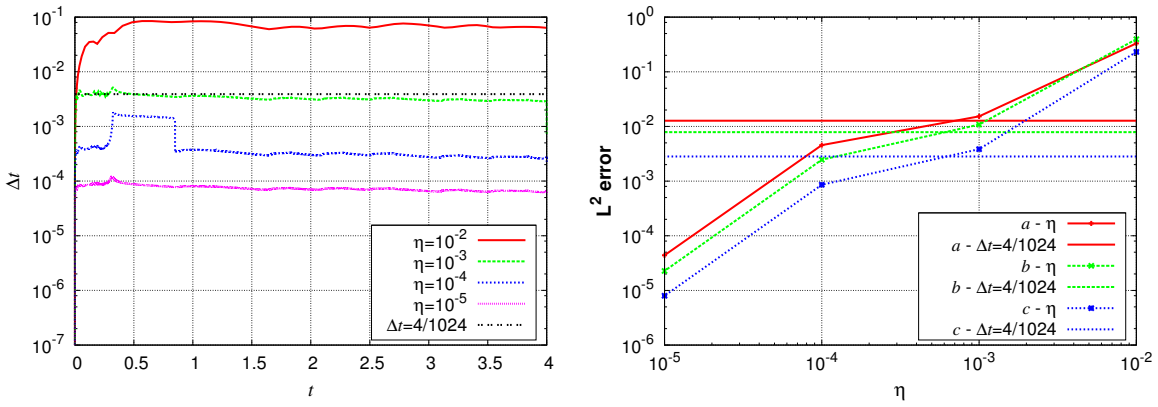

Fig. 5.7: *2D BZ spiral waves. Time evolution of splitting time steps (left) and normalized L^2 errors at $t^* = 4$ (right) according to (4.24), given by the time adaptive strategy for different accuracy tolerances η into (5.11) and with constant $\Delta t = 4/1024$. Uniform grid of 256^2.*

First, we consider the time adaptive scheme on a uniform grid of 256^2. Figure 5.7 shows the adaptive splitting time steps corresponding to different accuracy tolerances η according to (5.11). In all cases, the splitting time step is adapted from a chosen initial value of $\Delta t^0 = 10^{-7}$

to a roughly constant value that depends on the prescribed accuracy, due to the self-similar character of the wave like in the 1D case. The global time integration error is indeed controlled by the local error accuracy tolerance η into (5.11), as shown by the normalized L^2 errors at $t^* = 4$, according to (4.24). We see thus that for this kind of propagating phenomenon, a constant splitting time step computed based on an accuracy criterion as detailed in Chapter 4, is appropriate to describe the corresponding multi-scale features. Furthermore, a splitting time step of $\Delta t = 4/1024$ yields practically the same results as the adaptive splitting strategy with $\eta = 10^{-3}$. The overhead of estimating the local errors of the adaptive scheme can be thus saved, even though this overhead implies no more than 25 % of additional CPU time considering the embedded procedure. In general and for this kind of propagating phenomenon, the adaptive scheme can be used to initially compute the corresponding constant splitting time step for a given accuracy and therefore, preliminary computations are no longer necessary.

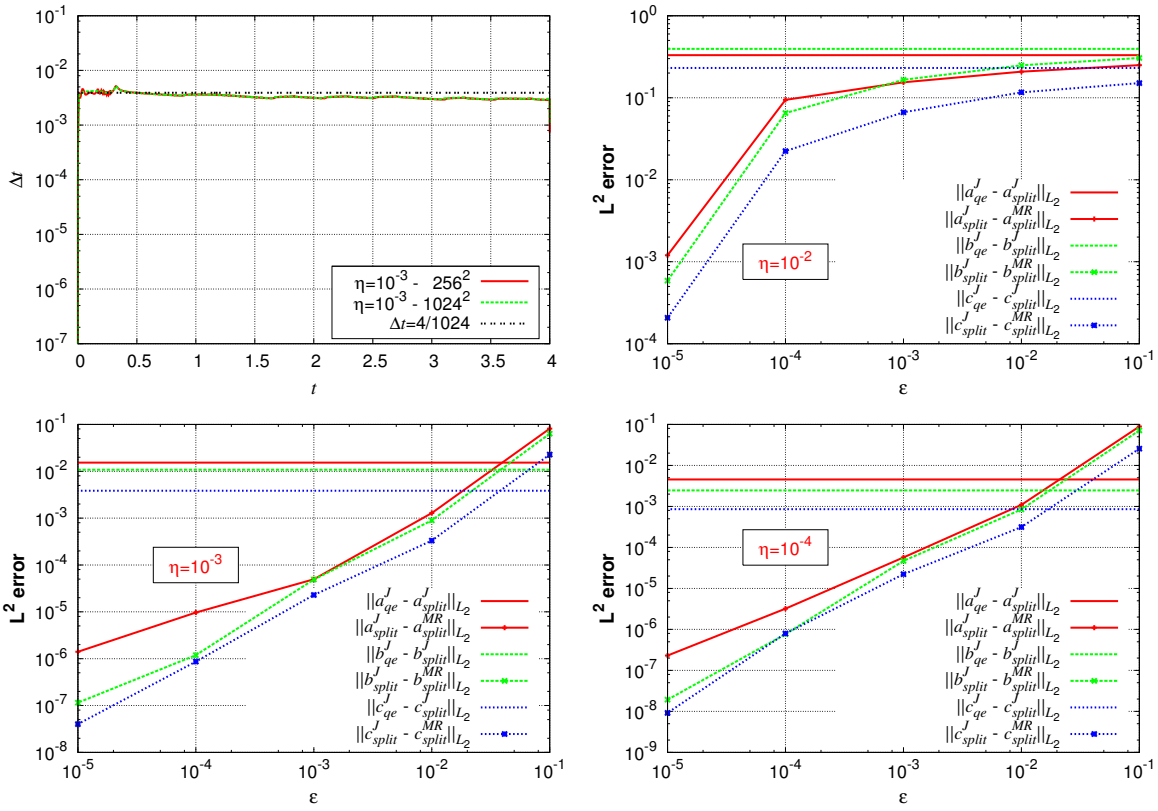

Fig. 5.8: *2D BZ spiral waves. Top left: time evolution of splitting time steps with* $\eta = 10^{-3}$ *and finest grids of* 256^2 *and* 1024^2*, and with* $\Delta t = 4/1024$*. From top right to bottom left: normalized* L^2 *errors at* $t^* = 4$ *given by the adaptive splitting technique with* $\eta = 10^{-2}$ *(top right),* 10^{-3} *(bottom left), and* 10^{-4} *(bottom right) on a uniform grid according to (4.24), and by the MR procedure according to (4.26), for several threshold values* ε *and the adaptive splitting scheme. Finest grid:* 256^2*.*

Considering a larger computational domain of 1024^2, Figure 5.8 (top left) shows the adaptive splitting time steps obtained with an accuracy tolerance of $\eta = 10^{-3}$. Once again this is shown to be almost equivalent to the constant splitting time step $\Delta t = 4/1024$, as in the previous 256^2 spatial discretization case, where these splitting errors are measured with respect to the corresponding coupled solution of the same semi-discretized problem. In this way, the splitting errors are practically independent of the spatial discretization of the problem for a sufficiently fine spatial representation.

We consider now the time/space adaptive technique by introducing the adaptive splitting scheme into the MR/splitting strategy. Figure 5.8 (from top right to bottom left) shows the corresponding time and space approximation errors given by (4.40), where $\mathbf{U}^J_{split}$ represents now the time adaptive splitting solution for some accuracy tolerance η into (5.11). The finest grid of the multiresolution representation corresponds to a 256^2 mesh discretization. We see in the representations of Figure 5.8 that:

- For $\eta = 10^{-2}$, we have time steps of about 7×10^{-2} as seen in Figure 5.7, *i.e.*, approximatively 7 times the bound $\Delta t_{\max}$ given by (4.27) in order to properly couple the splitting integration scheme with the multiresolution representation. This case is an illustrating counter example of a suitable coupling of the numerical methods, and highlights the importance of satisfying condition (4.27) for the maximum splitting time step $\Delta t_{\max}$.
- For $\eta \leq 10^{-3}$, we retrieve the same numerical behavior shown in Figure 4.6, which corresponds to the case $\eta = 10^{-3}$ in terms of accuracy.
- The time steps in Figure 5.7 show clearly that the splitting scheme is still working properly in the loss order region. In particular, for $\eta = 10^{-4}$ a threshold value given by $\varepsilon \leq 10^{-2}$ guarantees negligible spatial representation errors in front of the time integration ones; whereas for $\eta = 10^{-5}$, $\varepsilon \leq 10^{-4}$ is necessary.

To conclude, the time integration accuracy tolerance η describes the global accuracy of the numerical methods as long as $\eta \leq 10^{-3}$, and hence, $\Delta t < \Delta t_{\max}$ for the chosen refinement criterion of $\bar{k} = 1$ into (4.27). Then, $\varepsilon \leq 10^{-2}$ must be verified for $\eta = 10^{-3} - 10^{-4}$, whereas $\varepsilon \leq 10^{-4}$ for $\eta = 10^{-5}$.

5.8 Concluding Remarks

In the first part of this chapter we have introduced an adaptive splitting technique for the numerical solution of stiff reaction-diffusion problems. To the best of our knowledge, such a technique was never presented in the literature before and constitutes one of the major contributions of the present work. The mathematical analysis and description of the new scheme were also conducted which have allowed us to build the numerical method on a solid theoretical background. A complementary numerical procedure was also developed to ensure the validity of the local time integration error estimates *err*, for any arbitrary splitting time step Δt, by appropriately computing the shifting parameter δ in the embedded lower order Strang scheme, as well as the domain of application of the adaptive method defined by the critical splitting time step $\Delta t^\star$. In particular, a novel and general numerical procedure was introduced to estimate the truncated leading term of the local error expansion, which can be applied to any time integration scheme. The following remarks can be thus made related to the resulting numerical scheme:

- Since both parameters δ and $\Delta t^\star$ are linked through the expression (5.48), a sufficiently large value of δ might be sufficient in practical implementations, taking into account that the overhead of computing the lower order solution remains the same in (5.8).
- In principle, the latter overhead is given by an extra half step integration of the reaction problem, and one more step for the diffusion. Nevertheless, we will see in Chapter 8 how this overhead may be reduced in the practical implementation.
- The local error estimates *err* account only for the splitting errors since both the Strang and the shifted Strang schemes are applied to the same semi-discretized problem.

As previously envisioned, such a time adaptive scheme is then used to extend the domain of application of the numerical strategy introduced in Chapter 4, with exactly the same criteria, to more complex time dependent phenomena. Furthermore, the adaptive time stepping implies also a dynamic update of the local refinement ruled by the dynamics of the numerical solution. In this time/space adaptive technique, the accuracy of the numerical method is controlled by the parameters η and ε, for the time integration and the spatial representation errors. In order to better guarantee a decoupling of time- and space-related numerical errors and thus an overall accuracy dominated rather by the time integration procedure, a necessary but not sufficient appropriate choice will be in general given by

$$\varepsilon < \eta. \tag{5.62}$$

Notice that the configuration $\varepsilon = \eta$ might also yield good results taking into account the conservative refinement criteria adopted in Chapter 4 for the multiresolution process, and depending on the particular problem, even $\varepsilon \approx 10 \cdot \eta$, as shown in Figure 4.6 for the BZ problem (4.29). For all cases the accuracy estimates are only valid if the time evolution and the spatial representation of the solution are correctly coupled, as discussed in Chapter 4 and illustrated in Figure 4.6 for $\eta = 10^{-2}$. Therefore, criterion (5.62) becomes in general sufficient only if the dynamic splitting time steps satisfy

$$\Delta t \leq \Delta t_{\max}. \tag{5.63}$$

From a practical point of view and in order to avoid preliminary computations to define the maximum splitting time step $\Delta t_{\max}$ given by (4.27), a good level of coupling of the time and space resolutions is accomplished if for a given time integration accuracy tolerance η, numerical simulations with different threshold values ε behave qualitatively in a similar way. Since the main reactive activity is usually concentrated on the moving fronts, a deficient spatial representation during the time evolution step introduces severe perturbations in the numerical solution which are easily identified. Furthermore, a more conservative refinement criterion expressed by $\bar{k} > 1$ extra cells may be considered if larger splitting time steps are desired, or to better ensure the decoupling of space and time numerical errors. Alternatively, more sophisticated refinement criteria (like in [CKMP03]) may be required for highly complex configurations.

For the numerical simulation of phenomena for which a constant splitting time step Δt is sufficient, a hybrid strategy that considers a time step Δt computed within a prescribed accuracy with the current adaptive technique, might be the most convenient solution. The overhead of computing the local error estimates *err* throughout the entire time domain of simulation is thus avoided. Finally, considering supplementary operators like a convective term in the original reaction-diffusion system (5.1) does not change in general the order estimates conducted in this chapter for both the standard and shifted Strang schemes. The same follows for more general diffusion terms like the one in (1.52), for which the theoretical computations just become more technical.

Chapter 6

Parareal Operator Splitting for Multi-Scale Reaction Waves

In this chapter, we analyze a time parallelization technique for the solution of PDEs modeling stiff propagating waves, in order to improve the numerical performance of the simulations. As described in Chapter 1, and numerically analyzed in the previous chapters 4 and 5, the mathematical description of such problems imposes special restrictions on the numerical methods required to perform the numerical integration in time. In this context, the time operator splitting technique introduced in Chapter 4 has been coupled with the well known parareal algorithm initially proposed by Lions *et al.* in [LMT01]. A numerical analysis of the combined strategy is then mandatory to describe the mathematical behavior of such a method, and better explore its capabilities in the context of stiff problems. These issues motivated the following study which was recently published in ESAIM: Mathematical Modelling and Numerical Analysis journal [DDM11]. For the sake of completeness of the present work we include the entire article in what follows as a self-contained part.

PARAREAL OPERATOR SPLITTING TECHNIQUES FOR MULTI-SCALE REACTION WAVES: NUMERICAL ANALYSIS AND STRATEGIES

Max Duarte, Marc Massot, and Stéphane Descombes

Abstract. In this paper, we investigate the coupling between operator splitting techniques and a time parallelization scheme, the parareal algorithm, as a numerical strategy for the 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, spatially very localized. In a series of previous studies, the numerical analysis of the operator splitting as well as the parareal algorithm has been conducted and such approaches have shown a great potential in the framework of reaction-diffusion and convection-diffusion-reaction systems. However, complementary studies are needed for a more complete characterization of such techniques for these stiff configurations. Therefore, we conduct in this work a precise numerical analysis that considers the combination of time operator splitting and the parareal algorithm in the context of stiff reaction fronts. The impact of the stiffness featured by these fronts on the convergence of the method is thus quantified, and allows to conclude on an optimal strategy for the resolution of such problems. We finally perform some numerical simulations in the field of nonlinear chemical dynamics that validate the theoretical estimates and examine the performance of such strategies in the context of academical one-dimensional test cases as well as multi-dimensional configurations simulated on parallel architecture.

Keywords. *Parareal algorithm; Operator splitting; Convergence analysis; Reaction-diffusion; Multi-scale waves.*

Mathematics Subject Classification. 65Y05, 65M12, 65L04, 35A35, 35K57, 35C07.

ESAIM: M2AN 45 (2011) 825–852
DOI: 10.1051/m2an/2010104
Received May 6, 2010.
Published online February 23, 2011.

6.1 Introduction

Numerical simulations of multi-scale phenomena are commonly requested for modeling purposes in many applications such as combustion [Gio99, DB00, NWK98, KNW99, NK05], chemical vapor deposition [Gok88], or air pollution modeling [MGS82, Sun96, KC97]. The important development of the numerical strategies in these and in other fields such as nonlinear chemical dynamics for excitable media [Bar91, DMB97, DMD+12] or biomedical engineering [GDD+08b, DD08a, DDD+12] is mainly due to the constant increase of the computer power (see for instance [Ech09] for a recent review of methods applied to turbulent combustion). In general, all 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. Therefore, there are several numerical strategies in order to treat the induced stiffness for time dependent problems. In this particular 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.

The most natural idea to overcome these difficulties is to use dedicated numerical methods and to solve the complete models where diffusion, reaction and eventually convection are coupled together. In this context, we aim at solving strongly coupled nonlinear systems with either a fully implicit method, using eventually modified Newton methods for ill-conditioned problems [Deu74, Smo83, Deu04], or yet semi-implicit or linearized implicit methods instead (see [D'A94] and references therein). However, the strong stability restrictions for the latter when dealing with very fast temporal scales [D'A94, DL95b] as well as the computational cost and the huge memory requirements of these methods, even if adaptive grids are used, make these strategies difficult to handle. Nevertheless, these kind of coupled resolutions are useful when we need reference solutions for validation and study purposes but necessarily restricted to low dimensional configurations with not too many unknowns.
An alternative numerical strategy first introduced in [ACM98] is then to combine implicit and explicit schemes to discretize nonlinear evolution problems in time. Further studies into [SSV97, VS04] settled the appropriate numerical background for these methods called IMEX, which in particular might be conceived to solve stiff nonlinear problems as presented in [VSH04, SSV06]. These methods are usually very efficient (see for instance [NWK98] for a combustion application). Nevertheless, on the one hand, the feasibility of utilizing dedicated implicit solvers over a discretized domain become soon critical when treating large computational domains. And on the other hand, the time steps globally imposed over partial regions or the entire domain are strongly limited by either the stability restrictions of the explicit solver or by the fastest scales treated by the implicit scheme.
Nevertheless, in many multi-scale problems as for example the propagation of reaction waves, the fastest time scales do not play a leading role in the global physical phenomenon and thus, we might consider the possibility of using reduced models where these chemical scales have been previously relaxed [HW96]. 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. In particular, the derivation of the reduced model is usually accessible when the system is well-partitioned and the fast scales have been isolated [Spo99, SD00]. In this case, a rigorous singular perturbation analysis can be conducted even in the context of nonlinear source terms for numerical analysis purposes [Mas02, DM04]. 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 [Mar90] also called fractional step methods [Tém69a, Tém69b, Yan71] 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 [Des01, Sch02]. In practice, it is firstly necessary to decouple numerically the reaction part from the rest of the physical phenomena like convection, diffusion or both, for which there also exist dedicated numerical methods. Hence, operator splitting techniques allow a completely independent optimization of the resolution of each subsystem which normally yields lower requirements of computational resources.
In the context of multi-scales waves, the dedicated methods chosen for each subsystem are then responsible for dealing with the fast scales associated to each one of them, in a separate manner; then, the composition of the global solution based on the splitting scheme should

guarantee the good description of the global physical coupling; therefore, in order to verify this fundamental constraint, a rigorous numerical analysis is required. In fact, several works [D'A94, HW96, VS98, Spo00] proved that the classical numerical analysis of splitting schemes fails in presence of scales much faster than the splitting time step and motivated more rigorous studies for these stiff configurations [DM04]. In this way, complementary works described also the numerical behavior of these methods when spatial multi-scale phenomena arise mainly as a consequence of large spatial gradients [DDLM07], so that the influence of both spatial and time related stiffness has been and continues to be analyzed in detail for not arbitrarily small splitting time steps [DDL+12].

Thus, with the choice of the resolution technique properly justified, we investigate the coupling of operator splitting with a time parallelization scheme, pursuing even better performances in time consumption for multi-scale simulations on parallel architectures. In this context, many algorithms already proposed the solution of evolution problems in a time-parallel fashion (see [GV07] and the references therein for a historical review). However, the parareal algorithm, first presented in [LMT01], has received a lot of attention over the past few years in different applications in different domains [BBM+02, FC03, FHM03, GEF03, BM03], as a promising efficient numerical method to solve evolution problems in parallel. The general principle of the parareal algorithm combines a coarse and fast solver which is run sequentially, and a more accurate and expensive fine solver that should be run in parallel. In this way, we take advantage of parallel computations through an iterative process which yields convergence from a coarse initial approximation to the detailed dynamics of the system given by the accurate resolution of the fine solver. Several variants of the method have been also proposed (for example in [FC03, GLFE06]).

Up to these days, many theoretical analysis of the parareal algorithm have been conducted (see for instance [LMT01, MT02, Bal03, MT03, SR03, GV07, GH08]), which led to various estimates of convergence rates and descriptions of the stability behavior of the method in applications involving general linear and nonlinear systems of ODEs or PDEs. Nevertheless, to our knowledge, none of the studies previously conducted either in a linear or in a nonlinear framework, took explicitly into account stiff phenomena. As a result, there is the need of complementary studies to utterly predict the performance of the algorithm in such configurations in order to propose more efficient time parallelization schemes in multi-scale contexts.

Keeping this in mind, the present work conducts a detailed numerical analysis of parareal operator splitting techniques in the context of multi-scale reaction waves. In this way, new representation of the splitting local errors are deduced and a convergence analysis of the parareal operator splitting algorithm is performed for general linear reaction-diffusion systems. These results are then extended to the case of reaction waves through the associated linearized system of the original reaction-diffusion system modeling the multi-scale phenomenon, where stiffness is introduced by the presence of large spatial gradients. The negative influence of stiffness on the numerical behavior of parareal operator splitting schemes is then mathematically proven.

With this theoretical characterization achieved, we are able to build a new and simple numerical strategy for multi-scale phenomena, with the parareal algorithm and the operator splitting as fine solver, each one of them based on their corresponding theoretical background. The time operator splitting strategy adopted then considers on the one hand, a high order method like Radau5 [HW96], based on implicit Runge-Kutta schemes for stiff ODEs, that 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 [Abd02], based on explicit stabilized Runge-Kutta schemes, that solves the diffusion problem. The potential of this splitting configuration has already been tested and evaluated in previous studies [DDD+12, DMD+12]. The numerical simulations then performed show the influence of a stiff configuration on the

parareal performance, validating the theoretical estimates previously conducted. Finally, the discussions inspired by these numerical results gives some insights into alternative numerical strategies for this kind of difficult problems.
The paper is organized as follows: in a first part, we formally present the parareal algorithm as well as the splitting techniques which will be used in the proposed numerical strategy. The complete numerical analysis of the parareal operator splitting is conducted in the second part: a brief review on the theoretical results found in the literature is first presented, then the different local error estimates are carefully conducted to finish with the theoretical convergence analysis in a general linear reaction-diffusion configuration and in the case of multi-scale reaction waves through a linearized model of such configurations. In the last part, we first conduct a series of detailed and careful numerical simulations in a one-dimensional case in order to validate the previous theoretical results. Finally, the potential of the method is illustrated in the framework of a two-dimensional simulation which allows a detailed discussion of the capability and performance of the method. All the models belong to the field of nonlinear chemical dynamics.

6.2 Parareal Algorithm

6.2.1 Temporal Parallelization

We first consider a general method in order to compute the numerical solution of a system of ordinary differential equations (ODEs) of the form:

$$\mathbf{u}'(t) = \mathbf{f}\left(\mathbf{u}(t)\right), \quad t \in [0,T], \quad \mathbf{u}(0) = \mathbf{u}^0, \tag{6.1}$$

where for some integer M, $\mathbf{f} : \mathbb{R}^M \to \mathbb{R}^M$ and $\mathbf{u} : \mathbb{R} \to \mathbb{R}^M$. In order to achieve a time parallelization algorithm for (6.1), we decompose the time domain $I = [0,T]$ into N subdomains $I_n = [T_n, T_{n+1}[$, $n = 0, \ldots, N-1$, with $0 = T_0 < \ldots < T_{N-1} < T_N = T$, and $\Delta T_n := T_{n+1} - T_n$, so, that we consider for each time subdomain the evolution problem:

$$\mathbf{u}_n'(t) = \mathbf{f}\left(\mathbf{u}_n(t)\right), \quad t \in I_n, \quad \mathbf{u}_n(T_n) = \mathbf{U}_n, \tag{6.2}$$

where the initial values $\mathbf{U}_n$ are a necessary input in order to solve each of these evolution problems. The solutions $\mathbf{u}_n$ on the subdomains I_n in (6.2) should be consistent with $\mathbf{u}$ over I obtained out of (6.1), this means that the initial values $\mathbf{U}_n$ are intended to satisfy the system:

$$\mathbf{U}_0 = \mathbf{u}^0, \quad \mathbf{U}_n = \phi^{\Delta T_{n-1}}(\mathbf{U}_{n-1}), \quad n = 1, \ldots, N, \tag{6.3}$$

where the flow $\phi^{\Delta T_n}(\mathbf{U})$ denotes the solution of (6.1) with initial condition $\mathbf{U}$ after time ΔT_n. Thus, we consider N independent evolution problems given by (6.2) for each time subdomain I_n, so that each of them may be computed by a different processor in a parallel environment. Nevertheless, this can only be achieved if the initial conditions $\mathbf{U}_n$ of (6.2) are previously known or at least approximated, in order to have a proper Cauchy problem on each time subdomain. Therefore, any time parallelization algorithm will aim at approximating the initial $\mathbf{U}_n$ by the set $\tilde{\mathbf{U}}_n$ and then, at solving (6.2), even though condition (6.3) with $\tilde{\mathbf{U}}_n$ is not initially verified. In this context, the parareal algorithm gives us a way to perform this kind of parallelization in an efficient way.
This technique extends naturally to the numerical resolution of partial differential equations (PDEs), as it was initially proposed in [LMT01]. In fact, the multi-scale reaction waves that we consider in this study are modeled by means of parabolic PDEs.

6.2.2 Principle of the Parareal Algorithm

Considering system (6.2), the parareal algorithm is based on two propagation operators: $\mathcal{G}^{\Delta T_n}(\mathbf{U})$ and $\mathcal{F}^{\Delta T_n}(\mathbf{U})$, that provide respectively a coarse and an accurate approximation of $\phi^{\Delta T_n}(\mathbf{U})$. In this way, the algorithm starts with an initial approximation $\tilde{\mathbf{U}}_n^0$ given for example by the sequential computation:

$$\tilde{\mathbf{U}}_0^0 = \mathbf{u}^0, \quad \tilde{\mathbf{U}}_n^0 = \mathcal{G}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^0), \quad n = 1, \ldots, N, \tag{6.4}$$

and then performs for $i = 1, \ldots, i_{conv}$ the correction iterations:

$$\left.\begin{aligned} &\tilde{\mathbf{U}}_0^i = \mathbf{u}^0, \\ &\tilde{\mathbf{U}}_n^i = \mathcal{F}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^{i-1}) + \mathcal{G}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^i) - \mathcal{G}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^{i-1}), \quad n = 1, \ldots, N. \end{aligned}\right\} \tag{6.5}$$

Notice that i iterations imply i time subdomains integrated by $\mathcal{F}$, since $\tilde{\mathbf{U}}_n^i = \mathcal{F}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^{i-1})$ for $n \leq i$ and $\tilde{\mathbf{U}}_0^i = \mathbf{u}^0$ with $\mathbf{u}^0$ taken from the original problem (6.1). Then, for $i = N$, where N is the number of subdomains, the parareal algorithm (6.5) will generate a set of values $\tilde{\mathbf{U}}_n$ that satisfy $\tilde{\mathbf{U}}_n = \mathcal{F}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1})$. That is, the approximations at the time-points T_n will have achieved the accuracy of the propagator $\mathcal{F}$. Nevertheless, the main idea of the algorithm is to choose propagators $\mathcal{F}$ and $\mathcal{G}$ in order to achieve this level of accuracy without performing the N accurate resolutions. Therefore, after convergence of the algorithm (6.5) for $i = i_{conv}$, we shall obtain a solution $\tilde{\mathbf{U}}_n^{i_{conv}}$ with a fine accuracy respect to $\mathbf{U}_n$ in (6.3), for which only $i_{conv} \ll N$ subdomains would have been integrated by propagator $\mathcal{F}$. Thus from an initial coarse approximation (6.4), we might achieve an accurate resolution of problem (6.1) with important savings of computational time.

6.2.3 Interpretation of the Parareal Algorithm

The parareal algorithm can most naturally be interpreted as a classical deferred correction method in which the initial values $\mathbf{U}_n$ of (6.2) are corrected through a feedback mechanism based on both propagators $\mathcal{F}$ and $\mathcal{G}$, as it was presented in [BBM+02]. Therefore, for a initial condition given by (6.4), we define the correction

$$\mathcal{E}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}) = \mathcal{F}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}) - \mathcal{G}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}) \tag{6.6}$$

in order to obtain the new initial conditions for $n = 1, \ldots, N-1$ and the new value at $n = N$:

$$\tilde{\mathbf{U}}_n^1 = \mathcal{G}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^1) + \mathcal{E}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^0). \tag{6.7}$$

Thus, a recursive application leads us to

$$\tilde{\mathbf{U}}_0^i = \mathbf{u}^0, \quad \tilde{\mathbf{U}}_n^i = \mathcal{G}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^i) + \mathcal{E}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^{i-1}), \quad n = 1, \ldots, N, \tag{6.8}$$

which gives us the same scheme as (6.5).
Nevertheless, based on [CP93], the time decomposition method (6.2) can be also interpreted as a multiple shooting method for (6.1). In fact, considering $\mathbf{U} = (\mathbf{U}_0, \ldots, \mathbf{U}_N)^T$ as the unknowns, the system (6.3) can be written as

$$\mathbf{F}(\mathbf{U}) = \begin{pmatrix} \mathbf{U}_0 - \mathbf{u}^0 \\ \mathbf{U}_1 - \phi^{\Delta T_0}(\mathbf{U}_0) \\ \vdots \\ \mathbf{U}_N - \phi^{\Delta T_{N-1}}(\mathbf{U}_{N-1}) \end{pmatrix} = \mathbf{0}, \tag{6.9}$$

where $\mathbf{F} : \mathbb{R}^{M\cdot N+1} \to \mathbb{R}^{M\cdot N+1}$. In this way, we search the initial values $\mathbf{U}_n$ of (6.2) that correspond to the solution of original problem (6.1) through condition (6.3).
Solving this system with Newton's method, leads after a short calculation to

$$\left.\begin{aligned} &\tilde{\mathbf{U}}_0^i = \mathbf{u}^0, \\ &\tilde{\mathbf{U}}_n^i = \phi^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^{i-1}) + \frac{\partial \phi^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^{i-1})}{\partial \tilde{\mathbf{U}}_{n-1}^{i-1}} \left(\tilde{\mathbf{U}}_{n-1}^i - \tilde{\mathbf{U}}_{n-1}^{i-1}\right), \quad n = 1, \ldots, N. \end{aligned}\right\} \tag{6.10}$$

With this formulation, there are many ways to apply the multiple shooting algorithm to solve (6.10) numerically, as it is detailed in [GV07], and when the approximations are close enough to the solution, the convergence is guaranteed as shown in [CP93]. However, if we approximate the time subdomain integration in (6.10) by

$$\phi^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^{i-1}) \approx \mathcal{F}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^{i-1}), \tag{6.11}$$

and the Jacobian term by

$$\frac{\partial \phi^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^{i-1})}{\partial \tilde{\mathbf{U}}_{n-1}^{i-1}} \left(\tilde{\mathbf{U}}_{n-1}^i - \tilde{\mathbf{U}}_{n-1}^{i-1}\right) \approx \mathcal{G}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^i) - \mathcal{G}^{\Delta T_{n-1}}(\tilde{\mathbf{U}}_{n-1}^{i-1}), \tag{6.12}$$

then the multiple shooting method (6.10) and the parareal algorithm (6.5) coincide (see [GV07]).
The parareal algorithm can also be seen as a time-multigrid method as it was entirely proven in [GV07]. However, because of its clarity and simplicity, in this paper we will adopt the multiple shooting point of view, *i.e.,* a Newton's method approximation, even if the numerical analysis we will perform is practically independent of this choice. In fact, from a practical point of view, all the interpretations are equivalent and the algorithm is mainly implemented as a deferred correction method.

6.2.4 Considerations on the Propagation Operators

The parareal algorithm relies on two solvers, the fine and coarse propagators, $\mathcal{F}$ and $\mathcal{G}$. On the one hand, the fine solver produces an accurate approximation of the solution and its choice depends on the desired level of accuracy and on the nature of the problem (multi-scale phenomena, stiffness, large systems). In our particular case of multi-scale reaction waves, the time operator splitting reveals itself as a suitable resolution technique as it was previously discussed in the introduction. Its parallelization is then achieved via the parareal scheme. On the other hand, the coarse solver gives a coarser approximation that needs to be computed in a sequential way, therefore it should be as fast as possible in order to globally guarantee important savings in time consumption. The performance of the parareal algorithm then relies directly on the choice of this solver and an optimal balance must be found between its computational speed and its level of accuracy, *i.e.,* coarser approximations are faster but increase the number of iterations needed.
Let us now set the general mathematical framework in this work; in this context, we recall that a class of multi-scale phenomena can be modeled by general reaction-diffusion systems of type:

$$\left.\begin{aligned} &\partial_t \mathbf{u} - \partial_{\mathbf{x}} \left(\mathbf{D}(\mathbf{u}) \partial_{\mathbf{x}} \mathbf{u}\right) = \mathbf{f}\left(\mathbf{u}\right), && \mathbf{x} \in \mathbb{R}^d, \ t > 0, \\ &\mathbf{u}(0, \mathbf{x}) = \mathbf{u}_0(\mathbf{x}), && \mathbf{x} \in \mathbb{R}^d, \ t = 0, \end{aligned}\right\} \tag{6.13}$$

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

However, in this work we will only consider the simplified case of linear diagonal diffusion, in which case the elements of the diffusion matrix are written as $D_{i_1 i_2 i_3}(\mathbf{u}) = D_{i_3}\delta_{i_1 i_2}$, so that the diffusion operator reduces to the heat operator with some scalar diffusion coefficient D_{i_3} for component u_{i_3} of $\mathbf{u}$. Notice that as it was established in previous studies [DM04, DDLM07, DDL+12], on the one hand, a rigorous numerical analysis of this simpler class of reaction-diffusion systems is very useful in order to theoretically characterize the impact of stiffness on the numerical behavior of methods conceived to simulate these phenomena. In particular, the original multi-scale character is not withdrawn. And on the other hand, it shall give us some insights into more complex cases as well as complete convection-reaction-diffusion systems.
Finally, in order to simplify the presentation, we consider a one-dimensional model, taking into account that extension into higher dimensions of $\mathbf{x}$ or $\mathbf{u}$ is straightforward. From these considerations, we infer the following initial value problem:

$$\left.\begin{aligned} \partial_t u - D\,\partial_x^2 u &= kf(u), \quad x \in \mathbb{R},\ t > 0, \\ u(0,x) &= u_0(x), \qquad\quad x \in \mathbb{R},\ t = 0, \end{aligned}\right\} \tag{6.14}$$

where f and u_0 are smooth functions, with real coefficients D and k. Furthermore, we denote by $T^t(u_0)$ the semiflow associated to (6.14). In what follows, we shall present both the fine and the coarse solvers taken into consideration to solve problem (6.14) by means of the parareal algorithm (6.5).

Fine Solver: Time Operator Splitting

Let us first introduce the classical decoupling of the diffusion and reaction parts of (6.14). More precisely, we denote $X^t(u_0)$ the solution of the diffusion equation:

$$\partial_t u_D - D\,\partial_x^2 u_D = 0, \quad x \in \mathbb{R},\ t > 0, \tag{6.15}$$

with initial data $u_D(0,\cdot) = u_0(\cdot)$. We also denote by $Y^t(u_0)$ the solution of the reaction part where the spatial coordinate x can be considered as a parameter:

$$\partial_t u_R = kf(u_R), \quad x \in \mathbb{R}, t > 0, \tag{6.16}$$

with initial data $u_R(0,\cdot) = u_0(\cdot)$.
The two Lie approximation formulae of (6.14) are defined by

$$\mathcal{L}_1^t(u_0) = X^t Y^t(u_0), \quad \mathcal{L}_2^t(u_0) = Y^t X^t(u_0), \tag{6.17}$$

and the two Strang approximation formulae of (6.14) [Str63, Str68] are defined by

$$\mathcal{S}_1^t(u_0) = X^{t/2} Y^t X^{t/2}(u_0), \quad \mathcal{S}_2^t(u_0) = Y^{t/2} X^t Y^{t/2}(u_0). \tag{6.18}$$

It is well known that Lie formulae (6.17) (resp. Strang formulae (6.18)) are an approximation of order 1 (resp. 2) of the exact solution of (6.14). Nevertheless, these classical orders are no longer valid since we consider very stiff reactive or diffusive terms (see [DM04]). In fact, if the fastest time scales play a leading role in the global physics of the phenomenon, then the composed solution obtained by means of a splitting technique will surely fail to capture the final dynamics of the phenomenon, unless we consider splitting time steps of the same order of such scales.
In the opposite case where these fast scales are not directly related to the physical development of the phenomenon, larger splitting time steps might be considered, but order reductions may

then appear due to short-life transients associated to the fast variables. In particular, this is the case for propagating reaction waves. In this context, it has been proven in [DM04] that better performances are expected while ending the splitting scheme by the time integration of the reaction part (6.16) or in a more general case, the part involving the fastest time scales of the phenomenon (see the numerical application in [DDL+12]). In particular, in the case of linear diagonal diffusion problems, no order loss is expected for the $\mathcal{L}_2^t$ and $\mathcal{S}_2^t$ schemes when fast scales are present in the reactive term. Even more, as it was presented and analyzed in [DDLM07], the presence of high spatial gradients may also degrade the performance of these methods leading to order reductions coming from space multi-scale phenomena.
Keeping in mind these theoretical studies and considering the various numerical alternatives previously discussed, the time operator splitting remains as the most appropriate resolution scheme for general multi-scale problems and so far, the optimal choice for the fine solver. A complementary consideration is that suitable time integration methods in terms of order and stability must be chosen for each subsystem (6.15) and (6.16), in order to guarantee the accuracy of the estimates established by the corresponding numerical analysis. In fact, in all splitting order estimates, the solutions associated to these systems are supposed to be known exactly or with a sufficient accuracy (see for example [DS02a, HV03, DM04, HLW06]).

Coarse Approximation

When considering stiff problems of type (6.13) (or (6.14)), the choice of the coarse solver is not an easy task because we must look for fast and stable methods at the same time, considering that these computations will be performed in a sequential environment. In fact, depending on the stiffness of the system, we are almost constrained to choose more expensive but more stable methods (see [HW96] for more details on integration of stiff ODEs), otherwise, we would not be able to obtain coarser but still valid approximations.
Taking into account these requirements, several strategies might be considered. For instance, we can solve the coupled reaction-diffusion system (6.14) with a dedicated, stable but less accurate solver; which can be achieved by using larger integration time steps or a lower order method respect to the fine solver. Nevertheless, as previously discussed, the important computational requirements of these dedicated methods must be taken into account and thus, this alternative might not be always feasible.
Another technique could be the resolution of (6.14), discretized on a coarser spatial grid. Then, the crucial aspect would be the definition of proper inter-grid operators; that is, the operators allowing data exchange between the coarse and the fine grids. On the other hand, we might also consider the resolution of a reduced model of (6.14) instead, where the fastest scales have been relaxed; and thus, take this solution as the coarse approximation. Nevertheless, the previous knowledge of the fast scales is mandatory for straightforward implementations, and at the same time, we should also conceive an efficient reconstruction procedure of the fast variables not estimated by the coarse solver.
However, in the context of multi-scale reaction waves, a less accurate splitting operator might be a natural choice, conceived with larger splitting time steps or coarser time integrators of the split subsystems in order to accelerate computations. In particular, in this work we will be mainly focused on the detailed analysis of such splitting solvers as coarse propagators, considering their reputed pertinence on multi-scale problems. In this way, the numerical analysis presented in the next section describes the behavior of the splitting techniques as well as the parareal scheme itself, and shall lead us to further numerical studies and potential improvements of the numerical strategy.

6.3 Numerical Analysis of the Parareal Operator Splitting

In this section, we will first present important results from the literature that explain the behavior of the parareal algorithm and detail its convergence when applied to, first, linear and then, nonlinear problems. Those particularly relevant for our study will be described in details. Complementary information can be found in the indicated references. Then, a comprehensive numerical analysis of the parareal Lie (Strang) operator splitting will be presented in a linear framework that mimics the eventual influence of high spatial gradients for reaction waves. Throughout all this section, we will maintain the notation established in the previous one.

6.3.1 Review of the Literature and State of the Art

The Linear Case

We first consider a scalar linear problem of the form:

$$u'(t) = au, \quad t \in [0,T], \quad u(0) = u^0 \text{ with } a \in \mathbb{C}, \tag{6.19}$$

the following proposition is taken from the first publication on the parareal algorithm [LMT01].

Proposition 6.1. *Let $\Delta T = T/N$, $T_n = n\Delta T$ for $n = 0,\dots,N$. Consider (6.19) with $a \in \mathbb{R}$. Let $\mathcal{F}^{\Delta T}(U_n)$ be the exact solution at T_{n+1} of (6.19) with $u(T_n) = U_n$, and let $\mathcal{G}^{\Delta T}(U_n)$ be the corresponding backward Euler approximation with time step ΔT. Then,*

$$\max_{1\le n\le N} |u(T_n) - \tilde{U}_n^i| \le P_i\, \Delta T^{i+1}. \tag{6.20}$$

Thus, the algorithm converges and behaves in ΔT like a method of order $i+1$. Nevertheless, we must take into account that this result is only valid for a fixed iteration step i, since the constant P_i in (6.20) grows with i in the estimate of the proof in [LMT01], *i.e.*, the convergence is verified only for ΔT sufficiently small. Result (6.20) has been extended to more general and, in particular, higher order time-integration schemes. In fact, it was shown in [Bal03, BM03] that the parareal is a method of order $p(i+1)$ when a method of order p is used as the coarse propagator.
However, a very important work presented in [GV07] gives us complementary results that describe the behavior of the algorithm for any i and a fixed ΔT. As a matter of fact, the following corollary was stated and proven.

Corollary 6.2. *Let $T < \infty$,$\Delta T = T/N$, and $n = 0,\dots,N$. Consider (6.19) with $a \in \mathbb{C}$. Let $\mathcal{F}^{\Delta T}(U_n)$ be the exact solution at T_{n+1} of (6.19) with $u(T_n) = U_n$, and let $\mathcal{G}^{\Delta T}(U_n) = R(a\Delta T)U_n$ be a one-step method in its region of absolute stability. Then, we have the bound*

$$\max_{1\le n\le N} |u(T_n) - \tilde{U}_n^i| \le \frac{|\mathrm{e}^{a\Delta T} - R(a\Delta T)|^i}{i!} \prod_{j=1}^{i} (N-j) \max_{1\le n\le N} |u(T_n) - \tilde{U}_n^0|. \tag{6.21}$$

If the local truncation error of $\mathcal{G}$ is bounded by $C\Delta T^{p+1}$, with $p > 0$ and C a constant, then we have, for ΔT small enough,

$$\max_{1\le n\le N} |u(T_n) - \tilde{U}_n^i| \le \frac{(CT)^i}{i!} \Delta T^{pi} \max_{1\le n\le N} |u(T_n) - \tilde{U}_n^0|. \tag{6.22}$$

These results give us more precise information regarding constant P_i in (6.20). In fact, for finite T, the division by $i!$ in (6.21) and (6.22) shows that the algorithm converges superlinearly. Furthermore, result (6.22) presents the parareal as a method of order pi.

A linear convergence result was also demonstrated in [GV07] for an infinitely long time interval, *i.e.*, $T = \infty$, for problem (6.19). Moreover, similar results were obtained in [GV07] while investigating the performance of the parareal algorithm on PDEs, more precisely, a diffusion equation and an advection one. In fact, a Fourier transform in space converts these equations into linear systems of ODEs for each Fourier mode, and the convergence is demonstrated based on the previous results with some complementary hypothesis.

The Nonlinear Case

Taking into account the nonlinear system of ODEs (6.1), the superlinear convergence of the parareal algorithm has also been demonstrated in [GH08]. In fact, considering $\mathcal{F}$ as the exact solution of (6.1), it has been assumed that the difference between the approximate solution given by $\mathcal{G}$ and the exact solution can be expanded for ΔT small, in the following way:

$$\mathcal{F}^{\Delta T}(\mathbf{u}) - \mathcal{G}^{\Delta T}(\mathbf{u}) = c_{p+1}(\mathbf{u})\Delta T^{p+1} + c_{p+2}(\mathbf{u})\Delta T^{p+2} + \dots, \tag{6.23}$$

which is possible if the right hand side function $\mathbf{f}$ in (6.1) is smooth enough. Notice that this condition is no other than the series expansion of the local error of a numerical method $\mathcal{G}$ of order p (where $\mathcal{F}$ is the exact solution of (6.1)). Then, as a consequence of (6.23), we might assume that we have the following bound:

$$\|\mathcal{F}^{\Delta T}(\mathbf{u}) - \mathcal{G}^{\Delta T}(\mathbf{u}) - \left(\mathcal{F}^{\Delta T}(\mathbf{v}) - \mathcal{G}^{\Delta T}(\mathbf{v})\right)\| \leq C_1 \Delta T^{p+1} \|\mathbf{u} - \mathbf{v}\|, \tag{6.24}$$

as it has been supposed in [GH08].
Moreover, if $\mathcal{G}$ satisfies the Lipschitz condition:

$$\|\mathcal{G}^{\Delta T}(\mathbf{u}) - \mathcal{G}^{\Delta T}(\mathbf{v})\| \leq (1 + C_2 \Delta T)\|\mathbf{u} - \mathbf{v}\|, \tag{6.25}$$

the following theorem was stated and proven in [GH08].

Theorem 6.3. *Let $\mathcal{F}^{\Delta T}(\mathbf{U}_{n-1})$ be the exact solution on time subdomain I_{n-1}, and let $\mathcal{G}^{\Delta T}(\mathbf{U}_{n-1})$ be an approximate solution with local truncation error bounded by $C_3 \Delta T^{p+1}$, and satisfying (6.23), where the c_j, $j = p+1, p+2, \dots$ are continuously differentiable, and assume that $\mathcal{G}$ satisfies the Lipschitz condition (6.25). Then, at iteration i of the parareal algorithm (6.5), we have the bound*

$$\begin{aligned} \|\mathbf{u}(T_n) - \tilde{\mathbf{U}}_n^i\| &\leq \frac{C_3}{C_1} \frac{(C_1 \Delta T^{p+1})^{i+1}}{(i+1)!} (1 + C_2 \Delta T)^{n-i-1} \prod_{j=0}^{i} (n-j), \\ &\leq \frac{C_3}{C_1} \frac{(C_1 T_n)^{i+1}}{(i+1)!} \mathrm{e}^{C_2 (T_n - T_{i+1})} \Delta T^{p(i+1)}. \end{aligned} \tag{6.26}$$

Finally, we see that the parareal algorithm also converges superlinearly with respect to i and behaves as a method of order $p(i+1)$ for a nonlinear problem and a finite T. Hence, result (6.26) does not contradict the previous ones but extends them to the nonlinear case. Nevertheless, in a practical application, a more complete knowledge of the bound (6.26) might be necessary in order to fully describe the behavior of the algorithm. More precisely, constants C_1, C_2 and C_3 in (6.26) (as well as C into (6.22) for a linear application) should be properly estimated. This is one of the purposes of the present work in the case of an operator splitting strategy and the corresponding analysis will be presented in the next subsection.

6.3.2 Convergence Analysis of the Parareal Operator Splitting

In order to carry on our study, we will first present the mathematical problem that we want to solve. The link with multi-scale reaction waves will be then established.

Mathematical Framework

We consider the initial-value problem:

$$\left.\begin{array}{ll} \partial_t u - D\,\partial_x^2 u + kV(x)u = 0 & x \in \mathbb{R},\ t > 0, \\ u(x,0) = u_0(x) & x \in \mathbb{R},\ t = 0, \end{array}\right\} \tag{6.27}$$

where V is a bounded function from $\mathbb{R}$ to $\mathbb{R}$ of class $\mathcal{C}^\infty$ with all bounded derivatives and, D and k are some real positive coefficients such that $Dk = 1$. Moreover, u represents a dimensionless specie concentration.
For this study, we consider problem (6.27) as coming from the associated linearized system of (6.14) around u_0 after some simple change of variables. In this context, V is no other than the Jacobian $f'(u_0) = \partial f(u_0)/\partial u$ in (6.14). The diffusion and reaction coefficients, D and k in (6.14) and (6.27), might be seen as scale coefficients of time and space. In fact, in the context of reaction waves, we can obtain a dimensionless form of system (6.14) considering a dimensionless time τ and a dimensionless space r with

$$\tau = kt \quad \text{and} \quad r = (k/D)^{1/2}x. \tag{6.28}$$

See [HV03] for details on how reaction systems of ODEs can be built from a reaction scheme, and [GS94] for details on analysis of traveling reaction waves.
Then, without loss of generality, a dimensionless analysis of a traveling wave, as shown in [GS94], may allow us to establish the steady state of a reaction wave. Therefore, taking into account the scale coefficients, the dimensionless steady velocity of the wavefront given by the expression $\mathrm{d}r/\mathrm{d}\tau$ establishes that

$$\mathrm{d}_t x \propto (Dk)^{1/2}, \tag{6.29}$$

while coming back to the initial representation of time t and space x. Moreover, in a general way, the sharpness of the wave profile is measured by

$$\mathrm{d}_x u|_{\max} \propto (k/D)^{1/2}. \tag{6.30}$$

Note that condition $Dk = 1$ implies constant velocity for all $k = 1/D$; however, greater k (or smaller D) implies higher spatial gradients, and thus, stiffer configurations.
In the context of reaction waves, we have usually wavefronts of steep spatial gradient propagating with a steady constant speed. Therefore, we are interested in studying this kind of stiff configurations and their impact on a time parallelization algorithm such as the parareal scheme. In particular, the latter must be coupled with an appropriate stiff solver such as the operator splitting, considered in this work. Hence, in order to conduct a convergence analysis of the parareal scheme, we are first constrained to obtain some bounds (similar to those given by (6.23), (6.24) and (6.25)) drawn out of a more precise analysis of the operator splitting itself, applied to problem (6.27). This is the main goal of the following section.

Analysis of the Operator Splitting

For problem (6.27), we consider the linear operator b corresponding to the multiplication by kV and the operator $a = -D\partial_x^2$ (minus D times the second partial derivative with respect to

x in one dimension). Hence, their commutator is given by

$$\partial_a b = [a,b] = -(\partial_x^2 V) - 2(\partial_x V)\partial_x, \tag{6.31}$$

considering that $Dk = 1$. We first recall the following result which is proven in [DS02a] and gives an exact representation of the difference between the exponential of $a+b$, *i.e.*, the exact solution of (6.27), and its Lie (resp. Strang) approximation denoted by $\mathcal{L}$ (resp. $\mathcal{S}$).

Theorem 6.4. *The following identities hold*

$$\begin{aligned}
\mathcal{L}(t) &= \mathrm{e}^{-ta}\mathrm{e}^{-tb} \\
&= \mathrm{e}^{-t(a+b)} + \int_0^t \int_0^s \mathrm{e}^{-(t-s)(a+b)}\mathrm{e}^{-(s-r)a}\big(\partial_a b\big)\mathrm{e}^{-ra}\mathrm{e}^{-sb}\,\mathrm{d}r\,\mathrm{d}s, \qquad (6.32)\\
\mathcal{S}(t) &= \mathrm{e}^{-tb/2}\mathrm{e}^{-ta}\mathrm{e}^{-tb/2} = \mathrm{e}^{-t(a+b)} \\
&\quad + \frac{1}{4}\int_0^t \int_0^s (s-r)\mathrm{e}^{-(t-s)(a+b)}\mathrm{e}^{-(s-r)b/2}\big(\partial_b^2 a\big)\mathrm{e}^{-rb/2}\mathrm{e}^{-sa}\mathrm{e}^{-sb/2}\,\mathrm{d}r\,\mathrm{d}s \qquad (6.33)\\
&\quad - \frac{1}{2}\int_0^t \int_0^s (s-r)\mathrm{e}^{-(t-s)(a+b)}\mathrm{e}^{-sb/2}\mathrm{e}^{-ra}\big(\partial_a^2 b\big)\mathrm{e}^{-(s-r)a}\mathrm{e}^{-sb/2}\,\mathrm{d}r\,\mathrm{d}s.
\end{aligned}$$

In particular, for problem (6.27) we have

$$\partial_a^2 b = [a,[a,b]] = D(\partial_x^4 V) + 4D(\partial_x^3 V)\partial_x + 4D(\partial_x^2 V)\partial_x^2, \tag{6.34}$$

$$\partial_b^2 a = [b,[b,a]] = -2k(\partial_x V)^2. \tag{6.35}$$

Identities (6.32) and (6.33) are also valid for general nonlinear operators (or vector fields) a and b, in which case the commutators (6.31), (6.34) and (6.35) are computed as the Lie derivatives of the Lie bracket of the vector fields (see [HV03] and the complete analysis conducted in [Lub08] and [DDL+12]).

Nevertheless, even if (6.32) and (6.33) give us the exact representation of the local error of a Lie or Strang splitting approximation, we are more interested in defining some bounds on these local errors. More precisely, we will see in the next section that the key point of the proof of the convergence estimates for the parareal algorithm is the use of L^2 to L^2 estimates of the errors.

Therefore, considering the exact solution of problem (6.27), a simple computation shows that for $t \geq 0$,

$$\left\| \mathrm{e}^{t(D\partial_x^2 - kV)} \right\|_{\mathcal{L}(L^2,L^2)} \leq \mathrm{e}^{k\|V\|_\infty t}. \tag{6.36}$$

Moreover, with the exact representation of the local errors (6.32) and (6.33), and denoting by

$$\mathrm{E}_{\mathcal{L}}(t) = \mathrm{e}^{t(D\partial_x^2 - kV)} - \mathcal{L}(t) \quad \text{and} \quad \mathrm{E}_{\mathcal{S}}(t) = \mathrm{e}^{t(D\partial_x^2 - kV)} - \mathcal{S}(t), \tag{6.37}$$

it is straightforward to obtain the following bounds.

Lemma 6.5 (First estimate of the splitting local error)**.** *Consider problem (6.27) and $u_0 \in H^2(\mathbb{R})$, for $t \geq 0$ we have the following error bounds*

$$\begin{aligned}
\|\mathrm{E}_{\mathcal{L}}(t)u_0\|_{L^2} &\leq \left(\frac{t^2}{2}\|\partial_x^2 V\|_\infty + \frac{2t^3}{3}k\|\partial_x V\|_\infty^2\right)\mathrm{e}^{k\|V\|_\infty t}\|u_0\|_{L^2} \\
&\quad + t^2\|\partial_x V\|_\infty \mathrm{e}^{k\|V\|_\infty t}\|\partial_x u_0\|_{L^2}, \qquad (6.38)
\end{aligned}$$

and

$$\|\mathrm{E}_{\mathcal{S}}(t)u_0\|_{L^2} \leq \left(\frac{t^3}{12}\left(k\|\partial_x V\|_\infty^2 + D\|\partial_x^4 V\|_\infty\right) + \frac{t^4}{8}\left(\|\partial_x^3 V\|_\infty\|\partial_x V\|_\infty + \|\partial_x^2 V\|_\infty^2\right) + \frac{t^5}{20}k\|\partial_x^2 V\|_\infty\|\partial_x V\|_\infty^2\right)\mathrm{e}^{k\|V\|_\infty t}\|u_0\|_{L^2} + \left(\frac{t^3}{3}D\|\partial_x^3 V\|_\infty + \frac{t^4}{4}\|\partial_x^2 V\|_\infty\|\partial_x V\|_\infty\right)\mathrm{e}^{k\|V\|_\infty t}\|\partial_x u_0\|_{L^2} + \frac{t^3}{3}D\|\partial_x^2 V\|_\infty \mathrm{e}^{k\|V\|_\infty t}\|\partial_x^2 u_0\|_{L^2}. \tag{6.39}$$

Proof. Using (6.32), we have in $L^2(\mathbb{R})$,

$$\left\|\mathrm{e}^{t(D\partial_x^2 - kV)}u_0 - \mathrm{e}^{tD\partial_x^2}\mathrm{e}^{-tkV}u_0\right\|_{L^2} \leq \int_0^t\int_0^s \left\|\mathrm{e}^{(t-s)(D\partial_x^2-kV)}\mathrm{e}^{(s-r)D\partial_x^2}\partial_a b\,\mathrm{e}^{rD\partial_x^2}\mathrm{e}^{-skV}u_0\right\|_{L^2}\mathrm{d}r\,\mathrm{d}s. \tag{6.40}$$

And with (6.36) we have

$$\left\|\mathrm{e}^{t(D\partial_x^2 - kV)}u_0 - \mathrm{e}^{tD\partial_x^2}\mathrm{e}^{-tkV}u_0\right\|_{L^2} \leq \int_0^t\int_0^s \mathrm{e}^{(t-s)k\|V\|_\infty}\left\|\partial_a b\,\mathrm{e}^{rD\partial_x^2}\mathrm{e}^{-skV}u_0\right\|_{L^2}\mathrm{d}r\,\mathrm{d}s. \tag{6.41}$$

Since

$$\begin{aligned}\partial_a b\,\mathrm{e}^{rD\partial_x^2}\mathrm{e}^{-skV}u_0 &= -(\partial_x^2 V)\mathrm{e}^{rD\partial_x^2}\mathrm{e}^{-skV}u_0 - 2(\partial_x V)\partial_x\left(\mathrm{e}^{rD\partial_x^2}\mathrm{e}^{-skV}u_0\right)\\ &= -(\partial_x^2 V)\mathrm{e}^{rD\partial_x^2}\mathrm{e}^{-skV}u_0 - 2(\partial_x V)\mathrm{e}^{rD\partial_x^2}\partial_x\left(\mathrm{e}^{-skV}u_0\right)\\ &= -(\partial_x^2 V)\mathrm{e}^{rD\partial_x^2}\mathrm{e}^{-skV}u_0 - 2(\partial_x V)\mathrm{e}^{rD\partial_x^2}\mathrm{e}^{-skV}\partial_x u_0\\ &\quad + 2(\partial_x V)\mathrm{e}^{rD\partial_x^2}sk(\partial_x V)\mathrm{e}^{-skV}u_0,\end{aligned} \tag{6.42}$$

we obtain (6.38), integrating (6.41) and knowing that $\|\mathrm{e}^{-skV}\|_{\mathcal{L}(L^2,L^2)} \leq \mathrm{e}^{sk\|V\|_\infty}$.
Performing the same computations, we obtain (6.39) from (6.33), (6.34) and (6.35). □

However, we recall the fact that we are specifically searching for L^2 to L^2 error estimates in order to conduct the convergence analysis of the parareal scheme. Hence, thanks to the regularizing effect of the Laplacian, we have for all u_0 in L^2 and for $t > 0$,

$$\|\partial_x \mathrm{e}^{tD\partial_x^2}u_0\|_{L^2} \leq \frac{1}{\sqrt{2\mathrm{e}Dt}}\|u_0\|_{L^2}, \tag{6.43}$$

and then, we can conduct further calculations.

Lemma 6.6 (Second estimate of the splitting local error)**.** *Consider problem (6.27), for $t > 0$ we have the following error bounds*

$$\|\mathrm{E}_{\mathcal{L}}(t)u_0\|_{L^2} \leq \left(\frac{8}{3}t\sqrt{t}\left(\frac{\|\partial_x V\|_\infty}{\sqrt{2\mathrm{e}D}}\right) + \frac{t^2}{2}\|\partial_x^2 V\|_\infty\right)\mathrm{e}^{k\|V\|_\infty t}\|u_0\|_{L^2}, \tag{6.44}$$

and

$$\|\mathrm{E}_{\mathcal{S}}(t)u_0\|_{L^2} \leq \left(\frac{t^2}{4}\left(\frac{\pi}{\mathrm{e}}\|\partial_x^2 V\|_\infty\right) + \frac{t^3}{12}\left(k\|\partial_x V\|_\infty^2 + D\|\partial_x^4 V\|_\infty\right)\right)\mathrm{e}^{k\|V\|_\infty t}\|u_0\|_{L^2}. \tag{6.45}$$

Proof. We take into account that

$$\partial_a b\,\mathrm{e}^{rD\partial_x^2}\mathrm{e}^{-skV}u_0 = -(\partial_x^2 V)\mathrm{e}^{rD\partial_x^2}\mathrm{e}^{-skV}u_0 - 2(\partial_x V)\partial_x\left(\mathrm{e}^{rD\partial_x^2}\mathrm{e}^{-skV}u_0\right). \tag{6.46}$$

Coming back to (6.41), the regularizing effect of the Laplacian (6.43) yields

$$\begin{aligned}
&\left\|\mathrm{e}^{t(D\partial_x^2 - kV)}u_0 - \mathrm{e}^{tD\partial_x^2}\mathrm{e}^{-tkV}u_0\right\|_{L^2} \leq \\
&\int_0^t\int_0^s \mathrm{e}^{(t-s)k\|V\|_\infty}\left\|\partial_a b\,\mathrm{e}^{rD\partial_x^2}\mathrm{e}^{-skV}u_0\right\|_{L^2}\,\mathrm{d}r\,\mathrm{d}s \leq \\
&\int_0^t\int_0^s \mathrm{e}^{(t-s)k\|V\|_\infty}\left(\left(\frac{2\|\partial_x V\|_\infty}{\sqrt{2\mathrm{e}Dr}}\right) + \|\partial_x^2 V\|_\infty\right)\|\mathrm{e}^{-skV}u_0\|_{L^2}\,\mathrm{d}r\,\mathrm{d}s \leq \\
&\left(\frac{8}{3}t\sqrt{t}\left(\frac{\|\partial_x V\|_\infty}{\sqrt{2\mathrm{e}D}}\right) + \frac{t^2}{2}\|\partial_x^2 V\|_\infty\right)\mathrm{e}^{k\|V\|_\infty t}\|u_0\|_{L^2}.
\end{aligned} \tag{6.47}$$

Moreover, taking into account that

$$\partial_a^2 b = D(\partial_x^4 V) + 4D\partial_x\left((\partial_x^2 V)\partial_x\right) \tag{6.48}$$

and if we consider

$$\begin{aligned}
\mathrm{e}^{rD\partial_x^2}\partial_a^2 b\,\mathrm{e}^{(s-r)D\partial_x^2}\mathrm{e}^{-skV/2}u_0 =\ & D\mathrm{e}^{rD\partial_x^2}(\partial_x^4 V)\mathrm{e}^{(s-r)D\partial_x^2}\mathrm{e}^{-skV/2}u_0 \\
& + 4D\mathrm{e}^{rD\partial_x^2}\partial_x\left((\partial_x^2 V)\partial_x\right)\mathrm{e}^{(s-r)D\partial_x^2}\mathrm{e}^{-skV/2}u_0 \\
=\ & D\mathrm{e}^{rD\partial_x^2}(\partial_x^4 V)\mathrm{e}^{(s-r)D\partial_x^2}\mathrm{e}^{-skV/2}u_0 \\
& + 4D\partial_x\left(\mathrm{e}^{rD\partial_x^2}(\partial_x^2 V)\partial_x\mathrm{e}^{(s-r)D\partial_x^2}\mathrm{e}^{-skV/2}u_0\right),
\end{aligned} \tag{6.49}$$

we obtain with (6.43),

$$\begin{aligned}
&\left\|\mathrm{e}^{rD\partial_x^2}\partial_a^2 b\,\mathrm{e}^{(s-r)D\partial_x^2}\mathrm{e}^{-skV/2}u_0\right\|_{L^2} \leq \\
&D\|\partial_x^4 V\|_\infty\|\mathrm{e}^{-skV/2}u_0\|_{L^2} + \frac{4D}{\sqrt{2\mathrm{e}Dr}}\|\partial_x^2 V\|_\infty\left\|\partial_x\mathrm{e}^{(s-r)D\partial_x^2}\mathrm{e}^{-skV/2}u_0\right\|_{L^2} \leq \\
&D\|\partial_x^4 V\|_\infty\|\mathrm{e}^{-skV/2}u_0\|_{L^2} + \frac{2\|\partial_x^2 V\|_\infty}{\mathrm{e}\sqrt{r(s-r)}}\|\mathrm{e}^{-skV/2}u_0\|_{L^2} \leq \\
&\left(D\|\partial_x^4 V\|_\infty + \frac{2\|\partial_x^2 V\|_\infty}{\mathrm{e}\sqrt{r(s-r)}}\right)\mathrm{e}^{sk\|V\|_\infty/2}\|u_0\|_{L^2}.
\end{aligned} \tag{6.50}$$

Taking this into (6.33) and integrating yield (6.45). □

The main advantage of these local error bounds is that all terms in estimates (6.38), (6.39), (6.44) and (6.45) are known. No truncated terms in t are present indeed, since we have considered an exact representation of errors (6.32) and (6.33). Furthermore, first and second estimates of the splitting local error agree perfectly with those found in the literature (see [DDLM07]) for the case of PDEs with high spatial gradients. In particular, a more general and complete study of the Strang method can be found in [DS02a].
Notice that as it was stated in [DDLM07], an order reduction might appear in the local error. Thus, if we consider $k = D = 1$ and expand $\mathrm{e}^{\|V\|_\infty t}$, we infer from the previous results that for $t > 0$,

$$\|\mathrm{E}_{\mathcal{L}}(t)u_0\|_{L^2} \propto \max\left(\|\partial_x u_0\|_{L^2} t^2, \|u_0\|_{L^2} t^{1.5}\right) \tag{6.51}$$

and

$$\|\mathrm{E}_{\mathcal{S}}(t)u_0\|_{L^2} \propto \max\left(\|\partial_x u_0\|_{L^2} t^3, \|u_0\|_{L^2} t^2\right). \tag{6.52}$$

These estimates describe the behavior of the local errors; the first terms are more relevant when t is small and the second ones when t is not small enough and $\|\partial_x u_0\|_{L^2}$ is very high. More precisely there exists an explicit constant $\theta > 0$ such that for $t \leq \theta$, $\|\mathrm{E}_{\mathcal{L}}(t)u_0\|_{L^2}$ (resp. $\|\mathrm{E}_{\mathcal{S}}(t)u_0\|_{L^2}$) behaves like t^2 (resp. t^3) and for $t \geq \theta$, $\|\mathrm{E}_{\mathcal{L}}(t)u_0\|_{L^2}$ (resp. $\|\mathrm{E}_{\mathcal{S}}(t)u_0\|_{L^2}$) behaves like $t^{1.5}$ (resp. t^2).
These local error estimates are valid for general linear problems of type (6.27). However, in the context of propagating wavefronts, since the L^2-norm of $\partial_x u_0$ is normally very high, it is especially relevant to obtain alternative error estimates which do not involve the derivative of the initial condition. In our particular case, the L^2 to L^2 error estimates established in Lemma 6.6 allow us to obtain the following bounds which will be necessary for the convergence analysis of the parareal operator splitting scheme.

Lemma 6.7. *Consider problem (6.27) with initial conditions u_0 and v_0, then for a fixed $t > 0$, there exist $\kappa_{\mathcal{L}}, \kappa_{\mathcal{S}}, C_{\mathcal{L}}, C_{\mathcal{S}} \in \mathbb{R}^+$ such that we have the following bounds*

$$\|\mathcal{L}(t)u_0 - \mathcal{L}(t)v_0\|_{L^2} \leq \mathrm{e}^{k\|V\|_\infty t}\|u_0 - v_0\|_{L^2}, \tag{6.53}$$

$$\|\mathrm{E}_{\mathcal{L}}(t)u_0 - \mathrm{E}_{\mathcal{L}}(t)v_0\|_{L^2} \leq \kappa_{\mathcal{L}} C_{\mathcal{L}} t\sqrt{t}\mathrm{e}^{k\|V\|_\infty t}\|u_0 - v_0\|_{L^2}, \tag{6.54}$$

and

$$\|\mathcal{S}(t)u_0 - \mathcal{S}(t)v_0\|_{L^2} \leq \mathrm{e}^{k\|V\|_\infty t}\|u_0 - v_0\|_{L^2}, \tag{6.55}$$

$$\|\mathrm{E}_{\mathcal{S}}(t)u_0 - \mathrm{E}_{\mathcal{S}}(t)v_0\|_{L^2} \leq \kappa_{\mathcal{S}} C_{\mathcal{S}} t^2\mathrm{e}^{k\|V\|_\infty t}\|u_0 - v_0\|_{L^2}, \tag{6.56}$$

Proof. Simple calculation of $\|\mathrm{e}^{tD\partial_x^2}\mathrm{e}^{-tkV}(u_0 - v_0)\|_{L^2}$ yields (6.53), as well as $\|\mathrm{e}^{-tkV/2}\mathrm{e}^{tD\partial_x^2}\mathrm{e}^{-tkV/2}(u_0 - v_0)\|_{L^2}$ yields (6.55). Besides, estimates (6.54) and (6.56) come from the bounds (6.44) and (6.45) applied to $(u_0 - v_0)$ with

$$C_{\mathcal{L}} = \max\left(\frac{8\|\partial_x V\|_\infty}{3\sqrt{2\mathrm{e}D}}, \frac{\|\partial_x^2 V\|_\infty}{2}\right), \tag{6.57}$$

$$C_{\mathcal{S}} = \max\left(\frac{\pi\|\partial_x^2 V\|_\infty}{4\mathrm{e}}, \frac{k\|\partial_x V\|_\infty^2}{12}, \frac{D\|\partial_x^4 V\|_\infty}{12}\right), \tag{6.58}$$

$$\kappa_{\mathcal{L}} \geq 1 + \sqrt{t} \quad \text{and} \quad \kappa_{\mathcal{S}} \geq 1 + t. \tag{6.59}$$

□

Notice that bounds (6.54) (resp. (6.56)) and (6.53) (resp. (6.55)) correspond to estimates (6.24) and (6.25) respectively in [GH08], *i.e.*, the difference of local errors for different initial conditions and the Lipschitz condition related to the Lie (resp. Strang) approximation (we can even consider some $C \in \mathbb{R}^+$ such that $\mathrm{e}^{\|V\|_\infty t} \leq (1 + Ct)$ for $t \in (0, 1)$ into the different estimates). Moreover, series expansion (6.23) can be associated to (6.44) (resp. (6.45)). In this way, we have completely characterized the constants C_1, C_2 and C_3 appearing into classical bound (6.26) and a more detailed analysis of the algorithm can be performed.
The choice of the second estimate of the splitting local errors is justified by the fact that reaction waves phenomena involve usually wavefronts of high spatial gradients and that there is no such practical interest in utilizing very small time steps in order to simulate them. Besides, from a mathematical point of view, this allows us to obtain the necessary Lipschitz relations from L^2 to L^2 spaces.

Parareal Convergence Analysis

The following theorem gives us the convergence rate of the parareal operator splitting algorithm. Its demonstration is based on the preliminary lemmas stated before and on the convergence analysis developed by [GH08]. For reasons of simplicity, we assume that all the time subdomains are of the same size, *i.e.*, $\Delta T := T/N$, and $T_n = n\Delta T$ for $n = 0, 1, \ldots, N$.

Theorem 6.8. *Let $\mathcal{F}^{\Delta T}(U_{n-1})$ be the exact solution of problem (6.27) on time subdomain I_{n-1}. If $\mathcal{G}^{\Delta T}(U_{n-1})$ is the Lie approximate solution with local error bounded by (6.44) satisfying (6.53) and (6.54), then at iteration i of the parareal algorithm (6.5), there exist some finite $\kappa_{\mathcal{L}}, \kappa_{\mathcal{S}}, C_{\mathcal{L}}, C_{\mathcal{S}}, \Lambda_{T_n} \in \mathbb{R}^+$ such that we have the bound*

$$\|u(T_n) - \tilde{U}_n^i\|_{L^2} \leq \Lambda_{T_n} \frac{(\kappa_{\mathcal{L}} C_{\mathcal{L}} T_n)^{i+1}}{(i+1)!} \Delta T^{(i+1)/2} \tag{6.60}$$

with $\Lambda_{T_n} = \|u_0\|_{L^2} \mathrm{e}^{k\|V\|_\infty T_n}$. Otherwise, if $\mathcal{G}^{\Delta T}(U_{n-1})$ is the Strang approximate solution with local error bounded by (6.45) satisfying (6.55) and (6.56), then we have the bound

$$\|u(T_n) - \tilde{U}_n^i\|_{L^2} \leq \Lambda_{T_n} \frac{(\kappa_{\mathcal{S}} C_{\mathcal{S}} T_n)^{i+1}}{(i+1)!} \Delta T^{(i+1)}. \tag{6.61}$$

Proof. The proof is similar to the one conducted in [GH08]. From the parareal algorithm (6.5), considering that $\mathcal{F}$ is the exact solution of (6.27) and adding and subtracting $\mathcal{G}^{\Delta T}(u(T_{n-1}))$, we obtain

$$\begin{aligned} u(T_n) - \tilde{U}_n^{i+1} = \; & \mathcal{F}^{\Delta T}(u(T_{n-1})) - \mathcal{G}^{\Delta T}(u(T_{n-1})) \\ & - \left(\mathcal{F}^{\Delta T}(\tilde{U}_{n-1}^i) - \mathcal{G}^{\Delta T}(\tilde{U}_{n-1}^i)\right) + \mathcal{G}^{\Delta T}(u(T_{n-1})) - \mathcal{G}^{\Delta T}(\tilde{U}_{n-1}^{i+1}). \end{aligned} \tag{6.62}$$

Hence, taking norms and considering (6.54) (or (6.56)) and (6.53) (or (6.55)), there exist some α and β such that

$$\|u(T_n) - \tilde{U}_n^{i+1}\|_{L^2} \leq \alpha \|u(T_{n-1}) - \tilde{U}_{n-1}^i\|_{L^2} + \beta \|u(T_{n-1}) - \tilde{U}_{n-1}^{i+1}\|_{L^2}. \tag{6.63}$$

The classical convergence analysis lead us to study the recurrence relation

$$\mathrm{e}_n^{i+1} = \alpha \mathrm{e}_{n-1}^i + \beta \mathrm{e}_{n-1}^{i+1}, \quad \mathrm{e}_n^0 = \gamma + \beta \mathrm{e}_{n-1}^0, \tag{6.64}$$

where e_n^i is an upper bound on $\|u(T_n) - \tilde{U}_n^i\|_{L^2}$, the global error of the parareal scheme at T_n considering the exact solution $u(t)$. After induction and using the binomial series expansion, we obtain the bound [GH08]

$$e_n^i \leq \gamma \alpha^i \beta^{n-i-1} \left(\begin{array}{c} n \\ i+1 \end{array} \right). \tag{6.65}$$

If $\mathcal{G}$ is the Lie approximate solution,

$$\left. \begin{aligned} \alpha &= \kappa_{\mathcal{L}} C_{\mathcal{L}} \Delta T \sqrt{\Delta T} e^{k\|V\|_\infty \Delta T}, \\ \beta &= e^{k\|V\|_\infty \Delta T}, \\ \gamma &= \kappa_{\mathcal{L}} C_{\mathcal{L}} \Delta T \sqrt{\Delta T} e^{k\|V\|_\infty \Delta T} \|u_0\|_{L^2}, \end{aligned} \right\} \tag{6.66}$$

then,

$$\begin{aligned} \|u(T_n) - \tilde{U}_n^i\|_{L^2} \leq & \\ & \|u_0\|_{L^2} \frac{(\kappa_{\mathcal{L}} C_{\mathcal{L}} \Delta T \sqrt{\Delta T} e^{k\|V\|_\infty \Delta T})^{i+1}}{(i+1)!} (e^{k\|V\|_\infty \Delta T})^{n-i-1} \prod_{j=0}^{i} (n-j) \\ & \leq \|u_0\|_{L^2} (e^{k\|V\|_\infty \Delta T})^n \frac{(\kappa_{\mathcal{L}} C_{\mathcal{L}} n \Delta T \sqrt{\Delta T})^{i+1}}{(i+1)!} \\ & \leq \|u_0\|_{L^2} e^{k\|V\|_\infty T_n} \frac{(\kappa_{\mathcal{L}} C_{\mathcal{L}} T_n)^{i+1}}{(i+1)!} \Delta T^{(i+1)/2}. \end{aligned} \tag{6.67}$$

If $\mathcal{G}$ is the Strang approximate solution, we perform the same process to obtain (6.61) with

$$\left. \begin{aligned} \alpha &= \kappa_{\mathcal{S}} C_{\mathcal{S}} \Delta T^2 e^{k\|V\|_\infty \Delta T}, \\ \beta &= e^{k\|V\|_\infty \Delta T}, \\ \gamma &= \kappa_{\mathcal{S}} C_{\mathcal{S}} \Delta T^2 e^{k\|V\|_\infty \Delta T} \|u_0\|_{L^2}. \end{aligned} \right\} \tag{6.68}$$

□

These convergence results show that the parareal algorithm converges superlinearly with respect to i and behaves as a method of order $p(i+1)$ for a problem of type (6.27) and a finite T. Notice that this time $p = 1/2$ for the Lie formula instead of classical global order 1 and $p = 1$ for Strang instead of 2, according to the order reduction previously discussed and proven in [DDLM07]. Thus, results (6.60) and (6.61) do not contradict the classical results from the literature but complement them, giving new and more detailed insights to the performance of the algorithm. In fact, all the terms into bounds (6.60) and (6.61) are known or can be calculated for a general problem of type (6.27), even more, constants $C_{\mathcal{L}}$, $C_{\mathcal{S}}$, $\kappa_{\mathcal{L}}$ and $\kappa_{\mathcal{S}}$ have been established in the proof of Lemma 6.7. This means that more precise estimates can be obtained for a parareal operator splitting strategy.

6.3.3 Parareal Operator Splitting for Reaction Waves

Theorem 6.8 gives us the convergence rate while applying a parareal operator splitting strategy on a general problem of type (6.27). Nevertheless, we will now extend these results to

the particular case of reaction waves phenomena. In this context, we are interested in the propagation of steady self-similar waves, *i.e.*, parabolic PDEs of type (6.14) with solution

$$u(x,t) = u_0(x - ct), \tag{6.69}$$

where c is the steady speed of the wavefront.
We have already showed the link between problems (6.27) and (6.14), furthermore a simple dimensionless analysis of traveling waves yield expressions (6.29) and (6.30) with scaling constants D and k (see [GS94]). Therefore, we can easily show that for an integer $s > 0$,

$$\partial_x^s V(x) = \partial_x^s f'(u_0) = \partial_x^{s-1}(f''(u_0)\partial_x u_0). \tag{6.70}$$

Taking norms and taking into consideration (6.30) under condition $kD = 1$, we obtain that

$$\|\partial_x^s V(x)\|_\infty = \|\partial_x^s f'(u_0)\|_\infty = \|\partial_x^{s-1}(f''(u_0)\partial_x u_0)\|_\infty = k^s \|\partial_x^{s-1}(f''(u_0)\partial_x \bar{u}_0)\|_\infty, \tag{6.71}$$

where $\bar{u}(x,t) = \bar{u}_0(x - ct)$ is a reference solution of (6.14) when $k = 1$. Moreover, following (6.29), condition $Dk = 1$ implies constant speed c for all k. Then, we can rewrite Lemma 6.7 as follows.

