UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES

DUALITY METHODS FOR TIME-SPACE ADAPTIVITY TO CALCULATE THE NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

Tesis Doctoral

Jaime Carpio Huertas

Ingeniero Industrial y Licenciado en Ciencias Físicas

2008

DPTO. MATEMÁTICA APLICADA A LA INGENIERÍA INDUSTRIAL
ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES

DUALITY METHODS FOR TIME-SPACE ADAPTIVITY TO CALCULATE THE NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

(MÉTODOS DE DUALIDAD PARA ADAPTACIÓN ESPACIO-TEMPORAL
EN LA RESOLUCIÓN NUMÉRICA
DE ECUACIONES EN DERIVADAS PARCIALES)

Tesis Doctoral

Autor: Jaime Carpio Huertas
Ingeniero Industrial y Licenciado en Ciencias Físicas

Directores: Rodolfo Bermejo Bermejo
MSc PhD University of British Columbia. Vancouver (Canada)

Juan Antonio Alonso Miguel
Dr en Ciencias Físicas por la Universidad Autónoma de Madrid

2008

Tribunal nombrado por el Magfco. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid, el día 4 de Marzo de 2008.

Presidente: D. Luis Casasús Latorre

Vocal: D. Amable Liñán Matínez

Vocal: D. Luis Ferragut Canals

Vocal: D. Malte Braack

Secretario: D. Luis Sanz Lorenzo

Suplente: D. Antonio L. Sánchez Pérez

Suplente: D. Ángel M. Ramos del Olmo

Realizado el acto de defensa y lectura de la tesis el día 7 de Abril de 2008 en la E.T.S. Ingenieros Industriales.

CALIFICACIÓN:

EL PRESIDENTE LOS VOCALES

EL SECRETARIO

Ver un mundo en un grano de arena
y un firmamento en una flor silvestre,
dar cabida al infinito en la palma de la mano
y a la eternidad en una hora.

J.G. Taylor (La nueva Física)

A Álvaro Crooke S.J.

in memoriam

Contents

AGRADECIMIENTOS

Querido Álvaro,

ya estamos otra vez de nuevo juntos, hoy no tendrás diálisis -seguramente allá donde estés no te hará falta sustituir tus viejos riñones por uno artificial- así que podemos charlar tranquilamente. Hace más de 4 años que nos conocimos, ¿te acuerdas? Yo estaba a punto de terminar la carrera de Ingeniero Industrial y tú estabas recién llegado del Congo. En aquel tiempo andaba pensando si la vida académica me permitiría acercarme al conocimiento de la Verdad y sobre todo si podría trabajar por ella de forma eficaz. Tú me animaste a que iniciase el camino de la Tesis Doctoral diciéndome que en el mundo de la Ciencia hay mucha humanidad que descubrir y me acompañaste en prácticamente la totalidad del viaje. Solamente al final te faltaron las fuerzas, pero ya estaba casi todo terminado, sólo era ya cuestión de escribir y rematar los últimos detalles. Así pues, hoy también será para ti un día grande en el Cielo.

Es impresionante el mundo que nos rodea, la belleza de las ecuaciones que nos permiten explorar incluso lo desconocido y que hacen que nuestra visión de la naturaleza sea más completa y más útil cada día. Decía Galileo que "la naturaleza es el gran libro que permanece constantemente abierto ante nuestros ojos y que está escrito en lenguaje matemático", por lo tanto sólo hay que afanarse en aprender ese lenguaje para descifrarlo. Aunque en este descubrir y conocer corres a veces el peligro de perder al Tú en el camino si no reconoces, que por encima de los fascinantes enigmas que tiene la naturaleza lo más misterioso y digno de admiración es el ser humano, capaz de aventurarse en el maravilloso mundo del saber simplemente por la satisfacción de conocer. Por lo que es imposible separar la ciencia del humanismo, y que la grandeza científica debe estar unida necesariamente a la grandeza humana, pues lo primero sin lo segundo es algo estéril. La escuela que triunfa es aquella que de forma generosa y gratuita abre sus puertas difundiendo el saber a todo el que con humildad se acerca a ella.

Por lo tanto es justo decir que esta Tesis no es otra cosa que una recopilación de la generosidad y del cariño de todas aquellas personas que en el camino de estos 4 años me han hecho ser. Todos de una u otra forma han hecho que esta Tesis fuera posible. Así pues, ahora llega el tiempo de los agradecimientos, ya se sabe que en el agradecer mejor si sobra que si falta. El agradecer es hermoso e incluso rentable, por lo tanto "miel sobre hojuelas".

En primer lugar hay que señalar las fuentes financiadoras del trabajo, así que agradecer al Fondo Social Europeo y a la Comunidad de Madrid la ayuda económica de la beca FPI con la que he contado durante 2 años que me ha permitido desarrollar mis estudios de doctorado, así como realizar varias estancias breves en la Universidad de Heidelberg. También agradecer el mismo tipo de ayuda económica a la Univerdad Politécnica de Madrid por contratarme como profesor ayudante en el laboratorio de Estadística de la E.T.S.I. Industriales donde he desarrollado mi labor docente-investigadora este último año. ¡Ah!, y puestos a recordar, sería injusto no valorar el aporte económico realizado por mis alumnos en la "Academia Pepe", sin los cuales no

podría haber empezado mis estudios de Doctorado al carecer el primer año de ayuda económica para ello.

En segundo lugar y donde me extenderé más es en aquellas personas que me han dedicado su tiempo, aquellas personas que en este camino han compartido su vida conmigo (ya se sabe, quien da su tiempo se da a sí mismo, ya que al acabarse el tiempo se acaba el ser). Aquí la lista es grande, seguro que más extensa que la que pongo aquí, no están todos los que son pero si que son todos los que están, a todos gracias porque con vuestro amor me habéis hecho menos simple y a la vez más humilde.

Agradecerle a Juan Antonio Alonso, que tiene el reconocimiento de codirector de tesis, el permitirme iniciar en el departamento de Matemática Aplicada los estudios de doctorado, por apoyarme y estar disponible en todo momento. También tengo que agradecerle que me recomendara personalmente al profesor Rodolfo Bermejo para realizar con él la Tesis Doctoral y así aprender a su lado. También en este punto hago extensible mi afecto por todo el personal del Departamento de Matemática Aplicada que siempre me ha tratado con mucho cariño.

Agradecer a mi director de Tesis Rodolfo Bermejo, con el que he trabajado estrechamente durante estos años de Doctorado, todo el tiempo que me ha dedicado, todo el interés y el cariño que ha mostrado por el trabajo que estaba desarrollando. En todo este tiempo hemos mantenido una relación mucho más estrecha que la de simple alumno-profesor. Le doy las gracias por la paciencia que ha tenido en mi formación matemática más rigurosa y le pido perdón por mis impaciencias y desacuerdos. Ahora, como manchego que soy, hago mías las palabras que Sancho dijera a Don Quijote "quiero decir que la conversación de vuestra merced ha sido el estiércol que sobre la estéril tierra de mi seco ingenio ha caído; la cultivación, el tiempo que ha que le sirvo y comunico; y con esto espero de dar frutos de mí que sean de bendición, tales, que no desdigan ni deslicen de los senderos de la buena crianza que vuestra merced ha hecho en el agostado entendimiento mío".

Agradecer al profesor Amable Liñan que me enseñase la belleza de los procesos de combustión. Agradecerle que siempre estuviera disponible para responder con atención desmedida a todas mis dudas y preguntas. También extiendo este agradecimiento a las personas de su grupo de investigación que siempre han valorado muy positivamente mi trabajo.

Agradecer al profesor Malte Braack su cariño y su sentido del humor tan necesario cuando uno está en tierra extraña. Gracias por enseñarme abiertamente todos los entresijos de la técnica "Dual Weighted Residual" en el tiempo que yo estuve en Heidelberg. Espero que podamos seguir compartiendo buenos momentos y trabajos en común.

Agradecer a Antún, Luis, Rosa y a toda la "peña" de Heidelberg su compañía y cariño durante mi estancia allí. En Heidelberg es donde, sin duda, desarrollé gran parte de esta tesis, y donde he hecho muy buenos amigos, cuasi-hermanos. "Ich liebe euch von ganzen Herzen" y sin duda los mejores resultados de esta tesis son fundamentalmente obra vuestra.

Agradecer a mis compañeros de Estadística, con los que he compartido mi labor docente este último año, todo el apoyo y el cariño con el que me recibieron. Gracias a Jesús Juan por confiar en mi para la plaza de profesor ayudante y a Carol agradecerle su amistad.

Agradecer a Carlos Díaz y a todo el aula Marcelino Legido que me enseñaran que el hombre completo es aquél en el que ninguna dimensión humana esencial está radicalmente

oprimida. Que una humanidad completa necesita de ingenieros y místicos; incluso necesita hasta híbridos.

Agraceder a mis alumnos la paciencia que han tenido conmigo en mis primeras clases de universidad, sin vosotros no tendría sentido esta profesión. La investigación sin docencia es tan inútil como la docencia sin investigación, en este mundo hay que renovarse continuamente para poder ofrecer a los jovenes una enseñanza fresca que despierte la inquietud por la vida que todos ellos deben albergar en sus corazones. En el joven todo es futuro por venir, todo para él ha de ser esperanza y aliento para lograr transformar este mundo en algo más habitable.

Agradecer a Sam (Juan Luis Prieto) su inestimable ayuda a lo largo del doctorado, sin duda Frodo nunca hubiese llegado tan lejos sin ti. Por todas las páginas de tesis que te has leído, por todas las llamas que en tu ordenador has quemado, por todos los "marrones" en los que te he metido y sobre todo, por los buenos ratos compartidos. Mucho ánimo para la recta final de tu tesis.

Agradecer muy especialmente a mis padres, el maravilloso regalo de la vida, que me enseñaran el valor fundamental del trabajo, del esfuerzo diario, de la lucha incansable... que como buenos hombres de campo me enseñaran que las tierras secas, regándolas y cultivándolas bien, vienen a dar buenos frutos.

Por último agradecerte a ti Álvaro tu presencia y compañía en tus últimos años de vida, si esta Tesis empezó contigo bueno es que termine de igual forma. La muerte es sólo otro sendero que recorreremos todos, antes o después. Hasta entonces a mí me queda responder al cariño que de alguna u otra manera habéis depositado todos en mí, para que sepa llegar de forma eficaz a todos, especialmente a los más necesitados.

Muchas gracias.

RESUMEN

En esta tesis presentamos un método adaptativo espacio-temporal para calcular de forma eficiente la solución numérica de EDP's, especialmente en problemas dependientes del tiempo. Formularemos el método en primer lugar para ecuaciones de reacción-difusión y después generalizaremos para todo tipo de ecuaciones de convección-reacción-difusión (o sistema de ellas), tratando los términos convectivos mediante un esquema semi-lagrangiano. Nuestro algoritmo adaptativo, formulado bajo la teoría de elementos finitos, está basado en el método "Dual Weighted Residual (DWR)", que es presentado en [11] como una técnica para el cálculo del error a posteriori en cantidades de interés físico $J(u)$ de la solución. Esta técnica conduce a algoritmos eficientes de elementos finitos adaptativos en el caso estacionario, pero es todavía un problema abierto el conseguir algoritmos adaptativos eficientes con esta misma metodología en problemas dependientes del tiempo. El término eficiente lo empleamos aquí para designar a un algoritmo numérico que proporciona una buena estimación espacial y temporal del error a posteriori cometido, con el fin de generar particiones en espacio y tiempo que conduzcan a una equidistribución del error con un coste computacional óptimo en almacenamiento y tiempo de CPU.

Hasta ahora, la aplicación de las técnicas DWR en ecuaciones dependientes del tiempo llevaba asociado el problema de almacenar necesariamente tanto la topología como la solución de los cálculos numéricos realizados en todos los subintervalos de tiempo $I_n := (t_{n-1}, t_n]$ que forman el intervalo de integración temporal $[0, T]$, lo que requeriría enormes recursos computacionales. Para solventar este problema, proponemos en esta tesis un algoritmo adaptativo (en espacio y tiempo) que, basado en las ideas del método DWR, requiere solamente almacenar una malla espacial que es adaptada de forma adecuada a medida que el tiempo progresa. Nuestro algoritmo consiste en aplicar la técnica DWR localmente en cada subintervalo de tiempo $I_n := (t_{n-1}, t_n]$ al problema original, que ha sido perturbado con una condición inicial dada por la solución numérica en el instante t_{n-1}. Así, más que controlar el error en todo el intervalo temporal $[0, T]$, lo que hacemos ahora es controlar el error local en cada subintervalo I_n. Sin embargo, un punto débil de esta aproximación es que un buen control del error local no garantiza que el error global esté acotado $|J(u) - J(u_{h\Delta t})| < GTOL$, siendo $GTOL$ una tolerancia global impuesta. Es bien conocido el hecho de que la magnitud del error global depende de la rigidez del problema; si la rigidez es baja o moderada, acabaremos teniendo un error global pequeño si se controla bien el error local, pero si la rigidez es alta el error global puede ser grande incluso si se controlase el error local con una tolerancia razonable. Dicho esto, debemos mencionar algunas buenas propiedades del algoritmo presentado en esta tesis, las cuales permiten realizar una eficiente adaptación espacial y temporal. La primera es que es auto suficiente en proporcionar un criterio preciso para la adaptación del paso temporal Δt y del tamaño de malla h, tal que en cada subintervalo de tiempo I_n impondremos las tolerancias Tol_s y Tol_t ($TOL = Tol_s + Tol_t$) al error espacial y temporal estimado respectivamente. La segunda propiedad es que el algoritmo extiende la idea del post-proceso espacial del tradicional método DWR a mallas no estructuradas.

Para dotar a la tesis de un contenido práctico en el marco de la ingeniería, al final de la misma, se resolverá un interesante problema dentro del marco de la combustión, denominado "Problema de las llamas levitadas". El cálculo de la solución numérica a este problema es desafiante para los métodos numericos convencionales porque adicionalmente a las ecuaciones de Navier-Stokes compresibles tenemos fuertes no linealidades en las ecuaciones de convección-reacción-diffusion para las especies químicas y la temperatura. Esto conduce a fenomenos físicos que exhiben gran disparidad entre las escalas temporales y espaciales, y una superficie libre móvil representada por el frente de llama. Por consiguiente, una técnica numérica eficiente para resolver el problema debería hacer uso de métodos adaptativos tanto en tiempo como en espacio para resolver apropiadamente las escalas y la evolución de los frentes de llama.

ABSTRACT

In this thesis we present a space-time adaptive method to calculate, in an efficient manner, the numerical solution of PDE's, specially in time-dependent problems. Firstly we will formulate the adaptive method for reaction-diffusion equations and then we will apply it to solve all kinds of convection-diffusion-reaction equations (or system of them), dealing with the convective terms by means of a semi-Lagrangian scheme. Our method, formulated in the context of finite elements, is based on the philosophy of the *Dual Weighted Residual* (hereafter, DWR) methodology for goal-oriented error estimation. The conventional DWR method is presented in [11] as a technique to derive a posteriori error estimates of arbitrary quantities of physical interest $J(u)$ of the solution. This technique yields efficient adaptive finite element algorithms for stationary problems, but it is still an open question how to achieve an efficient adaptive algorithm via the application of the DWR methodology for time-dependent problems. Here, we use the term efficiency to denote an adaptive algorithm that provides both space and time error estimates, computed from the numerical solution, in order to allow the generation of space and time partitions that yield an equidistribution of the error at an optimal cost in computational resources such as storage and CPU time.

To date, the solution of multi-dimensional time-dependent problems with the DWR has the drawback of having to store the topology and related calculations of all time subintervals $I_n := (t_{n-1}, t_n]$ needed to cover the whole time integration interval $[0, T]$. Since this has to be done in a fully adaptive strategy, it would require a large amount of computational resources. To palliate this issue, we propose in this thesis a fully adaptive algorithm that is based on the DWR idea, but requiring to store only one spatial mesh, which is properly adapted as time progresses. Our algorithm consists of applying the DWR technique locally in each time subinterval $I_n := (t_{n-1}, t_n]$ to the original problem with a perturbed initial condition given by the numerical solution at time t_{n-1}. Thus, rather than controlling the error for the whole interval $[0, T]$ what we have now is a good local control of the error in each subinterval I_n. However a weak point of this approach is that having a good local error control does not guarantee that the global error will be bounded as $|J(u) - J(u_{h\Delta t})| < GTOL$, $GTOL$ being a prescribed global tolerance, for it is a well known fact that the magnitude of the global error depends on the stiffness of the problem. If the stiffness is low or moderate we will end up having a small global error if we control the local error well, but if the stiffness is large enough the global error will be large even if we control the local error with a reasonable tolerance. This being said, we should mention some good properties of the algorithm presented in this thesis. The first one is that it is self-sufficient in providing a precise criterium for the adaptation of both the time step Δt and the mesh size h, such that in each time subinterval I_n we prescribe the tolerances Tol_s and Tol_t ($TOL = Tol_s + Tol_t$) for the spatial and temporal estimator of the error respectively. The second property is that the algorithm extends the idea of the space post-processing of the DWR method to unstructured meshes made of simplices.

As an engineering application of the fully numerical adaptive procedure, at the end of

the thesis we will solve an interesting problem within the framework of the combustion named "Lifted flames problem". The calculation of the numerical solution of this problem is a challenging question for conventional numerical methods because, in addition to the compressible Navier-Stokes equations, one has to deal with strong non linear convection-reaction-diffusion equations for species concentrations and temperature, yielding phenomenae that exhibit a large disparity of both temporal and spatial scales**,** and a free surface represented by the flame front. Therefore, an efficient numerical approach to the problem should make use of adaptive methods both in time and space to properly solve the scales and the evolution of the flame fronts.

Chapter 1

INTRODUCTION

Engineers, mathematicians, physicists and scientists in general, working nowadays on a wide variety of problems, find useful to come up with mathematical models of the systems they are investigating. To do this, they give a simplified description of the problem which mainly takes the form of differential or integral equations. The first historical mention of the term "equatis differentialis" goes back to 1676, when Gottfried Wilhelm von Leibniz designated with this term a function that satisfies with one or more of its derivatives a given relation. From that moment on, the theory of differential equations experienced a great development. Unfortunately, though only in the very simplest cases are possible to find analytical solutions to equations of the model, and in general one has to rely on numerical techniques for finding approximate solutions.

The basic idea in any numerical method involving differential equations is to discretize the given continuous problem implying infinite degrees of freedom, so as to obtain a discrete problem or system of equations with only a finite number of unknowns that may be solved thereafter. One of the most widely used numerical methods to solve differential and integral equations in science and engineering is the *finite element method* (FEM). The method was first developed in 1943 by Richard Courant [27], who used the Ritz method of numerical analysis and minimization of variational calculus to obtain approximate solutions to vibration systems. Shortly thereafter, a paper published in 1956 by structural engineers [67] can be considered the first publication in the field of finite element methods. At this point, the method was thought of as a generalization of earlier methods in structural engineering for beams, frames and plates, where the structure was subdivided into small parts, the so-called finite elements, whose behaviour was known. When the mathematical study of the finite element method started in the mid 60's, it soon became clear that, in fact, the method is a general technique for the numerical solution of partial differential equations with roots in the variational methods in mathematics; first introduced at the beginning of the twentieth century. During the 60's and 70's the method was developed by engineers, mathematicians and numerical analysts up to a general method for the numerical solution of partial differential equations and integral equations with applications in more areas of science and engineering than structural engineering, including heat transfer, solid state diffusion and reactions with moving boundaries, fluid dynamics and electromagnetism.

With the rapid development of high speed computers over the past decades, the possibilities of efficiently utilizing this numerical method have dramatically increased. As it is, the underlying physical processes are often extremely complex and to obtain numerical solutions to the partial differential equations with high accuracy is not a trivial problem. For this reason new techniques have to be developed to reduce the numerical effort and make the simulation of complex problems feasible. In the late 70's, adaptive procedures appear in the numerical field of partial differential equations and have since spread out, so that, there are now standard tools

in science and engineering when making realistic and feasible computations in multi scale phenomena, specially in 3D. The goal in adaptation is to design economical meshes and time step sizes to find the solution of the problem based on information extracted from an a posteriori error estimator, which measures the difference between the unknown real solution u and the calculated discrete solution u_h (where $h \in \mathbb{R}_+$ is the discretization parameter).

Most of the existing work on a posteriori error estimation in Galerkin finite element methods is done with respect to the natural "energy norm" induced by the underlying differential operator, or the L^2 norm, usually involving unknown 'stability constants' and the residual of the differential equation $\rho(u_h)$, which is well defined in the context of the Galerkin finite element method.

$$\eta \leq c_s \left\| \rho^2(u_h) \right\| .$$

This approach was initiated by the pioneering work of Babuska and Rheinboldt in 1978 ([4] and [5]), and was then further developed by Ladeveze and Leguillon [51], Bank and Weiser [12], Babuska and Miller [6], to mention only a few of the most influential papers. For discussions and further references, we refer to the surveys by Verfürth [68] and Ainsworth and Oden [1]. Error estimation in the energy norm seems rather generic as it is directly based on the variational formulation of the problem and allows us to exploit its natural coercivity properties.

However, in most applications, the error in a global norm does not provide useful bounds for the errors of the quantities with real physical interest. Further, their sensibility to local error sources is not properly represented by global stability constants. These deficiencies are overcome by employing duality techniques, as is commonly done in a priori error analysis of finite element methods. Let $J(u)$ be a quantity of physical interest derived from the solution u, with the goal to control the error $J(u)-J(u_h)$ in terms of the local residuals $\rho_K(u_h)$ computable on each element K of the mesh cells $\mathbb{T}_h$ and certain local sensitivity factors. This leads to a 'weighted' a posteriori error estimator

$$\eta = |J(u) - J(u_h)| \approx \sum_{K \in \mathbb{T}_h} \left\| \rho(u_h) \cdot \omega_h(z) \right\|_K ,$$

where the sensitivity factor $\omega_h(z)$ is obtained solving a linear 'adjoint' (or dual) problem. The adjoint solution z may be viewed as a generalized Green's function with respect to the output functional $J(\cdot)$, and accordingly, the weight $\omega_h(z)$ describes the effect of local variations of the residual $\rho(u_h)$ on the error quantity $J(u) - J(u_h)$.

The use of duality arguments in a posteriori error estimation goes back to ideas of Babuska and Miller in 1984 ([7], [8] and [9]) in the context of post-processing of 'quantities of physical interest' in elliptic model problems. It has since been systematically pursued by Eriksson and Johnson ([31] and [32]) and their collaborators for more general situations (see also Johnson [48] and the survey paper by Eriksson, Estep, Hansbo and Johnson [36]). Here, sensitivity factor $\omega_h(z)$ for adjoint problems are mostly derived by analytical arguments. But in practice, is almost impossible to determine the importance of each residual for any target quantity by analytical means. In 1996 with Becker and Rannacher [13], this approach is further developed into a computation-based feedback method, called the 'dual-weighted residual method' (or, for short, 'DWR method'), where the sensitivity factor $\omega_h(z)$ is obtained by means of computational resolution of the dual problem and a post-processing of its solution. This approach is particularly designed for achieving high accuracy in the target functional at minimum computational cost, since the additional work required by the evaluation of the error bounds is usually acceptable, particularly in nonlinear cases, where it amounts only a moderate fraction of the

total cost. For surveys and applications to problems in solid, fluid mechanics, physics and chemistry see Rannacher and Braack ([58], [59] and [20]).

The performance of the DWR method has been shown for several linear and nonlinear model problems mainly from fluid and solid mechanics. But there are several open problems in the theoretical foundation of the DWR method, as well as in its practical implementation, that need further investigation. One of the major challenges for the DWR method are multidimensional time-dependent problems, where rigorous error control in a space-time frame requires to solve a backward space-time adjoint problem. Specially for nonlinear problems, this may be prohibitive with respect to storage space and computing time requirement. Temporal equations using the DWR technique have been treated by Hartmann [45], Bangerth and Rannacher [10], but the realization of these concepts for practical problems beyond simple model situations is still in an immature state. To this respect, it is worth mentioning the recent contribution of Schmich and Vexler [61], where some progress has been done in formulating efficient DWR adaptive finite elements for parabolic equations.

To alleviate this issue, we propose in this Thesis a space-time adaptive algorithm that based on the DWR idea, requires to store only one spatial mesh, which is properly adapted as time progresses. Our technique is placed between the traditional approach to perform adaptation and the classical DWR method, taking advantage of the best characteristics of both. In our case, the error can be expressed as a space-time integral of residuals of computed solution multiplied by weight functions coupled to the solution of an associated linear dual problem, but it is only defined in the actual time subinterval of integration $I_n := (t_{n-1}, t_n]$. Thus, rather than controlling the error for the whole interval $I := (0, T]$ what we have now is a good local control of the error in each subinterval I_n. A weak point of this approach is that having a good local error control does not guarantee that the global error will be bounded, since the time-accumulation of the local errors is not taken into account. However, our method has the following advantages:

1. We do not need to solve a dual problem backward in the whole time interval $(0, T]$ to obtain the a posteriori error estimator. We only solve the dual problem in each time subinterval $(t_{n-1}, t_n]$ to compute the error. Therefore computational resources such as storage and CPU time are smaller.

2. We can make an effective control on the size of the elements in the mesh and over the time step size while we perform the temporal integration of the problem. According to that, the time steps and the mesh sizes are automatically chosen with respect to a tolerance prescribed by the user.