Lemma 6.9. *Consider problem (6.27) with initial conditions u_0 and v_0. Furthermore, let us assume that $Dk = 1$ and that condition (6.71) is satisfied. Denoting by $\mathrm{E}_{\mathcal{L}}(t) = \mathrm{e}^{t(D\partial_x^2 - kf'(u_0))} - \mathcal{L}(t)$, $\mathrm{E}_{\mathcal{S}}(t) = \mathrm{e}^{t(D\partial_x^2 - kf'(u_0))} - \mathcal{S}(t)$ and $\tau = kt$, then for a fixed $t > 0$ there exist $\kappa_{\mathcal{L}}, \kappa_{\mathcal{S}}, C_{\mathcal{L}}, C_{\mathcal{S}} \in \mathbb{R}^+$ such that we have the following bounds*

$$\|\mathrm{E}_{\mathcal{L}}(t)u_0 - \mathrm{E}_{\mathcal{L}}(t)v_0\|_{L^2} \leq \kappa_{\mathcal{L}} C_{\mathcal{L}} \tau \sqrt{\tau} \mathrm{e}^{\|f'(u_0)\|_\infty \tau} \|u_0 - v_0\|_{L^2}, \tag{6.72}$$

and

$$\|\mathrm{E}_{\mathcal{S}}(t)u_0 - \mathrm{E}_{\mathcal{S}}(t)v_0\|_{L^2} \leq \kappa_{\mathcal{S}} C_{\mathcal{S}} \tau^2 \mathrm{e}^{\|f'(u_0)\|_\infty \tau} \|u_0 - v_0\|_{L^2}. \tag{6.73}$$

Proof. The proof of (6.72) and (6.73) is straightforward to obtain, considering (6.71) and rewriting (6.44) and (6.45) as

$$\|\mathrm{E}_{\mathcal{L}}(t)u_0\|_{L^2} \leq \left(\frac{8}{3} t\sqrt{t} \left(\frac{k\|g(u_0)\|_\infty}{\sqrt{2\mathrm{e}D}} \right) + \frac{t^2}{2} k^2 \|\partial_x g(u_0)\|_\infty \right) \mathrm{e}^{k\|f'(u_0)\|_\infty t} \|u_0\|_{L^2}, \tag{6.74}$$

$$\begin{aligned} \|\mathrm{E}_{\mathcal{S}}(t)u_0\|_{L^2} \leq \Bigg(& \frac{t^2}{4} \left(\frac{\pi}{\mathrm{e}} k^2 \|\partial_x g(u_0)\|_\infty \right) \\ & + \frac{t^3}{12} \left(k^3 \|g(u_0)\|_\infty^2 + Dk^4 \|\partial_x^3 g(u_0)\|_\infty \right) \Bigg) \mathrm{e}^{k\|f'(u_0)\|_\infty t} \|u_0\|_{L^2}, \end{aligned} \tag{6.75}$$

where $g(u_0) = f''(u_0)\partial_x \bar{u}_0$, then we take $Dk = 1$, $\tau = kt$ and redefine $C_{\mathcal{L}}$ and $C_{\mathcal{S}}$ as

$$C_{\mathcal{L}} = \max\left(\frac{8\|g(u_0)\|_\infty}{3\sqrt{2\mathrm{e}}}, \frac{\|\partial_x g(u_0)\|_\infty}{2} \right), \tag{6.76}$$

$$C_{\mathcal{S}} = \max\left(\frac{\pi\|\partial_x g(u_0)\|_\infty}{4\mathrm{e}}, \frac{\|g(u_0)\|_\infty^2}{12}, \frac{\|\partial_x^3 g(u_0)\|_\infty}{12} \right). \tag{6.77}$$

□

With this lemma, the following corollary of Theorem 6.8 can be obtained.

Corollary 6.10. *Let $\mathcal{F}^{\Delta T}(U_{n-1})$ be the exact solution of problem (6.27) on time subdomain I_{n-1}. Furthermore, let us assume that $Dk = 1$ and that condition (6.71) is satisfied. If $\mathcal{G}^{\Delta T}(U_{n-1})$ is the Lie approximate solution with local error bounded by (6.72) satisfying (6.53), then at iteration i of the parareal algorithm (6.5), there exist some finite $\kappa_{\mathcal{L}}, \kappa_{\mathcal{S}}, C_{\mathcal{L}}, C_{\mathcal{S}}, \Lambda_{T_n} \in \mathbb{R}^+$ such that we have the bound*

$$\|u(T_n) - \tilde{U}_n^i\|_{L^2} \leq \Lambda_{T_n} \frac{(\kappa_{\mathcal{L}} C_{\mathcal{L}} k T_n)^{i+1}}{(i+1)!} (k\Delta T)^{(i+1)/2} \tag{6.78}$$

with $\Lambda_{T_n} = \|u_0\|_{L^2} \mathrm{e}^{\|f'(u_0)\|_\infty k T_n}$. Otherwise, if $\mathcal{G}^{\Delta T}(U_{n-1})$ is the Strang approximate solution with local error bounded by (6.73) satisfying (6.55), then we have the bound

$$\|u(T_n) - \tilde{U}_n^i\|_{L^2} \leq \Lambda_{T_n} \frac{(\kappa_{\mathcal{S}} C_{\mathcal{S}} k T_n)^{i+1}}{(i+1)!} (k\Delta T)^{(i+1)}. \tag{6.79}$$

Proof. The proof comes out directly from Theorem 6.8 and Lemma 6.9, considering $Dk = 1$ and condition (6.71). □

We see that also in this case, all the terms in bounds (6.78) and (6.79) are known or can be calculated for a general problem of type (6.27) satisfying (6.71). Moreover, constants $C_{\mathcal{L}}$, $C_{\mathcal{S}}$, $\kappa_{\mathcal{L}}$ and $\kappa_{\mathcal{S}}$ have been established in the proof of Lemma 6.9 and this time, $C_{\mathcal{L}}$ and $C_{\mathcal{S}}$ are completely independent of coefficients k and D, so they are valid for any speed/gradient configuration of the wavefront. By the way, from bounds of Lemma 6.9 as well as those of Corollary 6.10, we can consider coefficient k as a time scaling parameter through $\tau = kt$.
From these results we conclude that the convergence rate of the algorithm in the Lie (resp. Strang) case behaves like $k^{1.5}$ (resp. k^2) for a fixed ΔT. And as established by condition (6.30), higher k implies the propagation of wavefronts with higher spatial gradients at the same speed. As a matter of fact, the following corollary establishes more precisely the influence of high spatial gradients onto the performance of the parareal algorithm.

Corollary 6.11. *Let us consider Lemma 6.9 and Corollary 6.10. Denoting by $E_n^i = \|u(T_n) - \tilde{U}_n^i\|_{L^2}$ with fixed $k \geq 1$ into (6.27). Then, at iteration i of the parareal algorithm (6.5), there exist some finite $A, B, q \in \mathbb{R}^+$ such that we have the bound*

$$E_n^i \leq \mathrm{e}^{A + (i+1)qB} \overline{E}_n^i, \tag{6.80}$$

where $\overline{E}_n^i$ is a fixed reference value of E_n^i with $k = D = 1$ in (6.27).

Proof. From Corollary 6.10, we have

$$E_n^i \leq \Lambda_{T_n} k^{q(i+1)} \frac{(\kappa_{\mathcal{G}} C_{\mathcal{G}} T_n)^{i+1}}{(i+1)!} \Delta T^{(q-1)(i+1)}, \tag{6.81}$$

with $q = 3/2$, $C_{\mathcal{G}} = C_{\mathcal{L}}$ and $\kappa_{\mathcal{G}} = \kappa_{\mathcal{L}}$ (resp. $q = 2$, $C_{\mathcal{G}} = C_{\mathcal{S}}$ and $\kappa_{\mathcal{G}} = \kappa_{\mathcal{S}}$) if $\mathcal{G}$ is the Lie (resp. Strang) approximate solution.
After simple computations, we obtain

$$\ln E_n^i \leq (k-1)\|f'(u_0)\|_\infty T_n + q(i+1)\ln k + \ln \overline{E}_n^i, \tag{6.82}$$

that yields (6.80) with

$$A \geq (k-1)\|f'(u_0)\|_\infty T_n \quad \text{and} \quad B \geq \ln k. \tag{6.83}$$

□

Thus, Corollary 6.11 establishes the bound on the convergence rate of the algorithm for a fixed k into (6.27); as a consequence, the convergence rate is directly related to the value of the highest spatial gradient of the wavefront. In particular, the simulation of sharper wavefronts (higher k) implies weaker convergence rates since constants A and B into (6.80) increase. This can be easily shown from (6.80) if we consider the line $r_n(i)$:

$$\ln E_n^i - \ln \overline{E}_n^i \leq r_n(i) = A + (i+1)qB, \tag{6.84}$$

where we see that its slope is directly proportional to $\ln k$ through $B \propto \ln k$ for the same approximate solution, *i.e.*, same q; and the convergence rate gets lower when sharper spatial gradients are present in the solution.
Now that we have obtained these results that explain the behavior of the parareal algorithm when we consider an operator splitting scheme as the coarse approximation technique, we need to validate them with some numerical examples. This is the goal of the next section.

6.4 Numerical Simulations

In this section, we will present some illustrating simulations in order to first, validate the theoretical results presented before, and secondly, to investigate the performance of such algorithms on multi-scale reaction wave phenomena. Both mathematical models considered in the following come from nonlinear chemical dynamics; nevertheless, the conclusions might be extended to similar reaction-diffusion models in other domains. In this way, the 1D KPP model is a clarifying example of the deduced behavior of the parareal splitting technique applied on such type of problems. Then, with a more efficient operator splitting strategy inspired by [DDD+12, DMD+12], we will consider a much more complex model, the 1D and 2D BZ system, that will also confirm the previous results and will give us some insights into the performance of the algorithm and the alternative strategies for these multi-scale problems.

6.4.1 1D KPP Equation

Following the theoretical investigations we have presented, we focus in this part on the numerical evidence of the convergence rate reduction associated to splitting solvers in a typical nonlinear framework of stiff traveling waves. Let us first recall the Kolmogorov-Petrovskii-Piskunov model. In their original paper dated in 1937 [KPP37], these authors introduced a model describing the propagation of a virus and the first rigorous analysis of a stable traveling wave solution of a nonlinear reaction-diffusion equation [GS94, VVV94]. The equation is the following:

$$\partial_t u - D\,\Delta u = k\,u^2(1-u), \tag{6.85}$$

with homogeneous Neumann boundary conditions.
The description of the dimensionless model and the structure of the exact solution can be found in [GS94]. Thus, the dimensionless analysis shows that in the case of $D = 1$ and $k = 1$, the velocity of the self-similar traveling wave is $c = 1/\sqrt{2}$ and the maximal gradient value reaches $1/\sqrt{32}$. The structure of the wave can be observed in Figure 6.1 with a discretization of 5001 points of the interval $[-70, 70]$ and a time varying in $[0, 30]$ divided into eight time intervals.
The key point of this illustration is that the velocity of the traveling wave is proportional to $(k\,D)^{1/2}$, whereas the maximal gradient is proportional to $(k/D)^{1/2}$. Thus, switching to values $k = 10.0$ and $D = 0.1$, the velocity is preserved, but the maximal gradient is multiplied by a factor of 10 and introduces stiffness in the equation, as presented in Figure 6.2. For the spatial discretizations considered, the wave, however "stiff", is always well solved on the considered

grid. This model is then a very suitable example because it coincides perfectly with the general reaction-diffusion structure considered throughout our theoretical study.

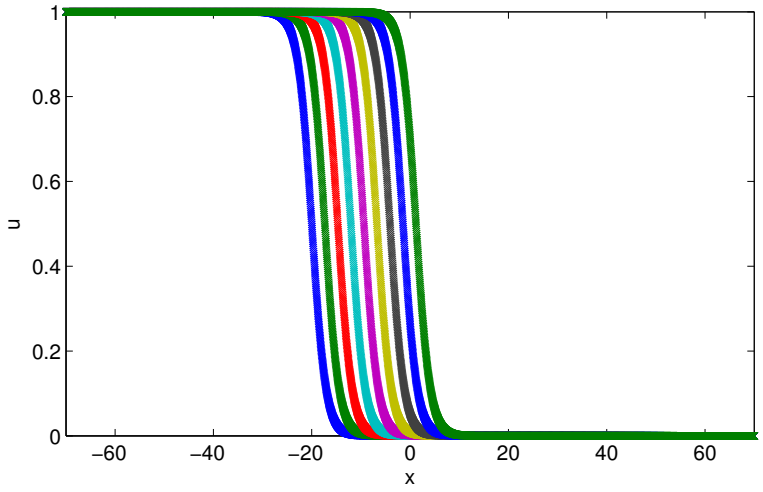

Fig. 6.1: *Standard KPP traveling wave, discretization with* 5001 *points on the* $[-70, 70]$ *region. Self-similar solutions for eight time intervals after the initial condition.*

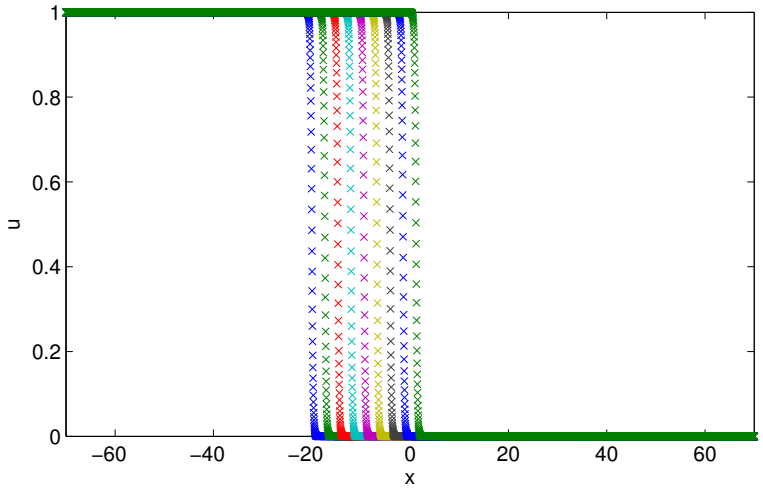

Fig. 6.2: *"Stiff" KPP traveling wave, discretization with* 5001 *points on the* $[-70, 70]$ *region. Self-similar solutions for eight time intervals after the initial condition.*

Application of the method of lines with a finite difference second order discretization in space implies a discretization of the Laplacian operator in (6.85) and thus, leads to a system of nonlinear ODEs. For the parareal scheme, we then need the fine and coarse temporal integrations of this semi-discretized problem. Based on the previous considerations, we choose a Lie (or Strang) split scheme as the coarse solver and a very accurate Lie (or Strang) split scheme as the fine one, the latter considers smaller splitting time step. In all cases, each of the time integration substeps (reaction and diffusion substeps) is integrated by an unique and very accurate solver in time: LSODE with very fine predefined tolerances. In this way, we decouple

the errors originated by the operator splitting itself from the errors coming from the temporal integration of the substeps, in order to remain coherent with the theoretical study conducted in the paper.
We start from an already self-similar solution and evaluate the ability of the parareal solution to reproduce the correct self-similar profile. The splitting time steps for both fine and coarse solvers, are defined such that the wave speed is correctly calculated. Figure 6.3 and 6.4 below show the convergence results of the parareal algorithm considering the convergence of the parareal iterative solutions towards the fine solution, computed separately and sequentially by the fine solver. The first iteration corresponds to the initial coarse approximation and the time domain is decomposed into $N = 128$ time subdomains.
In Figure 6.3, on the one hand we observe a clean and fast iterative evolution towards convergence of the Lie splitting technique applied to the standard KPP equation. In fact, only 10 over 128 subdomains have been solved directly by the fine solver to achieve the fine accuracy. On the other hand, a much more lower convergence rate of the algorithm is shown while applying the parareal Lie scheme to the "stiff" KPP equation. In fact, after 15 iterations we are very far from the accuracy obtained in the standard case.

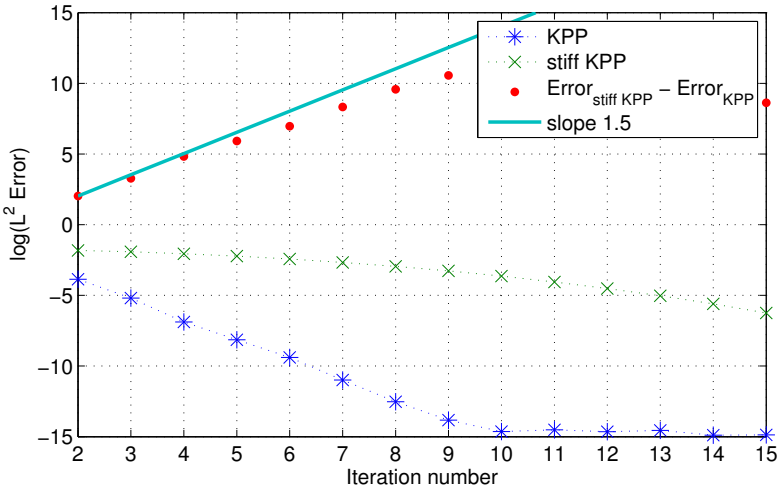

Fig. 6.3: *Comparison of convergence rates of the parareal operator splitting at time $t = 15$ for 1D KPP equation for the Lie splitting scheme.*

The key point in this paper is the reduction of the convergence rate exhibited by the "stiff" case, where the velocity of the wave is the same, but the maximal gradient is ten times higher. Thus, considering the line defined by (6.84) as a consequence of Corollary 6.11 where $B = 1$ since $k = 10$ (considering log instead of ln), we have plotted a straight line of slope $q = 1.5$ in order to show the very good agreement to the bound predicted by the theory.
The same conclusions arise from the Strang case, as it is shown in Figure 6.4, where this time $q = 2$, as established into Corollary 6.11. Notice that the parareal scheme applied to the standard KPP equation converges faster than in the previous Lie case, as a Strang scheme is naturally more accurate that a Lie one. Thus, we obtain less coarse approximations and convergence after 5 iterations. Notice also that the stiff case practically converges after 15 iterations, which was not the case in the previous Lie application (Figure 6.3). Therefore, let us consider a more accurate Lie application where the splitting time steps are smaller so that the stiff case converges completely. Figure 6.5 describes this situation where once again the theoretical results are validated and for which convergence is achieved after 5 (resp. 9)

iterations for the standard (resp. stiff) case.

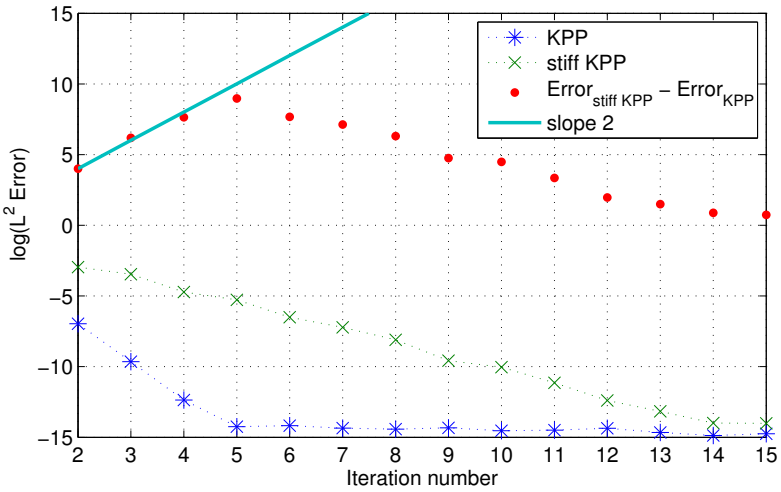

Fig. 6.4: *Comparison of convergence rates of the parareal operator splitting at time $t = 15$ for 1D KPP equation for the Strang splitting scheme.*

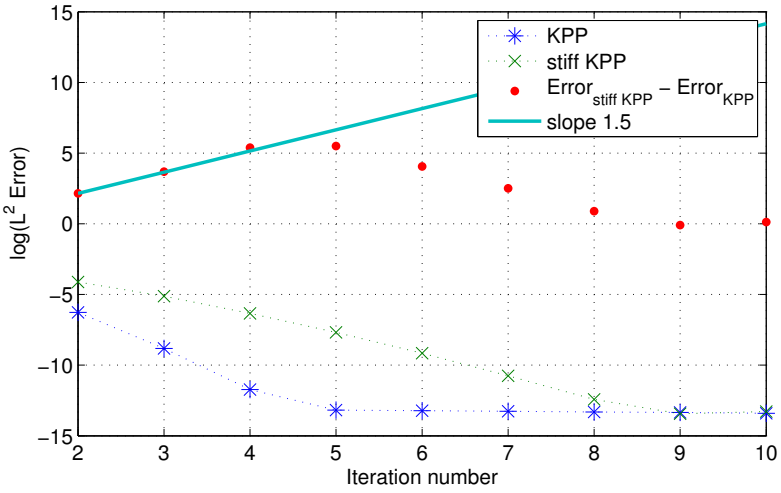

Fig. 6.5: *Comparison of convergence rates of the parareal operator splitting at time $t = 15$ for 1D KPP equation for a more accurate Lie scheme.*

Let us finally see what happens with a coarse solver that considers the initial coupled reaction-diffusion system instead of the split one. To limit important computational cost, let us consider an explicit solver such us ROCK4 [Abd02]. This alternative might not be always feasible because of the important stability restrictions of this method when treating problems with important imaginary part eigenvalues (see [Abd02] for more details). We reproduce in Figure 6.6, the former convergence rates found in the Lie application of Figure 6.3 for both, the standard and stiff KPP equation, and the new one obtained by the ROCK4-based coarse solver for the stiff case. The latter coupled coarse resolution clearly improves the convergence rate of the parareal scheme for the stiff KPP problem. Hence, we see that an operator splitting

approach as coarse approximation clearly implies an important reduction of the convergence rate of the parareal algorithm as it was proven in the previous section.

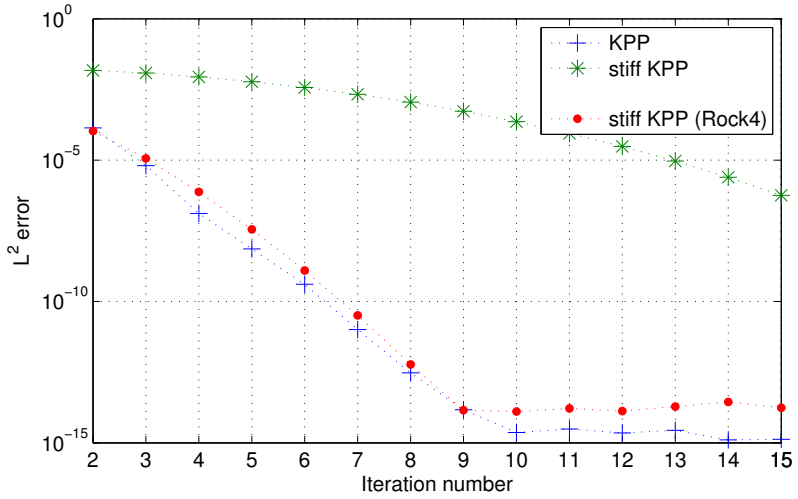

Fig. 6.6: *Comparison of convergence rates of the parareal algorithm with either the operator splitting or the ROCK4 solver as coarse solver, at time $t = 15$ for 1D KPP equation.*

6.4.2 BZ Equation

In this second illustration, we are concerned with the numerical approximation of another model coming from nonlinear chemical dynamics, the Belousov-Zhabotinski reaction, a catalyzed oxidation of an organic species by acid bromated ion (for more details and illustrations, see [EP98]). We can first consider the two-variable Oregonator model, studied in [JSW89]; 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 consider the following model:

$$\left.\begin{array}{rcl} \dfrac{\partial b}{\partial \tau} - D_b \Delta b &=& \dfrac{1}{\varepsilon}\left(b(1-b) + \dfrac{f(q-b)c}{q+b}\right), \\ \dfrac{\partial c}{\partial \tau} - D_c \Delta c &=& b - c, \end{array}\right\} \tag{6.86}$$

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 [GS94] and coming from the classic work of Field, Koros and Noyes (FKN) (1972), 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:

$$\left.\begin{array}{rcl} \dfrac{\partial a}{\partial \tau} - D_a \Delta a &=& \dfrac{1}{\mu}(-qa - ab + fc), \\ \dfrac{\partial b}{\partial \tau} - D_b \Delta b &=& \dfrac{1}{\varepsilon}\left(qa - ab + b(1-b)\right), \\ \dfrac{\partial c}{\partial \tau} - D_c \Delta c &=& b - c, \end{array}\right\} \tag{6.87}$$

with additional diffusion coefficient D_a and real positive parameter $\mu \ll \varepsilon$.
The dynamical systems associated to systems (6.86) and (6.87) model reactive excitable media with a large time scale spectrum (see [GS94] for more details). Moreover, the spatial configuration with addition of diffusion generates propagating wavefronts with steep spatial gradients. Thus, this model presents all the difficulties associated to a stiff multi-scale configuration. The advantages of applying a splitting strategy to these models have already been studied and presented in [DDM03, DMD+12]. In what follows, we will consider the 1D and 2D configurations of problem (6.87).

1D BZ Equation

Let us first consider the 1D case of problem (6.87) with homogeneous Neumann boundary conditions and the following parameters (based on [GS94]): $\varepsilon = 10^{-2}$, $\mu = 10^{-5}$, $f = 3$ and $q = 2.10^{-4}$, with diffusion coefficients $D_a = 1$, $D_b = 1$ and $D_c = 0.6$. Following the same partial discretization applied in the previous problem, the structure of the waves can be observed in Figure 6.7 with a discretization of 4001 points of the interval $[0, 80]$ and a time varying in $[0, 2]$ divided into eight time intervals.

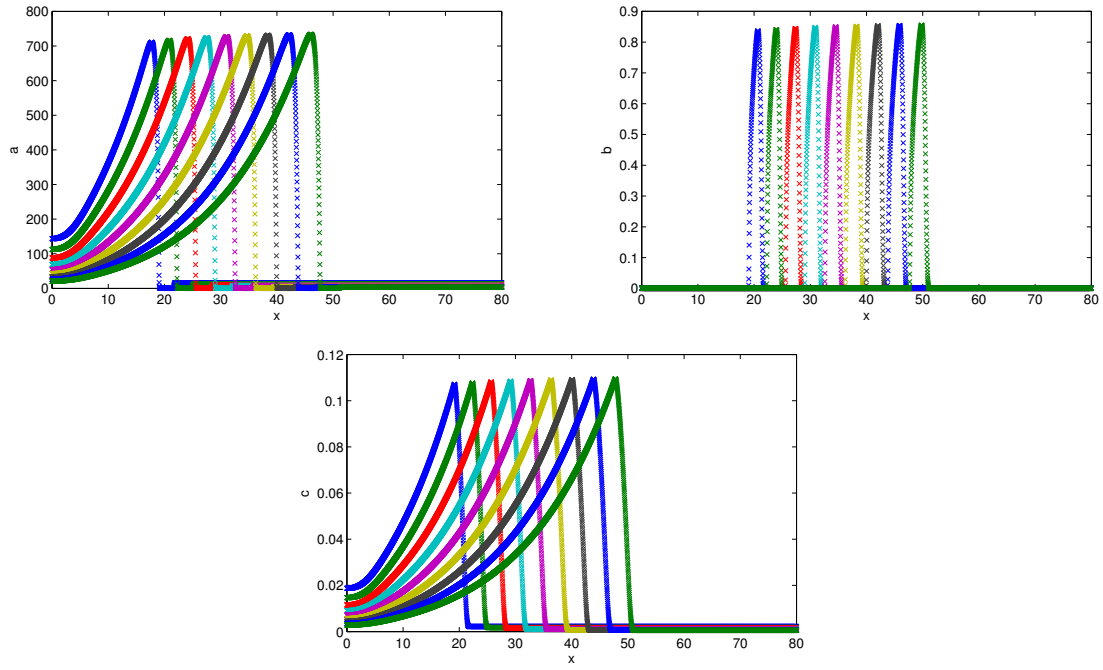

Fig. 6.7: *1D BZ traveling waves, discretization with* 4001 *points on the* $[0, 80]$ *region. Self-similar solutions for eight time intervals after the initial condition.*

We then construct an optimal operator splitting configuration already studied and validated [DDD+12, DMD+12] based on the RDR Strang $\mathcal{S}_2^t$ scheme for which, Radau5 is used for the time integration of the reaction term and ROCK4 for the diffusive part. In this context, Figure 6.8 shows the iterative evolution and the convergence rate obtained with a RDR Strang operator splitting scheme considered for both fine and coarse solvers, the latter with larger splitting time steps. In the same figure, we see that once again a coupled resolution of the initial reaction-diffusion system (6.87) as coarse solver yields better performances. Also in this illustration, we take ROCK4 as coarse solver of the coupled system.

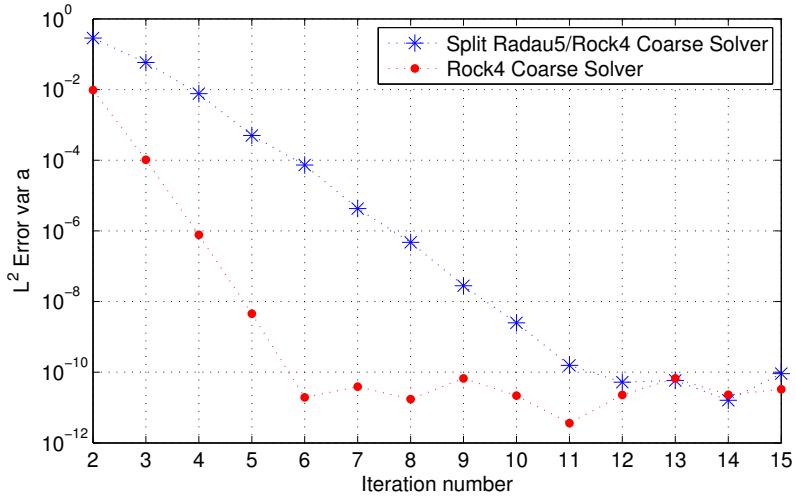

Fig. 6.8: *Comparison of convergence rates of the parareal algorithm with an operator splitting or ROCK4 as coarse solver, at time $t = 2$ for 1D BZ equation.*

2D BZ Equation

We now consider the 2D configuration of problem (6.87) with homogeneous Neumann boundary conditions and the following parameters (from [DDM03]): $\varepsilon = 10^{-2}$, $\mu = 10^{-5}$, $f = 1.6$ and $q = 2.10^{-3}$, with diffusion coefficients $D_a = 2.5 \times 10^{-3}$, $D_b = 2.5 \times 10^{-3}$ and $D_c = 1.5 \times 10^{-3}$. We follow the same partial discretization applied in the previous case and after integration over a time domain of $[0, 2]$, we see the developed spiral waves at final time $t = 2$ into Figure 6.9.

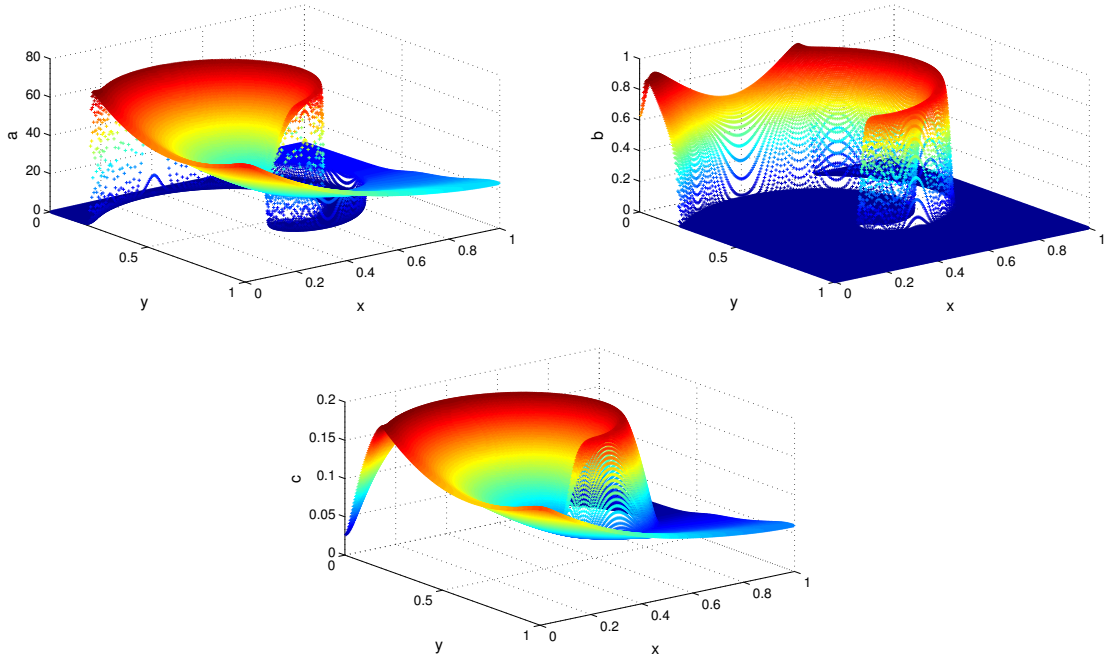

Fig. 6.9: *2D BZ spiral waves on a* $[257 \times 257]$ *grid at* $t = 2$.

In this illustration, we are concerned with the evaluation of the potential gains obtained in the computational time through a partial parallel computation process. In this way, let us first introduce the following notation:

T_{fine}: computation time of the fine solver over the whole time domain;

T_{coarse}: computation time of the coarse solver over the whole time domain;

N_{proc}: number of processors, *i.e.,* number of time subdomains considered; and

N_{ite}: number of iterations.

In a parallel environment, the computation time utilized by the parareal algorithm, T_{para}, should be equivalent to the time needed by the fine resolution divided by the number of processors available, that is

$$T_{para} \approx N_{ite} \times \frac{T_{fine}}{N_{proc}}, \tag{6.88}$$

taking into account the iterative procedure. We set this estimate as the optimal ratio, when coarse approximations are very fast. Nevertheless, a more realistic estimate should take into account all coarse approximations computed sequentially in order to initialize each time subdomain. Then, we have

$$T_{para} \approx (N_{ite} + 1) \times T_{coarse} + N_{ite} \times \frac{T_{fine}}{N_{proc}}, \tag{6.89}$$

with the initial coarse approximation needed to start the algorithm. Thus, the choice of the coarse solver is crucial and represents the major constraint on the success of the applications. In this context, let us make some computations using the RDR Strang $\mathcal{S}_2^t$ operator splitting

Table 6.1: *Computation time in seconds, 2D BZ.*

Grid	129×129		257×257	
Coarse solver	RDR Strang	ROCK4	RDR Strang	ROCK4
T_{fine}	2769.94	2757.52	11291.36	11149.42
T_{coarse}	228.53	256.07	1006.62	1177.89
T_{fine}/T_{coarse}	12.12	10.77	11.22	9.47
N_{ite}	4	2	4	2
T_{para}	1279.81	860.30	5581.32	3869.51

established in the previous 1D case and ROCK4, as coarse solvers for two different grids, $[129 \times 129]$ and $[257 \times 257]$, where the time domain has been decomposed into $N = N_{proc} = 64$ time subdomains. Naturally, the fine solver is an accurate RDR Strang operator splitting. According to the parareal scheme, this fine resolution is performed on a cluster made of 32 nodes with 2 processors AMD Opteron 64 bits dual core with speed 2.4 GHz; the numerical code is written in Fortran 95 and uses the MPI library for the parallel features. The results related to each configuration are summarized in Table 6.1, where the stopping criterion is based on the same order of difference between successive parareal solutions for both configurations. Notice that the splitting scheme is faster than a coupled resolution even if an explicit method such

Table 6.2: *Computation time ratios, 2D BZ.*

Grid	129×129		257×257	
Coarse solver	RDR Strang	ROCK4	RDR Strang	ROCK4
N_{proc}	64			
N_{proc}/N_{ite}	16	32	16	32
T_{fine}/T_{para}	**2.16**	**3.21**	**2.02**	**2.88**

as ROCK4 has been utilized. Nevertheless, the coupled computation as coarse approximation allows faster convergence and thus, a better performance of the algorithm.
This is also reflected in Table 6.2 where the computation time ratios have been estimated. N_{proc}/N_{ite} then represents the optimal ratio described into expression (6.88): 16 for the split coarse solver and 32 for the coupled one. Hence, the final ratios obtained ($\sim$ 2-3) reflect on the one hand, the important cost of the coarse solvers for stiff problems, and on the other hand, the reduction of convergence rate for an operator splitting scheme as coarse propagator, previously deduced.

6.5 Conclusions

In this paper, we have presented a new numerical strategy that couples an optimal operator splitting technique for multi-scale problems and the parareal algorithm, a promising time parallelization scheme. Considering the state of the art of the literature, we have then conducted a detailed numerical analysis of such techniques in order to introduce complementary and necessary characterization of the numerical behavior of operator splitting and parareal schemes in the context of multi-scale reaction waves. In particular, a precise representation of the impact of stiff configurations on the performance of the algorithm has been mathematically proven for reaction-diffusion models with large spatial gradients.
Then, a set of numerical illustrations have on the one hand, validated the previous theoretical results and explicitly showed the reduction of the convergence rate of the parareal algorithm when an operator splitting is performed for the coarse approximations of the method. And on the other hand, they have highlighted the need of efficient coarse solvers for stiff problems in order to construct more competitive methods.
Finally, it was also numerically demonstrated that a more suitable strategy considers a coupled resolution of the initial reaction-diffusion system as coarse solver. However, this alternative may not always be feasible and other approaches should be evaluated. For instance, the resolution of the reduced model of the BZ system or the set of slow variables for a general multi-scale problem could be considered as a coarse approximation of the complete fine model. Nevertheless, further studies that go out of the scope of this work are necessary in order to test such alternatives; this is the topic of our current research.

Acknowledgements. This research was supported by a fundamental project grant from ANR (French National Research Agency - ANR CIS) *PITAC* (Project leader Y. Maday), and by a CNRS Ph.D. grant for M. Duarte from the Mathematics (INSMI) and Engineering (INSIS) Institutes of CNRS. This work was granted access to the HPC resources of IDRIS

(CNRS Institute of Scientific Computing) under the allocation 2009-i2009066173 made by GENCI (Grand Equipement National de Calcul Intensif) where some of the simulations were performed. The authors are grateful to Stéphane de Chaisemartin, Thierry Dumont, Violaine Louvet and Frédérique Laurent for valuable discussions and important assistance throughout this project.

Part III

Algorithmic Description and Basic Implementation Issues: MBARETE Code

Introduction

In this part, we present some key aspects concerning the practical implementation of the numerical strategy established in chapters 4 and 5. These techniques gave birth to an academic numerical code called MBARETE, with time/space adaptive features for the solution of stiff PDEs. Chapter 7 describes the most important multiresolution algorithms included in the code and gives some details on the data and code structures.

Additionally, Chapter 8 details the main core of the time integration scheme implemented in the MBARETE code, for reaction-diffusion systems. The time adaptive and dedicated splitting techniques are also described as well as the complementary numerical procedure introduced in Chapter 5. Further extensions will be considered for the simulations illustrated in the next part.

Chapter 7

Description of Space Adaptive Multiresolution in the MBARETE Code

We detail in this chapter the algorithms and the main issues concerning the practical implementation of the multiresolution schemes detailed in Chapter 3, for the numerical strategy introduced in Chapter 4. The general algorithms can be found in the original articles of Harten [Har94a, Har95, BH97], and more details on these and other related matters in the book of Müller [Mül03]. The multiresolution analysis was implemented in our code based on the standard tree-structured data representation, as in the approach of Roussel *et al.* [RSTB03]. More efficient and sophisticated data structures for adaptive grid representation were developed by Brix *et al.* [BMV09], and successfully implemented for multiresolution applications (see, *e.g.*, [BBB+10] and references therein). The multiresolution technique implemented in the MBARETE code is largely based on of the multiresolution kernel of MR CHORUS, a code developed by Christian Tenaud for compressible Navier-Stokes equations. Further details on this multiresolution implementation are included in a tutorial that have been elaborated for a Summer School of CNRS *GDR Groupe Calcul*, on Multiresolution and Adaptive Mesh Refinement Methods, Fréjus, France (2010) [TD11].

7.1 Multiresolution Operations

Before introducing the algorithms for the different multiresolution operators, let us recall some notations introduced in Chapter 3. Considering a computational domain $\Omega \subset \mathbb{R}^d$, we define the set of disjoint partitions $(\Omega_\gamma)_{\gamma \in S_j}$ such that

$$\Omega = \bigcup_{\gamma \in S_j} \Omega_\gamma, \tag{7.1}$$

and

$$\Omega_\gamma \bigcap \Omega_\mu = \varnothing, \qquad \gamma \neq \mu;\ \gamma, \mu \in S_j. \tag{7.2}$$

The abbreviated notation $\Omega_\gamma := \Omega_{j,k}$ was defined with $j \in \mathbb{N}_0 \bigcap [j_0, J]$, and $k \in \mathbb{Z}^d$. The first index j stands for the grid level from the coarsest $j = j_0$ to the finest grid $j = J$, where the different grids are generated by a successive, dyadic, and nested partition. For each cell $\Omega_{j,k}$,

there are 2^d subcells $\Omega_{j+1,k'}$, such that $\Omega_{j+1,k'} \subset \Omega_{j,k}$, and

$$\Omega_{j,k} = \bigcup_{k' \in \mathrm{I}_k} \Omega_{j+1,k'}, \tag{7.3}$$

where I_k, with $\#(\mathrm{I}_k) = 2^d$, is the set of indices k' such that $\Omega_{j+1,k'} \subset \Omega_{j,k}$. In a Cartesian coordinates framework, we thus have

$$|\Omega_{j,k}| = \int_{\Omega_{j,k}} \mathrm{dx}, \qquad \mathrm{x} \in \mathbb{R}^d. \tag{7.4}$$

The second set of indices $k \in \mathbb{Z}^d$ into $(j,k) \in S_j$, accounts for the spatial position of $\Omega_{j,k}$ at grid level j, as illustrated by (3.66) for the 1D case.

7.1.1 Projection Operator

Considering finite volumes for the spatial discretization, we denote $\mathbf{U}_j := (u_{j,k})_{(j,k)\in S_j}$, as the spatial representation of a given function $u(t,\mathrm{x})$ on the level grid j, where $u_{j,k}$ represents the cell-average of $u : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}$ in $\Omega_{j,k}$ given by (3.67):

$$u_\gamma := |\Omega_\gamma|^{-1} \int_{\Omega_\gamma} u(t,\mathrm{x})\,\mathrm{dx}, \quad \mathrm{x} \in \mathbb{R}^d. \tag{7.5}$$

The cell-average values at two successive grid levels are related by the projection operator P_{j-1}^j, previously defined in (3.68):

$$u_{j,k} = \frac{1}{|\Omega_{j,k}|} \sum_{k' \in \mathrm{I}_k} |\Omega_{j+1,k'}| u_{j+1,k'}. \tag{7.6}$$

The latter procedure allows us to compute in a recursive way, all the values $u_{j,k}$ for all $j \in [j_0, j]$, from $\mathbf{U}_{j+1}$ at a given grid level $j+1$.

7.1.2 Prediction Operator

The computations from coarse to fine grids are performed by the prediction operator P_j^{j-1}, based on polynomial interpolation. For a 1D configuration, the interpolation stencil $R_{j+1,k'}$ (3.86) to approximate the values at grid level $j+1$: $\hat{u}_{j+1,2k}$ and $\hat{u}_{j+1,2k+1}$, contains the parent-cell $u_{j,k}$ and its nearest M neighbors. The centered polynomial interpolations of accuracy order $N = 2M+1$ might be written for the 1D case as

$$\left.\begin{aligned} \hat{u}_{j+1,2k_1} &= u_{j,k_1} + \sum_{d_1=1}^{M} \xi_{d_1} \left(u_{j,k_1+d_1} - u_{j,k_1-d_1}\right), \\ \hat{u}_{j+1,2k_1+1} &= u_{j,k_1} - \sum_{d_1=1}^{M} \xi_{d_1} \left(u_{j,k_1+d_1} - u_{j,k_1-d_1}\right), \end{aligned}\right\} \tag{7.7}$$

where $k = k_1 \in \mathbb{Z}$, and the coefficients ξ_{d_1} are given in Table 7.1 up to $M = 4$. The case $N = 3$ was already given in (3.85), whereas the case $N = 1$ corresponds to the Haar wavelet decomposition (3.14) into (3.80).

Extensions to multi-dimensional interpolations is straightforward based on the 1D configuration (7.7). Defining the expression Q^M as

$$Q^M(k_1, u_{j,k}) = \sum_{d_1=1}^{M} \xi_{d_1} \left(u_{j,k_1+d_1} - u_{j,k_1-d_1}\right), \tag{7.8}$$

N	M	ξ_1	ξ_2	ξ_3	ξ_4
1	0	0	0	0	0
3	1	$-1/8$	0	0	0
5	2	$-22/128$	$3/128$	0	0
7	3	$-201/1024$	$11/256$	$-5/1024$	0
9	4	$-3461/16384$	$949/16384$	$-185/16384$	$35/32768$

Table 7.1: *Prediction operator. Coefficients for polynomial interpolations of order* $N = 2M + 1$ *[Har94a].*

the 2D polynomial interpolation, proposed by Bihari & Harten [BH97], reads

$$\begin{aligned}\hat{u}_{j+1,2k_1+d_1,2k_2+d_2} &= u_{j,k} + (-1)^{d_1} Q^M(k_1, u_{j,k}) + (-1)^{d_2} Q^M(k_2, u_{j,k}) \\ &\quad -(-1)^{(d_1+d_2)} Q_2^M(k_1, k_2, u_{j,k}),\end{aligned} \tag{7.9}$$

with $k = (k_1, k_2) \in \mathbb{Z}^2$. The integers d_1 and d_2 are equal to either 0 or 1 depending on the child-cell considered, and Q^M (7.8) is used in both dimensions. The operator Q_2^M, derived from a tensor product is given by

$$\begin{aligned}Q_2^M(k_1, k_2, u_{j,k}) &= \sum_{d_1=1}^{M} \xi_{d_1} \sum_{d_2=1}^{M} \xi_{d_2} (u_{j,k_1+d_1,k_2+d_2} - u_{j,k_1-d_1,k_2+d_2} \\ &\quad -u_{j,k_1+d_1,k_2-d_2} + u_{j,k_1-d_1,k_2-d_2}).\end{aligned} \tag{7.10}$$

In the same way, 3D interpolations are defined by introducing the operator Q_3^M:

$$\begin{aligned}Q_3^M(k_1, k_2, k_3, u_{j,k}) &= \sum_{d_1=1}^{M} \xi_{d_1} \sum_{d_2=1}^{M} \xi_{d_2} \sum_{d_3=1}^{M} \xi_{d_3} (u_{j,k_1+d_1,k_2+d_2,k_3+d_3} \\ &\quad -u_{j,k_1-d_1,k_2+d_2,k_3+d_3} - u_{j,k_1+d_1,k_2-d_2,k_3+d_3} \\ &\quad -u_{j,k_1+d_1,k_2+d_2,k_3-d_3} + u_{j,k_1-d_1,k_2-d_2,k_3+d_3} \\ &\quad +u_{j,k_1-d_1,k_2+d_2,k_3-d_3} + u_{j,k_1+d_1,k_2-d_2,k_3-d_3} \\ &\quad -u_{j,k_1-d_1,k_2-d_2,k_3-d_3}).\end{aligned} \tag{7.11}$$

Hence,

$$\begin{aligned}\hat{u}_{j+1,2k_1+d_1,2k_2+d_2,2k_3+d_3} &= u_{j,k} + (-1)^{d_1} Q^M(k_1, u_{j,k}) + (-1)^{d_2} Q^M(k_2, u_{j,k}) \\ &\quad +(-1)^{d_3} Q^M(k_3, u_{j,k}) \\ &\quad -(-1)^{(d_1+d_2)} Q_2^M(k_1, k_2, u_{j,k}) \\ &\quad -(-1)^{(d_1+d_3)} Q_2^M(k_1, k_3, u_{j,k}) \\ &\quad -(-1)^{(d_2+d_3)} Q_2^M(k_2, k_3, u_{j,k}) \\ &\quad +(-1)^{(d_1+d_2+d_3)} Q_3^M(k_1, k_2, k_3, u_{j,k}),\end{aligned} \tag{7.12}$$

with $k = (k_1, k_2, k_3) \in \mathbb{Z}^3$. As before, d_1, d_2, and d_3 are equal to either 0 or 1.

7.1.3 Multiresolution Transform

With the projection and prediction inter-level operators, the solution $\mathbf{U}_J$ at the finest grid of size N_J is *encoded* by a multiresolution transform $\mathcal{M}$ into a multi-scale representation $\mathbf{M}_J$ (3.75), by means of the Algorithm 7.1. In general, the sets S_j with $j \in \mathbb{N}_0 \bigcap [j_0, J]$ are distributed in N_{R} trees Λ_r for $r = 1, \ldots, N_{\mathrm{R}}$, where N_{R} denotes the number of roots in the entire tree representation, and thus $N_J = N_{\mathrm{R}} 2^{dJ}$. For the moment this tree structure is useless because all cells are considered, and the representation on the leaves $\mathrm{L}(\Lambda_r)$ of size $N_{\mathrm{L}} = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\Lambda_r))$ coincides with the discretized solution at the finest grid, *i.e.*, $N_{\mathrm{L}} = N_J$.

Algorithm 7.1 Encoding by multiresolution transform $\mathcal{M} : \mathbf{U}_J \longmapsto \mathbf{M}_J$.

Input: $\mathbf{U}_J$ of size $N_{\mathrm{L}} = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\Lambda_r))$, given by cell-averaged values $u_{j,k}$ such that $(j,k) \in \bigcup_{r \in [1, N_{\mathrm{R}}]} \mathrm{L}(\Lambda_r)$.
for $r = 1 \to N_{\mathrm{R}}$ **do**
  for $j = J - 1 \to j_0$ **do**
    for k s.t. $(j,k) \in S_j \bigcap \Lambda_r$ **do**
      Compute all $u_{j,k}$ at grid level j, from $u_{j+1,k'}$ with $k' \in \mathrm{I}_k$ at level $j+1$, by using the projection operator P_{j-1}^{j} (7.6).
      Compute for each I_k, $2^d - 1$ predicted values $\hat{u}_{j+1,k'}$ by polynomial interpolation (7.8), (7.10), or (7.11), and the corresponding details defined by (3.71): $d_{j+1,k'} = u_{j+1,k'} - \hat{u}_{j+1,k'}$.
      Save details in the array $\mathbf{D}_{j+1}$, where $\#(\mathbf{D}_{j+1}) = r(2^d - 1)2^{dj}$ if $N_{\mathrm{L}} = N_J$. If needed, the last detail can be computed from $\sum_{k' \in \mathrm{I}_k} d_{j+1,k'} = 0$, according to (3.73).
      Encode the solution by replacing $\mathbf{U}_{j+1}$ by $(\mathbf{U}_j, \mathbf{D}_{j+1})$.
    end for
  end for
end for
Output: $\mathbf{M}_J = (\mathbf{U}_{j_0}, \mathbf{D}_{j_0+1}, \mathbf{D}_{j_0+2}, \cdots, \mathbf{D}_J)$ of size N_{L}.

With this new data representation on the wavelet space, the details in $\mathbf{D}_j$ account for the local spatial smoothness in the solution, according to (3.105). Nevertheless, a *decoding* procedure is also necessary to retrieve the representation on the physical space of the variables. The latter is done by means of the inverse multiresolution transform $\mathcal{M}^{-1}$, following the Algorithm 7.2.

Algorithm 7.2 Decoding by inverse multiresolution $\mathcal{M}^{-1} : \mathbf{M}_J \longmapsto \mathbf{U}_J$.

Input: $\mathbf{M}_J = (\mathbf{U}_{j_0}, \mathbf{D}_{j_0+1}, \cdots, \mathbf{D}_J)$ of size N_{L}, given by the representation on the coarsest grid: $\mathbf{U}_{j_0}$, and the set of detail arrays: $(\mathbf{D}_{j_0+1}, \cdots, \mathbf{D}_J)$.
for $r = 1 \to N_{\mathrm{R}}$ **do**
  for $j = j_0 \to J - 1$ **do**
    for k s.t. $(j,k) \in S_j \bigcap \Lambda_r$ **do**
      Compute for each $u_{j,k}$, $2^d - 1$ predicted $\hat{u}_{j+1,k'}$ such that $k' \in \mathrm{I}_k$, by polynomial interpolation (7.8), (7.10), or (7.11), and the corresponding $u_{j+1,k'}$ by (3.71): $u_{j+1,k'} = \hat{u}_{j+1,k'} + d_{j+1,k'}$.

Compute last remaining $u_{j+1,k'}$ from $u_{j,k}$ and the $2^d - 1$ computed $u_{j+1,k'}$, using the projection operator P_{j-1}^{j} (7.6).
Save $u_{j+1,k'}$ in the array $\mathbf{U}_{j+1}$, where $\#(\mathbf{U}_{j+1}) = r2^{d(j+1)}$ if $N_{\mathrm{L}} = N_J$.
Decode the solution by replacing $(\mathbf{U}_j, \mathbf{D}_{j+1})$ by $\mathbf{U}_{j+1}$.
end for
end for
end for
Output: $\mathbf{U}_J$ of size $N_{\mathrm{L}} = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\Lambda_r))$, given by cell-averaged values $u_{j,k}$ such that $(j,k) \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\Lambda_r)$.

Notice that the algorithms 7.1 and 7.2 remain valid even if only some indices are retained from the sets S_j, as long as the corresponding adapted trees Λ_r are graded. In this case, the representation of $\mathbf{U}_J$ on the leaves $\bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\Lambda_r)$ does not coincide anymore with the finest representation, *i.e.*, $N_{\mathrm{L}} \neq N_J$.

7.2 Tree-Structured Data

Introducing $\Lambda_{r,\varepsilon}$, the smallest graded tree containing the adapted thresholded tree Λ_r obtained by $\mathcal{T}_{\Lambda_r}$ (3.87), we detail in the following the thresholding procedure and the construction of the graded trees. The trees Λ_r are built based on the sets $\nabla_j \subset S_j$, 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 ensure the one-to-one correspondence (3.74). Enlarging Λ_r by adding the missing child-cells, generates the set $R(\Lambda_r)$. The refinement operator $\mathcal{R}$ builds the tree $\tilde{\Lambda}_{r,\varepsilon}$ containing $\Lambda_{r,\varepsilon}$, such that the Harten's heuristics (3.102) are guaranteed. In this part, we introduce a binary flag $t_{j,k}$ which indicates whether the index (j,k) is kept throughout the successive trees, from Λ_r to $\tilde{\Lambda}_{r,\varepsilon}$. Initially, $t_{j,k} =$ `.false.`, except for $j = j_0$, *i.e.*, $t_{j_0,k} =$ `.true.`.

7.2.1 Thresholding and Graduation

The Algorithm 7.3 illustrates the thresholding procedure that defines the trees Λ_r, according to (3.87).

Algorithm 7.3 Thresholding the multiresolution representation: $\mathcal{T}_{\Lambda_r}\mathbf{M}_J$.

Input: $\mathbf{M}_J = (\mathbf{U}_{j_0}, \mathbf{D}_{j_0+1}, \cdots, \mathbf{D}_J)$ of size $N_{\mathrm{L}} = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\Lambda_r))$.
for $r = 1 \to N_{\mathrm{R}}$ **do**
  for $j = J \to j_0 + 1$ **do**
    for k s.t. $(j,k) \in S_j \bigcap \Lambda_r$ **do**
      if $|d_{j,k}| \geq \varepsilon_j$ **then**
        $t_{j,k} =$ `.true.` $\Rightarrow (j,l) \in \Lambda_r$.
      end if
    end for
  end for
end for
Output: $\mathbf{M}_J = (\mathbf{U}_{j_0}, \mathbf{D}_{j_0+1}, \cdots, \mathbf{D}_J)$ of size $N_{\mathrm{L}} = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\Lambda_r))$.

Notice that according to (3.87), for all $t_{j,k} =$ `.false.`, *i.e.*, $|d_{j,k}| < \varepsilon_j$, we should impose $d_{j,k} = 0$ into Λ_r. This is not done at this stage because the graduation procedure might keep some cells

with $|d_{j,k}| < \varepsilon_j$, and thus imposing $d_{j,k} = 0$ will introduce some errors while coming back to the physical representation the in Algorithm 7.2.
In the original hyperbolic applications in [Har94a, Har95], the level dependent threshold values ε_j were given by (3.96):

$$\varepsilon_j = 2^{d(j-J)}\varepsilon, \quad j \in [j_0, J], \tag{7.13}$$

which in terms of the nonlinear approximation in (3.42), implies $q = 1$ and $p = \infty$ for the Hölder's inequality [CDD04]. The details $d_{j,k}$ defined by (3.71) were thus evaluated in the uniform norm $|d_{j,k}|_\infty$, as done in (3.92) [CKMP03, Mül03]. In this implementation, we adopt a set of ε_j given rather by (4.15):

$$\varepsilon_j = 2^{\frac{d}{2}(j-J)}\varepsilon, \quad j \in [j_0, J], \tag{7.14}$$

in order to be consistent with the evaluation of the approximation errors in an L^2-norm framework. Concerning (3.42), the latter choice implies $q = p = 2$ as in the general orthonormal and biorthogonal wavelets context. We must define in this case the measure $\|d_{j,k}\|_{L^2(\Sigma_{j,k})}$, where $\Sigma_{j,k}$ stands for the support of the dual wavelet $\tilde{\psi}_{j,k}$ into (3.80).
In this implementation, we have thus established a discretized ℓ^2-norm:

$$\|d_{j+1,k'}\|_{\ell^2(\chi_{\Omega_{j,k}})} = 2^{-d/2}\left(\sum_{k' \in \mathrm{I}_k} d^2_{j+1,k'}\right)^{-1/2}, \tag{7.15}$$

which considers that $\operatorname{supp}\tilde{\psi}_{j+1,k'} = \operatorname{supp}\tilde{\phi}_{j,k} = \chi_{\Omega_{j,k}}$ for the Haar wavelet into (3.80), *i.e.*, $N = 1$ and $M = 0$ for the polynomial interpolations in Table 7.1. In this way, according to (7.15) all the 2^d child-cells $\Omega_{j+1,k'}$ associated with $\Omega_{j,k}$ involve the same $\|d_{j+1,k'}\|_{\ell^2(\chi_{\Omega_{j,k}})}$, to compare with ε_{j+1} in the Algorithm 7.3. We have nevertheless kept the same estimate (7.15) for $N \geq 3$ and $M \geq 1$, to restrain the measure $\|d_{j+1,k'}\|_{\ell^2(\chi_{\Omega_{j,k}})}$ to only 2^d cells. In a more general case, for a vector function $U : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^m$, given by

$$U(t, \mathrm{x}) = (u^{(1)}(t, \mathrm{x}), u^{(2)}(t, \mathrm{x}), \ldots, u^{(m)}(t, \mathrm{x}))^T, \tag{7.16}$$

for m variables, where $u^{(i)}_{j,k}$ represents the cell-average of variable $u^{(i)}(t, \mathrm{x})$ in $\Omega_{j,k}$ given by (7.5). The final estimate of the local detail is computed by

$$\|d_{j+1,k'}\|_{\ell^2(\chi_{\Omega_{j,k}})} = m^{-1/2}\sum_{i=1}^{m}\|d^{(i)}_{j+1,k'}\|_{\ell^2(\chi_{\Omega_{j,k}})}\left(\max_{\mathrm{x}\in\Omega} u^{(i)}(t, \mathrm{x})\right)^{-1/2}, \tag{7.17}$$

based on (7.15) for the estimate for each variable, and normalized by the maximum value of the corresponding variable.
The Algorithm 7.4 allows us to consider only graded tree structures based on the stencil needed to perform the polynomial interpolations. Additionally, we consider that if a cell $(j+1, k')$ is contained in Λ_r, then the same holds for the $2^d - 1$ (resp., $2^d - 2$) other cells with $k' \in \mathrm{I}_k$ in the extended $R(\Lambda_r)$ (resp., in Λ_r).

Algorithm 7.4 Graduation of the tree structure: $\Lambda_r \to \Lambda_{r,\varepsilon}$.

1: **Input:** $\mathbf{M}_J = (\mathbf{U}_{j_0}, \mathbf{D}_{j_0+1}, \cdots, \mathbf{D}_J)$ of size $N_\mathrm{L} = \sum_{r=1}^{N_\mathrm{R}} \#(\mathrm{L}(\Lambda_r))$.
2: **for** $r = 1 \to N_\mathrm{R}$ **do**
3: **for** $j = J - 1 \to j_0$ **do**

for k' s.t. $(j+1,k') \in S_{j+1} \bigcap \Lambda_r$ **do**
  if $t_{j+1,k'} =$ `.true.` **then**
    for $k' \in \mathrm{I}_k$ **do**
      $t_{j+1,k'} =$ `.true.` $\Rightarrow (j+1,k') \in R(\Lambda_r)$.
    end for
    if $j \geq j_0 + 1$ **then**
      for $d_3 = -M \rightarrow M$ **do**
        for $d_2 = -M \rightarrow M$ **do**
          for $d_1 = -M \rightarrow M$ **do**
            $t_{j,k_1+d_1,k_2+d_2,k_3+d_3} =$ `.true.` $\Rightarrow (j,k_1+d_1,k_2+d_2,k_3+d_3) \in R(\Lambda_r)$.
          end for
        end for
      end for
    end if
  end if
end for
end for
end for
Output: $\Lambda_r \rightarrow \Lambda_{r,\varepsilon}$, with $\mathbf{M}_J = (\mathbf{U}_{j_0}, \mathbf{D}_{j_0+1}, \cdots, \mathbf{D}_J)$ of size $N_{\mathrm{L}} = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\Lambda_{r,\varepsilon}))$.

Algorithms 7.3 and 7.4 are finally represented by the combined threshold-graduation operator $\mathcal{T}_{\Lambda_{r,\varepsilon}}$, as seen in chapters 3 and 4.

7.2.2 Refinement of the Tree

The refinement operator $\mathcal{R}$ generates the final tree data structure $\tilde{\Lambda}_{r,\varepsilon}$, on which the time evolution of the solution $u(t,\mathrm{x})$ will be performed. The main concern is to ensure Harten's heuristics (3.102) during one time step evolution. Harten proposed in [Har94a, Har95] two refinement criteria given by (3.103) and (3.104) for the 1D dyadic configuration. In this implementation we consider the first Harten's criterion (3.103), and a more conservative second criterion (4.13), as established in Chapter 4. These criteria are implemented by the Algorithm 7.5.

Algorithm 7.5 Refinement of the tree structure: $\Lambda_{r,\varepsilon} \rightarrow \tilde{\Lambda}_{r,\varepsilon}$.

Input: $\mathbf{M}_J = (\mathbf{U}_{j_0}, \mathbf{D}_{j_0+1}, \cdots, \mathbf{D}_J)$ of size $N_{\mathrm{L}} = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\Lambda_{r,\varepsilon}))$.
for $r = 1 \rightarrow N_{\mathrm{R}}$ **do**
  for $j = J \rightarrow j_0$ **do**
    for k s.t. $(j,k) \in S_j \bigcap \mathrm{L}(\Lambda_{r,\varepsilon})$ **do**
      if $t_{j,k} =$ `.true.` **then**
        if $j \geq j_0 + 1$ **then** {First refinement criterion}
          for $i_d = 1 \rightarrow d$ **do**
            for $l = -\bar{k} \rightarrow \bar{k}$ **do**
              $t_{j,k(i_d)+l} =$ `.true.`
              if $(j, k(i_d)+l) \notin R(\Lambda_{r,\varepsilon})$ **then**
                Apply steps 6-17 of Algorithm 7.4 for graduation, with $j = j-1$.
                Set $d_{j,k(i_d)+l} = 0$.
                $(j, k(i_d)+l) \in R(\Lambda_{r,\varepsilon})$.
              end if

```
15:              end for
16:            end for
17:          end if
18:          if |d_{j,k}| ≥ ε_j and j < J then {Second refinement criterion}
19:            for k' ∈ I_k do
20:              t_{j+1,k'} = .true..
21:              if (j+1,k') ∉ R(Λ_{r,ε}) then
22:                Apply steps 6-17 of Algorithm 7.4 for graduation.
23:                Set d_{j+1,k'} = 0.
24:                (j+1,k') ∈ R(Λ_{r,ε}).
25:              end if
26:            end for
27:          end if
28:        end if
29:      end for
30:    end for
31: end for
32: Output: Λ_{r,ε} → Λ̃_{r,ε}, with M_J = (U_{j_0}, D_{j_0+1}, ··· , D_J) of size N_L = Σ_{r=1}^{N_R} #(L(Λ̃_{r,ε})).
```

Notice that the first refinement criterion in the Algorithm 7.5 can be performed only at the highest level $j = J$. Then, the graduation (Algorithm 7.4) can be applied with $\bar{M} = M + \lceil \bar{k}/2 \rceil$ as grading parameter for dimensions x, y, and z, and $\bar{M} = M$ for the remaining diagonal cells with respect to (j, k). Alternatively, the second refinement criterion can be directly implemented during the thresholding (Algorithm 7.3). Eventually, the final step summarized in Algorithm 7.6 deletes completely all cells that are not included in the thresholded, graded, and refined tree $\tilde{\Lambda}_{r,\varepsilon}$.

Algorithm 7.6 Deletion of superfluous cells.

```
1: Input: M_J = (U_{j_0}, D_{j_0+1}, ··· , D_J) of size N_L = Σ_{r=1}^{N_R} #(L(Λ̃_{r,ε})).
2: for r = 1 → N_R do
3:   for j = J − 1 → j_0 do
4:     for k' s.t. (j+1,k') ∈ S_{j+1} ⋂ Λ̃_{r,ε} do
5:       if t_{j+1,k'} = .false. then
6:         for k' ∈ I_k do
7:           Delete (j+1,k').
8:         end for
9:       end if
10:    end for
11:  end for
12: end for
13: Output: M_J = (U_{j_0}, D_{j_0+1}, ··· , D_J) of size N_L = Σ_{r=1}^{N_R} #(L(Λ̃_{r,ε})).
```

7.3 Fully Adaptive Multiresolution Scheme

In this part, we recall some issues concerning the adaptive multiresolution scheme established in Chapter 4, and first presented in Chapter 3, and introduce the adaptive multiresolution

algorithm implemented in the MBARETE code.

7.3.1 Data Initialization

Let us suppose that initially we have the function $u(t, \mathrm{x})$, discretized on a uniform mesh corresponding to the grid level $j^0 \in [j_0, J]$: $\mathbf{U}_{j^0}$, and thus $\mathrm{L}(\Lambda_r)$ initially coincides with S_{j^0} and $N_{\mathrm{L}} = N_{\mathrm{R}} 2^{dj^0}$. The following Algorithm 7.7 allows us to transform an initial representation on a uniform mesh into a compressed adapted one, over a set of grids of level $j \in [j_0, J]$.

Algorithm 7.7 Data initialization.

1: **Input:** $\mathbf{U}_{j^0}$ of size 2^{dj^0}, given by cell-averaged values $u_{j^0,k}$ with $(j^0, k) \in S_{j^0}$.
2: **for** $j = j^0 \to J - 1$ **do**
3: Encode values by multiresolution transform $\mathcal{M} : \mathbf{U}_j \mapsto \mathbf{M}_j$ with Algorithm 7.1.
4: Threshold data $\mathbf{M}_j$ by $\mathcal{T}_{\Lambda_r}$ with Algorithm 7.3.
5: Grade tree $\Lambda_r \to \Lambda_{r,\varepsilon}$ with Algorithm 7.4.
6: Delete superfluous cells in $\Lambda_{r,\varepsilon}$ considering $\tilde{\Lambda}_{r,\varepsilon} \equiv \Lambda_{r,\varepsilon}$ in Algorithm 7.6.
7: Refine tree $\Lambda_{r,\varepsilon} \to \tilde{\Lambda}_{r,\varepsilon}$ by locally adding cells at next grid levels by the second refinement criterion in Algorithm 7.5.
8: Decode values by inverse multiresolution transform $\mathcal{M}^{-1} : \mathbf{M}_{j+1} \mapsto \mathbf{U}_{j+1}$ with Algorithm 7.2.
9: If possible, compute average values $u_{j+1,k}$ at grid level $j+1$ within $\mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$.
10: **end for**
11: **Output:** $\mathbf{U}_J$ of size $N_{\mathrm{L}} = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\tilde{\Lambda}_{r,\varepsilon}))$, given by cell-averaged values $u_{j,k}$ with $(j,k) \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.

7.3.2 Introduction of Phantom Cells

In Chapter 4 we have decided to use phantom cells for the flux evaluations, as detailed in [Mül03] and implemented in [RSTB03]. The adapted representation is always given on the leaves of the adapted tree $\mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$, according to the procedures previously introduced. The cell phantoms are temporarily added to $R(\tilde{\Lambda}_{r,\varepsilon})$ to compute the fluxes, and their values are computed by polynomial interpolations. However, no other multiresolution or time integration operation is performed on them. We represent by $P(\tilde{\Lambda}_{r,\varepsilon})$ the set of indices containing $R(\tilde{\Lambda}_{r,\varepsilon})$ and the phantom cells. These cells must be introduced contiguously to the leaves of the tree, and their number is fixed by the stencil of the flux computation scheme on the outer direction, denoted R^F. The number of phantom cells added to the current tree is given by: $N_{\mathrm{P}} = \#(\bigcup_{r\in[1,N_{\mathrm{R}}]} P(\tilde{\Lambda}_{r,\varepsilon}) \backslash R(\tilde{\Lambda}_{r,\varepsilon}))$. The Algorithm 7.8 illustrates the introduction of the phantom cells in the tree structure, and the computation of their cell values.

Algorithm 7.8 Introduction of the phantom cells: $R(\tilde{\Lambda}_{r,\varepsilon}) \to P(\tilde{\Lambda}_{r,\varepsilon})$.

1: **Input:** $\mathbf{M}_J = (\mathbf{U}_{j_0}, \mathbf{D}_{j_0+1}, \cdots, \mathbf{D}_J)$ of size $N_{\mathrm{L}} = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\tilde{\Lambda}_{r,\varepsilon}))$.
2: **for** $r = 1 \to N_{\mathrm{R}}$ **do**
3: **for** $j = J \to j_0 + 1$ **do**
4: **for** k s.t. $(j,k) \in S_j \bigcap \mathrm{L}(\Lambda_{r,\varepsilon})$ **do**
5: **if** $t_{j,k} =$ `.true.` **then**

6: **for** $i_d = 1 \to d$ **do**
7: **for** $l = -R^F \to R^F$ **do**
8: **if** $(j, k(i_d) + l) \notin R(\Lambda_{r,\varepsilon})$ **then**
9: $t_{j,k(i_d)+l} =$ `.true.`
10: Apply steps 6-17 of Algorithm 7.4 for graduation, with $j = j - 1$.
11: Set $d_{j,k(i_d)+l} = 0$.
12: $(j, k(i_d) + l) \in P(\Lambda_{r,\varepsilon})$.
13: **end if**
14: **end for**
15: **end for**
16: **end if**
17: **end for**
18: **end for**
19: **end for**
20: **Output:** $\mathbf{M}_J = (\mathbf{U}_{j_0}, \mathbf{D}_{j_0+1}, \cdots, \mathbf{D}_J)$ of size $N_{\mathrm{L}} = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\tilde{\Lambda}_{r,\varepsilon}))$, and N_{P} phantom cells included in $\bigcup_{r\in[1,N_{\mathrm{R}}]} P(\tilde{\Lambda}_{r,\varepsilon}) \backslash R(\tilde{\Lambda}_{r,\varepsilon})$.

Notice that to perform the interpolations and estimate the N_{P} cell values introduced in Algorithm 7.8, the graduation of the tree must take into account the presence of phantom cells. Therefore, if phantom cells are considered, the graduation Algorithm 7.4 might be applied with $\bar{M} = M + \lceil R^F/2 \rceil$ instead of M as grading parameter, and thus the graduation step in the Algorithm 7.8 would not be necessary.

7.3.3 Construction of the Tree-Structured Data

The following Algorithm 7.10 takes simultaneously into account the thresholding, the graduation, and the refinement of the data tree, considering the phantom cells and the previous discussions. Nevertheless, we first introduce the Algorithm 7.9 that allows us to locally graduate the tree for each cell added or kept in the tree.

Algorithm 7.9 Local graduation of the tree.

1: **Input:** $(j + 1, k') \in \Lambda_r$.
2: **for** $k' \in \mathrm{I}_k$ **do**
3: $t_{j+1,k'} =$ `.true.` $\Rightarrow (j + 1, k') \in R(\Lambda_r)$.
4: **end for**
5: **if** $j \geq j_0 + 1$ **then**
6: **for** $d_3 = -\bar{M} \to \bar{M}$ **do**
7: **for** $d_2 = -\bar{M} \to \bar{M}$ **do**
8: **for** $d_1 = -\bar{M} \to \bar{M}$ **do**
9: $t_{j,k_1+d_1,k_2+d_2,k_3+d_3} =$ `.true.`.
10: **if** $(j, k_1 + d_1, k_2 + d_2, k_3 + d_3) \notin R(\Lambda_r)$ **then**
11: Set $d_{j,k_1+d_1,k_2+d_2,k_3+d_3} = 0$.
12: **end if**
13: $(j, k_1 + d_1, k_2 + d_2, k_3 + d_3) \in R(\Lambda_r)$.
14: Grade locally for $(j, k_1 + d_1, k_2 + d_2, k_3 + d_3)$ with Algorithm 7.9.
15: **end for**
16: **end for**
17: **end for**

18: **end if**
19: **Output:** Graded structure for level $j \in [j_0, j+1]$, related to $(j+1, k') \in \Lambda_r$.

The first step considers that if

$$(j+1, k') \in \Lambda_r \Rightarrow \forall\ k' \in \mathrm{I}_k,\ (j+1, k') \in R(\Lambda_r), \tag{7.18}$$

as previously established for this implementation. Moreover, the grading parameter is now defined by

$$\bar{M} = M + \lceil \bar{k}/2 \rceil + \lceil R^F/2 \rceil, \tag{7.19}$$

taking into account the interpolation stencil, the first refinement criterion, and the introduction of phantom cells. For the diagonal cells with respect to (j,k), it suffices to consider $\bar{M} = M + \lceil R^F/2 \rceil$. Notice that condition (7.18) involves $2\lceil \bar{k}/2 \rceil$ more cells at each side and on each dimension for the Harten's first refinement criterion (3.103). The tree data structure is finally built by means of Algorithm 7.10, recalling that initially $t_{j,k} =$ `.false.` except for $j = j_0$, *i.e.*, $t_{j_0,k} =$ `.true.`.

Algorithm 7.10 Construction of the tree-structured data: $\Lambda_r \to \tilde{\Lambda}_{r,\varepsilon}$.

Input: $\mathbf{M}_J = (\mathbf{U}_{j_0}, \mathbf{D}_{j_0+1}, \cdots, \mathbf{D}_J)$ of size $N_{\mathrm{L}} = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\Lambda_r))$.
for $r = 1 \to N_{\mathrm{R}}$ **do**
  for $j = J \to j_0 + 1$ **do**
    for k s.t. $(j,k) \in S_j \bigcap \Lambda_r$ **do**
      if $|d_{j,k}| \geq \varepsilon_j$ **then**
        $t_{j,k} =$ `.true.` $\Rightarrow (j,l) \in \Lambda_r$.
        Grade locally for (j,k) with Algorithm 7.9.
        if $j = J$ **then** {First refinement criterion at highest level}
          for $i_d = 1 \to d$ **do**
            for $l = -\bar{k} \to \bar{k}$ **do**
              $t_{j,k(i_d)+l} =$ `.true.`
              if $(j, k(i_d)+l) \notin R(\Lambda_r)$ **then**
                Set $d_{j,k(i_d)+l} = 0$.
              end if
              $(j, k(i_d)+l) \in R(\Lambda_r)$.
              Grade locally for $(j, k(i_d)+l)$ with Algorithm 7.9.
            end for
          end for
        else {Second refinement criterion}
          for $k' \in \mathrm{I}_k$ **do**
            $t_{j+1,k'} =$ `.true.`
            if $(j+1, k') \notin R(\Lambda_r)$ **then**
              Set $d_{j+1,k'} = 0$.
            end if
            $(j+1, k') \in R(\Lambda_r)$.
            Grade locally for $(j+1, k')$ with Algorithm 7.9.
          end for
        end if
        if k s.t. $(j,k) \in S_j \bigcap \mathrm{L}(\Lambda_r)$ **then** {Introduction of phantom cells}
          for $i_d = 1 \to d$ **do**

for $l = -R^F \to R^F$ **do**
if $(j, k(i_d) + l) \notin R(\Lambda_r)$ **then**
$t_{j,k(i_d)+l} =$ `.true.`
Set $d_{j,k(i_d)+l} = 0$.
$(j, k(i_d) + l) \in P(\Lambda_r)$.
end if
end for
end for
end if
end if
end for
end for
end for
Output: $\Lambda_r \to \tilde{\Lambda}_{r,\varepsilon}$ with $\mathbf{M}_J = (\mathbf{U}_{j_0}, \mathbf{D}_{j_0+1}, \cdots, \mathbf{D}_J)$ of size $N_{\mathrm{L}} = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\tilde{\Lambda}_{r,\varepsilon}))$, and N_{P} phantom cells included in $\bigcup_{r\in[1,N_{\mathrm{R}}]} P(\tilde{\Lambda}_{r,\varepsilon}) \backslash R(\tilde{\Lambda}_{r,\varepsilon})$.

7.3.4 Adaptive Multiresolution Algorithm

Based on the previous algorithms, the complete adaptive multiresolution scheme is implemented following the Algorithm 7.11, that details the numerical strategy introduced in Chapter 4 concerning dynamic mesh adaptation.

Algorithm 7.11 Fully adaptive multiresolution scheme.

INITIALIZATION:
Define: set of grid levels $[j_0, J]$, initialization grid j^0, number of roots N_{R}, threshold value ε, dimension of the problem d, computational domain $\Omega \subset \mathbb{R}^d$, time domain of integration $t \in [t_0, T]$.
The coarsest grid is given by N_{R} cells $\Omega_{j_0,k}$, such that $\bigcup_{j=j_0} \Omega_{j,k} = \Omega$.
for $j = j_0 \to j^0 - 1$ **do** {Create initial set of grids}
Create successive dyadic partitions $\Omega_{j+1,k'}$ of $\Omega_{j,k}$, such that (7.3) is verified.
end for
Compute $\mathbf{U}_{j^0}$ at uniform grid level j^0, following (7.5).
Data initialization with Algorithm 7.7.
$t = t_0$ and $n = 0$.
Input: $\mathbf{U}_J^0$ of size $N_{\mathrm{L}}^0 = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\tilde{\Lambda}_{r,\varepsilon}^0))$, given by cell values $u_{j,k}^0$ such that $(j,k) \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon}^0)$ and $j \in [j_0, J]$.
LOOP IN TIME:
while $t \leq T$ **do**
Encode values by multiresolution transform $\mathcal{M} : \mathbf{U}_J^n \mapsto \mathbf{M}_J^n$ with Algorithm 7.1.
Estimate details according to (7.15) and (7.17).
Built the tree-structured data $\tilde{\Lambda}_{r,\varepsilon}^n \to \tilde{\Lambda}_{r,\varepsilon}^{n+1}$ with Algorithm 7.10, and N_{P}^n phantom cells included in $\bigcup_{r\in[1,N_{\mathrm{R}}]} P(\tilde{\Lambda}_{r,\varepsilon}^{n+1}) \backslash R(\tilde{\Lambda}_{r,\varepsilon}^{n+1})$.
Delete superfluous cells in $\tilde{\Lambda}_{r,\varepsilon}^{n+1}$ with Algorithm 7.6.
Decode values by inverse multiresolution transform $\mathcal{M}^{-1} : \mathbf{M}_J^n \mapsto \mathbf{U}_J^n$ with Algorithm 7.2, and compute N_{P}^n phantom value cells by polynomial interpolation (7.8), (7.10), or (7.11).

Time integration of the solution on the leaves of the trees $\bigcup_{r\in[1,N_{\rm R}]} {\rm L}(\tilde{\Lambda}_{r,\varepsilon}^{n+1})$: $\mathbf{U}_J^n \to \mathbf{U}_J^{n+1}$, $n \to n+1$, and $t \to t+\Delta t$.
Delete the $N_{\rm P}^n$ phantom cells.
$\mathbf{U}_J^{n+1}$ of size $N_{\rm L}^{n+1} = \sum_{r=1}^{N_{\rm R}} \#({\rm L}(\tilde{\Lambda}_{r,\varepsilon}^{n+1}))$, given by cell values $u_{j,k}^{n+1}$ such that $(j,k) \in \bigcup_{r\in[1,N_{\rm R}]} {\rm L}(\tilde{\Lambda}_{r,\varepsilon}^{n+1})$ and $j \in [j_0, J]$.
end while
$t = T$ and $n = n_T$.
Output: $\mathbf{U}_J^T$ of size $N_{\rm L}^{n_T} = \sum_{r=1}^{N_{\rm R}} \#({\rm L}(\tilde{\Lambda}_{r,\varepsilon}^{n_T}))$, given by cell values $u_{j,k}^{n_T}$ such that $(j,k) \in \bigcup_{r\in[1,N_{\rm R}]} {\rm L}(\tilde{\Lambda}_{r,\varepsilon}^{n_T})$ and $j \in [j_0, J]$.

Notice that creating the initial set of grids for $j \in [j_0, j^0]$ implies defining the entire set of cells $\bigcup_{r\in[1,N_{\rm R}]} R(\Lambda_r)$, instead of $\bigcup_{r\in[1,N_{\rm R}]} \Lambda_r$ of size $N_{\rm L}$, according to the definition of ${\rm L}(\Lambda_r)$ introduced in Chapter 3. This means that we are creating more cells than needed since only $N_{\rm L}$ cells are needed to perform all the previous operations in the wavelet decomposition space, described by algorithms 7.1-7.10. Nevertheless, the numerical code, briefly depicted in the following, considers N_Λ cells composed of $N_{\rm L}$ leaves and the cells lying at the coarser grid levels, *i.e.*, $N_\Lambda = \sum_{r=1}^{N_{\rm R}} \#(R(\Lambda_r))$. The number of cells N_Λ is given by (4.28) when all cells are present in the different grids. Even though these underlying cells are not needed in the previous computations, the inclusion of all cells of the adapted tree eases the programming of the data structure in terms of navigation and location of cells within the tree.

7.4 Basic Code Implementation

The implemented code represents the tree-structured data as a set of cells linked by pointers in a Fortran 90/95 programming language. We present in this part the basic elements of the code for a straightforward implementation of multiresolution techniques as illustrated in [TD11]. See, *e.g.*, [VM99, BMV09] for details on optimization of code and data structures for these applications.

7.4.1 Data Structure

The cell representation is defined as a derived type in a general Fortran environment with the following components stocked within, as an illustration:

```
type cell
```

- A first flag analogous to $t_{j,k}$, to indicate whether cells are kept, added, or discarded in $\tilde{\Lambda}_{r,\varepsilon}^{n+1}$. The second optional flag allows us to identify phantom cells included in $P(\tilde{\Lambda}_{r,\varepsilon}^{n+1}) \backslash R(\tilde{\Lambda}_{r,\varepsilon}^{n+1})$, if necessary:
  `logical :: tree` $\Longrightarrow$ `.true.` if it must belong to the tree, and `.false.` otherwise.
  `logical :: leave` $\Longrightarrow$ `.true.` if it is a leaf, and `.false.` if it is a phantom leaf.

- The grid level j of the cell and its position (index) k on the corresponding grid for d dimensions. The couple (j,k) allows us to define an unique position of each cell into the whole set of nested grids:

```
integer :: level
integer, dimension(d) :: index
```

- Spatial coordinates of the center of the cell and spatial discretization. These values can be also calculated at any time knowing the grid level, the index, and the size of the computational domain:
```
double precision, dimension(d) ::  x
double precision, dimension(d) ::  dx
```

- Local threshold value ε_j. It depends on ε and the grid level j thought (7.14), and it can be also calculated on the fly:
```
double precision ::  epsilon
```

- Array U of size m to save the m variables $u_{j,k'}^{(i)}$, and the corresponding detail $|d_{j,k'}|$ estimated according to (7.15) and (7.17). Depending on the case, $2^d - 1$ child-cells of the same parent-cell will save the details $d_{j,k'}^{(i)}$ instead of $u_{j,k'}^{(i)}$, while the variable values of the parent-cell $u_{j-1,k}^{(i)}$ are saved by the remaining child-cell, during the multiresolution operations in algorithms 7.1 and 7.2:
```
double precision, dimension(m) ::  U
double precision ::  detail
```

- Right/left flux on each dimension if flux evaluations are necessary, and need to be saved throughout the time integration step:
```
double precision, dimension(m,d) ::  flux
```

- Pointers on the parent cell, its child cells, and its neighbors on each dimension:
```
type(cell), pointer ::  parent
type(child_cell), dimension(2**d) ::  children
type(neighbor_cell), dimension(d) ::  neighbors
```

```
end type cell
```

We define some new derived type structures of Fortran 90/95, called `child_cell` and `neighbor_cell`, to generate the array of pointers:

```
type child_cell
```

- `type(cell), pointer ::  child`

```
end type child_cell
```

```
type neighbor_cell
```

- `type(cell), pointer ::  previous`
- `type(cell), pointer ::  next`

```
end type neighbor_cell
```

Other computing parameters or local physical properties might be also saved into the cell-type. However, it is important to measure the potential benefits of whether saving or computing the various parameters or variables. A compromise must be made considering the availability of computing resources and the type of application.

7.4.2 General Code Structure

In a Fortran 90/95 environment, a pointer is just an alias to the target. Nevertheless, we take advantage of the fact that each pointer has a different state, depending on whether it is associated with another object. We work then with cells that lie at different grids and that are not necessarily arranged in a contiguous way. Hence, we must conceive the mechanisms to navigate through the tree structure. In this implementation, we adopt a recursive strategy in which one moves from one cell to another passing by the child-cell of the first, and by the consecutive children, until one gets to the desired cell. At each step, the state of the pointers tells us whether the target exists. In this recursive way, we are able to locate leaves or any cell, and the same kind of procedure is conducted in the opposite situation from leaves towards roots when necessary. Pointers to neighbors as well as other flags or indicators are not strictly necessary but eases considerably the searching process for certain routines. Notice that this is a fully local approach because we never consider more than one cell at the same time.
In order to illustrate this, let us consider a set of routines which are successively executed in the main program. For instance, the `evaluation` routine is called in the main program and involves computations at the leaves of the tree. The tree structure was previously built, from the roots, by similar navigating procedures. Then, each step is performed into the main program as follows:

```
do i3 = 1, N_R(3)
 do i2 = 1, N_R(2)
  do i1 = 1, N_R(1)
   current => root(i1, i2, i3)
   call evaluation(current)
  enddo
 enddo
enddo
```

where

```
type(child_cell), dimension(N_R(1),N_R(2),N_R(3)) ::  root
type(cell) ::  current
```

Into a general routine, in this example `evaluation`, the recursive scheme to evaluate a function only on the leaves, reads as follows:

```
recursive subroutine evaluation(current)
if (.not.associated(current%children(1))) then
   !  we are on a leaf
   current%u = ....
else
  do i = 1, 2**d
    call evaluation(current%children(i))
  enddo
endif
end subroutine evaluation
```

In this recursive way, we are able to locate the leaves by considering the state of the pointers that link cells at different levels. The same kind of procedure is conducted in the opposite case,

from the leaves towards the roots. This is illustrated in the following example where some value is known on the leaves and we would like to propagate the information to the roots:

```
recursive subroutine evaluation(current)
if (.not.associated(current%children(1))) then
   !  we are on a leaf, nothing is done
else
  do i = 1, 2**d
    call evaluation(current%children(i))
  enddo
  current%u = ....
endif
end subroutine evaluation
```

Other parameters as the indices of the cells or the pointers to the neighbors can be taken into consideration to accelerate the research, depending on the routine.

Chapter 8

Description of Time Operator Splitting in the MBARETE Code

In this chapter, we will discuss some particular features of the time operator splitting technique implemented in the MBARETE code. In particular, a different data structure is considered for the time integration operations. The reaction and diffusion algorithmic implementations will be explained in detail, and constitute the two possible configurations for local integration either leaf by leaf, or over the whole set of leaves of the tree structure, where the latter set corresponds to an adapted grid representation. Further extensions follow similar patterns.

8.1 Data Representation for Time Integration

We have seen in the previous Chapter 7 that a tree-structured data was implemented in the code to represent the dynamically adapted grid and the underlying meshes. Nevertheless, the time integration of the solution takes place only at the leaves of the tree, *i.e.,* the resulting adapted mesh. In order to avoid navigating the tree structure to locate the cells at the adapted mesh, we define $\Phi_{\mathrm{L}} = (\phi_i)_{i=1,\dots,N_{\mathrm{L}}}$, as the set of leaves of the tree representation:

$$\phi_i^n \longleftrightarrow \Omega_{j,k}, \text{ s.t. } (j,k) \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon}^n), \qquad i = 1,\dots,N_{\mathrm{L}}, \tag{8.1}$$

for a set of grid levels with $j \in [j_0, J]$. For the uniform mesh case, S_J is equivalent to $\bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon}^n)$. We define also $|\phi_i^n| := i_{\mathrm{L}} = (j,k)$, where i_{L} identifies each leaf-cell (j,k) within the set Φ_{L}^n. Algorithm 8.1 illustrates the definition of Φ_{L}^n given by (8.1).

Algorithm 8.1 Construction of the set of leaves of $\tilde{\Lambda}_{r,\varepsilon}$: Φ_{L}.

1: **Input:** $\mathbf{U}_J$ of size $N_{\mathrm{L}} = \sum_{r=1}^{N_{\mathrm{R}}} \#(\mathrm{L}(\tilde{\Lambda}_{r,\varepsilon}))$, given by cell values $u_{j,k}$ such that $(j,k) \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.
2: $i = 1$.
3: **for** $r = 1 \to N_{\mathrm{R}}$ **do**
4: **for** $j = j_0 \to J$ **do**
5: **for** k s.t. $(j,k) \in S_j \bigcap \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ **do**
6: $i_{\mathrm{L}} = (j,k)$.
7: $\Omega_{j,k} \longrightarrow \phi_i$.
8: $i = i + 1$.
9: **end for**

10: **end for**
11: **end for**
12: **Output:** $\mathbf{U}_J$ of size $N_{\mathrm{L}} = \#(\Phi_{\mathrm{L}})$, given by cell values $u_{i_{\mathrm{L}}}$ with $i_{\mathrm{L}} \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.

Notice that the introduction of the set Φ_{L} would not be necessary if we considered the trees Λ_r instead of $R(\Lambda_r)$, as tree representation in the code. The Algorithm 8.1 is performed during the construction of the tree structure in the global Algorithm 7.11.

Basic Code Implementation

The implementation of the set Φ_{L} is straightforward with the data structure considered for the multiresolution technique in Chapter 7. For instance, it suffices to consider:

```
type(child_cell), dimension(N_L) ::  PHI
```

with the same cell type as for the leaves. Additionally, the recursive operations are replaced by a single loop over the leaves of the tree. For instance:

```
subroutine evaluation
do i = 1, N_L
 PHI(i)%u = ....
enddo
end subroutine evaluation
```

The same information per cell is often not needed for both multiresolution and time integration procedures. Therefore, a better solution as finally implemented in the code, defines a new leaf type, similar to the previous cell type, such that the multiresolution or integration parameters are, respectively, saved in the cell or leaf types, depending on the status of the cell.

8.2 Time Operator Splitting Scheme

Let us now illustrate with more details the construction of the operator splitting technique introduced in Chapter 4. One of the key point of such an approach is that the implementation is given by an independent assemblage of the dedicated solvers for each split subproblem. In what follows, we illustrate the general procedure for the solution of the reaction and the diffusion problems, which remain similar for other time integration solvers different from the ones considered in this application.

8.2.1 Time Integration of the Reaction

Let us consider the vector function $U : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^m$ given by

$$U(t, \mathrm{x}) = (u^{(1)}(t, \mathrm{x}), u^{(2)}(t, \mathrm{x}), \ldots, u^{(m)}(t, \mathrm{x}))^T, \tag{8.2}$$

for m variables, in the general problem (1.52). The Algorithm 8.2 schematically illustrates the time integration of the associated reaction subproblem with Radau5 [HW96], over a time step Δt. The Radau5 solver is used with the parameters given by default (see [HW96]), except for the initial time step Δt_R, and the absolute and relative tolerances, *Atol* and *Rtol*, previously mentioned in Chapter 2. In particular, the cell (or leaf) type saves the last time step used in the previous time iteration on each leaf: $\Delta t_{R,i_{\mathrm{L}}}$, if the leaf was present during the iteration. For

the first time iteration, all the $\Delta t_{R,i_{\rm L}}$ are set equal to Δt. Concerning the tolerance parameters, they are in general fixed as $Atol = Rtol = \eta_{Radau5}$, as previously considered in the numerical illustrations of chapters 4 and 5. Radau5 is a fully implicit Runge-Kutta method as detailed in Chapter 2, and it often requires an important memory space to perform the various associated computations. For the cases that we are considering, the memory requirements are mainly set by the work array WRK_1 of size L_1:

$$L_1 = 4 \times m \times m + 12 \times m + 20, \tag{8.3}$$

which is rather minimal, compared with a coupled resolution of the problem with $m \times N_{\rm L}$ variables. Other dedicated solvers for stiff ODEs might be implemented in a similar way.

Algorithm 8.2 Integration of the reaction problem with Radau5.

1: **Input:** $\mathbf{U}_J$ of size $(m, N_{\rm L}) = (m, \#(\Phi_{\rm L}))$, given by m arrays $\mathbf{U}_J^{(i)}$ with cell values $u_{i_{\rm L}}^{(i)}$, $i = 1, \ldots, m$, $i_{\rm L} \in \bigcup_{r\in[1,N_{\rm R}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.
2: **for** $i_{\rm L} = 1 \to N_{\rm L}$ **do**
3: Set accuracy tolerances: $Atol = Rtol = \eta_{Radau5}$.
4: Guess initial time step integration: $\Delta t_R^0 = \max(\Delta t_{R,i_{\rm L}}, \Delta t)$.
5: Time integration of $(u_{i_{\rm L}}^{(i)})_{i=1}^m$ with Radau5: $t \to t + \Delta t$ with Δt_R^0, $Atol$, $Rtol$.
6: Update solution: $u_{i_{\rm L}}^{(i)} \to \bar{u}_{i_{\rm L}}^{(i)}$, $i = 1, \ldots, m$.
7: Update leaf time step: $\Delta t_{R,i_{\rm L}} = \Delta t_R$.
8: **end for**
9: **Output:** $\bar{\mathbf{U}}_J$ of size $(m, N_{\rm L}) = (m, \#(\Phi_{\rm L}))$, given by m arrays $\bar{\mathbf{U}}_J^{(i)}$ with cell values $\bar{u}_{i_{\rm L}}^{(i)}$, $i = 1, \ldots, m$, $i_{\rm L} \in \bigcup_{r\in[1,N_{\rm R}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.

The Jacobians must be computed by Radau5 either numerically according to (2.76), or analytically following a routine supplied by the user. The user must also supply the routine to locally evaluate the function $F : \mathbb{R}^m \to \mathbb{R}^m$, for instance, in (1.52), for each set $(u_{i_{\rm L}}^{(i)})_{i=1}^m$, within the time integration process. Let us recall that Radau5 will adapt its integration time steps Δt_R from the initial guess Δt_R^0, according to $Atol$ and $Rtol$. The last time step will be then saved as $\Delta t_{R,i_{\rm L}}$ for the next iteration. The previous algorithm can be easily parallelized by distributing the loop in $i_{\rm L}$ among $N_{\rm proc}$ processors, where each $i_{\rm L}$ represents an independent system of ODEs. In shared memory architectures, the latter procedure can be implemented, for instance, with the OpenMP library, with neither synchronization stages nor data exchange among the processors. The load balancing is hence practically optimal for this configuration. The work array WRK_1 must be this time of size $(N_{\rm proc}, L_1)$.

8.2.2 Time Integration of the Diffusion

The time integration of the diffusion problem, for instance, (4.2), is performed by the ROCK4 solver [Abd02]. The Algorithm 8.3 illustrates this integration over a time step Δt. Since the ROCK method is based on an explicit Runge-Kutta scheme, there are much less parameters to define in advance. In this implementation, the solution is progressively advanced in time with the same time step Δt_D, uniformly in space. These time steps are adapted according also to absolute and relative tolerances, $Atol$ and $Rtol$, fixed in this case as $Atol = Rtol = \eta_{ROCK4}$. As for the reaction integration, the last time step is saved for the next time iteration as $\Delta t_{D,\rm ite}$. For the first iteration, $\Delta t_{D,0}$ is taken equal to Δt. The spectral radius needed to estimate the

number of inner stages that ensures stability according to (4.12), is generally computed by ROCK4, although it can be also approximated by (2.18) for problems with linear diffusion. In the latter case, if we set ROCK4 to estimate the spectral radius, it is also set to do it only once during the global time step Δt. In a general configuration given, for instance, by the diffusion problem in (1.52), we have to simultaneously solve $m \times N_{\mathrm{L}}$ variables. The memory requirements of the ROCK4 solver are established in this case by the work array WRK_2 of size L_2:

$$L_2 = 8 \times m \times N_{\mathrm{L}}, \tag{8.4}$$

in the general case in which the spectral radius is computed by ROCK4. Otherwise, $L_2 = 7 \times m \times N_{\mathrm{L}}$.

Algorithm 8.3 Integration of the diffusion problem with ROCK4.

1: **Input:** $\mathbf{U}_J$ of size $(m, N_{\mathrm{L}}) = (m, \#(\Phi_{\mathrm{L}}))$, given by m arrays $\mathbf{U}_J^{(i)}$ with cell values $u_{i_{\mathrm{L}}}^{(i)}$, $i = 1, \ldots, m$, $i_{\mathrm{L}} \in \bigcup_{r \in [1, N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.
2: Set accuracy tolerances: $Atol = Rtol = \eta_{ROCK4}$.
3: Guess initial time step integration: $\Delta t_D^0 = \max(\Delta t_{D,\mathrm{ite}}, \Delta t)$.
4: Time integration of $\mathbf{U}_J$ with ROCK4: $t \to t + \Delta t$ with Δt_D^0, $Atol$, $Rtol$.
5: Update solution: $\mathbf{U}_J \to \bar{\mathbf{U}}_J$.
6: Update diffusion time step: $\Delta t_{D,\mathrm{ite}+1} = \Delta t_D$.
7: **Output:** $\bar{\mathbf{U}}_J$ of size $(m, N_{\mathrm{L}}) = (m, \#(\Phi_{\mathrm{L}}))$, given by m arrays $\bar{\mathbf{U}}_J^{(i)}$ with cell values $\bar{u}_{i_{\mathrm{L}}}^{(i)}$, $i = 1, \ldots, m$, $i_{\mathrm{L}} \in \bigcup_{r \in [1, N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.

In this implementation we have adopted a finite volume discretization approach for structured Cartesian meshes. The evaluation of the diffusion term: $\partial_{\mathrm{x}} \cdot (D(U) \partial_{\mathrm{x}} U)$ involves the computation of fluxes through the boundaries $\partial\Omega_{i_{\mathrm{L}}}$ of each leaf-cell $\Omega_{i_{\mathrm{L}}}$: $\mathrm{F}_{i_d}^+$ and $\mathrm{F}_{i_d}^-$, which account for the fluxes on the right and on the left of the cell in each direction, $i_d = 1, \ldots, d$. These fluxes are evaluated at the interfaces of the cell, as well as the diffusion coefficients inside, on a locally uniform mesh with the introduction of the phantom cells, as detailed in Chapter 7. The Algorithm 8.4 illustrates the steps to evaluate the fluxes within ROCK4, *i.e.*, at each internal stage s computed by (4.12). The latter procedure is general for any explicit solver or other transport operator like convective ones, in the case where local structured meshes are considered, as mentioned in Chapter 3. In this implementation a classical centered second order space discretization was implemented for the diffusion problem.

Algorithm 8.4 Flux evaluation with phantom cells.

1: **Input:** $\mathbf{U}_J$ of size $(m, N_{\mathrm{L}}) = (m, \#(\Phi_{\mathrm{L}}))$, given by m arrays $\mathbf{U}_J^{(i)}$ with cell values $u_{i_{\mathrm{L}}}^{(i)}$, $i = 1, \ldots, m$, $i_{\mathrm{L}} \in \bigcup_{r \in [1, N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.
2: **for** $r = 1 \to N_{\mathrm{R}}$ **do**
3: **for** $j = J - 1 \to j_0 - 1$ **do**
4: **for** k s.t. $(j, k) \in S_j \bigcap \tilde{\Lambda}_{r,\varepsilon}$ **do**
5: **for** $k' \in \mathrm{I}_k$ **do**
6: **if** $j \geq j_0$ **and** $(j+1, k') \in P(\tilde{\Lambda}_{r,\varepsilon}) \backslash R(\tilde{\Lambda}_{r,\varepsilon})$ **then** {Update phantom cells}
7: Update all $(u_{j,k(i_d)+l}^{(i)})_{i=1}^m$ with $i_d = 1, \ldots, d$, $|l| \leq M$, and such that $(j, k(i_d) + l) \notin \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ at grid level j, from $(u_{j+1,k''}^{(i)})_{i=1}^m$ with $k'' \in \mathrm{I}_{k(i_d)+l}$ at level $j+1$, by using the projection operator P_{j-1}^j (7.6).

Compute $(u^{(i)}_{j+1,k'})_{i=1}^m = (\hat{u}^{(i)}_{j+1,k'})_{i=1}^m$ by polynomial interpolation (7.8), (7.10), or (7.11).
end if
end for
end for
for k' s.t. $(j+1,k') \in S_{j+1} \bigcap \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ **do**
for $i_d = 1 \to d$ **do** {Compute fluxes}
Compute $\mathrm{F}^{+,(i)}_{i_d}$ and $\mathrm{F}^{-,(i)}_{i_d}$, for $i = 1, \ldots, m$ at grid level $j+1$.
end for
end for
end for
end for
Output: Numerical fluxes $\mathrm{F}^{+,(i)}_{i_d}$ and $\mathrm{F}^{-,(i)}_{i_d}$, $i = 1, \ldots, m$, at interfaces $\partial\Omega_{i_{\mathrm{L}}}$ with $i_{\mathrm{L}} \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.

The flux evaluations basically involves two stages. The first computations are needed to update the interpolation stencils of the phantom cells, since the time evolution is not performed on them. In the second part, we evaluate the fluxes direction by direction, as in a standard uniform mesh configuration. It can be inferred that a locally unstructured approach for the flux computations, *i.e.,* without phantom cells, will surely improve the global performance of the strategy, although the accuracy orders of the spatial discretizations are locally reduced. We can also conclude that a low number of stages s for the stabilized ROCK method will be advisable, and very convenient for the global performance of the method. For high number of stages, ROCK will however keep the advantage of being more stable than standard explicit Runge-Kutta methods, with less memory requirements than an implicit solver. Further developments and improvements on these issues constitute a continuous research in the field.
Alternatively, instead of Algorithm 8.4, another approach which was also implemented and tested in the code, considers the construction of the discretized diffusion operator. The idea is to accelerate the computations by performing matrix-vector products at each ROCK stage taking into consideration that the discretized operator results often in a sparse matrix representation, specially appropriate for linear diffusion problems as illustrated in [DDD+12]. In this case this matrix needs to be built only once for uniform meshes and once per time iteration for adapted grids. From the numerical tests performed with this approach, we can conclude that the sparsity of the matrix is importantly reduced with the inclusion of the interpolation stencils of the phantoms, specially in 3D. The phantom cells are not explicitly computed but represented as local functions of the leaf-cells through the combined projection/prediction operations used in Algorithm 8.4. These facts and the additional cost of building the matrix for a highly irregular configuration in terms of grid distribution, involve less important gains in CPU time. A comparison of the performance of both schemes reveals an important problem dependency. Alternatively, the Algorithm 8.4 has the advantage of being simpler and suitable for a larger range of problems from linear diffusion as in (1.53), to diffusion with variable dependent transport parameters like in (1.52).
For linear diffusion problems, the Algorithm 8.3 can be also parallelized by considering m independent integrations of $\mathbf{U}^{(i)}_J$ with $i = 1, \ldots, m$. The same follows for the flux evaluations in Algorithm 8.4. The parallelization of the loop in m implies a reduction of the work array WRK_2, if N_{proc} is smaller than m, because (8.4) becomes $L_2 = (N_{\mathrm{proc}}, 8 \times N_{\mathrm{L}})$, and the elimination of all loops in m for both algorithms which results in gains of CPU time.

8.2.3 Time Operator Splitting Algorithm

With the definition of the previous algorithms, the Algorithm 8.5 illustrates the implementation of the splitting technique during a splitting time step Δt, for the Strang scheme $\mathcal{S}_2^{\Delta t}$ in (4.5) and (5.5).

Algorithm 8.5 Time operation splitting scheme: $\mathcal{S}_2^{\Delta t}$.

1: **Input:** Solution $\mathbf{U}_J$ at $t = t$, of size $(m, N_{\mathrm{L}}) = (m, \#(\Phi_{\mathrm{L}}))$, given by m arrays $\mathbf{U}_J^{(i)}$ with cell values $u_{i_{\mathrm{L}}}^{(i)}$, $i = 1, \ldots, m$, $i_{\mathrm{L}} \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.
2: Set accuracy tolerances for split reaction and diffusion time integration: η_{Radau5} and η_{ROCK4}.
3: For $t \to t + \Delta t/2$: time integration of $\mathbf{U}_J$ for the reaction problem with Radau5 by Algorithm 8.2.
4: Update solution: $\bar{\mathbf{U}}_J \to \mathbf{U}_J$.
5: For $t \to t+\Delta t$: time integration of $\mathbf{U}_J$ for the diffusion problem with ROCK4 by Algorithm 8.3. At each internal stage s of ROCK4, the fluxes are evaluated with Algorithm 8.4.
6: Update solution: $\bar{\mathbf{U}}_J \to \mathbf{U}_J$.
7: For $t \to t + \Delta t/2$: time integration of $\mathbf{U}_J$ for the reaction problem with Radau5 by Algorithm 8.2.
8: Update solution: $\bar{\mathbf{U}}_J \to \mathbf{U}_J$.
9: Update time: $t \to t + \Delta t$.
10: **Output:** Strang solution $\mathbf{U}_J$ at $t = t + \Delta t$, of size $(m, N_{\mathrm{L}}) = (m, \#(\Phi_{\mathrm{L}}))$, given by m arrays $\mathbf{U}_J^{(i)}$ with cell values $u_{i_{\mathrm{L}}}^{(i)}$, $i = 1, \ldots, m$, $i_{\mathrm{L}} \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.

The splitting time integration scheme given by Algorithm 8.5 is hence introduced as the time evolution operator from t to $t+\Delta t$, in the global multiresolution scheme defined by Algorithm 7.11. The choice of the splitting time step Δt has been discussed in details in Chapter 4.

8.3 Time Adaptive Splitting Scheme

In this part, we detail the inclusion of a dynamic time stepping technique in the previous splitting strategy, based on the method introduced in Chapter 5.

8.3.1 Implementation of a Splitting Embedded Method

Let us consider again the Strang scheme $\mathcal{S}_2^{\Delta t}$ in (4.5) and (5.5), defined by Algorithm 8.5. The following Algorithm 8.6 considers the inclusion of the embedded shifted Strang formula (5.7): $\mathcal{S}_{2,\delta}^{\Delta t}$. We recall that the shifting time parameter δ verifies $\delta \leq |\delta_{max}| < 1/2$, and that the shifted operator will be applied on $l \leq m$ variables in the general problem (1.52): $\tilde{U} \in \mathbb{R}^l$, where Θ_l denotes the set of indices of these variables.

Algorithm 8.6 Standard and embedded splitting schemes: $\mathcal{S}_2^{\Delta t}$ and $\mathcal{S}_{2,\delta}^{\Delta t}$.

1: **Input:** $\mathbf{U}_J$ of size $(m, N_{\mathrm{L}}) = (m, \#(\Phi_{\mathrm{L}}))$, given by m arrays $\mathbf{U}_J^{(i)}$ with cell values $u_{i_{\mathrm{L}}}^{(i)}$, $i = 1, \ldots, m$, $i_{\mathrm{L}} \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.
2: For given accuracy tolerances: η_{Radau5} and η_{ROCK4}.
3: For $t \to t + \Delta t/2$: time integration of $\mathbf{U}_J$ for the reaction problem with Radau5 by Algorithm 8.2.

4: New Strang solution: $\bar{\mathbf{U}}_J \to \mathbf{U}_J^{\text{new}}$.
5: For $t \to t+\delta\Delta t$: time integration of l arrays $\mathbf{U}_J^{\text{new},(i)}$ with $i \in \Theta_l$, for the reaction problem with Radau5 by Algorithm 8.2.
6: New shifted Strang solution: $\bar{\mathbf{U}}_J^{(i)} \to \tilde{\mathbf{U}}_J^{(i)}$ for $i \in \Theta_l$.
7: For $t \to t+\Delta t$: time integration of $\mathbf{U}_J^\star = (\mathbf{U}_J^{\text{new}}, \tilde{\mathbf{U}}_J)^T$ of size $(m+l, N_{\text{L}})$, for the diffusion problem with ROCK4 by Algorithm 8.3. At each internal stage s of ROCK4, the fluxes are evaluated with Algorithm 8.4.
8: Update Strang solution: $\bar{\mathbf{U}}_J^{(i)} \to \mathbf{U}_J^{\text{new},(i)}$ for $i = 1, \ldots, m$.
9: Update shifted Strang solution: $\bar{\mathbf{U}}_J^{(m+i)} \to \tilde{\mathbf{U}}_J^{(i)}$ for $i = 1, \ldots, l$.
10: For $t \to t+(1/2-\delta)\Delta t$: time integration of $\mathbf{U}_J^\star = (\mathbf{U}_J^{\text{new}}, \tilde{\mathbf{U}}_J)^T$ of size $(m+l, N_{\text{L}})$ for the reaction problem with Radau5 by Algorithm 8.2.
11: Update Strang solution: $\bar{\mathbf{U}}_J^{(i)} \to \mathbf{U}_J^{\text{new},(i)}$ for $i = 1, \ldots, m$.
12: Update shifted Strang solution: $\bar{\mathbf{U}}_J^{(m+i)} \to \tilde{\mathbf{U}}_J^{(i)}$ for $i = 1, \ldots, l$.
13: For $t \to t+\delta\Delta t$: time integration of $\mathbf{U}_J^{\text{new}}$, for the reaction problem with Radau5 by Algorithm 8.2.
14: Update Strang solution: $\bar{\mathbf{U}}_J \to \mathbf{U}_J^{\text{new}}$.
15: **Output:** Initial $\mathbf{U}_J$ and Strang solution $\mathbf{U}_J^{\text{new}}$ of size $(m, N_{\text{L}}) = (m, \#(\Phi_{\text{L}}))$, given by m arrays $\mathbf{U}_J^{(i)}$ and $\mathbf{U}_J^{\text{new},(i)}$, with cell values $u_{i_{\text{L}}}^{(i)}$ and $u_{i_{\text{L}}}^{\text{new},(i)}$, $i = 1, \ldots, m$; and shifted Strang solution $\tilde{\mathbf{U}}_J$ of size $(l, N_{\text{L}}) = (l, \#(\Phi_{\text{L}}))$, given by l arrays $\tilde{\mathbf{U}}_J^{(i)}$, with cell values $\tilde{u}_{i_{\text{L}}}^{(i)}$, $i \in \Theta_l$; $i_{\text{L}} \in \bigcup_{r\in[1,N_{\text{R}}]} \text{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.

Let us make the following remarks concerning the latter algorithm:

- For the first reaction step: the time integration of the initial solution $\mathbf{U}_J$ is performed in practice for $t \to t + \max(1/2+\delta, 1/2) \times \Delta t$ using the Radau5 routine `SOLOUT` (see [HW96]) to extract the intermediate Strang solution at $t = \Delta t/2$ if $\delta > 0$, and the shifted Strang solution at $t = (1/2+\delta)\Delta t$ if $\delta < 0$.

- For the diffusion step: the advantage of integrating simultaneously both solutions $\mathbf{U}_J^\star = (\mathbf{U}_J^{\text{new}}, \tilde{\mathbf{U}}_J)^T$, is that the navigating procedure in Algorithm 8.4 to update the phantom-cells is performed only once for all $m+l$ variables. Nevertheless, the memory requirements are increased into (8.4): $L_2 = 8 \times (m+l) \times N_{\text{L}}$. In this case a parallel computation variable by variable is advisable to overcome memory restrictions whenever $m+l$ is larger than N_{proc}.

- For the second reaction step: the time integration of the combined solutions $\mathbf{U}_J^\star$ is also performed for $t \to t + \max(1/2-\delta, 1/2) \times \Delta t$, with the intermediate Strang solution extracted at $t = \Delta t/2$ if $\delta < 0$, and the shifted Strang solution at $t = (1/2-\delta)\Delta t$ if $\delta > 0$. In order to accelerate the computations, the Jacobian associated with $\mathbf{F}(\mathbf{U}_J^\star) = \mathbf{F}\left((\mathbf{U}_J^{\text{new}}, \tilde{\mathbf{U}}_J)^T\right)$ at each leaf i_{L}, can be also approximated by

$$\partial_{\mathbf{U}_J^\star}\mathbf{F} \approx \begin{pmatrix} \partial_{\mathbf{U}_J^{\text{new}}}\mathbf{F} & \\ & \partial_{\mathbf{U}_J^{\text{new},(i)}|_{i=\in\Theta_l}}\mathbf{F} \end{pmatrix} \tag{8.5}$$

  when δ is sufficiently small. This involves only the computation of the Jacobian associated with $\mathbf{U}_J^{\text{new}}$, as long as the l variables are decoupled in $F(U)$ from the remaining $m-l$ variables. If $l = m$, the approximation (8.5) follows naturally.

- Another alternative to accelerate computations considers larger tolerances η_{Radau5} and η_{ROCK4} during the shifted Strang computations, for the steps that are not used for the second order Strang solution.
- The global procedure needs to save in memory two arrays of size (m, N_{L}), and one of (l, N_{L}).

8.3.2 Computation of Splitting Time Step

The splitting time step is adapted at each time iteration in order to ensure computations within a prescribed accuracy tolerance η according to (5.11). First, the Algorithm 8.7 defines the L^2-error computation between two arrays, $\mathbf{A}_J$ and $\mathbf{B}_J$, represented on the same adapted grid, considered throughout this work.

Algorithm 8.7 Numerical estimate of the error: Δ.

1: **Input:** $\mathbf{A}_J$ and $\mathbf{B}_J$ of size $(l, N_{\mathrm{L}}) = (l, \#(\Phi_{\mathrm{L}}))$, given by l arrays $\mathbf{A}_J^{(i)}$ and $\mathbf{B}_J^{(i)}$ with cell values $a_{i_{\mathrm{L}}}^{(i)}$ and $b_{i_{\mathrm{L}}}^{(i)}$, $i = 1, \ldots, l$, $i_{\mathrm{L}} \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.
2: **for** $i = 1 \to l$ **do**
3: Compute

$$\Delta^{(i)} = |\Omega|^{-1/2} \left(\sum_{i_{\mathrm{L}}=1}^{N_{\mathrm{L}}} (a_{i_{\mathrm{L}}}^{(i)} - b_{i_{\mathrm{L}}}^{(i)})^2 |\Omega_{i_{\mathrm{L}}}| \right)^{-1/2} \tag{8.6}$$

and define $a_{\max}^{(i)} = \max_{i_{\mathrm{L}}=1,\ldots,N_{\mathrm{L}}}(|a_{i_{\mathrm{L}}}^{(i)}|)$.
4: **end for**
5: The numerical error Δ is finally defined by

$$\Delta = \max_{i=1,\ldots,l} \left(\frac{\Delta^{(i)}}{a_{\max}^{(i)}} \right). \tag{8.7}$$

6: **Output:** Numerical estimate of the error Δ between $\mathbf{A}_J$ and $\mathbf{B}_J$.

The Algorithm 8.8 establishes the general procedure to compute the splitting time step Δt, with security factor $0 < \upsilon \leq 1$ close to 1, based on the Strang and shifted Strang solutions issued from Algorithm 8.6.

Algorithm 8.8 Computation of the new splitting time step: $\Delta t \to \Delta t^{\mathrm{new}}$.

1: **Input:** Strang solution $\mathbf{U}_J^{\mathrm{new}}$ of size $(m, N_{\mathrm{L}}) = (m, \#(\Phi_{\mathrm{L}}))$, given by m arrays $\mathbf{U}_J^{\mathrm{new},(i)}$, with cell values $u_{i_{\mathrm{L}}}^{\mathrm{new},(i)}$, $i = 1, \ldots, m$; and shifted Strang solution $\tilde{\mathbf{U}}_J$ of size $(l, N_{\mathrm{L}}) = (l, \#(\Phi_{\mathrm{L}}))$, given by l arrays $\tilde{\mathbf{U}}_J^{(i)}$, with cell values $\tilde{u}_{i_{\mathrm{L}}}^{(i)}$, $i \in \Theta_l$; $i_{\mathrm{L}} \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.
2: Evaluate error Δ between $\mathbf{U}_J^{\mathrm{new},(i)}$ and $\tilde{\mathbf{U}}_J^{(i)}$ for $i \in \Theta_l$, with the Algorithm 8.7.
3: Estimate of time integration local error: $\Delta \to err$.
4: For a given accuracy tolerance η and a security factor υ, compute the new splitting time step Δt^{new}:

$$\Delta t^{\mathrm{new}} = \upsilon \, \Delta t \left(\frac{\eta}{err} \right)^{-1/2}. \tag{8.8}$$

5: **Output:** New splitting time step: $\Delta t \to \Delta t^{\text{new}}$, and estimate of time integration local error: err.

8.3.3 Time Adaptive Splitting Algorithm

With the previous elements, the Algorithm 8.9 introduces the general time adaptive splitting technique developed in Chapter 5, for both uniform and adapted spatial meshes.

Algorithm 8.9 Time adaptive splitting scheme: $\mathcal{S}_2^{\Delta t}$.

1: **Input:** Solution $\mathbf{U}_J$ at $t = t$, of size $(m, N_{\text{L}}) = (m, \#(\Phi_{\text{L}}))$, given by m arrays $\mathbf{U}_J^{(i)}$ with cell values $u_{i_{\text{L}}}^{(i)}$, $i = 1, \ldots, m$, $i_{\text{L}} \in \bigcup_{r \in [1, N_{\text{R}}]} \text{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.
2: Set the time integration accuracy tolerance: η, and a security factor for time step computation: υ.
3: Set accuracy tolerances for split reaction and diffusion time integration: η_{Radau5} and η_{ROCK4}.
4: Start time integration of the solution: $integrate$ = `.true.`
5: **while** $integrate$ = `.true.` **do**
6: Compute Strang and shifted Strang solutions: $\mathbf{U}_J^{\text{new}}$ and $\tilde{\mathbf{U}}_J$, from initial solution $\mathbf{U}_J$ with splitting time step Δt by Algorithm 8.6.
7: Obtain the new splitting time step: Δt^{new}, and the local error estimate: err, with Algorithm 8.8.
8: **if** $err \leq \eta$ **then** {Accept current Strang solution}
9: Update solution: $\mathbf{U}_J^{\text{new}} \to \mathbf{U}_J$.
10: Update time: $t \to t + \Delta t$.
11: Stop time integration: $integrate$ = `.false.`
12: **end if**
13: Update splitting time step: $\Delta t \to \Delta t^{\text{new}}$.
14: Limit time domain: $\Delta t = \min\{\Delta t, T - t\}$
15: **end while**
16: **Output:** Strang solution $\mathbf{U}_J$ at $t = t + \Delta t$, of size $(m, N_{\text{L}}) = (m, \#(\Phi_{\text{L}}))$, given by m arrays $\mathbf{U}_J^{(i)}$ with cell values $u_{i_{\text{L}}}^{(i)}$, $i = 1, \ldots, m$, $i_{\text{L}} \in \bigcup_{r \in [1, N_{\text{R}}]} \text{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.

As for the general splitting time integration scheme in Algorithm 8.5, we might consider the time adaptive Algorithm 8.9 for the time evolution operator from t to $t + \Delta t$, in the multiresolution scheme (Algorithm 7.11). In this way, a time/space adaptive numerical solution is achieved within a prescribed accuracy η, where the approximation errors introduced by the compressed spatial representations are also monitored by the multiresolution threshold value ε. Some numerical illustrations on this new numerical strategy are provided and discussed in forthcoming chapters. Let us also remark that both algorithms 8.5 and 8.9 are implemented in the same code, and that the user may finally select the most appropriate configuration.

8.3.4 Correction of Splitting Time Step

We have also analyzed in Chapter 5, the numerical behavior of the adaptive scheme for non-asymptotic regimes, *i.e.*, for larger splitting time steps Δt, for which the numerical estimates of

the error might become inaccurate. In the following, we detail the complementary numerical procedure introduced in this context to effectively handle these situations. Let us recall that the problem arises for relatively large tolerances η, and in particular for small time shifts δ. Therefore, after many numerical tests and from a practical point of view, we can conclude that by setting sufficiently large shifts δ, a compromise can be drawn, and the Algorithm 8.9 can be applied neglecting these issues. The code includes nevertheless the following algorithms which can be excluded by the user.
We recall that the numerical procedure is based on the approximation (5.48):

$$\Delta t^\star \approx \frac{\delta C_\delta}{C_0}, \tag{8.9}$$

where $\Delta t^\star$ denotes the critical splitting time step, *i.e.*, the maximum splitting time step Δt for which the local error estimates are completely reliable. The coefficient C_δ is approximated by

$$C_\delta \approx \frac{err}{\delta \Delta t^2}, \tag{8.10}$$

according to (5.46). Finally, C_0 is calculated from (5.56) by means of two local estimators: e_1 and e_2, given by (5.54) and (5.57):

$$e_1 = \mathcal{S}^{a_1\Delta t}u_0 - \mathcal{S}^{b_1\Delta t}(\mathcal{S}^{c_1\Delta t}u_0), \qquad e_2 = \mathcal{S}^{a_2\Delta t}u_0 - \mathcal{S}^{b_2\Delta t}(\mathcal{S}^{c_2\Delta t}u_0), \tag{8.11}$$

and defined by the sets: (a_1, b_1, c_1) and (a_2, b_2, c_2). Assuming the worst configuration in which both e_1 and e_2 are given by their maximum values in (5.56), some simple computations yield

$$e_1 \approx C_0 \Delta t^3 \left(\omega c_1^3 (a_1^3 - b_1^3)\right), \tag{8.12}$$

with

$$\omega \approx \frac{e_2(a_1^3 - b_1^3) - e_1(a_2^3 - b_2^3)}{e_1 c_2^3 - e_2 c_1^3}. \tag{8.13}$$

The computation of the critical splitting time step $\Delta t^\star$ is finally performed in two stages. The idea is to save memory resources and embed the computations with an appropriate choice of the sets (a_1, b_1, c_1) and (a_2, b_2, c_2), as detailed in Chapter 5. The following algorithms 8.10 and 8.11 illustrate the implementation of the numerical estimates in the code.

Algorithm 8.10 Computation of the critical splitting time step I: $\Delta t^\star$.

1: **Input:** Solution $\mathbf{U}_J$ of size $(m, N_{\mathrm{L}}) = (m, \#(\Phi_{\mathrm{L}}))$, given by m arrays $\mathbf{U}_J^{(i)}$ with cell values $u_{i_{\mathrm{L}}}^{(i)}$, $i = 1, \ldots, m$, $i_{\mathrm{L}} \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.
2: Build array $\tilde{\mathbf{U}}_J^{\mathbf{1}}$ of size (l, N_{L}), from $\mathbf{U}_J^{(i)}$ with $i \in \Theta_l$.
3: For $t \to t + c_2\Delta t$: time integration of $\tilde{\mathbf{U}}_J^{\mathbf{1}}$ with the standard Strang scheme by Algorithm 8.5.
4: For $t \to t + b_2\Delta t$: time integration of $\tilde{\mathbf{U}}_J^{\mathbf{1}}$ with the standard Strang scheme by Algorithm 8.5.
5: Build array $\tilde{\mathbf{U}}_J^{\mathbf{2}}$ of size (l, N_{L}), from $\mathbf{U}_J^{(i)}$ with $i \in \Theta_l$.
6: For $t \to t + c_1\Delta t$: time integration of $\tilde{\mathbf{U}}_J^{\mathbf{2}}$ with the standard Strang scheme by Algorithm 8.5.
7: Evaluate error Δ between $\tilde{\mathbf{U}}_J^{\mathbf{2}}$ and $\tilde{\mathbf{U}}_J^{\mathbf{1}}$ with Algorithm 8.7.
8: Estimate of local estimator e_2: $\Delta \to e_2$.

9: For $t \to t + b_1 \Delta t$: time integration of $\tilde{\mathbf{U}}_J^{\mathbf{2}}$ with the standard Strang scheme by Algorithm 8.5.
10: **Output:** Local estimator e_2 and solution $\tilde{\mathbf{U}}_J^{\mathbf{2}}$ at $t = t + \Delta t$, of size $(l, N_{\mathrm{L}}) = (l, \#(\Phi_{\mathrm{L}}))$, given by l arrays $\tilde{\mathbf{U}}_J^{\mathbf{2},(i)}$, with cell values $\tilde{u}_{i_{\mathrm{L}}}^{2,(i)}$, $i \in \Theta_l$, $i_{\mathrm{L}} \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.

Algorithm 8.11 Computation of the critical splitting time step II: $\Delta t^\star$.

1: **Input:** Strang solution $\mathbf{U}_J$ at $t = t + \Delta t$, of size $(m, N_{\mathrm{L}}) = (m, \#(\Phi_{\mathrm{L}}))$, given by m arrays $\mathbf{U}_J^{(i)}$ with cell values $u_{i_{\mathrm{L}}}^{(i)}$, $i = 1, \ldots, m$; and solution $\tilde{\mathbf{U}}_J^{\mathbf{2}}$ at $t = t + \Delta t$, of size $(l, N_{\mathrm{L}}) = (l, \#(\Phi_{\mathrm{L}}))$, given by l arrays $\tilde{\mathbf{U}}_J^{\mathbf{2},(i)}$, with cell values $\tilde{u}_{i_{\mathrm{L}}}^{2,(i)}$, $i \in \Theta_l$; $i_{\mathrm{L}} \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.
2: Evaluate error Δ between $\mathbf{U}_J^{(i)}$ and $\tilde{\mathbf{U}}_J^{\mathbf{2},(i)}$ for $i \in \Theta_l$ with Algorithm 8.7.
3: Estimate of local estimator e_1: $\Delta \to e_1$.
4: Compute C_δ with current shift δ in (8.10), ω in (8.13), and C_0 in (8.12).
5: Compute $\Delta t^\star$ from (8.9) and set for a given security factor ζ: $\Delta t^\star \to \zeta \Delta t^\star$.
6: **Output:** Critical splitting time step: $\Delta t^\star$.

A working region for the splitting time steps was also defined in Chapter 5 by (5.60): $\Delta t \in [\beta \Delta t^\star, \gamma \Delta t^\star]$, in order to ensure better predictions of the local error estimate with the appropriate time shift δ. The latter parameter is estimated by the following Algorithm 8.12.

Algorithm 8.12 Computation of the time shift: δ.

1: **Input:** Current splitting time step Δt, and estimates C_δ and C_0.
2: Compute δ from (8.9) by considering $\Delta t^\star = \Delta t$.
3: Set for a given security factor θ: $\delta = \min\{\theta\delta, \delta_{max}\}$.
4: Compute $\Delta t^\star$ from (8.9) with new δ.
5: **Output:** New time shift: δ, and corresponding critical splitting time step: $\Delta t^\star$.

The Algorithm 8.9 is finally rewritten with the inclusion of the previous tools in the Algorithm 8.13. With respect to the previous adaptive splitting implementation, this one needs to save one more array, of size (l, N_{L}). The computation of the critical splitting time step is done either periodically with a rate given by N_δ, or whenever the splitting time steps are casted away from the defined working region, indicated by the flag *estimate*, initially set to `.false.`. Moreover, C_δ^0 was introduced to reject the initial splitting time step Δt^0 at the first time iteration $n = 0$, if necessary. The Algorithm 8.13 contains thus all the elements introduced in this chapter, and can be coupled as well with the multiresolution scheme given by Algorithm 7.11, for the time/space adaptive numerical strategy.

Algorithm 8.13 Time adaptive splitting scheme II: $\mathcal{S}_2^{\Delta t}$.

1: **Input:** Solution $\mathbf{U}_J$ at $t = t$, of size $(m, N_{\mathrm{L}}) = (m, \#(\Phi_{\mathrm{L}}))$, given by m arrays $\mathbf{U}_J^{(i)}$ with cell values $u_{i_{\mathrm{L}}}^{(i)}$, $i = 1, \ldots, m$, $i_{\mathrm{L}} \in \bigcup_{r\in[1,N_{\mathrm{R}}]} \mathrm{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.
2: Set the time integration accuracy tolerance: η, and a security factor for time step computation: υ.

Set accuracy tolerances for split reaction and diffusion time integration: η_{Radau5} and η_{ROCK4}.
Set (a_1, b_1, c_1) and (a_2, b_2, c_2) for local estimators e_1 and e_2; and security factor $0 < \zeta \leq 1$, close to 1 for the critical time step $\Delta t^\star$.
Define the working region for splitting time steps: $0 < \beta < \gamma \leq 1$.
Define the maximum shift: δ_{max}, and a security factor $\theta \geq 1$, for computation of δ.
Start time integration of the solution: *integrate* = `.true.`
while *integrate* = `.true.` **do**
  if $n/N_\delta = \lfloor n/N_\delta \rfloor$ **or** *estimate* = `.true.` **then**
    Perform Algorithm 8.10 for critical splitting time step $\Delta t^\star$.
    Set *estimate* = `.true.`
  end if
  Compute Strang and shifted Strang solutions: $\mathbf{U}_J^{\text{new}}$ and $\tilde{\mathbf{U}}_J$, from initial solution $\mathbf{U}_J$ with splitting time step Δt by Algorithm 8.6.
  Obtain the new splitting time step: Δt^{new}, and the local error estimate: *err*, with Algorithm 8.8.
  if *estimate* = `.true.` **then**
    Compute critical splitting time step $\Delta t^\star$ with Algorithm 8.11: $\Delta t^\star$.
    Set *estimate* = `.false.`
    if $\Delta t \notin [\beta \Delta t^\star, \gamma \Delta t^\star]$ **then**
      Compute δ with Algorithm 8.12.
      Update δ and critical splitting time step $\Delta t^\star$.
    end if
  end if
  if $t = t_0$ **and** $\Delta t > \Delta t^\star$ **then**
    Set $err \to \eta + C_\delta^0$.
  end if
  if $\Delta t^{\text{new}} > \Delta t^\star$ **and** $\delta \neq \delta_{max}$ **then**
    Compute $\Delta t^\star$ at next time iteration: *estimate* = `.true.`
  end if
  if $err \leq \eta$ **then** {Accept current Strang solution}
    Update solution: $\mathbf{U}_J^{\text{new}} \to \mathbf{U}_J$.
    Update time: $t \to t + \Delta t$.
    Stop time integration: *integrate* = `.false.`
  end if
  Correct splitting time step: $\Delta t = \min\{\Delta t^{\text{new}}, \Delta t^\star\}$.
  Limit time domain: $\Delta t = \min\{\Delta t, T - t\}$
end while
Output: Strang solution $\mathbf{U}_J$ at $t = t + \Delta t$, of size $(m, N_{\text{L}}) = (m, \#(\Phi_{\text{L}}))$, given by m arrays $\mathbf{U}_J^{(i)}$ with cell values $u_{i_{\text{L}}}^{(i)}$, $i = 1, \ldots, m$, $i_{\text{L}} \in \bigcup_{r \in [1, N_{\text{R}}]} \text{L}(\tilde{\Lambda}_{r,\varepsilon})$ and $j \in [j_0, J]$.

Part IV

Application Framework

Introduction

In this last part, we present some numerical evaluations of the numerical techniques developed mainly in chapters 4 and 5, and implemented in the MBARETE code as described in chapters 7 and 8, for several problems coming from different domains. A first illustration was conducted throughout chapters 4 and 5 for a series of simulations of chemical waves that are often used to represent stiff features of general reacting fronts.

Chapter 9 is devoted to the numerical simulation of human brain strokes, modeled by reaction-diffusion systems with complex chemical mechanisms. The study included in this chapter might be seen as the continuation of the numerical evaluation performed in Chapter 4, this time for a much more complex model that requires the implementation of parallel computing techniques.

Numerical simulations of combustion fronts are considered in Chapter 10 for different physical configurations, modeled by reaction-diffusion-convection systems in a thermo-diffusive approach. The analysis of the numerical results assesses the predictive capabilities of the method in terms of accuracy, the performance of the time/space adaptive features of the method, as well as the potential extensions to more complex cases.

Chapter 11 is devoted to the numerical simulation of positive streamer discharges in plasma applications. The method introduced and evaluated in the previous chapters is integrated into a new second order method, specially conceived for the numerical solution of multi-scale plasma models. In this way, this chapter illustrates further extensions of this work, based on the numerical strategy developed in chapters 4 and 5. Numerical simulations are performed for propagating ionization waves and highly unsteady plasma discharge configurations, for which the capabilities of the new strategy are evaluated.

Chapter 9

Brain Stroke Simulations

We focus in this chapter on the numerical implementation of the strategy introduced in Chapter 4, for brain stroke simulations. Multi-scale propagating waves and stiff phenomena are associated with many biomedical applications. One example is given by electrocardiology simulations, modeled by reaction-diffusion systems, for which many numerical strategies were developed over the past years. The latter ones usually combine adaptive mesh refinement techniques with dedicated stiff solvers in order to cope with the numerical difficulties (see, *e.g.,* [CGH00, CGH03, CFP04, CFDE+06, YRH08] and references therein). In this context, splitting schemes have also proved to be very efficient to handle such problems, and have been largely implemented even since (see, *e.g.,* [TSA00, BWZ+02, TK04]). Splitting techniques were also considered to solve chemotaxis models, for instance, in [GP00, TSL00, GV02, RS09]. Let us also refer to [Mil08] and references therein for other type of application, in which the main goal of the numerical simulations is to predict drug targeting to inhibit biological malfunctions.

In the particular case of brain stroke simulation, the literature is much more reduced and it is mainly limited by the lack of comprehensive models, not yet available (see, *e.g.,* [DBG06] and references therein). The numerical difficulties to simulate ischemic strokes were first studied in [DD08b], based on a stroke model introduced by Dronne *et al.* in [DBG06]. These studies have led to numerical simulations on realistic 3D brain geometries, for the first time, based on the splitting strategy detailed in Chapter 4. The latter scheme was implemented with satisfactory results in a code called ZEBRE, developed by T. Dumont, which considers finite volumes of constant size [Dum07]. Nevertheless, in order to further reduce computational costs, the fully MR/splitting technique introduced in [DMD+12] was also implemented and confronted with the previous results on a simplified brain geometry. All these results have been recently submitted for publication [DDD+12].

In what follows, we will first reproduce the latter article, which stands as a complete study on the subject within the present work, and in particular allows us to partially validate the MBARETE code in terms of practical implementation, by comparing some numerical results with those obtained with the ZEBRE code. We remark that even though both codes consider the same time integration strategy, they are written in different programming languages, with also different data and code structures. In the second part of this chapter, we will present more details on the numerical simulations achieved with the MR/splitting numerical strategy presented in Chapter 4. The latter study has been presented during a Summer School of CNRS *GDR Groupe Calcul* on Multiresolution and Adaptive Mesh Refinement Methods, Fréjus, France (2010) [DMD+11b].

SIMULATION OF HUMAN ISCHEMIC STROKE IN REALISTIC 3D GEOMETRY

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

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

Keywords. *Ischemic stroke; Reaction-diffusion equations; Operator splitting; Parallel computing.*

Mathematics Subject Classification. 35A35, 35K57, 65L06, 65M08, 65M50, 65Y05, 92B05.

9.1 Introduction

Stroke is a major public health problem since it represents the second leading cause of death worldwide and the first cause of acquired disability in adults. In the United States, this disease strikes once every 40 seconds and causes death every 4 minutes, with an estimated 41.6% death rate in 2007 [LJAC+09]. Most frequently (80%) strokes result from the occlusion of one or several brain vessels and are thus called ischemic strokes (in the other cases, strokes are hemorrhagic strokes). Ischemic stroke involves many pathophysiological mechanisms causing devastating neurological damage (see for review [DIM99, MLI10]). Understanding these mechanisms is of the most importance to develop new therapeutic strategies since no treatments are currently available for most stroke patients. Currently, the only FDA-approved treatment for stroke patients is a thrombolytic agent (tPA) which can only be given to less than 10% of patients because of its narrow time-window and its hemorrhagic risks [Gra03]. Many neu-

roprotective agents (aimed at blocking the ischemic cascade) have also been developed but, although they had given very promising results in preclinical studies in rodent models, they appeared ineffective or even noxious during the clinical trials in stroke patients (see for review [DKSL99, BABS01, WA04, DT07]). This discrepancy between the results in rodents and in humans is partly due to the anatomic and histological differences between rodent and human brains. In this case, results in rodents are thus difficult to extrapolate to stroke patients. As a consequence, a mathematical model and its numerical simulations can help both to test some biological hypotheses concerning the involved mechanisms and to give new insights concerning the effects of these neuroprotective agents.

Previous works have been conducted on stroke modeling. One of these models [DBG06] is focused on the main mechanisms leading to cell death during the first hour of an ischemic stroke (such as ionic movements, glutamate excitotoxicity and cytotoxic edema). This model is based on a system of ordinary differential equations (ODEs) and is mainly an electrophysiological model. It describes the dynamics of membrane potentials, cell volumes and ionic concentrations (K^+, Na^+, Cl^-, Ca^{2+} and Glu^-) in brain cells and in the extracellular space during a stroke. This model was used to study the role of various cell types during ischemia [DGD+07] and to explore the effects of various neuroprotective agents in stroke patients [DDGG09]. Other models have been developed to simulate and study spreading depressions during a stroke. This phenomenon is characterized by a slowly propagating depolarization of brain cells along with drastic disruption of ionic gradients [Som01]. These spreading depressions have recently been observed in stroke patients [DSF+08] and are supposed to extent the ischemic damage [SDJ+05]. Some models reproduce and study the behavior of spreading depressions in neuronal cells [RRGR98, KWS00]. Others describe these depolarization waves though neuronal and glial cells [DGC+08]. Other models study the influence of the human brain cortex geometry on the propagation of these spreading depressions [DGC+08, GDD+08a]. All these models are based on reaction-diffusion systems and in this paper we choose to use the mathematical model [DBG06].

The final goal of our work is to utterly describe and reproduce precocious mechanisms of stroke (*i.e.,* ionic movements, glutamate excitotoxicity and cytotoxic edema) including the spreading depressions, for a realistic brain geometry. A first description of the algorithms used for the numerical solution of this stroke model on 1D and 2D geometries was presented in a previous article [DD08b]. However, since we need to take into account the anatomic and histological specificities of human brain, this model must be simulated on a 3D realistic geometry, which implies to develop powerful numerical methods able to deal with a broad spectrum of spatial and temporal scales. This paper focuses on the methods developed for the numerical solution of this model, with much more insights on the mathematical and numerical methods than in [DD08b]. The numerical method is based on operator splitting and explicit/implicit Runge-Kutta methods. A very important feature of this method is that no linear system (of large size) is solved. We then show, for the first time, numerical simulations in 3D obtained thanks to a particular implementation of parallelism in the framework of shared memory machines. Moreover, these 3D simulations are computed on realistic geometries, obtained from MRI of the human brain, on conventional computational resources, that is on nowadays reasonably powerful workstations; and they are shown to match the observed dynamics from MRI images in stroke patient. Since accuracy in 3D simulations is not yet optimal, the ability of extending the proposed numerical strategy to adaptive multiresolution is presented in the framework of preliminary computations in simple geometries, based on a strategy introduced in [DMD+12]. The idea is to increase the level of accuracy in order to match all the spatial scales, with a better computational efficiency; thanks to the fact that phenomenons in strokes are spatially localized, a local mesh adaptation (like multiresolution techniques) is the most suitable.

The paper is organized as follows: in a first part, we present the reaction-diffusion model of the precocious mechanisms. We then focus on numerical methods: we first mention the different approaches which can be used to discretize the system in time and explain why in the context of such a stiff and large system only very few are relevant. We then present our numerical strategy based on splitting methods; a grid adaptation technique is also proposed as a possible improvement of the numerical strategy, considering particular features of the phenomena. We present the parallel implementation on shared memory machines of the numerical strategy, and discuss the numerical validation of the results. In the next section, 2D and 3D numerical results of simulations with complex geometry are presented. Biological results obtained are compared with real observations and discussed in the penultimate section. Biomarkers are used in order to validate these computations. A brief and prospective study based on coupling the proposed strategy with adaptive multiresolution in space is conducted, whereas conclusion and future works are presented in the last section.

9.2 Stroke Modeling through Stiff Reaction-Diffusion Systems

In this section, we describe the model on which our study is based. This model includes ionic movements, glutamate excitotoxicity, cytotoxic edema and spreading depressions [DBG06, DGD+07]. It thus focuses on the first hour of a stroke, when the ionic exchanges are the main mechanisms leading to cell death. This model is based on a reaction-diffusion system (equations are given in what follows in Table 9.1).
In this model, brain tissue is composed of two cell types, namely neurons and glial cells, and of extracellular space. Two domains are considered: the white and the gray matter which differ in their glial cell composition (astrocytes in gray matter and oligodendrocytes in white matter) and in their "neuronal area" composition (neuronal somas in gray matter and neuronal axons in white matter). Human brain cortex is exclusively composed of gray matter whereas human brain space is mainly composed of white matter (except the gray kernels). For simplicity reasons, we consider in the model that brain cortex contains only gray matter and brain space contains only white matter. The ionic species considered in this model are K^+, Na^+, Cl^-, Ca^{2+} and the Glutamate (glu). They pass through neuronal and glial membranes via ionic channels (such as voltage-gated channels, receptor-channels, stretch-channels) and via ionic pumps and transporters (which are energy-dependent) (see Figure 9.1). The ionic exchanges through voltage-gated channels have been first modeled by Hodgkin and Huxley [HH52].
The main precocious mechanisms of ischemic stroke can be described as follows (see for review [DIM99, MLI10]): after the stroke onset, the cells in the ischemic area do not receive enough oxygen to maintain their production of energy. As a consequence, the activity of the ionic pumps decreases, which results in variations of ionic concentrations in the cells and in the extracellular space. These ionic variations have several consequences:

- The alteration of membrane potentials, resulting in membrane depolarization and in the opening of the voltage-gated channels;
- The cell swelling due to water influx;
- The increase of the neuronal concentration of Ca^{2+}, resulting in enzyme activation and leading cells towards necrosis;
- The increase of glutamate in the extracellular space, reinforcing the excitotoxic process;
- The increase of the concentration of K^+ propagating in the extracellular space and the increase of Ca^{2+} in the astrocytic synticium, creating waves of cortical spreading

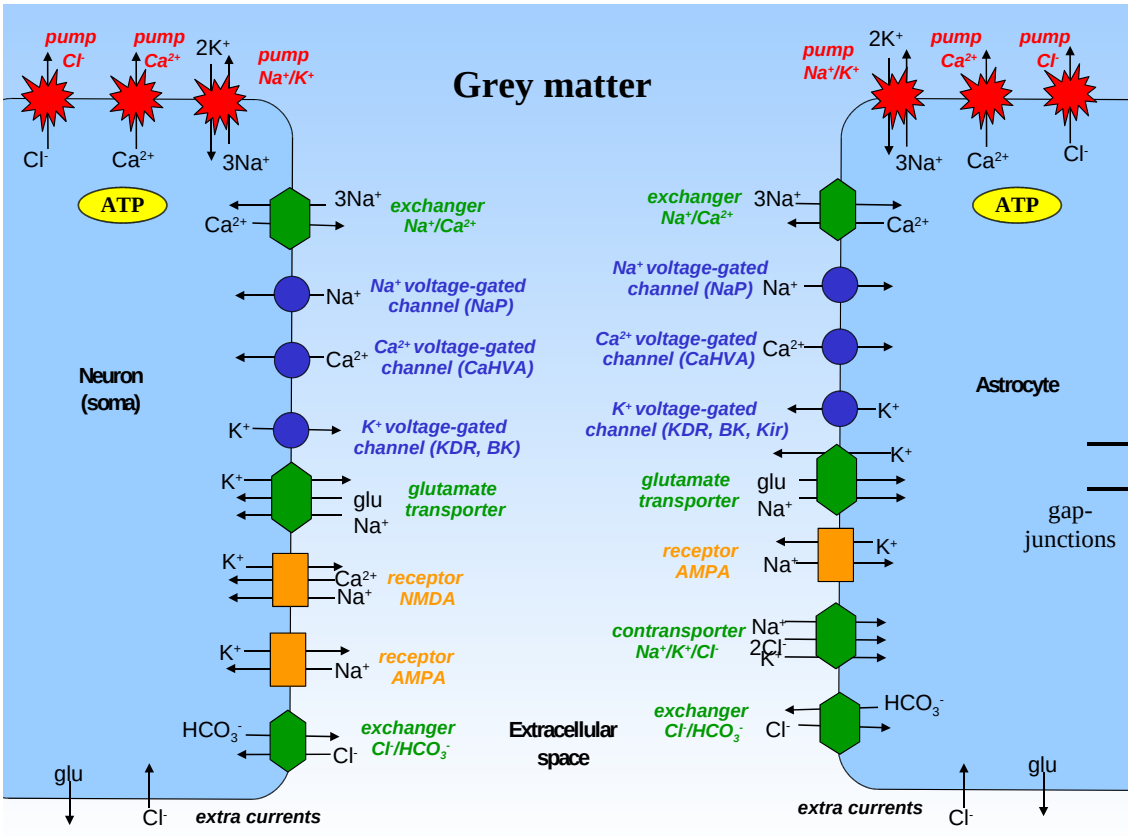

Fig. 9.1: *Ionic exchanges in gray matter between neurons, astrocytes and the extracellular space through voltage-gated channels, ionic transporters, receptor-channels and ionic pumps (from [DBG06]).*

depressions, opening further ionic channels and thus expanding the ischemic damage far from the ischemic core.

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

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

All the variables depend both from time and coordinates.
Altogether, the mean field model has $m = 19$ unknowns written as a reaction-diffusion system of equations. However, there is no diffusion for 4 unknowns, namely f_n, f_a, V_n and V_a and not all ion concentrations diffuse in gray matter and in white matter. Since gray matter contains astrocytes (which are linked into an astrocytic syncytium thanks to gap-junctions), ions are able to diffuse in the astrocytic space as well as in the extracellular space in gray matter. On the other side, as the main glial cells in white matter are oligodendrocytes (which do not have the same properties as astrocytes), ions are considered to be only able to diffuse in extracellular

space in white matter. As a consequence, the model contains 10 reaction-diffusion equations in gray matter (for the concentrations of K^+, Na^+, Cl^-, Ca^{2+} and Glu^- in astrocytes and in the extracellular space) and 5 reaction-diffusion equations in white matter (for the concentrations of (K^+, Na^+, Cl^-, Ca^{2+} and Glu^- in the extracellular space). To be more precise, in Table 9.1 we summarize the modeling equations of a human ischemic stroke, developed by Dronne *et al.* in [DBG06], using the following notations:

- S stands for the ionic species (Ca^{2+}, Na^+, K^+, Cl^- and Glu^- respectively).
- The subscript e stands for extracellular, n for neuronal and a for glial medium (astrocytes in grey matter and oligodendrocytes in white matter).
- $\varepsilon_{s,a}$ and $\varepsilon_{s,e}$ are the diffusion coefficients for each ion "S" in glial cells and in the extracellular space respectively. In white matter $\varepsilon_{s,a} = 0$.
- $It_{s,n}$ and $It_{s,a}$ are the global ionic currents for each ion S through neuronal membrane and through glial membrane respectively. For example in grey matter, $It_{s,n}$ is the sum of all the currents concerning the transport of ion S through neuronal membrane via the voltage-gated channels, transporters and receptor-channels represented in Figure 9.1. These current equations mainly rely on Hodgkin-Huxley equations and come from other electrophysiological models (neuronal and glial models: [Wal92, DMS98, YKA98, ROA00, Sha01] and cardiac models [DFN85, LRJ92] for some currents.

Other parameters are given by:

$N_{imp,a}$	:	number of moles of impermeant anions in the glial cells (constant)
$N_{imp,n}$	:	number of moles of impermeant anions in the neuron (constant)
n_n	:	number of neurons in each volume unit
n_a	:	number of glial cells in each volume unit
σ_n	:	neuron surface
σ_a	:	glial cells surface
z_s	:	valence of ion S
v	:	volume of each unit
c_n	:	neuron capacity
c_a	:	glial capacity
F	:	Faraday's constant

For α_n and α_a we follow [YFKP03]: $\alpha_n = \alpha_a = L_i RT/v$, with $L_i = 1.21 \times 10^{12}$ cm^3/(Pa$\cdot$min), $R = 8.3145$ J/(mol $\cdot$ K), and $T = 310.15$ K.
The PDEs in Table 9.1 can be written in the following general form:

$$\left.\begin{array}{l} \dfrac{\partial u_i}{\partial t}(x,t) - \mathrm{div}(\varepsilon_i(x)\mathrm{grad}\ u_i(x,t)) = f_i(u_1(x,t),\ldots,u_m(x,t)), \\ u_i(x,0) = u_i^0(x), \\ 1 \le i \le m,\ x \in \Omega. \end{array}\right\} \tag{9.1}$$

The domain Ω corresponds to a human brain and is divided in gray and white matter. These two matters differ in several coefficients in the reaction term (corresponding to the cell composition) and in their diffusion coefficients, as previously described. There are no fluxes of ions in and out of the brain and thus, the boundary conditions are of Neumann homogeneous type. For the initial conditions $u_i(x,0) = u_i^0(x)$, $1 \le i \le m$, a classical medical hypothesis is that the system is in a stable equilibrium: thus we take, and must find, a stable constant solution of system (9.1).

Table 9.1: *Model equations describing the dynamics of the ionic concentrations, cell volumes and membrane potentials with diffusion in neurons, in glial cells and in the extracellular space.*

Variables		Equations
Ionic concentrations	S_n	$\frac{\partial S_n}{\partial t} = -\frac{n_n\ \sigma_n\ It_{s,n}}{f_n\ z_s\ F\ v} - \frac{S_n}{f_n}\frac{\partial f_n}{\partial t}$
	S_a	$\frac{\partial S_a}{\partial t} = \mathrm{div}(\varepsilon_{s,a}\ \mathrm{grad}\ S_a) - \frac{n_a\ \sigma_a\ It_{s,a}}{f_a\ z_s\ F\ v} - \frac{S_a}{f_a}\frac{\partial f_a}{\partial t}$
	S_e	$\frac{\partial S_e}{\partial t} = \mathrm{div}(\varepsilon_{s,e}\ \mathrm{grad}\ S_e) + \frac{n_n\ \sigma_n\ It_{s,n} + n_a\ \sigma_a\ It_{s,a}}{(1-f_n-f_a)\ z_s\ F\ v} - \frac{S_e}{1-f_n-f_a}\frac{\partial(1-f_n-f_a)}{\partial t}$
Proportions of intracellular volumes	f_n	$\frac{\partial f_n}{\partial t} = \alpha_n\ \left(\sum S_n - \sum S_e + \frac{N_{imp,n}}{v\ f_n}\right)$
	f_a	$\frac{\partial f_a}{\partial t} = \alpha_a\ \left(\sum S_a - \sum S_e + \frac{N_{imp,a}}{v\ f_a}\right)$
Membrane potentials	V_n	$\frac{\partial V_n}{\partial t} = -\frac{\sigma_n}{c_n}\sum_s It_{s,n}$
	V_a	$\frac{\partial V_a}{\partial t} = -\frac{\sigma_a}{c_a}\sum_s It_{s,a}$

Let us mention some characteristics of the system which are very important in the choice of numerical schemes:

- The reaction term $F = (f_1, ..., f_m)^t$ is extremely stiff; that is to say that if we consider the system of differential equations $du/dt = F(u)$, it is a stiff system according to the definition given in [HW96]. To see this, we have performed, by numerical differentiation, a computation of the Jacobian matrix $(\partial f_i/\partial u_j)$, $1 \leq i, j \leq m$, near a stable stationary value $F(u) = 0$, and we found numerically negative eigenvalues with negligible imaginary parts but with real parts in the range from -10^8 to about -1. Moreover, it is impossible to separate fast and slow variables and even if this was possible, the voltage dependent gates would make this separation very local in time and space. We have to deal with the stiffness of the reactive term F, which is the core of the model and is a program of about 500 lines of C language.

- The diffusion coefficients $\varepsilon_i(x)$ are low: about 10^{-3} given by a non-dimensional analysis. The resulting splitting time step for a proper resolution of the propagating phenomenon resulting from the coupling with the reaction term will lead to the resolution of heat

equation in a mildly stiff framework. Exploiting this fact turns out to be very important: we can use stabilized explicit methods when solving the heat equation associated with the diffusion, with the advantage of good numerical performances, and an easy implementation of parallel computations.

The diffusion coefficients $\varepsilon_i(x)$ take two constant values in gray and white matter (respectively ε_i^g and ε_i^w). The interface conditions between gray and white matter are classical:

$$\varepsilon_i^g \text{ grad } u_i(x,t) \cdot \mathrm{n} = \varepsilon_i^w \text{ grad } u_i(x,t) \cdot \mathrm{n}, \tag{9.2}$$

where n is a normal unit vector to the boundary between gray and white matter. These conditions become Neumann homogeneous boundary conditions whenever one of the diffusion coefficients is zero.

9.3 Numerical Strategy: Operator Splitting and Time Integrators

One dimensional simulations are very useful to fit parameters such as the diffusion coefficients which are known in the literature only with limited accuracy; two dimensional ones are useful to validate numerical methods and programs, but only three dimensional simulations can be relevant from the medical point of view. From medical considerations, and also by some considerations on reaction-diffusion systems, we know that a precise description of the brain geometry is mandatory for the simulations, otherwise the plausible waves would be strongly perturbed, see for example [DDGG09]. We then have to think of a strategy dedicated to three dimensional simulations with a very fine spatial discretization allowing to resolve the broad spectrum of spatial and temporal scales of the system (9.1). The method developed has to be fast, robust and must take into account the properties of the model.
We describe now the methods introduced in this work, based on a spatial discretization which will be applied in dimension 2 and 3.
Concerning the spatial discretization, we have chosen a finite volume approach with a 5 points stencil in 2D, and a 7 points stencil in 3D. Our experience is that, with uniform finite volumes, at least $\ell = 10^7$ volumes are necessary for a realistic three dimensional simulation. The continuous unknown u is then replaced by a vector U belonging to $\mathbb{R}^{m\times\ell}$ corresponding to the m unknowns at each point x_i, $1 \le i \le \ell$. We use MRI pictures and we consider pixels as center of volumes of an uniform grid. When we apply this spatial discretization to the system (9.1), this yields a large system of ordinary differential equations. Let us write this system under the form

$$\frac{dU}{dt} = A_\varepsilon U + F(U), \tag{9.3}$$

A_ε being a matrix corresponding to the discretization of the diffusion operator; this is a classical 5 terms (resp. 7 terms) by line matrices in dimension 2 (resp. 3). We now present the different approaches which can be used to discretize this system in time and we explain why in the context of such a stiff and large system such as (9.1), only few are efficient.
The first idea is to use directly a solver of systems of ODEs, the so called method of lines, but due to the stiffness of the nonlinear term, a large system of algebraic equations should be solved at each time step, which is too much time consuming. It is then better to use different discretizations in time for the linear and the nonlinear terms. A first method is to use an Implicit–Explicit method by treating the linear term implicitly and the nonlinear term explicitly. If we denote by δt the time step and U_k the approximated solution at time $k\delta t$, the

simplest method is the following:

$$\frac{U_{k+1} - U_k}{\delta t} + A_\varepsilon U_{k+1} = F(U_k) \tag{9.4}$$

One must solve a linear system at each step since diffusion is taken implicitly but the nonlinear term is taken explicitly. This method is of order 1 in time. More accurate, but not really more expensive, methods of the same type and of order at most 6 are described and analyzed in [ACM98]. The main advantage of these methods is that only *linear* systems must be solved but the drawback is that, due to the explicit computation of the reaction terms, these methods are adapted only to systems with non stiff reaction terms. Let us recall that the system (9.1) is very stiff, and these methods can only work with time steps of the same order of the fastest time scale of the system which is about 10^{-8} seconds. This would result in an prohibitive computing time, about 4×10^{11} steps for simulating the first hour of the evolution of the stroke.
A better idea for the treatment of the linear and the nonlinear part in the context of a stiff nonlinear term is to "reverse" the numerical treatments: to solve explicitly the linear part and implicitly the nonlinear part. The discretization of the linear part is made using an explicit Runge-Kutta method with extended stability domain along the negative real axis. The papers [SSV97] and [VS04] settled the foundation for these methods called IMEX methods and particular methods devoted to stiff non linear problems are presented in [VSH04] and [SSV06]. The main advantage of these methods is that they treat diffusion terms explicitly and the stiff reaction terms implicitly. Furthermore, the stiff reaction term is decoupled over space grids and yields small sized systems. These methods are usually very efficient; nevertheless, the computational requirements associated mainly with an implicit solver over the discretized domain with the same time step become soon critical when treating large computational domains.
Finally, the only possible methods which can solve system (9.1) seem to be the so called splitting methods that we describe in details now[1].

9.4 Numerical Software

We have developed two different softwares for the solution of the system (9.1). Both implement the time integration strategy defined above in Chapter 4:

FM: (Fixed Mesh) a code using a fixed spatial discretization, with finite volumes of constant size [Dum07]. This code takes into account complex geometries in the following way: starting from MRI images, we take each pixel as the center of a finite volume; it aims to be a framework for testing and exploiting numerical methods for 1, 2 and 3D reaction-diffusion systems. It will be used in order to obtain the main results of the present contribution, that is numerical simulations of the detailed and stiff stroke model in complex 3D geometries.

MR: (Multi Resolution) a code using an adaptive multiresolution method as defined above in Chapter 4. In the framework of multiresolution, an important amount of work is still required in order to optimally combine all the numerical methods described here, the most difficult aspects are related to programming features such as data and code structures, as indicated in [DMD+12]. Nowadays, this program can only solve problems in simple domains like squares and cubes; simulations with an adaptive multiresolution

[1] The following section in [DDD+12] introduces and justifies the numerical technique detailed in Chapter 4. We omit it in this work in order to avoid redundancy. The complete text can be found in [DDD+12].

approach on a complex geometry are not yet available, and we will only present here 2D and 3D simulations in simplified geometries for the sake of assessing our results and perspectives in the field.

Let us remark that the (FM) code is a highly optimized and complete code for the simulation of reaction-diffusion equations[2]. In particular, stroke simulations in complex geometry can be performed for the first time, with standard computing resources, and constitutes the major advance of our contribution. On the other hand, the second code (MR) allows to validate to some extents the previous numerical results, and it is meant to be a potential extension to (FM) in future developments.

9.5 Numerical Results: Implementation Checkout and Accuracy Evaluation of the Code

In order to check out the implementation of the method in the codes (FM) and (MR), which use two different spatial discretizations and data structures, we have conducted a detailed comparison on a numerical test-case. Nevertheless, since we use the same numerical methods for the sub-steps integration, this does not result in a full validation. We have considered a 2D case in a regular geometry of $[0,5] \times [0,5]$ (cm), using two resolutions, one on a fixed grid computed with the code (FM) and the other on the adapted grid obtained by multiresolution with the code (MR). Both methods, based on Strang's splitting, use Radau5 and ROCK4 as time integrators for the reaction and for the diffusion problem. For the model parameters, we have considered only one domain, the gray matter. The time of integration was restricted to one hour, $t \in [0, 3600]$ seconds.
The splitting time step δt was chosen experimentally in order to obtain a good approximation of the velocity on a sufficiently 2D fine space discretization of 1024^2.

Table 9.2: *Minimum and maximum values of variables in the neurons, computed with the (MR) code, and normalized L^2 difference e of numerical results between uniform mesh and (MR). $t = 3600\,s$.*

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

We thus compare both solutions with a L^2-norm[3] at final time $t = 3600$, and define an error estimator e given by $\|u(.,t) - u_{MR}(.,t)\|_{L^2}$, where index MR denotes the use of multiresolution

[2] In particular, the ZEBRE code [Dum07] uses threads for the parallelization on shared memory architectures, based on the C++ boost-thread library [WK]. This implementation is described in [DDD+12], and is omitted in the present work for the sake of brevity.