3. The algorithm extends the idea of the space post-processing of the classical DWR method to unstructured meshes made of simplices.

4. With this scheme we can easily incorporate the error that we commit in the discretization of the convective terms within the semi-Lagrangian framework (a good exposition of the semi-Lagrangian scheme can be seen in [2]), and we give an easy algorithm to make adaptation in all kind of convection-diffusion-reaction equations with the DWR technique.

Therefore and to sum up, the main aim of this work is to develop a robust adaptive finite element code in the standard programming language C, that allows us to make goal oriented adaptation on unstructured triangular meshes in 2D with special interest to treat temporal

convection-diffusion-reaction equations. The main advantage of the presented approach will be more clear in 3D problems, where non-adaptive methods need high memory and computational requirements. Our code is designed for 2D situations but we can easily extend these ideas to 3D problems; it only requires a special care in setting up the data structure for mesh organization. The contents of the thesis are as follows:

- In Chapter 2 we expound the main ideas to design an adaptive finite element algorithm. We will explain the data structure for mesh organization and expound the refinement and coarsening algorithm by the bisectioning technique. We will additionally comment the selection of the optimal mesh and time step size using the information derived from the a posteriori error estimator.

- The underlying concept of the DWR method will be introduced in Chapter 3. We apply the method to reaction-diffusion equations as example of time-dependent problems. This chapter can be considered the core of our work.

- In Chapter 4 we will generalize the new proposed DWR method for convection problems introduced in the semi-Lagrangian framework and apply the new algorithm to solve different kinds of convection-diffusion-reaction equations.

- At the end of the thesis, in Chapter 5, will be devoted to an important problem in combustion theory, the so-called "lifted flames problem" [39], where a thin flame front is moving and presents a characteristic triple flame structure that has to be solved with a good resolution. The features of these equations may change in time and space. While they become reaction-dominated in reaction zones, in other parts of the domain the convection is more important and in boundary layers the diffusive part is dominant. Therefore, combustion problems are known to range among the most demanding problems for spatial-temporal adaptation.

Chapter 2

ADAPTIVE FINITE ELEMENTS

The efficient numerical treatment of multi scales phenomena and problems with poor regularity of the solution must be carried out with an adaptive method based on a posteriori error estimators to make realistic computation feasible, specially in 3D. To treat time-dependent problems, we split the time interval $I := (0, T]$ into half-open subintervals $I_n := (t_{n-1}, t_n]$ of length $\Delta t_n := t_n - t_{n-1}$,

$$0 = t_0 < ... < t_n < ... < t_N = T.$$

In each time subinterval I_n we generate a conforming triangulation $\mathbb{T}_h^n$ of the domain Ω and calculate with a time step size Δt_n the numerical solution $u_{h\Delta t}^n$. Then, an adaptive finite element method in time could consist of successive loops as shown in Figure 2.1.

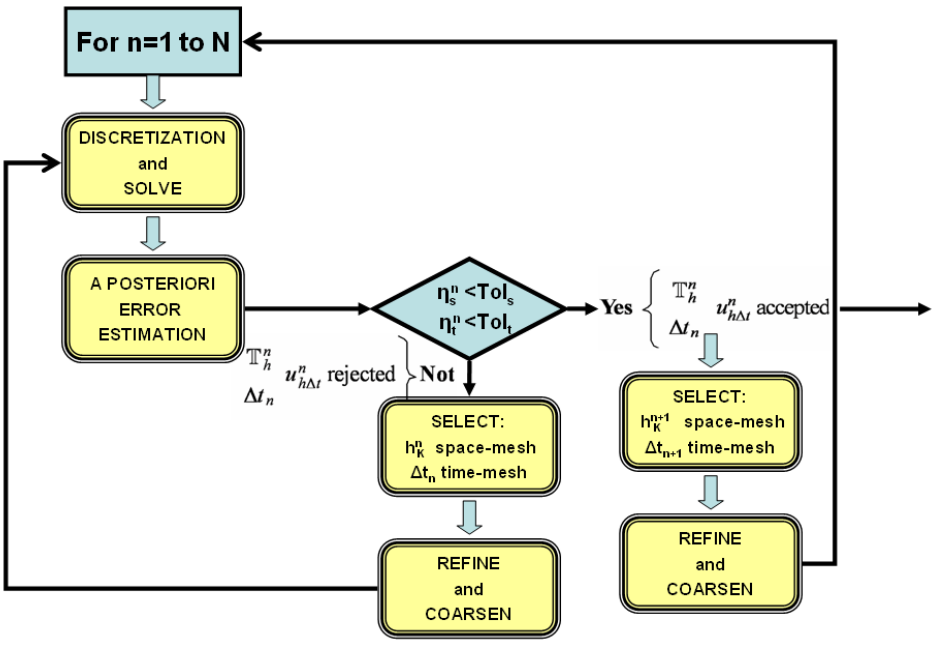

Figure 2.1: Scheme of the basic time-space adaptive algorithm for time-dependent PDE's.

1. **Discretization and solve.**

   The finite element method calculates an approximation to the true solution in some finite dimensional functional space. This space is built from local functions in space and time,

usually polynomials of low order on elements of a partitioning of the domain (the h space-mesh and Δt time-mesh). In each time subinterval $I_n := (t_{n-1}, t_n]$, of length $\Delta t_n := t_n - t_{n-1}$, we generate a regular triangulation $\mathbb{T}_h^n$ of the spatial domain Ω and define an associated space-time finite element space $V_{h\Delta t}^{(r)}$. The discretization leads to an algebraical system of equations that must be solved to obtain the numerical solution $u_{h\Delta t}^n$.

2. **A posteriori error estimation**

   An a posteriori error estimator, e, is needed to assess the difference between the unknown real solution $u(x,t)$ and the calculated discrete solution $u_{h\Delta t}(x,t)$. This estimator has to be reliable and efficient in order to perform error control. The traditional approach to adaptivity aims at estimating the error with respect to the energy-norm of the problem or the global L^2-norm, however this is generally not what applications need, so if we are interested in computing with accuracy locally defined scalar quantities of the global solution, we must use duality techniques to obtain an error estimator in a functional of physical interest $J(u)$ of the solution, $e = J(u) - J(u_{h\Delta t})$. For stationary problems, the application of duality techniques yield very accurate and efficient adaptive algorithms [11]; however, for time-dependent problems, the efficient application of such techniques in the framework of adaptivity is still an open problem as recognized in [11]. In this thesis we present a new treatment of the a posteriori error estimator for unstructured mesh and give a new point of view to treat time-dependent problems, where we give up to have control over the error in the whole time interval $[0,T]$ and we only consider the calculation of the local or truncated error at each time step I_n. With the a posteriori error estimator e^n, we will extract local error indicators η_s^n (in space) and η_t^n (in time) that will be used for practical mesh and time adaptation respectively.

3. **Select h_K space-mesh and Δt_n time-mesh**

   With the above error indicators, η_s^n and η_t^n, we generate space and time partitions that yield an equidistribution of the error at an optimal cost in computational resources such as storage and CPU time. We propose here a criteria to select both the optimal size of the element h_K and the optimal length of the time interval Δt_n, in order to have the spatial and temporal error indicators close but below of a fixed tolerance $\eta_s^n < Tol_s$ and $\eta_t^n < Tol_t$. With this choice we avoid, to recalculate the solution, in almost all of the time steps, and in the numerical examples less than 1% of the solutions are wrong and do not satisfy the tolerance criteria.

4. **Refine and coarsen**

   The spacial mesh $\mathbb{T}_h^n$ is an unstructured conforming triangulation of the computational domain Ω into triangles in 2D. The refinement and coarsening routines construct a sequence of nested meshes, from a given initial triangulation (macro triangulation), with a hierarchical structure. Here, the recursive refinement by bisection is implemented, where each triangle (father) will be divided in two triangles (children).

Next in this chapter, we expound in more detail all of these above four points.

2.1 Finite element discretization

Our method is based on conforming finite elements (for an introduction to **f**inite **e**lement **m**ethod (FEM) see for example [21] and [47]). This method computes an approximation to the true

solution in a finite dimensional function space. This space is built from local function in space and time, usually polynomials of low order on elements of a partitioning of the domain (the h space-mesh and Δt time-mesh). The choice of finite elements against other strategies as finite volume and finite difference methods is motivated for the following reasons:

- The finite element method allows the use of nearly arbitrary meshes, which are necessary for complex geometries and also for adaptive refinements.
- An important property of the finite elements methods is its Galerkin character, which leads to a posteriori error estimates forming the base for error control and local mesh refinement.

2.1.1 The spatial mesh

In order to compute the spatial numerical solution, we generate at each time level t_n a regular conforming triangulation $\mathbb{T}_h^n$ of the spatial domain Ω composed of elements K with Lipschitz boundary ∂K satisfying the following conditions.

(i) Let NE be the number of elements in $\mathbb{T}_h^n$, then $\overline{\Omega} = \cup_j \overline{K}_j, \quad j = 1, .., NE.$

(ii) For $1 \leq i, j \leq NE,\ i \neq j,$

$$\overline{K}_i \bigcap \overline{K}_j = \begin{cases} x_l, \text{ a mesh point, or} \\ \Gamma_{ij}, \text{ a common side, or} \\ \emptyset, \text{ void otherwise.} \end{cases}$$

(iii) There is a positive constant μ such that for all K, $\dfrac{h_K}{\rho_K} \leq \mu$, where ρ_K denotes the diameter of the circle inscribed in K and h_K is the length of the largest side of K.

Next, we associate a family of finite elements to the partition $\mathbb{T}_h^n$. To do so, we consider an element of reference $\widehat{K} \subset \mathbb{R}^2$ such that for each element K of $\mathbb{T}_h^n$ we can define the one-to-one mapping $F_K : \widehat{K} \to K$ with inverse F_K^{-1}, such that if $\widehat{P}_m(\widehat{K})$ is the set of polynomials $\widehat{p}(\widehat{x})$ of degree $\leq m$ (in this work $m = 1$ linear, or $m = 2$ quadratic finite elements) defined on $\widehat{K}$, then for each K we define

$$P_m(K) = \{p(x), x \in K : p(x) = \widehat{p}(F_K^{-1}(x))\}.$$

We now define the family of conforming finite element space $V_h^n \subset H_0^1(\Omega)$,

$$V_h^n = \{v_h \in C^0(\overline{\Omega}) : v_h|_K \in P_m(K)\ \forall K \in \mathbb{T}_h^n\} \tag{2.1}$$

if N_h^n is the number of mesh points, then any element of V_h^n is expressed as

$$v_h = \sum_{i=1}^{N_h^n} V_i \varphi_i^n(x), \tag{2.2}$$

where $V_i = v_h(x_i),\ x_i$ being the $i-th$ mesh point, and $\{\varphi_i^n\}$ is the set of global nodal basis functions of V_h^n characterized by the property $\varphi_i^n(x_j) = \delta_{ij}$.

2.1.2 The spatial-temporal mesh

For the time discretization, we split the time interval $I := (0, T]$ into half-open subintervals $I_n =: (t_{n-1}, t_n]$ of length $\Delta t_n := t_n - t_{n-1}$,

$$0 = t_0 < ... < t_n < ... < t_N = T.$$

At each time level t_n, let V_h^n be an appropriate finite dimensional space defined by (2.1) on spatial mesh $\mathbb{T}_h^n$ which may vary from time step to time step, but is constant inside of I_n. Extending the spatial meshes $\mathbb{T}_h^n$ to the time slabs $S_n = \Omega \times I_n$, we can define, for fixed integers m and r the following functional space $V_{h\Delta t}^{(r)}$:

$$\begin{cases} V_{h\Delta t}^{(r)} = \{v_{h\Delta t} : \overline{\Omega} \times \overline{I} \to \mathbb{R} : \ \forall n \text{ and } (x,t) \in S_n, \ v_{h\Delta t} \in C(\overline{I}_n; V_h^n), \ v_{h\Delta t}(x,0) \in V_h^0 \\ \qquad \text{and } \ v_{h\Delta t}(x,\cdot)|_{\overline{I}_n} \ \in \ P_r\}, \\ \\ V_h^n = \{v_h \in C^0(\overline{\Omega}) : v_h|_K \in P_m(K) \ \forall K \in \mathbb{T}_h^n\}, \end{cases} \tag{2.3}$$

where $P_r(I_n)$ is the set of polynomials of degree at most r defined on I_n.

The change of the spatial mesh from one time slab to the next one claims for an efficient algorithm which allows us to transfer the solution from the mesh $\mathbb{T}_h^{n-1}$ to the mesh $\mathbb{T}_h^n$ with the minimum loss of information. We will define the $L^2-projection$ or a high order $interpolation$ with a control of the error over the solution. Taking

- $Interpolation$: $v_{h\Delta t}^{n-1}(x) \in V_h^{n-1}$ and $I_h^n v_{h\Delta t}^{n-1}(x) \in V_h^n$, where I_h^n is the interpolation operator in V_h^n defined by:

  $$I_h^n v_{h\Delta t}^{n-1}(x_i) = v_{h\Delta t}^{n-1}(x_i),$$

  x_i being the $i-th$ mesh point.

- $L^2 - projection$: $v_{h\Delta t}^{n-1}(x) \in V_h^{n-1}$ and $P_h^n v_{h\Delta t}^{n-1}(x) \in V_h^n$, where P_h^n is the projection operator in V_h^n defined by:

  $$(P_h^n v_{h\Delta t}^{n-1}(x), \varphi_i)_\Omega = (v_{h\Delta t}^{n-1}(x), \varphi_i)_\Omega \quad \forall \varphi_i,$$

  where $(\cdot,\cdot)_\Omega$ is the scalar product in L^2 and φ_i the $i-th$ basic function of V_h^n.

Once the discretization and the assemblage of the matrices are made, we have to solve the so generated systems. The resolution of a discrete system is the core part of every finite element program and it is necessary after each partition (spatial or temporal) change. Additionally, this step is usually the most time consuming part of the above algorithm (Figure 2.1). A general finite element toolbox must provide flexibility while still providing enough efficiency for the resolution. To solve the nonlinear system, we will use a classical Newton algorithm, with the possibility of damping for those situations where the stiffness of the systems would require it. We solve the linear systems with Incomplete Cholesky Preconditioned Conjugate Gradient for symmetrical systems and a Diagonal Preconditioned BICGSTAB for the nonsymmetrical ones.

2.2 A posteriori error estimator

Let us consider a temporal problem written in the following way

$$\begin{cases} \partial_t u + A(u) = f(u) & \text{in } \Omega \times I, \\ u(x,0) = u^0 & \text{in } \Omega \qquad \text{Initial condition,} \\ u(x,t) = 0 & \text{on } \partial\Omega \times I \quad \text{Boundary condition.} \end{cases} \tag{2.4}$$

To make adaptation, we need to create an error indicator η derived from a posteriori error estimator e that allows us to measure the difference between the unknown real solution and the calculated discrete solution. Next, we show different approaches to measure the a posteriori error.

2.2.1 The traditional approach

The traditional approach to adaptivity aims at splitting the time intervals $I := (0, T]$ into half-open subintervals $I_n := (t_{n-1}, t_n]$, converting the above problem (2.4) into the following problem (2.5).

$$\begin{cases} \partial_t U + A(U) = f(u) & \text{in } \Omega \times I_n, \\ U(x,t_{n-1}) = u^{n-1}_{h\Delta t} & \text{in } \Omega \qquad \text{Initial condition,} \\ U(x,t) = 0 & \text{on } \partial\Omega \times I_n \quad \text{Boundary condition.} \end{cases} \tag{2.5}$$

Note that we have changed the notation from $u(x,t)$ (for the exact solution at time $t \in (0,T]$ of (2.4)) to $U(x,t)$ (for the exact solution at time $t \in (t_{n-1}, t_n]$ of the temporal problem with a perturbed initial condition $U(x,t_{n-1}) = u^{n-1}_{h\Delta t}$, the numerical solution at the previous time step).

A time-space discretization is needed to obtain the numerical solution $u^n_{h\Delta t}$ at time t_n. Therefore, the first stage is to discretize in time of the equation, with a known scheme, for instance, implicit Euler, Crank-Nicolson,... just to name a few of the most familiar schemes. For example, if we choose the implicit Euler scheme we have

$$\frac{u^n_{\Delta t} - u^{n-1}_{\Delta t}}{\Delta t_n} + A(u^n_{\Delta t}) = f(u^n_{\Delta t}), \tag{2.6}$$

and now we perform the finite element discretization in space for this equation. In the weak formulation we have to find the numerical solution $u^n_{h\Delta t}$ belonging to an appropriate discrete space V^n_h such that for all $\varphi_{h\Delta t} \in V^n_h$ the following equation is satisfied

$$\int_\Omega \left(\frac{u^n_{h\Delta t} - u^{n-1}_{h\Delta t}}{\Delta t_n} \cdot \varphi_{h\Delta t} \right) d\Omega + \int_\Omega A(u^n_{h\Delta t})(\varphi_{h\Delta t}) d\Omega = \int_\Omega f(u^n_{h\Delta t}) \varphi_{h\Delta t} d\Omega. \tag{2.7}$$

With both time and space discretization, we can measure the local or truncation error indicator in the time level t_n as the difference between the unknown real solution $U(x,t)$ and the calculated discrete solution $u_{h\Delta t}(x,t)$. However, to perform this operation, people usually choose heuristic indicators of the errors, for example:

- Spatial error:

  The L^2-norm of the residuals of the problem (2.6) is commonly employed as spatial error indicator

  $$\begin{cases} \eta^n_s = \|\rho^n(u_{h\Delta t})\|^2_{L^2} = \left(\int_\Omega (\rho^n(u_{h\Delta t}))^2 d\Omega\right) = \sum_{K \in \mathbb{T}^n_h} \left(\int_K (\rho^n(u_{h\Delta t}))^2 d\Omega\right), \\ \\ \eta^n_K = \int_K (\rho^n(u_{h\Delta t}))^2 d\Omega, \end{cases}$$

where $\rho^n(u_{h\Delta t})$ is the residual of the time discretization (2.6)

$$\rho^n(u_{h\Delta t}) = \frac{u^n_{h\Delta t} - u^{n-1}_{h\Delta t}}{\Delta t_n} + A(u^n_{h\Delta t}) - f(u^n_{h\Delta t}).$$

An easier formula, implemented in the majority of the commercial codes, is the jump of the gradient of the solution

$$\eta^n_K = \frac{h^2_K}{2} \sum_{E \subset \partial K \setminus \partial\Omega} \left(\left[\frac{\partial u^n_{h\Delta t}}{\partial n} \right] \right)^2_E,$$

where h_K is the length of the largest side of K and $[\partial u^n_{h\Delta t}/\partial n]$ denotes the jump of $\partial u^n_{h\Delta t}/\partial n$ across the inter-element edges.

- Temporal error:

  As error indicator in time, one could consider the local or truncation error in the approximation of (2.5). The temporal truncation error, can be calculated by a temporal Taylor expansion or to use a temporal scheme with higher order than the original one (2.6) and then, to compare both results (for instance, the classical embedded Runge Kutta formulae for ODE's). Obviously, the last approach is not very attractive, since we need to solve an additional problem more complicated, in general, than the original one.

This kind of method to calculate the error indicator combining error estimation in the L^2-norm of the residuals for spatial discretization with heuristic truncation error estimation, is proposed in very different works such as [73], [52], [65] and [17].

2.2.2 Classical DWR method

The error indicator in the energy-norm, or heuristic errors, are not what applications need, generally speaking. A new duality-based technique, the **D**ual **W**eighted **R**esidual method (or shortly DWR method) for goal-oriented error estimation was introduced in [36] and it seems to be one of the most robust techniques to implement adaptivity. This method aims at economical computation of arbitrary quantity $J(u)$ with physical interest for the solution. For example, the drag coefficient of a body immersed in a viscous flow, the mean heat flux along the cold wall, the reaction rate in combustion problems to determinate the right position of the flame front are computed, and so on.

To calculate the error for the functional $J(u)$ of the solution, $e = J(u) - J(u_{h\Delta t})$, we shall consider the original or primal problem (2.4) and its associated linear dual problem, which is easily defined following the general theory of the DWR method, and can be written as:

$$\begin{cases} -\partial_t z + A^*(u)(z) = f_J(u) & \text{in } \Omega \times I, \\ z(x,T) = z^T_J(u) & \text{in } \Omega \qquad \text{Final condition,} \\ z(x,t) = 0 & \text{on } \partial\Omega \times I \quad \text{Boundary condition,} \end{cases} \tag{2.8}$$

where the final condition $z^T_J(u)$ and the function $f_J(u)$ both depend on the chosen functional $J(u)$ to calculate the error. $A^*(u)(\cdot)$ is the adjoint associated to the original operator $A(u)$.

The theory of the DWR method shows that the error for a functional of the solution can be expressed by (2.9), as a space-time integral of residuals of a computed solution multiplied by weight functions associated with the solution of the linear dual problem (2.8).

$$J(u) - J(u_{h\Delta t}) = \int_0^T \left(\int_\Omega \rho \cdot \omega \cdot d\Omega \right) dt, \tag{2.9}$$

where ω is a weight function depending on the solution of an associated linear dual problem z.

For stationary problems, the application of duality techniques yields very accurate and efficient algorithms (see [11]); we only have the additional cost of solving a linear problem for the dual solution. But the use of DWR method for time-dependent problems in $\Omega \times I$ is still an open problem as recognized in [11] and [14]. The main reason is that we need to solve a dual problem backward in time from T to 0, and for doing so we must known the solution at $t = T$. Once we have calculated the primal and the dual solutions, we have to compute the integral (2.9) and give an error for each time level I_n and for each element K of the partition $\mathbb{T}_h^n$. Therefore, in multidimensional time dependent problems, there still remains the issue of having to store the topology and related calculations of as many spatial meshes as time subintervals I_n are needed to cover the integration interval $[0, T]$. Since this has to be done in a fully adaptive strategy, it requires a large amount of computational resources. To this respect, it is worth mentioning the recent work of Schmich and Vexler [61] where some progress has been done in formulating efficient DWR adaptive finite element algorithm for parabolic equations.

2.2.3 Local DWR method

We propose a new technique lying between the traditional approach to make adaptation and the classical DWR method, taking advantage of the good features of both methods. Our algorithm consists of applying the DWR technique locally in each time subinterval $I_n := (t_{n-1}, t_n]$ to the original problem with a perturbed initial condition given by the numerical solution at time t_{n-1}, as is shown in the following equation:

$$\begin{cases} \partial_t U + A(U) = f(U) & \text{in } \Omega \times (t_{n-1}, t_n], & \\ U(x, t_{n-1}) = u_{h\Delta t}^{n-1} & \text{in } \Omega & \text{Initial condition,} \\ U(x, t) = 0 & \text{on } \partial\Omega \times (t_{n-1}, t_n] & \text{Boundary condition.} \end{cases} \tag{2.10a}$$

We apply the general theory of the DWR method to calculate the error for the functional $J(U)$ at each time step. Then, the associated dual problem is

$$\begin{cases} -\partial_t z + A^*(U)(z) = f_J(U) & \text{in } \Omega \times [t_{n-1}, t_n), & \\ z(x, t_n) = z_J^n(U) & \text{in } \Omega & \text{Final condition,} \\ z(x, t) = 0 & \text{on } \partial\Omega \times [t_{n-1}, t_n) & \text{Boundary condition.} \end{cases} \tag{2.10b}$$

Note that the numerical solution of (2.4) and (2.10a) are the same $u_{h\Delta t}^n = U_{h\Delta t}^n$.

In this case, the error in the functional $J(U)$ can be also expressed as a space-time integral of residuals of a computed solution multiplied by weight functions associated with the solution of the linear dual problem. Now then, we only need to know the primal u and dual z solution in the time subinterval I_n.

$$J(U) - J(u_{h\Delta t}) = \int_{t_{n-1}}^{t_n} \left(\int_\Omega \rho \cdot \omega \cdot d\Omega \right) dt. \tag{2.11}$$

Thus, rather than controlling the error for the whole interval $[0, T]$ what we have now is a good local control of the error in each subinterval I_n. Figure 2.2 is a graphical representation of this idea where $J(u)$ denotes the output functional we wish to calculate accurately, $J(U)$ is the equivalent functional but for the locally perturbed solution in each interval I_n, and the error that we want to control is $J(U) - J(u_{h\Delta t})$.

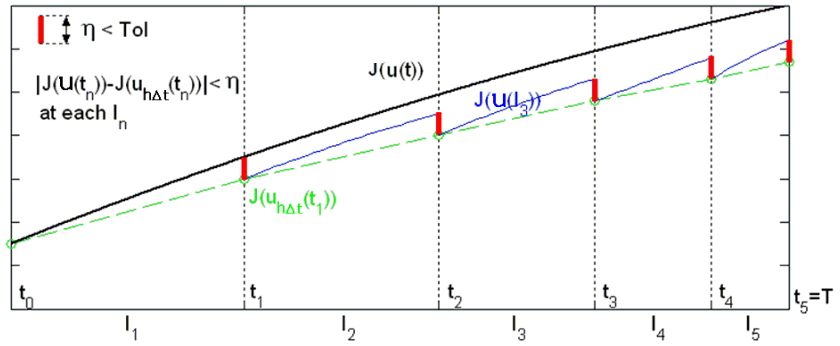

Figure 2.2: Control of the local error by application of the DWR method in each interval I_n.

The principal advantages of the method here proposed, are:

1. We extend the post-processing of the error for high order spatial finite elements for general unstructured meshes. The application will be triangular meshes in 2D but the same idea can be used with tetrahedra in 3D.

2. In time dependent problems, we estimate the local error at each time step for the functional $J(u)$, therefore:

   (a) We avoid solving the dual linear problem backward in the whole time interval I, and only have to solve a dual linear system at each time subinterval I_n.

   (b) This dual solution is self-sufficient in providing a precise criterium for adaptation of both the time step Δt and the mesh size h as the integration progresses.

   (c) In order to calculate the error, we need to store the previous and the actual mesh for the resolution of the problem; therefore, the computational resources, such as storage and CPU time, are minimal.

   (d) We approximate the time evolution of the functional $J(u(t))$ by $J(u_{h\Delta t}(t))$ evaluated in the numerical solution. One would hope that smaller local errors lead also to a decrease in the point-wise and global error between functionals $error(t) = J(u(t)) - J(u_{h\Delta t}(t))$.

However a weak point of this approach is that having a good local error control does not guarantee that the global error will be bounded as $\|J(u) - J(u_{h\Delta t})\| < GTOL$, $GTOL$ being a prescribed global tolerance, for it is a well known fact that the magnitude of the global error will depend on the stiffness of the problem. If the stiffness is low or moderate we will end up having a small global error if we control the local error well, but if the stiffness is large, the global error will be large even if we control the local error with a reasonable tolerance.

The a posteriori error estimator in the time subinterval $I_n := (t_{n-1}, t_n]$ will be derived from the primal and dual solutions $u^n_{h\Delta t}$ and $z^n_{h\Delta t}$ respectively by means of a post-processing

procedure. Thus, denoting by $\widetilde{e}^n_{sK}$ and $\widetilde{e}^n_{tK}$ the restriction to the element $K \in \mathbb{T}^n_h$ of the spatial and temporal *error estimators* respectively (after a post-processing procedure), we define the so-called *error indicators* that are used for practical adaptation η^n_{sK} and η^n_{tK}, as

$$\begin{cases} \eta^n_{sK} = \dfrac{|\widetilde{e}^n_{sK}|}{|J(u_{h\Delta t})|}, & \eta^n_s = \sum\limits_{K\in\mathbb{T}^n_h} \eta^n_{sK}, \\ \eta^n_{tK} = \dfrac{|\widetilde{e}^n_{tK}|}{|J(u_{h\Delta t})|}, & \eta^n_t = \sum\limits_{K\in\mathbb{T}^n_h} \eta^n_{tK}. \end{cases} . \tag{2.12}$$

2.3 Selection of h_K and Δt_n

In time dependent problems, there are several strategies to adapt both the spatial mesh and the size of the time step. The one we have chosen is reliable and efficient. It is reliable because it guarantees that the estimated error is below the prescribed tolerance, and it is efficient because it leads to economical meshes well adapted to the characteristics of the solution. Prescribing a tolerance $TOL = Tol_s + Tol_t$, where Tol_s and Tol_t are tolerances to check space and time errors respectively, the numerical solution $u^n_{h\Delta t}$, calculated in the partition $\mathbb{T}^n_h$ with the time step length Δt_n, is acceptable up to the tolerance TOL if

$$\eta^n_s \leq Tol_s \quad and \quad \eta^n_t \leq Tol_t.$$

To equidistribute the space and time errors one chooses $Tol_s \approx Tol_t$. Here, we propose a criterion to select the optimal size of the elements h_K (see [11] and [20]) in the spatial triangulation $\mathbb{T}^n_h$ and the time step size Δt_n (see [25]) to have an error indicator close to but lower than the fixed tolerance.

2.3.1 Spatial adaptation: Mesh-optimization strategy

Several adaptive strategies are proposed in the literature, all of them providing criteria to ascertain which mesh elements should be marked for refinement or coarsening and the number of times that we need to make it. All strategies are based on the idea of Babuska and Rheinboldt [4] that a mesh is almost optimal when the local errors are approximately equal on all its elements. So, elements where the error indicator is large will be marked for refinement, while elements with a small error indicator are coarsened or left unchanged.

We adapt the spatial mesh by using the *mesh-optimization strategy* presented in [11] and [20]. The idea is to find at each time level n the mesh with the smallest number of elements NE such that the error $\eta^n_s \leq Tol_s$. Clearly, this is an optimization problem where both NE and η^n_s will depend on h (the length of the elements). Assuming that the mesh $\mathbb{T}^n_h$ can be characterized by a continuous mesh-size function $h(x)$, the spatial error η^n_s and the number of elements NE can be written as a continuous limit of the form:

$$\begin{cases} \eta^n_s = \sum\limits_{K\in\mathbb{T}^n_h} \eta^n_{sK} \simeq \int_\Omega h(x)^\alpha \phi(x) dx =: \eta^n_s(x), \\ NE = \sum\limits_{K\in\mathbb{T}^n_h} h^d_K h^{-d}_K \simeq \int_\Omega h(x)^{-d} dx =: NE(x), \end{cases}$$

where:

- d is the spatial dimension of the problem, $d = 2$ *or* 3.

- α is the convergence rate of the spatial error, assuming $\eta_s^n = O(h^\alpha)$.

As we can see in [11] and [20], the solution of the optimization problem for h_{opt} is given by

$$h_{opt}(x) = \left(\frac{Tol_s}{W}\right)^{1/\alpha} \phi(x)^{-1/(\alpha+d)},$$

where the function $W := \int_\Omega \phi^{\frac{d}{d+\alpha}} dx < \infty$. With the discrete approximation $h|_K = h_K$ piecewise constant at each triangle K, we can write:

$$\begin{cases} \eta(x) := \int\limits_\Omega h(x)^\alpha \phi(x) dx \quad \Rightarrow \quad \phi_K \approx \dfrac{\eta_K^n}{h_K^{\alpha+d}}, \\ W := \int_\Omega \phi^{\frac{d}{d+\alpha}} dx \approx \sum_K h_K^d \phi^{\frac{d}{d+\alpha}} = \sum_K h_K^d \left(\dfrac{\eta_K^n}{h_K^{\alpha+d}}\right)^{\frac{d}{d+\alpha}} = \sum_K (\eta_K^n)^{\frac{d}{d+\alpha}}. \end{cases}$$

In this way, the optimal size for each triangle takes the form:

$$h_K^{opt} = h_K \left(\frac{Tol_s}{W}\right)^{1/\alpha} (\eta_K^n)^{-1/(\alpha+d)}. \tag{2.13}$$

Formula (2.13) gives a criterium to refine or coarsen the elements K of the triangular mesh $\mathbb{T}_h^n$. Specifically, by comparing the size of our actual triangle h_K with the optimal size h_K^{opt} we obtain the number of times that this element need to be refined or coarsened. We have to use the idea that we need to refine a triangle twice if we want to reduce its size by two, and analogously in the coarsening procedure, since we use the bisection criterion by the largest edge to divide triangles as we will see in section 2.4.

1. **Refining criterion:**

   If $\dfrac{h_K^{opt}}{h_K} \leq 1$,

   mark the element K to refine n_r times

$$n_r = Integer\ part \left[0.5 + 2\frac{\log(h_K/h_K^{opt})}{\log 2}\right]. \tag{2.14a}$$

2. **Coarsening criterion:**

   If $\dfrac{h_K^{opt}}{h_K} > 1$,

   mark the element K to coarsen n_c times

$$n_c = Integer\ part \left[2\frac{\log(h_K^{opt}/h_K)}{\log 2}\right]. \tag{2.14b}$$

At this point, several remarks are in order.

- For a general functional $J(U)$, the value of α in (2.13) is not known; however, there are cases in which the functional $J(U)$ is the $L^2 - norm$ error or the energy-norm error and then an a priori error estimator may yield values for α; but in a general case, α has to be estimated recursively. To do so, we use the formula

$$\alpha^{(i+1)} = \begin{cases} \alpha^{(i)} \dfrac{\log(\eta_s^{n(i)}/\eta_s^{n(i+1)})}{\log(\eta_s^{n(i)}/Tol_s)} & \text{if } \eta_s^{n(i+1)} < \eta_s^{n(i)}, \\ \alpha^{(i)} & \text{otherwise,} \end{cases}$$

  where i denotes the iteration number in the adaptive algorithm. It is not convenient in stationary problems to achieve a very good mesh in just one iteration because this will generate elements that are not really necessary. Therefore, it makes sense to start with a high value for α, say, $6 \leq \alpha \leq 8$, and let the adaptive algorithm to bring α near to its optimal value in a few iterations. Numerical experiments with Poisson equation show that the number of iterations of the adaptive algorithm to obtain the optimal solution is 2 or 3.

- When a mesh is close to the optimal one and $\alpha \in [0.75, 1.5]\alpha_{opt}$, the mesh-optimization strategy will give a good mesh in only one step. This fact is relevant for time dependent problems because the mesh changes somewhat smoothly from one step to the following.

- In the numerical experiments of this thesis, we have limited the number of refinements, n_r, or coarsenings, n_c, to 5 in each iteration. The purpose of limiting the number of refinements or coarsenings in each iteration is to guarantee the existence of smooth transition regions, thus avoiding the existence of patches in the mesh.

2.3.2 Time adaptation

The estimate η_t^n is used to adjust the size of the new time step and to determine whether the solution $u_{h\Delta t}^n$ is accepted or not. The current solution will be accepted if $\eta_t^n \leq Tol_t$, otherwise, it will be rejected and re-done. Similarly as we did in mesh-optimization, the adaptation of the time step size can be formulated as an optimization problem in which the parameter Tol_t will play the role of the Lagrange multiplier [25]. This approach, further limited by the condition that the size of the time step must not change too fast, leads to the formula

$$\Delta t_{new}^I = \min\left(fac_{\max},\ \max\left(fac_{\min}, \left(\frac{\kappa Tol_t}{\eta_t^n}\right)^{\frac{1}{\beta}}\right)\right)\Delta t_{old}, \tag{2.15a}$$

where:

- β is the order of the local truncation error, $\eta_t = O(\Delta t^\beta)$,

- $fac_{\max}$ and $fac_{\min}$ are factors limiting the maximum and minimum step sizes respectively, usually $fac_{\max} = 5$ and $fac_{\min} = 0.2$,

- κ is a security factor to prevent unnecessary rejections because they cause recomputation and, therefore, loss of performance. In our examples $\kappa = 0.7$.

In stiff problems the formula (2.15a) is not sufficient to achieve a nice behaviour of the time step size. This is due to over-corrections, for it may happen that the error estimator in one step be low, thus leading to a new step size too large for the calculation of the following time level. Consequently, the calculation must be rejected and re-done, initiating perhaps a series of wasteful increases and decreases. This undesirable behaviour is corrected by using the control-based approach of Gustafsson [44]. Then, the formula we use in the numerical examples is then

$$\Delta t_{new} = \min(5, \max(0.2, fac))\Delta t_{old},$$
$$fac = \begin{cases} \left(\left(\frac{\eta_t^{n-1}}{\eta_t^n}\right)^{1/\beta} \frac{\Delta t_n}{\Delta t_{n-1}}\right)\left(\frac{0.7Tol_t}{\eta_t^n}\right)^{1/\beta} & \text{when } u^n_{h\Delta t} \text{ is accepted,} \\ \left(\frac{0.7Tol_t}{\eta_t^n}\right)^{1/\beta} & \text{when } u^n_{h\Delta t} \text{ is rejected.} \end{cases} \tag{2.15b}$$

The coefficient β is not known, and we calculate it by the recursive procedure

$$\begin{cases} if\ 1 < \beta^n < 8 & \beta^n = \log\left(\frac{\eta_t^{n-1}}{\eta_t^n}\right) / \log\left(\frac{\Delta t_{n-1}}{\Delta t_n}\right), \\ else & \beta^n = \beta^{n-1}. \end{cases}$$

Formula (2.15b) gives very good results in the numerical examples and similar formulas for fac and β are used by other authors as [52].

2.4 Refining and coarsening

In this section, we describe the basic algorithms for the local refinement and coarsening of triangular meshes. To construct a sequence of conforming triangulations as we have defined in 2.1.1, we consider the following situation:

- An initial (coarse) triangulation $\mathbb{T}_{h,0}$ of the domain is given. We call it macro triangulation. It may be generated by hand or by some mesh generation algorithm, although it must be a conforming triangulation.
- To generate the mesh $\mathbb{T}_{h,i}$, some of the elements of the previous mesh $\mathbb{T}_{h,i-1}$ are marked to be refined depending on some error indicator as we have shown in 2.3. The marked elements are then refined, i.e. they are cut into smaller ones. After several refinements, some other triangles may be marked for coarsening. Coarsening tries to unite several triangles marked for coarsening into a bigger one, but elements of the macro triangulation can not be coarsened. A successive refinement and coarsening will produce a sequence of conforming triangulations $\mathbb{T}_{h,1}$, $\mathbb{T}_{h,2}$, ... generated from the given macro triangulation $\mathbb{T}_{h,0}$.

The refinement and coarsening routines construct a sequence of nested meshes with a hierarchical structure. Here, the recursive refinement by bisection is implemented following the ideas of ALBERTA code [62] and using the notation of Kossaczký [50]. The bisectioning refinement of elements naturally leads to nested meshes with hierarchical structure of binary trees, one tree for every element of the macro triangulation. Every interior element (father) of that tree is divided into two triangles (children) with the leaf elements of the trees being part of the current triangulation, which is used to define the finite element space.

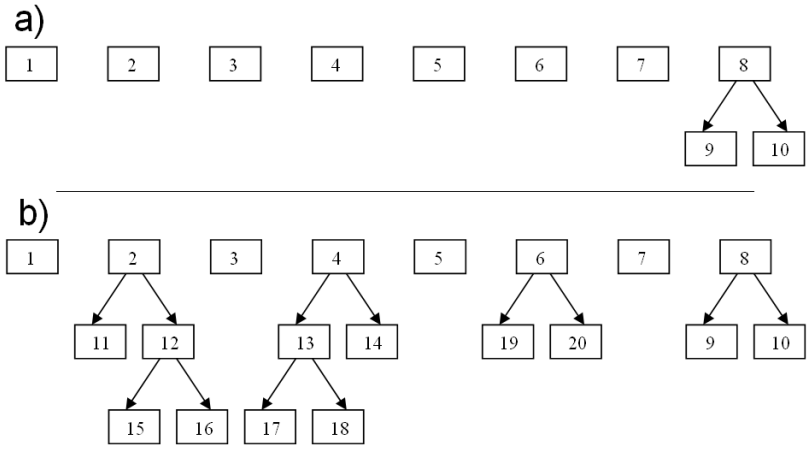

Figure 2.3: Tree structure for the situations shown in the Figure 2.6.

2.4.1 Refinement algorithm

For triangular elements, several refinement algorithms are widely used. One example is regular refinement ("red refinement"), which divides every triangle into four similar triangles. Unfortunately, hanging nodes arise during local regular refinement. To remove them and generate a conforming mesh in two dimensions, some triangles have to be bisected ("green closure").

The possibility that we use in this work is the bisection technique. For every element one of its edges is marked as the refinement edge, and the element is refined into two elements by cutting this edge at its midpoint. There are several possibilities to choose the refinement edge for a triangle; we use the longest edge as refinement edge (Mitchell in [54] compared different approaches).

In two dimensions we employ the newest vertex bisection (in Mitchell's notation [54]). We assume that all vertices of an element have fixed local indices: 0, 1 and 2. Now, the refinement edge for an element is fixed to be the edge between the vertices with local indices 0 and 1.

During refinement, the new vertex numbers, and therefore the refinement edges for the newly created child triangles, are prescribed by the refinement algorithm. For both children elements, the index of the newly generated vertex at the midpoint of the refinement edge has the highest local index (2 for triangles) and the other vertex are renumbered counterclockwise. These procedures are shown in Figure 2.4.

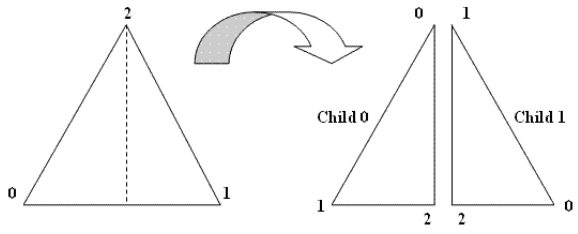

Figure 2.4: Numbering of nodes on parent and children for triangles.

By the above algorithm, the refinements of simplices are totally determined by local vertex numbering on the macro triangulation. The numbering and the definition of the refinement edge

during refinement of the elements guarantee that always the longest edge will be the refinement edge and will be bisected.

In order to keep the mesh conforming during refinement, the bisection of an edge is allowed only when such an edge is the refinement edge of all elements which share it. Bisection of an edge and, thus, of all the elements around the edge is the atomic refinement operation, and no other refinement operation is allowed, see Figure 2.5.

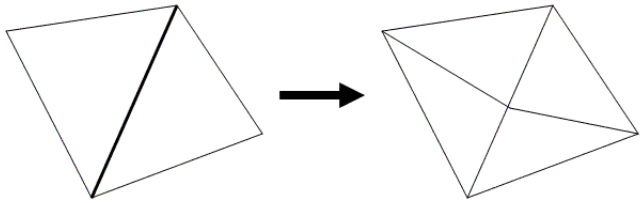

Figure 2.5: Atomic refinement operation in two dimensions. The common edge is the refinement edge for both triangles.

If an element has to be refined, we have to collect all elements at its refinement edge. In two dimensions this is either the neighbour opposite this edge or there is no other element in the case that the refinement edge belongs to the boundary. We can refine at the same time the two triangles or one by the boundary edge, by inserting a new vertex at the midpoint of the common refinement edge and bisecting every element of the patch. The resulting triangulation then is a conforming one.

But sometimes the refinement edge of a neighbour is not the common edge, then is necessary to use a recursive algorithm to refine first the neighbouring triangle by the refinement edge.

If we consider that:
$$\begin{cases} \text{The domain is } \Omega \text{ and its boundary is } \Gamma, \\ \text{The refinement edge of the element } K \text{ is } \Gamma_K^*. \\ K' \text{ is the neighbouring triangle by the refinement edge of } K \end{cases}$$

The recursive algorithm to refine a triangle K belonged to a triangulation $\mathbb{T}_h$ is the following:

subroutine *recursive_refine(K)*
{
  if ($\Gamma_K^* \in \Gamma$)
    boundary_refine(K)
  $else$
    if ($\Gamma_K^* = \Gamma_{K'}^*$)
      atomic_refinement(K, K')
    $else$
      recursive_refinement(K')
      Select the child of K' such that $\Gamma_K^* = \Gamma_{Child\_K'}^*$
      atomic_refinement(K, $Child\_K'$)
    $end\,if$
  $end\,if$
}

In Figure 2.6 we show a two-dimensional situation where recursion is needed. For all triangles, the longest edge is the refinement edge. Let us assume that triangles 8 and 6 are marked for refinement. In part a) of the figure, triangle 8 can be refined straightway, since its refinement edge is a boundary edge. For the refinement of triangle 6, in part b) of the figure, we have to refine recursively triangles 4 and 2. Again, triangle 2 can be directly refined, so recursion stops there. Back in triangle 4, this can now be refined together with its neighbours. After this, also triangle 6 can be refined together with its neighbours.

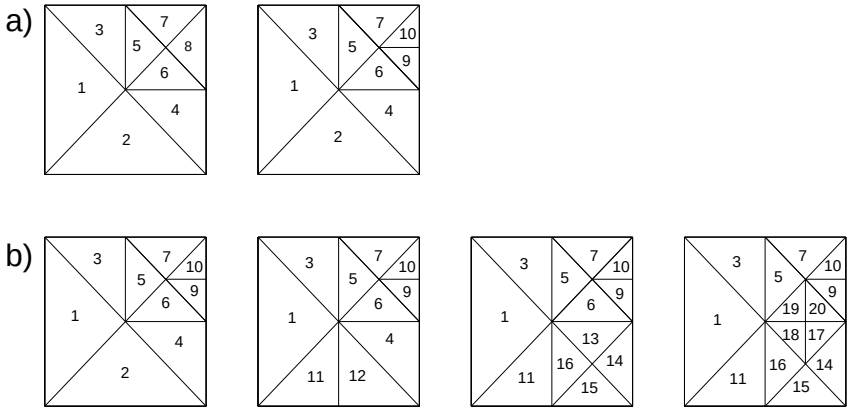

Figure 2.6: Recursive refinement in two dimensions. Triangles 8 and 6 are initially marked for refinement.

Since we use recursion, we have to guarantee that recursions terminates. Kossacký [50] and Mitchell [54] proved that it terminates and the regular shape of all elements at all levels is maintained if the macro triangulation fulfills certain criteria.

Remark:

1. A first observation is that elements initially not marked for refinement are indeed bisected, enforced by the refinement of marked elements. This is necessary to obtain a conforming triangulation, also in a regular refinement.

2. It is possible to mark an element for more than one bisection. The natural choice is to mark a element K for n_r bisections. After n_r refinement steps all original edges of K are bisected. An element K is refined n_r times refining its children $Child_0\_K$ and $Child_1\_K$ $n_r - 1$ times right after the refinement of K.

3. The recursion does not terminate for an arbitrary choice of refinement edges on the macro triangulation. In two dimensions, such a situation is shown in Figure 2.7. The selected refinement edges of the triangles are shown by dashed lines. One can easily see, that there are no patches for the atomic refinement operation. This triangulation can only be refined if other choices of refinement edges are made, or by a non-recursive algorithm.

4. In two dimensions, for every macro triangulation it is possible to choose the refinement edges in such a way that the recursion terminates (for example selecting the 'longest edge'). In three dimensions the situation is more complicated. But there may be a refined grid such that refinement edges can be chosen in such a way that recursion terminates [50].

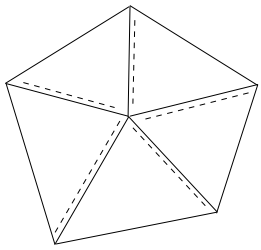

Figure 2.7: A macro triangulation where recursion does not stop.

2.4.2 Coarsening algorithm

The coarsening algorithm is approximately the inverse of the refinement algorithm. The basic idea is to collect all those elements that were created during the refinement at the same time, i.e. the parents of these elements build a compatible refinement patch. The elements must only be coarsened if all involved elements are marked for coarsening and if they are not elements of the macro triangulation. The actual coarsening again can be performed in an atomic coarsening operation without the handling of hanging nodes. Information is passed from all elements on the parents and the whole patch is coarsened at the same time by removing the vertex in the parent's common refinement edge (see Figure 2.8).

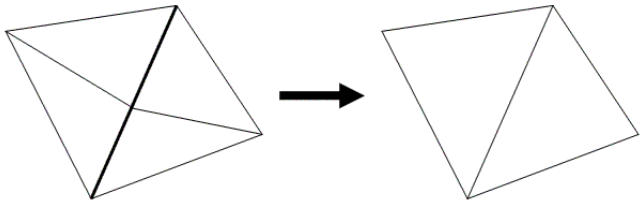

Figure 2.8: Atomic coarsening operation in two dimensions.

During refinement, the bisection of an element can enforce the refinement of an unmarked element in order to keep the mesh conforming. During coarsening, an element must only be coarsened if all elements involved in this operation are marked for coarsening, this is the main difference between refinement and coarsening. An adaptive method guarantees that elements with a large local error indicator marked for refinement are refined and no element is coarsened when the local error indicator is not small enough.

Since the coarsening process is the inverse of the refinement, refinement edges on parent elements are again at their original position. Thus, further refinement is possible with a terminating recursion and shape regularity for all resulting elements.

If we consider that:
$$\begin{cases} \text{The domain is } \Omega \text{ and its boundary is } \Gamma, \\ P_K \text{ is the father element of } K \\ \text{The refinement edge of the element } P_K \text{ is } \Gamma^*_{P_K}. \\ P_{K'} \text{ is the neighbouring triangle by the refinement edge of } P_K \end{cases}$$

A triangle K can be coarsened if
$$\begin{cases} K \text{ is a leaf and it is marked for coarsen} \\ K \text{ is a father and all of its leaves are marked for coarsen} \end{cases}$$

The recursive algorithm to coarsen a triangle K belonged to a triangulation $\mathbb{T}_h$ is the following:

subroutine *recursive_coarsen(K)*
{
  if ($P_K \neq \emptyset$) K does not belong to the macro triangulation
    if (($Child_0\_P_K$ can be coarsened) and ($Child_1\_P_K$ can be coarsened))
      if ($\Gamma^*_{P_K} \in \Gamma$)
        boundary_coarsening(P_K)
      $else$
        if (($Child_0\_P_{K'}$ can be coarsened) and ($Child_1\_P_{K'}$ can be coarsened))
          atomic_coarsening(P_K, $P_{K'}$)
        $end\,if$
      $end\,if$
    $end\,if$
  $end\,if$
}

Chapter 3

REACTION-DIFFUSION EQUATIONS

We will introduce here the main concepts of the Dual Weighted Residual method (or shortly DWR method) for goal-orientated error estimation in the Galerkin approximation of variational problems. To fix ideas and make the presentation simple, we shall consider the semi-linear reaction-diffusion equation

$$\begin{cases} \partial_t u - \epsilon \Delta u = f(u) \ \text{ in } \ \Omega \times (0,T], \\ \\ u(x,0) = u^0(x) \qquad \text{in } \ \Omega, \\ \\ u(x,t) = 0 \ \text{ in } \ \partial\Omega \times (0,T], \end{cases} \tag{3.1}$$

where ∂_t stands for $\dfrac{\partial}{\partial t}$; $T > 0$ and $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) is an open bounded domain with sufficiently smooth boundary $\partial\Omega$; the diffusion parameter $\epsilon > 0$ is assumed to be constant; the nonlinear reaction term $f : C^l(\mathbb{R}) \to \mathbb{R}$, l integer ≥ 1, satisfies suitable growth conditions [66] and the initial condition $u^0(x) \in L^2(\Omega)$. Under these assumptions there is an unique weak solution $u \in L^2(0,T; H_0^1(\Omega)) \cap L^p(\Omega \times (0,T)) \cap C([0,T]; L^2(\Omega))$ to problem (3.1), p being an integer $p \geq 2$. The weak solution $u(x,t)$ satisfies for all $v \in L^p(0,T; H_0^1(\Omega))$, $\partial_t v \in L^{p'}(0,T; H^{-1}(\Omega))$ and $v(x,T) = 0$,

$$-\int_0^T \int_\Omega u \partial_t v d\Omega dt + \int_0^T \int_\Omega \epsilon \nabla u \cdot \nabla v d\Omega dt = \int_\Omega u^0(x) v(x,0) d\Omega dt + \int_0^T \int_\Omega f(u) v d\Omega dt \tag{3.2}$$

for almost every $t \in (0,T]$. Here, H^{-1} is the dual space of $H_0^1(\Omega)$ and p' is the conjugate of p. Note the inclusions $H_0^1(\Omega) \subset\subset L^2(\Omega) \subset H^{-1}$. To simplify the notation in the formulas and equations that follow, we define the bilinear continuous form $a : H_0^1(\Omega) \times H_0^1(\Omega) \to \mathbb{R}$ as

$$a(u,v) = \int_\Omega \epsilon \nabla u \cdot \nabla v d\Omega$$

and denote the inner product in $L^2(\Omega)$ by $(b,c)_\Omega$.