[3] The L^2-norm $\|f\|_{L^2}$ of f is given by $\left(\int_\Omega f(x)^2\, dx\right)^{1/2}$.

Table 9.3: *Minimum and maximum values of variables in the astrocytes, computed with the (MR) code, and normalized L^2 difference e of numerical results between uniform mesh and (MR). $t = 3600\,s$.*

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

Table 9.4: *Minimum and maximum values of variables in the extracellular space, computed with the (MR) code, and L^2 normalized difference e of numerical results between uniform mesh and (MR). $t = 3600\,s$.*

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

techniques. The spatial discretization consists of an uniform mesh of 256^2 points. In the case of the spatial adaptive method, there is a set of nested grids arranged in 8 different levels from the coarsest to the finest discretization. The latter corresponds to the uniform mesh previously considered of 256^2 points. In order to compare these results, we must consider the same spatial discretization for both solutions: this is easily achieved with the mentioned projection/prediction operations on the adapted grid. Tables 9.2, 9.3 and 9.4 show the minimum and maximum values of variables in the neurons, the astrocytes and in the extracellular space respectively, as well as the normalized L^2 difference of the numerical results e obtained by the adaptive multiresolution strategy and our proposed numerical strategy.
Figure 9.2 shows the evolution of the propagating phenomenon on an adapted grid for variable K^+ in the neurons. The refined regions clearly correspond to the wavefront area where the steep spatial gradients are present. Finally, Figure 9.3 reveals in a qualitative way the different representations of the numerical solution on an uniform mesh and on the adapted one.
All these numerical results show a great accordance between the solutions of the two different codes: (MR) and (FM), in the 2D simulations. Let us recall that both codes rely on two well tested, robust and publicly available numerical routines: the Radau5 and ROCK4 methods; therefore, one can consider that they only differ by the different spatial discretizations and data structures they use: the comparison can thus be considered as a (partial) cross validation.
Besides such a level of comparisons, (MR) will be shown to pave the way towards higher levels of refinement for a better resolution of the details of the dynamics in 3D at a reasonable cost. In fact, considering this 2D numerical test-case, finer spatial discretizations yield naturally better resolution of both the wave velocity and the dynamics of the wavefront, as seen into

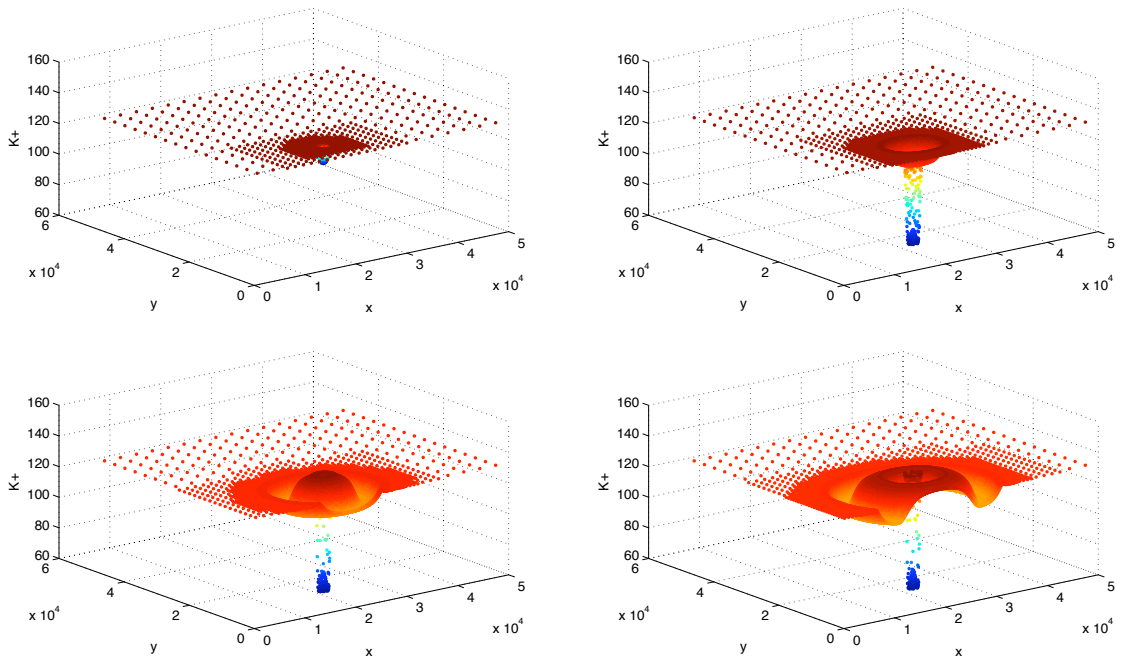

Fig. 9.2: *Evolution of K^+ in the neurons at $t = 100\,s$ (top left), $t = 1000\,s$ (top right), $t = 2000\,s$ (bottom left) and $t = 3000\,s$ (bottom right).*

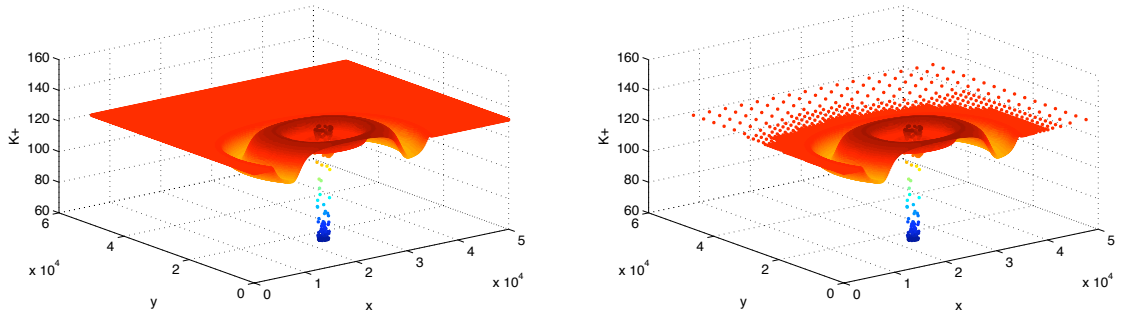

Fig. 9.3: *K^+ in the neurons at $t = 3600$ on a 256^2 uniform mesh (left) and the corresponding adapted grid (right).*

figure 9.4. See the corresponding (MR) adapted grids into Figure 9.5. Even if it is clear that the wave is better resolved on the finest grid (1024^2), in particular it is somewhat faster, the qualitative value of the wave velocity is correctly captured even on the coarsest grid (256^2), which corresponds roughly to the 3D simulations with (FM). However, we can not yet simulate with (MR) all the complex geometries of the brain we are investigating in this paper; and thus, all the results in complex geometries will be performed with (FM).

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

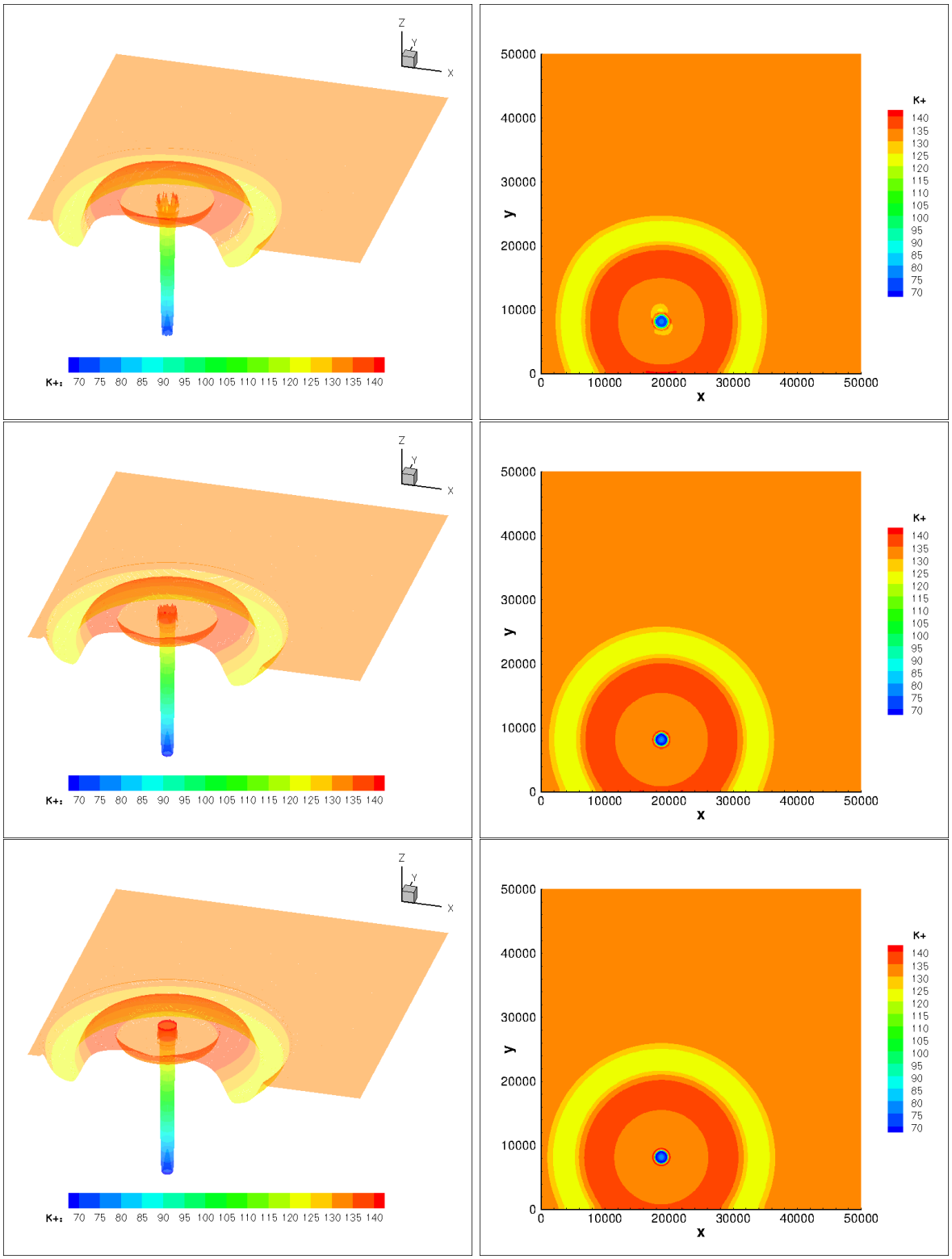

Fig. 9.4: *K^+ in the neurons at 3600 s for a 2D mesh of 256^2 (top), 512^2 (center) and 1024^2 (bottom).*

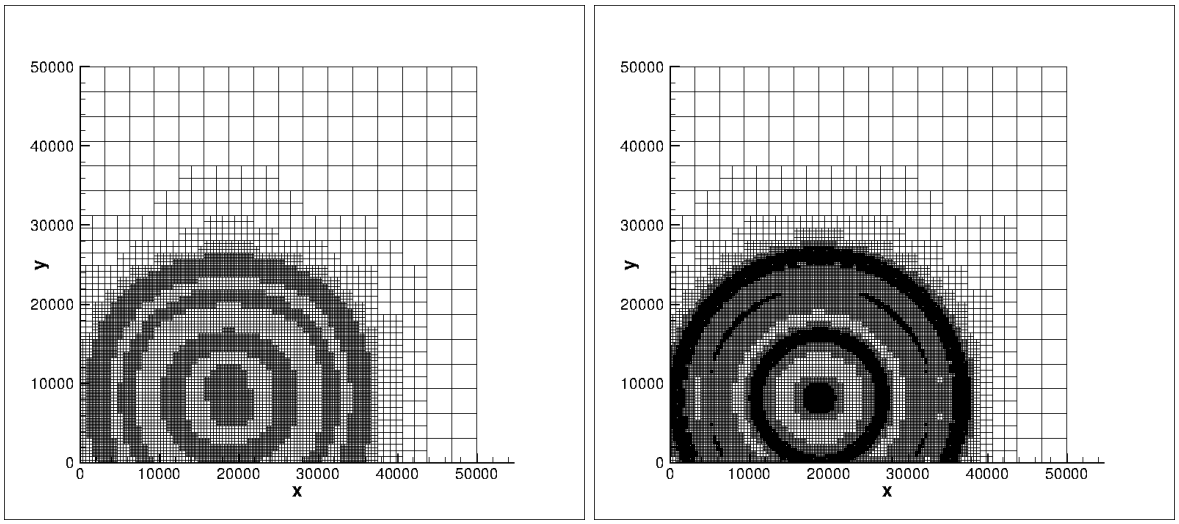

Fig. 9.5: *2D adapted meshes equivalent to* 256^2 *(left) and* 1024^2 *(right) spatial discretizations at the finest grid.*

9.5.1 Performances of the Diffusion Equation Solver

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

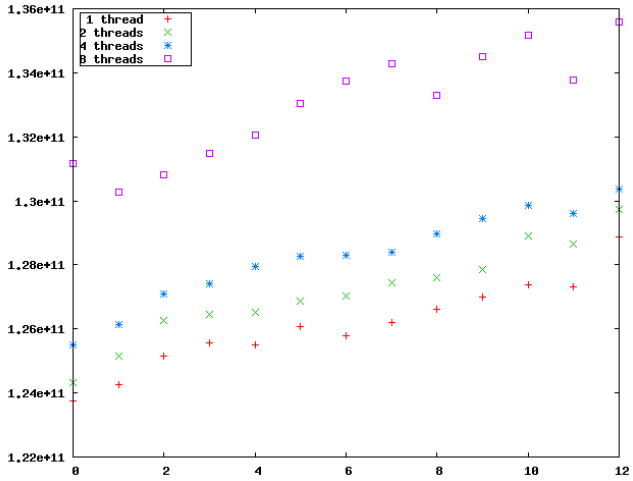

Fig. 9.6: *Performances of the multithreaded reaction solver along 12 time steps. Abscissa: time step. Ordinate: computing time in CPU clock ticks.*

9.5.2 Performances of the Reaction Solver

The main question concerns the efficiency of the multithreaded parallelism. Figure 9.6 shows the computing time with 1, 2, 4 and 8 threads, across 12 steps. The wall clock computing time is multiplied by the number of threads (unit is in number of CPU clock ticks); in case of perfect

scalability, all the points with the same number of threads should be at the same ordinate. This is roughly the case, considering that the comparison of computations obtained with 1 and 8 threads reveals a loss of scalability of only 6%. We conclude that this implementation is very efficient. This is a consequence of the complexity of the right hand side (the reaction term): even one single numerical evaluation of F is much more time consuming than the overhead introduced by the thread mechanism.
As a conclusion of this part, we can notice that our computing strategy combining splitting techniques with dedicated integration of each sub-step and multiresolution is compatible with parallelization.

9.6 Biological Results

We present and discuss here some simulation results obtained with the code FM on the complex geometry of the human brain. We simulate an ischemic stroke beginning in the cortex (in gray matter) and study the propagation of the ischemic damage. The input of the model is the decrease of the ionic currents through the ionic pumps. Two variables have been chosen for the model validation: the potassium concentration in the extracellular space ($[K^+]_e$) and the ratio of apparent diffusion coefficient of water (rADCw).

- The potassium concentration cannot be measured *in vivo* in the brain of stroke patients but it can be measured *ex vivo* or *in vitro* on brain tissues. These concentration values give some insights on the severity of the damage. The physiological value of $[K^+]_e$ is about 5 mM. It was observed to be able to increase up to 35 mM in areas of moderate ischemia where depolarization waves can spread [KN78] and up to 75-90 mM in areas of severe ischemia where most cells are dead [Han78]. The first step of the model validation is thus to compare the values of the $[K^+]_e$ obtained in the simulations with those values.

- The rADCw is a biomarker which can be estimated in the brain of stroke patients thanks to diffusion-weighted (DW) magnetic resonance (MRI) imaging. It reflects the severity of the cytotoxic edema and could thus be used to predict the ischemic damage and its extension [VBvdS+94, RHMD+09]. The value of this ratio is supposed to be 1 in physiological conditions and is known to decrease in ischemic areas. In several studies, this value in stroke patients was shown to be between 0.75 and 0.9 in areas of moderate ischemia and between 0.5 and 0.75 in areas of severe ischemia [DLR+01, FKR+01, ROS+01, SOG+01]. This biomarker can be related to the proportions of the intracellular volumes. It was shown to be proportional to the volume of the extracellular space [VBvdS+94]. Moreover, since the extracellular proportion was displayed to have a value of 0.2 in physiological conditions (*i.e.,* when rADCw=1) [MTD90], rADCw can be expressed as follows: $\text{rADCw} = 5\ (1 - f_n - f_a)$. Since f_n and f_a are two variables of the model, this ratio can be calculated for each time and for each coordinate. Another step of the model validation is thus to compare the calculated values of rADCw obtained in the simulations to the experimental values.

We present in Figures 9.7, 9.8, 9.9, 9.10, 9.11, and 9.12 some results of 2D and 3D simulations, showing the values of K^+ and rADCw biomarker in the extracellular space in different areas.

Let us make some biological comments about these results:

- First of all, we obtained depolarization waves after the simulation of a vessel occlusion in brain cortex and the depolarization waves spread in gray matter (*i.e.,* in brain cortex)

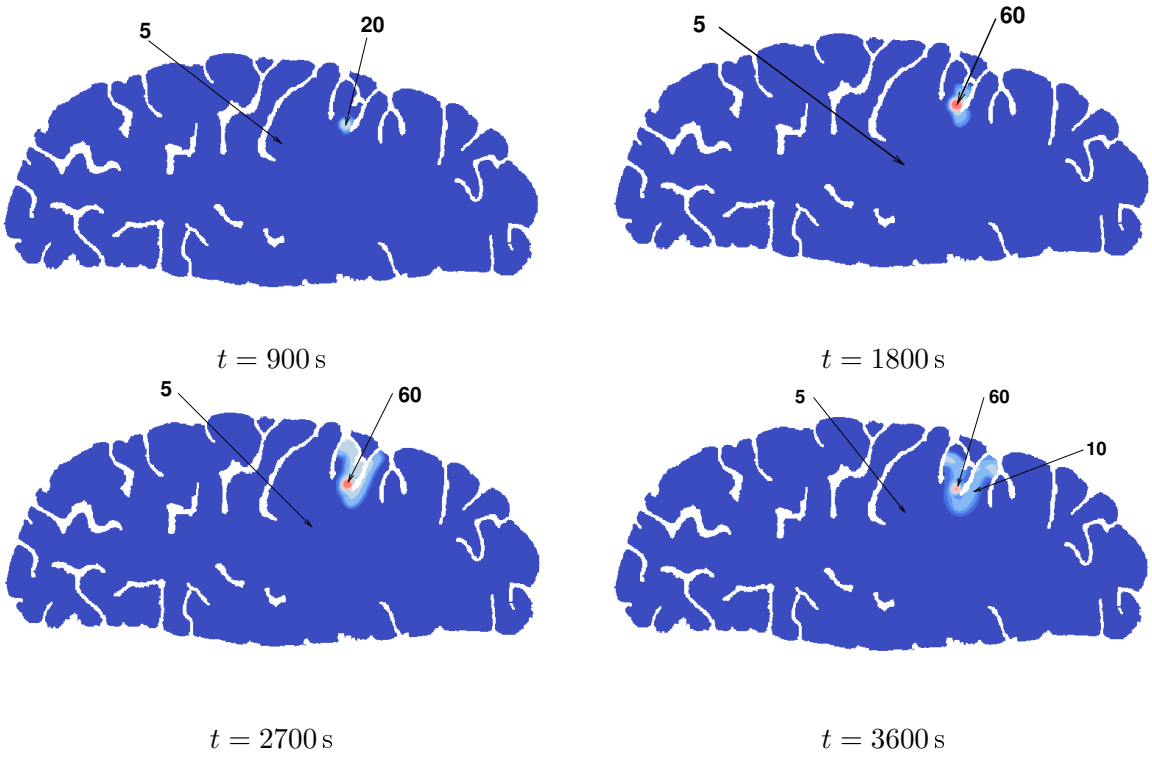

Fig. 9.7: *Evolution of K^+ in the extracellular space over one hour, 2D simulation (in millimolar (mM)), from left to right, top to bottom.*

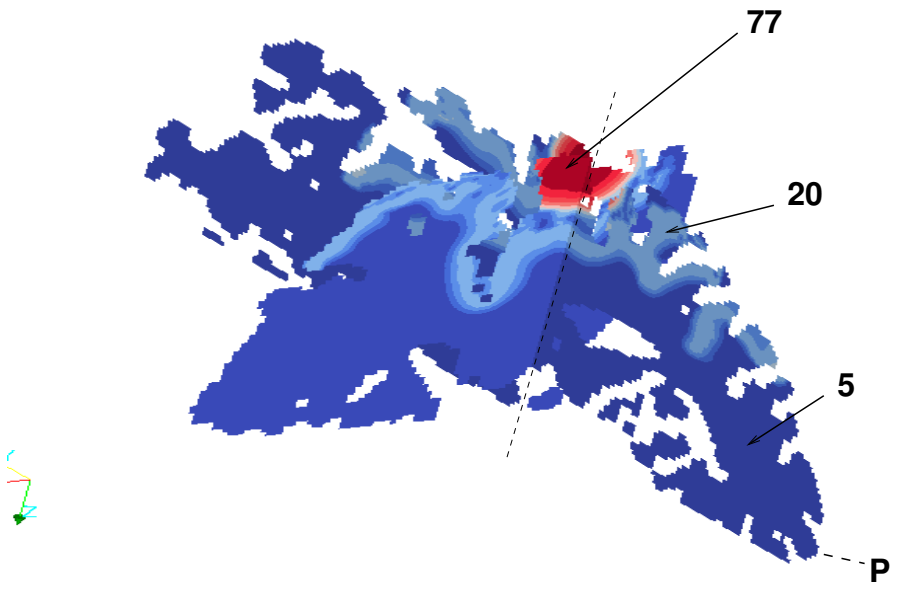

Fig. 9.8: *3D simulation; cut by two perpendicular planes of the K^+ in the extracellular space field at time $t = 3600\,s$ (in millimolar).*

and not in white matter, which is consistent with MRI images obtained in human brain [DSF+08].

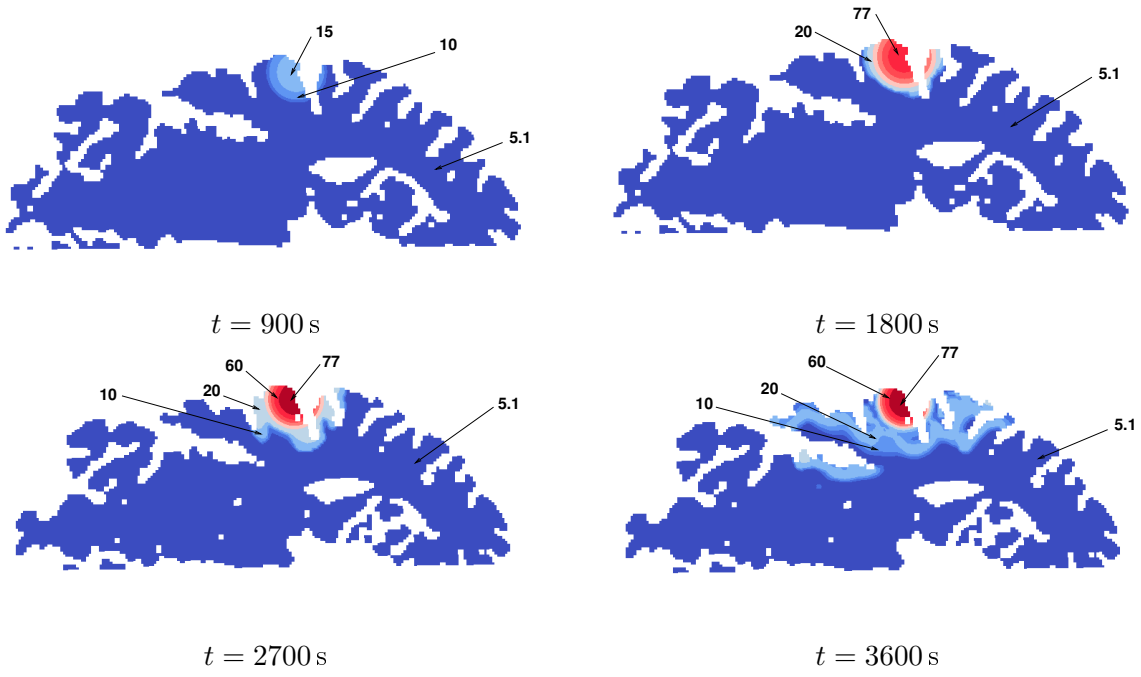

Fig. 9.9: *Evolution of K^+ in the extracellular space, 3D simulation. View in the plane* **P** *of Figure 9.8 (in mM).*

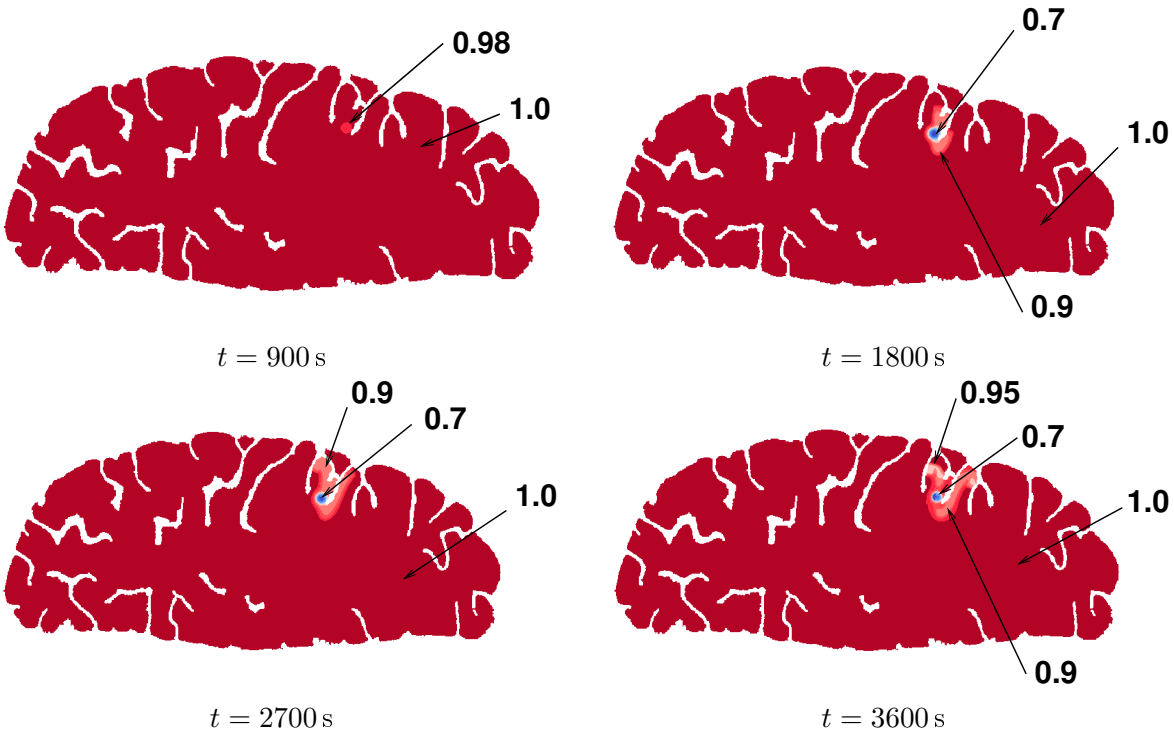

Fig. 9.10: *Evolution of rADCw over one hour, 2D simulation, from left to right, top to bottom, every 15 minutes.*

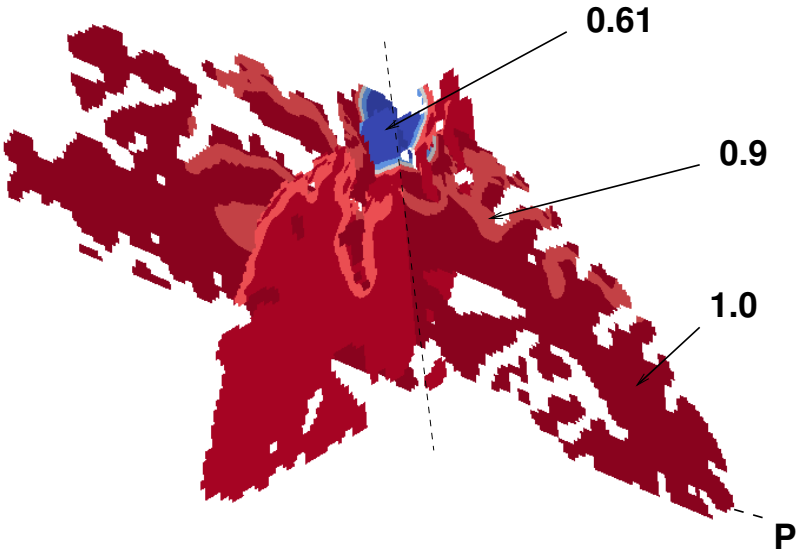

Fig. 9.11: *3D simulation; cut by two perpendicular planes of the rADCw field at time* $t = 3600\,s$.

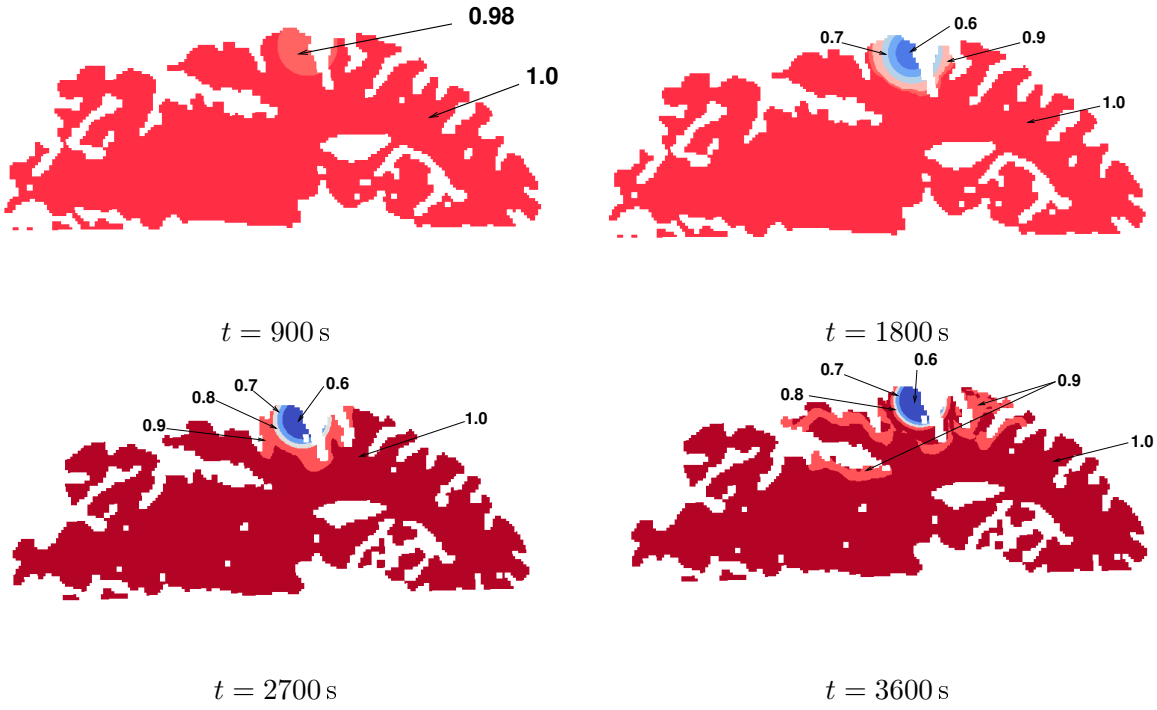

Fig. 9.12: *Evolution of rADCw in the extracellular space over one hour, 3D simulation. View in the plane* **P** *of Figure 9.11.*

- Concerning the potassium concentration in the extracellular space, Figures 9.7, 9.8, and 9.9 show that this concentration reaches values such as 77 mM in the areas where the

vessel was occluded, which is in accordance with the results obtained in the infarcted core in some experimental studies [Han78]. These figures also show that $[K^+]_e$ is about 20 mM in the areas where depolarization waves are spreading, which is consistent with the values obtained in penumbra (*i.e.,* areas of moderate ischemia, able to recover) during spreading depressions in several studies [KN78]. We can also notice that, in the safe areas, $[K^+]_e$ remains at its physiological value (5 mM).

- Concerning rADCw, Figures 9.10, 9.11, and 9.12 show that this ratio reaches values such as 0.6 in the areas where the vessel was occluded, which is in accordance with the values observed on MRI images in the infarcted core of the brain of the stroke patient [FKR+01, ROS+01, SOG+01]. These figures also show that rADCw has values between 0.75 and 0.9 in the areas where depolarization waves are spreading, which is consistent with the values obtained in penumbra during spreading depressions in stroke patients [FKR+01, SOG+01]. We can also notice that, in the safe areas, rADCw remains at its physiological value of 1.

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

9.7 Toward Better Computational Efficiency and Improved Accuracy: Adaptive Multiresolution

In the previous simulations, we notice that the simulated waves spread at a slightly slower speed. In several studies, spreading depressions were shown to spread at a rate of several millimeters per minute [MFNN00], which is not currently the case in our simulations. In fact, it is shown in [Kee87] that traveling waves solutions of reaction-diffusion equations can disappear in the numerical solution if the spatial discretization is too coarse; the velocity of the traveling waves is a function of the mesh size, and coarse meshes might perturb the accuracy of the computed wave velocity.

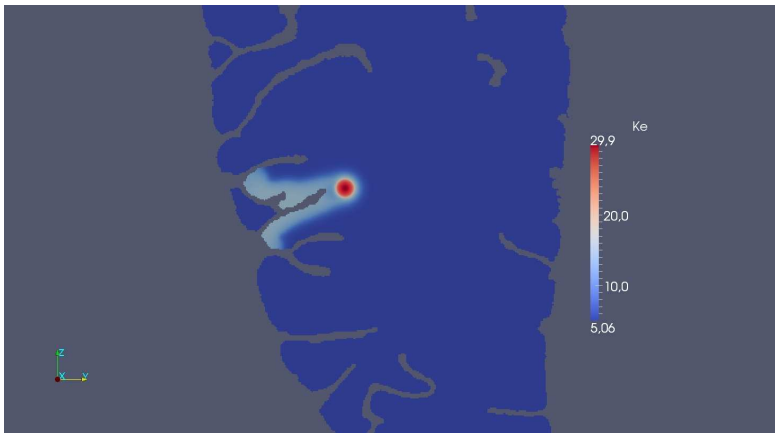

Fig. 9.13: *Local measurement of the computational cost of the reaction (in CPU clock tics).*

In particular, in the previous 3D simulations, the mesh we can use is not fine enough to obtain a correct level of accuracy for the wave velocities. In fact, coming back to the 2D numerical

test-case of section 9.5, we have seen in Figure 9.4, that a high number of volumes is needed to reproduce accurately the phenomenon, approximately 1000 per dimension.
We can also measure the computing time of the reaction at a typical step (see Figure 9.13 (one clock tick is about 0.35×10^{-9} second)). Clearly the most expensive nodes are about 37 times more expensive than the less ones! But on the other hand, the overwhelming part of the nodes are not expensive ones. Actually, 79% of the nodes (the less expensive ones, which cost less than 4.5×10^6 tics) takes 60% of the computing time.

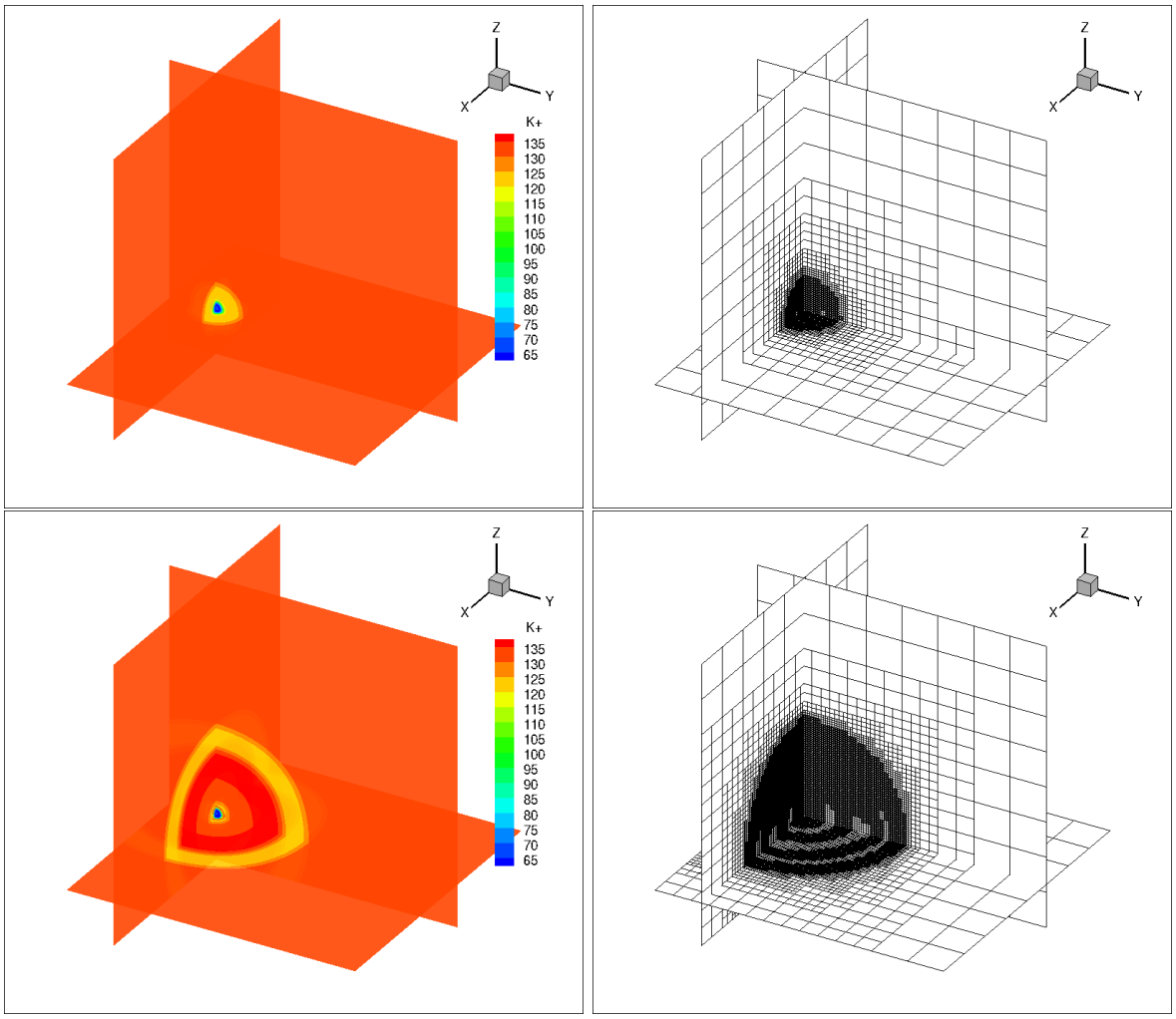

Fig. 9.14: *3D simulations with (MR). K^+ in the neurons (left) and corresponding adapted grid (right) at* 1000 *(top) and* 3600 *(bottom) seconds.*

Therefore, one way to improve both the performances and the accuracy of the resolution is to use an adaptive mesh: use a fine mesh in the ischemized zone, where the solution exhibits large gradients and wave propagation, and a coarser mesh far from this part of the domain: the multiresolution strategy, as implemented in the code (MR) is a step towards this goal.
In order to make this more concrete than just a statement, we eventually present here a first 3D multiresolution simulation in a cube, where the finest grid available has size 256^3, since complex geometry is not yet at hand (Figure 9.14). Therefore, considering the same computing resources, the computing time is reduced by a factor of about five with respect to the fixed mesh simulation with (FM) on the finest grid, even though trying to compare two so heterogeneous

codes is a very delicate task. In fact, the code (MR) considers a very low percentage ($\leq 10\%$) of 256^3, which coupled with what has been presented before, allows to explain the gain in CPU time. Nevertheless, at the current state of development of the code (MR), a lot of work remains to be done like taking into account complex geometry, improving performances of multiresolution methods by using adapted data structures and optimized routines, and finally, a high performance distributed parallelism implementation. Once this is achieved, an adaptive mesh approach shall overcome the natural limitations of accuracy and performance of even very performing strategies such as the implemented in (FM), and will allow one to solve entirely the multiscale dynamics of this kind of phenomena.
Finally, increasing the accuracy of the discretization, of the numerical methods and even of the implementation, is not sufficient to generate more precise and predictive simulations. This should be carefully coupled with the development of a more precise modeling of coefficients and boundary conditions, and it is beyond the scope of the present paper.

9.8 Conclusions and Future Work

We have presented for the first time numerical 3D simulations of an ischemic stroke in a realistic brain geometry, based on the model of Dronne *et al.* [DBG06]. Results are encouraging from numerical and medical points of view. This is a first major step towards an usable tool for predicting the evolution of a stroke. The next steps are to improve both numerical performances and modeling. For this, a lot of work remains to be done from the model to practical implementations. Concerning the numerical methods, many parameters in the model are known only with a coarse approximation. Thus, numerical simulations must be conducted to explore the sensitivity of the model to these parameters. The ultimate way to improve the performances is to switch from multithreaded parallelism to distributed parallelism, on massive parallel computers.
From a medical point of view, this model is of the most importance since it could be used to simulate on a realistic human brain geometry several neuroprotective agents aimed at blocking the ischemic cascade and at reducing the ischemic damage. Since the model contains many pharmacological targets (such as ionic transporters, voltage-gated channels, channel-receptors and stretch channels), it could be used to assess and study the effects of various therapeutic agents or associations of therapeutic agents. Moreover, since the model includes both ionic movements through the cells and their diffusion, we will be able to study the effects of these neuroprotective agents both on the severity and on the extension of the damage in each brain area. Developing powerful numerical methods are thus of the most importance to be able to simulate the time and spatial evolutions of these phenomena on a realistic human brain geometry.

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

9.9 Dedicated Splitting and Adaptive Multiresolution: Application to Human Ischemic Stroke

In the previous sections, the splitting strategy detailed in Chapter 4 have been successfully applied to 2D and 3D simulations of a stiff reaction-diffusion system modeling human ischemic stroke with a 19-species detailed chemistry [DBG06], in a realistic brain geometry [DDD+12]. In this part, we reconsider the same model solved this time on simplified brain geometries, with the MR/splitting technique introduced in Chapter 4. Exploiting the splitting configuration, a parallel computing technique was also implemented for the time integration stage in the framework of shared memory computing architectures, in order to reduce computing costs related to "detailed chemistry" features of the model. Some preliminary results have been shown in the previous part in another context without any detail or analysis of the numerical performance of the method. This is the main goal of this part and we will conduct a similar analysis to the one performed for the numerical simulations in Chapter 4 in order to extend the domain of application of the latter numerical strategy to stiffer and more complex configurations.
In what follows, we consider 2D and 3D configurations of the same reaction-diffusion system related to the ischemic stroke model in [DBG06] with $m = 19$ unknowns, recalling that there is no diffusion for 9 variables. The only simplification is that only gray matter has been taken into account in these simulations. All the simulations were performed on an 8 core (2x4) 64 bits machine (AMD Shanghai 2.7 GHz processors) with memory capacity of 32 GB.

9.9.1 2D Configuration

We first consider a computational domain of $[0, 50000]^2$ (μm), and simulate the phenomenon over one hour $t \in [0, 3600]$ (s). As previously done, we will refer to three ways to solve the corresponding semi-discretized system (4.1):

- The *quasi-exact* solution, which considers the coupled reaction-diffusion problem on a uniform mesh, computed by Radau5 with very fine tolerance;
- The *splitting* solution, which uses the RDR Strang $\mathcal{S}_2^{\Delta t}$ scheme with Radau5 for the time integration of the reaction term and ROCK4 for the diffusive part, also on a uniform grid, with tolerances set to $\eta_{Radau5} = 10^{-5}$ and $\eta_{ROCK4} = 10^{-7}$ after numerical experiments; and
- The proposed *MR/splitting* strategy, with the same $\mathcal{S}_2^{\Delta t}$ time integration scheme on an adapted mesh.

Computation of the Splitting Time Step

Taking into account that the solution of the coupled problem, *i.e.*, the quasi-exact solution, with Radau5 is very expensive, and that we do not dispose any theoretical information on the solution of the problem, a preliminary study is required to choose an appropriate splitting time step according to the criteria established in Chapter 4. The propagating nature of the phenomenon justifies the choice of a constant splitting time step. In this particular application, we are looking for a splitting time step Δt that approximates the wavefront speed v in (4.23), within a prescribed accuracy η_v.
Therefore, we perform 1D computations with the splitting solver on a uniform grid of 1024 points in order to define a reference wave velocity v into (4.23). Figure 9.15 (top) shows the propagation of the wavefront along the x-axis across the core of the initial perturbation (see Figure 9.16), and the time evolution of the wave velocities for solutions computed with different

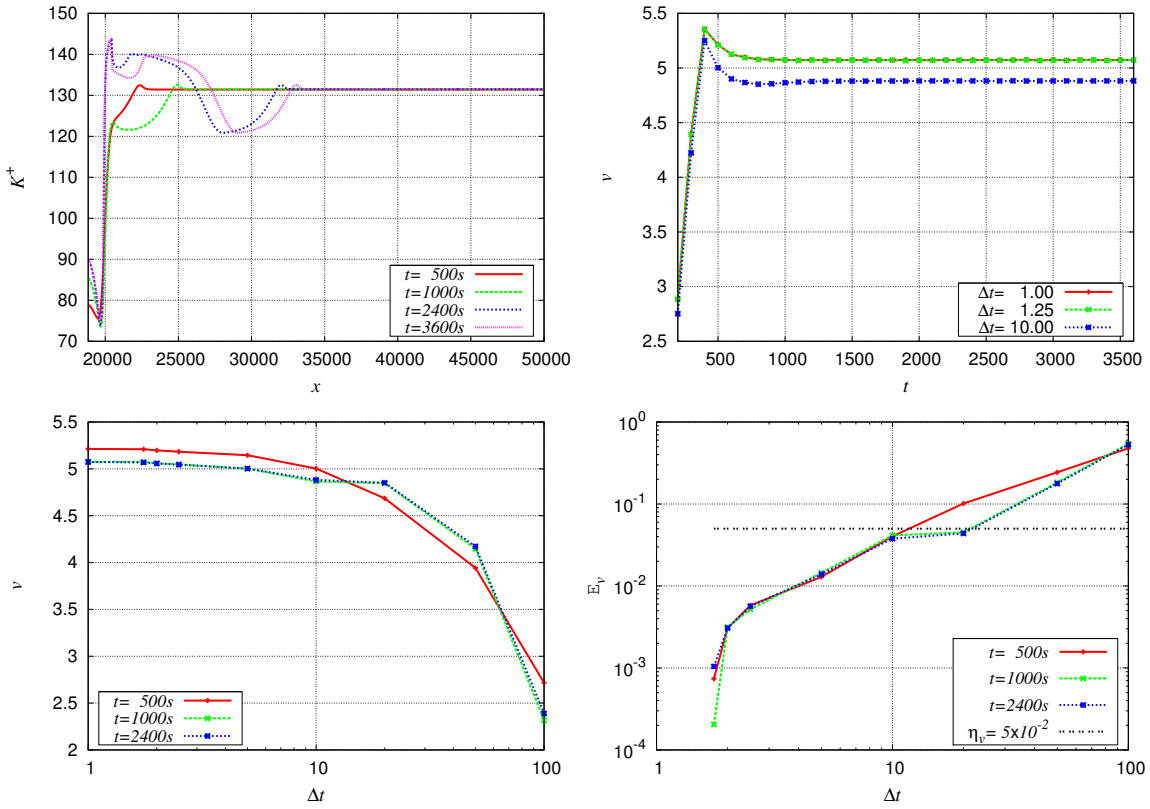

Fig. 9.15: *Stroke model. Top: time evolution of K^+ in the neurons along x-axis (left) and of the wave velocity v (right) for different splitting time steps Δt, computed on a uniform grid of* 1024 *points. Bottom: wave velocity v (left) and velocity error E_v computed by (4.23) at different times t for different splitting time steps Δt.*

splitting time steps. Smaller splitting time steps naturally involve a more accurate description of the global phenomenon, measured in this case by means of the wavefront speed (Figure 9.15 bottom left), and show convergence towards a roughly constant value of $v \approx 5.07$ once the propagating front is fully developed ($t \gtrsim 700\,\mathrm{s}$) and the wave speed becomes relatively constant (Figure 9.15 top right). We take this velocity computed with $\Delta t = 1$ as the reference one, and we choose just for illustrating purposes a rather large splitting time step $\Delta t = 10$ in order to show the potential decoupling of time steps for reaction, diffusion, and the time operator splitting for a given prescribed tolerance. This time step yields a relative velocity error of $\approx 3.8\%$ (Figure 9.15 bottom right), or equivalently an accuracy tolerance of $\eta_v = 5 \times 10^{-2}$ into (4.23).

According to the criterion (4.27) established in Chapter 4 to properly couple the splitting technique with the multiresolution representation, this time step is bounded by the maximum splitting time step given by $\Delta t_{\max} = 28.9$, for $\Delta x = \Delta y = 50000/1024$ and a refined region estimated by $\hat{k} = 3$ in each direction, *i.e.*, one extra cell ($\bar{k} = 1$) as enlarging criterion to predict the propagation of the solution, and two cells contained in the safety layer.

Performance of the Method: Data Compression and Memory Requirements

In order to analyze the performance of the MR/splitting strategy, we consider several values of J, that is, the number of nested dyadic grids that corresponds to a finest space resolution

equivalent to 256^2 ($J = 8$), 512^2 ($J = 9$), 1024^2 ($J = 10$), or 2048^2 ($J = 11$) grid points. For all the simulations, we considered a multiresolution threshold value of $\varepsilon = 10^{-2}$. This value yields a normalized L^2-error $\lessapprox 10^{-2}$ between splitting and MR/splitting solutions for all 19 variables, as previously shown in the numerical comparison of the first part [DDD+12]. Higher spatial discretizations naturally yield a better resolution of both the wave velocity and the dynamics of the wavefront. In particular, we have concluded that at least approximately 1024^2 points are needed to get a reasonably fine description of the phenomenon. Figure 9.16 shows some MR/splitting results for the time evolution of the concentration of K^+ on the corresponding adapted grids, for an equivalent finest grid of 1024^2 points ($J = 10$).

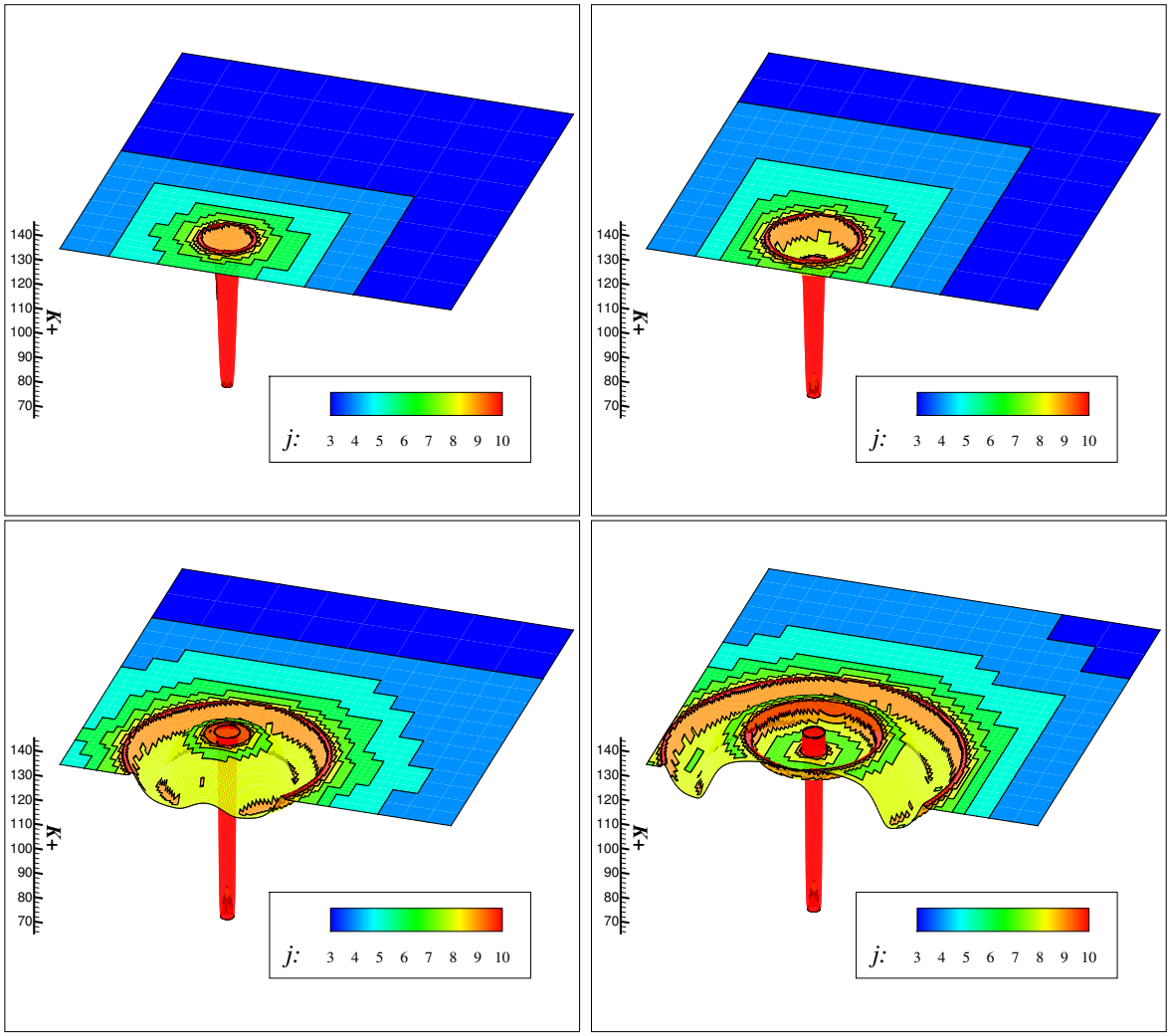

Fig. 9.16: *2D stroke model. K^+ in the neurons at 500 s (top left), 1000 s (top right), 2400 s (bottom left), and 3600 s (bottom right) represented on a dynamic adapted grid corresponding to 1024^2 points at the finest level $J = 10$.*

Figure 9.17 shows the time evolution of the achieved data compressions (DC) defined by (4.39) and the number of points in the adapted grids (AG). Data compression increases with the number of levels as the space scales present in the problem are better discriminated by locally finer spatial discretizations. For a given J, the number of points in the adapted grid AG is increasing linearly in time according to the growing radius with constant speed of the circular

propagation of the wavefront as seen in Figure 9.16. Moreover, at a given time, adding a new grid layer amounts to doubling the number of cells on the adapted grid as if we were refining only in one dimension, also because of the quasi axial symmetry of the phenomenon.

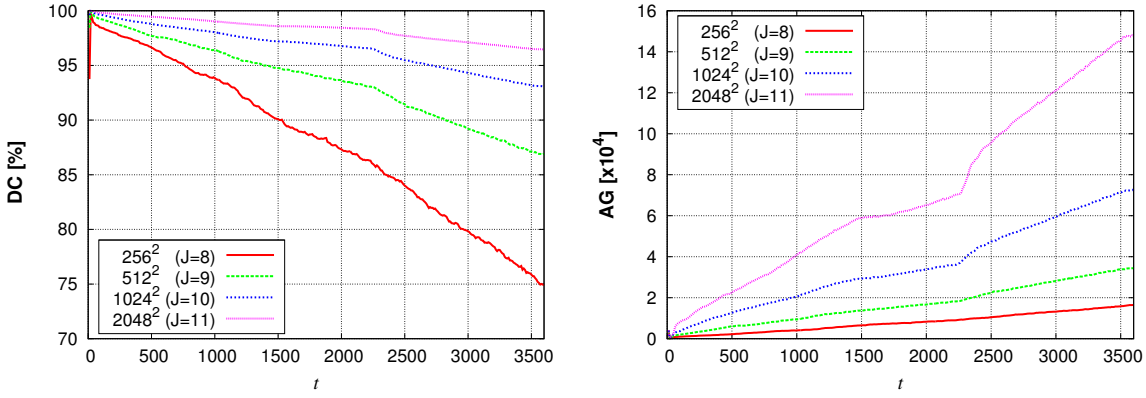

Fig. 9.17: *2D stroke model. Time evolution of data compressions* DC *given by (4.39) (left) and the number of points on the adapted grids* AG *(right) for different finest levels J.*

We now compute the same estimate on the memory requirements of each numerical strategy for a fine spatial resolution of 1024^2, where W_1 and W_2 are the number of unknowns solved, respectively, by Radau5 and ROCK4, and the total number of unknowns is given by $W = 19 \times 1024^2 \approx 1.99 \times 10^7$. The size of the working space L required by each solver is then:

1. Quasi-exact with Radau5: $W_1 = W \approx 1.99 \times 10^7$ and $L = L_1 \approx 1.6 \times 10^{15}$.
2. Splitting: $W_1 = 19$, $W_2 = 10 \times W/19 \approx 1.05 \times 10^7$ and $L = L_1 + L_2 \approx 8.4 \times 10^7$.
3. MR/Splitting with $\varepsilon = 10^{-2}$: $W_1 = 19$, $W_2 = 0.07 \times 10 \times W/19 \approx 7.34 \times 10^5$, and $L = L_1 + L_2 \approx 5.9 \times 10^6$; with a minimum data compression of 93%.

Considering a standard platform on which each double precision value is represented by 64 bits, each solver shall require 90.9 Pb, 5.0 Gb, and 360.1 Mb. For standard computational resources, an implicit resolution with Radau5 is completely out of reach. These expensive memory requirements are strongly reduced with a splitting strategy but further reductions are achieved by adding a multiresolution adaptive procedure.

Performance of the Method: Parallelization and CPU Times

Table 9.5 summarizes the CPU times (CT) of the simulations, performed with 8 cores in parallel, and the gain of parallelization (GP) which is defined as the ratio between the CPU time given by one single processor and the 8 cores in parallel. We obtain rather high GPs ($\max \mathrm{GP} = 8$) even though only the time integration procedure is parallelized. Actually, for this kind of highly localized and stiff problem the multiresolution operations take normally less than 5 % of the total time consumption. Parallelization of the reaction is practically optimal in the context of shared memory architectures because each core takes a new node immediately after finishing with the previous one, without any need of synchronizing or exchanging data with the other cores.
The integration of the stiff reaction problem is always much more expensive than any other computation, in terms of CPU time. The latter can be inferred from Figure 9.18 (left), where the ratios in time between the CPU times used to integrate the reaction and the diffusion

Table 9.5: *2D stroke model. CPU times* (CT) *in minutes and gain of parallelization* (GP)*, for several finest grids* (FG)*.*

FG	256^2	512^2	1024^2	2048^2
CT	10.92	31.37	82.45	214.43
GP	7.62	7.50	7.55	7.52

problems are shown for different spatial discretizations. Whenever we add a new layer grid to the computational domain, the reaction CPU integration time per number of active points remains almost the same, as seen in Figure 9.18 (right), because most of the new points will be naturally added at the wavefront where high chemical activity is present. Nevertheless, there is an overhead in the time consumption for two successive finest grid configurations J, as seen in Table 9.5. We can measure an increasing ratio of ~ 2.6, slightly higher than the ratio of 2 that we should have obtained taking into account the corresponding increasing ratio of the size of the adapted grids.

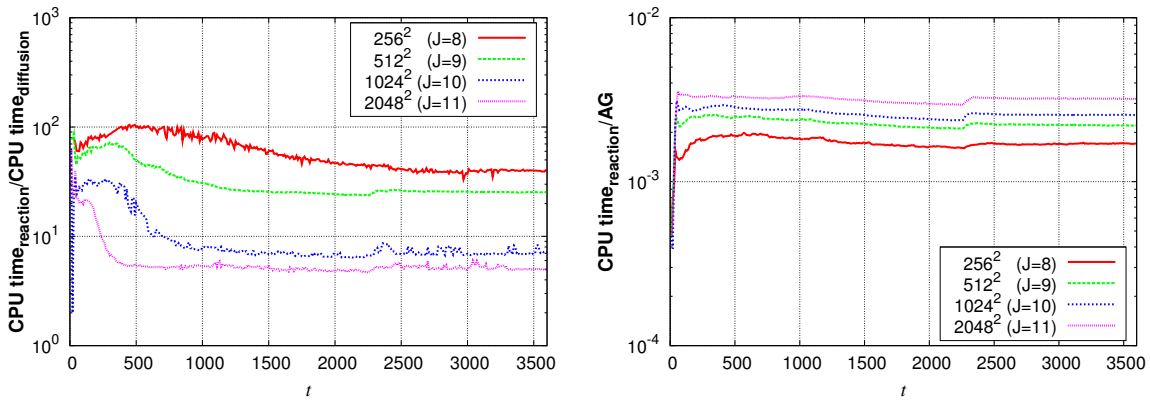

Fig. 9.18: *2D stroke model. Time evolution of the ratio between CPU times for the integration of the reaction and diffusion problems (left), and the CPU time per number of points in the adapted grid for the reaction solution (right) for different finest levels J.*

Actually, as shown in Figure 9.18, the integration of the diffusion problem becomes more expensive when the spatial discretization increases from 256^2 to 2048^2. This is because the diffusion time step Δt_D, dynamically chosen to guarantee the tolerance accuracy η_{ROCK4}, goes, respectively, from ~ 1.25 to ~ 0.27. Furthermore, the matrix-vector products computed by ROCK4 become also more expensive since the matrix representation of the discretized diffusion operator is less sparse when the computational domain includes more grid levels. This is because we locally reconstruct the values at the interface of grids with different discretization (by means of the phantom cells) to guarantee more accurate computations (see chapters 3 and 4).

Decoupling of the Time Scale Spectrum

The latter values of Δt_D for different J imply, respectively, from 8 to 36 diffusion time steps within the splitting time step $\Delta t = 10$, for which most of the time the minimum of $s = 5$

stages are needed to guarantee the stability of the method, according to (4.12).

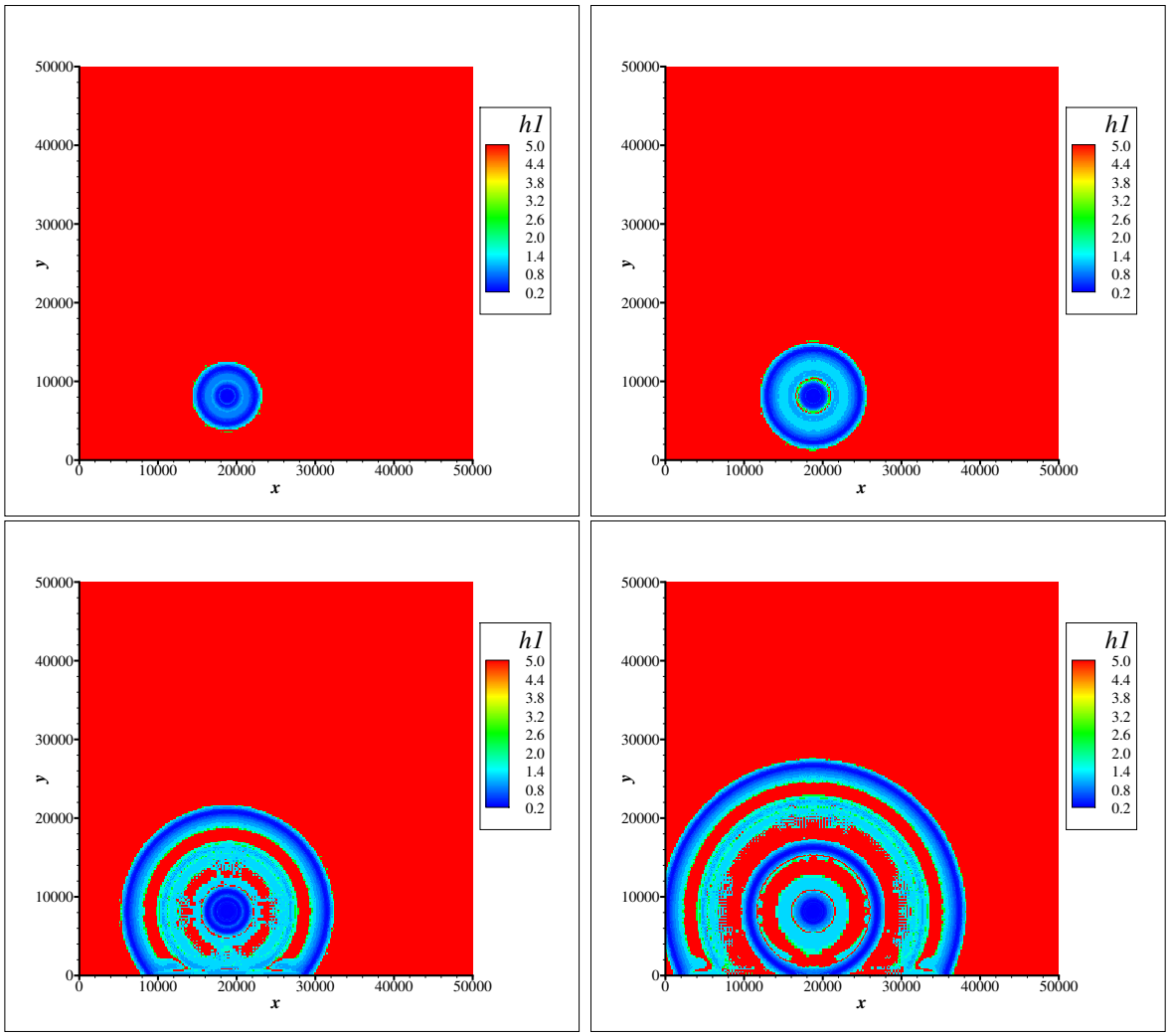

Fig. 9.19: *2D stroke model. Local reaction time steps* $h1$ *at* $500\,s$ *(top left),* $1000\,s$ *(top right),* $2400\,s$ *(bottom left), and* $3600\,s$ *(bottom right) within a half-splitting time step* $\Delta t/2 = 5$*, represented on a dynamic adapted grid corresponding to* 1024^2 *points at the finest level* $J = 10$*.*

Figure 9.19 shows the corresponding local reaction time steps averaged within $\Delta t/2$ for each point. The imposed tolerance for the reaction resolution η_{Radau5} involves time steps varying from ~ 0.2 to $\Delta t/2 = 5$ for points located, respectively, in the neighborhood of the reactive fronts and in regions of reduced chemical activity . Considering the extension of the refined regions, most of the computational domain is integrated with reaction time steps 25 time larger than the smallest time steps at the wavefronts. In this way, an effective decoupling of the time scale spectrum is achieved for the reaction, the diffusion, and the splitting schemes with an important gain of efficiency, whenever this is permitted by the physics of the problem, so that different physical or numerical time scales associated with each problem can be isolated and treated independently by each numerical method.

9.9.2 3D Configuration

Let us give more details on the 3D configuration previously considered in the first part, with the same parameters of the 2D case, in a spatial region of $[0, 50000]^3$ (μm).

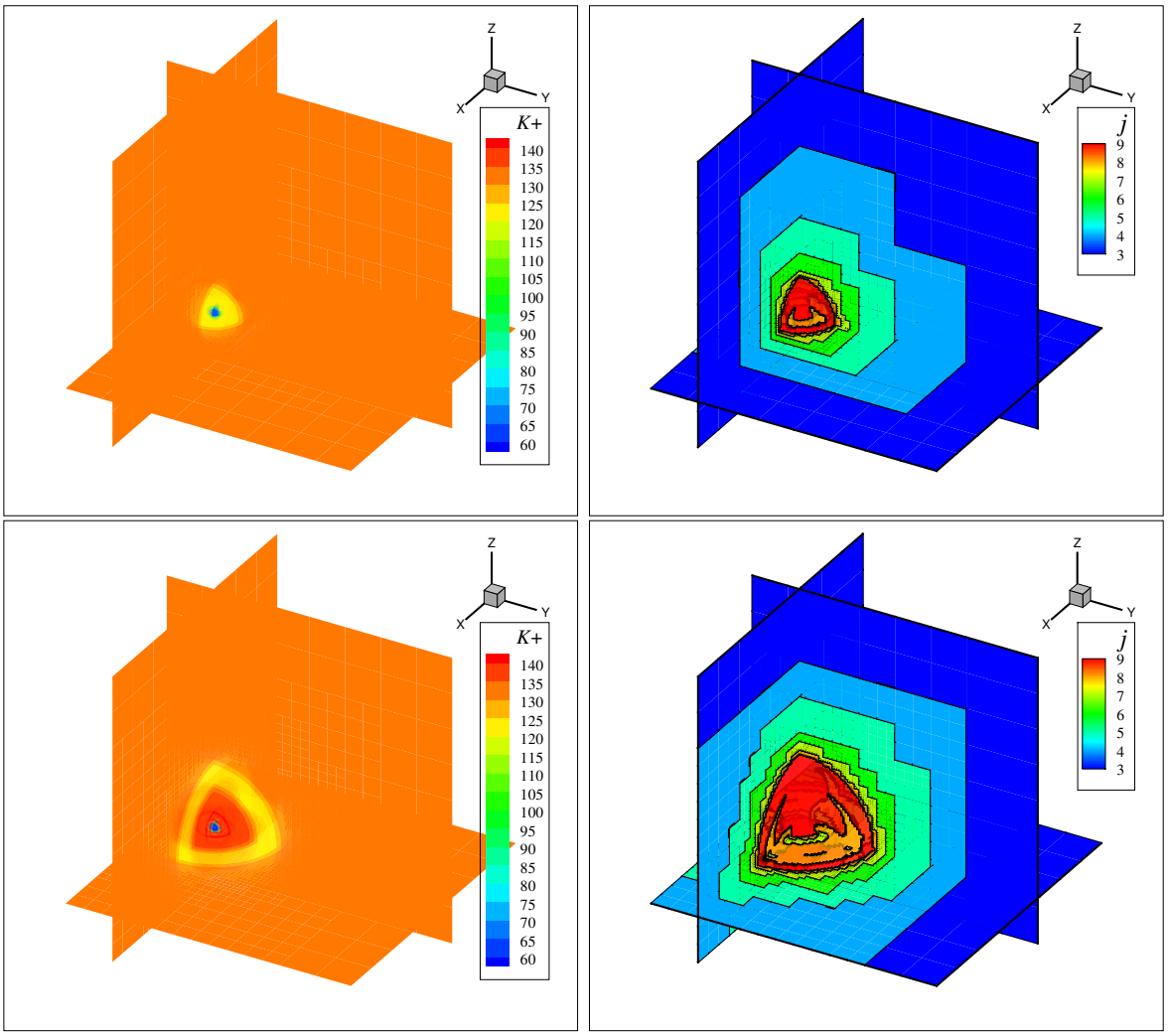

Fig. 9.20: *3D stroke model. K^+ in the neurons (left) and dynamic adapted grids (right) corresponding to 512^3 points at the finest level $J = 9$ at 1000 s (top) and 2000 s (bottom).*

In order to explore the feasibility and potential advantages of the method, we consider a larger computational domain given by $J = 9$ nested dyadic grids corresponding to 512^3 points in the finest grid J. For the 256^3 configuration with $J = 8$ shown in the first part, the achieved data compression DC were of 98.84 %, 95.27 %, and 87.23 % at times 1080 s, 2160 s, and 3600 s, respectively. The computing time CT was of about 16.68 hours with a gain of parallelization GP of 7.02.

Figure 9.20 shows the concentration of K^+ in the neurons and the corresponding adapted grids at 1080 s (DC = 99.14 %) and 2520 s (DC = 94.94 %) for the 512^3 case; CT $\approx$ 45.88 hours for $t \in [0, 2520]$ (s) and GP = 7.14. Longer simulations times yielded larger simulation domains which were no longer feasible with the considered computing resource and the current state of development of the code.

Performing the same comparison concerning memory requirements, the total number of unknowns for the 512^3 case is $W = 19 \times 512^3 \approx 2.55 \times 10^9$, and the global size of L required by

each solver is:

1. Quasi-exact with Radau5: $W_1 = W \approx 2.55 \times 10^9$ and $L = L_1 \approx 2.6 \times 10^{19}$.
2. Splitting: $W_1 = 19$, $W_2 = 10 \times W/19 \approx 1.34 \times 10^9$ and $L = L_1 + L_2 \approx 1.1 \times 10^{10}$.
3. MR/Splitting with $\varepsilon = 10^{-2}$: $W_1 = 19$, $W_2 = 0.04 \times 10 \times W/19 \approx 5.37 \times 10^7$, and $L = L_1 + L_2 \approx 4.3 \times 10^8$; with a minimum data compression of 96%.

Therefore, each solver shall require at least 1.4 Zb, 655.7 Gb, and 25.6 Gb of memory capacity.

9.9.3 Concluding Remarks

The present numerical study has mainly illustrated two aspects of the numerical strategy established in Chapter 4:

- The treatment of highly stiff problems, given by an important number of variables related by detailed and computationally expensive chemical terms.
- A straightforward parallelization in shared memory architectures, that turns out to be very efficient at least for problems for which the computational cost is dominated by the numerical solution of the source term.

As a consequence, these highly stiff phenomena modeling complex chemical mechanisms, previously out of reach, can be successfully simulated over large computational domains with standard computational resources.
Nevertheless, there are still many modeling and numerical points that require further improvement to efficiently conduct numerical simulations of human ischemic brain strokes, as detailed in the first part of this chapter. Regarding the present numerical strategy, we can cite, for instance, two key issues that need to be taken into consideration:

- The development and introduction of numerical tools that will allow us to consider complex computational domains to numerically represent realistic brain geometries.
- The implementation of efficient techniques to parallelize expensive computations over several processors, in a distributed or hybrid distributed/shared memory framework.

Both points still require an important amount of work, and constitute current topics of our research.

Chapter 10

Numerical Simulation of Combustion Fronts

We consider in this chapter the implementation of the numerical strategy introduced in chapters 4 and 5, for the numerical simulation of flames issued from combustion applications. This study will be performed in a classical context of laminar flames interacting with vortex structures, for propagating premixed flames and self-ignition of reactive mixtures. The hydrodynamics is decoupled from the transport equations by adopting a standard thermo-diffusive approach. In this context and for the considered applications, adaptive space meshing is advantageous because of the presence of localized fronts, whereas the important transient phases due to the imposed velocity fields as well as sudden physical variations given, for instance, by the ignition of a mixture, require an adequate time adaptation in order to efficiently describe these phenomena. The coupling of the space and time adaptive techniques enunciated in Chapter 5, turns out to be a powerful tool to cope with these problems.
In the first part of this chapter, we give a brief recapitulation of splitting methods in the literature for combustion applications. Then, we detail the basic mathematical formulation for the problems studied in these applications. The time/space adaptive technique is introduced in the next part as well as the extension to reaction-diffusion-convection systems. The two last parts are devoted to the numerical simulation of propagating premixed flames in 2D/3D, and the ignition and propagation of diffusion flames in 2D. Some preliminary results are reported considering more complex source terms as in previous Chapter 9, given this time by detailed chemical kinetics. Part of these results were presented during the Finite Volumes for Complex Applications VI International Symposium, Prague, Czech Republic (2011) [DMDD11], and published in the Annual Research Briefs 2011 of the Center for Turbulence Research, Stanford University, USA [DMD+11a].

10.1 Time Operator Splitting for Combustion Problems

Numerical simulation of combustion fronts usually involves numerous numerical difficulties arising from the broad spectrum of time and space scales present in the problem, and the induced stiffness of the governing equations. In this context and with the continuous development of dedicated stiff implicit methods, several numerical strategies consider the treatment of stiff terms usually related to detailed chemical kinetics, with these solvers instead of more classical and less efficient explicit or linearized implicit methods. Since a simultaneous implicit treatment of reaction, diffusion, and convection terms involves often prohibitive computational costs, a more efficient numerical alternative considers a combination of implicit and

explicit schemes, *i.e.*, the so-called IMEX methods. Similarly, we have also seen that time operator splitting techniques offer the same kind of advantage with usually a more reduced computational complexity, but with the additional constraint that appropriate criteria must be introduced to efficiently decouple the physical phenomena and control the splitting errors. Splitting methods were thus largely implemented and studied for combustion problems (see, *e.g.*, [GPMD88, DL95a, YP98, OB01, SLGS03, RP08]), which resulted in several numerical strategies over the past years.

Taking advantages of the particular split configuration of these multi-scale problems, Knio *et al.* introduced in [KNW99] an efficient stiff operator splitting technique for low Mach formulation of reacting flows[1]. These authors considered a split numerical solution for the energy and species concentrations transport equations, which included detailed chemical kinetics and transport parameters. In a previous work [NWK98][2], the same authors introduced an IMEX technique that considered implicit integration of the reactive term by the dedicated stiff solver VODE [BBH89], based on multi-step BDF formulae [Gea71]. The latter scheme achieved much better performances than standard explicit schemes by considering time steps much larger for the reactive term. Nevertheless, a strong limitation was given this time by the very restrictive stability conditions for the explicit treatment of the diffusive term for fine spatial discretizations The stability constraints for the diffusion problem became then of the same order of those issued from an explicit treatment of the reaction (about 2 ns).
In order to relieve this constraint, the splitting technique introduced in [KNW99] was built such that several diffusion steps given by the stability criteria, can be performed within the splitting time step. The method considered then a splitting time step multiple of the stability diffusion time scale, and a Strang scheme with half step integrations of the diffusion and a centered full step reaction integration. The reaction problem was solved by VODE, whereas the convection and diffusion problems were integrated with a second order multi-step Adam-Bashforth scheme. In this way, numerical experiments showed that the method was globally of order 2 for the range of time steps considered (less than 200 ns), where the splitting time steps were suggested to be chosen as a small fraction of the physical time scale of the flow, to ensure negligible splitting errors. In the numerical illustrations, a sufficiently small splitting step was consequently chosen for which several diffusion substeps were evaluated (from 2 to 32), as well as different accuracy tolerances for the stiff solver. The resulting global error of the scheme was then dominated either by the stiff implicit solution for small splitting (and diffusion) time steps and large accuracy tolerances, or by the second order explicit integration for larger splitting time steps. In the first case when the implicit errors were comparable with the remaining second order errors, the global order of the method is likely lower than 2. These conclusions were drawn in [KNW99] out of numerical tests performed on a simplified reaction-diffusion system, and on a 1D methane-air premixed flame. A 2D case was also considered for a counter rotating vortex pair interacting with the premixed flame, for which better efficiencies were found with respect to the previous IMEX approach [NWK98], mainly because of the larger time evolution steps.
The same solver was later enhanced in [NK05] with the inclusion of a second order RKC scheme [vdHS80, Ver96] for the diffusion problem, and an extrapolation procedure for the computation of the transport coefficients. The use of approximate Jacobians was also evaluated for the stiff implicit solver. The RKC solver with extended stability domain, briefly described in Chapter 2, allowed larger splitting time steps for the same previous splitting technique. The second order

[1] See, *e.g.*, [PV05] for the modeling equations.
[2] In particular, the reader can find in [NWK98, KNW99] a detailed state of the art of numerical methods considered at the time for the numerical simulation of combustion fronts.

of the scheme was shown to be conserved for the previous range of splitting time steps (this time for less than 100 ns), and the introduction of the second order RKC scheme showed to keep the same previous numerical behavior in terms of global errors, but with larger splitting time steps. In this way, the time integration errors were once again associated with the accuracy of the inner implicit and explicit solvers, and thus the splitting errors were supposed negligible, even though the choice of the splitting step was not explicitly discussed. In all these works, the research effort was mainly focused on the numerical solution of the reaction and diffusion problems. Stability conditions for the explicit time discretization of diffusive terms are usually related to Δx^{-2}, compared with Δx^{-1} for the convective ones. Furthermore, these works considered detailed transport coefficients which display multi-scale and potentially stiff features (see, *e.g.,* [Gio99]).
An interesting study in [NK05] on splitting errors showed that a time adaptive stiff solver effectively handles the initial fast transients introduced in the reactive step after the diffusion step due to the splitting procedure, as previously reported and treated in [VBvLS96, SVdZ+98, VSBH99]. With these results, these authors justified their diffusion ending splitting scheme by arguing that the adaptive stiff solver succeeds to guarantee the prescribed accuracy even with important transient phases, although this will not be the case after the second and last diffusion half step. Nevertheless, the global errors are piloted by the inner time integrators in this implementation, so that it becomes more difficult to relate this approach with the previous theoretical results [Spo00, DM04] discussed in Chapter 1, for which the split subproblems were assumed to be exactly solved. Notice that these transients imply naturally an overhead in the stiff solvers as reported also in [KNW99]. As in [KNW99], the number of constant diffusion substeps defines the splitting time step, and both this number and the number of stages of the RKC scheme needed to be defined in advance. The time stepping features of RKC with error control [SSV97] were not considered.
More recently in [SPN06b], Singer *et al.* coupled the previous splitting scheme of [NK05] with ISAT, an adaptive chemistry tabulation technique for detailed chemical kinetics [Pop97], to handle the reactive terms. The capabilities of this coupling were first conducted in [SP04]. Numerical experiments were thus performed in [SPN06b] in the same previous context of [KNW99, NK05], to analyze the approximation errors introduced by ISAT and to verify the second order of the method. Splitting time steps should be considered sufficiently small and therefore were not discussed in the paper as well as the order reduction appreciated for splitting time steps larger than $\sim$ 200 ns. Order reductions were previously reported in [GPMD88, DL95a, YP98] for flames with complex chemistry in combustion applications. The numerical strategy proposed in [SPN06b] showed to be very performing, and implemented hybrid parallel computing techniques for distributed and shared memory architectures that yield important speed-ups in the simulations, as reported and analyzed in [SPN06a]. Complementary studies in [RP08] introduced alternative techniques like previously in [NK05], to reduce the computational cost of the transport substeps while maintaining in practice the second order accuracy of the Strang scheme.