For the time discretization, we split the time interval $I = (0,T]$ into half-open subintervals $I_n = (t_{n-1}, t_n]$ of length $\Delta t_n = t_n - t_{n-1}$,

$$0 = t_0 < t_1 < ... < t_{n-1} < t_n < ... < t_N = T, \qquad n = 1, 2, ...N,$$

such that $\overline{I} = \sum_{n=1}^N I_n$, and using the notation

$$v^{n+}(x) := \lim_{t \to t_n^+} v(x,t), \qquad v^{n-}(x) := \lim_{t \to t_n^-} v(x,t), \qquad [v]^n := v^{n+}(x) - v^{n-}(x),$$

it is easy to see that for all I_n, with v being zero outside the interior of I_n, the weak solution of (3.2) satisfies

$$\int_{I_n} \{\langle \partial_t u, v\rangle_\Omega + a(u,v)\}\, dt + ([u]^{n-1}, v^{n-1+})_\Omega = \int_{I_n} \langle f(u), v\rangle_\Omega\, dt, \tag{3.3}$$

where $\langle\cdot,\cdot\rangle_\Omega$ denotes the duality pairing for H^{-1} and $H_0^1(\Omega)$.

To calculate a numerical approximation $u_{h\Delta t}(x,t)$ to the weak solution $u(x,t)$ in each interval I_n, we calculate a regular triangulation $\mathbb{T}_h^n$ and define the finite element space $V_h^n \subset H_0^1(\Omega)$ associated to it as

$$V_h^n = \{v_h \in C^0(\overline{\Omega}) : v_h|_K \in P_m(K)\ \forall K \in \mathbb{T}_h^n\}.$$

Likewise, extending the spatial meshes $\mathbb{T}_h^n$ constant in time to the time slabs $S_n = \Omega \times I_n$, we can define the trial and test spaces for the pair of integer indexes (m,r) and (m,s):

$$\begin{aligned} V_{h\Delta t}^{(r)} = \{&\varphi_{h\Delta t} : \overline{\Omega}\times\overline{I} \to \mathbb{R} :\ \forall n \text{ and } (x,t) \in S_n,\ \varphi_{h\Delta t} \in C(\overline{I}_n; V_h^n),\ \varphi_{h\Delta t}(x,0) \in V_h^0 \\ &\text{and }\ \varphi_{h\Delta t}(x,\cdot)|_{\overline{I}_n} \ \in\ P_r\}, \end{aligned} \tag{3.4}$$

$$\begin{aligned} W_{h\Delta t}^{(s)} = \{&\psi_{h\Delta t} : \overline{\Omega}\times\overline{I} \to \mathbb{R} :\ \forall n \text{ and } (x,t) \in S_n,\ \psi_{h\Delta t} \in L^p(I_n; V_h^n),\ \psi_{h\Delta t}(x,0) \in V_h^0 \\ &\text{and }\ \psi_{h\Delta t}(x,\cdot)|_{I_n} \ \in\ P_s\}; \end{aligned} \tag{3.5}$$

here, P_r and P_s are the set of polynomials of degrees at most r and s respectively defined on I_n. Note that for $r = 0$ the space $V_{h\Delta t}^{(0)}$ coincides with $W_{h\Delta t}^{(0)}$.

Taking at the first time step

$$u_{h\Delta t}(x,0) = P_h^0 u^0 \in V_h^0, \tag{3.6}$$

where $P_h^0 : L^2(\Omega) \to V_h^0$ is the orthogonal L^2-projector. For the next time subintervals I_n we have to project the solution $u_{h\Delta t}^{n-1}(x) \in V_h^{n-1}$ as the initial condition on the mesh $\mathbb{T}_h^n$ and we denote by $\overline{u}_{h\Delta t}^{n-1}(x)$ the interpolatory projection onto the finite element space V_h^n,

$$\overline{u}_{h\Delta t}^{n-1}(x) = I_h^n u_{h\Delta t}^{n-1}(x) \in V_h^n. \tag{3.7}$$

In the numerical examples we use a quasi-monotone interpolatory scheme of order 2 with a control of the error following ideas of the goal oriented adaptation. But this procedure and the calculation of the error will be seen in the next chapter, where we introduce them to treat the convective term in a semi-Lagrangian framework. Then, the numerical approximation $u_{h\Delta t}(x,t)$ is the solution of the following sequence of problems:

For $n = 1, 2, .., N$, find $u_{h\Delta t}(x,t) \in V_{h\Delta t}^{(r)}$ such that for all $\psi_{h\Delta t}(x,t) \in W_{h\Delta t}^{(s)}$

$$\begin{cases} \displaystyle\int_{I_n} \{(\partial_t u_{h\Delta t}, \psi_{h\Delta t})_\Omega + a(u_{h\Delta t}, \psi_{h\Delta t})\}\, dt + ([u_{h\Delta t}]^{n-1}, \psi_{h\Delta t}^{n-1+})_\Omega = \int_{I_n} (f(u_{h\Delta t}), \psi_{h\Delta t})_\Omega\, dt, \\ \\ u_{h\Delta t}(x, t_{n-1}) = \overline{u}_{h\Delta t}^{n-1}(x). \end{cases} \tag{3.8}$$

- With $r = 1$ and $s = 0$, (3.8) yields a version of the Crank-Nicolson scheme:

  For all n and for all $\psi_{h\Delta t} \in W^{(s)}_{h\Delta t}$

$$(u^n_{h\Delta t} - \overline{u}^{n-1}_{h\Delta t}, \psi_{h\Delta t})_\Omega + \frac{\Delta t_n}{2} a(u^n_{h\Delta t} + \overline{u}^{n-1}_{h\Delta t}, \psi_{h\Delta t}) = \int_{I_n} (f(u_{h\Delta t}), \psi_{h\Delta t})_\Omega \, dt \quad (3.9)$$

  and

$$u_{h\Delta t}(x,t)|_{I_n} = \overline{u}^{n-1}_{h\Delta t}(x) + \frac{t - t_{n-1}}{\Delta t_n}(u^n_{h\Delta t}(x) - \overline{u}^{n-1}_{h\Delta t}(x)).$$

- Also, as is well known, when $r = s = 0$ (3.8) yields the conventional Euler implicit scheme:

  For all n and for all $\psi_{h\Delta t} \in W^{(s)}_{h\Delta t}$

$$(u^n_{h\Delta t} - \overline{u}^{n-1}_{h\Delta t}, \psi_{h\Delta t})_\Omega + \Delta t_n a(u^n_{h\Delta t}, \psi_{h\Delta t}) = \Delta t_n \, (f(u^n_{h\Delta t}), \psi_{h\Delta t})_\Omega \quad (3.10)$$

  and

$$u_{h\Delta t}(x,t)|_{I_n} = u^n_{h\Delta t}(x).$$

To apply the DWR methodology to control the local error in each I_n we shall consider the auxiliary problem:

Find $U(x,t) \in V^n := L^2(I_n; H^1_0(\Omega)) \cap L^p(S_n) \cap C^0(\overline{I}_n; L^2(\Omega))$ such that for all $\psi(x,t) \in W^n := L^p(I_n; H^1_0(\Omega))$, $\psi(x, t_n) = 0$

$$\int_{I_n} \{\langle \partial_t U, \psi \rangle_\Omega + a(U, \psi) - \langle f(U), \psi \rangle_\Omega\} \, dt + (U^{n-1+}, \psi^{n-1+})_\Omega = (\overline{u}^{n-1}_{h\Delta t}, \psi^{n-1+})_\Omega. \quad (3.11a)$$

Here, V^n and W^n are the local restrictions of the spaces $L^2(0,T; H^1_0(\Omega)) \cap L^p(\Omega \times (0,T)) \cap C([0,T]; L^2(\Omega))$ and $L^p(0,T; H^1_0(\Omega))$, respectively, to the interval I_n and $\overline{u}^{n-1}_{h\Delta t}(x)$ is the solution of (3.8) at time instant t_{n-1}. Note that (3.11a) is the weak formulation of the problem

$$\begin{cases} \partial_t U - \epsilon \Delta U = f(U) & \text{in } \Omega \times I_n, \\ U(x, t_{n-1}) = \overline{u}^{n-1}_{h\Delta t}(x) & \text{in } \Omega, \\ U(x,t) = 0 \text{ in } \partial\Omega \times I_n. \end{cases} \quad (3.11b)$$

Next, setting $U^{n-1-} = \overline{u}^{n-1}_{h\Delta t}$ in (3.11a) we define the semi-linear form $A : V^n \times W^n \to \mathbb{R}$ as

$$A(U)(z) = \int_{I_n} \{\langle \partial_t U, z \rangle_\Omega + a(U, z) - \langle f(U), z \rangle_\Omega\} \, dt + ([U]^{n-1}, z^{n-1+})_\Omega \quad (3.12)$$

and choose an output functional $J : V^n \to \mathbb{R}$, such that we can define the Lagrangian $\mathcal{L} : V^n \times W^n \to \mathbb{R}$ as

$$\mathcal{L}(U; z) := J(U) - A(U)(z).$$

Then, we calculate the stationary points $(U, z) \in V^n \times W^n$ of $\mathcal{L}(U; z)$ which are solution of

$$\mathcal{L}'(U; z)(\varphi, \psi) = 0;$$

that is, considering that U^{n-1-} is a known function (as we have just said before), we have to find the pair $(U, z) \in V^n \times W^n$ that satisfies

$$-\int_{I_n} \{\langle \partial_t U, \psi\rangle_\Omega + a(U,\psi) - \langle f(U), \psi\rangle_\Omega\}\, dt - ([U]^{n-1}, \psi^{n-1+})_\Omega = 0 \ \ \forall \psi \in W^n, \quad (3.13)$$

and

$$J'(U)(\varphi) - \int_{I_n} \{\langle \partial_t \varphi, z\rangle_\Omega + a(\varphi, z) - \langle f'(U)\varphi, z\rangle_\Omega\}\, dt - (\varphi^{n-1+}, z^{n-1+})_\Omega = 0, \ \ \forall \varphi \in V^n, \quad (3.14)$$

where, by noting that

$$\int_{I_n} \langle \partial_t \varphi, z\rangle_\Omega\, dt = -\int_{I_n} \langle \partial_t z, \varphi\rangle_\Omega\, dt + (\varphi^{n-}, z^{n-})_\Omega - (\varphi^{n-1+}, z^{n-1+})_\Omega,$$

yields

$$J'(U)(\varphi) - \int_{I_n} \{-\langle \partial_t z, \varphi\rangle_\Omega + a(z,\varphi) - \langle f'(U)\varphi, z\rangle_\Omega\}\, dt - (\varphi^{n-}, z^{n-})_\Omega = 0, \ \forall \varphi \in V^n. \quad (3.15)$$

Problems (3.13) and (3.15) are termed *primal* and *dual problems* respectively in the slab S_n. Note that for the primal problem we have that $U^{n-1-} = \overline{u}^{n-1}_{h\Delta t}$, so that, this problem is the weak formulation (3.11a) of the auxiliary problem (3.11b).

The Galerkin approximation, $(U_{h\Delta t}, z_{h\Delta t}) \in V^{(r)}_{h\Delta t} \times W^{(s)}_{h\Delta t}$, to the primal and dual problems by time-space finite elements in each slab S_n satisfies for all $(\varphi_{h\Delta t}, \psi_{h\Delta t}) \in V^{(r)}_{h\Delta t} \times W^{(s)}_{h\Delta t}$ the equation

$$\mathcal{L}'(U_{h\Delta t}; z_{h\Delta t})(\varphi_{h\Delta t}, \psi_{h\Delta t}) = 0;$$

that is, $(U_{h\Delta t}, z_{h\Delta t})$ is the unique solution of the following problem:

For each I_n, find the pair $(U_{h\Delta t}, z_{h\Delta t}) \in V^{(r)}_{h\Delta t} \times W^{(s)}_{h\Delta t}$ such that for all $(\varphi_{h\Delta t}, \psi_{h\Delta t}) \in V^{(r)}_{h\Delta t} \times W^{(s)}_{h\Delta t}$

$$-\int_{I_n} \{(\partial_t U_{h\Delta t}, \psi_{h\Delta t})_\Omega + a(U_{h\Delta t}, \psi_{h\Delta t}) - (f(U_{h\Delta t}), \psi_{h\Delta t})_\Omega\}\, dt - ([U_{h\Delta t}]^{n-1}, \psi^{n-1+}_{h\Delta t})_\Omega = 0, \quad (3.16)$$

and

$$J'(U_{h\Delta t})(\varphi_{h\Delta t}) - \int_{I_n} \{(-\partial_t z_{h\Delta t}, \varphi_{h\Delta t})_\Omega + a(z_{h\Delta t}, \varphi_{h\Delta t}) - (f'(U_{h\Delta t})\varphi_{h\Delta t}, z_{h\Delta t})_\Omega\}\, dt - (\varphi^{n-}_{h\Delta t}, z^{n-}_{h\Delta t})_\Omega = 0. \quad (3.17)$$

Since (3.16) coincides with (3.8) in each interval I_n; then, the solution $U_{h\Delta t}$ is precisely the finite element solution $u_{h\Delta t}$ to problem (3.1) for the time interval I_n. From (3.16) and (3.17) together with Proposition 6.1 of [11] (that can be seen in the Appendix) is easy to obtain the following result.

Proposition 1. *For each I_n, let (U, z) and $(u_{h\Delta t}, z_{h\Delta t})$ be solutions of ((3.13), (3.15)) and ((3.16),(3.17)) respectively. Assume that the functional $J : V^n \to \mathbb{R}$ and the semi-linear form $A : V^n \times W^n \to \mathbb{R}$ have directional derivatives up to order three. Then, we have the following error representation*

$$J(U) - J(u_{h\Delta t}) = \frac{1}{2}\rho(u_{h\Delta t})(z - \psi_{h\Delta t}) + \frac{1}{2}\rho^*(u_{h\Delta t}, z_{h\Delta t})(U - \varphi_{h\Delta t}) + \mathcal{R}^{(3)}_{h\Delta t}, \quad (3.18)$$

where the primal residual $\rho(u_{h\Delta t})(\cdot)$ and the dual residual $\rho^*(u_{h\Delta t}, z_{h\Delta t})(\cdot)$ are given in terms of the element residuals $R_{h\Delta t}$ and $R^*_{h\Delta t}$, and the edge residuals $r_{h\Delta t}$ and $r^*_{h\Delta t}$ as:

$$\begin{cases} \rho(u_{h\Delta t})(\cdot) = \sum\limits_{K\in\mathbb{T}_h^n} \int_{I_n} \{(R_{h\Delta t},\cdot)_K + (r_{h\Delta t},\cdot)_{\partial K}\}\, dt - ([u_{h\Delta t}]^{n-1},(\cdot)^{n-1+})_K, \\ R_{h\Delta t} = f(u_{h\Delta t}) - \partial_t u_{h\Delta t} + \epsilon\Delta u_{h\Delta t} \quad \text{and} \quad r_{h\Delta t} = \begin{cases} \frac{\epsilon}{2}\,[\partial_n u_{h\Delta t}]_\Gamma \ \text{ if } \ \Gamma\subset \partial K\backslash\partial\Omega, \\ 0 \ \text{ if } \ \Gamma\subset\partial\Omega. \end{cases} \end{cases} \tag{3.19}$$

and

$$\begin{cases} \rho^*(u_{h\Delta t}, z_{h\Delta t})(\cdot) = \sum\limits_{K\in\mathbb{T}_h^n} \int_{I_n} \{(R^*_{h\Delta t},\cdot)_K + (r^*_{h\Delta t},\cdot)_{\partial K}\}\, dt - J'(u_{h\Delta t})(\cdot)_K - (z^n_{h\Delta t},(\cdot)^{n-})_K, \\ R^*_{h\Delta t} = f'(u_{h\Delta t}) z_{h\Delta t} + \partial_t z_{h\Delta t} + \epsilon\Delta z_{h\Delta t} \quad \text{and} \quad r^*_{h\Delta t} = \begin{cases} \frac{\epsilon}{2}\,[\partial_n z_{h\Delta t}]_\Gamma \ \text{ if } \ \Gamma\subset \partial K\backslash\partial\Omega, \\ 0 \ \text{ if } \ \Gamma\subset\partial\Omega. \end{cases} \end{cases} \tag{3.20}$$

where $[\partial_n u_{h\Delta t}]_\Gamma$ (resp. $[\partial_n z_{h\Delta t}]_\Gamma$) denotes the jump of gradient $\nabla u_{h\Delta t}$ (resp. $\nabla z_{h\Delta t}$) across the inter-element edges, i.e., for two neighboring cells K and K' with common edge Γ and normal unit vector $\mathbf{n}$ pointing from K to K' we set

$$[\partial_n u_{h\Delta t}]_\Gamma = [\mathbf{n}\cdot\nabla u_{h\Delta t}]_\Gamma := \mathbf{n}\cdot\left(\nabla u_{h\Delta t}|_{K'\cap\Gamma} - \nabla u_{h\Delta t}|_{K\cap\Gamma}\right).$$

The reminder term

$$\begin{aligned} \mathcal{R}^{(3)}_{h\Delta t} = \frac{1}{2}\int_0^1 \{&J'''(u_{h\Delta t}+se)(e,e,e) + A'''(u_{h\Delta t}+se)(e,e,e,z_{h\Delta t}+se^*) - \\ &-3A''(u_{h\Delta t}+se)(e,e,e^*)\}s(s-1)ds, \end{aligned}$$

where $e = U - u_{h\Delta t}$ and $e^* = z - z_{h\Delta t}$.

The terms $(z-\psi_{h\Delta t})$ and $(U-\varphi_{h\Delta t})$ are the so called weights. The term $\mathcal{R}^{(3)}_{h\Delta t}$ vanishes only if $A(u)(\cdot)$ is linear and if $J(u)$ is quadratic, but it is usually small; so that, it will be neglected when Proposition 1 is used for mesh adaptation; nevertheless, there are problems in which $\mathcal{R}^{(3)}_{h\Delta t}$ may become large, but the DWR theoretical approach is still valid and must be applied with care [11].

Note that the a posteriori error estimate (3.18) involves the exact primal and dual solutions through the weights $(z-\psi_{h\Delta t})$ and $(U-\varphi_{h\Delta t})$. Since z and U are unknown, in practice they are estimated from their corresponding numerical solutions via a post-processing procedure. In this chapter we propose a post-processing procedure to separate the error contribution into two parts, namely, a part due to time discretization and another one due to space discretization. This splitting of errors is important because one wishes to adapt both time and space meshes and, therefore, one needs an estimator for space that, in general, will take different values that the estimator used for time.

3.1 Post-processing procedure to evaluate the weights of the residuals

In time dependent problems where the goal is to adapt both the space and time partitions (meshes), it is convenient to evaluate the contributions of time and space discretizations to

the error $J(U) - J(u_{h\Delta t})$ in each interval I_n. To do so, we shall formulate a procedure which is based on arguments presented in Chapters 4 and 5 of [11] to approximate the weights of the residuals in the application of DWR to stationary problems. The idea goes as follows. Since the main purpose of the a posteriori error analysis is to obtain estimates of the numerical errors in time and space, which are used for the formulation of criteria to adapt the meshes, then it will be sufficient to substitute $z(x,t)$ and $U(x,t)$ in (3.18) by approximations, say $\widetilde{z}_{h\Delta t}$ and $\widetilde{u}_{h\Delta t}(x,t)$, sufficiently close to $z(x,t)$ and $u(x,t)$ as to have the error estimates (in both time and space) $\|\widetilde{z}_{h\Delta t} - z_{h\Delta t}\| = O\left(\|z - z_{h\Delta t}\|\right)$ and $\|\widetilde{u}_{h\Delta t} - u_{h\Delta t}\| = O\left(\|U - u_{h\Delta t}\|\right)$ respectively. Such approximations will be obtained by some post-processing procedure of the numerical solutions $z_{h\Delta t}$ and $u_{h\Delta t}$, such as higher order methods for both time and space, although it is not clear that this approach is the best for adaptation. A second approach that is computationally efficient and may lead to satisfactory results, see the examples presented in Chapter 4 of [11], as well as the numerical example of a stationary problem below, it is the method of *patch-wise higher order interpolation recovery* in both time and space that has also be employed in [61]. To implement such a method we need to introduce the auxiliary semidiscrete primal and dual functions $u_h(x,t)$ and $z_h(x,t)$ defined as follows:

Definition 1. *(A) For $t = 0$, $u_h(x,0)$ is the L^2-projection of $u(x,0)$ onto V_h^0, and for all I_n, $u_h \in L^2(I_n; V_{h0}^n) \cap L^p(\Omega \times I_n) \cap C(\overline{I}_n; L^2(\Omega))$ satisfies*

$$-\int_{I_n} \{(\partial_t u_h, \psi_h)_\Omega + a(u_h, \psi_h) - (f(u_h), \psi_h)_\Omega\}\, dt + \left([u_h]^{n-1}, \psi_h^{n-1+}\right)_\Omega = 0 \;\; \forall \psi_h \in V_{h0}^n. \tag{3.21}$$

(B) For all I_n, $z_h \in L^2(I_n; V_{h0}^n)$ satisfies

$$J'(u_h)(\varphi_h) - \int_{I_n} \left\{(\partial_t \varphi_h, z_h)_\Omega + a(\varphi_h, z_h) - \left(f'(u_h)\varphi_h, z_h\right)_\Omega\right\} dt - \left(\varphi_h^{n-1+}, z_h^{n-1+}\right)_\Omega = 0 \;\; \forall \varphi_h \in V_{h0}^n. \tag{3.22}$$

Note that (u_h, z_h) is solution of

$$\mathcal{L}'(u_h; z_h)(\varphi_h, \psi_h) = 0 \; \forall \varphi_h, \psi_h \in V_{h0}^n.$$

Next, we consider $z - z_{h\Delta t}$ and $U - u_{h\Delta t}$, which are split as

$$\begin{cases} z - z_{h\Delta t} = (z - z_h) + (z_h - z_{h\Delta t}), \\ \\ U - u_{h\Delta t} = (U - u_h) + (u_h - u_{h\Delta t}) \end{cases}$$

and note that the terms $z - z_h$ (resp. $U - u_h$) and $z_h - z_{h\Delta t}$ (resp. $u_h - u_{h\Delta t}$) measure respectively the space and time discretization errors in the approximation to z (resp. U) by $z_{h\Delta t}$ (resp. $u_{h\Delta t}$). Based on this simple observation and recalling that we can set $\psi_{h\Delta t} = z_{h\Delta t}$ and $\varphi_{h\Delta t} = u_{h\Delta t}$ in (3.18) it follows that

$$\rho(u_{h\Delta t})(z - z_{h\Delta t}) = \rho(u_{h\Delta t})(z - z_h) + \rho(u_{h\Delta t})(z_h - z_{h\Delta t})$$

and

$$\rho^*(u_{h\Delta t}, z_{h\Delta t})(U - u_{h\Delta t}) = \rho^*(u_{h\Delta t}, z_{h\Delta t})(U - u_h) + \rho^*(u_{h\Delta t}, z_{h\Delta t})(u_h - u_{h\Delta t}),$$

Hence, collecting these results we have the following proposition.