Another efficient numerical strategy also in the context of low Mach regimes that considers operator splitting techniques, was introduced by Day & Bell in [DB00]. Even though this scheme included some similar ideas to those of [NWK98, KNW99], it differs mainly in the projection method used to solve the combined momentum and transport equations, and that the AMR technique introduced in [Ber82, BO84, BBSW94], was implemented. The operator splitting considered a Strang scheme with half steps of reaction integration with VODE [BBH89], and a combined convection-diffusion centered full step. The latter step is solved on the adapted mesh by considering local CFL time steps, according to the grid size and with a scheme developed

in [ABC+98, PHB+98] in the framework of AMR techniques for reacting flows. Numerical experiments in [DB00] showed convergence order of, or close to 2, by using reference solutions computed with the Premix code from the Chemkin library [KGM+98]. Further validations were conducted through qualitative analysis of particular physical features of the models under study. In these configurations the accuracy tolerance of the stiff solver is set as small as possible (*i.e.,* 10^{-16} in [DB00]), and the global time evolution step is related to the local time stepping procedure based on local CFL conditions rather than accuracy [BDA+06]. With these bases, further developments in terms of algorithm implementation and parallel computing techniques led to the effective simulation of 3D turbulent premixed flames with detailed chemistry (see, *e.g.,* [BDG02, BDA+06]) and outstanding achievements for laboratory-scale turbulent flames (see, *e.g.,* [BDS+05, BDG+07]). For instance, a 3D turbulent flame with complex chemistry was simulated in [BDG+07] in a computational domain of $[7.5, 5, 10]$ (cm) over three grid levels, with a corresponding resolution of $480 \times 320 \times 640$ in the finest mesh.

The numerical methods previously described represent some examples of splitting implementations for combustion applications, and reflect the interest of such techniques to overcome classical restrictions of computational resources. Such an approach allows a suitable choice of numerical time integration solvers and a straightforward coupling with other techniques like chemistry tabulation or grid adaptation procedures, with important gains of numerical performance due to the reduced computational complexity, and thus higher degree of liberty with respect to other numerical methods. In this context and with this background, the present work has presented in Chapter 4 a new splitting scheme for this kind of multi-scale problem, that further develops and introduces some new elements to the classical approach.
Compared with previous works, this scheme considers a new approach in the construction of splitting schemes in which the time integration errors are uniquely related to the splitting errors, even for large splitting time scales, based on mathematical studies conducted mainly in [DM04, DDLM07]. The underlying idea is to decouple time integration errors by choosing high order dedicated methods for the split subproblems, and thus the global error is mainly piloted by the splitting time step, defined according to the decoupling capabilities of the phenomenon. This novel point of view allows us to considerably reduce the number of simulation parameters and to easily generalize the numerical methodology. Additionally, the numerical accuracy of the simulation and the corresponding choice of the splitting time step, which were often not explicitly detailed or directly related to the fastest numerical or physical time scale in the classical literature, become the core of this numerical strategy. The independent choice of the splitting time step settles the numerical accuracy of the simulation and yields an effective decoupling of the time scale spectrum depending on the physics of the problem, with important computational gains as illustrated in previous chapters. The latter procedure may be considerably eased by the dynamic splitting error control introduced in Chapter 5, which extend the previous strategy with constant splitting time step as in the referred literature, to a more general framework potentially given by highly unsteady physics. With the simulation criteria introduced in this work, we look for a further decoupling between the time integration errors and the spatial errors introduced by the compressed spatial representations, taking into account that the mathematical background of multiresolution schemes [Har95, CKMP03, Mül03] offers a better quantification of such representation errors.

In the following, we implement the proposed numerical strategy for the simulation of laminar flames interacting with vortices in a thermo-diffusive approach. These configurations were extensively used in the combustion domain to describe and to study several phenomena (see, *e.g.,* [RTRC00]), and even compared with experimental results (see, *e.g.,* [TRR+96,

RRTC99]). The first 2D/3D simulations with adaptive multiresolution for combustion problems were performed by Roussel *et al.* in [RS02, RS05b, RS06], also in a thermo-diffusive approach with standard explicit time integration solvers.

10.2 Laminar Flames Coupled with Vortex Structures

In this study, we are concerned with the numerical simulation of diffusion and premixed flames interacting with vortex structures. We describe in this part the general configuration for laminar flames. In a first step and for the sake of simplicity, the chemistry is modeled by a global, single step, irreversible reaction given by

$$\nu_{\mathrm{F}}\mathrm{F} + \nu_{\mathrm{O}}\mathrm{O} \rightarrow \nu_{\mathrm{P}}\mathrm{P}, \tag{10.1}$$

where ν_k, $k = \mathrm{F}, \mathrm{O}, \mathrm{P}$, stand for the stoichiometric coefficients for the fuel F, the oxidizer O, and the combustion products P. The latter reaction is usually modeled by an Arrhenius law as we shall see in the following applications. Moreover, the following standard modeling assumptions are also considered throughout this study:

1. Mass diffusion velocities of chemical species are expressed by Fick's law.
2. Thermal diffusion of species (Soret-Dufour effect) is neglected.
3. Different species have constant and equal diffusion coefficients with respect to the mixture, noted D.
4. Constant pressure specific heats of all species are constant and are given by the same value c_p.
5. Lewis numbers corresponding to all species are equal to 1.
6. The rate of pressure change in time is negligible.
7. Density variations associated with chemical heat release are neglected.

The assumption 7 is commonly known as the thermo-diffusive approach in laminar flame theory. In practice, this constant-density character decouples the velocity field computation from the determination of species mass fractions and temperature. Known solutions of incompressible Navier-Stokes equations may then be imposed, and the problem is reduced to solving the following species and energy balance equations:

$$\left.\begin{aligned}
\partial_t Y_{\mathrm{F}} + \mathrm{v}\cdot\partial_{\mathrm{x}} Y_{\mathrm{F}} - D\,\partial^2_{\mathrm{x}} Y_{\mathrm{F}} &= -\frac{\nu_{\mathrm{F}} W_{\mathrm{F}}}{\rho}\dot{w},\\
\partial_t Y_{\mathrm{O}} + \mathrm{v}\cdot\partial_{\mathrm{x}} Y_{\mathrm{O}} - D\,\partial^2_{\mathrm{x}} Y_{\mathrm{O}} &= -\frac{\nu_{\mathrm{O}} W_{\mathrm{O}}}{\rho}\dot{w},\\
\partial_t Y_{\mathrm{P}} + \mathrm{v}\cdot\partial_{\mathrm{x}} Y_{\mathrm{P}} - D\,\partial^2_{\mathrm{x}} Y_{\mathrm{P}} &= \frac{\nu_{\mathrm{P}} W_{\mathrm{P}}}{\rho}\dot{w},\\
\partial_t T + \mathrm{v}\cdot\partial_{\mathrm{x}} T - D\,\partial^2_{\mathrm{x}} T &= \frac{\nu_{\mathrm{F}} W_{\mathrm{F}} Q}{\rho c_p}\dot{w},
\end{aligned}\right\} \tag{10.2}$$

$\mathrm{x} \in \mathbb{R}^d$, where W_k is the molar mass, and Y_k the corresponding mass fraction for $k = \mathrm{F}, \mathrm{O}, \mathrm{P}$. Variable T accounts for the temperature, and the reaction rate of progress $\dot{w}$ is related to the rate of consumption of fuel $\dot{w}_{\mathrm{F}}$, oxidizer $\dot{w}_{\mathrm{O}}$, and products $\dot{w}_{\mathrm{P}}$ by

$$\dot{w} = -\frac{\dot{w}_{\mathrm{F}}}{\nu_{\mathrm{F}}} = -\frac{\dot{w}_{\mathrm{O}}}{\nu_{\mathrm{O}}} = \frac{\dot{w}_{\mathrm{P}}}{\nu_{\mathrm{P}}}. \tag{10.3}$$

The heat release per unit mass of fuel Q is defined by

$$Q = h_{\mathrm{F}} + \frac{\nu_{\mathrm{O}} W_{\mathrm{O}}}{\nu_{\mathrm{F}} W_{\mathrm{F}}} h_{\mathrm{O}} - \frac{\nu_{\mathrm{P}} W_{\mathrm{P}}}{\nu_{\mathrm{F}} W_{\mathrm{F}}} h_{\mathrm{P}}, \tag{10.4}$$

where h_k, $k = \mathrm{F}, \mathrm{O}, \mathrm{P}$, is the corresponding species enthalpy. If we consider also a non-reacting diluent, noted by index N, the following equation must be included in (10.2):

$$\partial_t Y_{\mathrm{N}} + \mathrm{v} \cdot \partial_{\mathrm{x}} Y_{\mathrm{N}} - D\, \partial_{\mathrm{x}}^2 Y_{\mathrm{N}} = 0. \tag{10.5}$$

In any case, mass fractions verify by definition

$$Y_{\mathrm{F}} + Y_{\mathrm{O}} + Y_{\mathrm{P}} + Y_{\mathrm{N}} = 1. \tag{10.6}$$

In the thermo-diffusive approach, the velocity field $\mathrm{v}(\mathrm{x}, t)$ is usually computed analytically and imposed into (10.2). Throughout this study, we consider a 2D vortex configuration which features an azimuthal velocity of the form:

$$v_\theta(r,t) = \frac{\Gamma}{2\pi r}\left(1 - \mathrm{e}^{-r^2/4\nu t}\right), \tag{10.7}$$

where Γ denotes the vortex circulation; $r(x,y)$, the distance to the vortex center; and ν, the kinematic viscosity. This velocity field has a viscous core with a typical dimension of $R_\nu \approx (\nu t)^{1/2}$. Inside the core the velocity increases linearly with the radius, and the fluid rotates like a solid body:

$$v_\theta(r,t) \approx \frac{\Gamma r}{8\pi\nu t}, \qquad r < R_\nu. \tag{10.8}$$

A fast decay occurs immediately outside the core, whereas at large distances the flow approaches that of an ideal line flow:

$$v_\theta(r,t) \approx \frac{\Gamma}{2\pi r}, \qquad r \gg R_\nu. \tag{10.9}$$

10.3 Time/Space Adaptive Technique

For the previous reaction-diffusion-convection system given by (10.2), we implement the MR/ splitting strategy introduced in Chapter 4, complemented by the adaptive splitting technique introduced in Chapter 5. We recall that the main idea is to independently consider high order dedicated methods for each subsystem to handle the fastest physical/numerical scales associated with each one. The solution is then reconstructed by the splitting scheme that decouples the global physics within a prescribed accuracy. Additionally, the multiresolution analysis yields adaptive mesh representations with important gains of computational performance.
A second order Strang scheme is considered for the general problem (10.2):

$$\mathcal{S}^{\Delta t} U_0 = \mathcal{R}^{\Delta t/2} \mathcal{D}^{\Delta t/2} \mathcal{C}^{\Delta t} \mathcal{D}^{\Delta t/2} \mathcal{R}^{\Delta t/2} U_0, \tag{10.10}$$

with $U = (Y_{\mathrm{F}}, Y_{\mathrm{O}}, Y_{\mathrm{P}}, Y_{\mathrm{N}}, T)^T$, and splitting time step Δt. The operators $\mathcal{R}$, $\mathcal{D}$, and $\mathcal{C}$ indicate, respectively, the numerical integration of the reaction, diffusion, and convection problems. The splitting time steps are dynamically computed by

$$\Delta t^{\mathrm{new}} = \upsilon\, \Delta t \sqrt{\frac{\eta}{\left\| \mathcal{S}^{\Delta t} U_0 - \widetilde{\mathcal{S}}^{\Delta t} U_0 \right\|}}, \tag{10.11}$$

where $\widetilde{\mathcal{S}}^{\Delta t}$ stands for the embedded and lower order Strang splitting method, introduced in Chapter 5.
The reaction and diffusion problems are solved, respectively, by the Radau5 and ROCK4 solvers as in the previous chapters. An explicit high order in time and space, one-step monotonicity preserving scheme, *OSMP*, developed by Daru & Tenaud in [DT04], is implemented as convective scheme. It combines monotonicity preserving constraints for non-monotone data to avoid extrema clipping, with TVD features to prevent spurious oscillations around discontinuities or sharp spatial gradients. Considering its explicit character, standard CFL stability restrictions are imposed within each splitting time step Δt. In this way, adaptive time stepping is considered for all three operators in (10.10) within the corresponding splitting time steps. Although this dynamic step size selection is made within a prescribed accuracy tolerance for Radau5 and ROCK4, this is not currently the case for the convective scheme for which time stepping is done based on stability constraints. As previously stated and considered in previous works (see, *e.g.*, [NK05]), we have focused our attention on the numerical solution of stiff reaction-diffusion systems[3]. We have thus extended the numerical strategy to reaction-diffusion-convection problems in which the main constraint for the convective term is given by the small time steps resulting from stability conditions related to quite fine spatial discretizations. A convective scheme that guarantees at least second order should be considered, such that the numerical errors associated with the convection problem are expected to be negligible in front of the splitting errors taking into account the small values of the convection time steps[4]. The introduction of the convection problem, and the OSMP scheme in the numerical strategy, is discussed in the following.

10.3.1 Time Integration of the Convection

We are concerned with a general convection problem issued from the time operator splitting scheme:

$$\partial_t U_C + \partial_{\mathrm{x}} \cdot G(U_C) = 0, \qquad \mathrm{x} \in \mathbb{R}^d,\ t > 0, \tag{10.12}$$

with initial data $U_C(0,\cdot) = U_0(\cdot)$, where $U_C : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^m$, $U_0 \in \mathbb{R}^m$, and $G(U_C) = (G_i(U_C))_{i=1,\ldots,d}$ with $G_i : \mathbb{R}^m \to \mathbb{R}^m$. Taking into account that a general implicit solution of problem (10.12) is usually expensive in terms of computational resources, an explicit time integration is preferred. Moreover, explicit schemes are usually easy to implement and many performing and dedicated methods exist (see, *e.g.*, [LeV92, LeV02]). A review and comparison of different schemes can be found, for instance, in [DT01, LW03]. Nevertheless, a common drawback of these techniques is that the specific accuracy and stability constraints, deduced usually for 1D configurations, become often difficult to extend to multi-dimensional cases, and require more sophisticated developments similar to those encountered in coupled multi-dimensional schemes. In this context, one way to keep the original 1D schemes with often important gains in efficiency, considers *dimensional splitting* techniques (see, *e.g.*, [HV03]).
We define then the time operators $\mathcal{C}_x^t$, $\mathcal{C}_y^t$, and $\mathcal{C}_z^t$ associated, respectively, with the solutions at some time t of the 1D problems:

$$\left.\begin{aligned} \partial_t U_C + \partial_x G_1(U_C) &= 0, \\ \partial_t U_C + \partial_y G_2(U_C) &= 0, \\ \partial_t U_C + \partial_z G_3(U_C) &= 0, \end{aligned}\right\} \tag{10.13}$$

[3] In Chapter 2 we have only considered time integration solvers potentially suitable for reaction or diffusion problems in the context of PDEs.
[4] We will see in the following that the convective time step is at most equal to half a splitting time step.

issued from (10.12) for the x-, y- and, z-direction. A first order approximation is given, for instance, by

$$\mathcal{C}^{\Delta t_C} U_0 = \mathcal{C}_z^{\Delta t_C} \mathcal{C}_y^{\Delta t_C} \mathcal{C}_x^{\Delta t_C} U_0; \tag{10.14}$$

whereas a more suitable symmetric second order Strang scheme [Str68] considers, for example,

$$\mathcal{C}^{\Delta t_C} U_0 = \mathcal{C}_x^{\Delta t_C/2} \mathcal{C}_y^{\Delta t_C/2} \mathcal{C}_z^{\Delta t_C} \mathcal{C}_y^{\Delta t_C/2} \mathcal{C}_x^{\Delta t_C/2} U_0, \tag{10.15}$$

where the convection steps Δt_C are limited by the stability restrictions of the numerical scheme. Since the latter constraints take usually into account the spatial discretization steps as in the standard CFL condition (10.30) (proportional to Δx^{-1}), a better solution considers

$$\mathcal{C}^{2\Delta t_C} U_0 = \mathcal{C}_x^{\Delta t_C} \mathcal{C}_y^{\Delta t_C} \mathcal{C}_z^{\Delta t_C} \mathcal{C}_z^{\Delta t_C} \mathcal{C}_y^{\Delta t_C} \mathcal{C}_x^{\Delta t_C} U_0, \tag{10.16}$$

instead of (10.15) to better ensure the same numerical diffusion in all three directions and to preserve thus the isotropy of the computations (see also [dC09]). Furthermore, at each time step Δt_C we need to perform three steps in (10.16) to advance the solution instead of five in (10.15), with a possibly better efficiency of the scheme.
For the splitting scheme (10.10), the operator $\mathcal{C}$ is then given by

$$\mathcal{C}^{\Delta t} = \prod_{i=1}^{I_C} \mathcal{C}^{2\Delta t_{C,i}}, \tag{10.17}$$

for $2I_C$ convection steps $\Delta t_{C,i}$ within the global splitting step Δt, and (10.16):

$$\mathcal{C}^{2\Delta t_{C,i}} U_0 = \mathcal{C}_x^{\Delta t_{C,i}} \mathcal{C}_y^{\Delta t_{C,i}} \mathcal{C}_z^{\Delta t_{C,i}} \mathcal{C}_z^{\Delta t_{C,i}} \mathcal{C}_y^{\Delta t_{C,i}} \mathcal{C}_x^{\Delta t_{C,i}} U_0. \tag{10.18}$$

The intermediate time step $\Delta t_{C,i}$ is the same for all points over the computational domain, and it is computed such that

$$\Delta t_{C,i} = \min \left(\Delta t_{C,i}^{\max}, \frac{\Delta t}{2} - \sum_{i'=1}^{i-1} \Delta t_{C,i'} \right), \tag{10.19}$$

where $\Delta t_{C,i}^{\max}$ is the current maximum convection time step within the stability domain of the numerical scheme. The previous procedure is general and remains valid for any convective scheme and for both linear and nonlinear transport problems, with time- and space-varying transport velocities.
Some of the main advantages of a dimensional splitting technique are its easy implementation and the straightforward extensions to multi-dimensional configurations. Furthermore, stability restrictions in 1D are usually less restrictive than for multi-dimensional configurations, *i.e.*, $\max_{i\in[1,d]} \kappa_i \leq C$ instead of $\sum_{i=1}^{d} \kappa_i \leq C$. Usually, there are also fewer flux evaluations to compute, whereas the memory requirements are settled by a 1D configuration, considerably lower than those for fully multi-dimensional schemes. Nevertheless, the main disadvantages related to dimensional splitting are that it is not suitable for unstructured mesh, and that it is usually limited to low order schemes. Another inconvenience for some particular applications will be discussed in the following.

Dimensional Splitting and Divergence Free Velocity Fields

For most of the configurations and models that we will consider in this work (and the ones we would like to investigate in the future), the convective part is in a general conservative form like

in (10.12). The velocity field is either a variable to solve such as in the compressible Navier-Stokes or Euler equations, or a function of other variables such as in drift-diffusion equations modeling plasma discharges. In this context, the dimensional splitting will be a performing tool for structured meshes if coupled with a dedicated numerical method for conservation laws that ensures an adequate treatment of steep gradients, even shocks, such as the OSMP scheme [DT04]. In the particular context of the present chapter, we will consider a decoupled velocity field which will be imposed and divergence free, according to the thermo-diffusive approach. The scalar fields given by the temperature and the species mass fractions will be transported by such a velocity field along the fluid particles in (10.2). Hence, the relevant model is naturally in a rather non-conservative form with a velocity field that preserves the measure as well as constant temperature/species fields.
Nevertheless, since we want to have a generic approach for the convective subproblem, we will recast the convective system into a conservative form. A common problem that arise when using dimensional splitting is that such an approach may introduce some unwanted spatial deformations in the numerical solution of convection problems in conservative form like (10.12), with divergence free velocity fields. These issues are discussed with more details in [HV03]; they were originally noticed in [Bot92] and further investigated in [LLM96]. A simple example is the case of spatially constant solutions transported by a divergence free velocity field. In order to better illustrate this, let us consider one of the convective problems coming from (10.2):

$$\partial_t T + \mathrm{v} \cdot \partial_{\mathrm{x}} T = 0. \tag{10.20}$$

We switch to a conservative form by making use of the zero divergence of the velocity field[5]

$$\partial_t T + \partial_{\mathrm{x}} \cdot (\mathrm{v} T) = 0, \tag{10.21}$$

taking into account that both representation are equivalent as long as the velocity field v is divergence free:

$$\partial_{\mathrm{x}} \cdot \mathrm{v} = 0. \tag{10.22}$$

This is the case for the 2D velocity field given by (10.7), which have in particular motivated this short and complementary study.
If we now consider a spatially constant $T(\mathrm{x}) = T^\star$, we can see that the numerical time integration of (10.21) should naturally yield $T(\mathrm{x}, t) = T^\star$ after any time t. Nevertheless, this will not be necessarily the case for the dimensional splitting solution that independently considers problems:

$$\left.\begin{aligned} \partial_t T + \partial_x (v_x T) = 0, \\ \partial_t T + \partial_y (v_y T) = 0, \\ \partial_t T + \partial_z (v_z T) = 0, \end{aligned}\right\} \tag{10.23}$$

unless

$$\partial_x v_x = \partial_y v_y = \partial_z v_z = 0. \tag{10.24}$$

The resulting variations are thus artificial and may look as spatial deformations, but the cause is purely related to the coupling of dimensional splitting with a conservative formulation[6]. In particular, this deficiency may result in qualitatively bad results for problems with space-varying flow fields and spatially constant solutions in part of the domain, which are usually

[5] Notice also that the OSMP scheme was conceived for hyperbolic problems in conservative form.
[6] Let us underline that the use of dimensional splitting in advection form (10.20) would lead to no spatial oscillation but the resulting scheme would be non-conservative.

background concentration values in applications [LLM96, HV03]. Although second order dimensional splitting considerably reduces this problem, small oscillations may remain/appear depending on the spatial distribution of the problem.
This kind of inconvenience might be observed in the numerical simulation of problem (10.2), taking into account that we aim at solving localized propagating fronts in semi-equilibrium media. We have thus implemented the following simple procedure[7]. Instead of considering the convection subproblem (10.21), we apply the second order dimensional splitting technique to the equivalent problem:

$$\partial_t T + \partial_{\mathrm{x}} \cdot (\mathrm{v}T) = T(\partial_{\mathrm{x}} \cdot \mathrm{v}). \tag{10.25}$$

This amounts to consider problems:

$$\left.\begin{aligned} \partial_t T + \partial_x(v_x T) &= T(\partial_x v_x), \\ \partial_t T + \partial_y(v_y T) &= T(\partial_y v_y), \\ \partial_t T + \partial_z(v_z T) &= T(\partial_z v_z), \end{aligned}\right\} \tag{10.26}$$

for which the convective numerical scheme is implemented for the left side term as in the standard case (10.13), whereas the right side term is introduced as a corrective source term.

10.3.2 The OSMP Scheme

The OSMP scheme is built upon a one-step approach based on a Lax-Wendroff approximation [LW60]. Considering a linear scalar transport equation:

$$\partial_t u + a\,\partial_x u = 0, \tag{10.27}$$

with a constant velocity a, and denoting by u_j^{n+1} the numerical solution of (10.27) at time $t = t_0 + n\delta t$ and position $x = x_0 + j\delta x$, the main idea is to express u_j^{n+1} by using a Taylor series expansion where the time derivatives are substituted with space derivatives according to the exact equation (10.27). The latter procedure applied up to order 2 with an upwind discretization for odd derivatives and centered formulae for even derivatives, yields the second-order Lax-Wendroff scheme:

$$u_j^{n+1} = u_j^n - \frac{\delta t}{\delta x}(\mathrm{F}^{\mathrm{lw}}_{j+1/2} - \mathrm{F}^{\mathrm{lw}}_{j-1/2}), \tag{10.28}$$

where $\mathrm{F}^{\mathrm{lw}}_{j+1/2}$ is the Lax-Wendroff numerical flux:

$$\mathrm{F}^{\mathrm{lw}}_{j+1/2} = a u_j^n + \frac{(1-\kappa)}{2} a(u_{j+1}^n - u_j^n), \tag{10.29}$$

and κ is the local CFL number:

$$\kappa = a\frac{\delta t}{\delta x}. \tag{10.30}$$

The modified equation for this scheme reads

$$u_t + a\,\partial_x u = a\frac{\delta x^2}{6}(\kappa^2 - 1)\partial_x^3 u. \tag{10.31}$$

[7] Other corrective techniques are discussed in [LLM96] and [HV03].

By subtracting from the Lax-Wendroff scheme an upwind term issued from the discretization of the right hand side of (10.31), one obtains the classical third order upwind-biased scheme with a numerical flux given by

$$\mathrm{F}^3_{j+1/2} = a u^n_j + \frac{(1-\kappa)}{2} a \left(u^n_{j+1} - u^n_j - \frac{1+\kappa}{3}(u^n_{j+1} - 2u^n_j + u^n_{j-1}) \right). \tag{10.32}$$

Proceeding in the same way, the correction of the higher order error terms of the successive modified equations yields a recurrence relation that allows us to construct a family of schemes with arbitrary order of accuracy in time and space [DT04]. In this case, the numerical fluxes can be recast in a generic form:

$$\mathrm{F}^p_{j+1/2} = a u^n_j + \phi^p_{j+1/2} \frac{(1-\kappa)}{2} a (u^n_{j+1} - u^n_j), \tag{10.33}$$

such that the time integration is written as

$$u^{n+1}_j = u^n_j - \frac{\delta t}{\delta x} (\mathrm{F}^p_{j+1/2} - \mathrm{F}^p_{j-1/2}). \tag{10.34}$$

Hence, a scheme of order p is expressed in the usual form of a second order flux limiter scheme, and the $\phi^p_{j+1/2}$ plays the role of an accuracy function that settles the order of accuracy of the scheme. For the third order, we have, for instance,

$$\phi^3_{j+1/2} = 1 - \frac{1+\kappa}{3}(1 - r_{j+1/2}), \tag{10.35}$$

with

$$r_{j+1/2} = \frac{u^n_j - u^n_{j-1}}{u^n_{j+1} - u^n_j}. \tag{10.36}$$

The ϕ^p functions up the the 7-th order of accuracy were deduced in [DT04], as well as extensions to the nonlinear case:

$$\partial_t u + \partial_x g(u) = 0, \tag{10.37}$$

by considering the Jacobian $a(u) = \partial_u g$, and thus (10.34) holds with

$$\mathrm{F}^p_{j+1/2} = g^n_j + \phi^p_{j+1/2} \frac{(1-\kappa)_j}{2} (g^n_{j+1} - g^n_j). \tag{10.38}$$

For the third order, (10.35) becomes

$$\phi^3_{j+1/2} = 1 - \frac{1}{3} \frac{(1-\kappa^2)_{j+1/2} - (1-\kappa^2)_{j-1/2}\, r_{j+1/2}}{(1-\kappa)_{j+1/2}}, \tag{10.39}$$

with

$$r_{j+1/2} = \frac{g^n_j - g^n_{j-1}}{g^n_{j+1} - g^n_j}. \tag{10.40}$$

These schemes have the same order of accuracy in time and space, and have the property of giving the exact solution if the CFL number is equal to 1. In practice, a standard CFL stability condition is considered: $0 < \kappa \leq 1$. This family of schemes is rather simple to implement and considers fixed stencils for the flux evaluations. In terms of performance, it was shown in [DT09] that the 7th-order scheme is at least six times more efficient in CPU time with respect to a method of lines using a Runge-Kutta time integration or WENO schemes for space discretizations that represent the same accuracy. For the OSMP scheme, a stencil of eight points is required for 7-th order in time and space. In general, the stencils are constituted of $p+1$ points. Finally, the original monotonicity preserving constraints of Suresh & Huynh [SH97] have been recast in a TVD framework in order to preserve accuracy near extrema. In practice, these criteria are expressed in terms of flux limitation constraints applied through ϕ^p into (10.33) and (10.38), as developed in [DT04].

Implementation of the OSMP Scheme

We describe in this part the practical implementation of the OSMP scheme (10.38) for subproblem (10.25), following the alternative approach introduced to reduce the potential oscillations related to the dimensional splitting procedure for these particular problems.
Considering the split convection subproblem (10.21) and the standard OSMP scheme (10.38), the numerical solution in the x-direction after a time step Δt_C is given by

$$T_j^{n+1} = T_j^n - \frac{\Delta t_C}{\Delta x}(\mathrm{F}^p_{j+1/2} - \mathrm{F}^p_{j-1/2}), \tag{10.41}$$

with the fluxes

$$\mathrm{F}^p_{j+1/2} = \mathrm{F}^{\mathrm{roe}}_{j+1/2} + \phi^p_{j+1/2}\frac{(1-\kappa)_j}{2}|v_{x,j+1/2}|(T^n_{j+1} - T^n_j), \tag{10.42}$$

where $\mathrm{F}^{\mathrm{roe}}_{j+1/2}$ is the standard Roe numerical flux:

$$\mathrm{F}^{\mathrm{roe}}_{j+1/2} = \frac{1}{2}\left[v_{x,j+1/2}(T^n_{j+1} + T^n_j) - |v_{x,j+1/2}|(T^n_{j+1} - T^n_j)\right]. \tag{10.43}$$

All the computations are entirely evaluated on a locally uniform mesh of size Δx. The same follows for directions y and z.
According to (10.25), a corrective term should be introduced into the time evolution scheme (10.41) for this particular configuration. The scheme is thus written as

$$T_j^{n+1} = T_j^n - \frac{\Delta t_C}{\Delta x}(\mathrm{F}^p_{j+1/2} - \mathrm{F}^p_{j-1/2}) + \frac{\Delta t_C}{\Delta x}T^n_j(v_{x,j+1/2} - v_{x,j-1/2}), \tag{10.44}$$

for a one-step numerical scheme, that along with (10.42) and (10.43) yields a modified OSMP scheme. The numerical simulations of problem (10.2) show that the modified scheme (10.44) allows us to considerably reduce local accumulations in spatially constant regions. The oscillations introduced by the dimensional Strang splitting are however very small.

10.4 Propagation of Premixed Flames

In the framework of problem (10.2) that models laminar flames interacting with vortices, we conduct in the following a study on the performance of the proposed numerical strategy for the simulation of premixed flames in 2D and 3D configurations. The model under consideration comes from a configuration investigated by Laverdant & Candel in [LC89].

10.4.1 Model Formulation

We establish in this part the mathematical formulation of the model introduced in [LC89] for a 2D configuration. Extensions to 3D cases are straightforward and will be detailed afterward. We consider a square computational domain where a mixture of fuel and oxidizer constitutes the lower half-plane, while products occupy the upper half-plane. The chemistry is fast but with a finite rate, and a thin premixed laminar flame is initially located at the mid-plane. In this case, the reaction rate is modeled by the following Arrhenius law [LC89]:

$$\dot{w} = \frac{B_1}{W_{\mathrm{O}}W_{\mathrm{F}}}\rho^2 Y_{\mathrm{O}}Y_{\mathrm{F}}T^2\mathrm{e}^{-T_a/T}, \tag{10.45}$$

where B_1 is a preexponential factor and T_a, the activation energy. For premixed laminar flames, the mixture may be assumed to be fuel lean with a high diluent concentration. Hence,

the reaction rate is controlled by the fuel concentration, whereas the oxidizer mass fraction is nearly constant and equal to its upstream value $Y_{\mathrm{O}} = Y_{\mathrm{O}o}$. Therefore, (10.45) becomes

$$\dot{w} = \frac{B_1}{W_{\mathrm{O}} W_{\mathrm{F}}} \rho^2 Y_{\mathrm{O}o} Y_{\mathrm{F}} T^2 \mathrm{e}^{-T_a/T}. \tag{10.46}$$

Subscripts $(\,)_o$, $(\,)_b$, and $(\,)_\star$ indicate, respectively, fresh mixture zone, burnt product zone, and dimensionless variables.
The variation of the ratios $\rho_o/\rho_b = T_b/T_o$ is assumed negligible because the pressure is essentially constant and hypothesis 6. With these hypotheses, the composite Schvab-Zeldo'vich variable:

$$\theta_Z = T + \frac{Q}{c_p} Y_{\mathrm{F}}, \tag{10.47}$$

verifies a time dependent equation of type (10.2) without source term like (10.5), whereas from a simple energy balance relation in an adiabatic framework, we get

$$c_p(T_b - T_o) = Q(Y_{\mathrm{F}o} - Y_{\mathrm{F}b}). \tag{10.48}$$

By evaluating θ_{Zo} and θ_{Zb} in (10.47) and from (10.48), it can be seen that θ_Z is constant throughout the flame. Consequently, a progress variable $c(x, y, t)$ can be introduced:

$$c = \frac{T - T_o}{T_b - T_o} = \frac{Y_{\mathrm{F}o} - Y_{\mathrm{F}}}{Y_{\mathrm{F}o} - Y_{\mathrm{F}b}}. \tag{10.49}$$

Using $T/T_b = 1 + \tau c$, where $\tau = T_b/T_o - 1$, the reaction rate (10.46) becomes[8]

$$\dot{w} = B_\star Y_{\mathrm{F}o} (1 - c) \mathrm{e}^{-T_a/(T_o(1+\tau c))}, \tag{10.50}$$

with

$$B_\star = \frac{B_1}{W_{\mathrm{O}} W_{\mathrm{F}}} \rho_o^2 Y_{\mathrm{O}o} T_o^2. \tag{10.51}$$

Therefore, for the fuel mass fraction equation in (10.2), we now have

$$\partial_t Y_{\mathrm{F}} + v_x \partial_x Y_{\mathrm{F}} + v_y \partial_y Y_{\mathrm{F}} - D\left(\partial_x^2 Y_{\mathrm{F}} + \partial_y^2 Y_{\mathrm{F}}\right) = -\frac{B_\star}{\rho_o} Y_{\mathrm{F}o} (1 - c) \mathrm{e}^{-T_a/(T_o(1+\tau c))}, \tag{10.52}$$

which may be written as

$$\partial_t c + v_x \partial_x c + v_y \partial_y c - D\left(\partial_x^2 c + \partial_y^2 c\right) = \frac{B_\star}{\rho_o} (1 - c) \mathrm{e}^{-T_a/(T_o(1+\tau c))}. \tag{10.53}$$

Considering a square computational domain of size $2L$, a characteristic diffusion time $\tau_d = L^2/D$, and a velocity $V = D/L$, we define the following dimensionless variables:

$$x_\star = \frac{x}{L}, \qquad y_\star = \frac{y}{L}, \qquad v_{x,\star} = \frac{v_x}{V}, \qquad v_{y,\star} = \frac{v_y}{V}, \qquad t_\star = \frac{t}{\tau_d}. \tag{10.54}$$

We finally obtain [LC89]:

$$\partial_{t_\star} c + v_{x,\star} \partial_{x_\star} c + v_{y,\star} \partial_{y_\star} c - \left(\partial_{x_\star}^2 c + \partial_{y_\star}^2 c\right) = \mathrm{Da}\,(1 - c) \mathrm{e}^{-T_a/(T_o(1+\tau c))}, \tag{10.55}$$

where $\mathrm{Da} = B_\star \tau_d/\rho_o = \tau_d/\tau_{ch}$ is a Damköhler number and $\tau_{ch} = B_\star/\rho_o$, a chemical time.

[8] Taking into account that $Y_{\mathrm{F}b} = 0$ into (10.49).

The dimensionless tangential velocity induced by the viscous core vortex coming from (10.7), becomes

$$v_{\theta,\star}(r_\star, t_\star) = \frac{\mathrm{Re}\,\mathrm{Sc}}{r_\star}\left(1 - \mathrm{e}^{-r_\star^2/(4\,\mathrm{Sc}\,t_\star)}\right), \tag{10.56}$$

where the Reynolds and Schmidt numbers are defined by

$$\mathrm{Re} = \frac{\Gamma}{2\pi\nu}, \quad \mathrm{Sc} = \frac{\nu}{D}. \tag{10.57}$$

In Cartesian coordinates, the velocity is given by

$$\left.\begin{aligned} v_{x,\star} &= \left(\frac{y_\star - y_{0,\star}}{r_\star}\right) v_{\theta,\star}, \\ v_{y,\star} &= -\left(\frac{x_\star - x_{0,\star}}{r_\star}\right) v_{\theta,\star}, \end{aligned}\right\} \tag{10.58}$$

for a counterclockwise rotating vortex with radius

$$r_\star^2 = (x_\star - x_{0,\star})^2 + (y_\star - y_{0,\star})^2, \tag{10.59}$$

where $(x_{0,\star}, y_{0,\star})$ is the center of the vortex.

10.4.2 Numerical Simulations: 2D Configuration

In this application we consider two dynamic counter rotating vortices, each one modeled by (10.56), interacting with a premixed flame governed by (10.55) in a 2D computational domain. All these simulations have been performed on an AMD Shanghai 2.7 GHz processor with memory capacity of 32 GB.

Data Initialization and Simulation Parameters

We solve problem (10.55) with Neumann homogeneous boundary conditions in a 2D computational dimensionless domain of $[-1,1]^2$, *i.e.*, $L = 1$ into (10.54). The initial condition corresponds to a premixed flame in the limit of large activation energy [LC89]:

$$c = \begin{cases} \mathrm{e}^{(y_\star - y_{0,\star})/\Delta_\star}, & y_\star \le y_{0,\star}, \\ 1, & y_\star > y_{0,\star}, \end{cases} \tag{10.60}$$

where $\Delta_\star$ is the preheat zone thickness. For the following computations, the following modeling values were considered into equations (10.55), (10.56), and (10.60):

$$\mathrm{Da} = 2.5 \times 10^9, \quad T_a = 20000\,\mathrm{K}, \quad T_o = 300\,\mathrm{K} \quad \tau = 6.72, \tag{10.61}$$

$$\mathrm{Re} = 1000, \quad \mathrm{Sc} = 1, \tag{10.62}$$

$$\Delta_\star = 0.02. \tag{10.63}$$

The velocity field is given by the superposition of two vortices with opposite signs in (10.58), centered at $(-0.25, -0.5)$ for the counterclockwise vortex, and $(0.25, -0.5)$ for the clockwise one. The velocities are computed before each time integration of the convection problem and updated after two time steps $\Delta t_{C,i}$ according to (10.17), as well as the maximum time step $\Delta t_{C,i}^{\max}$.

For the numerical strategy, the adaptive splitting accuracy tolerance in (10.11) was set to $\eta = 10^{-3}$, unless noted otherwise, with $\eta_{ROCK4} = 10^{-5}$ and $\eta_{Radau5} = 10^{-7}$ for the ROCK4 and Radau5 solvers. The third order OSMP scheme was considered for the convection problem with a stability CFL condition equal to 1. The time domain of integration was given by $t_\star \in [0, 4 \times 10^{-3}]$. For the multiresolution analysis and in order to be consistent with the tolerance of the time integration solver, $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-3}$ were chosen as multiresolution threshold values in the following illustrations. The finest grid corresponds to a spatial discretization of 1024^2 points, *i.e.*, $J = 10$ as finest grid level.

Numerical Results

According to the definition of the progress variable c in (10.49), the fresh mixture is given by $c = 0$, whereas $c = 1$ corresponds to the burnt gases. Starting from the planar premixed flame (10.60) at $y_\star = -0.5$, with fresh gases in the lower part (blue zone in the figure), Figure 10.2 shows the time evolution of c and the interaction of the two imposed vortices with the flame front. The velocity field generated by the vortices is shown in Figure 10.1, and is characterized by high values with localized strong gradients for the considered Reynolds number of Re = 1000. As a consequence, the fresh mixture is drawn up towards the hot region (red zone in Figure 10.2) in the center region, beating the advance of the reacting front, whereas hot gases propagate faster in the outer zones around the vortices cores.

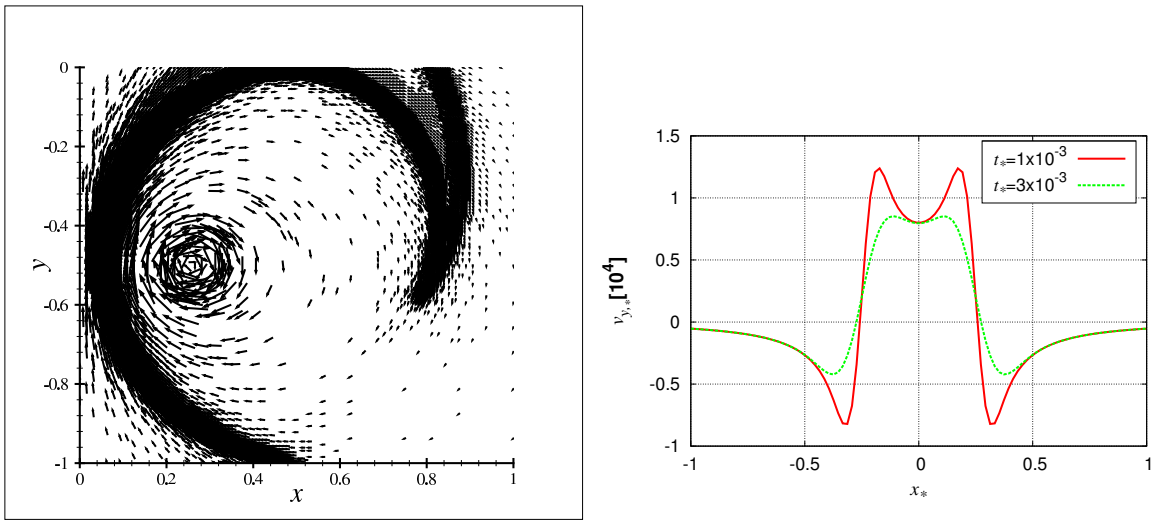

Fig. 10.1: *2D propagating flame. Velocity field at* $t_\star = 10^{-3}$ *(left) and* $v_{y,\star}$ *at* $y_\star = -0.5$*, at* $t_\star = 10^{-3}$ *and* $t_\star = 3 \times 10^{-3}$ *(right).*

The contour lines in Figure 10.2 account for the spatial thickness of the flame in which the fresh gases react and burn, and where an important numerical effort is usually required to precisely describe the phenomenon. In this configuration the flame thickness is reduced from about 0.05 in the standard planar configuration, to approximatively 0.025 at regions where the flame surface is sheared by the locally high velocity gradients as seen in Figure 10.1. A spatial mesh of 1024^2 points involves approximatively 10 discretization points throughout the flame front, and generates a reasonably good numerical representation of the problem. From a numerical point of view, the latter issue imposes a fine spatial discretization for a localized structure, 100 times smaller than the global scale of the computational domain, and thus an adaptive mesh refinement technique seems natural.

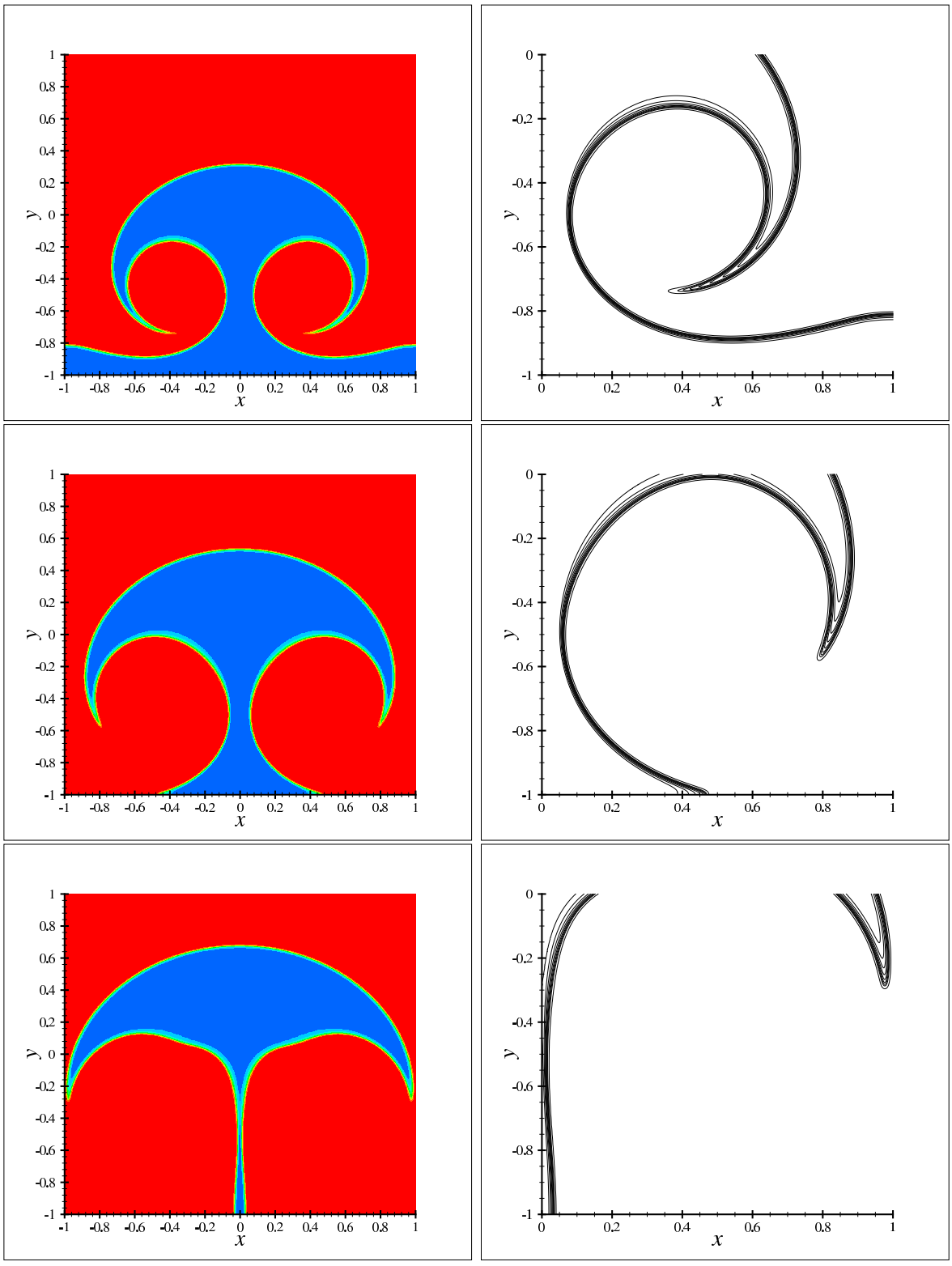

Fig. 10.2: *2D propagating flame. Time evolution of variable c at $t_\star = 5 \times 10^{-4}$ (top), 10^{-3} (middle), and 1.5×10^{-3} (bottom). Left: red (resp., blue) zone corresponds to burnt (resp., fresh) gases, $c = 1$ (resp., $c = 0$). Right: contour lines with $c = 0 - 0.99$ and $\Delta c = 0.11$.*

Figure 10.3 shows the corresponding adapted grids. The representation involves 7 levels of different spatial discretization where the finest regions coincide with the propagating fronts. Notice that the highest velocity values do not necessarily correspond to the flame front as seen in Figure 10.1, and thus we will have constant concentration regions with strong variations of the transport field. In the context of dimensional splitting, the latter feature introduces some numerical errors as previously discussed, and the corrective term introduced in (10.44) allows us to reduce local accumulations of c from a maximum of 1.01 to approximatively 1.0001 in the burnt regions. The multiresolution decomposition naturally identifies this behavior, and consequently refines these zones as seen in Figures 10.1 and 10.3. In uniform grid simulations, no accumulation was detected when using (10.44), and c is strictly lower than or equal to 1. Finally, the data compressions DC defined by (4.39) and achieved in this case with $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-3}$, show that no more than, respectively, 10 % or 15 % of the 1024^2 points are necessary to represent the flame front within the prescribed tolerance.

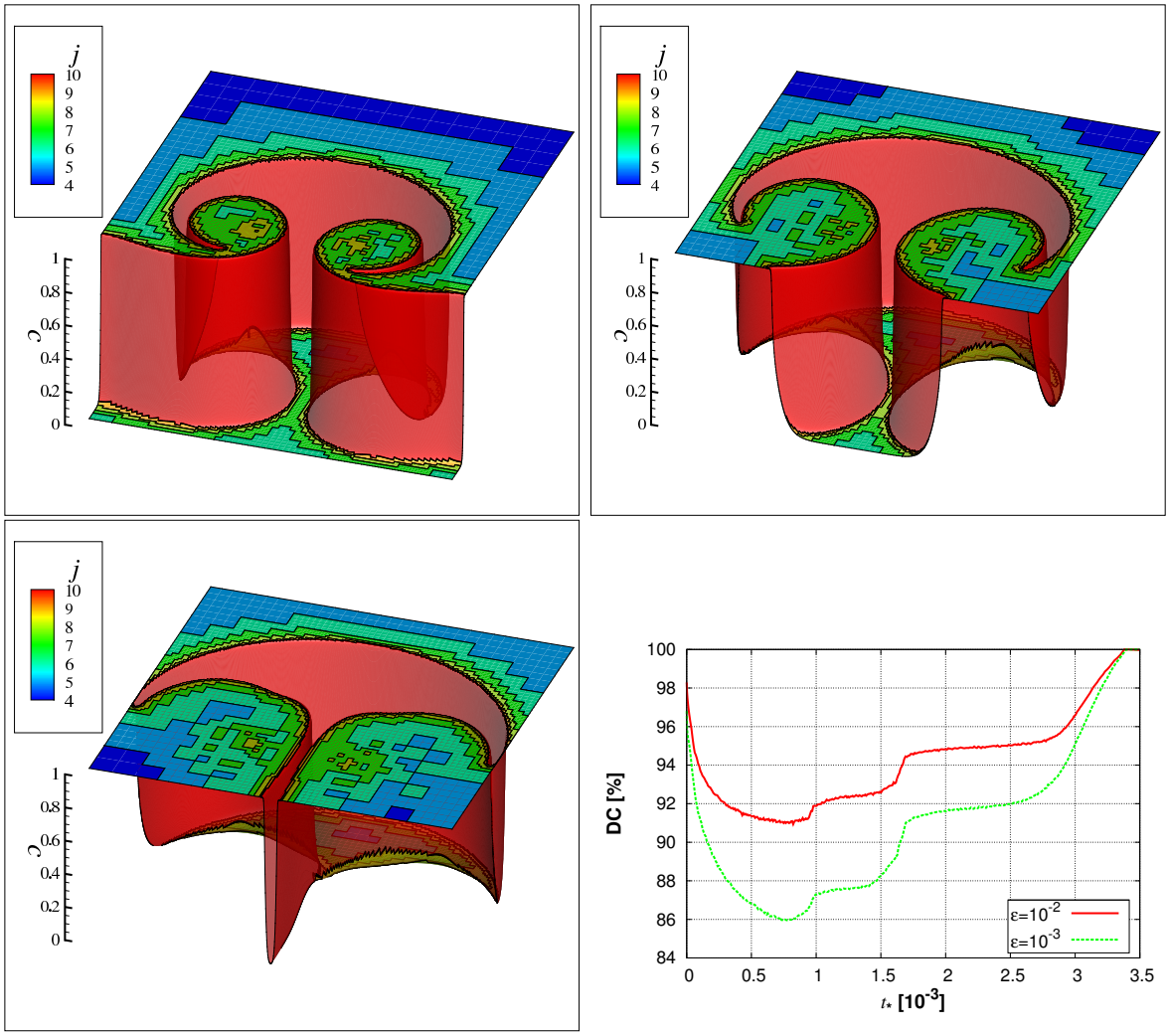

Fig. 10.3: *2D propagating flame. Variable* c *at* $t_\star = 5\times10^{-4}$ *(top left),* 10^{-3} *(top right), and* 1.5×10^{-3} *(bottom left) represented on a dynamic adapted grid corresponding to* 1024^2 *points at the finest level* $J = 10$ *with* $\varepsilon = 10^{-3}$*. Bottom right: time evolution of data compressions* DC *given by (4.39) for* $\varepsilon = 10^{-2}$ *and* $\varepsilon = 10^{-3}$*.*

In order to verify that the accuracy of the computations is settled by the accuracy tolerances,

we define for problem (10.55) discretized on a uniform mesh of 1024^2:

- A *quasi-exact* reference solution $\mathbf{c}^J_{qe}$, obtained with the Strang scheme (10.10) with a small fixed splitting time step of $\Delta t = 10^{-7}$;
- The *splitting* solution $\mathbf{c}^J_{split}$ is now obtained with the adaptive splitting scheme (10.10), with (10.11) and accuracy tolerance of $\eta = 10^{-3}$, computed also on the uniform grid; and
- The *time/space adaptive* solution $\mathbf{c}^{MR}_{split}$, composed of the adaptive splitting technique and the multiresolution representation with the equivalent of 1024^2 points in the finest grid level $J = 10$.

Defining also the numerical errors:

$$\mathrm{E}^J_{split} = \left\|\mathbf{c}^J_{qe} - \mathbf{c}^J_{split}\right\|_{L^2}, \tag{10.64}$$

$$\mathrm{E}^J_{MR} = \left\|\mathbf{c}^J_{split} - \mathbf{c}^{MR}_{split}\right\|_{L^2}, \tag{10.65}$$

$$\mathrm{E}^{MR}_{split} = \left\|\mathbf{c}^J_{qe} - \mathbf{c}^{MR}_{split}\right\|_{L^2}, \tag{10.66}$$

corresponding, respectively, to the time adaptive splitting, space adaptive multiresolution, and time/space adaptive approximations, the following Table 10.1 summarizes these errors where the solutions on adapted grids $\mathbf{c}^{MR}_{split}$, were reconstructed on the finest grid.

Table 10.1: *2D propagating flame. L^2 numerical errors for the time adaptive splitting (E^J_{split}), space adaptive multiresolution (E^J_{MR}), and time/space adaptive (E^{MR}_{split}) strategies evaluated at different times. Finest grid:* 1024^2.

$t_\star\,[10^{-3}]$	$\mathrm{E}^J_{split}\,[10^{-2}]$	$\mathrm{E}^J_{MR}\,[10^{-3}]$		$\mathrm{E}^{MR}_{split}\,[10^{-2}]$	
		$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$
0.5	1.45	5.28	1.79	1.14	1.16
1.0	2.71	6.36	1.69	2.17	2.28
1.5	4.74	7.37	3.12	4.14	4.34
2.0	5.74	7.05	3.47	5.21	5.44

With this choice of parameters, the global accuracy of the numerical strategy E^{MR}_{split} is indeed ruled by the approximation error of the time integration E^J_{split}, which is related to the local error tolerance η. The latter global error will remain practically independent of the multiresolution errors E^J_{MR} as seen in Table 10.1, as long as appropriate values of data thresholding are taken into account (see discussions in chapters 4 and 5). Concerning the multiresolution errors, the proportionality with respect to ε is verified but with a lower rate than in previous results. This rate reduction arises either because some error compensation takes place in the case with $\varepsilon = 10^{-2}$, or because of the spurious values introduced by the dimensional splitting which are not present in the reference solutions on uniform grid, used to compute these errors. For this particular problem, the global accuracy is nevertheless preserved for both thresholding values, even for $\varepsilon = 10 \cdot \eta$ (see final remarks in Chapter 5).

Table 10.2: *2D propagating flame. CPU time in minutes for the quasi-exact, the time adaptive splitting, and the time/space adaptive strategies for* $t_\star \in [0, 2 \times 10^{-3}]$. *Finest grid:* 1024^2.

	MR/splitting $\varepsilon =$		splitting	quasi-exact
	10^{-2}	10^{-3}		
CPU time (m)	56.27	71.05	589.00	6603.26

Table 10.2 includes the CPU times for half the time domain of study $t_\star \in [0, 2 \times 10^{-3}]$, taking into account that the reference quasi-exact solution is very expensive to compute. The total CPU time for $t_\star \in [0, 4 \times 10^{-3}]$ was about 80.73 and 98.38 minutes for $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-3}$, respectively. Notice that the splitting CPU time accounts for the cost reduction with respect to the quasi-exact solution that considers a small time step of the order of the fastest numerical scale (the convective CFL condition in this case). Nevertheless, much more performing strategies can be implemented to obtain the coupled reference solution, and the previous values should be taken as one possible numerical indicator. Additionally, the CPU times associated with the time/space adaptive technique account for the gain issued from the compressed data representation, if one compares them with the splitting CPU time. In this case, these gains are entirely coherent with the corresponding data compression achieved with each threshold value in Figure 10.3. This is a sign that the most expensive part of computations is related to the spatial representation, and thus either to the convection, the diffusion or both problems. This is confirmed by the CPU time per time step spent on the solution of each problem, given in Table 10.3. We note that in this implementation and for the diffusion problem, we do not use the previous matrix representation of the Laplacian operator, and thus better performances might be achieved considering its sparse structure at least for the uniform grid problem as previously discussed in this work.

Table 10.3: *2D propagating flame. CPU time per splitting time step (in seconds) for the reaction, diffusion, and convection time integrations for the time adaptive splitting and the time/space adaptive techniques,* $t_\star = 1.5 \times 10^{-3}$, *and* $\Delta t \approx 8.77 \times 10^{-6}$ *for the three solutions. Finest grid:* 1024^2.

	CPU time per time step (s)		
	splitting	MR/splitting	
		$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$
Reaction	4.87	1.09	1.37
Diffusion	19.93	5.09	4.48
Convection	68.65	6.79	10.26

Finally, Figure 10.4 illustrates the adaptive time steps considered in this problem. The splitting time steps are practically the same for both multiresolution tolerances. This indicates a good decoupling of spatial and temporal errors and also a proper coupling of time evolution and spatial representations. Otherwise, any deficiency will be reflected by the local error estimates, and thus by the splitting time steps issued from the time adaptive scheme, as discussed in Chapter 5. An initial value of $\Delta t^0 = 10^{-8}$ was chosen in order to cope with the sudden apparition of the velocity field with very high maximum values of about 4×10^5, with $\mathrm{Re} = 1000$

into (10.56). Taking into account that the global physics is piloted by the propagation of the flame, the splitting time steps evolve until a practically constant value of $\Delta t \approx 9 \times 10^{-6}$. Nevertheless, time adaptation is needed to handle fast variations in the beginning, and for the final total combustion of the fresh gases at some unknown time. In this configuration, all gases are burnt by $t_\star = 3.5 \times 10^{-3}$. Figure 10.4 shows that an important decoupling of time scales is possible, where the splitting time step Δt is globally at least 10 times larger than the inner integration steps for the split reaction, diffusion, and convection problems. This naturally yields important gains of computational efficiency, always within a prescribed accuracy.

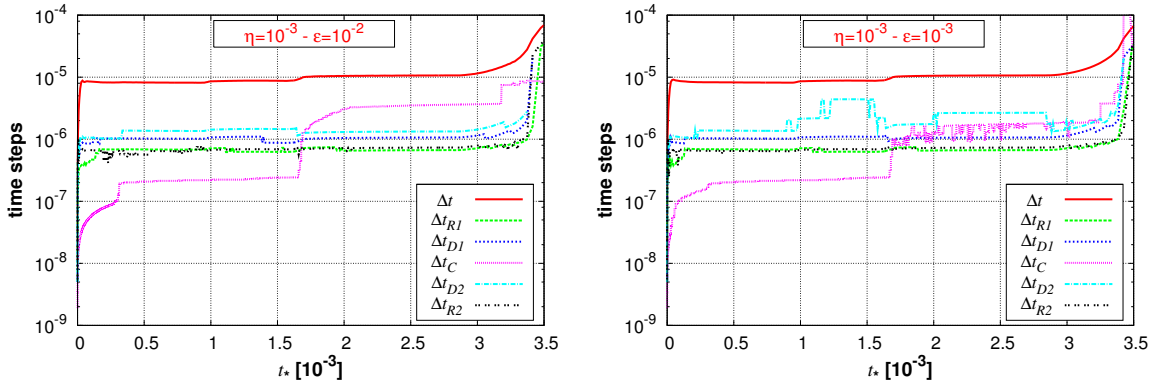

Fig. 10.4: *2D propagating flame. Time evolution of the splitting time step Δt, the reactive Δt_{R1} and Δt_{R2}, diffusive Δt_{D1} and Δt_{D2}, and convective Δt_C substeps according to the Strang scheme (10.10) with tolerances $\eta = 10^{-3}$ and $\varepsilon = 10^{-2}$ (left) or $\varepsilon = 10^{-3}$ (right).*

The reaction and diffusion time steps are chosen based on the accuracy tolerances of the solvers η_{Radau5} and η_{ROCK4}, and for each half splitting time step we represent in Figure 10.4 the averaged values of the inner reaction and diffusion substeps. This data post treatment serves only to obtain clearer representations of the time evolution of the time steps without including the often much smaller finishing substeps within the current (half) splitting time step. Reaction steps are of the order of $\Delta t_R \approx 7 \times 10^{-7}$ at the flame front (shown in Figure 10.4), and they progressively increase up to $\Delta t_R = \Delta t/2$ away from the highly reacting area. Diffusion time steps are of the order of $\Delta t_D \approx 5 \times 10^{-6}$ (averaged in the representation) with $s = 6$ inner stages for ROCK4, for a spectral radius of about 2.2×10^6. The convection time steps are computed by (10.19), based on the maximum stability time steps $\Delta t_{C,i}^{\max}$ which are illustrated in Figure 10.4, to avoid the previous representation problem. This convective step ranges from $\Delta t_C \approx 2 \times 10^{-9}$ in the beginning to $\Delta t_C \approx 2 \times 10^{-7}$ and $\Delta t_C \approx 5 \times 10^{-6}$, due mainly to the constraining high Reynolds number considered. This reasonably explains the high cost of the convective step per splitting time step in Table 10.3. The variations in the time evolution of the convection time step are originated by the fact that the highest velocity values are not necessarily at the finest grid during the whole phenomenon, as previously noted. It is important to notice that a local time stepping strategy is not easy to implement in such a configuration because of the previous remark, and also because the velocity field is highly nonlinear in space and evolves in time. As a consequence, a simple procedure to consistently partition the largest convection time step and to define local time steps over the whole domain, is not straightforward. Another alternative that has not been studied in this work, considers an adaptive mesh refinement based not only on the unknown variables of the problem, but also on the velocity field.

10.4.3 Numerical Simulations: 3D Configuration

The time/space adaptive technique is easily extended to 3D configurations. In order to illustrate this, we consider the solution of (10.55) over a dimensionless computational domain of $[-1,1]^3$. The same data initialization is considered with $z_\star$ and $z_{0,\star}$, instead of $y_\star$ and $y_{0,\star}$, respectively, into (10.60), as well as the same modeling parameters (10.61)-(10.63).

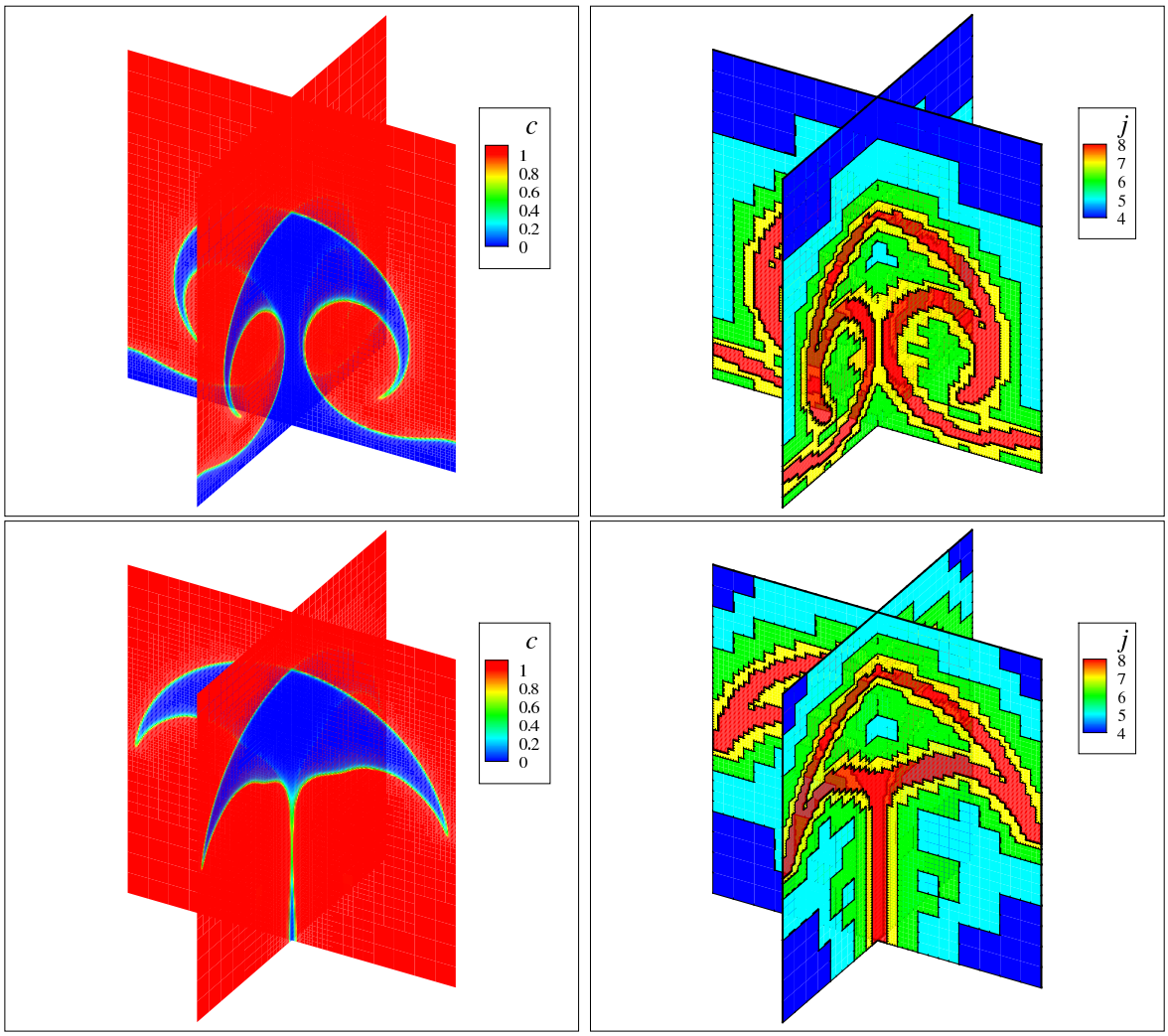

Fig. 10.5: *3D propagating flame. Time evolution of variable c (left) and dynamic adapted grids (right) corresponding to* 256^3 *points at the finest level* $J = 8$*, at* $t_\star = 5 \times 10^{-4}$ *(top) and* 1.5×10^{-3} *(bottom).*

The adaptive splitting accuracy tolerance in (10.11) is also set to $\eta = 10^{-3}$, with $\eta_{ROCK4} = 10^{-5}$, $\eta_{Radau5} = 10^{-7}$, and $\varepsilon = 10^{-2}$ for the multiresolution threshold value. The time domain of integration is given by $t_\star \in [0, 3.5 \times 10^{-3}]$, whereas the finest grid corresponds to a spatial discretization of 256^3 points, *i.e.*, $J = 8$ as finest grid level. A 3D velocity field is defined for this configuration by a toroidal vortex directly inspired by the previous 2D velocity field. At each plane containing the z-axis, we consider thus a pair of counter rotating vortices computed as usual with (10.56) with radius

$$r_\star^2 = (x_\star - x_{0,\star})^2 + (y_\star - y_{0,\star})^2 + (z_\star - z_{0,\star})^2, \tag{10.67}$$

centered at $\sqrt{x_{0,\star}^2 + y_{0,\star}^2} = 0.25$, $z_{0,\star} = -0.5$. Although the resulting field is not divergence

free, it suffices to construct a 3D configuration to illustrate the numerical capabilities of the method. The same simulations can be performed exactly in the same way with more physically consistent velocity fields. Figure 10.5 shows the interaction of the initial premixed flame with the toroidal vortex, and the corresponding adapted grids on which the solutions are computed. Figure 10.6 shows the corresponding time steps of integration. We retrieve qualitatively a similar behavior with respect to the previous 2D case, in terms of splitting time steps and the time stepping for each split subproblem. The fresh gases are completely burnt this time by $t_\star = 3 \times 10^{-3}$. Considering the obtained data compression, no more than 18 % of the 256^3 points are required. This simulation took approximatively 17.26 hours of CPU time.

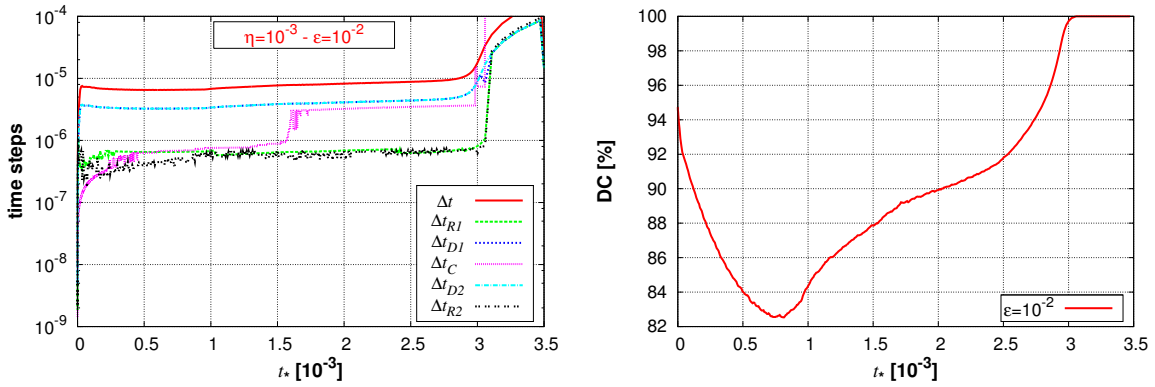

Fig. 10.6: *3D propagating flame. Left: time evolution of the splitting time step* Δt*, the reactive* Δt_{R1} *and* Δt_{R2}*, diffusive* Δt_{D1} *and* Δt_{D2}*, and convective* Δt_C *substeps with tolerances* $\eta = 10^{-3}$ *and* $\varepsilon = 10^{-2}$*. Right: time evolution of data compressions* DC *given by (4.39),* $\varepsilon = 10^{-2}$*.*

10.4.4 Introduction of Complex Chemistry

In this part, we consider the same previous 2D application but including detailed chemical kinetics in the reaction term. The main goal is to illustrate the previous numerical implementation in a more complex context by considering a simple extension of the previous model. Even though the following procedure to implement complex chemistry in the previous modeling framework is not completely rigorous from a physical point of view, it allows us to obtain some preliminary results of the proposed numerical strategy, applied to combustion problems with detailed chemistry. For more details on combustion modeling with detailed chemistry we refer to the books of Giovangigli [Gio99] and Poinsot & Veynante [PV05].

Detailed Chemical Kinetics

Instead of the single step reaction (10.1), we now consider a system of N_s species reacting according to N_R reactions:

$$\sum_{k=1}^{N_s} \nu'_{k,j} \chi_k \rightleftharpoons \sum_{k=1}^{N_s} \nu''_{k,j} \chi_k \qquad j = 1, \ldots, N_R, \tag{10.68}$$

where χ_k stands for species k, and $\nu'_{k,j}$ and $\nu''_{k,j}$ are the molar stoichiometric coefficients of species k in reaction j. Taking into account the mass conservation yields

$$\sum_{k=1}^{N_s} \nu'_{k,j} W_k = \sum_{k=1}^{N_s} \nu''_{k,j} W_k, \tag{10.69}$$

and thus

$$\sum_{k=1}^{N_s} \nu_{k,j} W_k = 0, \qquad j = 1, \ldots, N_R, \tag{10.70}$$

where

$$\nu_{k,j} = \nu''_{k,j} - \nu'_{k,j}. \tag{10.71}$$

In this context, the reaction rate $\dot{w}_k$ of species k considers the rate of production of χ_k by all N_R reactions (10.68), and it is given by

$$\dot{w}_k = \sum_{j=1}^{N_R} \nu_{k,j} \mathcal{Q}_k, \tag{10.72}$$

where $\mathcal{Q}_k$ is the rate of progress of reaction j, which is written as

$$\mathcal{Q}_k = K_{f,j} \prod_{k=1}^{N_s} \left(\frac{\rho Y_k}{W_k} \right)^{\nu'_{k,j}} - K_{r,j} \prod_{k=1}^{N_s} \left(\frac{\rho Y_k}{W_k} \right)^{\nu''_{k,j}}, \tag{10.73}$$

where $K_{f,j}$ and $K_{r,j}$ define, respectively, the forward and reverse rates of reaction j. The determination of these rate constants constitute a central problem in combustion modeling [PV05], and they are usually modeled by an empirical Arrhenius law (as seen before, for instance, for (10.45)):

$$K_{f,j} = A_{f,j} T^{\beta_j} \mathrm{e}^{-T_{a,j}/T}. \tag{10.74}$$

The backward rates $K_{r,j}$ are then computed based on the forward ones (10.74), and thus computing the progress rates $\mathcal{Q}_k$ by (10.73) for each reaction j, involves providing data for the preexponential constants $A_{f,j}$, the temperature exponents β_j, and the activation temperatures $T_{a,j}$.

Implementation of Complex Chemistry in the Model

Considering the same modeling hypotheses, the general problem (10.2) can be rewritten for $k = 1, \ldots, N_s$, as

$$\left.\begin{aligned} \partial_t Y_k + \mathrm{v} \cdot \partial_{\mathrm{x}} Y_k - D\, \partial^2_{\mathrm{x}} Y_k &= \frac{W_k}{\rho} \dot{w}_k, \\ \partial_t T + \mathrm{v} \cdot \partial_{\mathrm{x}} T - D\, \partial^2_{\mathrm{x}} T &= -\frac{1}{\rho c_p} \sum_{k=1}^{N_s} h_k W_k \dot{w}_k, \end{aligned}\right\} \tag{10.75}$$

according to (10.3) and (10.4), where the reaction rates $\dot{w}_k$ are given now by (10.72).
The main difficulty of this problem is that the thermo-diffusive assumption 7 of constant density is not consistent with the local variation of species concentrations, on which the reaction rates depend. Actually, in a low Mach regime for which the previous balance equations are still valid (see, *e.g.*, [PV05]), the local variation of density should verify the state equation:

$$\rho \frac{R}{W} T = p_0, \tag{10.76}$$

for a constant pressure p_0, where $R = 8.314\,\mathrm{J/(mol \cdot K)}$ is the perfect gas constant, and W is the mean molecular weight of the mixture expressed by

$$\frac{1}{W} = \sum_{k=1}^{N_s} \frac{Y_k}{W_k}. \tag{10.77}$$

In practice, this inconsistency naturally leads to strong numerical problems. In order to overcome this difficulty while keeping at the same time the decoupling between the velocity field and the determination of species and temperature (*i.e.,* an imposed velocity field into (10.75)), we have made the following assumption. The density is supposed constant on the left hand-side of the equations in (10.75) such that we keep exactly the previous approach, whereas it is updated and locally computed on the right hand-side by the state equation (10.76) with the current concentrations and temperature.
Furthermore, considering the reaction problem coming from (10.75):

$$\left.\begin{aligned} \rho\partial_t Y_k &= W_k\dot{w}_k, \\ \rho c_p\partial_t T &= -\sum_{k=1}^{N_s} h_k W_k\dot{w}_k, \end{aligned}\right\} \tag{10.78}$$

and that by definition

$$c_p = \sum_{k=1}^{N_s} c_{p,k}Y_k, \tag{10.79}$$

we can easily see that

$$\rho\partial_t h = \rho c_p\partial_t T + \sum_{k=1}^{N_s} h_k\rho\partial_t Y_k = 0. \tag{10.80}$$

Therefore, since the enthalpy is uniformly constant for problem (10.75)-(10.76), *i.e.,* $h = h_0$ throughout the flame front, and that

$$h_0 = \sum_{k=1}^{N_s} h_k Y_k = \Psi(Y_1, \ldots, Y_{N_s}, T), \tag{10.81}$$

where the species enthalpies locally depend on the temperature, we can compute the local temperatures from (10.81) with the current set of species concentrations, rather than from the time dependent equation for the temperature in (10.75). Although solving (10.81) implies the numerical solution of an implicit nonlinear system at each point, this alternative allows us to consistently solve the right hand side by locally considering the right temperature, according to the constant and known enthalpy of the mixture, and thus the corresponding density according to the state equation (10.76). Additionally, we can correctly couple the left and right side terms, for which the same enthalpy is conserved in time and space.
Finally, considering the same dimensionless variables (10.54), we aim at solving

$$\partial_{t_\star} Y_k + \mathrm{v}_\star \cdot \partial_{\mathrm{x}_\star} Y_k - D\,\partial^2_{\mathrm{x}_\star} Y_k = \tau_d \frac{W_k}{\rho}\dot{w}_k, \tag{10.82}$$

for $k = 1, \ldots, N_s$, where the velocity field $\mathrm{v}_\star$ is also based on (10.56). With the previous assumptions, the reaction problem

$$\partial_{t_\star} Y_k = \tau_d \frac{W_k}{\rho}\dot{w}_k, \tag{10.83}$$

is locally solved, by considering the uniformly constant enthalpy h_0, and that

1. The local temperature T is computed from (10.81) with the current set of species concentrations $(Y_1, \ldots, Y_{N_s})$;

2. The density ρ is computed from the state equation (10.76) and the current values of T and $(Y_1, \ldots, Y_{N_s})$; and

3. The reaction rates $\dot{w}_k$ are computed from (10.72) with the current values of ρ, T, and $(Y_1, \ldots, Y_{N_s})$.

Once again, this numerical procedure will not be necessary for a more consistent physical model for the transport equations and the corresponding hydrodynamics, but allows us to perform some numerical tests of the proposed strategy for models including detailed chemistry.

Data Initialization and Simulation Parameters

We consider the system (10.82) of N_s equations in a 2D square computational domain of 5 cm of side, *i.e.*, $[-6.25, 6.25]^2$ in dimensionless units, where we have considered typical diffusion length and time scale of $L = 4 \times 10^{-3}$ m and $\tau_d = L^2/D$ with $D = 2.26 \times 10^{-5}\,\mathrm{m}^2/\mathrm{s}$.
A methane CH_4 premixed flame is taken into account for which the global reaction is written as

$$CH_4 + 2(O_2 + 3.76N_2) \rightarrow CO_2 + 2H_2O + 7.52N_2 \tag{10.84}$$

with a *mass stoichiometric ratio* st of 4 according to

$$\mathrm{st} = \frac{\nu'_{\mathrm{O}} W_{\mathrm{O}}}{\nu'_{\mathrm{F}} W_{\mathrm{F}}}. \tag{10.85}$$

The detailed kinetics (10.68) associated with the global reaction (10.84) is modeled by $N_s = 49$ species and $N_R = 299$ reactions, following a reactive scheme developed by Lindstedt & Leung (1998). The reaction rates (10.72) and in general all species or mixture related variables are computed using the Chemkin library [KRM80].
The fresh mixture is given by

$$Y_{CH_4} = 0.06, \qquad Y_{O_2} = 0.3, \qquad Y_{N_2} = 0.64, \tag{10.86}$$

at temperature $T_o = 300\,K$, and atmospheric pressure p_0, which involves an adiabatic flame temperature of $T_b = 2349.67\,K$ and a mixture enthalpy of $h_0 \approx -2.8 \times 10^9$, according to (10.81), for the given set of species and reactions. The initial mixture (10.86) corresponds to a lean regime in which the oxidizer is in excess with respect to the fuel, and thus the *mixture equivalence ratio* ϕ_M is lower than 1:

$$\phi_M = \mathrm{st}\frac{Y_{\mathrm{F}}}{Y_{\mathrm{O}}}. \tag{10.87}$$

In order to initialize the premixed flame we solve the general problem (10.2) for Y_{F}, Y_{O}, and T in a 1D configuration without convection, in a simplified chemistry framework. When the flame front has converged and propagates in a self-similar way, the obtained profile is reconstructed for all species by an interpolation technique based on the one performed initially on Y_{F}, *i.e.*, Y_{CH_4}, considering the known, fresh and burnt, concentration values. A 2D plane flame with all N_s species is thus generated, and it is again integrated without convection with a constant splitting time step of $\Delta t = 10^{-5}$ during $t_\star \in [0, 10^{-4}]$, in order to obtain a sufficiently consistent flame profile for all species. At $t_\star = 10^{-4}$, the velocity field is turned on, as well as the time adaptive scheme for the complete reaction-diffusion-convection problem with an initial splitting time step of $\Delta t^0 = 10^{-8}$, and an accuracy tolerance of $\eta = 10^{-2}$. We keep the same previous accuracy tolerances for ROCK4 and Radau5, *i.e.*, $\eta_{ROCK4} = 10^{-5}$ and $\eta_{Radau5} = 10^{-7}$.

The space adaptive technique is used from the beginning of computations with a threshold value of $\varepsilon = 10^{-2}$, for which the finest grid corresponds to a spatial discretization of 1024^2 points, *i.e.,* $J = 10$ as finest grid level. Another standard initialization procedure considers, for instance, an initial 1D flame obtained with the Premix solver of Chemkin [KRM80] for the given kinetics, instead of the simple chemistry flame (see, *e.g.,* [DB00, NK05, SPN06a]).
As in the previous 2D case, the velocity field is given by the superposition of two vortices with opposite signs in (10.58), centered at $(-2, 2.5)$ for the counterclockwise vortex and $(2, 2.5)$ for the clockwise one, where $y_\star = 2.5$ is approximatively the initial position of the plane front. Moreover, $\mathrm{Re} = 625$ and $\mathrm{Sc} = 1$ are considered for the velocity evaluation. In order to avoid the sudden and nonphysical strong increase of local values of $v_{\theta,\star}$, we rewrite (10.56) as

$$v_{\theta,\star}(r_\star, t_\star) = \frac{\mathrm{Re}\,\mathrm{Sc}}{r_\star}\left(1 - \mathrm{e}^{-r_\star^2/(\mathrm{Sc}(R_\star^2 + 4\,t_\star))}\right), \tag{10.88}$$

for $t_\star \geq 10^{-4}$, where $R_\star = 10^{-1}$ is an initial radius for the viscous core [LC89].
As in the previous simulations of Chapter 9 with detailed chemistry, there is an important increase of the computational cost. Therefore, the same parallel computation technique for shared memory architectures was implemented for this configuration. It considers parallel and independent solutions of the reaction problem by space points, and by species for the diffusion and convection terms. The following simulations were performed on a 12 core (2x6) 64 bits machine (AMD-Shanghai 2.7 GHz processors) with memory capacity of 48 GB.

Numerical Results

Figure 10.7 shows the concentration of methane CH_4 and an intermediate species, the OH radical, at $t_\star = 5 \times 10^{-3}$. The latter one is often used to localize the flame front in experimental devices. The adaptive multiresolution shows also a good behavior in terms of local refinement even for a much larger set of variables, 49 compared with 19 in Chapter 9.
Although we have changed the dimensionless units with respect to the previous simple chemistry case, and that we are using a lower Reynolds number for the velocity field, the contour lines for Y_{CH_4} show a much thiner flame thickness as a result of the interaction of the premixed flame with the vortices. We recall that thermal expansion is neglected in this model and hence, thicker flames are expected otherwise. This flame thickness is of about 0.05, *i.e.,* approximatively 250 times smaller than the global scale of the computational domain. In order to have the same spatial resolution as in the previous case, we should use either at least a two times finer discretization, *i.e.,* a spatial mesh of 2048^2 points, or rescale more appropriately the problem. In particular, the representation of Y_{OH} in Figure 10.7 clearly shows that more points are needed for a better description of the front. The data compressions achieved for the current configuration are very high, more than 95 % according to (4.39), taking into account that the flame front is much more localized within the entire computational domain.
Figure 10.8 illustrates the velocity field computed according to (10.88) with a Reynolds number of $\mathrm{Re} = 625$. A less intense velocity field was considered in this case taking into account the inconsistencies coming from the modeling assumptions that were introduced to easily implement a detailed chemistry in the general problem (10.2)[9]. The splitting time step, dynamically adapted to handle the initial vortices, is adapted from $\Delta t^0 = 10^{-8}$ to an almost constant value of $\Delta t \approx 2 \times 10^{-6}$ ($\approx 1.4\,\mu$s) for an accuracy tolerance of $\eta = 10^{-2}$. For this configuration, there were always two convection time steps within the splitting time step according to

[9] For instance, for $\mathrm{Re} = 1000$ as in the previous cases, we observed convergence problems with the very simple Newton's method used to solve the nonlinear system (10.81) in our implementation. As previously discussed in the General Introduction, more sophisticated techniques are often developed and implemented for implicit solvers to efficiently solve these systems with complex chemistry features.

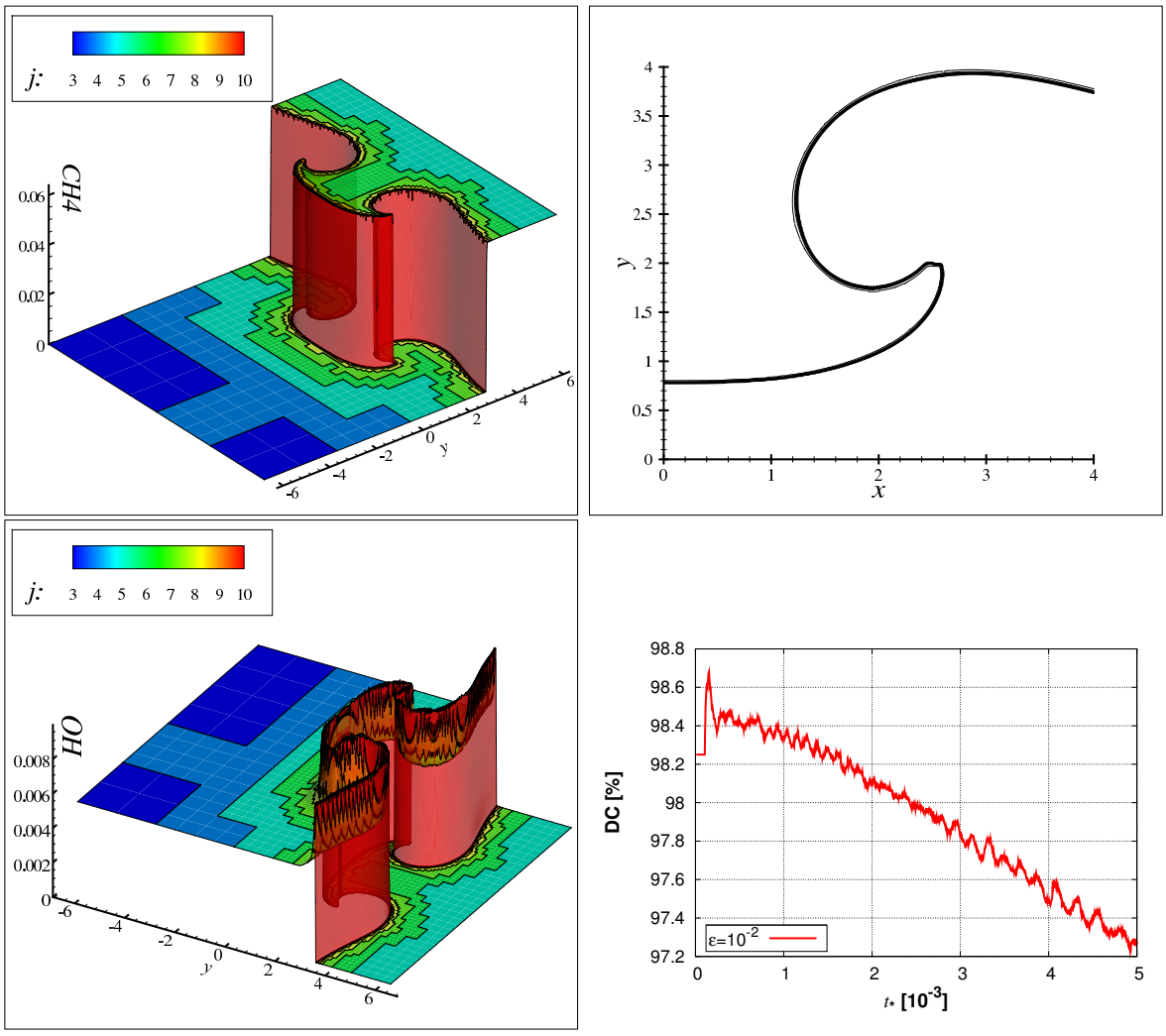

Fig. 10.7: *2D propagating flame with complex chemistry. Left: concentrations of* CH_4 *(top) and* OH *(bottom) at* $t_\star = 5 \times 10^{-3}$*, represented on an adapted grid corresponding to* 1024^2 *points at the finest level* $J = 10$ *with* $\varepsilon = 10^{-2}$*. Right: contour lines for* $Y_{CH_4} = 0 - 0.59$ *with* $\Delta Y_{CH_4} = 0.059$ *(top), and time evolution of data compressions* DC *given by (4.39) for* $\varepsilon = 10^{-2}$ *(bottom).*

(10.19), *i.e.*, $\Delta t_{C,i} = \Delta t/2$, $i = 1, 2$. For the given η_{ROCK4} and η_{Radau5} tolerances, the diffusion time steps were of the order of the splitting time step, *i.e.*, $\Delta t_D \approx \Delta t/2$, whereas the minimum reaction time steps at the front were of approximatively $\Delta t_R \approx 2 \times 10^{-7}$ (≈ 142 ns), gradually increasing up to $\Delta t_R = \Delta t/2$ elsewhere[10]. We recall that all these solvers are based on high order schemes, so that much smaller time steps would be required for standard lower order methods for each subproblem.

The CPU time for this simulation was of about 18.99 hours for $t_\star \in [0, 5.5 \times 10^{-3}]$ with a

[10] Even though a detailed comparison would be completely inappropriate, mainly because we do not consider exactly the same problem, let us give some details on the numerical simulations performed by Knio *et al.* in [KNW99] for a low Mach 2D methane-air premixed flame, interacting with a pair of counter-rotating vortices, and with chemical kinetics modeled by 32 species and 177 reactions. These authors calculated a maximum diffusion time step of 20 ns to ensure the stability of a second order explicit multi-step scheme and from there, a splitting time step of 200 ns. The minimum reaction time steps were of approximatively 8 ns for a tolerance of 10^{-6} for the implicit multi-step VODE solver [BBH89].

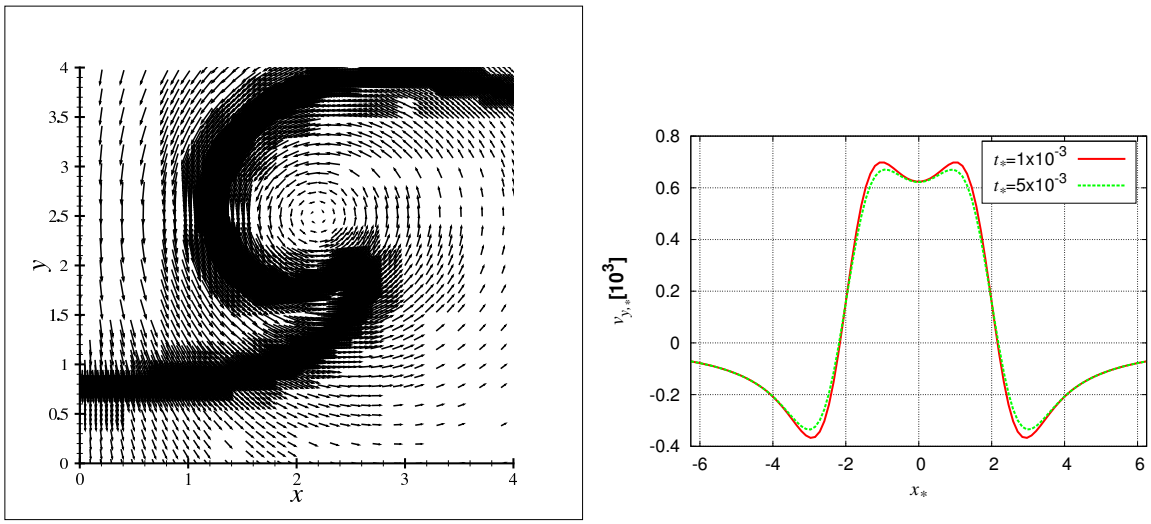

Fig. 10.8: *2D propagating flame with complex chemistry. Velocity field at* $t_\star = 5 \times 10^{-3}$ *(left), and* $v_{y,\star}$ *at* $y_\star = 2.5$*, at* $t_\star = 10^{-3}$ *and* $t_\star = 5 \times 10^{-3}$ *(right).*

gain of parallelization of 11.54, which is defined as the ratio between the computing time given by one single processor and the 12 cores in parallel. The ratio of CPU time per splitting time step corresponding to the reaction, diffusion, and convection time integration was of approximatively $23 : 5 : 1$, which justifies the high performance of the parallel technique for this particular configuration. Nevertheless, a more detailed study should be performed based on a more consistent model[11], and considering also a better initializing procedure, for instance, by means of Premix, as previously mentioned. These developments represent an important amount of work that is still in progress, and that is not sufficiently finalized to be presented here. These results prove nevertheless that the general algorithm and the numerical code can handle very complex reactive schemes, and in particular it can be easily coupled with Chemkin or any other chemistry library. In this way, this study illustrates the potential capabilities of the proposed numerical strategy, and constitutes a first preliminary stage towards the solution of fully transport and hydrodynamics models for reactive media.

10.5 Ignition of Diffusion Flames

In the same framework of problem (10.2) and the previous study on propagation of premixed flames, we illustrate in this part the performance of the proposed strategy to numerically describe the ignition dynamics of a diffusion flame interacting with a vortex. The mathematical model considered in this section was taken from a study conducted by Thévenin & Candel in [TC95]. A complete numerical and physical study on ignition dynamics based on these models can be found in [Thé92].

10.5.1 Model Formulation

Let us consider a 2D computational domain where pure and fresh hydrogen with mass fraction $Y_{\mathrm{F},0}$ at temperature $T_{\mathrm{F},0}$, occupies initially the half upper part. The remaining lower part is occupied by hot air at $T_{\mathrm{O},0}$ with $Y_{\mathrm{O},0}$. A single vortex modeled by (10.7) and centered on the

[11] In particular, we would not need to provide a numerical solver for the nonlinear system (10.81).

planar interface between the two media, is then introduced to transport and accelerate the mixture of both fluids. The governing equations of the physical phenomenon are thus given by the set of equations (10.2) with (10.5), with a reaction rate modeled by

$$\dot{w} = \frac{\rho^2}{\nu_{\mathrm{F}} W_{\mathrm{F}}} A Y_{\mathrm{O}} Y_{\mathrm{F}} \mathrm{e}^{(-T_a/T)}, \tag{10.89}$$

where A is a preexponential factor [TC95].
If we construct Schvab-Zeldo'vich variables by combining the reacting species mass fractions with proper coefficients, one obtains a balance equation without source term, analogous to the equation (10.5) governing Y_{N}. Thus, introducing the reduced total heat released χ; the normalized temperature difference between reactants τ; the product to fuel stoichiometric ratio σ; and the absolute equivalence ratio corresponding to a complete mixing between reactants in their initial state ϕ, defined, respectively, by

$$\chi = \frac{Q Y_{\mathrm{F},0}}{c_p T_{\mathrm{O},0}}, \quad \tau = \frac{T_{\mathrm{F},0} - T_{\mathrm{O},0}}{T_{\mathrm{O},0}}, \quad \sigma = \frac{\nu_{\mathrm{P}} W_{\mathrm{P}}}{\nu_{\mathrm{F}} W_{\mathrm{F}}}, \quad \phi = \mathrm{st} \frac{Y_{\mathrm{F},0}}{Y_{\mathrm{O},0}}, \tag{10.90}$$

with the stoichiometric factor st defined by (10.85), one may define the following variables:

$$\left.\begin{aligned} Z_1 &= \frac{\chi Y_{\mathrm{F}}/Y_{\mathrm{F},0} + \tau\theta}{\chi + \tau}, \\ Z_2 &= \frac{\chi Y_{\mathrm{O}}/(\phi Y_{\mathrm{O},0}) - \chi/\phi + \tau\theta}{-\chi/\phi + \tau}, \\ Z_3 &= \frac{-\chi Y_{\mathrm{P}}/(\sigma Y_{\mathrm{F},0}) + \tau\theta}{\tau}, \end{aligned}\right\} \tag{10.91}$$

where θ is the reduced temperature given by

$$\theta = \frac{T - T_{\mathrm{O},0}}{T_{\mathrm{F},0} - T_{\mathrm{O},0}}. \tag{10.92}$$

The set of variables (Z_1, Z_2, Z_3) are initially equal and follow the same balance equation without reaction term and with the same boundary conditions. Therefore, they are equal at each point and for all times to the same value Z. Introducing the same dimensionless variables previously defined in (10.54), we obtain a reduced system of equations of the form:

$$\left.\begin{aligned} &\partial_{t_\star} Z + v_{x,\star} \partial_{x_\star} Z + v_{y,\star} \partial_{y_\star} Z - \left(\partial^2_{x_\star} Z + \partial^2_{y_\star} Z\right) = 0, \\ &\partial_{t_\star} \theta + v_{x,\star} \partial_{x_\star} \theta + v_{y,\star} \partial_{y_\star} \theta - \left(\partial^2_{x_\star} \theta + \partial^2_{y_\star} \theta\right) = F(Z, \theta), \end{aligned}\right\} \tag{10.93}$$

with

$$F(Z,\theta) = \mathrm{Da}\, \phi \chi Y_{\mathrm{O},0} \left[\frac{1-Z}{\phi\tau} + \frac{1}{\chi}(Z-\theta)\right] \left[Z + \frac{\tau}{\chi}(Z-\theta)\right] \mathrm{e}^{(-\tau_a/(1+\tau\theta))}, \tag{10.94}$$

where $\tau_a = T_a/T_{\mathrm{O},0}$ is the reduced activation temperature, and the Damköhler number defined by $\mathrm{Da} = \rho A \tau_d$.

10.5.2 Numerical Simulations

We consider a 2D computational domain with initially two different media, given by fresh fuel and hot air. A single vortex modeled by (10.56) constitute the velocity field imposed on the domain. The following simulations were performed on an AMD Shanghai 2.7 GHz processor with memory capacity of 32 GB.

Data Initialization and Simulation Parameters

We consider the numerical solution of (10.93) with Neumann homogeneous boundary conditions in a 2D computational dimensionless domain of $[-1,1]^2$, *i.e.,* $L=1$ into (10.54). The following initial condition, taken from [Thé92], is considered:

$$Z(x_\star,y_\star)=\theta(x_\star,y_\star)=\frac{1}{2}\left[1+\tanh(\alpha y_\star)\right], \tag{10.95}$$

where $\alpha=200$. Notice that according to (10.91) and (10.92), $Z(x_\star,y_\star)=\theta(x_\star,y_\star)=1$ in the upper half-plane for the fresh fuel, and $Z(x_\star,y_\star)=\theta(x_\star,y_\star)=0$ in the lower part for the hot air. Therefore, considering the time dependent equation for Z in (10.93), Z must verify $Z\in[0,1]$ at any time. Similarly, $\theta\leq 1$ according to (10.92). The initial mass fractions and temperatures of the gases are given by

$$Y_{\mathrm{F},0}=1,\quad Y_{\mathrm{O},0}=0.23,\quad T_{\mathrm{F},0}=300\,\mathrm{K},\quad T_{\mathrm{O},0}=1000\,\mathrm{K}. \tag{10.96}$$

The following values together with (10.96) define the characteristic parameters (10.90), that allow the evaluation of the reaction rate (10.94) and the velocity field (10.56):

$$\frac{Q}{c_p}=5\times10^4\,\mathrm{K},\quad \mathrm{st}=8,\quad \mathrm{Da}=1.65\times10^7,\quad T_a=8000\,\mathrm{K}, \tag{10.97}$$

$$\mathrm{Re}=1000,\quad \mathrm{Sc}=1. \tag{10.98}$$

The velocity field is given by the counterclockwise vortex computed with (10.58) and centered at $(0,0)$. The accuracy parameters were fixed as for the corresponding 2D propagating flame with simple chemistry, *i.e.,* $\eta=10^{-3}$ in (10.11) with $\eta_{ROCK4}=10^{-5}$ and $\eta_{Radau5}=10^{-7}$. The time domain of integration is given by $t_\star\in[0,1.5\times10^{-4}]$. The multiresolution analysis was performed with $\varepsilon=10^{-2}$ and $\varepsilon=10^{-3}$ for a finest grid corresponding to a spatial discretization of 1024^2 points, *i.e.,* $J=10$ as finest grid level.

Numerical Results

Figure 10.9 shows the evolution of the temperature in the domain. The temperature is computed from θ by (10.92). There is initially fresh fuel at $T_{\mathrm{F},0}=300\,\mathrm{K}$ in the upper half-plane, whereas the remaining lower half contains hot air at $T_{\mathrm{O},0}=1000\,\mathrm{K}$. A counterclockwise rotating vortex modeled by (10.58) and centered at the planar interface is introduced immediately at $t_\star=0$. The resulting forced transport superposes to the diffusive mechanisms and accelerates the mixture of the gases. As a consequence, the local temperatures increase progressively from the vortex braids toward the vortex core along the contact surface of both media. A diffusion flame ignites along the contact surface. Notice that the velocity field entrains initially fresh gas into the vortex core and, subsequently, delays the fuel consumption of this central core by the diffusion flame. This behavior clearly depends on the initial configuration of the gases (10.96) and on the imposed velocity field, as studied in details in [Thé92, TC95]. The configuration described in this part reproduces the physics encountered in these previous studies and it was chosen in this work because it features important numerical difficulties. These issues are related to the severe transport conditions and the stiffness of the governing equations. Furthermore, this case is characterized by a sudden change of the physics at some instants which are not known at hand.

The corresponding adapted grids are also shown in Figure 10.9. We notice that the mixture lengths and the corresponding flame thickness are of the order of 0.025, similar to the previous

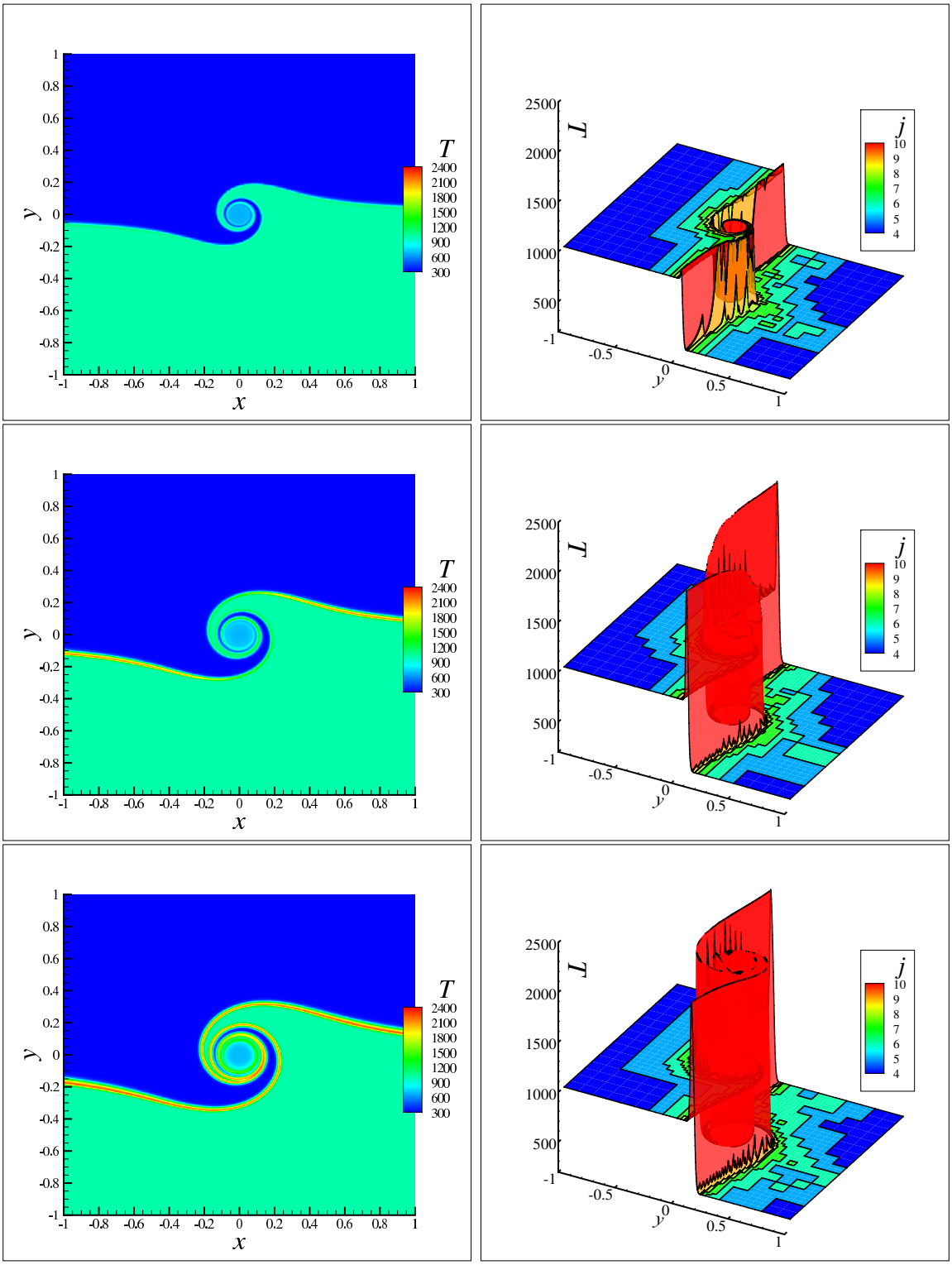

Fig. 10.9: *2D flame ignition. Time evolution of the temperature T calculated from θ in (10.92) at $t_\star = 5 \times 10^{-5}$ (top), 10^{-4} (middle), and 1.5×10^{-4} (bottom). Initial temperature of the fresh fuel: $T_{\mathrm{F},0} = 300\,K$, and of the hot air: $T_{\mathrm{O},0} = 1000\,K$. Right: dynamic adapted grid corresponding to 1024^2 points at the finest level $J = 10$ with $\varepsilon = 10^{-3}$.*

propagating case with simple chemistry. A spatial discretization of 1024^2 points is thus reasonably accurate. For this configuration, the finest regions dynamically identify the local rise in temperature until the ignition of the entire contact surface. In particular, the initial fronts do not require a full representation on the finest grid for a threshold tolerance of $\varepsilon = 10^{-3}$, recalling that the multiresolution analysis is performed on the dimensionless variables, *i.e.*, θ instead of T. Notice that for lower threshold values, the front will remain in the finest grid during the whole time period because the ignition and the subsequent formation of steeper gradients happens precisely at the flame front.

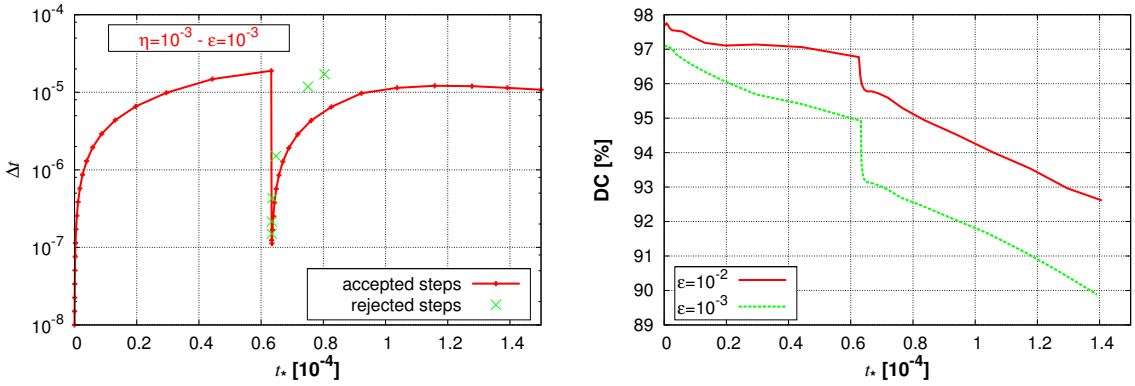

Fig. 10.10: *2D flame ignition. Time and space adaptation given by the time evolution of splitting time steps Δt with accuracy tolerance $\eta = 10^{-3}$ (left), and of data compressions* DC *given by (4.39) for $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-3}$ (right).*

Figure 10.10 illustrates the time and space adaptivity featured by the numerical strategy. An initial splitting time step of $\Delta t^0 = 10^{-8}$ was considered to handle the inclusion of the vortex and avoid unnecessary rejections at the beginning of computations. The splitting step increases until $t_\star \approx 6.5\times10^{-5}$ ($\Delta t \approx 2\times10^{-5}$) during the mixing phase, and then a series of rejected steps occurs for the given accuracy tolerance. The splitting time step is thus reduced down to the time scale needed to guarantee the prescribed accuracy: $\Delta t \approx 10^{-7}$. This behavior coincides naturally with the sudden ignition of the flame and the subsequent fast propagation along the contact surface, once a certain temperature is locally reached after the initial mixture of the gases. A dynamic adaptation of the splitting time step is hence mandatory to identify these changes in the physical behavior of the phenomenon, and to properly describe this process. This reduction of the evolution time steps allows us to update the spatial representation and to consequently refine as much as necessary the spatial configuration of the new physical scenario, as seen in Figure 10.10 for the time evolution of the data compressions. The capability of the method to rapidly update the mesh is illustrated by this limit case for which the chosen threshold parameter does not suffice to completely refine the initial front. Additionally, the high values of the achieved data compression justify a spatial adapted representation of the problem.

Other possible scenarios are illustrated in Figure 10.11 for different initial temperatures of the air $T_{O,0}$ and the same fuel temperature $T_{F,0} = 300\,\mathrm{K}$. Each configuration involves different dynamics in terms of time scales and final temperatures. It can be seen that for the time window of $t_\star \in [0, 1.5\times10^{-4}]$, there is only mixing for $T_{O,0} = 800\,\mathrm{K}$, whereas for $T_{O,0} > 1100\,\mathrm{K}$ ignition happens during the initial transition phase of the splitting time step so that no step reduction is needed. The interesting cases are obviously given by those in which ignition occurs at some unknown intermediate time. In any case, it can be seen that such an adaptive scheme can

handle all the possibilities without any preliminary information.

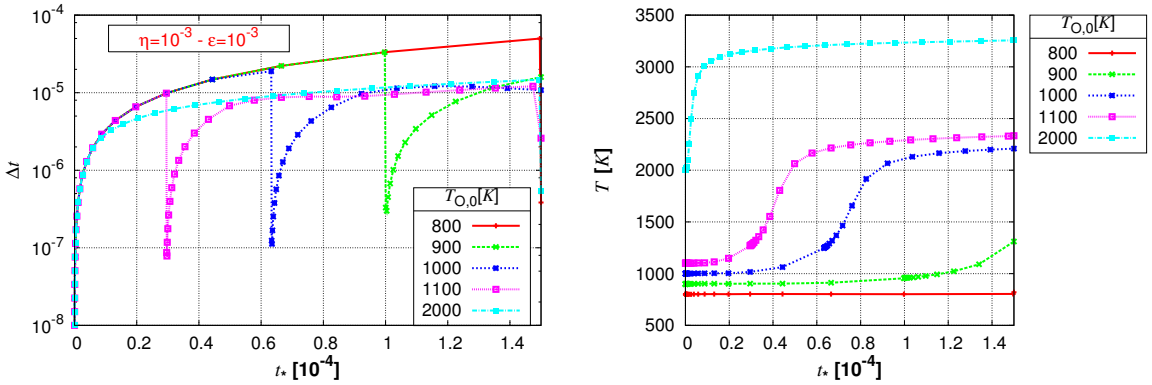

Fig. 10.11: *2D flame ignition. Time evolution of the splitting time step Δt and the maximum temperature for different initial temperatures of the hot air $T_{\mathrm{O},0}$ and $T_{\mathrm{F},0} = 300\,K$. Tolerances $\eta = 10^{-3}$ and $\varepsilon = 10^{-3}$.*

Table 10.4: *2D flame ignition. L^2 numerical errors for the time adaptive splitting (E^J_{split}), space adaptive multiresolution (E^J_{MR}), and time/space adaptive (E^{MR}_{split}) solutions evaluated at different times. Finest grid:* 1024^2.

$t_\star\,[10^{-4}]$	$\mathrm{E}^J_{split}\,[10^{-3}]$	$\mathrm{E}^J_{MR}\,[10^{-3}]$		$\mathrm{E}^{MR}_{split}\,[10^{-3}]$	
		$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$
0.5	0.46	6.48	2.12	6.48	2.11
1.0	2.25	13.1	5.77	12.9	5.35
1.5	1.53	37.7	2.69	37.9	3.65

Defining the splitting, multiresolution, and combined time/space adaptive approximation errors, respectively, by (10.64), (10.65), and (10.66), Table 10.4 shows these estimates evaluated for θ at different times $t_\star$. It can be seen that the splitting errors E^J_{split} are effectively controlled by the local error accuracy η, and that the global integration errors are approximatively of the same order taking into account that for this particular case, only a few time steps are required within the time interval of study. The multiresolution errors E^J_{MR} verify the proportionality with the threshold value ε. We remark that this behavior would not be possible without an adequate updating of the mesh issued from the adaptive time stepping technique during the ignition of the flame front, which proves the efficiency of the coupled adaptive time/space procedure. The numerical errors introduced by the dimensional splitting are also less important because the highest velocity gradients correspond approximatively to the spatial gradients of the solution, and thus to important local variations of the variables. Finally, we can see that for this configuration the global error E^{MR}_{split} of the method is mainly controlled by the multiresolution errors for $\varepsilon = 10^{-2}$, whereas for $\varepsilon = 10^{-3}$ the global error is a combination of time and space errors because both the splitting and the multiresolution errors are approximatively of the same order. Based on these and the previous results, we can verify that considering in general the same time and space accuracy tolerances, *i.e.*, $\eta = \varepsilon$, constitutes a good compromise

and yields satisfactory results in terms of accuracy and computational costs (see discussions in Chapter 5).
Table 10.5 summarizes the CPU times in minutes for the three alternatives. Important gains in CPU time are achieved with the adaptive splitting technique, which are moreover improved with the time/space adaptive strategy. The reductions related to the multiresolution representation are consistent with the achieved data compressions in Figure 10.10. Notice that the gains related to the adaptive splitting technique are this time less important because of the influence of the initialization phase in a rather short time domain. Finally, using only space adaptive multiresolution with constant splitting time step $\Delta t = 10^{-7}$, as in the quasi-exact approach, needs about 56.80 minutes of simulation.

Table 10.5: *2D flame ignition. CPU time in minutes for the quasi-exact, the time adaptive splitting, and the time/space adaptive strategies for* $t_\star \in [0, 1.5 \times 10^{-4}]$. *Finest grid:* 1024^2.

	MR/splitting $\varepsilon =$		splitting	quasi-exact
	10^{-2}	10^{-3}		
CPU time (m)	6.43	8.93	207.52	674.69

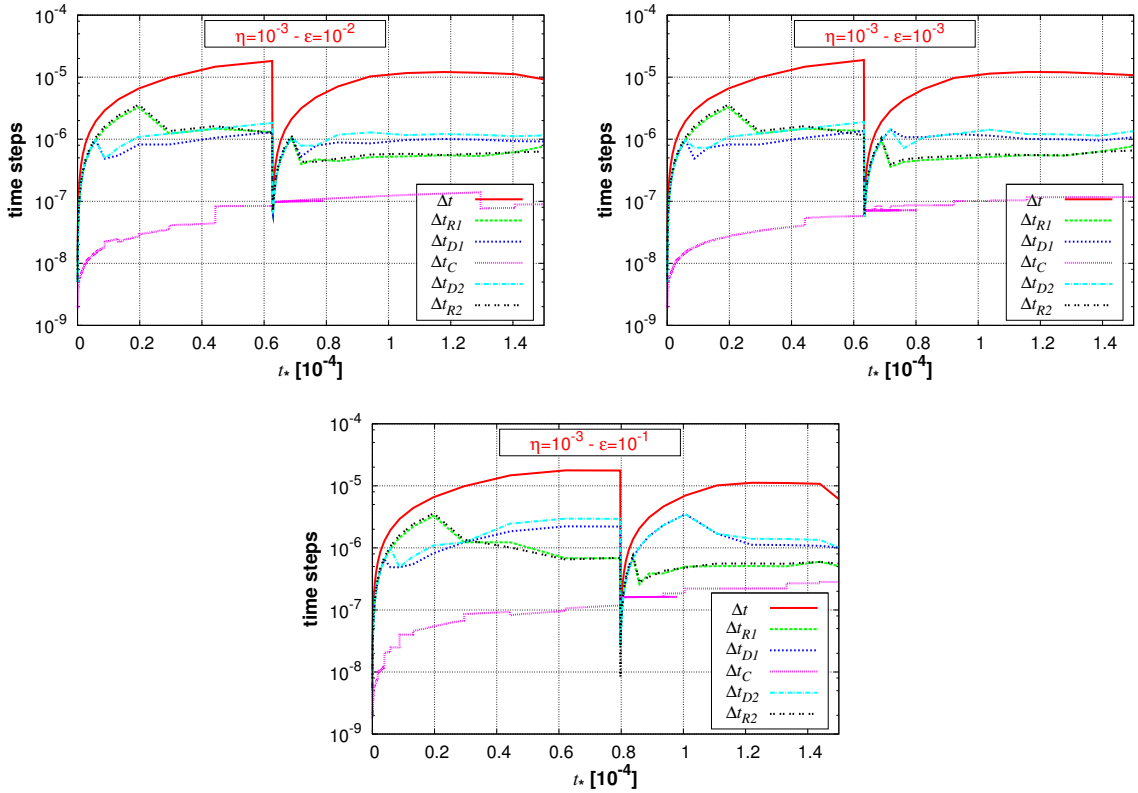

Fig. 10.12: *2D flame ignition. Top: time evolution of the splitting time step* Δt, *the reactive* Δt_{R1} *and* Δt_{R2}, *diffusive* Δt_{D1} *and* Δt_{D2}, *and convective* Δt_C *substeps according to the Strang scheme (10.10) with tolerances* $\eta = 10^{-3}$ *and* $\varepsilon = 10^{-2}$ *(left) or* $\varepsilon = 10^{-3}$ *(right). Bottom: illustrating case with coupling of time and space errors with* $\eta = 10^{-3}$ *and* $\varepsilon = 10^{-1}$.

The work distribution per splitting time step for the reaction, diffusion, and convection problems, is this time quite uneven in time as it can be inferred from Figure 10.12 that shows the corresponding time integrations steps for $\eta = 10^{-3}$, and $\varepsilon = 10^{-2}$ or $\varepsilon = 10^{-3}$. Actually, the work distribution is similar to the one given in Table 10.3 during the mixture and post-ignition phases. During the first ignitions and the small splitting time steps in the initial transient phases, the charge is rather balanced. Once again, the convection time step is the most constraining step considering the high Reynolds values in the velocity field. The diffusion time stepping performed by ROCK4 is quite similar to the previous case for we are considering the same spatial discretization in a computational domain of the same size, *i.e.,* the spectral radius of the Laplacian operators are of the same order. Notice that there is a slight shift in $t_\star$ for the splitting time steps with threshold values $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-3}$ in Figure 10.12. Nevertheless, the normal behavior of the multiresolution errors in Table 10.4 confirms that the coupling of time and space errors is still acceptable. A counter example is given by the case with $\varepsilon = 10^{-1}$ for which there is surely an important coupling of errors that results in a different and likely wrong numerical description of the physical behavior. In this way, we illustrate that the choice of appropriate accuracy tolerances, highly problem dependent, can be easily evaluated in practice, as explained in Chapter 5.

10.6 Concluding Remarks

In this chapter, we have implemented the time/space adaptive numerical strategy developed in chapters 4 and 5, for the numerical simulation of combustion fronts modeled by reaction-diffusion-convection systems. We have thus included a convective term to be handled by the operator splitting technique. Let us summarize some key aspects of this configuration:

- We have implemented the OSMP method for the numerical solution of the convection subproblem, following the general precepts discussed in Chapter 4 for the operator splitting method. The OSMP method is a one-step high order scheme, developed by Daru & Tenaud in [DT04].
- The extension to multi-dimensional configurations have been done by means of a dimensional splitting, which reduces considerably the algorithmic complexity as well as the computational requirements.
- Second order Strang dimensional splitting was implemented as well as a simple corrective procedure to reduce potential numerical oscillations introduced by the splitting approach for divergence free velocity fields.
- Taking into account the fine spatial meshes often needed to simulate propagating fronts, a dynamic time stepping was considered for the convection time steps within each splitting time step, computed based on stability constraints.

The numerical method has been then evaluated for different problems issued from combustion applications and studies, namely:

- The propagation of premixed flames in 2D/3D with simplified or complex chemical source terms.
- The ignition of a reactive mixture and the subsequent propagation of the resulting diffusion flames in 2D.

The numerical results obtained with this technique allow us to conclude that different physical configurations can be successfully simulated with the same time/space adaptive strategy, either in 2D or 3D, providing effective error control of the approximation errors related to the numerical methods. Along with other advantages already illustrated and discussed in previous chapters, the time adaptation procedure turns out to be a critical tool to efficiently simulate both general configurations, not only to ensure a prescribed accuracy of the computations but also to properly handle the initial strong transients associated with the velocity fields. Furthermore, the efficiency of the method is assessed for a difficult problem such as the sudden ignition of a flame, with remarkable different physics and time scale spectra with respect to the initial and final configurations. In all cases, important gains in computational efficiency are achieved, related to compressed data representations as well as a dedicated splitting technique with adequate solvers and independent time stepping procedures.
We recall that all inner solvers are based on high order schemes so that considerably smaller time steps would have been required for standard low order schemes. Nevertheless, we can still observe an important decoupling of the time scale spectra which allows us to overcome stability restrictions associated with the explicit solvers, and further gains in efficiency for splitting time steps not limited by the numerical particularities associated with each subproblem. Preliminary results were also presented for problems that include detailed chemical features in order to show the capability of the method to treat more complex configurations. However, let us remark that the simplified chemical terms retain and mimic some fast scale features and thus numerically stiff behaviors of the comprehensive formulation. This can be inferred, for instance, from the rather small time steps needed to solve the reaction problem, although a fifth order scheme like Radau5 has been implemented.
A key question in the previous numerical illustrations is related to the numerical accuracy of the numerical results. We have seen that the present numerical strategy allows us to better control the numerical errors of the simulations in a very easy and simple way. The level of computational complexity can be thus illustrated by the fact that for all the problems considered, we have only needed to settle two parameters:

- The threshold value ε of the multiresolution decomposition, which balances data compression and numerical errors associated with compressed data representations; and
- The accuracy tolerance η of the splitting time integration technique, which limits the degree of decoupling of the physical phenomena and hence, controls the numerical time integration errors.

We have in particular illustrated how these parameters are selected in practice, based on principles previously established in Chapter 5.
These numerical results are very satisfactory and have demonstrated that the method is capable to deal with different configurations for general multi-scale problems modeled by time dependent PDEs. Nevertheless, there are at least two main aspects that still require further developments and studies:

- A time stepping procedure for the convective scheme that takes also into account the accuracy of the computations, specially for problems with potentially large convective time integration steps; and
- The coupling of the numerical method with a hydrodynamics solver in order to properly evaluate the performance of the method and to extend the applicability of the strategy to this kind of problem.

Both matters are part of our current research.

Chapter 11

Plasma Application: Positive Streamer Simulations

We focus in this chapter on the numerical simulation of plasma discharges at atmospheric pressure, physically modeled by highly nonlinear ionizing waves called streamers. The detailed study and comprehensive description of such phenomena is of the utmost importance for many modern applications, and their numerical simulation constitutes a powerful tool in this regard. Nevertheless, the detailed physics associated with plasma discharges reveals an important time-space multi-scale character which demands great investments in accurate mathematical modeling as well as dedicated and efficient numerical schemes. In this context, the study of streamer discharges becomes an appropriate framework to evaluate and further develop the numerical strategy introduced in this work.
Streamer discharges are usually modeled by stiff time dependent PDEs of drift-diffusion type coupled with a Poisson's equation for the computation of the electric field. Therefore, the time/space adaptive numerical strategy detailed in the previous Chapter 10 for stiff reaction-diffusion-convection equations, can be implemented to efficiently solve the drift-diffusion equations of the plasma model. Nevertheless, the solution of the electric field is also required to simulate such models, and further developments and extensions to the present strategy are consequently needed. In order to conduct these simulations, we have developed a new second order strategy that couples the solution of the electric field with the drift-diffusion equations. The latter scheme also features time adaptivity with error control. Important gains in the numerical efficiency of the method are thus achieved for highly unsteady problems, while a prescribed accuracy is guaranteed by dynamic local error evaluations. The time/space numerical strategy presented in the previous chapter was therefore embedded into a second time adaptive scheme to solve these new set of governing equations for plasma applications.
This new solution scheme was numerically evaluated with satisfactory results for the simulation of propagating streamers, and in the context of highly nanosecond repetitively pulsed discharges in 1D. Extensions to multi-dimensional configurations are straightforward with an adequate numerical solver of the Poisson's equation. This study has motivated an article which has been recently published in Journal of Computational Physics in a special issue on Computational Plasma Physics [DBM+12]. We reproduce in what follows this article in its integral version because it constitutes an extension to the present numerical strategy for this particular application framework. Additionally, we first present a preliminary study on the solution of a simplified reaction-diffusion plasma model by the adaptive splitting technique of Chapter 5. The main goal is to evaluate the numerical capabilities of the method in the context of multi-pulsed gas discharges involving several dynamics with very different typical time scales.

In particular, this study settled the foundations of the numerical method developed in the article, presented in the second part of this chapter.

11.1 Adaptive Splitting on a Simplified Plasma Model

We present in this part a brief study on the adaptive splitting scheme introduced in Chapter 5 for the solution of a simplified plasma model. This study will also give a first insight into the numerical difficulties encountered during the simulation of such multi-scale problems, and serves as an appropriate introduction to the second part.

11.1.1 Mathematical Model

Let us consider a simplified reaction-diffusion model:

$$\left.\begin{aligned} \partial_t n_{\mathrm{e}} - D\,\partial_x^2 n_{\mathrm{e}} &= n_{\mathrm{e}}\alpha|\mathrm{v_e}| - n_{\mathrm{e}}\eta|\mathrm{v_e}| + n_{\mathrm{e}}n_{\mathrm{p}}\beta_{\mathrm{ep}},\\ \partial_t n_{\mathrm{p}} - D\,\partial_x^2 n_{\mathrm{p}} &= n_{\mathrm{e}}\alpha|\mathrm{v_e}| - n_{\mathrm{e}}n_{\mathrm{p}}\beta_{\mathrm{ep}} + n_{\mathrm{n}}n_{\mathrm{p}}\beta_{\mathrm{np}},\\ \partial_t n_{\mathrm{n}} - D\,\partial_x^2 n_{\mathrm{n}} &= n_{\mathrm{e}}\eta|\mathrm{v_e}| - n_{\mathrm{n}}n_{\mathrm{p}}\beta_{\mathrm{np}}, \end{aligned}\right\} \tag{11.1}$$

based on the drift-diffusion equations usually defined to describe the motion of ions and electrons in plasma models [BN96, Kul97], where n_i is the density of species i =e, p, n (e: electrons, p: positive ions, n: negative ions); $\mathrm{v_e}$ is the electron drift velocity; α is the impact ionization coefficient; η stands for the electron attachment on neutral molecules; and β_{ep} and β_{np} accounts, respectively, for the electron-positive ion and negative-positive ion recombinations. All the coefficients of the model are functions of the local reduced electric field E/N_{gas}, where E is the electric field magnitude and N_{gas} is the air neutral density. The reaction parameters for the air are taken from [ML97], with attachment coefficients taken from [KKMS92].

11.1.2 Numerical Configuration

In this numerical illustration, we consider as computational domain an air gap of 0.5 cm, where we have a high initial distribution of electrons and ions over the region $[0, 0.01]$ (cm). A constant electric field of $\sim 40\,\mathrm{kV/cm}$ is then repetitively applied over this region during 10 ns with a period of $1\,\mu$s. All parameters in (11.1) are computed with the imposed field. Finally, we consider a constant diffusion coefficient: $D = 50\,\mathrm{cm}^2/\mathrm{s}$, and a spatial discretization of 1001 points for which we have negligible spatial discretization errors with respect to the ones coming from the numerical time integration. Figure 11.1 shows the spatial distribution of electron density just before and after each pulse. Generally speaking, there are at least two completely different physical configurations given either by a high reactive activity when the electric field is applied, or by the propagative nature of the post-discharge phase.
For the solution of system (11.1) we consider the Strang scheme $\mathcal{S}_2^t$ in (5.5) and the embedded one $\mathcal{S}_{2,\delta}^t$ (5.7), considering the adaptive strategy described in Chapter 5 with initial splitting time step $\Delta t^0 = 10^{-10}$ and $\delta^0 = 0.05$, and with the following parameters previously considered:

- $\delta_{max} = 0.999$;
- $a_1 = 1$, $b_1 = c_1 = a_2 = 1/2$, $b_2 = 2/5$, and $c_2 = 1/10$ for the intermediate time steps, as in (5.59);
- $\zeta = 0.9$ as security factor of the critical $\Delta t^\star$ estimate;
- $\beta = 0.1$ and $\gamma = 0.95$ to define the working region (5.60);

- $\theta = 10$ as security factor of the δ estimate;
- $C_\delta^0 = 10$ to potentially reject the initial time step Δt^0;
- and $\upsilon = 0.9$ as security factor of the Δt^{new} estimate.

The time integration is performed with the dedicated splitting technique introduced in Chapter 4, with Radau5 for the reactive term and the ROCK4 method for the diffusion problem, both with fine tolerances, $\eta_{Radau5} = \eta_{ROCK4} = 10^{-10}$. Let us remark that the computation must be initialized with a time step included in the pulse duration.

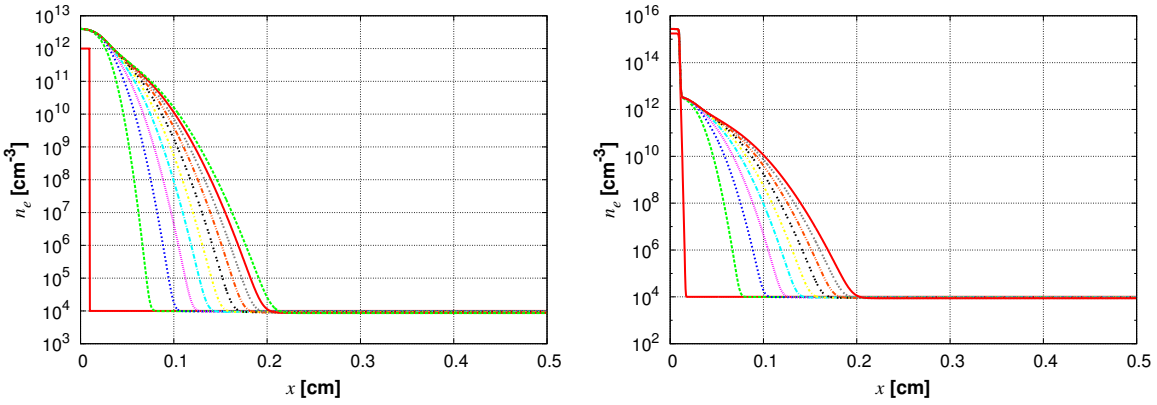

Fig. 11.1: *1D simplified plasma model. Spatial distribution of electron density before (left) and after (right) each pulse, starting from a step-like initial distribution (left) and for a duration of ten pulses.*

11.1.3 Numerical Results

Figure 11.2 shows the corresponding splitting time steps for a tolerance of $\eta = 10^{-3}$ in (5.11). The splitting time step features a periodic behavior and succeeds to consistently adapt itself to the discharge/post-discharge phenomena. This yields high varying time steps going from $\sim 10^{-10}$ to $\sim 10^{-7}$. Therefore, after each post-discharge phase, since the new time step is computed based on the previous one according to (5.11), this new time step will surely skip the next pulse. In order to avoid this, each time we get into a new period, we initialize the time step with the length of the pulse: $\Delta t = 10\,\text{ns}$. This time step is obviously rejected as seen in Figure 11.2, as well as the next ones, until we are able to retrieve the right dynamics of the phenomenon for the required accuracy tolerance. No other intervention is needed either in terms of modeling parameters or for the numerical solvers, in order to automatically adapt the time step to the several time scales of the phenomenon within a prescribed accuracy.
For this application, we compute the critical $\Delta t^\star$ and possibly δ for $N_\delta = 10$ and $N_\delta = 100$ in each period, in order to perform these computations at least once during the discharge and post-discharge regimes. For example, for $t \in [5, 6]\,\mu$s as in Figure 11.2, $\delta = \delta_{max}$ with $\Delta t^\star \approx 4.3 \times 10^{-9}$ during the pulse, and $\delta \approx 0.26$ with $\Delta t^\star \approx 1.6 \times 10^{-7}$ for the rest of the period. Similar values are found for the other periods. Notice that after each pulse, $\Delta t^\star$ is automatically updated because Δt increases and then Δt gets equal to $\Delta t^\star$. In particular, the important difference between $\Delta t^\star$ for each region, results naturally from the completely different modeling parameters and hence, physics description of each regime.
An effective error control is achieved for each phase of the phenomenon, as we can deduce from the global error between splitting and reference solutions at the end of the pulse ($t = 5.01\,\mu$s),

and at the end of the post-discharge phase ($t = 6\,\mu$s). The reference solution is computed by a coupled resolution of system (11.1) with Radau5 and fine tolerance $\eta_{Radau5} = 10^{-10}$. If we compare these results with the ones obtained without estimating either $\Delta t^\star$ or δ, but with $\delta = \delta^0 = 0.05$, we can draw the same conclusions as in the BZ application in Chapter 5. For less accurate resolutions with high tolerances, the proposed strategy corrects the local error estimates computed $\delta = \delta^0 = 0.05$. In particular, for $\eta = 10^{-3}$ there is a ratio of about 10 between both solutions. For higher tolerances, $\eta \geq 10^{-2}$, both methods yield a time step equal to the pulse duration, $\Delta t = 10\,$ns. For the smallest tolerances, slightly more accurate solutions are obtained with a fixed $\delta = \delta^0$ because smaller splitting time steps are used.

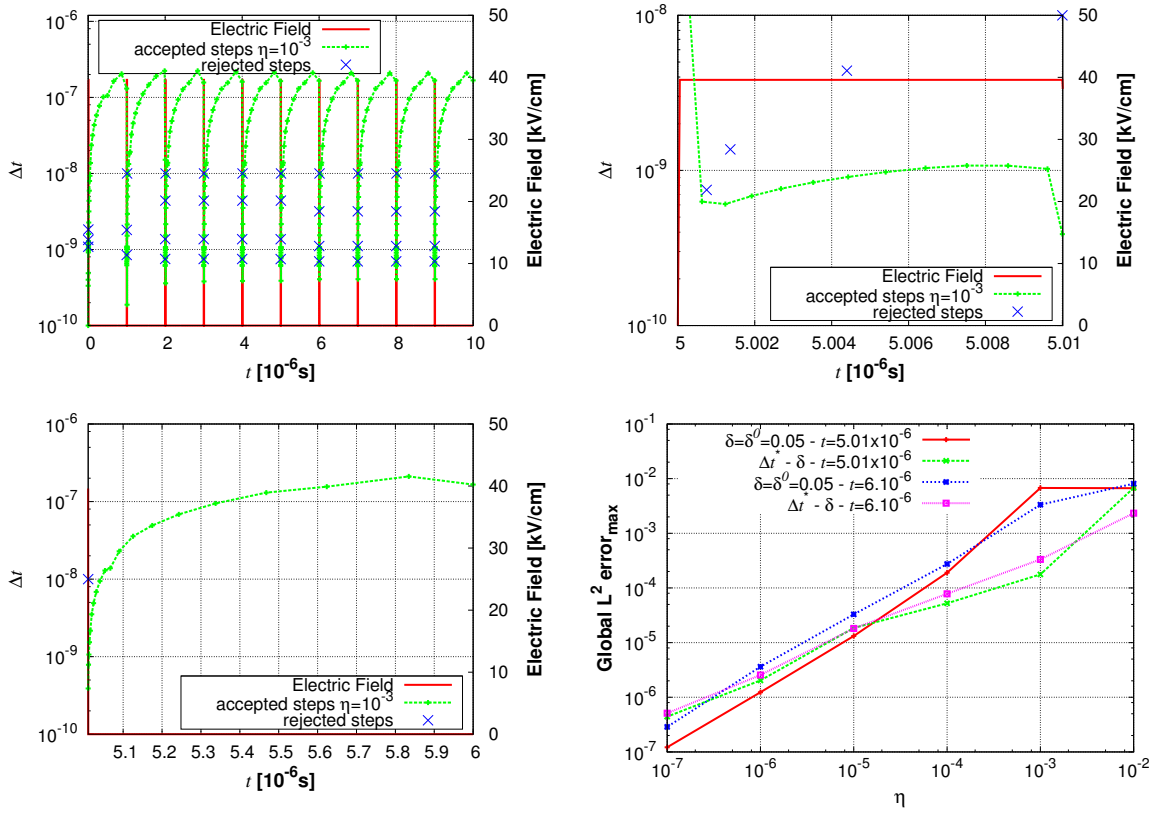

Fig. 11.2: *1D simplified plasma model. Time evolution of accepted and rejected splitting time steps, and the imposed electric field for $t \in [0, 10]\,\mu s$ (top left), during the pulse $t \in [5, 5.01]\,\mu s$ (top right), and the post-discharge phase $t \in [5.01, 6]\,\mu s$ (bottom left). Bottom right: global L^2 errors at the end of the pulse ($t = 5.01\,\mu s$) and the post-discharge phase ($t = 6\,\mu s$), with and without computation of critical $\Delta t^\star$ and time shift δ.*

To conclude this preliminary study, let us emphasize that compared with a standard procedure for which the accuracy is guaranteed by considering time steps of the order of the fastest scale, the error control featured by this adaptive method implies an effective accurate resolution of problems modeling various physical scenarios, independent of the fastest physical or numerical time scale. Additionally, an important improvement of computational efficiency is achieved for highly unsteady phenomena. In particular, the technique was successfully applied to this simplified model of plasma discharges that nevertheless exhibits a broad time scale spectrum coming from the modeling equations, and also important and discontinuous variation of parameters in time and in space that notably increase the numerical complexity of the problem. The complete drift-diffusion equations with the computation of the electric field will be treated

in details in the following part, in which the same adaptive technique will be embedded into a new one for plasma models.

A NEW NUMERICAL STRATEGY WITH SPACE-TIME ADAPTIVITY AND ERROR CONTROL FOR MULTI-SCALE STREAMER DISCHARGE SIMULATIONS

Max Duarte, Zdeněk Bonaventura, Marc Massot, Anne Bourdon, Stéphane Descombes, and Thierry Dumont

Abstract. This paper presents a new resolution strategy for multi-scale streamer discharge simulations based on a second order time adaptive integration and space adaptive multiresolution. A classical fluid model is used to describe plasma discharges, considering drift-diffusion equations and the computation of electric field. The proposed numerical method provides a time-space accuracy control of the solution, and thus, an effective accurate resolution independent of the fastest physical time scale. An important improvement of the computational efficiency is achieved whenever the required time steps go beyond standard stability constraints associated with mesh size or source time scales for the resolution of the drift-diffusion equations, whereas the stability constraint related to the dielectric relaxation time scale is respected but with a second order precision. Numerical illustrations show that the strategy can be efficiently applied to simulate the propagation of highly nonlinear ionizing waves as streamer discharges, as well as highly multi-scale nanosecond repetitively pulsed discharges, describing consistently a broad spectrum of space and time scales as well as different physical scenarios for consecutive discharge/post-discharge phases, out of reach of standard non-adaptive methods.

Keywords. *Multi-scale discharge; Time adaptive integration; Space adaptive multiresolution; Error control.*

Mathematics Subject Classification. 65M08, 65M50, 65Z05, 65G20.

Journal of Computational Physics 231 (2012) 1002–1019
DOI: 10.1016/j.jcp.2011.07.002
Published online July 19, 2011.

11.2 Introduction

In recent years, plasma discharges at atmospheric pressure have been studied for an increasing list of applications such as chemical and biological decontamination, aerodynamic flow control and combustion [vV00, FCG05]. In all these physical configurations, the discharges take usually the form of thin plasma filaments driven by highly nonlinear ionizing waves, also called streamers. These ionizing waves occur as a consequence of the high electric field induced by the fast variations of the net charge density ahead of an electron avalanche with large amplification. The streamer discharge dynamics are mainly governed by the Courant, the effective ionization and the dielectric relaxation times scales [VPB94], which are usually of the order of $10^{-14} - 10^{-12}$s, whereas the typical time scale of the discharge propagation in centimeter gaps, is about a few tens of nanoseconds. On the other hand, a large variation of space scales needs also to be taken into account, since the Debye length at atmospheric pressure can be as small as a few micrometers, while the inter-electrode gaps, where discharges propagate, are usually of the order of a few centimeters. As a result, the detailed physics of the discharges reveals an important time-space multi-scale character [UBRT10, EBD+11].
More complex applications include plasma assisted combustion or flow control, for which the enhancement of the gas flow chemistry or momentum transfer during typical time scales of the flow of $10^{-4} - 10^{-3}$s, is due to consecutive discharges generated by high frequency (in the kHz range) sinusoidal or pulsed applied voltages [PGL+06, OSM+08]. Therefore, during the post-

discharge phases of the order of tens of microseconds, not only the time scales are very different from those of the discharge phases of a few tens of nanoseconds, but a completely different physics is taking place. Then, to the rapid multi-scale configuration during discharges, we have to add other rather slower multi-scale phenomena in the post-discharge, such as recombination of charged species, heavy-species chemistry, diffusion, gas heating and convection. Therefore, it is very challenging to accurately simulate the physics of plasma/flow interactions due to the synergy effects between the consecutive discharge/post-discharge phases.

In most numerical models of streamer discharges, the motion of electrons and ions is governed by drift-diffusion equations coupled with Poisson's equation. Early simulation studies were limited to simplified situations where the streamer is considered as a cylinder of constant radius [DJE64, DDE71, AB80, Mor85], in which the charged particle densities are assumed to be constant along the radial extension of the streamer: the *1.5D model* approach. In this model, the spatio-temporal evolution of the charged particle densities is solved only along one spatial dimension in the direction of propagation, whereas the electric field is calculated in two dimensions using the so-called *disc method*, based on a direct integration of analytical results. A 2D model for the electric field is indeed essential to properly calculate the electric field enhancement by the space charge in the streamer head. After the first 2D streamer simulations using the Poisson's equation resolution were performed [DW87], many studies have been carried out in 2D [VPB94, BN97, Kul00, PSS01, AEH02, CBZ+09, BBC10] and 3D [NAK08, PSCB08, LEH08, PMG11].

Being aware of the complexity of fully coupled resolutions of these modeling equations, a decoupling strategy is usually adopted, which considers an independent and successive numerical resolution of Poisson's equation with a fixed charge distribution, and of the drift-diffusion equations with a fixed electric field during each decoupling time step. These computations might be performed explicitly in time with standard first or even second order schemes [MHE06, BPL+07]. In these cases, the time steps are usually limited for the sake of stability by the various characteristic times scales (Courant, ionization, dielectric relaxation), whereas the accuracy of simulations is assumed to be given by the resolution of the fastest physical time scale. In order to somehow overcome the dielectric relaxation limitation, some semi-implicit approaches were developed [VSHK93, CDW99, HK00], based on a predictive approximation of the space charge ahead in time during the electric field computation, even though the other time scale constraints remain. This gain of stability allows important improvements in terms of computational efficiency but the accuracy of simulations becomes rather difficult to quantify.

Another performing technique to improve the efficiency of simulations considers an asynchronous explicit time integration of the drift-diffusion equations with self-adaptive local time-stepping, for which the local time steps are based either on local dynamic increments of the solution [KDOO05, OK06] or on local Courant conditions [UBRT07]. These techniques are the subject of several studies [CNPT08, DGRS08, CNPT10] and are mainly conceived to avoid expensive computations whenever the whole system is unnecessarily advanced in time with a global time step prescribed by the fastest scale. Even though these methods yield efficient strategies, specially in terms of CPU time savings, with stable and flux-conserving time integrations, it is rather difficult to conduct an accuracy control on the resolution of the time dependent equations or on their coupling with the electric field resolution for plasma models.

In this work, a numerical study is conducted in order to build a second order explicit in time decoupling scheme for the resolution of the electric field and the electron and ion densities. A lower order and embedded method is taken into account to dynamically compute the decoupling time steps that guarantee an accurate description with error control of the global physical coupling. At this stage, the only limiting time scale is the dielectric relaxation characteristic time for stability reasons. In a second level, the drift-diffusion equations are solved using a

Strang second order operator splitting scheme in order to guarantee the global order of the strategy [DMD+12, DDD+12]. This time integration scheme considers high order dedicated methods during each splitting time step, which is dynamically adapted by an error control procedure [DDD+11]. In this way, even though there is a global advance in time given by the splitting time step, the latter is determined by the desired accuracy of the global physics, which is not necessarily related to the stability constraints associated with the mesh size or the fastest source time scales as demonstrated in [DMDD11]. As a consequence, this technique provides an error control procedure and stands as an alternative way to local stepping schemes to overcome time step limitations related to the reaction, diffusion and convection phenomena. Both the electric field and density resolutions are performed on an adapted mesh obtained by a spatial multiresolution method, based on Harten's pioneering work [Har95] and further developed in [CKMP03], taking into account the spatial multi-scale features of these phenomena with steep spatial gradients. In particular, some grid adaptation techniques for 2D structured meshes were already used [MHE06, PSCB08, UBRT10] and extensions to 3D have been also proposed [PSCB08, NAK08] for streamer simulations. However, one of the main advantages of the multiresolution approach is that it is based on a wavelet representation technique and an error of the spatial approximation can be then mathematically estimated. Consequently, an effective error control is achieved for both the time and space resolution of the multi-scale phenomena under study.
The performance of the method is first evaluated for a propagating streamer problem with the multi-scale features previously discussed, for which the various simulation parameters are studied. Once the physical configuration is settled, a 1.5D streamer model is adopted in order to obtain an electric field resolution strategy based on direct computations and derived from analytical expressions, suitable for adapted finite volume discretizations [BPB+07]. In a second step, a more complex physical configuration is considered for the simulation of repetitively pulsed discharges, for which a time-space adaptive method is required to efficiently overcome some highly multi-scale features in order to fully describe the various physical phenomena. In this work, only a 1.5D model is considered but extensions to higher dimensions is straightforward for instance with a Poisson's equation solver for adapted grids as it has been implemented in [MHE06, PSCB08, UBRT10]. However, in this paper we focus on the development and validation of new numerical methods for the resolution of the drift-diffusion equations and its coupling with the electric field computation, which are independent of the dimension of the problem. Numerical illustrations of multidimensional problems with the same time-space adaptive strategy with error control will be the subject of future work.
The paper is organized as follows: in Section 11.3, we present the physical configuration and the modeling equations. The numerical strategy is presented in Section 11.4, in which the second order adaptive time integration technique is detailed along with the resolution of drift-diffusion equations and the electric field, as well as the spatial multiresolution adaptive procedure. Numerical illustrations are summarized in Section 11.5 for two configurations given by single propagating and multi-pulsed discharges. We end with some concluding remarks and prospects on future developments and applications.

11.3 Model Formulation

In this work, we consider positive streamer discharges in air at atmospheric pressure in a point-to-plane geometry, as shown in Figure 11.3. The tip of the anode is placed 1 cm from the planar cathode and the radius of curvature of the anode is 324 μm. The most common and effective model to study streamer dynamics is based on the following drift-diffusion equations

for electrons and ions, coupled with Poisson's equation [BN96, Kul97]:

$$\left.\begin{aligned}
\partial_t n_\mathrm{e} - \partial_\mathrm{x}\cdot n_\mathrm{e}\,\mathrm{v}_\mathrm{e} - \partial_\mathrm{x}\cdot(D_\mathrm{e}\,\partial_\mathrm{x} n_\mathrm{e}) &= n_\mathrm{e}\alpha|\mathrm{v}_\mathrm{e}| - n_\mathrm{e}\eta|\mathrm{v}_\mathrm{e}| + n_\mathrm{e}n_\mathrm{p}\beta_\mathrm{ep} + n_\mathrm{n}\gamma,\\
\partial_t n_\mathrm{p} + \partial_\mathrm{x}\cdot n_\mathrm{p}\mathrm{v}_\mathrm{p} - \partial_\mathrm{x}\cdot(D_\mathrm{p}\,\partial_\mathrm{x} n_\mathrm{p}) &= n_\mathrm{e}\alpha|\mathrm{v}_\mathrm{e}| - n_\mathrm{e}n_\mathrm{p}\beta_\mathrm{ep} + n_\mathrm{n}n_\mathrm{p}\beta_\mathrm{np},\\
\partial_t n_\mathrm{n} - \partial_\mathrm{x}\cdot n_\mathrm{n}\mathrm{v}_\mathrm{n} - \partial_\mathrm{x}\cdot(D_\mathrm{n}\,\partial_\mathrm{x} n_\mathrm{n}) &= n_\mathrm{e}\eta|\mathrm{v}_\mathrm{e}| - n_\mathrm{n}n_\mathrm{p}\beta_\mathrm{np} - n_\mathrm{n}\gamma,
\end{aligned}\right\} \tag{11.2}$$

$$\varepsilon_0\,\partial_\mathrm{x}^2 V = -q_\mathrm{e}(n_\mathrm{p} - n_\mathrm{n} - n_\mathrm{e}), \tag{11.3}$$

where $\mathrm{x}\in\mathbb{R}^d$, n_i is the density of species i (e: electrons, p: positive ions, n: negative ions), V is the electric potential, $\mathrm{v}_i = \mu_i \mathrm{E}$ (E being the electric field) is the drift velocity. D_i and μ_i, are the diffusion coefficient and the absolute value of mobility of the charged species i, q_e is the absolute value of an electron charge, and ε_0 is the permittivity of free space. α is the impact ionization coefficient, η stands for the electron attachment on neutral molecules, β_ep and β_np account respectively for the electron-positive ion and the negative-positive ion recombination, and γ is the detachment coefficient.

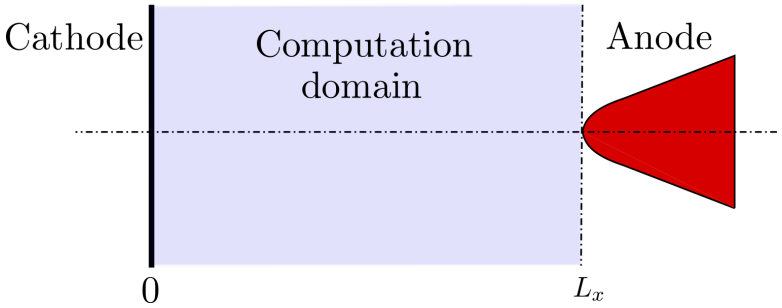

Fig. 11.3: *Computational domain for the studied point-to-plane geometry.*

The electric field E and the potential V are related by

$$\mathrm{E} = -\partial_\mathrm{x} V, \tag{11.4}$$

and thus, the Poisson's equation (11.3) becomes:

$$\varepsilon_0\,\partial_\mathrm{x}\cdot\mathrm{E} = q_\mathrm{e}(n_\mathrm{p} - n_\mathrm{n} - n_\mathrm{e}). \tag{11.5}$$

All the coefficients of the model are assumed to be functions of the local reduced electric field E/N_gas, where E is the electric field magnitude and N_gas is the air neutral density. For test studies presented in this paper, the transport parameters for air are taken from [ML97]; detachment and attachment coefficients, respectively from [BN03] and [KKMS92]; and other reaction rates, also from [ML97]. Diffusion coefficients for ions are derived from mobilities using classical Einstein relations

In simulations of positive streamer discharges in air at atmospheric pressure without any preionization, the photoionization term is crucial to produce seed charges in front of the streamer head and then to ensure the streamer propagation [BPL+07]. However, in repetitive discharges, [Pan05] and recently [WPL+10] have shown that even at low frequency, a significant amount of seed charges from previous discharges may be present in the inter-electrode gap. In this work, we have neglected the photoionization source term and considered discharge conditions with a preionization background to ensure a stable propagation of the discharge without impacting the main discharge characteristics [PSS01, Pan05, Cel08, BBC10].

11.4 Construction of the Numerical Strategy

In this section, we introduce a new numerical technique for multi-scale streamer discharge simulations, based on a second order decoupled resolution of the electric field and the drift-diffusion equations for electrons and ions, with self-adaptive decoupling time steps with error control. The drift-diffusion equations are then solved using a dedicated Strang time operator splitting scheme for multi-scale phenomena. On the other hand, the electric field is computed based on a parallel computing method, specially conceived for the configuration under study in 1.5D geometry. Both resolutions are conducted on a dynamic adaptive mesh using spatial multiresolution transformation with error control of the spatial adapted representation.

11.4.1 Second Order Adaptive Time Integration Strategy

Let us write the semi-discretized equations (11.2) and (11.5) in the following way just for analysis purposes:

$$\left.\begin{array}{rcl} \mathrm{d}_t\psi &=& \Psi(\psi,\phi), \\ 0 &=& \Phi(\psi,\phi), \end{array}\right\} \tag{11.6}$$

for $t > t_0$, where $\psi : \mathbb{R} \to \mathbb{R}^{N\times m}$ and $\phi : \mathbb{R} \to \mathbb{R}^{N\times d}$ stand respectively for the spatial discretization of $(n_\mathrm{e}, n_\mathrm{p}, n_\mathrm{n})$, *i.e.*, $m = 3$, and of E over N points. Supposing that all functions are sufficiently differentiable in all their variables and using the Taylor expansion of the true solution, one can write after some time Δt from initial time t_0,

$$\psi(t_0+\Delta t) = \psi_0 + \Delta t\Psi(\psi_0,\phi_0) + \frac{\Delta t^2}{2}\left[\partial_\psi\Psi\,\Psi + \partial_\phi\Psi\,\mathrm{d}_t\phi\right]_{t=t_0} + \mathcal{O}(\Delta t^3), \tag{11.7}$$

with $\psi_0 = \psi(t_0)$, $\phi_0 = \phi(t_0)$.
A second order in time resolution of system (11.6) must then verify (11.7) locally for each Δt. However, as it was stated before, solving simultaneously (11.2) and (11.3) (or (11.5)), or equivalently (11.6), involves important numerical difficulties, considering for instance the different nature of equations (11.2) and (11.3) (or (11.5)). Therefore, a decoupled approach is often used in which one aims at solving the drift-diffusion equations and the electric field independently. This amounts to solve

$$\mathrm{d}_t\tilde{\psi} = \Psi(\tilde{\psi},\phi^\star), \quad t \in]t_0, t_0+\Delta t], \tag{11.8}$$

with fixed $\phi^\star = \phi(t^\star)$, $t^\star \in [t_0, t_0+\Delta t]$ and $\tilde{\psi}(t_0) = \psi_0$.
The most common technique considers $t^\star = t_0$, that is, to previously compute the electric field at t_0 from $\Phi(\psi_0,\phi_0) = 0$, and then solve (11.8) with $\phi^\star = \phi_0$. This can be interpreted as a standard first order operator splitting method that yields an approximation of order 1, $\tilde{\psi}_1(t)$, of the exact solution, $\psi(t)$, based on classical numerical analysis results obtained by confronting (11.7) with

$$\tilde{\psi}_1(t_0+\Delta t) = \psi_0 + \Delta t\Psi(\psi_0,\phi_0) + \frac{\Delta t^2}{2}\left[\partial_\psi\Psi\,\Psi\right]_{t=t_0} + \mathcal{O}(\Delta t^3). \tag{11.9}$$

The same result follows for $\tilde{\phi}_1(t_0+\Delta t)$ computed out of $\Phi(\tilde{\psi}_1(t_0+\Delta t), \tilde{\phi}_1(t_0+\Delta t)) = 0$ or equivalently, out of its explicit representation $\tilde{\phi}_1(t_0+\Delta t) = \Upsilon(\tilde{\psi}_1(t_0+\Delta t))$, assuming a Lipschitz condition:

$$\|\Upsilon(\psi) - \Upsilon(\psi^\star)\| \le L\,\|\psi - \psi^\star\|\,. \tag{11.10}$$

Considering now any $t^\star \in [t_0, t_0+\Delta t]$ into (11.8), the only second order solution, $(\tilde{\psi}_2(t), \tilde{\phi}_2(t))$, will be given by resolution of (11.8) with $\phi^\star = \phi_{\frac{1}{2}}$ for $t^\star = t_0 + \Delta t/2$, for which

$$\tilde{\psi}_2(t_0+\Delta t) = \psi_0 + \Delta t \Psi(\psi_0, \phi_{\frac{1}{2}}) + \frac{\Delta t^2}{2} \left[\partial_\psi \Psi \, \Psi\right]_{\psi\to\psi_0, \phi_{\frac{1}{2}}} + \mathcal{O}(\Delta t^3), \tag{11.11}$$

where

$$\begin{aligned} \Psi(\psi_0, \phi_{\frac{1}{2}}) &= \Psi\left(\psi_0, \phi\left(t_0 + \frac{\Delta t}{2}\right)\right) \\ &= \Psi\left(\psi_0, \phi_0 + \frac{\Delta t}{2} \, \mathrm{d}_t\phi|_{t=t_0} + \mathcal{O}(\Delta t^2)\right) \\ &= \Psi(\psi_0, \phi_0) + \frac{\Delta t}{2} \left[\partial_\phi \Psi \, \mathrm{d}_t\phi\right]_{t=t_0} + \mathcal{O}(\Delta t^2), \end{aligned} \tag{11.12}$$

and hence,

$$\tilde{\psi}_2(t_0+\Delta t) = \psi_0 + \Delta t \Psi(\psi_0, \phi_0) + \frac{\Delta t^2}{2} \left[\partial_\psi \Psi \, \Psi + \partial_\phi \Psi \, \mathrm{d}_t\phi\right]_{t=t_0} + \mathcal{O}(\Delta t^3); \tag{11.13}$$

and

$$\tilde{\phi}_2(t_0+\Delta t) = \Upsilon(\tilde{\psi}_2(t_0+\Delta t)). \tag{11.14}$$

Nevertheless, this second order approximation, $\tilde{\psi}_2(t)$, is based on the previous knowledge of $\phi_{\frac{1}{2}} = \phi(t_0 + \Delta t/2)$, and thus, of $\psi(t_0 + \Delta t/2)$. In order to overcome this difficulty, one can solve (11.8) with $\phi^\star = \tilde{\phi}_1(t_0+\Delta t/2) = \Upsilon(\tilde{\psi}_1(t_0+\Delta t/2))$, that is, computing first $\tilde{\psi}_1(t_0+\Delta t/2)$ with the first order method. In particular, this does not change the previous order estimates as it follows from

$$\begin{aligned} \psi(t_0+\Delta t) - \tilde{\psi}_2(t_0+\Delta t) &= \frac{\Delta t^2}{2} \left[\partial_\phi \Psi \, \mathrm{d}_t(\phi - \tilde{\phi}_1)\right]_{t=t_0} + \mathcal{O}(\Delta t^3) \\ &= \frac{\Delta t^2}{2} \left[\partial_\phi \Psi \, \partial_\psi \Upsilon \, \mathrm{d}_t(\psi - \tilde{\psi}_1)\right]_{t=t_0} + \mathcal{O}(\Delta t^3) \\ &= \mathcal{O}(\Delta t^3). \end{aligned} \tag{11.15}$$

Taking into account both methods,

$$\begin{pmatrix} \tilde{\psi}_1(t_0+\Delta t) \\ \tilde{\phi}_1(t_0+\Delta t) \end{pmatrix} = \mathcal{T}_1^{\Delta t} \begin{pmatrix} \psi_0 \\ \phi_0 \end{pmatrix}, \quad \begin{pmatrix} \tilde{\psi}_2(t_0+\Delta t) \\ \tilde{\phi}_2(t_0+\Delta t) \end{pmatrix} = \mathcal{T}_2^{\Delta t} \begin{pmatrix} \psi_0 \\ \phi_0 \end{pmatrix}, \tag{11.16}$$

we perform computations with a second order scheme $\mathcal{T}_2^{\Delta t}$, which uses an embedded and lower order scheme $\mathcal{T}_1^{\Delta t/2}$, as it was previously detailed. An adaptive time step strategy is then implemented in order to control the accuracy of computations by tuning the duration of the decoupled resolution. It is based on a local error estimate, dynamically computed at the end of each decoupling time step Δt, given by

$$\left\| \mathcal{T}_2^{\Delta t}(\psi_0, \phi_0)^T - \mathcal{T}_1^{\Delta t}(\psi_0, \phi_0)^T \right\| \approx \mathcal{O}(\Delta t^2). \tag{11.17}$$

Therefore, for a given accuracy tolerance $\eta_{\mathcal{T}}$,

$$\left\| \mathcal{T}_2^{\Delta t}(\psi_0, \phi_0)^T - \mathcal{T}_1^{\Delta t}(\psi_0, \phi_0)^T \right\| < \eta_{\mathcal{T}} \tag{11.18}$$

must be verified in order to accept the current computation with Δt, while the new time step is calculated by

$$\Delta t^{\text{new}} = \Delta t \sqrt{\frac{\eta_{\mathcal{T}}}{\left\| \mathcal{T}_2^{\Delta t}(\psi_0, \phi_0)^T - \mathcal{T}_1^{\Delta t}(\psi_0, \phi_0)^T \right\|}}. \tag{11.19}$$

Several dedicated solvers can be then implemented for each subproblem (11.2) and (11.3) while the theoretical error estimates of the decoupling schemes analyzed in this section remain valid. In this way, the independent choice of appropriate numerical schemes allows to strongly reduce the computational complexity of the global numerical strategy, and an error control procedure such as the one proposed in this work allows to effectively calibrate this decoupling within a prescribed accuracy tolerance.