Proposition 2. *Assuming the hypotheses of Proposition 1 hold, then the a posteriori error estimate representation for the functional $J(U)$ in each I_n is given by*

$$J(U) - J(u_{h\Delta t}) = e_s^n + e_t^n + \mathcal{R}_{h\Delta t}^{(3)}, \tag{3.23a}$$

where the time component of the error is

$$e_t^n = \tfrac{1}{2}\rho(u_{h\Delta t})(z_h - z_{h\Delta t}) + \tfrac{1}{2}\rho^*(u_{h\Delta t}, z_{h\Delta t})(u_h - u_{h\Delta t}), \tag{3.23b}$$

and the space component of the error is

$$e_s^n = \tfrac{1}{2}\rho(u_{h\Delta t})(z - z_h) + \tfrac{1}{2}\rho^*(u_{h\Delta t}, z_{h\Delta t})(U - u_h). \tag{3.23c}$$

As an immediate consequence of Proposition 2 and applying Proposition 1 to the difference between $J(u_h) - J(u_{h\Delta t})$, we obtain the following corollary.

Corollary 1. *Let*

$$J(U) - J(u_{h\Delta t}) = J(U) - J(u_h) + J(u_h) - J(u_{h\Delta t}).$$

Then, assuming the hypotheses of Proposition 1 hold, the a posteriori error estimation for the difference between the functional $J(u_h)$ and $J(u_{h\Delta t})$ in each I_n will be associated with e_t^n by

$$J(u_h) - J(u_{h\Delta t}) = e_t^n + \mathcal{R}_{\Delta t}^{(3)}, \tag{3.24a}$$

where the reminder term takes the form

$$\mathcal{R}_{\Delta t}^{(3)} = \frac{1}{2}\int_0^1 \{J'''(u_{h\Delta t} + se)(e,e,e) + A'''(u_{h\Delta t} + se)(e,e,e, z_{h\Delta t} + se^*) - \\ -3A''(u_{h\Delta t} + se)(e,e,e^*)\} s(s-1)ds,$$

with $e = u_h - u_{h\Delta t}$ and $e^* = z_h - z_{h\Delta t}$.

And the error between $J(U)$ and $J(u_h)$ will be associated with e_s^n by

$$J(U) - J(u_h) = e_s^n + \mathcal{R}_{h\Delta t}^{(3)} - \mathcal{R}_{\Delta t}^{(3)}. \tag{3.24b}$$

Notice that e_s^n and e_t^n involve the unknown terms $z_h(x,t)$, $z(x,t)$, $U(x,t)$ and $u_h(x,t)$. For practical use of this error representation we shall approximate such terms by a patch-wise higher order interpolation recovery technique applied to the numerical solutions $u_{h\Delta t}$ and $z_{h\Delta t}$. The basic idea of recovering by patch-wise higher order interpolation is as follows. If the numerical solution approximates the exact solution in space by a polynomial of degree, say m, we shall construct from the numerical solution a space approximation that is a piecewise polynomial of degree $2m$; likewise, if the numerical solution approximates the time dependence of the exact solution by a piecewise polynomial of degree, say r, then we shall use the numerical solution to construct an approximation in time that will be a piecewise polynomial of degree $r+1$. So that, the recovery technique consists of two stages, namely, recovery in space and recovery in time. First, we explain the technique to recover in space and then the technique for time.

3.1.1 Recovery technique in space

Although our algorithms are specially designed for unstructured meshes of simplices, the patch-wise higher order recovery technique in space we propose in this paper follows the spirit of the biquadratic interpolation procedure introduced in [11] for space recovery in structured meshes made of squares (or hexahedral). To describe our procedure, let the integer m be the degree of the polynomials of the finite elements we use for the computations of both the dual and primal solutions, and for all n, let $I_{2h}^{2m} : V_h^n \to V_{2h}^n$ denote the interpolation operator of degree $2m$ where

$$V_{2h}^n = \left\{ v_{2h} \in C^0(\overline{\Omega}) : v_{2h} \mid_{\widetilde{K}} \in P(\widetilde{K}) \ \ \forall \widetilde{K} \in \mathbb{T}_{2h}^n \right\},$$

$$P(\widetilde{K}) = \left\{ p(x),\ x \in \widetilde{K} :\ p(x) = \widehat{p} \circ F_{\widetilde{K}}^{-1}(x),\ \ \widehat{p}(\widehat{x}) \in P_{2m}(\widehat{K}) \right\},$$

$F_{\widetilde{K}} : \widehat{K} \to \widetilde{K}$ is the affine invertible function and $P_{2m}(\widehat{K})$ the set of polynomials of degree $2m$ defined on the reference element $\widehat{K}$. Thus, if we consider a mesh of linear triangles ($m = 1$), to define the interpolation operator I_{2h}^2 on such a mesh we associate to each element K a larger element $\widetilde{K}$ defined by six nodes as is shown on the left hand side panel of Figure 3.1. $\widetilde{K}$ is constructed from the shadowed triangle K by doubling the length of its edges.

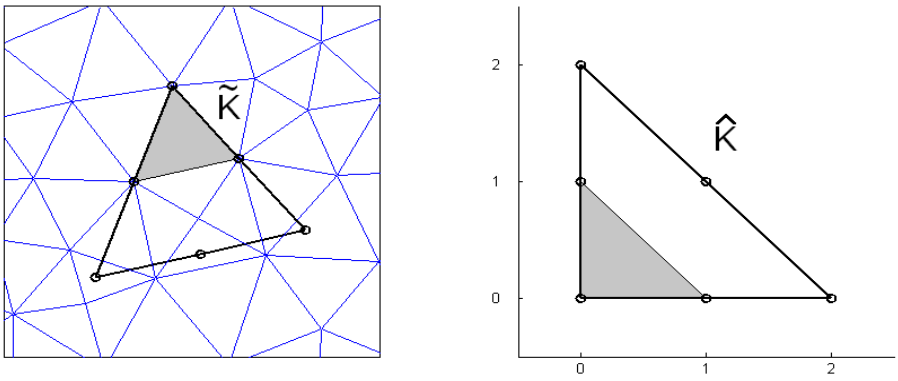

Figure 3.1: Patch-wise quadratic interpolation for linear triangles.

Similarly, if the mesh triangle K is of type ($m = 2$); that is, a quadratic triangle, the interpolation operator is for this case I_{2h}^4, and this means that the associated element $\widetilde{K}$ has to be of type ($2m = 4$); that is, a quartic triangle. Now, the element $\widetilde{K}$ is constructed as in the linear case, first doubling the size of the edges of K and then setting up the nodes, which are not vertices, as is usual in finite element technology.

We should note that the nodes of K belong to the set of nodes of $\widetilde{K}$ and, therefore, $K \subset \widetilde{K}$ by construction. This implies that the interpolation operator $I_{2h}^{2m} \mid_{\widetilde{K}}$ is also well defined on K. It is clear now the formula to calculate the coordinates of the vertices of $\widetilde{K}$ in terms of the vertices of K. For instance, let (a_1, a_2, a_3) be the vertices of K and let $(\widetilde{a}_1, \widetilde{a}_2, \widetilde{a}_3)$ be the vertices of $\widetilde{K}$. Choose a_1 to be a vertex of $\widetilde{K}$, so $\widetilde{a}_1 = a_1$, then it is easy to see that $\widetilde{a}_2 = 2a_2 - a_1$ and $\widetilde{a}_3 = 2a_3 - a_1$. One may prefer to choose a_2, o even a_3, instead of a_1 as the common vertex of K and $\widetilde{K}$, in this case the procedure to calculate the other two vertices is the same as in the previous case. Knowing the coordinates of the vertices $(\widetilde{a}_1, \widetilde{a}_2, \widetilde{a}_3)$ one proceeds as usual in finite element technology with the calculation of the coordinates of the other nodes defining the element $\widetilde{K}$.

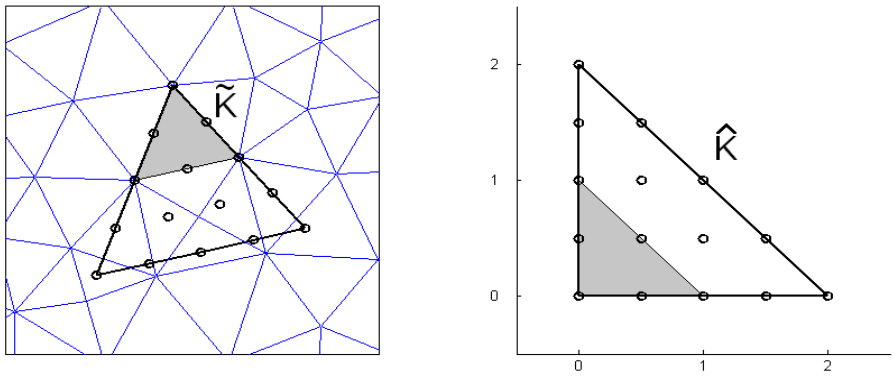

Figure 3.2: Patch-wise quartic interpolation for quadratic triangles.

Once the coordinates of all points of $\widetilde{K}$ have been calculated one needs to know the value of the solution $u_{h\Delta t}$ (and $z_{h\Delta t}$ too) at such nodes to define the interpolation operator $I_{2h}^{2m} u_{h\Delta t} \mid_{\widetilde{K}}$. Noting that $u_{h\Delta t}$ is known at those nodes of $\widetilde{K}$, which are also nodes of K, then it remains to evaluate $u_{h\Delta t}$ at the nodes which are not. To do so, we use the search-locate algorithm of [3] that consists of first searching the element K' containing the node at which one wishes to evaluate, say $u_{h\Delta t}$, and then evaluating $u_{h\Delta t}$ at this point by finite element interpolation using the local basis function associated to the element K'. In applying this procedure, we should be aware of the following situations that may appear in elements close to the boundary.

- In choosing a vertex of K, say a_1, to be also vertex of $\widetilde{K}$ one may find that some of the other vertices of $\widetilde{K}$ is outside $\overline{\Omega}$, then one should take a_2 as the common vertex and proceed as before. If the a_2-choice does not work either, then one moves to vertex a_3.
- It may happen that for any vertex of K the element $\widetilde{K}$ has vertices outside $\overline{\Omega}$, there may be one vertex or at most two vertices for triangles (or three vertices for tetrahedra), in this case one keeps $\widetilde{K}$ and the values of $u_{h\Delta t}$ assigned at the nodes outside Ω are the values of $u_{h\Delta t}$ at points inside Ω which are symmetric of the outside nodes with respect to the common vertex of K and $\widetilde{K}$. See Figure 3.3 where it is shown a typical example of a triangle K close to a corner of the domain.

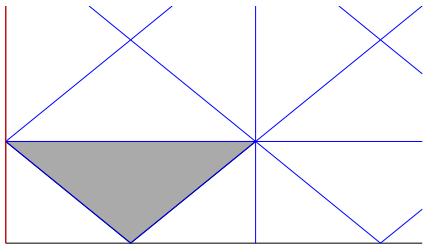

Figure 3.3: An example of a triangle close to a corner in the patch-wise higher order interpolation.

By virtue of the definitions of $u_h(x,t)$ and $z_h(x,t)$, they represent for all t the space semidiscrete approximations to $u(x,t)$ and $z(x,t)$, respectively, in the corresponding finite element spaces V_h^n. Then, the errors $z - z_h$ and $u - u_h$ are caused by the spatial discretization as in the stationary problems; hence, guided by the post-processing procedure in stationary problems

we propose for the evaluation of the space component of the error e_s^n in (3.23c) the following approximations:

$$\begin{cases} z(x,t) - z_h(x,t) \approx I_{2h}^{2m} z_{h\Delta t}(x,t) - z_{h\Delta t}(x,t), \\ \\ U(x,t) - u_h(x,t) \approx I_{2h}^{2m} u_{h\Delta t}(x,t) - u_{h\Delta t}(x,t). \end{cases} \tag{3.25}$$

3.1.2 Recovery technique in time

To recover a "good" approximation in time for the primal and dual solutions $u_h(x,t)$ and $z_h(x,t)$ from the calculated solutions $u_{h\Delta t}(x,t)$ and $z_{h\Delta t}(x,t)$ respectively, we adopt a scheme similar to the one introduced in [61]. Thus, for the sake of concreteness, we shall restrict ourselves to the cases in which the degree of time polynomials for the primal solution are 1 or 0, and for the dual solution are 0; specifically, recalling the definitions of the spaces $V_{h\Delta t}^{(r)}$, and $W_{h\Delta t}^{(s)}$ above, see (3.4) and (3.5), we shall consider the following cases:

Case1: r=s=0 (discontinuous in time both primal and dual solutions).

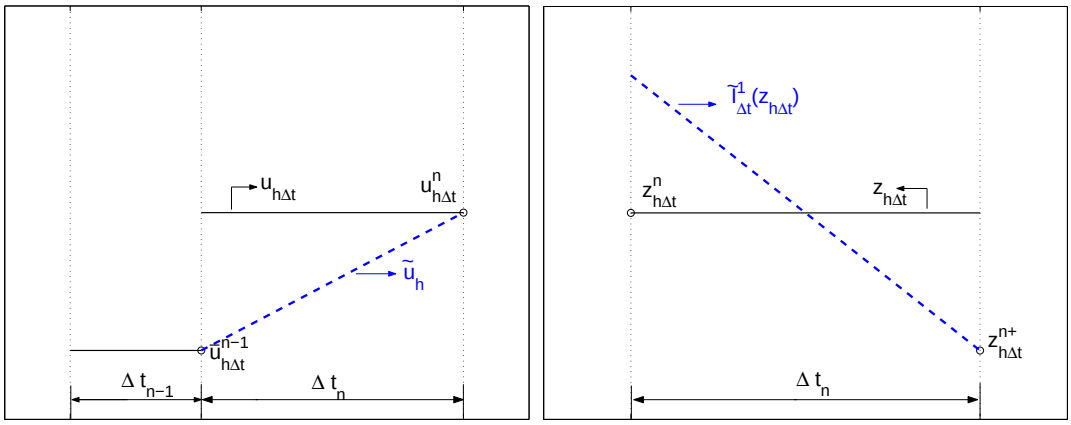

Figure 3.4: Linear recovery in time for both the primal solution when $r = 0$ (left panel) and the dual solution when $s = 0$ (right panel).

For each I_n, we have that

$$u_{h\Delta t}(x,t)|_{I_n} = u_{h\Delta t}^n(x) \;\text{ and }\; z_{h\Delta t}(x,t)|_{I_n} = z_{h\Delta t}^n(x),$$

with the particularity that $u_{h\Delta t}(x,t)$ is solution of a forward in time problem, whereas $z_{h\Delta t}(x,t_n)$ is the solution of the backward in time adjoint problem associated to (3.11a). We hypothesize that a better representation for the time dependence of the solutions $u_h(x,t)$ and $z_h(x,t)$ in each I_n is a linear polynomial instead of the polynomial of degree zero, which is the polynomial used to calculate the corresponding numerical solutions. Then the formulae to recovery in time are:

$$\begin{cases} \widetilde{I}_{\Delta t}^1 z_{h\Delta t}(x,t) = z_{h\Delta t}^{n+}(x) + 2\left(z_{h\Delta t}^n(x) - z_{h\Delta t}^{n+}(x)\right)\dfrac{t_n - t}{\Delta t_n}, \\ \\ I_{\Delta t}^1 u_{h\Delta t}(x,t) = \overline{u}_{h\Delta t}^{n-1}(x) + \dfrac{t - t_{n-1}}{\Delta t_n}\left(u_{h\Delta t}^n(x) - \overline{u}_{h\Delta t}^{n-1}(x)\right). \end{cases} \tag{3.26}$$

We show in Figure 3.4 a graphical representation of $I_{\Delta t}^1 u_{h\Delta t}(x,t)$ and $\widetilde{I}_{\Delta t}^1 z_{h\Delta t}(x,t)$ in I_n. We remark that $I_{\Delta t}^1 u_{h\Delta t}(x,t)$ is the typical linear Lagrange polynomial for the interpolation points

$(t_{n-1}, \overline{u}^{n-1}_{h\Delta t}(x))$ and $(t_n, u^n_{h\Delta t}(x))$ [61]; however, $\widetilde{I}^1_{\Delta t} z_{h\Delta t}$ is the linear polynomial for the interpolation points $(t_n, z^{n+}_{h\Delta t}(x))$ and $(t_{n-1}, \left(2z^n_{h\Delta t} - z^{n+}_{h\Delta t}\right)(x))$. Here, $z^{n+}_{h\Delta t}(x)$ denotes the initial value of the adjoint problem associated to (3.11a) which has to be solved in each I_n; this means that $z^{n+}_{h\Delta t}(x)$ will depend on the target functional $J(U)$. Thus, for instance:

i) when $J(U) = \dfrac{1}{\Delta t_n}\displaystyle\int_{t_{n-1}}^{t_n}\int_\Omega j(U)d\Omega dt$, the dual problem to be solved in each I_n is:

$$\begin{cases} -\dfrac{\partial z}{\partial t} - \epsilon\Delta z = f'(U)z - \dfrac{1}{\Delta t_n} j'(U) \quad \text{in} \quad \Omega \times [t_{n-1},\ t_n), \\ z^{n+} = 0 \qquad \text{in} \quad \Omega, \\ z = 0 \qquad \text{in} \ \partial\Omega \times [t_{n-1},\ t_n); \end{cases}$$

ii) when $J(U) = \displaystyle\int_\Omega j(U^n)d\Omega$, the dual problem to be solved in each I_n is:

$$\begin{cases} -\dfrac{\partial z}{\partial t} - \epsilon\Delta z = f'(U)z \quad \text{in} \quad \Omega \times [t_{n-1},\ t_n), \\ z^{n+} = j'(U^n) \qquad \text{in} \quad \Omega, \\ z = 0 \qquad \text{in} \ \partial\Omega \times [t_{n-1},\ t_n). \end{cases}$$

$j(U)$ is a measurable function of U. Typical expressions for $j(U)$ are:

$$\begin{cases} j(U) = U, & j'(U) = 1; \\ j(U) = U^2 \log U, & j'(U) = 2U \log U + U. \end{cases}$$

Case 2: r=1 (continuous in time the primal solution) and s=0 (discontinuous in time the dual solution).

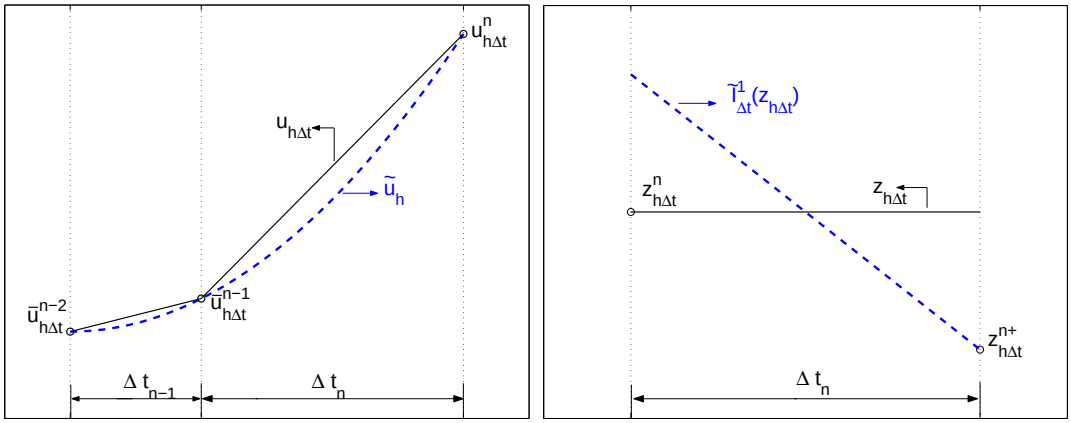

Figure 3.5: Quadratic recovery in time for the primal solution when $r = 1$ (left panel) and linear recovery in time for the dual solution when $s = 0$ (right panel).

In this case, for each I_n we have that

$$u_{h\Delta t}(x,t)\mid_{I_n} = \overline{u}^{n-1}_{h\Delta t}(x) + \frac{t-t_n}{\Delta t_n}(u^n_{h\Delta t}(x) - \overline{u}^{n-1}_{h\Delta t}(x)) \text{ and } z_{h\Delta t}(x,t)\mid_{I_n} = z^n_{h\Delta t}(x), \quad (3.27)$$

and the recovery formulae are

$$\begin{cases} \widetilde{I}^1_{\Delta t} z_{h\Delta t}(x,t) = z^{n+}_h(x) + 2\left(z^n_h(x) - z^{n+}_h(x)\right)\dfrac{t_n - t}{\Delta t_n}, \\ \\ I^2_{2\Delta t} u_{h\Delta t}(x,t) = \dfrac{t-t_{n-1}}{\Delta t_{n-1}}\dfrac{t-t_n}{(\Delta t_{n-1}+\Delta t_n)}\overline{u}^{n-2}_{h\Delta t} + \dfrac{t-t_{n-2}}{\Delta t_{n-1}}\dfrac{t_n - t}{\Delta t_n}\overline{u}^{n-1}_{h\Delta t} + \dfrac{t-t_{n-2}}{(\Delta t_{n-1}+\Delta t_n)}\dfrac{t-t_{n-1}}{\Delta t_n}u^n_{h\Delta t}. \end{cases} \quad (3.28)$$

As in the previous case, Figure 3.5 shows a graphical representation of $I^2_{2\Delta t}u_{h\Delta t}(x,t)$ and $\widetilde{I}^1_{\Delta t}z_{h\Delta t}(x,t)$.

For the evaluation of the time component of the error e^n_t in (3.23c) we propose the following approximations:

Case 1: r=s=0 (discontinuous in time both primal and dual solutions).

$$\begin{cases} z_h(x,t) - z_{h\Delta t}(x,t) \approx \widetilde{I}^1_{\Delta t} z_{h\Delta t}(x,t) - z_{h\Delta t}(x,t), \\ \\ u_h(x,t) - u_{h\Delta t}(x,t) \approx I^1_{\Delta t} u_{h\Delta t}(x,t) - u_{h\Delta t}(x,t). \end{cases} \quad (3.29a)$$

Case 2: r=1 (continuous in time the primal solution) and s=0 (discontinuous in time the dual solution).

$$\begin{cases} z_h(x,t) - z_{h\Delta t}(x,t) \approx \widetilde{I}^1_{\Delta t} z_{h\Delta t}(x,t) - z_{h\Delta t}(x,t), \\ \\ u_h(x,t) - u_{h\Delta t}(x,t) \approx I^2_{2\Delta t} u_{h\Delta t}(x,t) - u_{h\Delta t}(x,t). \end{cases} \quad (3.29b)$$

3.2 Strategy for adaptation in time and space

In time dependent problems there are several strategies to adapt both the spatial mesh and the size of the time step. The one we have chosen is reliable and efficient. It is reliable because it guaranties that the estimated error is below the prescribed tolerance, and it is efficient because it leads to economical meshes well adapted to the characteristics of the solution. Thus, assuming that at time level t_{n-1} we know the partition $\mathbb{T}^{n-1}_h$ and the primal and dual solutions $u^{n-1}_{h\Delta t}$ and $z^{n-1}_{h\Delta t}$ respectively, then, we will proceed to calculate the primal and dual solutions $u^n_{h\Delta t}$ and $z^n_{h\Delta t}$ in a new mesh $\mathbb{T}^n_h$ with a time step Δt_n based on the information provided by the a posteriori space and time error estimators $\widetilde{e}^n_s$ and $\widetilde{e}^n_t$, respectively, which are derived from those defined in Proposition 2 using the post-processing procedure. Thus, denoting by $\widetilde{e}^n_{sK}$ and $\widetilde{e}^n_{tK}$ the restriction to the element K of $\widetilde{e}^n_s$ and $\widetilde{e}^n_t$ respectively, we have then by virtue of Proposition 2 and the recovery formulas

$$\begin{cases} \widetilde{e}^n_{sK} := \overbrace{\dfrac{1}{2}\rho(u_{h\Delta t})(I^{2m}_{2h} z_{h\Delta t} - z_{h\Delta t})_K}^{\widetilde{e}^n_{sK,primal}} + \overbrace{\dfrac{1}{2}\rho^*(u_{h\Delta t}, z_{h\Delta t})(I^{2m}_{2h} u_{h\Delta t} - u_{h\Delta t})_K}^{\widetilde{e}^n_{sK,dual}}, \\ \\ \widetilde{e}^n_s = \widetilde{e}^n_{s,primal} + \widetilde{e}^n_{s,dual} = \displaystyle\sum_{K\in\mathbb{T}^n_h}\left(\widetilde{e}^n_{sK,primal} + \widetilde{e}^n_{sK,dual}\right) \end{cases} \quad (3.30a)$$

and

$$\begin{cases} \widetilde{e}^n_{tK} := \overbrace{\frac{1}{2}\rho(u_{h\Delta t})(\widetilde{I}^1_{\Delta t} z_{h\Delta t} - z_{h\Delta t})_K}^{\widetilde{e}^n_{tK,primal}} + \overbrace{\frac{1}{2}\rho^*(u_{h\Delta t}, z_{h\Delta t})(I^r_{(r+1)\Delta t} u_{h\Delta t} - u_{h\Delta t})_K}^{\widetilde{e}^n_{tK,dual}}, \\ \widetilde{e}^n_t = \widetilde{e}^n_{t,primal} + \widetilde{e}^n_{t,dual} = \sum_{K\in\mathbb{T}^n_h} \left(\widetilde{e}^n_{tK,primal} + \widetilde{e}^n_{tK,dual}\right). \end{cases} \tag{3.30b}$$

From these error estimators we define the so called *error indicators* that are used for practical adaptation. Thus,

$$\eta^n_{sK} = \frac{|\widetilde{e}^n_{sK}|}{|J(u_{h\Delta t})|}, \qquad \eta^n_s = \sum_{K\in\mathbb{T}^n_h} \eta^n_{sK} \tag{3.31a}$$

and

$$\eta^n_{tK} = \frac{|\widetilde{e}^n_{tK}|}{|J(u_{h\Delta t})|}, \qquad \eta^n_t = \sum_{K\in\mathbb{T}^n_h} \eta^n_{tK}. \tag{3.31b}$$

The strategy for adaptation will check whether the primal solution $u^n_{h\Delta t}$ is acceptable or not; if not, we proceed to calculate a new Δt_n and a new $\mathbb{T}^n_h$ by local refinements and coarsenings, and then to recalculate the primal and dual solutions $u^n_{h\Delta t}$ and $z^n_{h\Delta t}$ as well as the error indicators η^n_t and η^n_s. Prescribing a tolerance $TOL = Tol_t + Tol_s$, where Tol_t and Tol_s are tolerances to check time and space errors respectively, the primal solution $u_{h\Delta t}$ at each time level t_n is acceptable up to the tolerance TOL if

$$\eta^n_t \le Tol_t \ \text{ and } \ \eta^n_s \le Tol_s.$$

To balance the space and time errors one chooses $Tol_s \approx Tol_t$.

3.2.1 Mesh adaptation: mesh-optimization strategy.

The criterium to adapt the spatial mesh consists of calculating such a mesh with the minimum number of elements NE to satisfy $\eta^n_s \le Tol_s$. This yields the optimal size h^{opt}_K of the element $K \in \mathbb{T}^n_h$ as we have seen in Chapter 2

$$h^{opt}_K = h_K \left(\frac{Tol_s}{W}\right) (\eta^n_K)^{-1/(\alpha+d)}, \tag{3.32}$$

where $W := \sum_{K\in\mathbb{T}^n_h} (\eta^n_K)^{\frac{d}{d+\alpha}} < \infty$; $\eta^n_K = \eta^n_s|_K$ is the spatial error indicator for the element K; α is the convergence rate of the spatial error, it is assumed that $\eta^n_s = O(h^\alpha)$ and $d = 2$ or 3 is the spatial dimension. Formula (3.32) gives a criterion to refine or coarsen the element K. Specifically, comparing the size of our actual triangle h_K with the optimal size h^{opt}_K we obtain the number of times that this element needs to be refined or coarsened. Thus, we adopt the following refining and coarsening criteria:

Refining criterion:

If $\dfrac{h^{opt}_K}{h_K} \le 1$, mark the element K to refine n_r times,

$$n_r = Integer\ part\left[0.5 + 2\frac{\log(h_K/h^{opt}_K)}{\log 2}\right]. \tag{3.33a}$$

Coarsening criterion:

If $\dfrac{h_K^{opt}}{h_K} > 1$, mark the element K to coarsen n_c times,

$$n_c = Integer\ part\left[2\frac{\log(h_K^{opt}/h_K)}{\log 2}\right]. \tag{3.33b}$$

The refinement of marked elements is made bisecting the largest edge by joining its midpoint with the opposite vertex and taking the vertices thus created as the vertices of a new refinement. To maintain the regularity of the mesh in the refining and coarsening procedure we follow the strategy of [62].

3.2.2 Adaptation of the time step size

The error indicator η_t^n is used to adjust the size of the new time step whether the solution $u_{h\Delta t}^n$ is accepted or not. Following the strategy of the numerical ODE community we adjust the time step by the formula

$$\begin{aligned}
&\Delta t_{new} = \min(fac_{\max}, \max(fac_{\min}, fac))\Delta t_{old},\\
&fac = \begin{cases} \left(\left(\dfrac{\eta_t^{n-1}}{\eta_t^n}\right)^{1/\beta}\dfrac{\Delta t_n}{\Delta t_{n-1}}\right)\left(\dfrac{\kappa\cdot Tol_t}{\eta_t^n}\right)^{1/\beta} & \text{when } u_{h\Delta t}^n \text{ is accepted,}\\ \left(\dfrac{\kappa\cdot Tol_t}{\eta_t^n}\right)^{1/\beta} & \text{when } u_{h\Delta t}^n \text{ is rejected,}\end{cases}
\end{aligned} \tag{3.34}$$

where β is an unknown coefficient, which is equal to the order of the time local truncation error, and is calculated by a recursive procedure (see Chapter 2); $fac_{\max}$ and $fac_{\min}$ are factors limiting the maximum and minimum step sizes respectively, usually $fac_{\max} = 5$ and $fac_{\min} = 0.2$; and κ is a security factor to prevent unnecessary rejections because they cause recomputation and, therefore, loss of performance. In our examples $\kappa = 0.7$.

An algorithmic presentation of the adaptive strategy is the following.

Space – time adaptive algorithm :

Given the tolerances Tol_s, Tol_t, the macro triangulation $\mathbb{T}_{h,0}$ and the time step Δt_0.

Obtain an initial mesh $\mathbb{T}_h^0$ by local refinement and project the initial datum u^0 in that mesh as $u_{h\Delta t}^0 \in V_h^0$.

$n = 1$;

while $(t_{n-1} < T)$

{

$\mathbb{T}_{h,0}^n = \mathbb{T}_h^{n-1}$.

$i = 1$;

while $(k == 0)$

{

1. Adapt the mesh $\mathbb{T}_{h,i-1}^n$ to generate a new mesh $\mathbb{T}_{h,i}^n$.

2. Project the solution $u_{h\Delta t}^{n-1}$ onto the finite element space $V_{h,i}^n$ and obtain $\overline{u}_{h\Delta t}^{n-1}$.

3. Compute the solution $u_{h\Delta t,i}^n$ on the mesh $\mathbb{T}_{h,i}^n$ by calculate (3.16).

4. Calculate the error indicator with DWR technique in time η_t^n and in space η_s^n.

5. Select elements to be refined or coarsened with formulae (3.33a) and (3.33b).

If $[(\eta_s^n > Tol_s)\ and\ (\eta_t^n \leq Tol_t)]$

Reject step and maintain the time step length Δt_n. Step reject $k = 0$ and $i = i + 1$.

else

Calculate a new time step Δt_{new} with the formula (3.34).

if $(\eta_t^n > Tol_t)$

$t_n = t_{n-1} + \Delta t_{new}$. Reject step: $k = 0$ and $i = i + 1$.

else

$t_{n+1} = t_n + \Delta t_{new}$ and $k = 1$.

end if

end if

}

$u_{h\Delta t}^n = u_{h\Delta t,i}^n$.

$\mathbb{T}_h^n = \mathbb{T}_{h,i}^n$.

}

(3.35)

To illustrate the performance of the DWR proposed in this paper, we shall show some numerical examples that may be considered representative to test the capabilities of the method.

3.3 The stationary problem

The aim in executing this example is to test the patch-wise higher-order interpolation recovery in space in unstructured meshes. Thus, we consider the problem

$$-\Delta u = f \ \text{ in } \ \Omega := (0,1)^2, \ \ u = 0 \ \text{ on } \ \partial\Omega, \tag{3.36a}$$

with exact solution

$$u(x,y) = \frac{\left(1-x^2\right)^2\left(1-y^2\right)^2}{10x^2+0.1}, \tag{3.36b}$$

whose graphical representation is shown in Figure 3.6. On the left panel we have the solution in 3D and on the right panels are displayed the cross sections at $x = 0$ (top) and $y = 0$ (bottom).

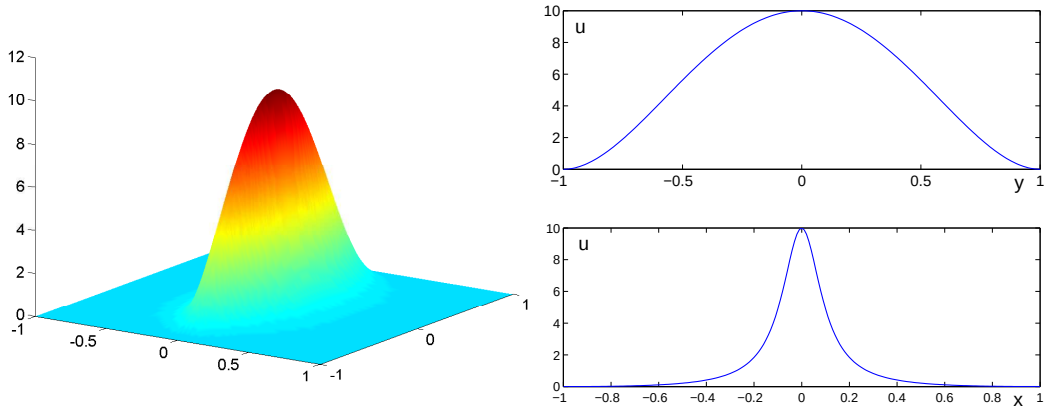

Figure 3.6: 3D solution $u(x, y)$ (left) and cross sections of it at $x = 0$ (top) and at $y = 0$ (bottom).

In the application of DWR we control the error with respect to the output functionals

$$J_1(u) = \int_\Omega u d\Omega, \qquad j_1(u) = u, \tag{3.37a}$$

and

$$J_2(u) = \frac{1}{2}\int_\Omega u^2 d\Omega, \qquad j_2(u) = \frac{1}{2}u^2. \tag{3.37b}$$

The general theory of the DWR method for linear elliptic problems ([11], Chapter 2) yields the following primal and dual problems

$$\begin{cases} -(\nabla u_h, \nabla \psi_h) + (f, \psi_h) = 0 \quad \forall \psi_h \in V_h, \\ \\ -(\nabla z_h, \nabla \varphi_h) + J'(u_h)(\varphi_h) = 0 \ \ \forall \varphi_h \in V_h, \end{cases} \tag{3.38}$$

where $V_h \subset H_0^1(\Omega)$ is the finite element associated to the regular partition $\mathbb{T}_h$. The error for the functional $J(u)$ can be written as

$$\begin{cases} J(u) - J(u_h) \simeq \sum\limits_{K\in\mathbb{T}_h} e_K = \sum\limits_{K\in\mathbb{T}_h} \frac{1}{2}\rho(u_h)(z-\psi_h)_K + \frac{1}{2}\rho^*(u_h, z_h)(u-\varphi_h)_K, \\ e_K := \frac{1}{2}\rho(u_h)(z-\psi_h)_K \quad + \quad \frac{1}{2}\rho^*(u_h, z_h)(u-\varphi_h)_K, \end{cases} \tag{3.39}$$

where the residuals $e_{K,primal} = \frac{1}{2}\rho(u_h)(z-\psi_h)_K$ and $e_{K,dual} = \frac{1}{2}\rho^*(u_h, z_h)(u-\varphi_h)_K$ are approximated by the patch-wise higher-order interpolation recovery described above. Thus, we have

$$\begin{cases} \widetilde{e}_{K,primal} = \frac{1}{2}\rho(u_h)(I_{2h}^{2m} z - z_h)_K, \\ \\ \widetilde{e}_{K,dual} = \frac{1}{2}\rho^*(u_h, z_h)(I_{2h}^{2m} u - u_h)_K, \end{cases} \tag{3.40}$$

primal residual:

$$\{\rho(u_h)(z-\psi_h)_K \simeq (R_h, I_{2h}^{2m} z - z_h)_K + (r_h, I_{2h}^{2m} z - z_h)_{\partial K}\ ,$$

$$\begin{cases} R_h = \Delta u_h + f(u_h), \\ r_h = \begin{cases} \frac{1}{2}\left[\partial_n u_h\right]_\Gamma & if \quad \Gamma \subset \partial K \backslash \partial\Omega, \\ 0 & if \quad \Gamma \subset \partial\Omega, \end{cases} \end{cases} \tag{3.41a}$$

and

dual residual:

$$\{\rho^*(u_h, z_h)(u-\varphi_h)_K \simeq (R_h^*, I_{2h}^{2m}u - u_h)_K + (r_h^*, I_{2h}^{2m}u - u_h)_{\partial K}\ ,$$

$$\left\{ \begin{array}{l} R_h^* = \Delta z_h + j'(u_h), \\ r_h^* = \left\{ \begin{array}{ll} \frac{1}{2}\left[\partial_n z_h\right]_\Gamma & if \quad \Gamma \subset \partial K \backslash \partial\Omega, \\ 0 & if \quad \Gamma \subset \partial\Omega. \end{array} \right. \end{array} \right. \tag{3.41b}$$

The numerical algorithm presents this easy form:

Space adaptive algorithm :

Given Tol and the macro triangulation $\mathbb{T}_{h,0}$.
$i = 0$.
while $(\eta > Tol)$
{
1. Compute the solution $u_{h,i}$ on the mesh $\mathbb{T}_{h,i}$.
2. Calculate the error indicator η with DWR technique.
3. Select elements to be refined or coarsened, with the algorithm described in (3.33a) and (3.33b).

If $(\eta > Tol)$
Adapt the mesh $\mathbb{T}_{h,i}$ generating a new $\mathbb{T}_{h,i+1}$.
$i = i + 1$.
end if
}
$u_h = u_{h,i}$.
$\mathbb{T}_h = \mathbb{T}_{h,i}$.

To calculate the numerical solution we use linear and quadratic elements on triangular meshes. We shall analyze two basic aspects of the DWR method. First, we test the validity of the patch-wise higher order interpolation recovery proposed in this thesis. To do so, we calculate the (inverse) effectivity index defined as [11].

$$I_{eff} := \left| \frac{\widetilde{e}_{primal} + \widetilde{e}_{dual}}{J(u) - J(u_h)} \right|$$

where $\widetilde{e} = \widetilde{e}_{primal} + \widetilde{e}_{dual}$ is an estimation of the error calculated by (3.40), whereas $J(u) - J(u_h)$ is the error committed by using the exact solution $u(x, y)$ to calculate $J(u)$. The recovery procedure will be acceptable if it yields an effectivity index I_{eff} close to 1; in fact, the recovery procedure is optimal if $\lim_{Tol \to 0} I_{eff} \to 1$. As we observe in Tables 1a and 1b, where the variation of I_{eff} for the functional $J_1(u)$ is shown as a function of the number of elements NE in the mesh for linear and quadratic elements respectively, the patch-wise higher order interpolation recovery works.

Linear

NE	$\widetilde{e}_{primal}$	$\widetilde{e}_{dual}$	$\widetilde{e}_{primal} + \widetilde{e}_{dual}$	$J(u) - J(u_h)$	I_{eff}
934	$-4.35 \cdot 10^{-3}$	$-4.19 \cdot 10^{-3}$	$-8.55 \cdot 10^{-3}$	$-6.82 \cdot 10^{-3}$	1.24
2106	$-1.01 \cdot 10^{-3}$	$-0.96 \cdot 10^{-3}$	$-1.97 \cdot 10^{-3}$	$-1.96 \cdot 10^{-3}$	1.00
4512	$-0.72 \cdot 10^{-3}$	$-0.64 \cdot 10^{-3}$	$-1.36 \cdot 10^{-3}$	$-1.56 \cdot 10^{-3}$	0.87
9472	$-1.94 \cdot 10^{-4}$	$-1.90 \cdot 10^{-4}$	$-3.84 \cdot 10^{-4}$	$-4.50 \cdot 10^{-4}$	0.86
19536	$-1.33 \cdot 10^{-4}$	$-1.29 \cdot 10^{-4}$	$-2.62 \cdot 10^{-4}$	$-3.09 \cdot 10^{-4}$	0.85
39968	$-4.31 \cdot 10^{-5}$	$-4.20 \cdot 10^{-5}$	$-8.51 \cdot 10^{-5}$	$-9.29 \cdot 10^{-5}$	0.87
81120	$-3.03 \cdot 10^{-5}$	$-2.90 \cdot 10^{-5}$	$-5.93 \cdot 10^{-5}$	$-6.92 \cdot 10^{-5}$	0.85

Table 1a: Variation of the error and I_{eff} for the functional $J_1(u)$ as a function of the number of linear elements.

Quadratic

NE	$\widetilde{e}_{primal}$	$\widetilde{e}_{dual}$	$\widetilde{e}_{primal} + \widetilde{e}_{dual}$	$J(u) - J(u_h)$	I_{eff}
832	$0.72 \cdot 10^{-4}$	$0.61 \cdot 10^{-4}$	$1.32 \cdot 10^{-4}$	$-3.22 \cdot 10^{-3}$	0.04
1910	$-2.76 \cdot 10^{-6}$	$-2.48 \cdot 10^{-6}$	$-5.24 \cdot 10^{-6}$	$-3.02 \cdot 10^{-4}$	0.02
4148	$-1.60 \cdot 10^{-6}$	$-1.52 \cdot 10^{-6}$	$-3.12 \cdot 10^{-6}$	$-7.96 \cdot 10^{-6}$	0.39
8780	$-1.81 \cdot 10^{-7}$	$-1.73 \cdot 10^{-7}$	$-3.54 \cdot 10^{-7}$	$-4.50 \cdot 10^{-7}$	0.77
18216	$-0.75 \cdot 10^{-7}$	$-0.66 \cdot 10^{-7}$	$-1.41 \cdot 10^{-7}$	$-1.23 \cdot 10^{-7}$	1.20
37400	$-1.23 \cdot 10^{-8}$	$-1.03 \cdot 10^{-8}$	$-2.26 \cdot 10^{-8}$	$-2.75 \cdot 10^{-8}$	0.81
76112	$-3.88 \cdot 10^{-9}$	$-3.57 \cdot 10^{-9}$	$-7.45 \cdot 10^{-9}$	$-7.52 \cdot 10^{-9}$	0.98

Table 1b: Variation of the error and I_{eff} for the functional $J_1(u)$ as a function of the number of quadratic elements.

We have repeated the same type of calculations for the output functional $J_2(u)$, showing the results in Tables 2a and 2b.

Linear

NE	$\widetilde{e}_{primal}$	$\widetilde{e}_{dual}$	$\widetilde{e}_{primal} + \widetilde{e}_{dual}$	$J(u) - J(u_h)$	I_{eff}
934	$0.68 \cdot 10^{-1}$	$0.64 \cdot 10^{-1}$	$1.32 \cdot 10^{-1}$	$1.17 \cdot 10^{-1}$	0.77
2106	$3.12 \cdot 10^{-2}$	$3.15 \cdot 10^{-2}$	$6.17 \cdot 10^{-2}$	$7.03 \cdot 10^{-2}$	0.87
4512	$1.93 \cdot 10^{-2}$	$1.89 \cdot 10^{-2}$	$3.82 \cdot 10^{-2}$	$4.21 \cdot 10^{-2}$	0.89
9472	$0.76 \cdot 10^{-2}$	$0.78 \cdot 10^{-2}$	$1.54 \cdot 10^{-2}$	$1.72 \cdot 10^{-2}$	0.88
19536	$0.56 \cdot 10^{-2}$	$0.46 \cdot 10^{-2}$	$1.02 \cdot 10^{-2}$	$1.06 \cdot 10^{-2}$	0.95
39968	$2.01 \cdot 10^{-3}$	$1.91 \cdot 10^{-3}$	$3.92 \cdot 10^{-3}$	$4.29 \cdot 10^{-3}$	0.94
81120	$1.28 \cdot 10^{-3}$	$1.25 \cdot 10^{-3}$	$2.53 \cdot 10^{-3}$	$2.63 \cdot 10^{-3}$	0.95

Table 2a: Variation of the error and I_{eff} for the functional $J_2(u)$ as a function of the number of linear elements.

Quadratic

NE	$\widetilde{e}_{primal}$	$\widetilde{e}_{dual}$	$\widetilde{e}_{primal} + \widetilde{e}_{dual}$	$J(u) - J(u_h)$	I_{eff}
832	$1.46 \cdot 10^{-3}$	$1.38 \cdot 10^{-3}$	$2.84 \cdot 10^{-3}$	$8.02 \cdot 10^{-3}$	0.35
1910	$2.11 \cdot 10^{-3}$	$2.02 \cdot 10^{-3}$	$4.13 \cdot 10^{-3}$	$8.36 \cdot 10^{-3}$	0.49
4148	$0.66 \cdot 10^{-4}$	$0.65 \cdot 10^{-4}$	$1.31 \cdot 10^{-4}$	$1.34 \cdot 10^{-4}$	0.97
8780	$1.16 \cdot 10^{-5}$	$1.16 \cdot 10^{-5}$	$2.32 \cdot 10^{-5}$	$2.23 \cdot 10^{-5}$	1.01
18216	$3.88 \cdot 10^{-6}$	$3.84 \cdot 10^{-6}$	$7.72 \cdot 10^{-6}$	$7.81 \cdot 10^{-6}$	0.98
37400	$0.67 \cdot 10^{-6}$	$0.65 \cdot 10^{-6}$	$1.32 \cdot 10^{-6}$	$1.55 \cdot 10^{-6}$	0.86
76112	$2.40 \cdot 10^{-7}$	$2.32 \cdot 10^{-7}$	$4.72 \cdot 10^{-7}$	$4.85 \cdot 10^{-7}$	0.96

Table 2b: Variation of the error and I_{eff} for the functional $J_2(u)$ as a function of the number of quadratic elements.

Now, we turn our attention to the mesh-optimization algorithm in order to have an idea about the number of iterations which are necessary to get an optimal mesh. Also, we wish to estimate the convergence rate α when the output functionals $J_1(u)$ and $J_2(u)$ are used. The results are shown in Table 3.

	$J(u) = \int_\Omega u d\Omega$	$J(u) = \int_\Omega \frac{1}{2} u^2 d\Omega$
Tol Linear	$3 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
α_l^{end}	2.26	2.29
Tol Quadratic	$3 \cdot 10^{-6}$	$1 \cdot 10^{-5}$
α_c^{end}	5.57	4.63

Table 3: The convergence rate α for the output functionals $J_1(u)$ and $J_2(u)$ with linear and quadratic elements.

To estimate α we start with the initial values $\alpha^{(0)} = 6$ for linear elements and $\alpha^{(0)} = 8$ for quadratic elements and after 2 or 3 iteration we reach the optimal mesh and the end values of α. Figure 3.7 shows the optimal meshes for linear elements and functionals $J_1(u)$ and $J_2(u)$.

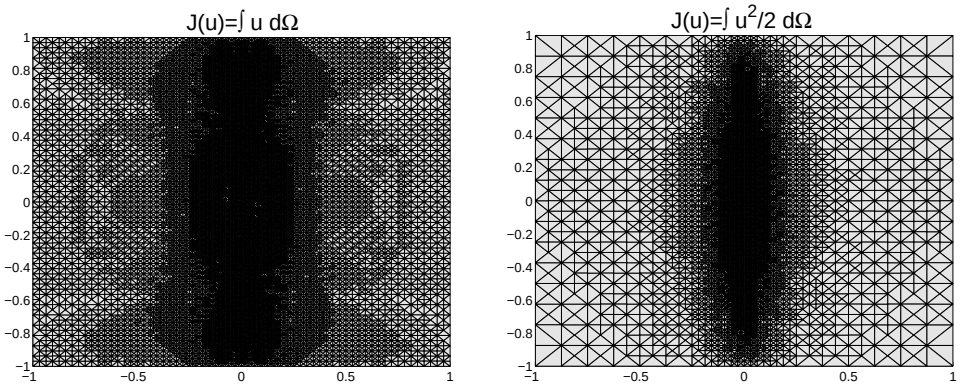

Figure 3.7: Optimal meshes for the functionals $J_1(u)$ and $J_2(u)$ with linear elements. The number of elements on the left NE=63210 and on the right NE=41832.

3.4 A semilinear time dependent problem

Our next numerical example will simulate the propagation of laminar flames according to the model equations proposed in [52]. Combustion problems are known to range among the most demanding for spatial adaptivity when the thin flame front has to be resolved numerically. Moreover, if there are ignition and extinction processes, complicated geometries and non-uniform material, the relevant time scales can change by orders of magnitude, then an adaptive time step size control is needed too.

For this example we use quadratic elements ($m = 2$) for space discretization in both solutions, and for time discretization cG(1) (continuous Galerkin $r = 0$) for the primal solution and dG(0) (discontinuous Galerkin $s = 0$) for the dual solution. It is known that this combination of time discretizations yields a Crank-Nicolson type scheme for the primal solution. At every time step one obtains a discrete nonlinear system of equations that is solved by the classical Newton algorithm; however, we allow the use of a damped Newton algorithm for those cases where the high stiffness of the nonlinear system requires to do so. The linear systems of algebraic equations are solved by a diagonal preconditioned conjugate gradient if they are symmetric and a diagonal preconditioned BICGSTAB if they are not symmetric.

We consider in $\Omega \times (0, T]$ a thermo-diffusive model where u_1 and u_2 are the temperature and the mass fraction respectively, the equations of which are:

$$\left\{ \begin{array}{l} \left\{ \begin{array}{l} \dfrac{\partial u_1}{\partial t} - \Delta u_1 = \omega(u_1, u_2), \\ \\ \dfrac{\partial u_2}{\partial t} - \dfrac{1}{Le}\Delta u_2 = -\omega(u_1, u_2). \end{array} \right. \\ \\ \textit{Boundary conditions} : \forall t > 0, \left\{ \begin{array}{ll} u_1 = 1 & \text{on } \Gamma_D, \\ \partial_n u_1 = 0 & \text{on } \Gamma_N, \\ \partial_n u_1 = -k(u_1 - u_{1ref}) & \text{on } \Gamma_R, \end{array} \right. \textit{and} \left\{ \begin{array}{ll} u_2 = 0 & \text{on } \Gamma_D, \\ \partial_n u_2 = 0 & \text{on } \Gamma_N \cup \Gamma_R. \end{array} \right. \\ \\ \textit{Initial condition} : u_1(x, 0) = u_1^0, \quad u_2(x, 0) = u_2^0. \end{array} \right. \tag{3.42}$$

We use a simple one-species reaction mechanism governed by an Arrhenius law in which the reaction rate ω is given by

$$\omega = \frac{\beta^2}{2Le} u_2 \cdot \exp\left(\frac{\beta(u_1 - 1)}{1 + \alpha(u_1 - 1)} \right). \tag{3.43}$$

Here, Le denotes the Lewis Number, α is the temperature ratio that determines the gas expansion in non-constant density flows and β denotes the Zeldovich number. In the numerical examples $Le = 1$, $\alpha = 0.8$ and $\beta = 10$.