11.4.2 Resolution of the Drift-Diffusion Equations

We consider now the numerical resolution of the drift-diffusion equations (11.2), that can be written in the general form of a convection-reaction-diffusion system of equations:

$$\left.\begin{aligned} \partial_t u - \partial_{\mathrm{x}} \left(F(u) + D(u) \partial_{\mathrm{x}} u \right) = f(u), \quad t > t_0, \\ u(t_0, \mathrm{x}) = u_0(\mathrm{x}), \quad t = t_0, \end{aligned}\right\} \tag{11.20}$$

where F, $f : \mathbb{R}^m \to \mathbb{R}^m$ and $u : \mathbb{R} \times \mathbb{R}^d \to \mathbb{R}^m$, with a tensor of order $d \times d \times m$ as diffusion matrix $D(u)$. In particular, $u = (n_{\mathrm{e}}, n_{\mathrm{p}}, n_{\mathrm{n}})^T$ with $m = 3$ in this study.
The system (11.20) corresponds to problem (11.8) for a fixed electric field, and it is solved during each decoupling time step Δt into $\mathcal{T}_2$ (or $\mathcal{T}_1$) scheme, using a Strang time operator scheme with dedicated high order time integrators on a dynamic adaptive mesh, based on a strategy introduced in [DMD+12]. This resolution is briefly detailed in following sections[1].

Time Operator Splitting

An operator splitting procedure allows to consider dedicated solvers for the reaction part which is decoupled from other physical phenomena like convection, diffusion or both, for which there also exist dedicated numerical methods. These dedicated methods chosen for each subsystem are then responsible for dealing with the fastest scales associated with each one of them, in a separate manner, while the reconstruction of the global solution by the splitting scheme should guarantee an accurate description with error control of the global physical coupling, without being related to the stability constraints of the numerical resolution of each subsystem.
Considering problem (11.20) and in order to remain consistent with the second order $\mathcal{T}_2$ scheme, a second order Strang scheme is implemented [Str68]

$$\mathcal{S}^{\Delta t_{\mathrm{s}}}(u_0) = \mathcal{R}^{\Delta t_{\mathrm{s}}/2} \mathcal{D}^{\Delta t_{\mathrm{s}}/2} \mathcal{C}^{\Delta t_{\mathrm{s}}} \mathcal{D}^{\Delta t_{\mathrm{s}}/2} \mathcal{R}^{\Delta t_{\mathrm{s}}/2}(u_0), \tag{11.21}$$

where operators $\mathcal{R}$, $\mathcal{D}$, $\mathcal{C}$ indicate respectively the independent resolution of the reaction, diffusion and convection problems with a splitting time step, Δt_{s}, taken inside the overall decoupling time step, $\Delta t_{\mathrm{s}} \leq \Delta t$. Usually, for propagating reaction waves where for instance, the speed of propagation is much slower than some of the chemical scales, the fastest scales are not directly related to the global physics of the phenomenon, and thus, larger splitting

[1] For the sake of brevity, we will omit in what follows the description of the mesh refinement technique by multiresolution analysis on variables $u = (n_{\mathrm{e}}, n_{\mathrm{p}}, n_{\mathrm{n}})^T$, included in [DBM+12]. The corresponding description can be found in chapters 3 and 4. The threshold parameter ε in (4.26) will be noted as η_{MR} in the following. Nevertheless, we reproduce the splitting technique described in Chapter 10 because it is particularly embedded in the time adaptive strategy of Section 11.4.1.

time steps might be considered [DMD+12, DDD+12]. Nevertheless, order reductions may then appear due to short-life transients associated with fast variables and in these cases, it has been proven in [DM04] that better performances are expected while ending the splitting scheme by operator $\mathcal{R}$ or in a more general case, the part involving the fastest time scales of the phenomenon.
The resolution of (11.20) should be precise enough to guarantee theoretical estimates given in Section 11.4.1. Therefore, an adaptive splitting time step strategy, based on a local error estimate at the end of each splitting time step Δt_s, is also implemented in order to control the accuracy of computations [DMDD11]. In this context, a second, embedded and lower order Strang splitting method $\widetilde{\mathcal{S}}^{\Delta t_s}$ was developed by [DDD+11], which allows to dynamically calculate a local error estimate that should verify

$$\left\| \mathcal{S}^{\Delta t_s}(u_0) - \widetilde{\mathcal{S}}^{\Delta t_s}(u_0) \right\| \approx \mathcal{O}({\Delta t_s}^2) < \eta_{\text{split}}, \tag{11.22}$$

in order to accept the current computation with Δt_s, and thus, the new splitting time step is given by

$$\Delta t_s^{\text{new}} = \min \left(\Delta t_s \sqrt{\frac{\eta_{\text{split}}}{\left\| \mathcal{S}^{\Delta t_s}(u_0) - \widetilde{\mathcal{S}}^{\Delta t_s}(u_0) \right\|}},\ t_0 + \Delta t - \hat{t} \right), \tag{11.23}$$

with $\eta_{\text{split}} \leq \eta_{\mathcal{T}}$ and $\hat{t} = \sum_i \Delta t_{s_i}$ while $\hat{t} \in]t_0, t_0 + \Delta t]$.
The choice of suitable time integration methods to numerically approximate $\mathcal{R}$, $\mathcal{D}$ and $\mathcal{C}$ during each Δt_s is mandatory not only to guarantee the theoretical framework of the numerical analysis but also to take advantage of the particular features of each independent subproblem. A new operator splitting for reaction-diffusion systems was recently introduced [DMD+12, DDD+12], which considers a high fifth order, A-stable, L-stable method like Radau5 [HW96], based on implicit Runge-Kutta schemes for stiff ODEs, that solves with a local cell by cell approach the reaction term: a system of stiff ODEs without spatial coupling in a splitting context. For the diffusion problem, another high fourth order method like ROCK4 [Abd02] is considered, which is based on explicit stabilized Runge-Kutta schemes that feature extended stability domains along the negative real axis. The ROCK4 solver is then very appropriate for diffusion problems because of the usual predominance of negative real eigenvalues. Both methods incorporate adaptive time integration tools, similar to (11.19) and (11.23), in order to control the accuracy of the integrations for given accuracy tolerances η_{Radau5} and η_{ROCK4}, chosen such that $\eta_{\text{Radau5}} < \eta_{\text{split}}$ and $\eta_{\text{ROCK4}} < \eta_{\text{split}}$. In particular, in the case of multi-scale propagating waves, it can be proven that the local treatment plus the adaptive time stepping of the reaction solver allow to discriminate the cells of high reactive activity only present in the neighborhood of the localized wavefront, saving as a consequence a large quantity of integration time [DDD+12].
An explicit high order in time and in space one step monotonicity preserving scheme OSMP [DT04] is used as convective scheme. It combines monotonicity preserving constraints for non-monotone data to avoid extrema clipping, with TVD features to prevent spurious oscillations around discontinuities or sharp spatial gradients. Classical CFL stability restrictions are though imposed inside each splitting time step Δt_s for operator $\mathcal{C}^{\Delta t_s}$. The overall combination of an explicit treatment of the spatial phenomena as convection and diffusion, with a local implicit integration of stiff reaction implies important savings in computing time and memory resources [DMD+12], as well as an important reduction of computational complexity with respect to a fully implicit coupled resolution of problem (11.20). On the other hand an explicit coupled treatment of (11.20) will have a very limited efficiency for stiff problems unless more sophisticated strategies as the asynchronous local time-stepping techniques

[KDOO05, OK06, UBRT07] are considered even though these schemes do not provide a precise measurement of the accuracy of the integration.
Finally, the numerical errors of the splitting scheme are effectively handled by an error control procedure which furthermore allows to determine the coupling time scales of the global phenomenon that can be several orders of magnitude slower than the fastest time scales of each subproblem treated by each dedicated solver. In this way a decoupling of the time scale spectrum of the problem is achieved that leads to more efficient performances within a prescribed accuracy tolerance whenever this decomposition of scales is possible.

11.4.3 Computation of the Electric Field

In this part, we are concerned with the resolution of the electric field according to the $\mathcal{T}_2$ (or $\mathcal{T}_1$) scheme at some fixed time for a given distribution of charges (n_e, n_p, n_n), considering a 1.5D model. This computation is also performed on the adapted mesh obtained by the previous multiresolution analysis.

Discretization of the Computational Domain

According to Figure 11.3, the computational domain is limited by a planar cathode at $x = 0$ and the tip of a hyperbolic anode at $x = L_x$. The anode is not included in the domain. We consider streamers of fixed radius R_s along the axis of symmetry. The computational domain is divided into n_x cells of different size corresponding to the multiresolution adapted mesh, with faces x_f^i, where $i \in [0, n_x]$ and cell centers x_c^j, where $j \in [1, n_x]$. The face x_f^0 corresponds to the position of the cathode and $x_f^{n_x}$ corresponds to the position of the tip of the anode. Therefore for each cell x_c^i, there is its left face x_f^{i-1}, and its right face x_f^i. For each cell x_c^j we define a width $w_j = x_f^j - x_f^{j-1}$ (see Figure 11.4).

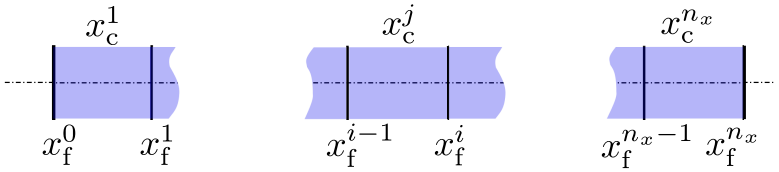

Fig. 11.4: *Definition of the grid: the cell centers are located at x_c^j, whereas cell faces are located at x_f^i. The domain is bounded by faces x_f^0 (cathode) and $x_f^{n_x}$ (tip of the anode).*

Resolution of the Electric Field in a 1.5D Model

To determine the electric field during the propagation of the streamer, the space charge of the streamer is considered as a set of finite cylinders of width w_j, bounded by cell faces x_f^{j-1} and x_f^j. As the computational domain is bounded by conducting electrodes of fixed potential, each volume charge ρ_j creates an infinite series of image charges [DJE64, DDE71]. Then the principle of superposition is used to sum individual contributions from all the cylindrical space charges in the domain, their image charges, and the Laplacian electric field (computed based on classical results [EMM28]). An advantage of this approach dwells in the fact that the electric field contributions from individual cylinders can be expressed analytically in a simple form and the determination of the electric field in each point of the domain can be performed in parallel. In the configurations we have studied, the cathode is grounded whereas an electric voltage is applied on the anode. These boundary conditions are taken into account by the Laplacian

electric field and by including a series of image charges of the charges in the gap. It is important to note that the computation of the Laplacian electric field takes into account the real geometry of electrodes as shown in Figure 11.3. However, in this work, to simplify the computation of image charges we have assumed that both electrodes are planar. For a volume charge ρ_j centered at x_{c}^j, there exist image charges of the first order with charge $-\rho_j$ at $x = 2L_x - x_{\mathrm{c}}^j$ mirrored through the anode, see Figure 11.5a, and at $x = -x_{\mathrm{c}}^j$ mirrored through the cathode, see Figure 11.5b. And for each of these image charges there exist higher order image charges of opposite signs and so forth. All the image charges of ρ_j up to order three are depicted in Figure 11.5c.

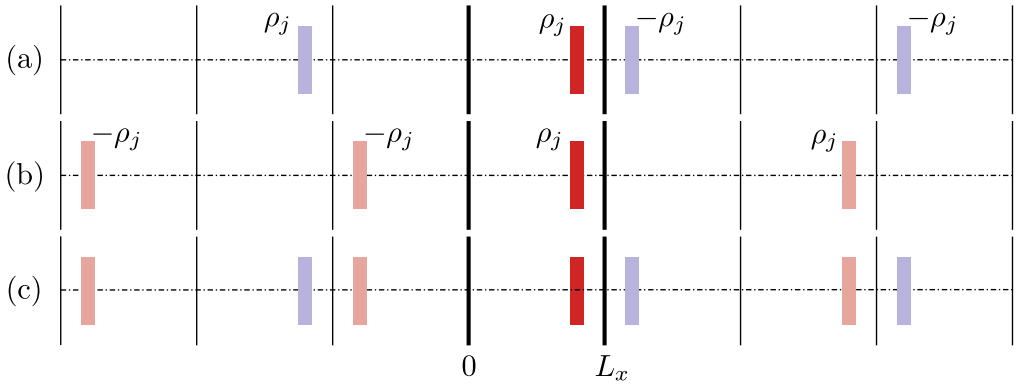

Fig. 11.5: *Image charges up to the third order: (a) charge ρ_j is first mirrored behind the anode ($x = L_x$), (b) charge ρ_j is first mirrored behind the cathode ($x = 0$), (c) charge ρ_j and its images.*

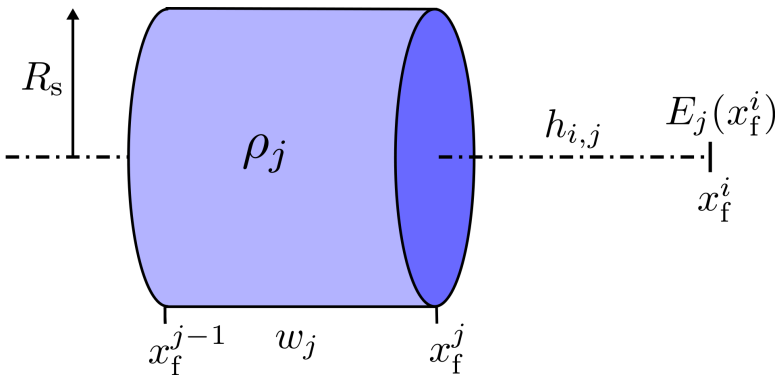

Fig. 11.6: *Charged cylinder considered to compute the electric field in the 1.5D model.*

Integrating the generalized Coulomb's law [Jac99] and using the principle of superposition, we find that the cylinder charges of cells $j \in [1, n_x]$ of width w_j, radius R_{s}, charged with densities ρ_j (see Figure 11.6), and the Laplacian electric field $E_L(x_{\mathrm{f}}^i)$ at x_{f}^i [EMM28], create the electric field E at position x_{f}^i as follows:

$$E(x_{\mathrm{f}}^i) = E_L(x_{\mathrm{f}}^i) + \sum_{j=1}^{n_x} s \frac{\rho_j w_j}{2\varepsilon_0} \left(1 - \frac{w_j + 2h_{i,j}}{\sqrt{h_{i,j}^2 + R_{\mathrm{s}}^2} + \sqrt{(h_{i,j} + w_j)^2 + R_{\mathrm{s}}^2}} \right), \tag{11.24}$$

where

$$h_{i,j} = \begin{cases} x_{\mathrm{f}}^{i} - x_{\mathrm{f}}^{j} & \text{for } i \geq j, \\ x_{\mathrm{f}}^{j-1} - x_{\mathrm{f}}^{i} & \text{for } i < j, \end{cases} \quad \text{and} \quad s = \begin{cases} +1 & \text{for } i \geq j, \\ -1 & \text{for } i < j. \end{cases} \tag{11.25}$$

The positive sign of s accounts for the electric field calculated on the right from the position of the charged cylinder and vice-versa. The same formula applies for the image charges, but an appropriate sign of the charge has to be carefully taken into account according to Figure 11.5. In particular, in a shared memory computing environment, a straightforward parallelization is accomplished for equation (11.24), in which each core solves successively the electric field on one single position x_{f}^{i}, and where neither synchronization stages nor data exchange are needed among nodes.

Note that for $R_{\mathrm{s}} \to \infty$ (infinite plane charges), equation (11.24) yields the exact electric field for a planar front:

$$E_{\mathrm{inf}} = \sum_{j=1}^{n_x} s \frac{\rho_j w_j}{2\varepsilon_0}. \tag{11.26}$$

For finite radius R_{s} the solution (11.24) is valid only on the axis of the discharge, but when applied to a discharge of a small radius, the electric field will vary only negligibly over the cross section of the discharge. This approach is expected to be more accurate for any finite radius than any discretization of Poisson's equation [DJE64].

11.5 Numerical Results

In this section, we present some numerical illustrations of the proposed numerical strategy for the simulations of positive streamers using a 1.5D model in a point-to-plane geometry. First, we will consider a discharge propagation with constant applied voltage for which different features of the numerical strategy are discussed, *e.g.*, error estimates, data compression values and computing time, in order to properly choose the simulation parameters. Then, the potential of the method is fully exploited for a more complex configuration of repetitive discharges generated by high frequency pulsed applied voltages, followed by a long time scale relaxation, for which a complete physical description of the discharge and the post-discharge phases is achieved.

11.5.1 Propagation of a Positive Streamer with Constant Applied Voltage

We consider a point-to-plane geometry with a 1 cm gap between the tip of the electrode and the plane, and a constant applied voltage of 13 kV at $x = L_x$. For the following simulations, the discharge is initiated by placing a neutral plasma cloud with a Gaussian distribution close to the tip of the anode. The initial distributions of electrons and ions are then given by

$$n_{\mathrm{e,p}}(x)|_{t=0} = n_{\max} \exp\left(-(x-c)^2/w^2\right) + n_0, \quad n_{\mathrm{n}}(x)|_{t=0} = 0, \tag{11.27}$$

where $w = 0.027\,\mathrm{cm}$, $c = 1\,\mathrm{cm}$, $n_{\max} = 10^{14}\,\mathrm{cm}^{-3}$, and with a preionization of $n_0 = 10^{8}\,\mathrm{cm}^{-3}$. There are no negative ions as initial condition. The streamer radius is set to $R_{\mathrm{s}} = 0.05\,\mathrm{cm}$ to have a typical electric field magnitude in the streamer head of 120 kV/cm [Kul98]. Homogeneous Neumann boundary conditions were considered for the drift-diffusion equations.

Two instances of the discharge propagation are shown in Figure 11.7, for 12 nested grids equivalent to 4096 cells on the finest grid, $L = 12$, and for accuracy tolerances of $\eta_{\mathcal{T}} = \eta_{\mathrm{split}} = \eta_{\mathrm{MR}} = 10^{-4}$; the spatial refinement takes place only where it is required. Fine tolerances

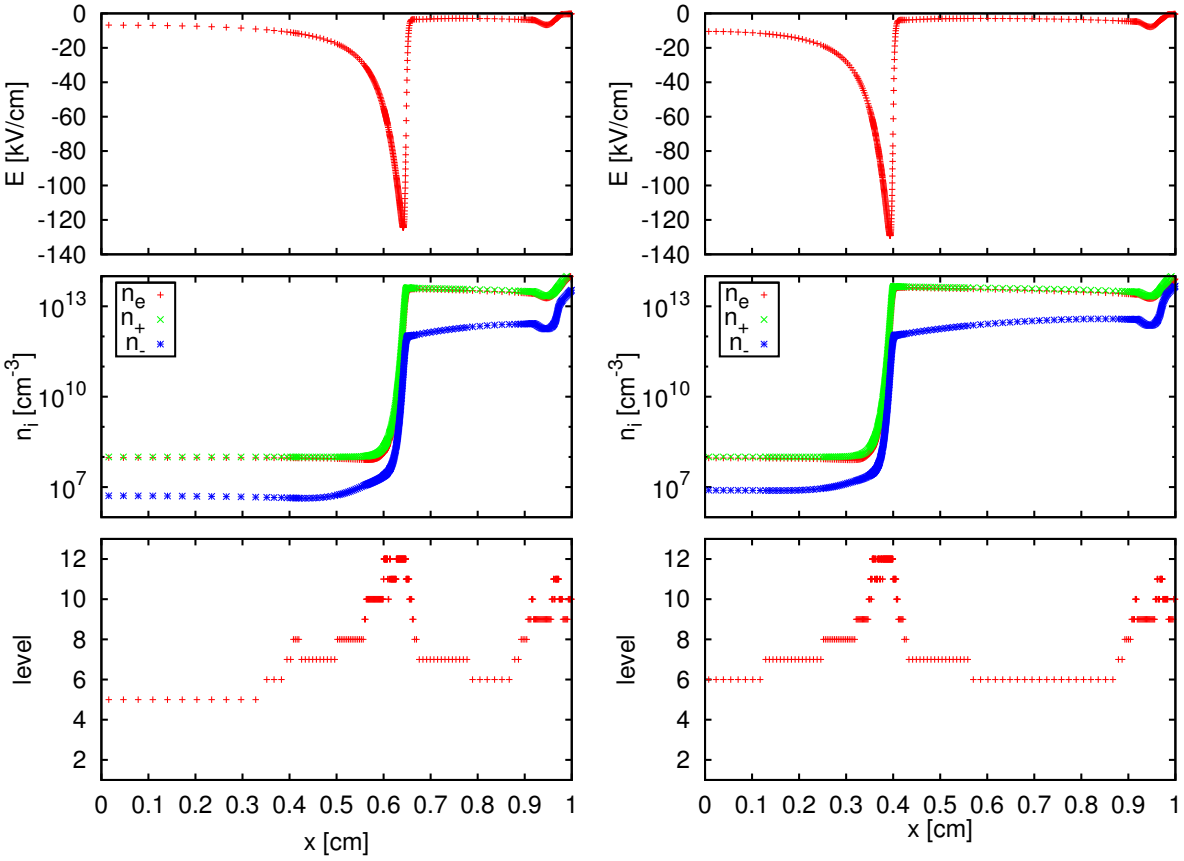

Fig. 11.7: *Positive streamer propagation at $t = 6\,ns$ (left) and $t = 10\,ns$ (right). Top: electric field; middle: charged species density; and bottom: grid levels. Finest grid: 4096, $\eta_{\mathcal{T}} = \eta_{\text{split}} = \eta_{\text{MR}} = 10^{-4}$.*

were chosen in all cases for the solvers, $\eta_{\text{Radau5}} = \eta_{\text{ROCK4}} = 10^{-7}$, to guarantee accurate integrations. For all the simulation cases, the *detail* in each cell is taken as the maximum of the *details* computed according to (3.72) for each variable, where the prediction operator is a polynomial interpolation of degree 2, performed on normalized $\log \mathrm{u}$ of the density variables in order to properly discriminate the streamer heads from the highly ionized plasma channel; this logarithmic scale guarantees a correct spatial representation of the phenomenon as seen in Figure 11.7 for the density profiles.

In order to perform an analysis of the numerical results, we define the *reference* solution as a fine resolution with the $\mathcal{T}_2$ scheme that considers a fixed decoupling time step, $\Delta t = 10^{-14}\,\mathrm{s}$ and a uniform grid of 4096 cells. For this *reference* solution, the memory requirements are acceptable and the simulation is still feasible, but it requires about 14 days of real simulation time on an AMD Opteron 6136 Processor cluster, while running the electric field computation in parallel on 16 CPU cores. In this case, the computation of the electric field, based on a direct integration of individual contributions of the charged cylinders, represents 80% of total CPU time per time step (about 3.2 s).

First of all, we must verify the previous order estimates for the $\mathcal{T}_1$ and $\mathcal{T}_2$ schemes given in Section 11.4.1. We consider as initial condition the *reference* solution at $t = 10\,\mathrm{ns}$. In order to only evaluate errors coming from the decoupling techniques, $\mathcal{T}_1$ and $\mathcal{T}_2$, we consider a fine splitting time step, $\Delta t_{\mathrm{s}} = 10^{-14}\,\mathrm{s}$, to solve the drift-diffusion problem (11.2) and a

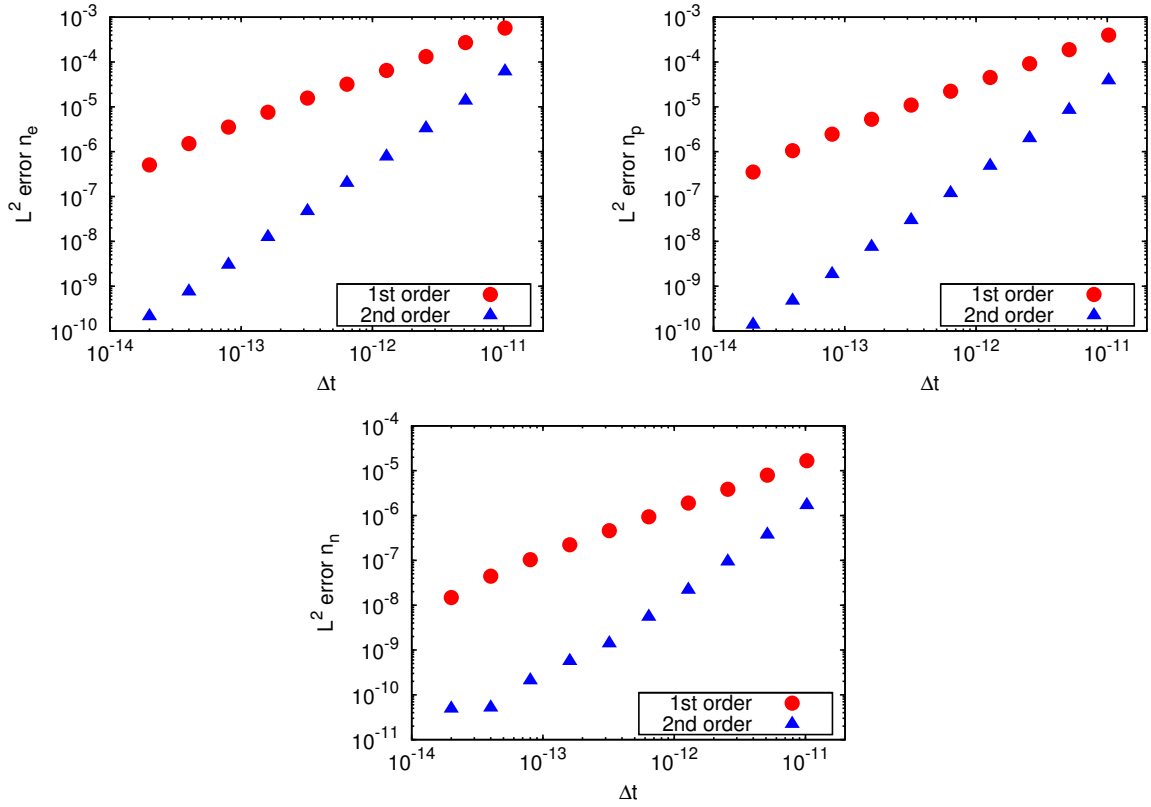

Fig. 11.8: *Normalized L^2 errors between the reference and the $\mathcal{T}_1$ (first order) and $\mathcal{T}_2$ (second order) solutions for several decoupling time steps Δt on a uniform grid of 4096 cells. Top: electron (left) and positive ions (right); and bottom: negative ions.*

uniform grid; then, we solve (11.6) with both schemes for several decoupling time steps Δt_i, and calculate the normalized L^2 error between the first/second order and *reference* solutions after time $t = 2^{10}\Delta t_\mathrm{s} = 1.024 \times 10^{-11}$ s. Figure 11.8 shows results with $\Delta t_i = 2^i \Delta t_\mathrm{s}$, where $i \in [1, 10]$, which clearly verify first and second order in time for the $\mathcal{T}_1$ and $\mathcal{T}_2$ schemes, respectively, and prove important gains in accuracy for same time steps. For instance, for $\Delta t \leq 10^{-12}$ s the second order scheme provides solutions with L^2 errors at least 100 times lower than those obtained with the first order method.

Figure 11.9 shows the time evolution of the normalized L^2 error for each variable between the time-space adapted and *reference* solutions for several tolerances, $\eta_\mathcal{T} = \eta_\mathrm{split} = \eta_\mathrm{MR} = 10^{-4}$, 10^{-3}, and 10^{-2}. These are rather approximations of the error since the *reference* and adapted solutions are not evaluated exactly at the same time, and therefore, they are often slightly shifted of about $\sim 10^{-14} - 10^{-13}$s. In these tests, the decoupling time steps Δt were limited by the dielectric relaxation time step, Δt_{DR}, after noticing an important amount of rejections of computed time steps according to (11.19), whenever $\Delta t \gtrsim 1.5 \times \Delta t_{DR}$. Otherwise, Δt is dynamically chosen in order to locally satisfy the required accuracy $\eta_\mathcal{T}$, but it does not show important variations considering the self-similar propagating phenomenon.

In Figure 11.10, we can see the corresponding adapted grid to each previous configuration with different tolerances. The representation of the electric field and the densities shows that for $\eta_\mathcal{T} = \eta_\mathrm{split} = \eta_\mathrm{MR} = 10^{-2}$, the streamer front propagates faster than in the *reference* case, with a slightly higher peak of the electric field in the front. On the other hand, for

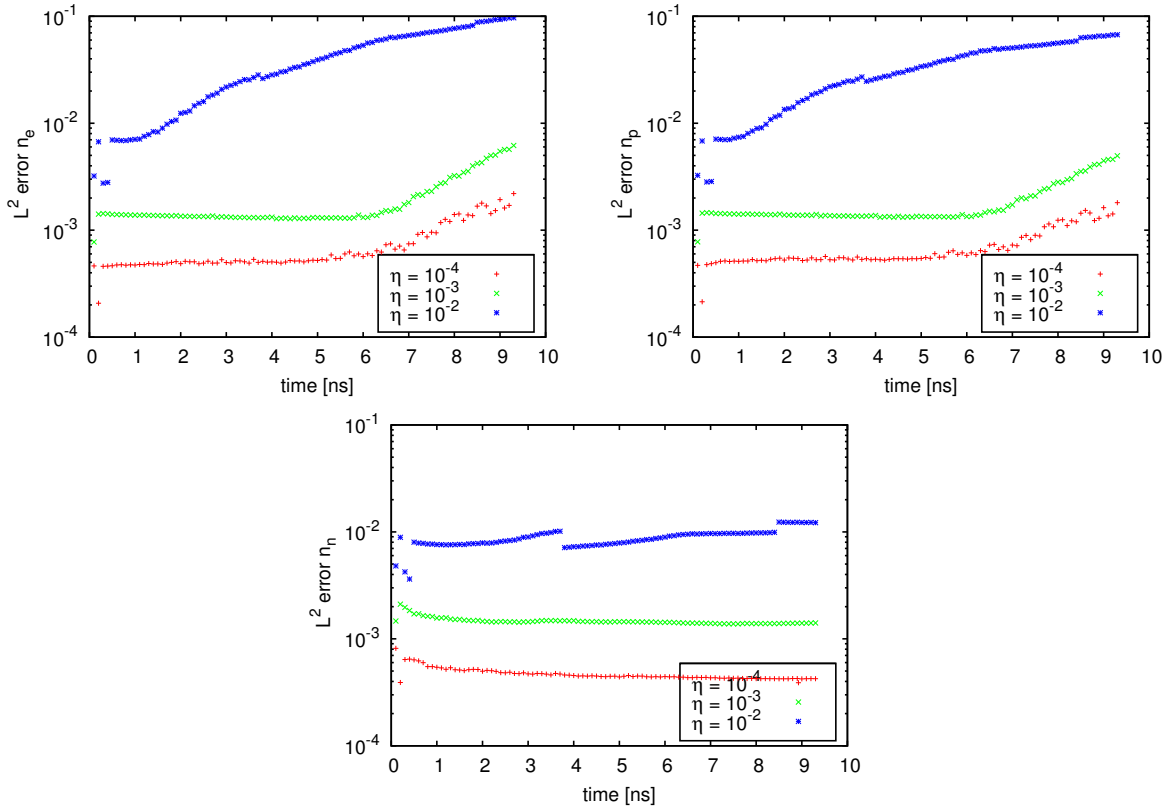

Fig. 11.9: *Time evolution of the normalized L^2 errors between the reference and adapted solutions with $\eta = \eta_{\mathcal{T}} = \eta_{\text{split}} = \eta_{\text{MR}} = 10^{-4}$, 10^{-3}, and 10^{-2}, and 4096 cells corresponding to the finest discretization. Top: electron (left) and positive ions (right); and bottom: negative ions.*

$\eta_{\mathcal{T}} = \eta_{\text{split}} = \eta_{\text{MR}} \leq 10^{-3}$, we observe a quite good agreement between the adapted and *reference* resolutions.

We consider now an accurate enough resolution with $\eta_{\mathcal{T}} = \eta_{\text{split}} = \eta_{\text{MR}} = 10^{-4}$ and investigate the influence of the number of grids, that is, the finest spatial discretization at level L that should be taken into account. Figure 11.11 shows the adapted grids for $L = 10$, 11 and 12, respectively equivalent to 1024, 2048 and 4096 cells in the finest grid; and a close-up of the corresponding electric fields in the discharge head at $t = 8$ ns. We see that for this level of tolerances, the streamer front propagates slightly slower than the *reference* case for $L = 10$, whereas $L = 11$ gives already good resolutions compared with the *reference* solution and with $L = 12$. In particular, higher values of L would need lower tolerances in order to retain regions at the finest level; this is already the case for $L = 13$ (equivalent to 8192 cells). Therefore, $L = 11$ with 2048 cells at the finest level seems to be an appropriate choice for this level of accuracy.

Table 11.1 summarizes the number of cells in the adapted grid ($\#AG$) at time $t = 8$ns, and the corresponding data compression (DC) defined as the percentage of active cells with respect to the equivalent number of cells for the finest discretization, in this case 2048 for $L = 11$. For this propagating case, the data compression remains of the same order during the time simulation interval. The CPU computing times correspond to a time domain of study of $t \in [0, 10]$ ns computed by one sole CPU core. If we consider for example total computing time for $L = 11$ and tolerances $\eta_{\mathcal{T}} = \eta_{\text{split}} = \eta_{\text{MR}} = 10^{-4}$, it is ~ 44 times less expensive with respect to a

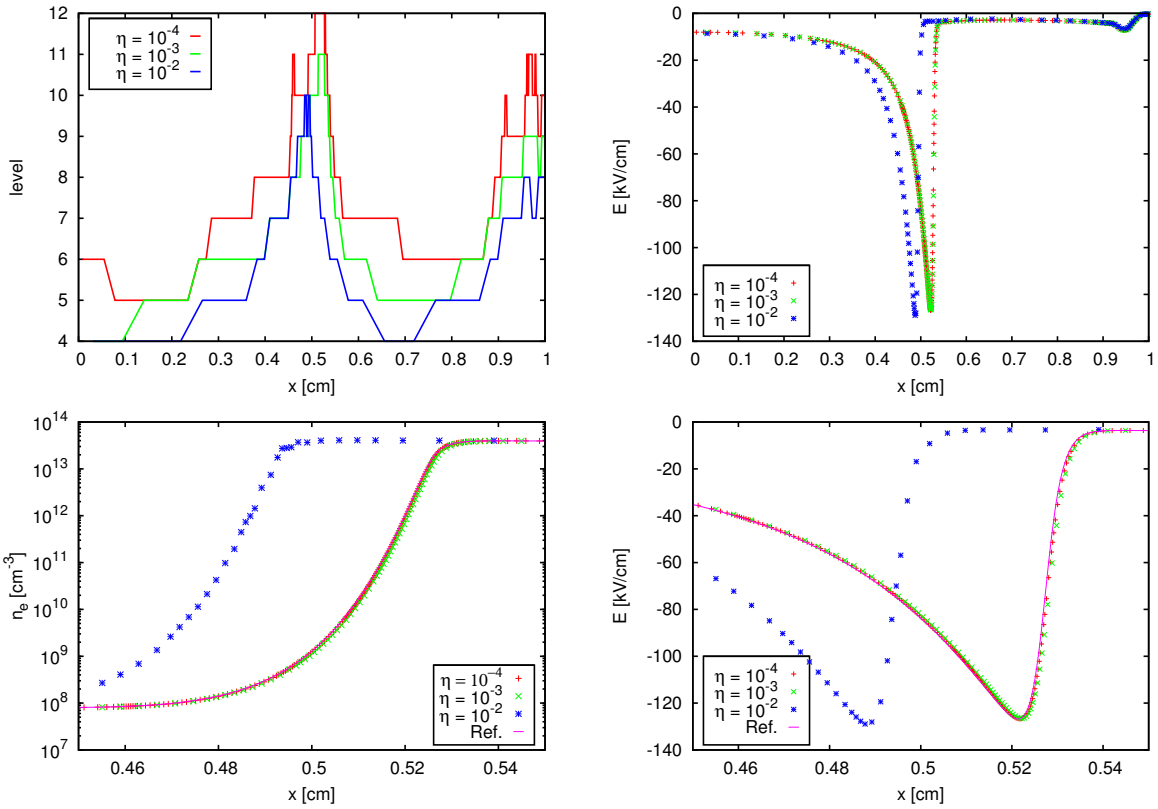

Fig. 11.10: *Top: adapted grids (left) and electric fields (right) at $t = 8$ ns with 4096 cells corresponding to the finest discretization, and $\eta = \eta_{\mathcal{T}} = \eta_{\text{split}} = \eta_{\text{MR}} = 10^{-4}$, 10^{-3}, and 10^{-2}. Bottom: zoom on the electron distributions (left) and the electric field (right) with the same parameters, and the reference solution.*

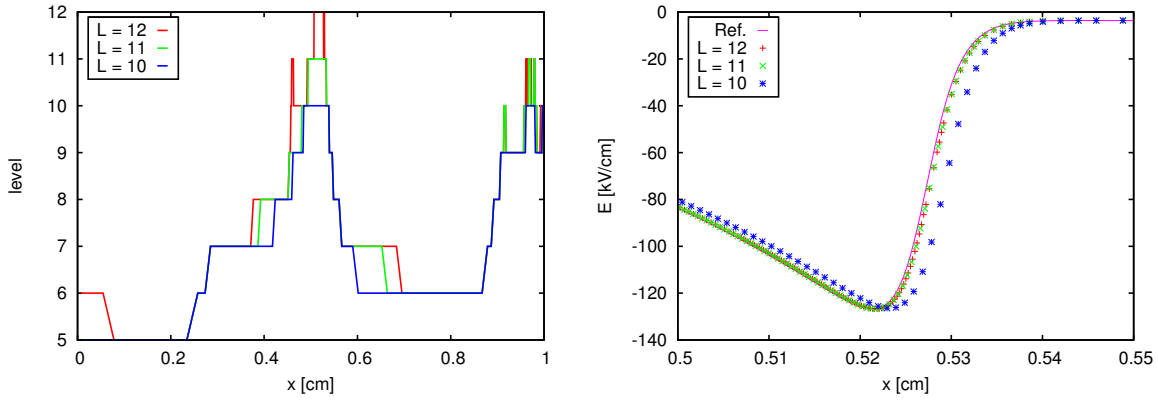

Fig. 11.11: *Adapted grids (left) and electric fields (right) at $t = 8$ ns, for several finest spatial discretization $L = 10$, 11 and 12, $\eta_{\mathcal{T}} = \eta_{\text{split}} = \eta_{\text{MR}} = 10^{-4}$, and the reference solution.*

resolution on a uniform grid with 2048 cells and $\eta_{\mathcal{T}} = \eta_{\text{split}} = 10^{-4}$ (CPU time of 8552 s). This is quite reasonable, taking into account that the computing time for the electric field resolution is proportional to at least $\mathcal{O}(N^2)$ for N computing cells, after (11.24).

In conclusion, in this section we have shown that the numerical strategy developed can be

Table 11.1: *Number of cells in the adapted grid (#AG) and data compression (DC) at time $t = 8ns$, CPU computing time for $t \in [0, 10]$ ns, $L = 11$, and several tolerances $\eta = \eta_{\mathcal{T}} = \eta_{\text{split}} = \eta_{\text{MR}}$.*

η	$\#AG$	$DC\%$	CPU(s)
10^{-6}	724	35.35	1360
10^{-5}	421	20.56	517
10^{-4}	263	12.84	193
10^{-3}	138	6.74	66
10^{-2}	70	3.42	24

efficiently applied to simulate the propagation of highly nonlinear ionizing waves as streamer discharges. An important reduction of computing time results from significant data compression with still accurate resolutions. In addition, this study allows to properly tune the various simulation parameters in order to guarantee a fine resolution of more complex configurations, based on the time-space accuracy control capabilities of the method.

11.5.2 Simulation of Multi-Pulsed Discharges

In this section, we analyze the performance of the proposed numerical strategy on the simulation of nanosecond repetitively pulsed discharges [PGL+06, PLL10]. The applied voltage profile for this type of discharges is a high voltage *pulse* followed by a zero voltage *relaxation* phase. The typical pulse duration is $\sim 10^{-8}$ s, while the relaxation phase takes over $\sim 10^{-4}$ s. The detailed experimental study of these discharges in air has shown that the cumulative effect of repeated pulsing achieves a steady-state behavior [PLL10]. In the following illustrations, we choose a pulse duration of $T_{\text{p}} = 15$ ns, which is approximately equal to the time that is needed for the discharge to cross the inter-electrode gap. The rise time considers the time needed to go from zero to the maximum voltage and it is set to $T_{\text{r}} = 2$ ns. The pulse repetition period is set to $T_{\text{P}} = 10^{-4}$ s, equal to 10 kHz of repetition frequency, a typical value used in experiments [PGL+06]. We model the voltage pulse P by using sigmoid functions

$$P(t, s, r, p) = 1 - \sigma(-t, -s, r) - \sigma(t, s + p, r), \tag{11.28}$$

with

$$\sigma(t, s, r) = \frac{1}{1 + \exp(-8(t - s)/r)}, \tag{11.29}$$

for time t, where s indicates when the pulse starts; r is the rise time; and p is the pulse duration; t, s, r, $p \in [0, T_{\text{P}}]$. With a maximum applied voltage $V_{\max}$, the applied voltage $V(t)$ is computed by

$$V(t) = V_{\max} \cdot P\left(t - \left\lfloor \frac{t}{T_{\text{P}}} \right\rfloor \cdot T_{\text{P}}, T_{\text{r}}, T_{\text{r}}, T_{\text{p}}\right). \tag{11.30}$$

In repetitively pulse discharges at atmospheric pressure and 300 K, as discussed in [Pan05, WPL+10], electrons attach rapidly to O_2 molecules during the interpulse to form negative ions (characteristic time scale of 20 ns). Then, the rate of the plasma decay is determined by ion-ion recombination [KKMS92, Pan05, WPL+10]. When the next voltage pulse is applied, electrons are detached with a rate taken from [BN03]. Therefore, as initial condition we assume a distribution similar to the end of the interpulse phase with a homogeneous preionization

consisting of positive and negative ions with a density of $10^9\,\mathrm{cm}^{-3}$. For electrons, we consider a low homogeneous background of $10^1\,\mathrm{cm}^{-3}$. This small amount of electrons as initial condition has a negligible influence on the results.

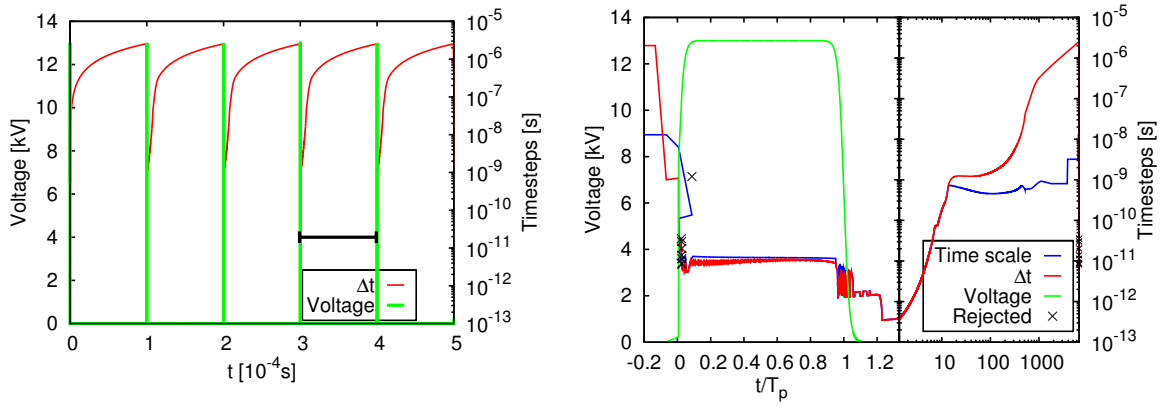

Fig. 11.12: *Time evolution of the applied voltage and the decoupling time steps Δt for a multi-pulse simulation for the first 6 pulses (left) and for the 4th one (right) with its subsequent relaxation. Rejected time steps are marked with black crosses, while the minimum time scale corresponds to the blue line.*

We set the tolerances to $\eta_{\mathcal{T}} = \eta_{\text{split}} = \eta_{\text{MR}} = 10^{-4}$ and consider $L = 11$ grid levels, equivalent to 2048 cells in the finest grid. As in the previous configuration, homogeneous Neumann boundary conditions were considered for the drift-diffusion equations. Figure 11.12 shows the time evolution of the decoupling time steps and the applied voltage for the first six pulses, even though simulation was performed for 100 pulses, that is $t \in [0, 10^{-2}]\,\mathrm{s}$. This simulation took over 8h44m while running the electric field computation in parallel on 6 CPU cores of the same AMD Opteron 6136 Processor cluster; this gives an average of 5.24 minutes per pulse period. Figure 11.12 shows also the fourth pulse for which the steady-state of the periodic phenomenon was already reached and almost the same numerical performance is reproduced during the rest of computations. The time steps are about $\sim 10^{-11}\,\mathrm{s}$ during pulses, then increase from $\sim 10^{-12}\,\mathrm{s}$ up to about $\sim 10^{-6}\,\mathrm{s}$ during a period ~ 6000 times longer, for which standard stability constraints are widely overcome according to the required accuracy tolerance. Solving this problem for such different scales with a constant time step is out of question and even a standard strategy that considers the minimum of all time scales would limit considerably the efficiency of the method as it is shown in the representation. In this particular case, the dielectric relaxation is the governing time scale during the discharge as in the previous case with constant applied voltage, whereas the post-discharge phase is alternatively ruled by diffusive or convective CFL, or by ionization time scale, with all security factors and CFL conditions set to one in Figure 11.12.

The computation is initialized with a time step included in the pulse duration. Nevertheless, after each relaxation phase, since the new time step is computed based on the previous one according to (11.19), this new time step will surely skip the next pulse. In order to avoid this, each time we get into a new period, that is when $\lfloor t/T_{\text{P}} \rfloor$ changes, we initialize the time step with $\Delta t = 0.5 T_{\text{r}} = 1\,\mathrm{ns}$. This time step is obviously rejected as seen in Figure 11.12, as well as the next ones, until we are able to retrieve the right dynamics of the phenomenon for the required accuracy tolerance. No other intervention is needed neither for modeling parameters nor for numerical solvers in order to automatically adapt the time step needed to describe the various time scales of the phenomenon within a prescribed accuracy.

Figure 11.13 represents the time evolution of the data compression which ranges from $\sim 2\%$ up

to $\sim 16\%$ during each pulse period. Regarding only the electric field resolution with the same time integration strategy, a grid adaptation technique involves resolutions ~ 39 to ~ 2500 times faster, based on a really rough estimate for $\mathcal{O}(N^2)$ operations.

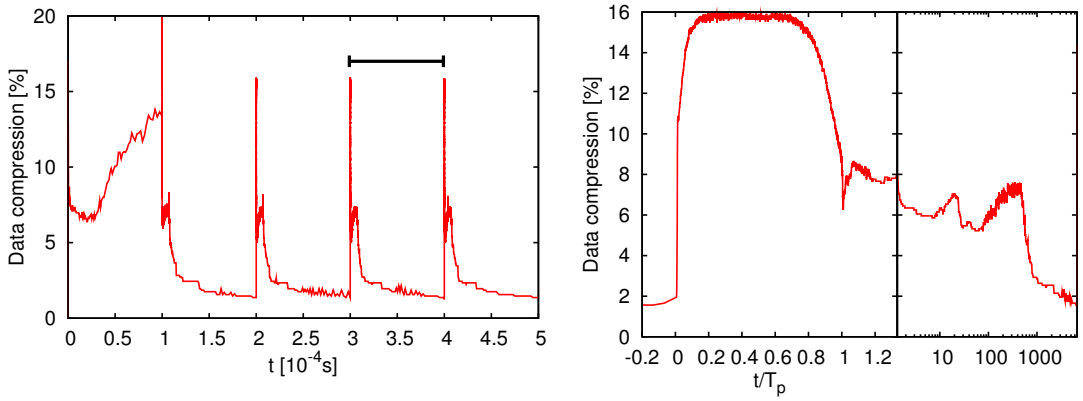

Fig. 11.13: *Time evolution of the data compression for a multi-pulse simulation for the first six pulses (left) and for the fourth one (right) with its subsequent relaxation.*

Figure 11.14 presents the discharge dynamics for the first period. First, we observe at $t = 10\,\text{ns}$ after the beginning of the pulse, the propagation of a positive streamer in the gap. In Section 11.5.1, a preionization of positive ions and electrons was used to ensure the positive streamer propagation. In this section, seed electrons ahead of the streamer front are created as the front propagates by detachment of negative ions initially present. We note that at 15 ns, which corresponds to almost the end of the plateau before the decrease of the applied voltage, the discharge has crossed $\sim 0.75\,\text{cm}$ of the 1 cm gap. As a consequence, during the voltage decrease and at the beginning of the relaxation phase where the applied voltage is zero, there is a remaining space charge and steep gradients of charged species densities in the gap. Then for $t = 50\,\text{ns}$, Figure 11.14 shows that the electric field in the discharge is almost equal to zero except in a small area where steep gradients of the electric field are observed but with peak values of only $30\,\text{V/cm}$. We have checked that this area corresponds to the location of the streamer head at the end of its propagation as it is seen in the representation. We note that in the post-discharge, electrons are attaching and then at $t = 50\,\text{ns}$, the density of positive ions is almost equal to the density of negative ions in the whole gap. At $t = 99972\,\text{ns}$, the densities of charged species have significantly decreased due to charged species recombination. However, it is interesting to note that the location of the previous streamer head can still be observed at the same location as at $t = 50\,\text{ns}$, but with much smaller gradients of charged species densities and a very small electric field. This final state is the initial condition of the second pulse with a non-uniform axial preionization with positive and negative ions and a much smaller density of electrons.

After a few repetitive pulses, we have observed that the discharge dynamics reached a steady-state behavior as observed in the experiments. To show the characteristics of the discharge when the steady-state is reached, Figure 11.15 shows the discharge dynamics of the 100th period. The sequence of images is the same as in Figure 11.14. At the end of the 99th pulse, we have observed that the axial distribution of charged species in the gap is uniform and that the level of preionization is $5 \times 10^{10}\,\text{cm}^{-3}$ positive and negative ions and $10^4\,\text{cm}^{-3}$ electrons. We note that 10 ns after the beginning of the 100th pulse the propagation of the discharge is faster than for the first pulse.

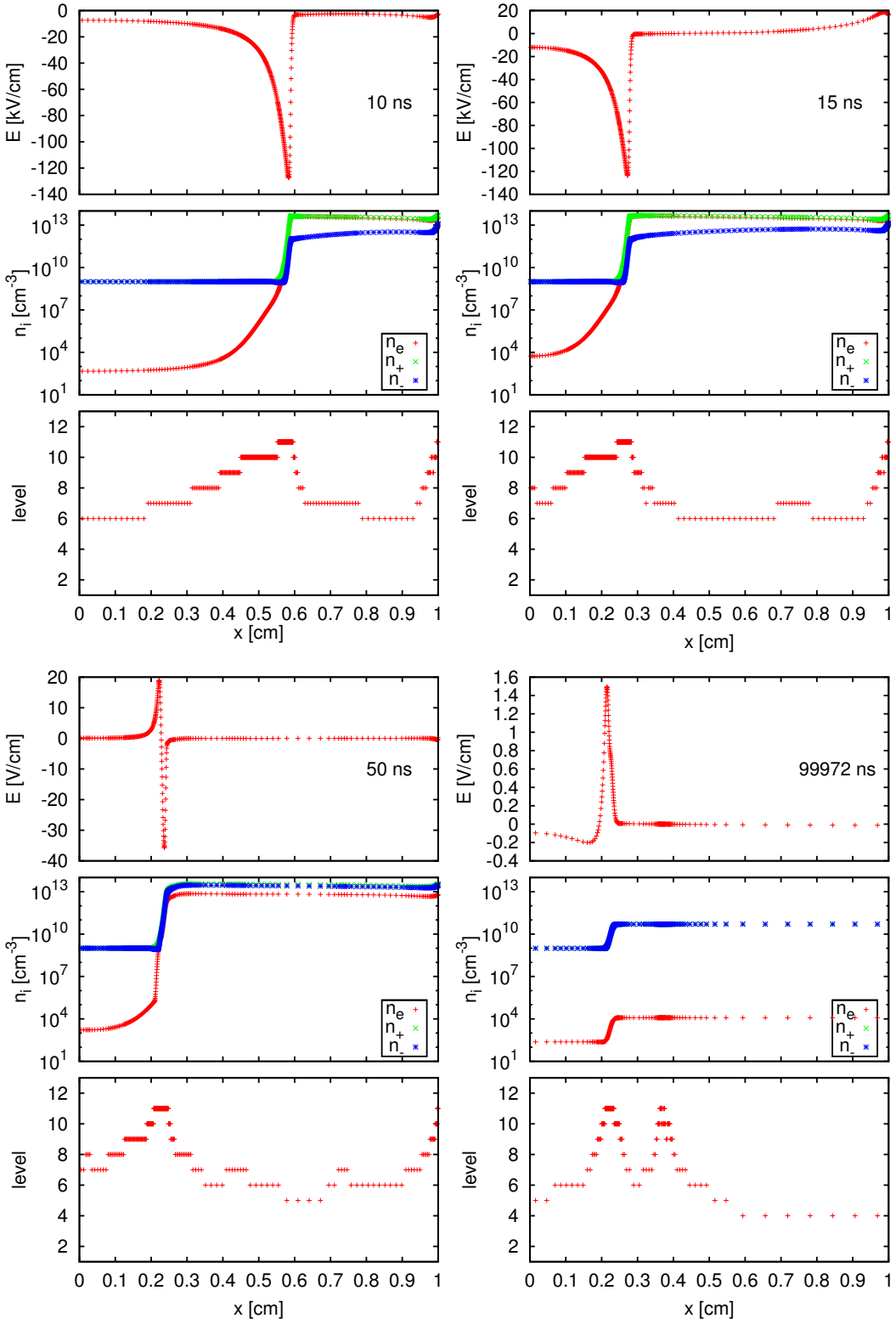

Fig. 11.14: *First period of pulsed discharges. Top: propagation of the discharge in the domain at $t = 10\,ns$ after the beginning of the pulse (left); and at $t = 15\,ns$ (right). Bottom: relaxation on the short time scale $t = 50\,ns$; and end of the relaxation phase after $t = 99972\,ns$ (right).*

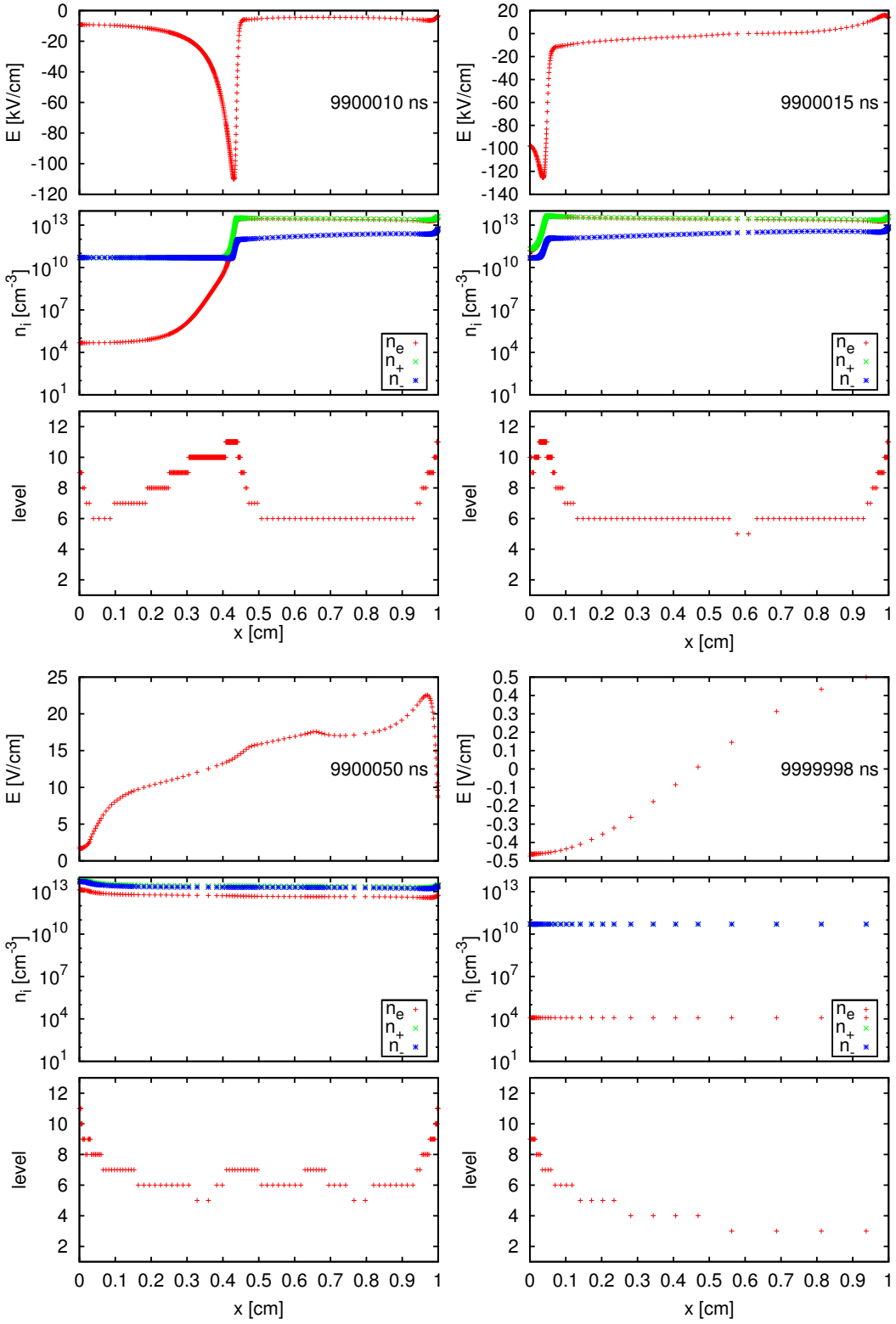

Fig. 11.15: *Steady-state of pulsed discharges (last period). Top: propagation of the discharge in the domain at* $t = 9900010\,ns$ *after the beginning of the pulse (left); and at* $t = 9900015\,ns$ *(right). Bottom: relaxation on short time scale* $t = 9900050\,ns$*; and end of the relaxation phase* $t = 9999998\,ns$ *(right).*

This faster propagation is mostly due to the higher preionization level of positive and negative ions in the gap in comparison of the first voltage pulse. We observe that for the 100th pulse, 15 ns after the beginning of the pulse the discharge has almost completely crossed the inter-electrode gap and then during the relaxation phase, there is no remaining space charge in the whole gap. Consequently, 50 ns after the beginning of the 100th pulse, axial distributions of all charged species are uniform. As already observed for the first pulse, at 50 ns after the beginning of the voltage pulse most electrons have attached and then, the density of positive ions is almost equal to the density of negative ions in the whole gap. We see that the corresponding electric field distribution is not uniform at 50 ns, but no steep gradients are observed as for the first voltage pulse. At $t = 9999998$ ns, that is to say at the end of the 100th period, we note that a very low electric field is obtained in the gap. An axially uniform distribution of charges is obtained with $5 \times 10^{10}\,\mathrm{cm}^{-3}$ for positive and negative ions and $10^4\,\mathrm{cm}^{-3}$ for electrons, which was the initial condition of the 100th pulse. This demonstrates the existence of a steady-state behavior of these nanosecond repetitively pulsed discharges.

11.6 Conclusions

The present work proposes a new numerical strategy for multi-scale streamer simulations. It is based on an adaptive second order time integration strategy that allows to discriminate time scales-related features of the phenomena, given a required level of accuracy of computations. Compared with a standard procedure for which accuracy is guaranteed by considering time steps of the order of the fastest scale, the control error approach implies on the one hand, an effective accurate resolution independent of the fastest physical time scale, and on the other hand, an important improvement of computational efficiency whenever the required time steps go beyond standard stability constraints. The latter is a direct consequence of the self-adaptive time step strategy for the resolution of the drift-diffusion equations which considers splitting time steps not limited by stability constraints for reaction, diffusion and convection phenomena. So far, the global decoupling time steps are limited by the dielectric relaxation stability constraint but with a second order accuracy. Nevertheless, we have also demonstrated that the decoupling time steps are rather chosen based on an accuracy criterion. Besides, if a technique such as a semi-implicit approach is implemented, the same ideas of the proposed adaptive strategy remain valid.

An adaptive multiresolution technique was also proposed in order to provide error control of the spatial adapted representation. The numerical results have proven a natural coupling between time and space accuracy requirements and how the set of time-space accuracy tolerances tunes the precise description of the overall time-space multi-scale phenomenon. As a consequence, the numerical results for multi-pulsed discharge configurations prove that this kind of multi-scale phenomena, previously out of reach, can be successfully simulated with conventional computing resources by this time-space adaptive strategy. And they also show that a consistent physical description is achieved for a broad spectrum of space and time scales as well as different physical scenarios.

In this work, we focused on a 1.5D model in order to evaluate the numerical performance of the strategy. However, the dimension of the problem will only have an influence on the computational efficiency measurements but not on any space-time accuracy or stability aspects. At this stage of development, the same numerical strategy can be coupled with a multi-dimensional Poisson's equation solver, even for adapted grid configurations as developed recently in [MHE06, PSCB08, UBRT10]. Finally, an important amount of work is still in progress concerning programming features such as data structures, optimized routines and

parallelization strategies. On the other hand, numerical analysis of theoretical aspects is also underway to extend and further improve the proposed numerical strategy. These issues constitute particular topics 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), and by a DIGITEO RTRA project: *MUSE* (project leader M. Massot).
Authors express special thanks to Christian Tenaud (LIMSI-CNRS) for providing the basis of the multiresolution kernel of MR_CHORUS, code developed for compressible Navier-Stokes equations (Déclaration d'Invention DI 03760-01). Support of Ecole Centrale Paris is gratefully acknowledged for several month stay of Z. Bonaventura at Laboratory EM2C as visiting Professor. Z. Bonaventura is also grateful to the Ministry of Education, Youth and Sports of the Czech Republic under project CZ.1.05/2.1.00/03.0086 and project MSM 0021622411.

General Conclusion and Prospects

The general framework of this work was settled by the development of efficient numerical strategies for the numerical simulation of problems with an important range of time and space scales describing real physical phenomena. In this large context, this work has particularly introduced a few mathematical and numerical tools that have yielded a general time/space adaptive numerical strategy for the numerical solution of stiff PDEs modeling reacting fronts. The practical implementation of the strategy has been carried out by the development of a numerical code, whereas the numerical simulation of several applications coming from different domains, namely nonlinear chemical dynamics, biomedical engineering, combustion, and plasma fields, has assessed the capabilities and potentials of the numerical approach. The theoretical background and some new achievements in terms of mathematical results as well as numerical tools, were thus put into practice to perform numerical simulations of models studied for scientific and technological purposes, coupling elements of applied mathematics and scientific computing for problems of interest for real applications. A key issue considered and discussed extensively in this work deals with the objective of providing valid error estimates for the numerical simulations. A special care was thus given to this aspect in the proposed numerical strategy in order to define general criteria to monitor and likely control the approximation errors introduced by the numerical methods. The latter issue is often underestimated or vaguely referred in most of the numerical simulations presented in the literature, in which no information is usually available concerning the accuracy achieved with the numerical methods. The numerical techniques developed in this work settle thus some fundamental bases to perform accurate and feasible numerical simulations of varied stiff problems, for large computational domains out of reach of standard techniques, with conventional computational resources.

The main contributions of this work, previously discussed in details, are summarized in the following:

- In terms of mathematical analysis:
  - The numerical analysis of the shifted and standard Strang splitting techniques for general nonlinear reaction-diffusion systems, published in [DDD+11]; and the theoretical characterization of numerical errors for self-similar solutions.
  - A numerical procedure to evaluate the Strang local truncation error and the study of non-asymptotic regimes for large splitting time steps [DDD+11].
  - New error estimates of splitting techniques for reaction-diffusion systems featuring solutions with high spatial gradients, published in [DDM11].
  - The numerical analysis of a parareal operator splitting technique for reaction-diffusion systems [DDM11]; and a theoretical characterization of numerical errors for self-similar solutions.
  - The construction of a second order scheme for plasma modeling equations, published in [DBM+12].

- In terms of numerical scheme:
  - An original splitting scheme for reaction-diffusion systems featuring dedicated, one-step and high order schemes for each split subproblem (introduced in [DMD+12, DDD+12] and presented in [DML+11]), such that the time integration errors for each split system are decoupled from the global splitting error[2].
  - A new parareal operator splitting scheme for PDEs modeling multi-scale reaction waves [DDM11]. To our knowledge, the parareal algorithm was not previously implemented and studied for stiff PDEs.
  - A space multiresolution technique coupled with a dedicated splitting technique for stiff reaction-diffusion systems [DMD+12, DML+11]. To our knowledge, this constitutes the first combination of such methods and the first use of multiresolution techniques for general stiff problems[3].
  - A time stepping splitting technique with local error control for stiff reaction-diffusion systems, and extensible to more general time dependent PDEs [DDD+11]. In our opinion, this constitutes a major contribution to the field because it opens a whole new set of possibilities for the numerical simulation of highly unsteady phenomena with splitting techniques[4].
  - A new time/space adaptive numerical technique with error control for stiff propagating waves [DMDD11, DMD+11a]. Although dynamic error control is common practice for stiff ODEs, this is usually not the case for time integration of PDEs. To our knowledge, this is one of the first numerical strategies that accounts for such an issue, and proposes general criteria to define the level of accuracy of the numerical simulation by decoupling and tracking time and space representation errors[5].
  - A new time stepping procedure with local error control for plasma modeling equations [DBM+12], based on a general second order scheme that decouples electric field computation from the solution of the drift-diffusion equations[6].
  - A time/space general adaptive numerical strategy with error control for plasma modeling equations [DBM+12]. To the best of our knowledge this is a novelty in the domain for the numerical simulation of streamers.
- In terms of scientific computing:
  - A new academic, generic 1D/2D/3D code has been developed for the numerical solution of stiff reaction-diffusion-convection problems with time/space adaptive

[2] This is the opposite of standard practices that consider small splitting time steps in practice to ensure negligible splitting errors, where the global errors are ruled by the inner time integration solvers (see, *e.g.*, [NK05, SPN06a] and detailed discussions in Chapter 4).

[3] For instance, the contemporaneous implementation of multiresolution analysis to simulate waves in excitable media in [BRBS10], considered a time integration technique specially conceived for the particular models under study.

[4] Contemporaneous works like the one in [KT11] or the Richardson extrapolation approach in [GV02], remain suitable only for non stiff problems. However, to the best of our knowledge, no other numerical simulation of large size or for varied applications, including this functionality, has been reported in the literature, except for the ones illustrated in this work.

[5] In other type of time/space adaptive techniques, such like local time stepping techniques conceived mainly for adaptive mesh configurations, the time adaptive criteria are usually based on local stability issues [CNPT08, DGRS08, CNPT10] rather than accuracy. The numerical technique presented in [DRS09] considers time step error control with multiresolution techniques and explicit solvers, appropriate to non stiff problems.

[6] Previous time adaptive procedures for these models considered local time stepping based either on local, dynamic increments of the solution [KDOO05, OK06] or on local stability conditions [UBRT07], but not on accuracy criteria.

features and with parallel routines for shared memory architectures. The code considers a tree-structured data for the multiresolution representation and a modular configuration to easily couple solvers of different nature. It is coded in the Fortran 90/95 programming language, and considers the OpenMP library for parallel procedures.

Additionally, let us highlight some achievements in terms of numerical simulations that to our knowledge, were never performed before in their respective domains:

- The numerical simulation of 3D scroll waves with an equivalent spatial discretization of 512^3 on a standard workstation of 32 GB.
- The numerical simulation of a 19-variable human brain stroke model on a simplified 3D geometry with an equivalent spatial discretization of 512^3 on a standard workstation of 32 GB.
- The consistent numerical simulation of different physical scenarios, depicted by the self-ignition of a reactive mixture with the subsequent generation and propagation of the flame, by means of dynamic time/space adaptivity tools.
- The consistent numerical simulation of highly multi-scale nanosecond repetitively pulsed discharges, describing a broad spectrum of space and time scales as well as different physical scenarios for consecutive discharge/post-discharge phases.

Finally, let us mention some further developments envisioned in a short term:

- A straightforward extension to the practical implementation of the numerical strategy developed for the simulation of streamers in [DBM+12], considers the inclusion of a Poisson's equation solver for adaptive grids in order to simulate multi-dimensional cases[7]. This kind of solver is available in the literature, and has been recently implemented, for instance, in [MHE06, PSCB08] for streamer simulation on adaptive grids. We recall that the second order method as well as the time stepping procedure in [DBM+12] are independent of the dimension of the problem.
- In the same way, a hydrodynamic solver can be coupled with the numerical strategy established in Chapter 10 for the simulation of combustion fronts, in order to consider complete physical models with complex chemistry and detailed transport, without adopting a thermo-diffusive approach[8].
- From a theoretical point of view, we have established some global accuracy criteria, based on the decoupling of time and space approximation errors for, respectively, the time integration method and the compressed spatial representation. The numerical evaluation

[7] This is a work in progress in collaboration with Zdeněk Bonaventura and the technical support of Laurent Series at the Computational Service Unit of Ecole Centrale Paris.

[8] This is a work in progress in collaboration with Violaine Louvet and Frédérique Laurent, for the practical implementation of detailed chemistry features. In general, the whole work is part of the *Séchelles* project that aims at developing a new, general, and optimized numerical code for the simulation of multi-scale problems. Thierry Dumont is currently leading the conception and construction process of the general structure of this code, that assembles all these recent numerical tools. We count also on the collaboration of Christian Tenaud regarding the implementation and numerical simulation of combustion fronts in the context of the Digiteo *MUSE* project.

of the simulations showed a good agreement between the numerical results and the prescribed tolerances in terms of accuracy. Nevertheless, there is naturally a certain dose of heuristics that can be certainly improved, by considering a more detailed mathematical analysis of the various elements of the strategy, *e.g.*, study of global time/space errors, characterization of coupled time/space approximation errors, mathematical description of wave profile and speed approximations.

References

[AB80] I. Abbas and P. Bayle. A critical analysis of ionizing wave-propagation mechanisms in breakdown. *J. Phys. D: Appl. Phys.*, 13(6):1055–1068, 1980.

[ABC+98] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, and M.L. Welcome. A conservative adaptive projection method for the variable density incompressible Navier-Stokes equations. *J. Comput. Phys.*, 142(1):1–46, 1998.

[Abd00] A. Abdulle. On roots and error constants of optimal stability polynomials. *BIT Numer. Math.*, 40(1):177–182, 2000.

[Abd02] A. Abdulle. Fourth order Chebyshev methods with recurrence relation. *SIAM J. Sci. Comput.*, 23(6):2041–2054, 2002.

[Abg95] R. Abgrall. Multiresolution analysis on unstructured meshes: application to CFD. In K.W. Morton and M.J. Baines, editors, *Numerical methods for fluid dynamics*, volume 5, pages 271–276. Oxford Science Publications, 1995.

[ABS05] M.J.H. Anthonissen, B.A.V. Bennett, and M.D. Smooke. An adaptive multilevel local defect correction technique with application to combustion. *Combust. Theory Modelling*, 9(2):273–299, 2005.

[ACM98] G. Akrivis, M. Crouzeix, and C. Makridakis. Implicit-explicit multistep finite element methods for nonlinear parabolic problems. *Math. Comp.*, 67(222):457–477, 1998.

[AEH02] M. Arrayás, U. Ebert, and W. Hundsdorfer. Spontaneous branching of anode-directed streamers between planar electrodes. *Phys. Rev. Lett.*, 88:174502(R), 2002.

[AH98] R. Abgrall and A. Harten. Multiresolution representation in unstructured meshes. *SIAM J. Numer. Anal.*, 35(6):2128–2146, 1998.

[AKV06] J.M. Alam, N.K.-R. Kevlahan, and O.V. Vasilyev. Simultaneous space-time adaptive wavelet solution of nonlinear parabolic differential equations. *J. Comput. Phys.*, 214(2):829–857, 2006.

[AM01] A. Abdulle and A.A. Medovikov. Second order Chebyshev methods based on orthogonal polynomials. *Numer. Math.*, 90(1):1–18, 2001.

[ARS97] U.M. Ascher, S.J. Ruuth, and R.J. Spiteri. Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. *Appl. Numer. Math.*, 25(2-3):151–167, 1997.

[ARW95] U.M. Ascher, S.J. Ruuth, and B.T.R. Wetton. Implicit-explicit methods for time-dependent partial differential equations. *SIAM Journal on Numerical Analysis*, 32(3):797–823, 1995.

[Axe72] O. Axelsson. A note on a class of strongly A-stable methods. *BIT Numer. Math.*, 12:1–4, 1972.

[BABS01] P.A. Barber, R.N. Auer, A.M. Buchan, and G.R. Sutherland. Understanding and managing ischemic stroke. *Can. J. Physiol. Pharmacol.*, 79(3):283–296, 2001.

[Bal03] G. Bal. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In *Proceedings of the 15th International Domain Decomposition Conference, Lect. Notes Comput. Sci. Eng. 40*, pages 426–432. Springer, Berlin, 2003.

[Bar91] D. Barkley. A model for fast computer simulation of waves in excitable media. *Physica D:Nonlinear Phenomena*, 49:61–70, 1991.

[BBB+10] J. Ballmann, M. Behr, K. Brix, W. Dahmen, C. Hohn, R. Massjung, S. Melian, S. Müller, and G. Schieffer. Parallel and adaptive methods for fluid-structure-interactions. In W. Schröder, editor, *Summary of Flow Modulation and Fluid-Structure Interaction Findings*, volume 109 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 265–294. Springer Berlin Heidelberg, 2010.

[BBC10] A. Bourdon, Z. Bonaventura, and S. Celestin. Influence of the pre-ionization background and simulation of the optical emission of a streamer discharge in preheated air at atmospheric pressure between two point electrodes. *Plasma Sources Sci. Technol.*, 19(3):034012, 2010.

[BBH89] P.N. Brown, G.D. Byrne, and A.C. Hindmarsh. VODE: A variable-coefficient ODE solver. *SIAM J. Sci. Stat. Comput.*, 10:1038–1051, 1989.

[BBM+02] L. Baffico, S. Bernard, Y. Maday, G. Turinici, and G. Zérah. Parallel-in-time molecular-dynamics simulations. *Physical Review E*, 66(057706):1–4, 2002.

[BBRBS09] M. Bendahmane, R. Bürger, R. Ruiz-Baier, and K. Schneider. Adaptive multiresolution schemes with local time stepping for two-dimensional degenerate reaction-diffusion systems. *Appl. Numer. Math.*, 59(7):1668–1692, 2009.

[BBSW94] J. Bell, M.J. Berger, J. Saltzman, and M. Welcome. Three-dimensional adaptive mesh refinement for hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 15:127–138, 1994.

[BC89] M.J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:67–84, 1989.

[BCR91] G. Beylkin, R. Coifman, and V. Rokhlin. Fast wavelet transforms and numerical algorithms I. *Comm. Pure and Applied Math.*, 44(2):141–183, 1991.

[BCVK02] H. Bijl, M.H. Carpenter, V.N. Vatsa, and C.A. Kennedy. Implicit time integration schemes for the unsteady compressible Navier-Stokes equations: Laminar flow. *J. Comput. Phys.*, 179(1):313–329, 2002.