The time of integration is $T = 60$. Figure 3.8 shows the domain Ω with length $L = 60$, height $H = 16$, with a double obstacle of length $L/4$ and height $h = 8$.

In the obstacle there is an absorption of heat modelled by the Robin boundary condition on Γ_R, where k is the heat loss parameter and u_{1ref} is the reference temperature outside the domain. In our numerical simulation these magnitudes have the values $k = 0.1$ and $u_{1ref} = 0$. On the left boundary of the domain Γ_D a Dirichlet condition corresponding to the burnt state is imposed, while the remaining boundary conditions are of homogeneous Neumann type on Γ_N.

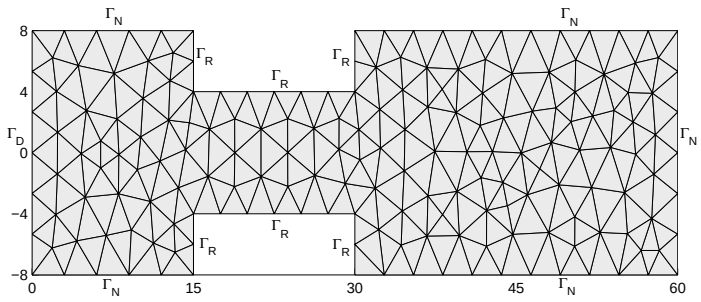

Figure 3.8: The domain for the combustion problem

The initial condition is the analytical solution of a one-dimensional right-travelling flame in the limit $\beta \to \infty$ located left of the obstacle:

$$\begin{cases} u_1^0(x,0) = \begin{cases} 1 & if\ x \le x_0, \\ e^{x_0 - x} & if\ x > x_0, \end{cases} \\ u_2^0(x,0) = \begin{cases} 1 & if\ x \le x_0, \\ 1 - e^{Le(x_0 - x)} & if\ x > x_0. \end{cases} \end{cases}$$

We take $x_0 = 9$.

In our numerical experiment the flame becomes curved and slows down in the interior of the channel, but it manages to pass through because the value of k is small. For stronger heat losses the flame would be extinguished, actually, for $k = 0.2$ this would be the case. This behavior of the flame at different time instants can be observed in Figure 3.9.

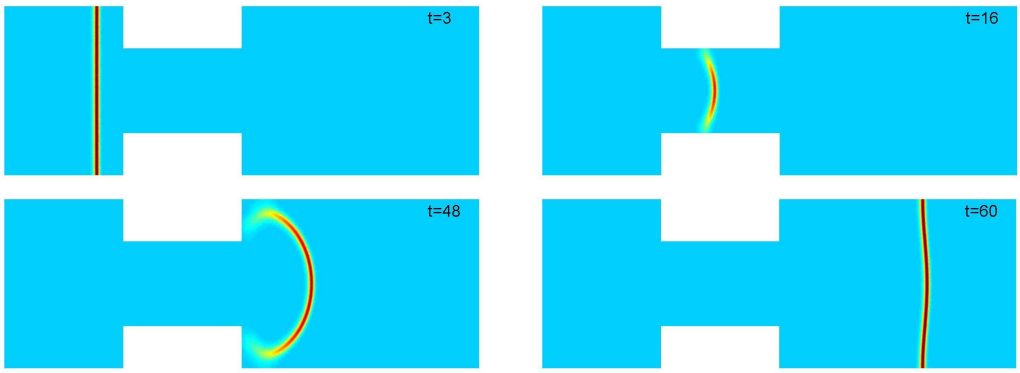

Figure 3.9: Evolution of the flame at time t = 3, 16, 48 and 60.

We shall test the performance of our algorithm with the local (in time) output functionals in each subinterval I_n

$$J_1(U) = J(U_1, U_2) = \int_\Omega \omega(U_1^n, U_2^n) d\Omega, \qquad j_1(U) = \omega(U), \qquad \text{1st functional}$$

and

$$J_2(U) = J(U_1, U_2) = \int_\Omega U_1^n d\Omega, \qquad j_2(U) = U_1, \qquad \text{2nd functional.}$$

For these functionals, the adjoint problem associated to (3.42) in each I_n is the following.

For $1 \leq i, j \leq 2$

$$-\frac{\partial z_i}{\partial t} - \epsilon_i \Delta z_i = \sum_{j=1}^{2} \frac{\partial \omega_j(U)}{\partial U_i} z_j \quad \text{in } S_n, \tag{3.44}$$

with boundary conditions for all $t \in I_n$

$$z_i \mid_{\Gamma_D} = 0, \quad \left.\frac{\partial z_i}{\partial n}\right|_{\Gamma_N} = 0 \quad \text{and} \quad \left.\frac{\partial z_i}{\partial n}\right|_{\Gamma_R} = \begin{cases} -k z_1 & \text{for } i = 1, \\ 0 & \text{for } i = 2, \end{cases} \tag{3.45}$$

and initial conditions

$$z_1^{n+} = \begin{cases} \left.\dfrac{\partial \omega(U)}{\partial U_1}\right|_{t_n} & \text{for } J_1(U), \\ 1 \text{ for } J_2(U). \end{cases} \quad z_2^{n+} = \begin{cases} \left.\dfrac{\partial \omega(U)}{\partial U_2}\right|_{t_n} & \text{for } J_1(U), \\ 0 \text{ for } J_2(U). \end{cases} \tag{3.46}$$

In these equations,

$$\begin{cases} U = (U_1, U_2), \\ \epsilon_1 = 1, \quad \epsilon_2 = 1/Le, \\ \omega_1(U_1, U_2) = -\omega_2(U_1, U_2) = \omega(U), \\ \dfrac{\partial \omega(U)}{\partial U_1} = \dfrac{\beta^2}{2Le} \dfrac{\beta U_2}{(1 + \alpha(U_1 - 1))^2} \exp\left(\dfrac{\beta(U_1 - 1)}{1 + \alpha(U_1 - 1)}\right), \quad \dfrac{\partial \omega(U)}{\partial U_2} = \dfrac{\beta^2}{2Le} \exp\left(\dfrac{\beta(U_1 - 1)}{1 + \alpha(U_1 - 1)}\right). \end{cases}$$

In the sequel we use the shorthand notations

$$\begin{cases} \omega'_{ij}(U) = \dfrac{\partial \omega_j(U_1, U_2)}{\partial U_i}, \\ j'_i = \dfrac{\partial j(U_1, U_2)}{\partial U_i}. \end{cases}$$

As we mention above, we calculate the numerical solution with continuous in time first order polynomials ($r = 1$) for the primal solution and piecewise constant in time polynomials ($s = 0$) for the dual solution, and with quadratic finite elements ($m = 2$) for both solutions. Considering the boundary conditions of the problem and its associated adjoint problem we shall seek the numerical solutions in the finite dimensional spaces

$$\begin{cases} V^{(1)}_{h\Delta tD} = \{\varphi_{h\Delta t} : \overline{\Omega} \times \overline{I} \to \mathbb{R} : \ \forall n \text{ and } (x,t) \in S_n, \ \varphi_{h\Delta t} \in C(\overline{I}_n; V_h^n), \ \varphi_{h\Delta t}(x,0) \in V_h^0, \\ \qquad \varphi_{h\Delta t} \mid_{\Gamma_D} = 1 \text{ and } \varphi_{h\Delta t}(x,\cdot)|_{\overline{I}_n} \in P_1\}, \\ \\ V^{(1)}_{h\Delta t0} = \{\varphi_{h\Delta t} : \overline{\Omega} \times \overline{I} \to \mathbb{R} : \ \forall n \text{ and } (x,t) \in S_n, \ \varphi_{h\Delta t} \in C(\overline{I}_n; V_h^n), \ \varphi_{h\Delta t}(x,0) \in V_h^0, \\ \qquad \varphi_{h\Delta t} \mid_{\Gamma_D} = 0 \text{ and } \varphi_{h\Delta t}(x,\cdot)|_{\overline{I}_n} \in P_1\}, \end{cases} \tag{3.47}$$

and

$$W^{(0)}_{h\Delta t0} = \{\psi_{h\Delta t} : \overline{\Omega} \times \overline{I} \to \mathbb{R} : \ \forall n \text{ and } (x,t) \in S_n, \ \psi_{h\Delta t} \in L^p(I_n; V_h^n), \ \psi_{h\Delta t}(x,0) \in V_h^0 \\ \psi_{h\Delta t} \mid_{\Gamma_D} = 0 \text{ and } \psi_{h\Delta t}(x,\cdot)|_{I_n} \in P_0\}. \tag{3.48}$$

The primal and dual finite element solutions for (3.42) are:

Primal solution: *For each I_n, find* $(u_{1h\Delta t}, u_{2h\Delta t}) \in V^{(1)}_{h\Delta tD} \times V^{(1)}_{h\Delta t0}$ *such that* $\forall (\psi_{1h\Delta t}, \psi_{2h\Delta t}) \in W^{(0)}_{h\Delta t0} \times W^{(0)}_{h\Delta t0}$:

$$\left\{ \begin{array}{l} (u^n_{ih\Delta t} - \overline{u}^{n-1}_{ih\Delta t}, \psi^n_{ih\Delta t})_\Omega + \dfrac{1}{2}\Delta t_n a_i(u^n_{ih\Delta t} + \overline{u}^{n-1}_{ih\Delta t}, \psi^n_{ih\Delta t}) + \\ \\ + \dfrac{\gamma_i k}{2}\Delta t_n \left(u^n_{ih\Delta t} + \overline{u}^{n-1}_{ih\Delta t}, \psi_{ih\Delta t}\right)_{\Gamma_R} - \displaystyle\int_{I_n} (\omega_i(u_{h\Delta t}), \psi_{ih\Delta t})_\Omega \, dt = 0. \end{array} \right. \tag{3.49a}$$

Dual Solution: *For each I_n, find* $(z_{1h\Delta t}, z_{2h\Delta t}) \in W^{(0)}_{h\Delta t0} \times W^{(0)}_{h\Delta t0}$ *such that* $\forall (\varphi_{1h\Delta t}, \varphi_{2h\Delta t}) \in V^{(1)}_{h\Delta t0} \times V^{(1)}_{h\Delta t0}$:

$$\left\{ \begin{array}{l} (j_i'(u^n_{h\Delta t}) - z^n_{ih\Delta t}, \varphi^n_{ih\Delta t})_\Omega - \dfrac{1}{2}\Delta t_n a_i(z^n_{ih\Delta t}, \varphi^n_{ih\Delta t}) - \\ \\ - \dfrac{\gamma_i k}{2}\Delta t_n \left(z^n_{ih\Delta t}, \varphi^n_{ih\Delta t}\right)_{\Gamma_R} + \displaystyle\int_{I_n} \left(\sum_{j=1}^{2} \omega_{ij}'(u_{h\Delta t}) \cdot z^n_{jh\Delta t}, \varphi_{ih\Delta t} \right)_\Omega dt = 0. \end{array} \right. \tag{3.49b}$$

In these equations $i = 1, 2$;

$$\gamma_1 = 1, \;\; \gamma_2 = 0 \quad \text{and} \quad a_i(u_i, v_i) = \int_\Omega \epsilon_i \nabla u_i \cdot \nabla v_i d\Omega.$$

Moreover, we point out that we take $\varphi_{ih\Delta t}(x,t) \in V^{(1)}_{h\Delta t0}$ as $\varphi_{ih\Delta t}(x,t) = \dfrac{t - t_{n-1}}{\Delta t_n}\varphi_h(x)$, where $\varphi_h(x) \in V^n_h$ is such that $\varphi_h(x)|_{\Gamma_D} = 0$. Note that $\varphi_{ih\Delta t}(x,t) \in C(\overline{I}_n; V^n_h)$. As for the functions $\psi_{ih\Delta t}(x,t)$, we note on one side that they are piecewise constant in time for $t \in I_n$ and on the other side they are in V^n_h satisfying the homogeneous Dirichlet boundary condition on Γ_D; so that, for $t \in I_n$, we take $\psi_{ih\Delta t}(x,t) = \varphi_h(x) \in V^n_h$ such that $\varphi_h(x)|_{\Gamma_D} = 0$. For each I_n, $u_{h\Delta t} \in V^{(1)}_{h\Delta tD}$ is given by

$$u_{ih\Delta t} = \overline{u}^{n-1}_{ih\Delta t}(x) + \frac{t - t_{n-1}}{\Delta t_n}(u^n_{ih\Delta t}(x) - \overline{u}^{n-1}_{ih\Delta t}(x)),$$

where $\overline{u}^{n-1}_{ih\Delta t}(x), u^n_{ih\Delta t}(x) \in V^n_h$ and satisfy the boundary conditions.

To estimate the error $J(U) - J(u_{h\Delta t})$ we apply Proposition 1 and estimate the residuals $\rho(u_{h\Delta t})(\cdot)$ and $\rho^*(u_{h\Delta t}, z_{h\Delta t})(\cdot)$ in terms of the local residuals.

For each element K, the **primal residual** takes the form $\rho_K(u_{h\Delta t})(\cdot) := \rho_{1K}(u_{h\Delta t})(\cdot) + \rho_{2K}(u_{h\Delta t})(\cdot)$, with $i = 1, 2$:

$$\rho_{iK}(u_{h\Delta t})(\cdot) = \int_{I_n} \{(R_{ih\Delta t}, \cdot)_K + (r_{ih\Delta t}, \cdot)_{\partial K}\} dt,$$

$$\left\{ \begin{array}{l} R_{ih\Delta t} = \frac{1}{\Delta t_n}(\overline{u}^{n-1}_{ih\Delta t} - u^n_{ih\Delta t}) + \epsilon_i \Delta u_{ih\Delta t} + \omega_i(u_{h\Delta t}), \\ \\ r_{ih\Delta t} = \left\{ \begin{array}{ll} \frac{1}{2}\left[\partial_n u_{ih\Delta t}\right]_\Gamma & \text{if} \;\; \Gamma \subset \partial K \backslash \partial\Omega, \\ -\partial_n u_{ih\Delta t} & \text{on} \;\; \Gamma_N, \\ -\partial_n u_{ih\Delta t} - \gamma_i k u_{ih\Delta t} & \text{on} \;\; \Gamma_R, \\ 0 & \text{on} \;\; \Gamma_D. \end{array} \right. \end{array} \right. \tag{3.50a}$$

Similarly, the **dual residual** $\rho^*_{iK}(u_{h\Delta t}, z_{h\Delta t})(\cdot) := \rho^*_{1K}(u_{h\Delta t}, z_{h\Delta t})(\cdot) + \rho^*_{2K}(u_{h\Delta t}, z_{h\Delta t})(\cdot)$, with $i = 1, 2$

$$\rho^*_{iK}(u_{h\Delta t}, z_{h\Delta t})(\cdot) = \int_{I_n} \{(R^*_{ih\Delta t}, \cdot)_K + (r^*_{ih\Delta t},)_{\partial K}\} dt + (j_i'(u^n_{h\Delta t}) - z^n_{ih\Delta t}, \cdot)_K,$$

$$\left\{ \begin{array}{l} R^*_{ih\Delta t} = \epsilon_i \Delta z^n_{ih\Delta t} + \sum_{j=1}^2 \omega'_{ij}(u_{h\Delta t}) z^n_{jh\Delta t}, \\ r^*_{ih\Delta t} = \left\{ \begin{array}{ll} \frac{1}{2}\left[\partial_n z^n_{ih\Delta t}\right]_\Gamma & \text{if } \Gamma \subset \partial K \backslash \partial\Omega, \\ -\partial_n z^n_{ih\Delta t} & \text{on } \Gamma_N, \\ -\partial_n z^n_{ih\Delta t} - \gamma_i k z^n_{ih\Delta t} & \text{on } \Gamma_R, \\ 0 & \text{on } \Gamma_D. \end{array} \right. \end{array} \right. \tag{3.50b}$$

We calculate the a posteriori error estimators $\eta^n_t = \sum_{K\in\mathbb{T}^n_h} \eta^n_{tK}$ and $\eta^n_s = \sum_{K\in\mathbb{T}^n_h} \eta^n_{sK}$ substituting (3.50a) and (3.50b) into (3.30a and 3.30b) and approximating the integrals by quadrature rules.

Next, we shall apply the proposed adaptive algorithm to calculate the numerical solution to (3.42) with the first and second functionals introduced above as output functionals.

3.4.1 Functional $J_1(U) = \int_\Omega \omega(U^n_1, U^n_2) d\Omega$

To calculate the numerical solution to (3.42) with this functional we choose $Tol_t = Tol_s = 5 \cdot 10^{-4}$. Moreover, we shall compute another numerical solution with $Tol_t = Tol_s = 10^{-4}$ considering this solution as the "exact" solution to compare with.

Figure 3.10 shows the evolution in time of the number of nodes (left panel) and the magnitude of the time step Δt (right panel) for the numerical and "exact" solutions. The number of time steps needed to reach $T = 60$ is $N_{\Delta t} = 2981$, with 29 time steps rejected (only 1% of time steps rejected).

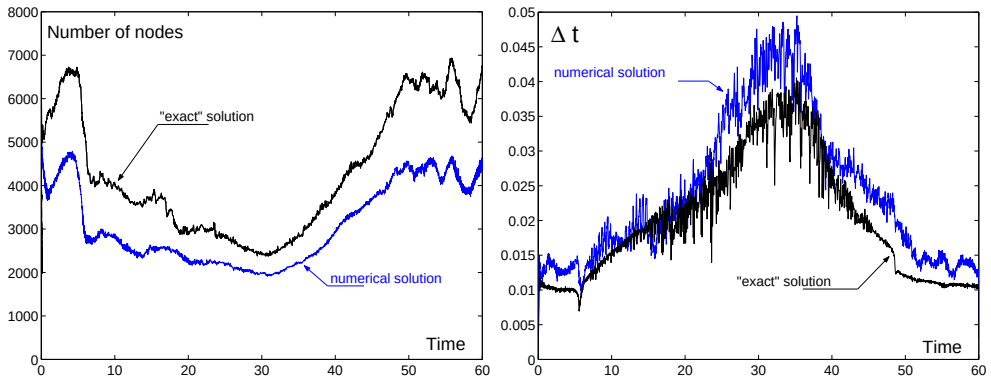

Figure 3.10: Evolution of the number of nodes N_h (left) and the time step size Δt (right) with the time for the functional $J_1(U)$.

In Figure 3.11 we show the evolution in time of the functional for the "exact" solution (left panel) and the point-wise and global relative errors (right panel). The point-wise error oscillates in the interval $[-2 \cdot 10^{-4},\ 4 \cdot 10^{-4}]$ when $J(U)$ is smooth, for instance, in the intervals $[0, 5]$, $[6, 30]$ and $[50, 60]$; however, in those regions where $J(U)$ loses smoothness or experiences a rapid variation the error grows. It is remarkable the variation of the point-wise error between $1.8 \cdot 10^{-2}$ and $-3 \cdot 10^{-2}$ in a neighborhood of $t = 50$ which is the region of fastest variation of

$J(U)$. Also, it is interesting to note that when $J(U)$ returns to be smooth the point-wise error becomes again very small, as it happens in the interval $(50, 60]$. A similar behavior is exhibited by the global error.

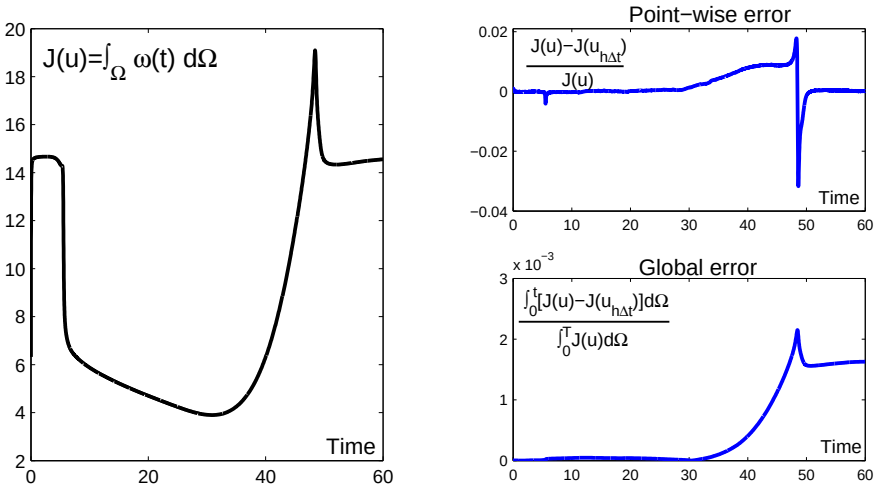

Figure 3.11: Evolution in time of the function $J(U) = \int_\Omega \omega(t) d\Omega$ (left) and point-wise (top right) and global (bottom right) errors in this quantity.

The calculations were carried out on a Pentium Dual Core 3GHz, 2GB RAM and the distribution of the CPU time in seconds for each part of the method is displayed in Figure 3.12. Note in this figure that the calculation of the primal solution spends almost a sixty per cent of the total of CPU time. This numerical solution is calculated using Newton method.

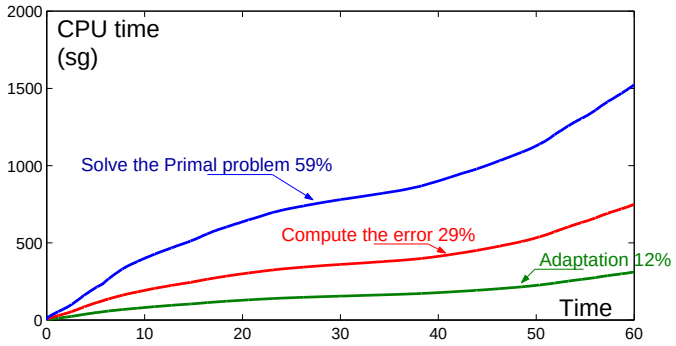

Figure 3.12: CPU time for each part of the adaptive algorithm for the functional $J_1(U)$.

Our next concern is to ascertain the reliability of the a posteriori error estimators proposed in this thesis, in particular, we are interested in knowing the values of the index of effectivity I_{eff} in each interval I_n. Ideally, we could do so for each time interval I_n if we could be able to compare the a posteriori error estimator with the true error $J(U) - J(u_{h\Delta t})$. In each time interval I_n we have the numerical solutions $\overline{u}^{n-1}_{h\Delta t}$, $u^n_{h\Delta t}$ in the mesh $\mathbb{T}^n_h$ and both the temporal and spatial a posteriori error estimators that have been calculated using the post-processing procedure, see (3.30a and 3.30b). And we shall take as "exact" solution $U = (U_1, U_2)$ the one computed in a mesh $\frac{\mathbb{T}^n_h}{2}$ with a time step $\frac{\Delta t_n}{4}$.

In Table 4a we show the results of this numerical experiment carried out in the intervals I_n (first column) of length Δt_n (second column) in a mesh of N_h nodes (third column). Note that

the end points of these intervals correspond to the time instants of the solution snapshots shown in Figure 3.9. In the fifth and sixth columns appear the temporal and spatial errors respectively. Finally, in the seventh and eighth columns are represented the total error estimator and the real error to define the effectivity index I_{eff} in the last column

$$I_{eff} = \frac{|\widetilde{e}_t^n + \widetilde{e}_s^n|}{|J(U) - J(u_{h\Delta t})|}. \tag{3.51}$$

I_n	Δt	N_h	$J(u_{h\Delta t})$	$\widetilde{e}_t^n$	$\widetilde{e}_s^n$	$\widetilde{e}_t^n + \widetilde{e}_s^n$	$J(U) - J(u_{h\Delta t})$	$\mathbf{I}_{eff}$
$[2.99, 3.01]$	0.012	4582	14.67	$5.68 \cdot 10^{-4}$	$5.53 \cdot 10^{-4}$	$1.12 \cdot 10^{-3}$	$1.39 \cdot 10^{-3}$	**0.81**
$[15.99, 16.01]$	0.016	2511	5.11	$2.50 \cdot 10^{-4}$	$5.31 \cdot 10^{-4}$	$7.81 \cdot 10^{-4}$	$6.68 \cdot 10^{-4}$	**1.17**
$[47.99, 48.01]$	0.022	4287	17.32	$2.63 \cdot 10^{-4}$	$6.08 \cdot 10^{-4}$	$8.71 \cdot 10^{-4}$	$1.14 \cdot 10^{-3}$	**0.76**
$[59.99, 60.01]$	0.013	4606	14.56	$8.35 \cdot 10^{-4}$	$7.00 \cdot 10^{-4}$	$1.53 \cdot 10^{-3}$	$1.82 \cdot 10^{-3}$	**0.84**

Table 4a: Errors and I_{eff} in different time intervals I_n for the functional $J_1(U)$.