[BDA+06] J.B. Bell, M.S. Day, A. Almgren, M. Lijewski, C. Rendleman, R. Cheng, and I. Shepherd. Simulation of lean premixed turbulent combustion. *Journal of Physics: Conference Series*, 46(1):1, 2006.

[BDG02] J.B. Bell, M.S. Day, and J.F. Grcar. Numerical simulation of premixed turbulent methane combustion. *Proc. Combust. Inst.*, 29(2):1987–1993, 2002.

[BDG+07] J.B. Bell, M.S. Day, J.F. Grcar, M.J. Lijewski, J.F. Driscoll, and S.A. Filatyev. Numerical simulation of a laboratory-scale turbulent slot flame. *Proc. Combust. Inst.*, 31(1):1299–1307, 2007.

[BDS+05] J.B. Bell, M.S. Day, I.G. Shepherd, M.R. Johnson, R.K. Cheng, J.F. Grcar, V.E. Beckner, and M.J. Lijewski. Numerical simulation of a laboratory-scale turbulent V-flame. *Proc. Nat. Acad. Science*, 1021:10006–10011, 2005.

[Ber82] M.J. Berger. *Adaptive Mesh Refinement for Hyperbolic Differential Equations.* PhD thesis, Stanford University, 1982.

[BG02] A. Bartel and M. Günther. A multirate W-method for electrical networks in state-space formulation. *J. Comput. Appl. Math.*, 147(2):411–425, 2002.

[BGMH+03] F. Bramkamp, B. Gottschlich-Müller, M. Hesse, P. Lamby, S. Müller, J. Ballmann, K.H. Brakhage, and W. Dahmen. H-adaptive multiscale schemes for compressible Navier-Stokes equations - polyhedral discretization, data compression and mesh generation. In J. Ballmann, editor, *Flow Modulation and Fluid-Structure Interaction at Airplane Wings*, volume 84 of *Numerical notes on Fluid Mechanics*, pages 125–204. Springer, 2003.

[BH97] B.L. Bihari and A. Harten. Multiresolution schemes for the numerical solution of 2-D conservation laws I. *SIAM J. Sci. Comput.*, 18(2):315–354, 1997.

[BL98] M.J. Berger and R.J. LeVeque. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM J. Numer. Anal.*, 35:2298–2316, 1998.

[BLM04] F. Bramkamp, P. Lamby, and S. Müller. An adaptive multiscale finite volume solver for unsteady and steady state flow computations. *J. Comput. Phys.*, 197(2):460–490, 2004.

[BM03] G. Bal and Y. Maday. A "parareal" time discretization for non-linear PDE's with application to the pricing of an american put. In *Recent Developments in Domain Decomposition Methods, Lect. Notes Comput. Sci. Eng. 23*, pages 189–202. Springer, Berlin, 2003.

[BMMS09] K. Brix, S. Melian, S. Müller, and G. Schieffer. Parallelisation of multiscale-based grid adaptation using space-filling curves. *ESAIM: Proc.*, 29:108–129, 2009.

[BMP92] E. Bacry, S. Mallat, and G. Papanicolaou. A wavelet based space-time adaptive numerical method for partial differential equations. *Math. Model. Numer. Anal.*, 26:793–834, 1992.

[BMP+09] B.A.V. Bennett, C.S. McEnally, L.D. Pfefferle, M.D. Smooke, and M.B. Colket. Computational and experimental study of the effects of adding dimethyl ether and ethanol to nonpremixed ethylene/air flames. *Combust. and Flame*, 156(6):1289–1302, 2009.

[BMV09] K. Brix, R. Massjung, and A. Voss. A hash data structure for adaptive PDE-solvers based on discontinuous Galerkin discretizations. *IGPM-Rep. 302, RWTH Aachen*, 2009.

[BN96] N.Y. Babaeva and G.V. Naidis. Two-dimensional modelling of positive streamer dynamics in non-uniform electric fields in air. *J. Phys. D: Appl. Phys.*, 29:2423–2431, 1996.

[BN97] N.Y. Babaeva and G.V. Naidis. Dynamics of positive and negative streamers in air in weak uniform electric fields. *IEEE Trans. Plasma Sci.*, 25:375–379, 1997.

[BN03] M.S. Benilov and G.V. Naidis. Modelling of low-current discharges in atmospheric-pressure air taking account of non-equilibrium effects. *J. Phys. D: Appl. Phys.*, 36(15):1834–1841, 2003.

[BO84] M.J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512, 1984.

[Bot92] A. Bott. Monotone flux limitation in the area-preserving flux-form advection algorithm. *Monthly Weather Review*, 120(11):2592–2602, 1992.

[BPB+07] D. Bessieres, J. Paillol, A. Bourdon, P. Segur, and E. Marode. A new one-dimensional moving mesh method applied to the simulation of streamer discharges. *J. Phys. D: Appl. Phys.*, 40(21):6559–6570, 2007.

[BPL+07] A. Bourdon, V.P. Pasko, N.Y. Liu, S. Celestin, P. Segur, and E. Marode. Efficient models for photoionization produced by non-thermal gas discharges in air based on radiative transfer and the Helmholtz equations. *Plasma Sources Sci. Technol.*, 16(3):656–678, 2007.

[Bra77] A. Brandt. Multi-level adaptive solutions to boundary value problems. *Math. Comp.*, 31:333–390, 1977.

[BRBS10] R. Bürger, R. Ruiz-Baier, and K. Schneider. Adaptive multiresolution methods for the simulation of waves in excitable media. *J. Scientific Computing*, 43:261–290, 2010.

[BRBSS08] R. Bürger, R. Ruiz-Baier, K. Schneider, and M. Sepúlveda. Fully adaptive multiresolution schemes for strongly degenerate parabolic equations in one space dimension. *ESAIM: Math. Model. Numer. Anal.*, 42:535–563, 2008.

[But64a] J.C. Butcher. Implicit Runge-Kutta processes. *Math. Comp.*, 18:50–64, 1964.

[But64b] J.C. Butcher. Integration processes based on Radau quadrature formulas. *Math. Comp.*, 18:233–244, 1964.

[But64c] J.C. Butcher. On Runge-Kutta processes of high order. *J. Austral. Math. Soc.*, 4:179–194, 1964.

[But64d] J.C. Butcher. On the attainable order of Runge-Kutta methods. *Math. Comp.*, 19:408–417, 1964.

[But76] J.C. Butcher. On the implementation of implicit Runge-Kutta methods. *BIT Numer. Math.*, 6:237–240, 1976.

[BWZ+02] O. Bernus, R. Wilders, C.W. Zemlin, H. Verschelde, and A.V. Panfilovz. A computationally efficient electrophysiological model of human ventricular cells. *Am. J. Physiol. Heart Circ. Physiol.*, 282(6):H2296–H2308, 2002.

[CBZ+09] S. Celestin, Z. Bonaventura, B. Zeghondy, A. Bourdon, and P. Segur. The use of the ghost fluid method for Poisson's equation to simulate streamer propagation in point-to-plane and point-to-point geometries. *J. Phys. D: Appl. Phys.*, 42(6):065203, 2009.

[CCdS+09] J.H. Chen, A. Choudhary, B. de Supinski, M. DeVries, E.R. Hawkes, S. Klasky, W.K. Liao, K.L. Ma, J. Mellor-Crummey, N. Podhorszki, R. Sankaran, S. Shende, and C.S. Yoo. Terascale direct numerical simulations of turbulent combustion using S3D. *Comp. Science & Discovery*, 2(1):015001, 2009.

[CCDV09] F. Castella, P. Chartier, S. Descombes, and G. Vilmart. Splitting methods with complex times for parabolic equations. *BIT Numer. Math.*, 49:487–508, 2009.

[CD01] G. Chiavassa and R. Donat. Point value multiscale algorithms for 2D compressible flows. *SIAM J. Sci. Comput.*, 23:805–823, 2001.

[CD11] B. Chabaud and Q. Du. A hybrid implicit-explicit adaptive multirate numerical scheme for time-dependent equations. *J. Scientific Computing*, pages 1–23, 2011.

[CDD04] A. Cohen, W. Dahmen, and R. DeVore. *Adaptive Wavelet Techniques in Numerical Simulation.* John Wiley & Sons, Ltd., 2004.

[CDF92] A. Cohen, I. Daubechies, and J.-C. Feauveau. Biorthogonal bases of compactly supported wavelets. *Comm. Pure and Applied Math.*, 45(5):485–560, 1992.

[CDKP00] A. Cohen, N. Dyn, S.M. Kaber, and M. Postel. Multiresolution finite volume schemes on triangles. *J. Comput. Phys.*, 161:264–286, 2000.

[CDW99] P. Colella, M.R. Dorr, and D.D. Wake. A conservative finite difference method for the numerical solution of plasma fluid equations. *J. Comput. Phys.*, 149(1):168–193, 1999.

[Cel08] S. Celestin. *Study of the streamer dynamics in air at atmospheric pressure.* PhD thesis, Ecole Centrale Paris, France, 2008.

[CFDE+06] P. Colli Franzone, P. Deuflhard, B. Erdmann, J. Lang, and L.F. Pavarino. Adaptivity in space and time for reaction-diffusion systems in electrocardiology. *SIAM J. Sci. Comput.*, 28:942–962, 2006.

[CFP04] P. Colli Franzone and L.F. Pavarino. A parallel solver for reaction–diffusion systems in computational electrocardiology. *Math. Models Methods Appl. Sci.*, 14:883–911, 2004.

[CGH00] E.M. Cherry, H.S. Greenside, and C.S. Henriquez. A space-time adaptive method for simulating complex cardiac dynamics. *Phys. Rev. Lett.*, 84(6):1343–1346, 2000.

[CGH03] E.M. Cherry, H.S. Greenside, and C.S. Henriquez. Efficient simulation of three-dimensional anisotropic cardiac tissue using an adaptive mesh refinement method. *Chaos*, 13(3):853–865, 2003.

[Che11] J.H. Chen. Petascale direct numerical simulation of turbulent combustion-fundamental insights towards predictive models. *Proc. Combust. Inst.*, 33(1):99–123, 2011.

[Chi71] F.H. Chipman. A-stable Runge-Kutta processes. *BIT Numer. Math.*, 11:384–388, 1971.

[CKB+05] M. Carpenter, C. Kennedy, H. Bijl, S. Viken, and V. Vatsa. Fourth-order Runge-Kutta schemes for fluid mechanics applications. *J. Scientific Computing*, 25:157–194, 2005.

[CKMP03] A. Cohen, S.M. Kaber, S. Müller, and M. Postel. Fully adaptive multiresolution finite volume schemes for conservation laws. *Math. Comp.*, 72:183–225, 2003.

[CNPT08] F. Coquel, Q.L. Nguyen, M. Postel, and Q.H. Tran. Local time stepping for implicit-explicit methods on time varying grids. In K. Kunisch *et al.*, editor, *Numerical Mathematics and Advanced Applications*, pages 257–264. Springer Berlin Heidelberg, 2008.

[CNPT09] F. Coquel, Q.L. Nguyen, M. Postel, and Q.H. Tran. Local time stepping with adaptive time step control for a two-phase fluid system. *ESAIM: Proc.*, 29:73–88, 2009.

[CNPT10] F. Coquel, Q.L. Nguyen, M. Postel, and Q.H. Tran. Local time stepping applied to implicit-explicit methods for hyperbolic systems. *Multiscale Model. Simul.*, 8(2):540–570, 2010.

[Coh00] A. Cohen. *Wavelet Methods in Numerical Analysis*, volume 7. Elsevier, Amsterdam, 2000.

[CP93] P. Chartier and B. Philippe. A parallel shooting technique for solving dissipative ODEs. *Computing*, 51:209–236, 1993.

[CPPT06] F. Coquel, M. Postel, N. Poussineau, and Q.H. Tran. Multiresolution technique and explicit-implicit scheme for multicomponent flows. *J. of Num. Math*, 14:187–216, 2006.

[CPT11] F. Coquel, M. Postel, and Q.H. Tran. Convergence of time-space adaptive algorithms for nonlinear conservation laws. *Preprint, available at http://www.ann.jussieu.fr/~postel*, 2011.

[Cro80] M. Crouzeix. Une méthode multipas implicite-explicite pour l'approximation des équations d'évolution paraboliques. *Numer. Math.*, 35:257–276, 1980.

[CW93] W.Y. Crutchfield and M.L. Welcome. Object-oriented implementation of adaptive mesh refinement algorithms. *Sci. Program.*, 2:145–156, 1993.

[D'A94] Y. D'Angelo. *Analyse et simulation numérique de phénomènes liés à la combustion supersonique*. PhD thesis, Ecole Nationale des Ponts et Chaussées, 1994.

[Dah63] G. Dahlquist. A special stability problem for linear multistep methods. *Nordisk Tidskr. Informations-Behandling*, 3:27–43, 1963.

[Dau88] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Comm. Pure and Applied Math.*, 41(7):909–996, 1988.

[Dau92] I. Daubechies. *Ten Lectures on Wavelets*, volume 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.

[DB00] M.S. Day and J.B. Bell. Numerical simulation of laminar reacting flows with complex chemistry. *Combust. Theory Modelling*, 4:535–556, 2000.

[DBG06] M.-A. Dronne, J.-P. Boissel, and E. Grenier. A mathematical model of ion movements in grey matter during a stroke. *J. of Theoretical Biology*, 240(4):599–615, 2006.

[DBM+12] M. Duarte, Z. Bonaventura, M. Massot, A. Bourdon, S. Descombes, and T. Dumont. A new numerical strategy with space-time adaptivity and error control for multi-scale streamer discharge simulations. *J. Comput. Phys.*, 231:1002–1019, 2012.

[dC09] S. de Chaisemartin. *Eulerian models and numerical simulation of turbulent dispersion for polydisperse evaporation sprays*. PhD thesis, Ecole Centrale Paris, 2009.

[DCB+09] S.B. Dworkin, J.A. Cooke, B.A.V. Bennett, B.C. Connelly, M.B. Long, M.D. Smooke, R.J. Hall, and M.B. Colket. Distributed-memory parallel computation of a forced, time-dependent, sooting, ethylene/air coflow diffusion flame. *Combust. Theory Modelling*, 13(5):795–822, 2009.

[DCS+07] S.B. Dworkin, B.C. Connelly, A.M. Schaffer, B.A.V. Bennett, M.B. Long, M.D. Smooke, M.P. Puccio, B. McAndrews, and J.H. Miller. Computational and experimental study of a forced, time-dependent, methane-air coflow diffusion flame. *Proc. Combust. Inst.*, 31(1):971–978, 2007.

[DD08a] S. Descombes and T. Dumont. Numerical simulation of a stroke: Computational problems and methodology. *Progress in Biophysics Molecular Biology*, 97:40–53, 2008.

[DD08b] S. Descombes and T. Dumont. Numerical simulation of a stroke: Computational problems and methodology. *Progress in Biophysics & Molecular Biology*, 97(1):40–53, 2008.

[DDD+11] S. Descombes, M. Duarte, T. Dumont, V. Louvet, and M. Massot. Adaptive time splitting method for multi-scale evolutionary partial differential equations. *Confluentes Mathematici*, 3(3):413–443, 2011.

[DDD+12] T. Dumont, M. Duarte, S. Descombes, M.A. Dronne, M. Massot, and V. Louvet. Simulation of human ischemic stroke in realistic 3D geometry: A numerical strategy. *Submitted, available at HAL (http://hal.archives-ouvertes.fr/hal-00546223)*, 2012.

[DDE71] A.J. Davies, C.S. Davies, and C.J. Evans. Computer simulation of rapidly developing gaseous discharges. *Proc. of the Institution of Electrical Engineers-London*, 118(6):816–823, 1971.

[DDG+09] R. Deiterding, M.O. Domingues, S.M. Gomes, O. Roussel, and K. Schneider. Adaptive multiresolution or adaptive mesh refinement? A case study for 2D Euler equations. *ESAIM: Proc.*, 29:28–42, 2009.

[DDGG09] M.-A. Dronne, S. Descombes, E. Grenier, and H. Gilquin. Examples of the influence of the geometry on the propagation of progressive waves. *Math. Comput. Modelling*, 49(11-12):2138–2144, 2009.

[DDL+12] S. Descombes, T. Dumont, V. Louvet, M. Massot, F. Laurent, and J. Beaulaurier. Operator splitting techniques for multi-scale reacting waves and application to low Mach number flames with complex chemistry: Theoretical and numerical aspects. *In preparation*, 2012.

[DDLM07] S. Descombes, T. Dumont, V. Louvet, and M. Massot. 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, 2007.

[DDM03] S. Descombes, T. Dumont, and M. Massot. Operator splitting for stiff nonlinear reaction-diffusion systems: Order reduction and application to spiral waves. In *Patterns and waves (Saint Petersburg, 2002)*, pages 386–482. AkademPrint, St. Petersburg, 2003.

[DDM11] M. Duarte, S. Descombes, and M. Massot. Parareal operator splitting techniques for multi-scale reaction waves: Numerical analysis and strategies. *ESAIM: Math. Model. Numer. Anal.*, 45:825–852, 2011.

[Dei05] R. Deiterding. Construction and application of an amr algorithm for distributed memory computers. In T.J. Barth *et al.*, editor, *Adaptive Mesh Refinement - Theory and Applications*, volume 41 of *Lecture Notes in Computational Science and Engineering*, pages 361–372. Springer Berlin Heidelberg, 2005.

[Des01] S. Descombes. Convergence of a splitting method of high order for reaction-diffusion systems. *Math. Comp.*, 70(236):1481–1501, 2001.

[Deu74] P. Deuflhard. A modified Newton method for the solution of ill-conditioned systems of nonlinear equations with application to multiple shooting. *Numer. Math.*, 22:289–315, 1974.

[Deu83] P. Deuflhard. Order and stepsize control in extrapolation methods. *Numer. Math.*, 41:399–422, 1983.

[Deu04] P. Deuflhard. *Newton Methods for Nonlinear Problems.* Springer-Verlag, 2004. Affine Invariance and Adaptive Algorithms.

[DFN85] D. Di Francesco and D. Noble. A model of cardiac electrical activity incorporating ionic pumps and concentration changes. *Philos. Trans. R. Soc. Lond. B Biol. Sci.*, 307(1133):353–398, 1985.

[DGC+08] M.-A. Dronne, E. Grenier, G Chapuisat, M. Hommel, and J.-P. Boissel. A modelling approach to explore some hypotheses of the failure of neuroprotective trials in ischemic stroke patient. *Progress in Biophysics & Molecular Biology*, 97:60–78, 2008.

[DGD+07] M.-A. Dronne, E. Grenier, T. Dumont, M. Hommel, and J.-P. Boissel. Role of astrocytes in grey matter during a stroke: A modelling approach. *Brain Research*, 1138:231–242, 2007.

[DGMM01] W. Dahmen, B. Gottschlich-Müller, and S. Müller. Multiresolution schemes for conservation laws. *Numer. Math.*, 88:399–443, 2001.

[DGOZ04] I. Dimov, K. Georgiev, T. Ostromsky, and Z. Zlatev. Computational challenges in the numerical treatment of large air pollution models. *Ecological Modelling*, 179(2):187–203, 2004.

[DGRS08] M.O. Domingues, S.M. Gomes, O. Roussel, and K. Schneider. An adaptive multiresolution scheme with local time stepping for evolutionary PDEs. *J. Comp. Phys.*, 227(8):3758–3780, 2008.

[DGRS09] M.O. Domingues, S.M. Gomes, O. Roussel, and K. Schneider. Space-time adaptive multiresolution methods for hyperbolic conservation laws: Applications to compressible Euler equations. *Appl. Numer. Math.*, 59(9):2303–2321, 2009.

[DIM99] U. Dirnagl, C. Iadecola, and M.A. Moskowitz. Pathobiology of ischaemic stroke: An integrated view. *Trends in Neurosciences*, 22(9):391–397, 1999.

[DJE64] A.J. Davies, F.L. Jones, and C.J. Evans. Electrical breakdown of gases - Spatio-temporal growth of ionization in fields distorted by space charge. *Proc. Royal Soc. London S. A-Math. and Phys. Scien.*, 281(1385):164–183, 1964.

[DKSL99] J. De Keyser, G. Sulter, and P. Luiten. Clinical trials with neuroprotective drugs in acute ischaemic stroke: are we doing the right thing? *Trends in Neurosciences*, 22(12):535–40, 1999.

[DL95a] Y. D'Angelo and B. Larrouturou. Comparison and analysis of some numerical schemes for stiff complex chemistry problems. *RAIRO Modél. Math. Anal. Numér.*, 29(3):259–301, 1995.

[DL95b] Y. D'Angelo and B. Larrouturou. Comparison and analysis of some numerical schemes for stiff complex chemistry problems. *RAIRO Modél. Math. Anal. Numér.*, 29(3):259–301, 1995.

[DLR+01] P.M. Desmond, A.C. Lovell, A.A. Rawlinson, M.W. Parsons, P.A. Barber, Qing Yang, Ting Li, D.G. Darby, R.P. Gerraty, S.M. Davis, and B.M. Tress. The value of apparent diffusion coefficient maps in early cerebral ischemia. *Am. J. Neuroradiol.*, 22(7):1260–1267, 2001.

[DM04] S. Descombes and M. Massot. Operator splitting for nonlinear reaction-diffusion systems with an entropic structure: Singular perturbation and order reduction. *Numer. Math.*, 97(4):667–698, 2004.

[DMB97] M. Dowle, R.M. Mantel, and D. Barkley. Fast simulations of waves in three-dimensional excitable media. *Int. J. Bif. Chaos*, 7:2529–2545, 1997.

[DMD+11a] M. Duarte, M. Massot, S. Descombes, C. Tenaud, and S. Candel. Time-space adaptive numerical methods for the simulation of combustion fronts. In *Annual Research Briefs 2011*, pages 347–358. Center for Turbulence Research, Stanford University, 2011.

[DMD+11b] M. Duarte, M. Massot, S. Descombes, C. Tenaud, T. Dumont, V. Louvet, and F. Laurent. New resolution strategy for multi-scale reaction waves using time operator splitting and space adaptive multiresolution: Application to human ischemic stroke. *ESAIM: Proc.*, 34:277–290, 2011.

[DMD+12] M. Duarte, M. Massot, S. Descombes, C. Tenaud, T. Dumont, V. Louvet, and F. Laurent. New resolution strategy for multi-scale reaction waves using time operator splitting, space adaptive multiresolution and dedicated high order implicit/explicit time integrators. *SIAM J. Sci. Comput.*, 34(1):A76–A104, 2012.

[DMDD11] M. Duarte, M. Massot, S. Descombes, and T. Dumont. Adaptive time-space algorithms for the simulation of multi-scale reaction waves. In J. Fořt *et al.*, editor, *Finite Volumes for Complex Applications VI Problems & Perspectives*, volume 4, pages 379–387. Springer Berlin Heidelberg, 2011.

[DML+11] M. Duarte, M. Massot, F. Laurent, S. Descombes, C. Tenaud, T. Dumont, and V. Louvet. New resolution strategies for multi-scale reaction waves: Optimal time operator splitting and space adaptive multiresolution. In L. Cernuzzi *et al.*, editor, *Special issue of best papers presented at CLEI'2010, Asunción, Paraguay*, volume 14, page 14. CLEI electronic journal, 2011.

[DMS98] A. Destexhe, Z.F. Mainen, and T.J. Sejnowski. *Kinetic models of synaptic transmission*, chapter 1. MIT Press,Cambridge, MA, 1998.

[DRS09] M.O. Domingues, O. Roussel, and K. Schneider. An adaptive multiresolution method for parabolic PDEs with time-step control. *International Journal for Numerical Methods in Engineering*, 78(6):652–670, 2009.

[DS84] R. DeVore and R. Sharpley. *Maximal Functions Measuring Smoothness*, volume 47. Mem. Am. Math. Soc., 1984.

[DS02a] S. Descombes and M. Schatzman. Strang's formula for holomorphic semi-groups. *J. Math. Pures Appl.*, 81(1):93–114, 2002.

[DS02b] R. Djouad and B. Sportisse. Partitioning techniques and lumping computation for reducing chemical kinetics. APLA: An automatic partitioning and lumping algorithm. *Appl. Numer. Math.*, 43(4):383–398, 2002.

[DS03] R. Djouad and B. Sportisse. Solving reduced chemical models in air pollution modelling. *Appl. Numer. Math.*, 44(1-2):49–61, 2003.

[DS10] R. Dobbins and M. Smooke. A fully implicit, compact finite difference method for the numerical solution of unsteady laminar flames. *Applied Scientific Research*, 85(3-4):763–799, 2010.

[DSF+08] C. Dohmen, O.W. Sakowitz, M. Fabricius, B. Bosche, T. Reithmeier, R.-I. Ernestus, G. Brinker, J.P. Dreier, J. Woitzik, A.J. Strong, and R. Graf. Spreading depolarizations occur in human ischemic stroke with high incidence. *Annals of Neurology*, 63:720–728, 2008.

[DT01] V. Daru and C. Tenaud. Evaluation of TVD high resolution schemes for unsteady viscous shocked flows. *Computers & Fluids*, 30:89–113, 2001.

[DT04] V. Daru and C. Tenaud. High order one-step monotonicity-preserving schemes for unsteady compressible flow calculations. *J. Comput. Phys.*, 193(2):563–594, 2004.

[DT07] A. Durukan and T. Tatlisumak. Acute ischemic stroke: Overview of major experimental rodent models, pathophysiology, and therapy of focal cerebral ischemia. *Pharmacology Biochemistry and Behavior*, 87(1):179–197, 2007.

[DT09] V. Daru and C. Tenaud. Numerical simulation of the viscous shock tube problem by using a high resolution monotonicity-preserving scheme. *Computers & Fluids*, 38:664–676, 2009.

[DT10] S. Descombes and M. Thalhammer. An exact local error representation of exponential operator splitting methods for evolutionary problems and applications to linear Schrödinger equations in the semi-classical regime. *BIT Numer. Math.*, 50:729–749, 2010.

[DT11] S. Descombes and M. Thalhammer. The Lie-Trotter splitting method for nonlinear evolutionary problems involving critical parameters. An exact local error representation and application to nonlinear Schrödinger equations in the semiclassical regime. *Preprint, available at HAL (http://hal.archives-ouvertes.fr/hal-00557593)*, 2011.

[Dum07] T. Dumont. ZEBRE: Numerical software for reaction-diffusion systems. Source code and documentation at: http://math.univ-lyon1.fr/~tdumont/zebre/, 2007.

[DW87] S.K. Dhali and P.F. Williams. Two-dimensional studies of streamers in gases. *J. Appl. Phys.*, 62:4696–4707, 1987.

[EBD+11] U. Ebert, F. Brau, G. Derks, W. Hundsdorfer, C.-Y. Kao, C. Li, A. Luque, B. Meulenbroek, S. Nijdam, V. Ratushnaya, L. Schäfer, and S. Tanveer. Multiple scales in streamer discharges, with an emphasis on moving boundary approximations. *Nonlinearity*, 24(1):C1–C26, 2011.

[ECG+08] D. Estep, V. Carey, V. Ginting, S. Tavener, and T. Wildey. A posteriori error analysis of multiscale operator decomposition methods for multiphysics models. *Journal of Physics: Conference Series*, 125(1):012075, 2008.

[Ech09] T. Echekki. Multiscale methods in turbulent combustion: strategies and computational challenges. *Comp. Science & Discovery*, 2:013001, 2009.

[EGR+08] D. Estep, V. Ginting, D. Ropp, J.N. Shadid, and S. Tavener. An a posteriori-a priori analysis of multiscale operator splitting. *SIAM J. Numer. Anal.*, 46:1116–1146, 2008.

[Ehl68] B.L. Ehle. High order A-stable methods for the numerical solution of systems of DEs. *BIT Numer. Math.*, 8:276–278, 1968.

[Ehl69] B.L. Ehle. On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems. *Research Report CSRR 2010*, 1969.

[Ehl73] B.L. Ehle. *A*-stable methods and Padé approximations to the exponential. *SIAM J. Math. Anal.*, 4:671–680, 1973.

[EMB+06] U. Ebert, C. Montijn, T. M. P. Briels, W. Hundsdorfer, B. Meulenbroek, A. Rocco, and E.M. van Veldhuizen. The multiscale nature of streamers. *Plasma Sources Sci. Technol.*, 15:S118–S129, 2006.

[EMM28] C.F. Eyring, S.S. Mackeown, and R.A. Millikan. Fields currents from points. *Phys. Rev.*, 31(5):900–909, 1928.

[EOZ94] B. Engquist, S. Osher, and S. Zhong. Fast wavelet based algorithms for linear evolution equations. *SIAM J. Sci. Comput.*, 15(4):755–775, 1994.

[EP98] I.R. Epstein and J.A. Pojman. *An Introduction to Nonlinear Chemical Dynamics.* Oxford University Press, 1998. Oscillations, Waves, Patterns and Chaos.

[FC03] C. Farhat and M. Chandesris. Time-decomposed parallel time-integrators: Theory and feasibility studies for fluid, structure, and fluid-structure applications. *Int. J. Numer. Meth. Engng.*, 58(9):1397–1434, 2003.

[FCG05] A. Fridman, A. Chirokov, and A. Gutsol. Non-thermal atmospheric pressure discharges. *J. Phys. D: Appl. Phys.*, 38:R1–R24, 2005.

[FGM97] H. Friedel, R. Grauer, and C. Marliani. Adaptive mesh refinement for singular current sheets in incompressible magnetohydrodynamic flows. *J. Comput. Phys.*, 134(1):190–198, 1997.

[FHM03] F. Fischer, F. Hecht, and Y. Maday. A parareal in time semi-implicit approximation of the Navier-Stokes equations. In *Proceedings of the 15th International Domain Decomposition Conference, Lect. Notes Comput. Sci. Eng. 40*, pages 433–440. Springer, Berlin, 2003.

[FHV97] J. Frank, W. Hundsdorfer, and J.G. Verwer. On the stability of implicit-explicit linear multistep methods. *Appl. Numer. Math.*, 25(2-3):193–205, 1997.

[FKN72] R.J. Field, E. Koros, and R.M. Noyes. Oscillations in chemical systems. II. Thorough analysis of temporal oscillation in the bromate-cerium-malonic acid system. *J. Amer. Chem. Soc.*, 94(25):8649–8664, 1972.

[FKR+01] J. Fiehler, R. Knab, J.R. Reichenbach, C. Fitzek, C. Weiller, and J. Röther. Apparent diffusion coefficient decreases and magnetic resonance imaging perfusion parameters are associated in ischemic tissue of acute stroke patients. *J. Cereb. Blood Flow & Metab.*, 97(1):54–89, 2001.

[GDD+08a] E. Grenier, M.-A. Dronne, S. Descombes, H. Gilquin, A. Jaillard, M. Hommel, and J.-P. Boissel. A numerical study of the blocking of migraine by Rolando sulcus. *Progress in Biophysics & Molecular Biology*, 97(1):54–59, 2008.

[GDD+08b] E. Grenier, M.A. Dronne, S. Descombes, H. Gilquin, A. Jaillard, M. Hommel, and J.P. Boissel. A numerical study of the blocking of migraine by Rolando sulcus. *Progress in Biophysics and Molecular Biology*, 97:54–59, 2008.

[Gea71] C.W. Gear. *Numerical Initial Value Problems in Ordinary Differential Equations.* Prentice-Hall series in automatic computation. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[GEF03] I. Garrido, M.S. Espedal, and G.E. Fladmark. A convergence algorithm for time parallelization applied to reservoir simulation. In *Proceedings of the 15th International Domain Decomposition Conference, Lect. Notes Comput. Sci. Eng. 40*, pages 469–476. Springer, Berlin, 2003.

[GH08] M. Gander and E. Hairer. Nonlinear convergence analysis for the parareal algorithm. In *Domain Decomposition Methods in Science and Engineering XVII*, pages 45–56. Springer, Berlin, 2008.

[Gin10] V. Ginting. An a posteriori analysis of multiscale operator decomposition. In G. Kreiss *et al.*, editor, *Numerical Mathematics and Advanced Applications 2009*. Springer Berlin Heidelberg, 2010.

[Gio99] V. Giovangigli. *Multicomponent Flow Modeling*. Birkhäuser Boston Inc., Boston, MA, 1999.

[Gje07] T. Gjesdal. Implicit–explicit methods based on strong stability preserving multistep time discretizations. *Appl. Numer. Math.*, 57(8):911–919, 2007.

[GKR01] M. Günther, A. Kværnø, and P. Rentrop. Multirate partitioned Runge-Kutta methods. *BIT Numer. Math.*, 41:504–514, 2001.

[GL61] A. Guillou and B. Lago. Domaine de stabilité associé aux formules d'intégration numérique d'équations différentielles à pas séparés et à pas liés. Recherche de formules à grands rayons de stabilité. *1er Cong. Assoc. Fran. Calcul, AFCAL, Grenoble*, pages 43–56, 1961.

[GLFE06] I. Garrido, B. Lee, G.E. Fladmark, and M.S. Espedal. Convergent iterative schemes for time parallelization. *Math. Comp.*, (26):1403–1428, 2006.

[Gok88] S.A. Gokoglu. Significance of vapor phase chemical reactions on cvd rates predicted by chemically frozen and local thermochemical equilibrium boundary layer theories. *J. Electrochem. Soc.*, 1988.

[GP00] A. Gerisch and H. Podhaisky. Splitting methods for the simulation of tumor angiogenesis models. In *Proceedings of the 16th IMACS World Congress*, pages 3–9522075, 2000.

[GPMD88] G. Goyal, P.J. Paul, H.S. Mukunda, and S.M. Deshpande. Time dependent operator-split and unsplit schemes for one dimensional premixed flames. *Combust. Sci. Technol.*, 60:167–189, 1988.

[Gra03] G.D. Graham. Tissue plasminogen activator for acute ischemic stroke. *Stroke*, 34:2847–2850, 2003.

[Grö67] W. Gröbner. *Die Liereihen und ihre Anwendungen*. VEB Deutscher Verlag der Wiss., Berlin 1960, 1967. 2nd Edition.

[GS69] A. Guillon and F.L. Soulé. La résolution numérique des problèmes différentiels aux conditions initiales par des méthodes de collocation. *RAIRO Anal. Numér. Ser. Rouge, v. R-3*, pages 17–44, 1969.

[GS94] P. Gray and S.K. Scott. *Chemical Oscillations and Instabilites*. Oxford Univ. Press, 1994.

[Gus94] K. Gustafsson. Control-theoretic techniques for stepsize selection in implicit Runge-Kutta methods. *ACM Trans. Math. Softw.*, 20:496–517, 1994.

[GV02] A. Gerisch and J.G. Verwer. Operator splitting and approximate factorization for taxis-diffusion-reaction models. *Appl. Numer. Math.*, 42(1-3):159–176, 2002.

[GV07] M. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.*, 29(2):556–578, 2007.

[GW84] C.W. Gear and D.R. Wells. Multirate linear multistep methods. *BIT Numer. Math.*, 24:484–502, 1984.

[Han78] A.J. Hansen. The extracellular potassium concentration in brain cortex following ischemia in hypo- and hyperglycemic rats. *Acta Physiologica Scandinavica*, 102(3):500–544, 1978.

[Har94a] A. Harten. Adaptive multiresolution schemes for shock computations. *J. Comput. Phys.*, 115:319–338, 1994.

[Har94b] A. Harten. Multiresolution representation and numerical algorithms: A brief review. *ICASE Rep. 94-59*, 1994.

[Har95] A. Harten. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Comm. Pure and Applied Math.*, 48:1305–1342, 1995.

[HH52] A.L. Hodgkin and A.F. Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.*, 117(4):500–544, 1952.

[Hin80] A.C. Hindmarsh. LSODE and LSODI, two new initial value ordinary differential equation solvers. *SIGNUM Newsl.*, 15:10–11, 1980.

[Hin83] A.C. Hindmarsh. ODEPACK, a systematized collection of ODE solvers. In *Scientific computing (Montreal, Que., 1982)*, pages 55–64. IMACS, New Brunswick, NJ, 1983.

[HK00] G.J.M. Hagelaar and G.M.W. Kroesen. Speeding up fluid models for gas discharges by implicit treatment of the electron energy source term. *J. Comp. Phys.*, 159(1):1–12, 2000.

[HLR88] E. Hairer, C. Lubich, and M. Roche. Error of Runge-Kutta methods for stiff problems studied via differential algebraic equations. *BIT Numer. Math.*, 28:678–700, 1988.

[HLW06] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration*. Springer-Verlag, Berlin, 2nd edition, 2006. Structure-Preserving Algorithms for Ordinary Differential Equations.

[HM10] N. Hovhannisyan and S. Müller. On the stability of fully adaptive multiscale schemes for conservation laws using approximate flux and source reconstruction strategies. *IMA J. of Numer. Anal.*, 30(4):1256–1295, 2010.

[HMSW99] P. Houston, J.A. Mackenzie, E. Süli, and G. Warnecke. A posteriori error analysis for numerical approximations of Friedrichs systems. *Numer. Math.*, 82:433–470, 1999.

[HNW87] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I.* Springer-Verlag, Berlin, 1987. Nonstiff Problems.

[HO09] E. Hansen and A. Ostermann. High order splitting methods for analytic semigroups exist. *BIT Numer. Math.*, 49:527–542, 2009.

[Hof76] E. Hofer. A partially implicit method for large stiff systems of ODEs with only few equations introducing small time-constants. *SIAM Journal on Numerical Analysis*, 13(5):645–663, 1976.

[HR07] W. Hundsdorfer and S.J. Ruuth. IMEX extensions of linear multistep methods with general monotonicity and boundedness properties. *J. Comput. Phys.*, 225(2):2016–2042, 2007.

[HV95] W. Hundsdorfer and J.G. Verwer. A note on splitting errors for advection-reaction equations. *Appl. Numer. Math.*, 18(1-3):191–199, 1995.

[HV03] W. Hundsdorfer and J.G. Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer-Verlag, Berlin, 2003.

[HW96] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer-Verlag, Berlin, 2nd edition, 1996. Stiff and Differential-Algebraic Problems.

[HW06] W. Heineken and G. Warnecke. Partitioning methods for reaction-diffusion problems. *Appl. Numer. Math.*, 56(7):981–1000, 2006.

[HWK06] R. Hornung, A. Wissink, and S. Kohn. Managing complex data and geometry in parallel structured AMR applications. *Engineering with Computers*, 22:181–195, 2006.

[IZ04] S. Isono and D. Zingg. A Runge-Kutta-Newton-Krylov algorithm for fourth-order implicit time marching applied to unsteady flows. *AIAA Paper 2004-0433*, 2004.

[Jac99] J.D. Jackson. *Classical Electrodynamics*. John Wiley and Sons, Inc., 3rd edition, 1999. Page 26, equation (1.5).

[JSW89] W. Jahnke, W.E. Skaggs, and A.T. Winfree. Chemical vortex dynamics in the Belousov-Zhabotinsky reaction and in the two-variable Oregonator model. *J. Phys. Chem.*, 93:740–749, 1989.

[KB95] S.R. Kohn and S.B. Baden. A parallel software infrastructure for structured adaptive mesh methods. In *Proc. of the Conf. on Supercomputing '95*, page 36, Los Alamitos, CA, USA, 1995. IEEE Computer Society.

[KC97] J. Kim and S.Y. Cho. Computation accuracy and efficiency of the time-splitting method in solving atmospheric transport-chemistry equations. *Atmos. Environ.*, 1997.

[KC03] C.A. Kennedy and M.H. Carpenter. Additive Runge-Kutta schemes for convection-diffusion-reaction equations. *Appl. Numer. Math.*, 44(1-2), 2003.

[KC07] T.E. Karakasidis and C.A. Charitidis. Multiscale modeling in nanomaterials science. *Materials Science and Engineering: C*, 27(5-8):1082–1089, 2007.

[KDK+07] C.R. Kleijn, R. Dorsman, K.J. Kuijlaars, M. Okkerse, and H. van Santen. Multiscale modeling of chemical vapor deposition processes for thin film technology. *Journal of Crystal Growth*, 303(1):362–380, 2007.

[KDOO05] H. Karimabadi, J. Driscoll, Y.A. Omelchenko, and N. Omidi. A new asynchronous methodology for modeling of physical systems: Breaking the curse of courant condition. *J. Comp. Phys.*, 205(2):755–775, 2005.

[Kee87] J.P. Keener. Propagation and its failure in coupled systems of discrete excitable cells. *SIAM J. Appl. Math.*, 47(3):556–572, 1987.

[KGM+98] R.J. Kee, J.F. Grcar, J.A. Miller, E. Meeks, and M. Smooke. PREMIX Users Manual. *Reaction Design, San Diego, CA (www.ReactionDesign.com)*, 1998.

[KK04] D.A. Knoll and D.E. Keyes. Jacobian-free Newton-Krylov methods: A survey of approaches and applications. *J. Comput. Phys.*, 193(2):357–397, 2004.

[KKMS92] I.A. Kossyi, A. Yu Kostinsky, A.A. Matveyev, and V.P. Silakov. Kinetic scheme of the non-equilibrium discharge in nitrogen-oxygen mixtures. *Plasma Sources Sci. Technol.*, 1(3):207–220, 1992.

[KN78] R.P. Kraig and C. Nicholson. Extracellular ionic variations during spreading depression. *Neuroscience,*, 3(11):1045–1059, 1978.

[KNW99] O.M. Knio, H.N. Najm, and P.S. Wyckoff. A semi-implicit numerical scheme for reacting flow. II. Stiff, operator-split formulation. *J. Comput. Phys.*, 154:482–467, 1999.

[KO00] D. Kröner and M. Ohlberger. A posteriori error estimates for upwind finite volume schemes for nonlinear conservation laws in multi dimensions. *Math. Comput.*, 69:25–39, 2000.

[KPP37] A.N. Kolmogoroff, I.G. Petrovsky, and N.S. Piscounoff. Etude de l'équation de la diffusion avec croissance de la quantité de matière et son application a un problème biologique. *Bulletin de l'Université d'état Moscou, Série Internationale Section A Mathématiques et Mécanique*, 1:1–25, 1937.

[KRM80] R.J. Kee, F.M. Rupley, and J.A. Miller. CHEMKIN-II: A Fortran chemical kinetics package for the analysis of gas-phase chemical kinetics. *Tech. Rep. SAND89-8009, Sandia National Lab., Livermore, CA*, 1980.

[KrO04] R. Kozlov, A. Kværnø, and B. Owren. The behaviour of the local error in splitting methods applied to stiff problems. *J. Comput. Phys.*, 195(2):576–593, 2004.

[KT11] O. Koch and M. Thalhammer. Embedded exponential operator splitting methods for the time integration of nonlinear evolution equations. *Preprint, available at http://techmath.uibk.ac.at/mecht/research/Papers/Preprint4.pdf*, 2011.

[Kul97] A.A. Kulikovsky. Positive streamer between parallel plate electrodes in atmospheric pressure air. *J. Phys. D: Appl. Phys.*, 30:441–450, 1997.

[Kul98] A.A. Kulikovsky. Positive streamer in a weak field in air: A moving avalanche-to-streamer transition. *Phys. Rev. E*, 57(6):7066–7074, 1998.

[Kul00] A.A. Kulikovsky. The role of photoionization in positive streamer dynamics. *J. Phys. D: Appl. Phys.*, 33:1514–1524, 2000.

[KWS00] H. Kager, W.J. Wadman, and G.G. Somjen. Simulated seizures and spreading depression in a neuron model incorporating interstitial space and ion concentrations. *J. Neurophysiol.*, 84(1):495–512, 2000.

[LC89] A. Laverdant and S. Candel. Computation of diffusion and premixed flames rolled up in vortex structures. *J. Propulsion and Power*, 5:134–143, 1989.

[Leb89] V.I. Lebedev. Explicit difference schemes with time-variable steps for solving stiff systems of equations. *Sov. J. Numer. Anal. Math. Modelling*, 4:111–135, 1989.

[Leb93a] V.I. Lebedev. A new method for determining the zeros of polynomials of least deviation on a segment with weight and subject to additional conditions. part I. *Russian J. Numer. Anal. Math. Modelling*, 8:195–222, 1993.

[Leb93b] V.I. Lebedev. A new method for determining the zeros of polynomials of least deviation on a segment with weight and subject to additional conditions. part II. *Russian J. Numer. Anal. Math. Modelling*, 8:397–426, 1993.

[Leb94] V.I. Lebedev. How to solve stiff systems of differential equations by explicit methods. In *Numerical Methods and Applications*, pages 45–80. Boca Raton:CRC Press, 1994.

[Leb00] V.I. Lebedev. Explicit difference schemes for solving stiff problems with a complex or separable spectrum. *Comput. Math. and Math. Phys.*, 40(12):1729–1740, 2000.

[LEH08] A. Luque, U. Ebert, and W. Hundsdorfer. Interaction of streamer discharges in air and other oxygen-nitrogen mixtures. *Phys. Rev. Lett.*, 101(7):075005, 2008.

[LeV92] R.J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser, 2nd edition, 1992.

[LeV02] R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[LG94] S.H. Lam and D.A. Goussis. The CSP method for simplifying kinetics. *International Journal of Chemical Kinetics*, 26(4):461–486, 1994.

[LJAC+09] D. Lloyd-Jones, R. Adams, M. Carnethon, G. De Simone, T.B. Ferguson, K. Flegal, E. Ford, K. Furie, A. Go, K. Greenlund, N. Haase, S. Hailpern, M. Ho, V. Howard, B. Kissela, S. Kittner, D. Lackland, L. Lisabeth, A. Marelli, M. McDermott, J. Meigs, D. Mozaffarian, G. Nichol, C. O'Donnell, V. Roger, W. Rosamond, R. Sacco, P. Sorlie, R. Stafford, J. Steinberger, T. Thom, S. Wasserthiel-Smoller, N. Wong, J. Wylie-Rosett, and Y. Hong. Heart disease and stroke statistics–2009 update: A report from the American Heart Association Statistics Committee and Stroke Statistics Subcommittee. *Circulation*, 119(3):e21–181, 2009.

[LL09] T. Lu and C.K. Law. Toward accommodating realistic fuel chemistry in large-scale computations. *Progress in Energy and Combustion Science*, 35(2):192–215, 2009.

[LLM96] B.P. Leonard, A.P. Lock, and M.K. MacVean. Conservative explicit unrestricted-time-step multidimensional constancy-preserving advection schemes. *Monthly Weather Review*, 124:2588–2606, 1996.

[LM98] V.I. Lebedev and A.A. Medovikov. An explicit method of the second order of accuracy for solving stiff systems of ordinary differential equations. *Russian Izv. Vyssh. Uchebn. Zaved. Mat.*, 9:55–63, 1998.

[LMT01] J.L. Lions, Y. Maday, and G. Turinici. Résolution d'EDP par un schéma en temps "pararéel". *C. R. Acad. Sci. Paris Sér. I Math.*, 332(7):661–668, 2001.

[Log02] A. Logg. Multi-adaptive Galerkin methods for ODEs I. *SIAM J. Sci. Comput.*, 24:1879–1902, 2002.

[Log03] A. Logg. Multi-adaptive Galerkin methods for ODEs II: Implementation and applications. *SIAM J. Sci. Comput.*, 25:1119–1141, 2003.

[LRJ92] D.R. Lemieux, F.A. Roberge, and D. Joly. Modeling the dynamic features of the electrogenic na,k pump of cardiac cells. *J. Theor. Biol.*, 3:335–358, 1992.

[LT90] J. Liandrat and P. Tchamitchian. Resolution of the 1D regularized Burgers' equation using a spatial wavelet approximation. *ICASE Rep. 90-83*, 1990.

[Lub08] C. Lubich. On splitting methods for Schrödinger-Poisson and cubic nonlinear Schrödinger equations. *Math. Comp.*, 77(264):2141–2153, 2008.

[LV99] D. Lanser and J.G. Verwer. Analysis of operator splitting for advection-diffusion-reaction problems from air pollution modelling. *J. Comput. Appl. Math.*, 111(1-2):201–216, 1999.

[LW60] D. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure and Applied Math.*, 13:217–237, 1960.

[LW03] R. Liska and B. Wendroff. Comparison of several difference schemes on 1D and 2D test problems for the Euler equations. *SIAM J. Sci. Comput.*, 25:995–1017, 2003.

[LZ06] H. Liu and J. Zou. Some new additive Runge-Kutta methods and their applications. *J. Comput. Appl. Math.*, 190(1-2):74–98, 2006.

[Mal89] S. Mallat. Multiresolution approximation and wavelets orthonormal bases of $L^2(R)$. *Trans. Amer. Math. Soc.*, 315:69–87, 1989.

[Mar68] G.I. Marchuk. Some application of splitting-up methods to the solution of mathematical physics problems. *Applications of Mathematics*, 13(2):103–132, 1968.

[Mar75] G.I. Marchuk. *Methods of Numerical Mathematics.* Appl. Math. Springer, New York, NY, 1975. Trans. from the Russian.

[Mar90] G.I. Marchuk. Splitting and alternating direction methods. In *Handbook of Numerical Analysis, Vol. I*, pages 197–462. North-Holland, Amsterdam, 1990.

[Mar09] A.A. Markov. Multi scale numerical simulation of dispersed reacting flow, with application to chemical vapor deposition of alumina. In H. Deconinck and E. Dick, editors, *Computational Fluid Dynamics 2006*, pages 753–758. Springer Berlin Heidelberg, 2009.

[Mas02] M. Massot. 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, 2002.

[Med98] A. Medovikov. High order explicit methods for parabolic equations. *BIT Numer. Math.*, 38:372–390, 1998.

[Mey90] Y. Meyer. *Ondelettes et Opérateurs.* Hermann, Paris, 1990.

[MFNN00] H. Martins-Ferreira, M. Nedergaard, and C. Nicholson. Perspectives on spreading depression. *Brain Research Reviews*, 31(1):215–234, 2000.

[MGS82] G.J. McRae, W.R. Goodin, and J.H. Seinfeld. Numerical solution of the atmospheric diffusion equation for chemically reacting flows. *J. Comput. Phys.*, 45(1):1–42, 1982.

[MHE06] C. Montijn, W. Hundsdorfer, and U. Ebert. An adaptive grid refinement strategy for the simulation of negative streamers. *J. Comput. Phys.*, 219(2):801–835, 2006.

[Mil08] D.J. Miller. *New Methods in Computational Systems Biology*. PhD thesis, Faculty of Drexel University, 2008.

[ML97] R. Morrow and J.J. Lowke. Streamer propagation in air. *J. Phys. D: Appl. Phys.*, 30:614–627, 1997.

[MLI10] M.A. Moskowitz, E.H. Lo, and C. Iadecola. The science of stroke: Mechanisms in search of treatments. *Neuron*, 67(2):181–198, 2010.

[MOM+00] P. MacNeice, K.M. Olson, C. Mobarry, R. de Fainchtein, and C. Packer. PARAMESH: A parallel adaptive mesh refinement community toolkit. *Computer Physics Communications*, 126(3):330–354, 2000.

[Mor85] R. Morrow. Theory of negative corona in oxygen. *Phys. Rev. A*, 32(3):1799–1809, 1985.

[MP92] U. Maas and S.B. Pope. Simplifying chemical kinetics: Intrinsic low-dimensional manifolds in composition space. *Combust. and Flame*, 88(3-4):239–264, 1992.

[MQ02] R.I. McLachlan and R.W. Quispel. Splitting methods. *Acta Numerica*, 11:341–434, 2002.

[MR92] Y. Maday and J.C. Ravel. Adaptivité par ondelettes: Condition aux limites et dimensions supérieures. *Tech. Rep., Université Pierre et Marie Curie, Lab. D'Analyse Numérique*, 1992.

[MS07] S. Müller and Y. Stiriba. Fully adaptive multiscale schemes for conservation laws employing locally varying time stepping. *J. Scientific Computing*, 30:493–531, 2007.

[MT02] Y. Maday and G. Turinici. A parareal in time procedure for the control of partial differential equations. *C. R. Math. Acad. Sci. Paris Sér. I Math.*, (335):387–391, 2002.

[MT03] Y. Maday and G. Turinici. The parareal in time iterative solver: A further direction to parallel implementation. In *Proceedings of the 15th International Domain Decomposition Conference, Lect. Notes Comput. Sci. Eng. 40*, pages 441–448. Springer, Berlin, 2003.

[MTD90] C.J. McBain, S.F. Traynelis, and R. Dingledine. Regional variation of extracellular space in the hippocampus. *Science*, 249 (4969), 1990.

[Mül03] S. Müller. *Adaptive Multiscale Schemes for Conservation Laws*, volume 27. Springer-Verlag, 2003.

[NAK08] D.S. Nikandrov, R.R. Arslanbekov, and V.I. Kolobov. Streamer simulations with dynamically adaptive cartesian mesh. *IEEE Trans. Plasma Sci.*, 36(4):932–933, 2008.

[NBS07] M. Noskov, M. Benzi, and M.D. Smooke. An implicit compact scheme solver for two-dimensional multicomponent flows. *Computers & Fluids*, 36(2):376–397, 2007.

[NK05] H.N. Najm and O.M. Knio. Modeling Low Mach number reacting flow with detailed chemistry and transport. *J. Scientific Computing*, 25(1/2):263–287, 2005.

[NS05] M. Noskov and M.D. Smooke. An implicit compact scheme solver with application to chemically reacting flows. *J. Comput. Phys.*, 203(2):700–730, 2005.

[NWK98] H.N. Najm, P.S. Wyckoff, and O.M. Knio. A semi-implicit numerical scheme for reacting flow. I. Stiff chemistry. *J. Comput. Phys.*, 143:381–402, 1998.

[OB01] E.S. Oran and J.P. Boris. *Numerical Simulation of Reacting Flows.* Cambridge University Press, 2001. Second Edition.

[OK06] Y.A. Omelchenko and H. Karimabadi. Self-adaptive time integration of flux-conservative equations with sources. *J. Comp. Phys.*, 216(1):179–194, 2006.

[OOZ01] T. Ostromsky, W. Owczarz, and Z. Zlatev. Computational challenges in large-scale air pollution modelling. In *Proceedings of the 15th International Conference on Supercomputing*, ICS '01, pages 407–418, 2001.

[OSM+08] D.F. Opaits, M.N. Shneider, R. B. Miles, A.V. Likhanskii, and S.O. Macheret. Surface charge in dielectric barrier discharge plasma actuators. *Phys. Plasmas*, 15(7):073505, 2008.

[Pan05] S.V. Pancheshnyi. Role of electronegative gas admixtures in streamer start, propagation and branching phenomena. *Plasma Sources Sci. Technol.*, 14(4):645–653, 2005.

[Pet09] N. Peters. Multiscale combustion and turbulence. *Proc. Combust. Inst.*, 32:1–25, 2009.

[PGL+06] G. Pilla, D. Galley, D. Lacoste, F. Lacas, D. Veynante, and C.O. Laux. Stabilization of a turbulent premixed flame using a nanosecond repetitively pulsed plasma. *IEEE Trans. Plasma Sci.*, 34(6, Part 1):2471–2477, 2006.

[PHB+98] R.B. Pember, L.H. Howell, J.B. Bell, P. Colella, W.Y. Crutchfield, W.A. Fiveland, and J.P. Jessee. An adaptive projection method for unsteady, low-Mach number combustion. *Combust. Sci. Technol.*, 140(1-6):123–168, 1998.

[Pit06] H. Pitsch. Large-Eddy Simulation of turbulent combustion. *Annual Review of Fluid Mechanics*, 38(1):453–482, 2006.

[PLL10] D.Z. Pai, D.A. Lacoste, and C.O. Laux. Transitions between corona, glow, and spark regimes of nanosecond repetitively pulsed discharges in air at atmospheric pressure. *J. Appl. Phys.*, 107(9):093303, 2010.

[PMG11] L. Papageorgiou, A.C. Metaxas, and G.E. Georghiou. Three-dimensional numerical modelling of gas discharges at atmospheric pressure incorporating photoionization phenomena. *J. Phys. D: Appl. Phys.*, 44(4):045203, 2011.

[Pop97] S.B. Pope. Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation. *Combust. Theory Modelling*, 1(1):41–63, 1997.

[Pos05] M. Postel. Approximations multiéchelles. In *Neuvième École Mécanique des Fluides Numérique, Roscoff*, pages 1–59. Université Pierre et Marie Curie, Lab. Jacques-Louis Lions, 2005. Lesson Notes, available at http://www.ann.jussieu.fr/~postel.

[PR74] A. Prothero and A. Robinson. On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations. *Math. Comp.*, 28(125):145–162, 1974.

[PR09] S. Pope and Z. Ren. Efficient implementation of chemistry in computational combustion. *Flow, Turbulence and Combustion*, 82:437–453, 2009.

[PSCB08] S. Pancheshnyi, P. Segur, J. Capeillere, and A. Bourdon. Numerical simulation of filamentary discharges with parallel adaptive mesh refinement. *J. Comp. Phys.*, 227(13):6574–6590, 2008.

[PSS01] S.V. Pancheshnyi, S.M. Starikovskaia, and A.Y. Starikovskii. Role of photoionization processes in propagation of cathode-directed streamer. *J. Phys. D: Appl. Phys.*, 34:105–115, 2001.

[PV05] T. Poinsot and D. Veynante. *Theoretical and Numerical Combustion*. Edwards, 2nd edition edition, 2005.

[Rad80] R. Radau. Étude sur les formules d'approximation qui servent à calculer la valeur numérique d'une intégrale definie. *J. Math. Pures Appl.*, 6:283–336, 1880.

[RBL+00] C.A. Rendleman, V.E. Beckner, M. Lijewski, W. Crutchfield, and J.B. Bell. Parallelization of structured, hierarchical adaptive mesh refinement algorithms. *Computing and Visualization in Science*, 3:147–157, 2000.

[RHMD+09] C. Rosso, N. Hevia-Montiel, S. Deltour, E. Bardinet, D. Dormont, S. Crozier, S. Baillet, and Y. Samson. Prediction of infarct growth based on apparent diffusion coefficients: penumbral assessment without intravenous contrast material. *Radiology*, 95(3):450–458, 2009.

[ROA00] D.J. Rossi, T. Oshima, and D. Attwell. Glutamate release in severe brain ischaemia is mainly by reversed uptake. *Nature*, 403:316–321, 2000.

[ROS+01] L. Røhl, L. Østergaard, C.Z. Simonsen, P. Vestergaard-Poulsen, G. Andersen, M. Sakoh, D. Le Bihan, and C. Gyldensted. Viability thresholds of ischemic penumbra of hyperacute stroke defined by perfusion-weighted MRI and apparent diffusion coefficient. *Stroke*, 32(5):1140–1146, 2001.

[RP08] Z. Ren and S.B. Pope. Second-order splitting schemes for a class of reactive systems. *J. Comput. Phys.*, 227(17):8165–8176, 2008.

[RRGR98] K. Revett, E. Ruppin, S. Goodall, and J.A. Reggia. Spreading depression in focal ischemia: a computational study. *J. Cereb. Blood Flow & Metab.*, 18(9):998–1007, 1998.

[RRTC99] P.-H. Renard, J.C. Rolon, D. Thévenin, and S. Candel. Investigations of heat release, extinction, and time evolution of the flame surface, for a nonpremixed flame interacting with a vortex. *Combust. and Flame*, 117(1-2):189–205, 1999.

[RS02] O. Roussel and K. Schneider. A fully adaptive multiresolution scheme for 3D reaction-diffusion equations. In R. Herbin and D. Kröner, editors, *Finite Volumes for Complex Applications*, volume 3, pages 833–840. Hermes Penton Science, 2002.

[RS05a] D.L. Ropp and J.N. Shadid. Stability of operator splitting methods for systems with indefinite operators: Reaction-diffusion systems. *J. Comput. Phys.*, 203(2):449–466, 2005.

[RS05b] O. Roussel and K. Schneider. An adaptive multiresolution method for combustion problems: Application to flame ball-vortex interaction. *Computers & Fluids*, 34(7):817–831, 2005.

[RS06] O. Roussel and K. Schneider. Numerical study of thermodiffusive flame structures interacting with adiabatic walls using an adaptive multiresolution scheme. *Combust. Theory Modelling*, 10(2):273–288, 2006.

[RS09] D.L. Ropp and J.N. Shadid. Stability of operator splitting methods for systems with indefinite operators: Advection-diffusion-reaction systems. *J. Comput. Phys.*, 228(9):3508–3516, 2009.

[RSO04] D.L. Ropp, J.N. Shadid, and C.C. Ober. Studies of the accuracy of time integration methods for reaction-diffusion equations. *J. Comput. Phys.*, 194(2):544–574, 2004.

[RSTB03] O. Roussel, K. Schneider, A. Tsigulin, and H. Bockhorn. A conservative fully adaptive multiresolution algorithm for parabolic PDEs. *J. Comput. Phys.*, 188(2):493–523, 2003.

[RTRC00] P.-H. Renard, D. Thévenin, J.C. Rolon, and S. Candel. Dynamics of flame/vortex interactions. *Progress in Energy and Combustion Science*, 26(3):225–282, 2000.

[Sau60] V.K. Saul'ev. Integration of parabolic type equations with the method of nets. *Moscow, Fizmatgiz*, 1960. In Russian.

[SBP00] B. Sportisse, G. Bencteux, and P. Plion. Method of Lines versus Operator Splitting for reaction–diffusiuon systems with fast chemistry. *Environmental Modelling & Software*, 15(6-7):673–679, 2000.

[Sch02] M. Schatzman. Toward non commutative numerical analysis: High order integration in time. *J. Scientific Computing*, 17(1-3):107–125, 2002.

[SD00] B. Sportisse and R. Djouad. Reduction of chemical kinetics in air pollution modeling. *J. Comput. Phys.*, 164(2):354–376, 2000.

[SDJ+05] H.K. Shin, A.K. Dunn, P.B. Jones, D.A. Boas, M.A. Moskowitz, and C. Ayata. Vasoconstrictive neurovascular coupling during focal ischemic depolarizations. *J. Cereb. Blood Flow & Metab.*, 26(8):1018–1030, 2005.

[SGP07] G. Staffelbach, L.Y.M. Gicquel, and T. Poinsot. Highly parallel large eddy simulations of multiburner configurations in industrial gas turbines. In S.C. Kassinos *et al.*, editor, *Complex Effects in Large Eddy Simulations*, volume 56 of *Lecture Notes in Computational Science and Engineering*, pages 325–336. Springer Berlin Heidelberg, 2007.

[SH77] L.F. Shampine and K.L. Hiebert. Detecting stiffness with the Fehlberg (4,5) formulas. *Computers & Mathematics with Applications*, 3(1):41–46, 1977.

[SH97] A. Suresh and H.T. Huynh. Accurate monotonicity-preserving schemes with Runge-Kutta time stepping. *J. Comput. Phys.*, 136:83–99, 1997.

[Sha77] L.F. Shampine. Stiffness and nonstiff differential equation solvers, II: Detecting stiffness with Runge-Kutta methods. *ACM Trans. Math. Softw.*, 3:44–53, 1977.

[Sha87] L.F. Shampine. Control of step size and order in extrapolation codes. *J. Comput. Appl. Math.*, 18(1):3–16, 1987.

[Sha01] B.E. Shapiro. Osmotic forces and gap junctions in spreading depression: A computational model. *J. Comput. Neurosci.*, 10:877–896, 2001.

[SHH94] T. Sonar, V. Hannemann, and D. Hempel. Dynamic adaptivity and residual control in unsteady compressible flow computation. *Mathematical and Computer Modelling*, 20(10-11):201–213, 1994.

[SHV07] V. Savcenco, W. Hundsdorfer, and J.G. Verwer. A multirate time stepping strategy for stiff ordinary differential equations. *BIT Numer. Math.*, 47:137–155, 2007.

[SJWO11] Y. Shi, W.H. Green Jr., H.-W. Wong, and O.O. Oluwole. Redesigning combustion modeling algorithms for the Graphics Processing Unit (GPU): Chemical kinetic rate evaluation and ordinary differential equation integration. *Combust. and Flame*, 158(5):836–847, 2011.

[SLGS03] D.A. Schwer, P. Lu, W.H. Green, and V. Semião. A consistent-splitting approach to computing stiff steady-state reacting flows with adaptive chemistry. *Combust. Theory Modelling*, 7(2):383–399, 2003.

[SM10] V. Savcenco and R.M.M. Mattheij. Multirate numerical integration for stiff ODEs. In H.-G. Bock *et al.*, editor, *Progress in Industrial Mathematics at ECMI 2008*, volume 15 of *Mathematics in Industry*, pages 327–332. Springer Berlin Heidelberg, 2010.

[SMN10] C. Steiner, S. Müller, and S. Noelle. Adaptive timestep control for nonstationary solutions of the Euler equations. *SIAM J. Sci. Comput.*, 32(3):1617–1651, 2010.

[Smo83] M.D. Smooke. 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, 1983.

[SN08] C. Steiner and S. Noelle. On adaptive timestepping for weakly instationary solutions of hyperbolic conservation laws via adjoint error control. *Comm. Numer. Meth. Eng*, 2008.

[SOG+01] M. Sakoh, L. Østergaard, A. Gjedde, L. Røhl, P. Vestergaard-Poulsen, D.F. Smith, D. Le Bihan, S. Sakaki, and C. Gyldensted. Prediction of tissue survival after middle cerebral artery occlusion based on changes in the apparent diffusion of water. *J. Neurosurg.*, 95(3):450–458, 2001.

[Som01] G.G. Somjen. Mechanisms of spreading depression and hypoxic spreading depression-like depolarization. *Physiol. Rev.*, 81(3):96–100, 2001.

[SP04] M.A. Singer and S.B. Pope. Exploiting ISAT to solve the reaction–diffusion equation. *Combust. Theory Modelling*, 8(2):361–383, 2004.

[SPN06a] M.A. Singer, S.B. Pope, and H.N. Najm. Modeling unsteady reacting flow with operator splitting and ISAT. *Combust. and Flame*, 147(1-2):150–162, 2006.

[SPN06b] M.A. Singer, S.B. Pope, and H.N. Najm. Operator-splitting with ISAT to model reacting flow with detailed chemistry. *Combust. Theory Modelling*, 10(2):199–217, 2006.

[Spo99] B. Sportisse. *Contribution à la modélisation des écoulements réactifs: Réduction des modèles de cinétique chimique et simulation de la pollution atmosphérique.* PhD thesis, Ecole Polytechnique, 1999.

[Spo00] B. Sportisse. An analysis of operator splitting techniques in the stiff case. *J. Comput. Phys.*, 161(1):140–168, 2000.

[Spo07] B. Sportisse. A review of current issues in air pollution modeling and simulation. *Computational Geosciences*, 11:159–181, 2007.

[SR03] G.A. Staff and E.M. Rønquist. Stability of the parareal algorithm. In *Proceedings of the 15th International Domain Decomposition Conference, Lect. Notes Comput. Sci. Eng. 40*, pages 449–456. Springer, Berlin, 2003.

[SSC94] J.M. Sanz-Serna and M.P. Calvo. *Numerical Hamiltonian Problems.* Chapman & Hall, London, 1994.

[SSGP08] G. Staffelbach, J. Senoner, L. Gicquel, and T. Poinsot. Large eddy simulation of combustion on massively parallel machines. In J. Palma *et al.*, editor, *High Performance Computing for Computational Science - VECPAR 2008*, volume 5336 of *Lecture Notes in Computer Science*, pages 444–464. Springer Berlin Heidelberg, 2008.

[SSV97] B.P. Sommeijer, L.F. Shampine, and J.G. Verwer. RKC: An explicit solver for parabolic PDEs. *J. Comput. Appl. Math.*, 88(2):315–326, 1997.

[SSV06] L.F. Shampine, B.P. Sommeijer, and J.G. Verwer. IRKC: An IMEX solver for stiff diffusion-reaction PDEs. *J. Comput. Appl. Math.*, 196(2):485–497, 2006.

[Ste08] C. Steiner. *Adaptive Timestepping for Conservation Laws via Adjoint Error Representation.* PhD thesis, RWTH Aachen University, 2008.

[Str63] G. Strang. Accurate partial difference methods. I. Linear Cauchy problems. *Arch. Ration. Mech. Anal.*, 12:392–402, 1963.

[Str68] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506–517, 1968.

[Sül99] E. Süli. A posteriori error analysis and adaptivity for finite element approximations of hyperbolic problems. In *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws (Freiburg/Littenweiler, 1997)*, volume 5 of *Lecture Notes in Computational Science and Engineering*, pages 123–194. Springer Berlin, 1999.

[Sun96] P. Sun. A pseudo non-time splitting method in air quality modeling. *J. Comput. Phys.*, 1996.

[SV80] B.P. Sommeijer and J.G. Verwer. *A Performance Evaluation of a Class of Runge-Kutta-Chebyshev Methods for Solving Semidiscrete Parabolic Differential Equations.* Afdeling Numerieke Wiskunde [Department of Numerical Mathematics], 91. Mathematisch Centrum, Amsterdam, 1980.

[SV10] K. Schneider and O.V. Vasilyev. Wavelet methods in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 42:473–503, 2010.

[SVdZ+98] E.J. Spee, J.G. Verwer, P.M. de Zeeuw, J.G. Blom, and W. Hundsdorfer. A numerical study for global atmospheric transport-chemistry problems. *Mathematics and Computers in Simulation*, 48(2):177–204, 1998.

[Swe95] W. Sweldens. Lifting scheme: A new philosophy in biorthogonal wavelet constructions. In A.F. Laine *et al.*, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 2569 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 68–79, 1995.

[Swe98] W. Sweldens. The lifting scheme: A construction of second generation wavelets. *SIAM J. Math. Anal.*, 29(2):511–546, 1998.

[TBS11] L. Tosatto, B.A.V. Bennett, and M.D. Smooke. Parallelization strategies for an implicit Newton-based reactive flow solver. *Combust. Theory Modelling*, 15(4):455–486, 2011.

[TC95] D. Thévenin and S. Candel. Ignition dynamics of a diffusion flame rolled up in a vortex. *Phys. Fluids*, 7(2):434–445, 1995.

[TD11] C. Tenaud and M. Duarte. Tutorials on adaptive multiresolution for mesh refinement applied to fluid dynamics and reactive media problems. *ESAIM: Proc.*, 34:184–239, 2011.

[Tém69a] R. Témam. 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, 1969.

[Tém69b] R. Témam. 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, 1969.

[Tey02] R. Teyssier. Cosmology hydrodynamics with adaptive mesh refinement. A new high resolution code called RAMSES. *Astronomy & Astrophysics*, 385:337–364, 2002.

[TGHB00] A.S. Tomlin, S. Ghorai, G. Hart, and M. Berzins. 3-D multi-scale air pollution modelling using adaptive unstructured meshes. *Environmental Modelling & Software*, 15(6-7):681–692, 2000.

[Tha08] M. Thalhammer. High-order exponential operator splitting methods for time-dependent Schrödinger equations. *SIAM J. Numer. Anal.*, 46(4):2022–2038, 2008.

[Thé92] D. Thévenin. Dynamique de l'allumage de flammes de diffusion dans des écoulements cisaillés - étude théorique et numérique, 1992.

[TK04] J.A. Trangenstein and C. Kim. Operator splitting and adaptive mesh refinement for the Luo-Rudy I model. *J. Comput. Phys.*, 196(2):645–679, 2004.

[Tro59] H.F. Trotter. On the product of semi-groups of operators. *Proc. Am. Math. Soc.*, 10:545–551, 1959.

[TRR+96] D. Thévenin, J.C. Rolon, P.-H. Renard, D.W. Kendrick, D. Veynante, and S. Candel. Structure of a non-premixed flame interacting with counterrotating vortices. *Symposium (International) on Combustion*, 26(1):1079–1086, 1996.

[TSA00] J.A. Trangenstein, K. Skouibine, and W.K. Allard. Operator splitting and adaptive mesh refinement for the Fitzhugh-Nagumo problem. *In press, available atn http://www.math.duke.edu/~johnt/fhn.ps*, 2000.

[TSL00] R. Tyson, L.G. Stern, and R.J. LeVeque. Fractional step methods applied to a chemotaxis model. *Journal of Mathematical Biology*, 41:455–475, 2000.

[TVS85] A.N. Tikhonov, A.B. Vasil'eva, and A.G. Sveshnikov. *Differential Equations.* Springer-Verlag, Berlin, 1985.

[UBRT07] T. Unfer, J.-P. Boeuf, F. Rogier, and F. Thivet. An asynchronous scheme with local time stepping for multi-scale transport problems: Application to gas discharges. *J. Comp. Phys.*, 227(2):898–918, 2007.

[UBRT10] T. Unfer, J.-P. Boeuf, F. Rogier, and F. Thivet. Multi-scale gas discharge simulations using asynchronous adaptive mesh refinement. *Computer Physics Communications*, 181(2):247–258, 2010.

[Var74] V.S. Varadarajan. *Lie Groups, Lie Algebras and their Representations.* Prentice-Hall, Englewood Cliffs, New Jersey, 1974.

[Vas03] O.V. Vasilyev. Solving multi-dimensional evolution problems with localized structures using second generation wavelets. *International Journal of Computational Fluid Dynamics*, 17:151–168, 2003.

[VB00] O.V. Vasilyev and C. Bowman. Second-generation wavelet collocation method for the solution of partial differential equations. *J. Comput. Phys.*, 165(2):660–693, 2000.

[VBvdS+94] H.B. Verheul, R. Balazs, J.W. Berkelbach van der Sprenkel, C.A.F. Tulleken, K. Nicolay, K.S. Tamminga, and M. van Lookeren Campagne. Comparison of diffusion-weighted MRI with changes in cell volume in a rat model of brain injury. *NMR Biomed.*, 7(1–2):96–100, 1994.

[VBvLS96] J.G. Verwer, J.G. Blom, M. van Loon, and E.J. Spee. A comparison of stiff ODE solvers for atmospheric chemistry problems. *Atmospheric Environment*, 30(1):49–58, 1996.

[vdH77] P.J. van der Houwen. *Construction of Integration Formulas for Initial Value Problems.* North-Holland Pub.Co., 1977.

[vdHS80] P.J. van der Houwen and B.P. Sommeijer. On the internal stability of explicit, m-stage Runge-Kutta methods for large m-values. *Z. Angew. Math. Mech.*, 60(10):479–485, 1980.

[Ver96] J.G. Verwer. Explicit Runge-Kutta methods for parabolic partial differential equations. *Appl. Numer. Math.*, 22(1-3):359–379, 1996.

[Vey09] D. Veynante. Large eddy simulations of turbulent combustion. In M. Deville *et al.*, editor, *Turbulence and Interactions*, volume 105 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, pages 113–138. Springer Berlin Heidelberg, 2009.

[VG01] M. Valorani and D.A. Goussis. Explicit time-scale splitting algorithm for stiff problems: Auto-ignition of gaseous mixtures behind a steady shock. *J. Comput. Phys.*, 169(1):44–79, 2001.

[VHS90] J.G. Verwer, W.H. Hundsdorfer, and B.P. Sommeijer. Convergence properties of the Runge-Kutta-Chebyshev method. *Numer. Math.*, 57:157–178, 1990.

[VM99] A. Voss and S. Müller. A manual for the template class library *igpm_t_lib. IGPM-Rep. 197, RWTH Aachen*, 1999.

[VP96] O.V. Vasilyev and S. Paolucci. A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain. *J. Comput. Phys.*, 125(2):498–512, 1996.

[VP97] O.V. Vasilyev and S. Paolucci. A fast adaptive wavelet collocation algorithm for multidimensional PDEs. *J. Comput. Phys.*, 138(1):16–56, 1997.

[VPB94] P.A. Vitello, B.M. Penetrante, and J.N. Bardsley. Simulation of negative-streamer dynamics in nitrogen. *Phys. Rev. E*, 49:5574–5598, 1994.

[VPS95] O.V. Vasilyev, S. Paolucci, and M. Sen. A multilevel wavelet collocation method for solving partial differential equations in a finite domain. *J. Comput. Phys.*, 120(1):33–47, 1995.

[VS98] J.G. Verwer and B. Sportisse. Note on operator splitting in a stiff linear case. *Rep. MAS-R9830*, 1998.

[VS04] J.G. Verwer and B.P. Sommeijer. An implicit-explicit Runge-Kutta-Chebyshev scheme for diffusion-reaction equations. *SIAM J. Sci. Comput.*, 25(5):1824–1835, 2004.

[VSBH99] J.G. Verwer, E.J. Spee, J.G. Blom, and W. Hundsdorfer. A second-order Rosenbrock method applied to photochemical dispersion problems. *SIAM J. Sci. Comput.*, 20:1456–1480, 1999.

[VSH04] J.G. Verwer, B.P. Sommeijer, and W. Hundsdorfer. RKC time-stepping for advection-diffusion-reaction problems. *J. Comput. Phys.*, 201(1):61–79, 2004.

[VSHK93] P.L.G. Ventzek, T.J. Sommerer, R.J. Hoekstra, and M.J. Kushner. Two-dimensional hybrid model of inductively coupled plasma sources for etching. *Appl. Phys. Lett.*, 63(5):605–607, 1993.

[vV00] E.M. van Veldhuizen, editor. *Electrical Discharges for Environmental Purposes: Fundamentals and Applications.* Nova Science, New York, 2000.

[VVV94] A.I. Volpert, V.A. Volpert, and V.A. Volpert. *Traveling Wave Solutions of Parabolic Systems.* American Mathematical Society, Providence, RI, 1994.

[WA04] N.G. Wahlgren and N. Ahmed. Neuroprotection in cerebral ischaemia: Facts and fancies–the need for new approaches. *Cerebrovasc. Dis.*, 17:155–166, 2004.

[Wal92] W. Walz. Role of na/k/cl cotransport in astrocytes. *Can. J. Physiol. Pharmacol*, 70:S260–S262, 1992.

[Wat84] H.A. Watts. Step size control in ordinary differential equation solvers. *Trans. Soc. Computer Simulation*, 1:15–25, 1984.

[Wid67] O.B. Widlund. A note on unconditionally stable linear multistep methods. *BIT Numer. Math.*, 7:65–70, 1967.

[WK] A. Williams and W. Kempf. *The boost thread library. Portable C++ multi-threading.* http://www.boost.org/.

[WPL+10] G. Wormeester, S. Pancheshnyi, A. Luque, S. Nijdam, and U. Ebert. Probing photo-ionization: simulations of positive streamers in varying N2:O2 mixtures. *J. Phys. D: Appl. Phys.*, 43(50):505201, 2010.

[WR08] D. Wang and S.J. Ruuth. A hybrid implicit-explicit adaptive multirate numerical scheme for time-dependent equations. *J. Comp. Mathematics*, 26(6):838–855, 2008.

[Wri71] K Wright. Some relationships between implicit Runge-Kutta, collocation and Lanczos τ methods, and their stability properties. *BIT Numer. Math.*, 10:217–227, 1971.

[Yan71] N.N. Yanenko. *The Method of Fractional Steps. The Solution of Problems of Mathematical Physics in Several Variables.* Springer-Verlag, New York, 1971.

[YFKP03] C.S. Yi, A.L. Fogelson, J.P. Keener, and C.S. Peskin. A mathematical study of volume shifts and ionic concentration changes during ischemia and hypoxia. *J Theor Biol.*, 220(1):83–106, 2003.

[YKA98] W.M. Yamada, C. Koch, and P.R. Adams. *Methods in Neuronal Modeling: From Ions to Networks*, chapter 1, pages 137–170. MIT Press, Cambridge, MA, 1998.

[Yos90] H. Yoshida. Construction of higher order symplectic integrators. *Physics Letters A*, 150(5-7):262–268, 1990.

[YP98] B. Yang and S.B. Pope. An investigation of the accuracy of manifold methods and splitting schemes in the computational implementation of combustion chemistry. *Combust. and Flame*, 112(1-2):16–32, 1998.

[YRH08] W. Ying, D.J. Rose, and C.S. Henriquez. Efficient fully implicit time integration methods for modeling cardiac dynamics. *IEEE Transactions on Biomedical Engineering*, 55(12):2701–2711, 2008.

[ZC09] D.W. Zingg and T.T. Chisholm. Jacobian-free Newton-Krylov methods: Issues and solutions. In H. Deconinck *et al.*, editor, *Computational Fluid Dynamics 2006*, pages 237–242. Springer Berlin Heidelberg, 2009.