We can also see in Table 4a the equidistribution of the error in temporal $\widetilde{e}_t^n$ and spatial part $\widetilde{e}_s^n$. In Figure 3.13 we have plotted the effectivity index I_{eff} every tenth time intervals I_n from $t = 0$ up to $t = T = 60$. Some statistics of I_{eff} are the following: mean value $E(I_{eff})= 0.805$, median $M(I_{eff})= 0.797$ and the standard deviation $\sigma(I_{eff}) = 0.1268$. These values of the effectivity index I_{eff} close to 1 indicate that the post-processing procedure works very well for this example.

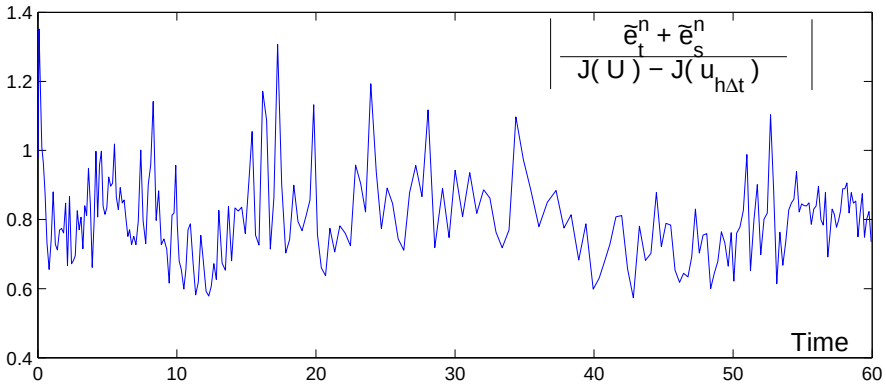

Figure 3.13: Effectivity index I_{eff} in different time intervals I_n for the functional $J_1(U)$.

Now, we are going to deal with separately the temporal and spatial error. Using Corollary 1, we can compare the a posteriori error estimator in time with the real error $J(u_h) - J(u_{h\Delta t})$ (3.24a) and the a posteriori error estimator in space with the real error $J(U) - J(u_h)$ (3.24b). The solution $u_h = (u_{1h}, u_{2h})$ is computed in the same mesh $\mathbb{T}_h^n$ as $u_{h\Delta t}^n$ with a time step $\frac{\Delta t_n}{4}$.

Temporal error:

In Table 4b we show the results of this numerical experiment carried out in the intervals I_n (first column) of length Δt_n (second column) in a mesh of N_h nodes (third column) for the temporal errors. Note that the end points of these intervals correspond to the time instants of the solution snapshots shown in Figure 3.9. In the fifth and sixth columns appear the primal and dual temporal errors respectively. Finally, in the seventh and eighth columns are represented the temporal error estimator and the real error in time respectively, and the effectivity index $I_{t,eff}$

is calculated in the ninth column.

$$I_{t,eff} = \frac{\left|\widetilde{e}^n_{t,primal} + \widetilde{e}^n_{t,dual}\right|}{|J(u_h) - J(u_{h\Delta t})|} = \frac{|\widetilde{e}^n_t|}{|J(u_h) - J(u_{h\Delta t})|} \tag{3.52}$$

I_n	Δt	N_h	$J(u_{h\Delta t})$	$\widetilde{e}^n_{t,primal}$	$\widetilde{e}^n_{t,dual}$	$\widetilde{e}^n_t$	$J(u_h) - J(u_{h\Delta t})$	$\mathbf{I}_{t,eff}$
$[2.99, 3.01]$	0.012	4582	14.67	$2.71 \cdot 10^{-4}$	$2.97 \cdot 10^{-4}$	$5.68 \cdot 10^{-4}$	$4.67 \cdot 10^{-4}$	**1.22**
$[15.99, 16.01]$	0.016	2511	5.11	$1.64 \cdot 10^{-4}$	$8.64 \cdot 10^{-5}$	$2.50 \cdot 10^{-4}$	$2.31 \cdot 10^{-4}$	**1.08**
$[47.99, 48.01]$	0.022	4287	17.32	$1.34 \cdot 10^{-4}$	$1.29 \cdot 10^{-4}$	$2.63 \cdot 10^{-4}$	$2.27 \cdot 10^{-4}$	**1.16**
$[59.99, 60.01]$	0.013	4606	14.56	$4.00 \cdot 10^{-4}$	$4.35 \cdot 10^{-4}$	$8.35 \cdot 10^{-4}$	$7.23 \cdot 10^{-4}$	**1.15**

Table 4b: Temporal Errors and $I_{t,eff}$ in different time intervals I_n for the functional $J_1(U)$.

Spatial error:

In Table 4c we show similar results as in Table 4b, but only for spatial errors. In the fifth and sixth columns appear the primal and dual temporal errors respectively. Finally, in the seventh and eighth columns are represented the temporal error estimator and the real error in time respectively, and the effectivity index $I_{s,eff}$ is calculated in the ninth column.

$$I_{s,eff} = \frac{\left|\widetilde{e}^n_{s,primal} + \widetilde{e}^n_{s,dual}\right|}{|J(U) - J(u_h)|} = \frac{|\widetilde{e}^n_s|}{|J(U) - J(u_h)|} \tag{3.53}$$

I_n	Δt	N_h	$J(u_{h\Delta t})$	$\widetilde{e}^n_{s,primal}$	$\widetilde{e}^n_{s,dual}$	$\widetilde{e}^n_s$	$J(U) - J(u_h)$	$\mathbf{I}_{s,eff}$
$[2.99, 3.01]$	0.012	4582	14.67	$3.55 \cdot 10^{-4}$	$1.98 \cdot 10^{-4}$	$5.53 \cdot 10^{-4}$	$9.23 \cdot 10^{-4}$	**0.60**
$[15.99, 16.01]$	0.016	2511	5.11	$3.36 \cdot 10^{-4}$	$1.95 \cdot 10^{-4}$	$5.31 \cdot 10^{-4}$	$4.37 \cdot 10^{-4}$	**1.22**
$[47.99, 48.01]$	0.022	4287	17.32	$4.30 \cdot 10^{-4}$	$1.78 \cdot 10^{-4}$	$6.08 \cdot 10^{-4}$	$9.11 \cdot 10^{-4}$	**0.67**
$[59.99, 60.01]$	0.013	4606	14.56	$4.65 \cdot 10^{-4}$	$2.35 \cdot 10^{-4}$	$7.00 \cdot 10^{-4}$	$1.10 \cdot 10^{-3}$	**0.64**

Table 4c: Spatial errors and $I_{s,eff}$ in different time intervals I_n for the functional $J_1(U)$.

3.4.2 Functional $J_2(U) = \int_\Omega U_1^n d\Omega$

To calculate the numerical solution to (3.42) with this functional we choose $Tol_t = Tol_s = 1 \cdot 10^{-6}$. Moreover, we shall compute another numerical solution with $Tol_t = Tol_s = 2 \cdot 10^{-7}$ considering this solution as the "exact" solution to compare with.

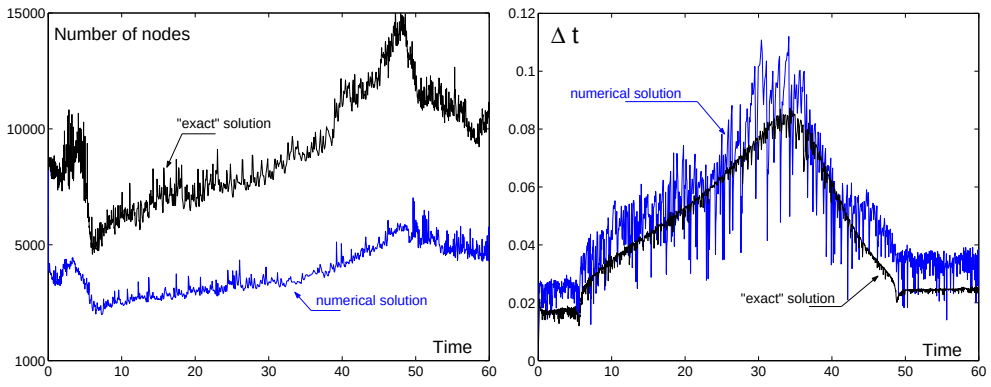

Figure 3.14: Evolution of the number of nodes N_h (left) and the time step size Δt (right) with the time for the functional $J_2(U)$.

Figure 3.14 shows the evolution in time of the number of nodes (left panel) and the magnitude of the time step Δt (right panel) for the numerical and "exact" solutions. For this example, the number of time steps needed to reach $T = 60$ is $N_{\Delta t} = 1386$, with 24 time steps rejected (only 1.7% of time steps rejected).

In Figure 3.15 we show the evolution in time of the functional for the exact solution (left panel) and the point-wise and global relative errors (right panel). The behavior of the errors now is similar to the previous example, that is, in regions where the functional is smooth, the point-wise error is small, specifically, it takes values in the interval $[0,\ 1 \cdot 10^{-5}]$, to grow when $J(U)$ loses regularity.

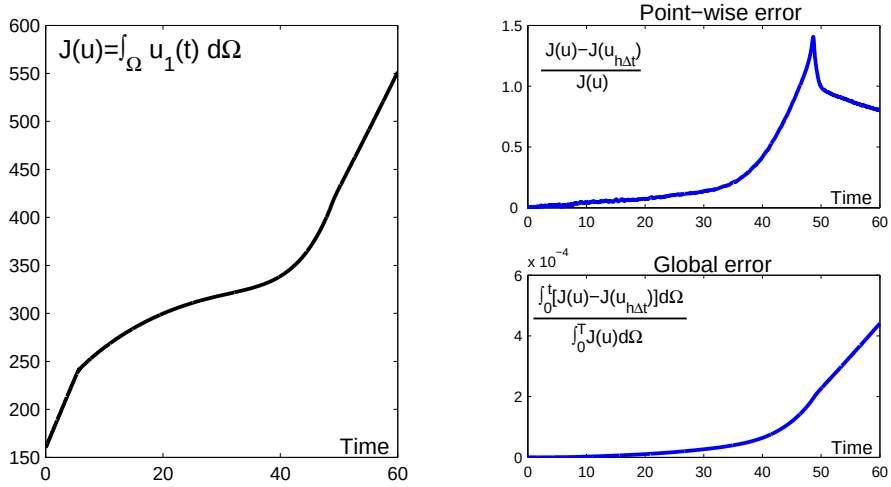

Figure 3.15: Evolution in time of the function $J(U) = \int_\Omega u_1(t) d\Omega$ (left) and point-wise (top right) and global (bottom right) errors in this quantity.

The distribution of the CPU time in seconds for each part of the method is displayed in Figure 3.16. Note that the calculation of the primal solution spends more than a sixty per cent of the total of CPU time.

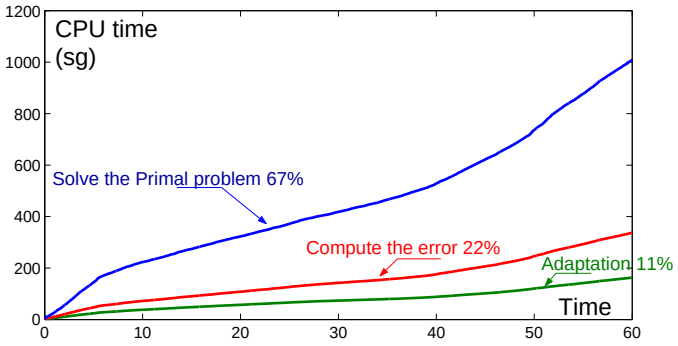

Figure 3.16: CPU time for each part of the adaptive algorithm for the functional $J_2(U)$.

Our next concern is to ascertain the reliability of the a posteriori error estimators knowing the values of the index of effectivity I_{eff} in each interval I_n. We follow the same procedure as for the functional $J_1(U)$. Thus, Tables 5a and Figure 3.17 are the analogous of Tables 4a and Figure 3.13. We note in this example that in a few intervals the effectivity index is much larger

than 1, in fact, there are 7 intervals in which $I_{eff} > 10$. Eliminating these atypical values we have that the mean $E(I_{eff}) = 1.0452$, the median $M(I_{eff}) = 0.828$ and the standard deviation $\sigma(I_{eff}) = 1.0831$. Again, we can consider that for this case the post-processing works well.

I_n	Δt	N_h	$J(u_{h\Delta t})$	$\widetilde{e}_t^n$	$\widetilde{e}_s^n$	$\widetilde{e}_t^n + \widetilde{e}_s^n$	$J(U) - J(u_{h\Delta t})$	$\mathbf{I}_{eff}$
$[2.99, 3.01]$	0.024	4295	203.4	$-3.67 \cdot 10^{-5}$	$2.26 \cdot 10^{-5}$	$-1.41 \cdot 10^{-5}$	$-1.20 \cdot 10^{-5}$	**1.17**
$[15.99, 16.05]$	0.056	2916	288.2	$-1.43 \cdot 10^{-5}$	$3.14 \cdot 10^{-6}$	$-1.12 \cdot 10^{-5}$	$-1.12 \cdot 10^{-5}$	**1.00**
$[47.99, 48.03]$	0.037	5743	402.3	$-1.23 \cdot 10^{-5}$	$2.89 \cdot 10^{-6}$	$-9.41 \cdot 10^{-6}$	$-1.20 \cdot 10^{-5}$	**0.78**
$[59.99, 60.02]$	0.032	5348	551.6	$-3.90 \cdot 10^{-5}$	$1.25 \cdot 10^{-5}$	$-2.65 \cdot 10^{-5}$	$-2.21 \cdot 10^{-5}$	**1.20**

Table 5a: Errors and I_{eff} in different time intervals I_n for the functional $J_2(U)$.

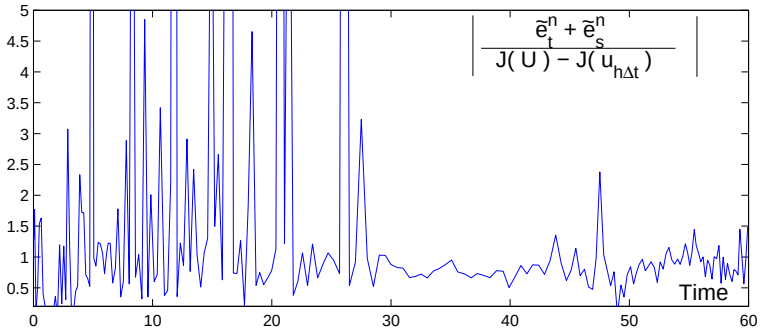

Figure 3.17: Effectivity index I_{eff} in different time intervals I_n for the functional $J_2(U)$.

Then, we show the effect of both temporal and spatial post-processing procedure. We calculate the effectivity index in time $I_{t,eff}$ and the effectivity index in space $I_{s,eff}$ using the Corollary 1. Thus, Table 5a and 5b are analogous of Table 4a and 4b for the functional $J_2(U)$.

Temporal error:

In Table 5b we show the temporal results of the numerical experiment for the functional J_2. The effectivity index $I_{t,eff}$ is calculated in the last column as (3.52).

I_n	Δt	N_h	$J(u_{h\Delta t})$	$\widetilde{e}_{t,primal}^n$	$\widetilde{e}_{t,dual}^n$	$\widetilde{e}_t^n$	$J(u_h) - J(u_{h\Delta t})$	$\mathbf{I}_{t,eff}$
$[2.99, 3.01]$	0.024	4295	203.4	$-1.76 \cdot 10^{-5}$	$-1.91 \cdot 10^{-5}$	$-3.67 \cdot 10^{-5}$	$-3.10 \cdot 10^{-5}$	**1.18**
$[15.99, 16.05]$	0.056	2916	288.2	$-6.20 \cdot 10^{-6}$	$-8.06 \cdot 10^{-6}$	$-1.43 \cdot 10^{-5}$	$-9.90 \cdot 10^{-6}$	**1.44**
$[47.99, 48.03]$	0.037	5743	402.3	$-8.80 \cdot 10^{-6}$	$-3.53 \cdot 10^{-6}$	$-1.23 \cdot 10^{-5}$	$-1.37 \cdot 10^{-5}$	**0.90**
$[59.99, 60.02]$	0.032	5348	551.6	$-1.92 \cdot 10^{-5}$	$-1.98 \cdot 10^{-5}$	$-3.90 \cdot 10^{-5}$	$-3.40 \cdot 10^{-5}$	**1.15**

Table 5b: Temporal errors and $I_{t,eff}$ in different time intervals I_n for the functional $J_2(U)$.

Spatial error:

In Table 5c we show the spatial results of the numerical experiment for the functional J_2. The effectivity index $I_{s,eff}$ is calculated by the formula (3.53) in the last column.

I_n	Δt	N_h	$J(u_{h\Delta t})$	$\widetilde{e}_{s,primal}^n$	$\widetilde{e}_{s,dual}^n$	$\widetilde{e}_s^n$	$J(U) - J(u_h)$	$\mathbf{I}_{s,eff}$
$[2.99, 3.01]$	0.024	4295	203.4	$6.56 \cdot 10^{-6}$	$1.60 \cdot 10^{-5}$	$2.26 \cdot 10^{-5}$	$1.91 \cdot 10^{-5}$	**1.18**
$[15.99, 16.05]$	0.056	2916	288.2	$1.07 \cdot 10^{-6}$	$2.07 \cdot 10^{-6}$	$3.14 \cdot 10^{-6}$	$8.79 \cdot 10^{-6}$	**0.36**
$[47.99, 48.03]$	0.037	5743	402.3	$-9.95 \cdot 10^{-7}$	$3.89 \cdot 10^{-6}$	$2.89 \cdot 10^{-6}$	$1.26 \cdot 10^{-6}$	**2.29**
$[59.99, 60.02]$	0.032	5348	551.6	$7.07 \cdot 10^{-7}$	$5.44 \cdot 10^{-6}$	$1.25 \cdot 10^{-5}$	$1.20 \cdot 10^{-5}$	**1.04**

Table 5c: Spatial errors and $I_{s,eff}$ in different time intervals I_n for the functional $J_2(U)$.

3.4.3 Comparison of the results for both functionals

In this subsection we will compare the different meshes and time step sizes for the two functionals. The evolution of the number of nodes and the time step size can be observed in Figure 3.18, where we can see that the number of nodes and the size of the time steps are larger in the second functional than in the first. In fact, for the first functional the number of the time steps needed to reach $t = T = 60$ is 2981, whereas for the second functional the number of time steps is 1386, less than half of the time steps needed for the first one.

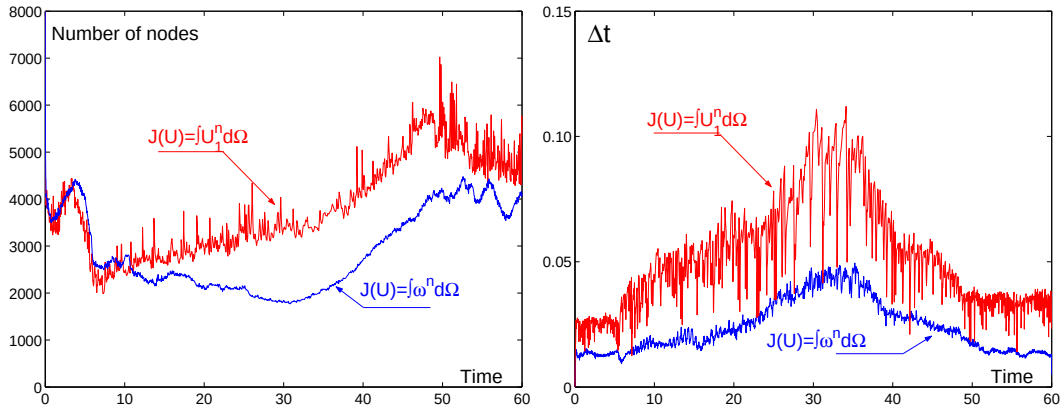

Figure 3.18: Comparative between the evolution of the number of nodes N_h (left) and the time step Δt (right) for both functionals $J_1(U)$ and $J_2(U)$.

Figure 3.19 shows the meshes obtained for the two functionals at different time instants. Using the first functional many elements are necessary to have a good resolution of the flame front; however, to control the second functional we must extend the refinement in the domain having an especial care in the non-convex corners.

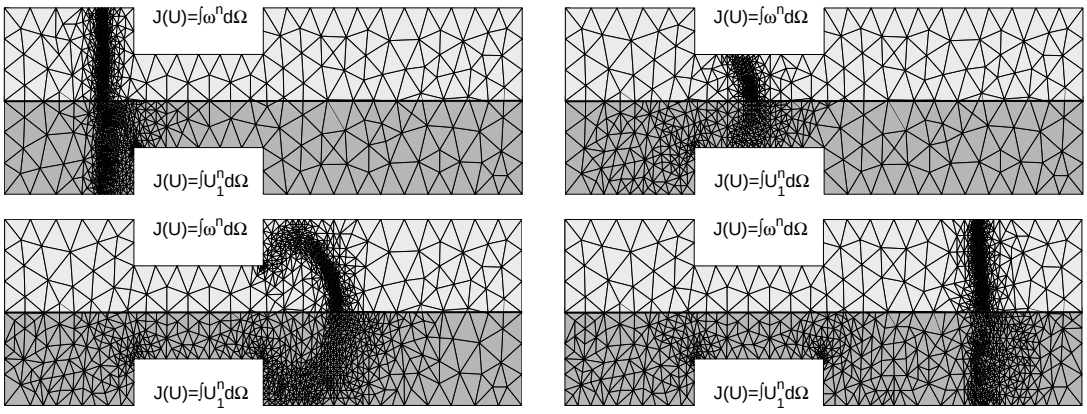

Figure 3.19: Comparative between the mesh at time t = 3, 16, 48 and 60 for both functionals $J_1(U)$ and $J_2(U)$.

3.4.4 Comparison with the results of the global DWR method of M. Schmich and B. Vexler

Now, we are interested in comparing our results obtained by the local DWR methodology with those provided by the global DWR method implemented by M. Schmich and B. Vexler [61]. The primal problem (3.42) is the same in both cases and to define the mesh $\mathbb{T}_h^n$ and the time step Δt_n size we use, to control the error, the local in time functional $J_1(U) = \int_\Omega \omega(U_1^n, U_2^n) d\Omega$, whereas M. Schmich and B. Vexler employ a global in time functional $J(u) = \frac{1}{60\,|\Omega|}\int_0^T \left(\int_\Omega \omega(u_1,u_2)d\Omega\right) dt$ (however, both functionals are global in space, since in time dependent problems our method does not work well with local in space functionals such as line integrals).

For each numerical experiment, we are interested in the following quantities: N_t number of time steps, N_{max} maximum number of nodes for the spatial mesh and the value of error $J(u) - J(u_{h\Delta t})$ in the functional

$$\begin{cases} J(u) = \dfrac{1}{60\,|\Omega|}\displaystyle\int_0^T \left(\int_\Omega \omega(u_1,u_2)d\Omega\right) dt, \\ |\Omega| = 840, \quad T = 60. \end{cases}$$

In Table 6a, we have copied Table 5.1 from [61]. To obtain these results, they use bilinear finite elements on structured quadrilateral meshes with hanging nodes for spacial discretization and cG(1)dG(0) scheme for time discretization.

N_t	$N_{\max}$	$J(u) - J(u_{h\Delta t})$
256	985	$-1.703 \cdot 10^{-3}$
396	985	$-1.413 \cdot 10^{-3}$
616	1427	$-8.718 \cdot 10^{-4}$
872	2309	$-4.013 \cdot 10^{-4}$
1370	3927	$-1.569 \cdot 10^{-4}$
1528	6927	$-8.630 \cdot 10^{-5}$
1772	14683	$-5.941 \cdot 10^{-5}$

Table 6a: Bilinear cG(1)dG(0) M. Schmich and B. Vexler.

The following tables, Table 6b and Table 6c, have been calculated using our local DWR technique to make error control. We have always used the cG(1)dG(0) scheme in time and both linear and quadratic finite elements in space.

N_t	$N_{\max}$	$J(u) - J(u_{h\Delta t})$	$Tol_s = Tol_t$
1215	3927	$-3.35 \cdot 10^{-5}$	$5 \cdot 10^{-3}$
1760	6172	$-8.56 \cdot 10^{-5}$	$2 \cdot 10^{-3}$
2596	9631	$-2.23 \cdot 10^{-5}$	$1 \cdot 10^{-3}$

Table 6b: Linear cG(1)dG(0)

N_t	$N_{\max}$	$J(u) - J(u_{h\Delta t})$	$Tol_s = Tol_t$
1193	2150	$7.95 \cdot 10^{-5}$	$5 \cdot 10^{-3}$
1692	2760	$1.09 \cdot 10^{-5}$	$2 \cdot 10^{-3}$
2273	3150	$-3.78 \cdot 10^{-6}$	$1 \cdot 10^{-3}$

Table 6c: Quadratic cG(1)dG(0)

Local DWR technique

Results for linear finite elements with the local DWR scheme (Table 6b) are similar to those of M. Schmich and B. Vexler (Table 6a). However, using quadratic finite elements (Table 6c), our results are better than those showed in Table 6a.

3.5 DWR methodology with Runge-Kutta-Chebyshev schemes in time

We show again the primal problem to be solved in each time interval $I_n := (t_{n-1}, t_n]$.

$$\textit{For } n = 1, 2, .., N, \textit{find } u_{h\Delta t}(x,t) \in V^{(r)}_{h\Delta t} \textit{ such that for all } \psi_{h\Delta t}(x,t) \in W^{(s)}_{h\Delta t}$$

$$\begin{cases} \displaystyle\int_{I_n} \{(\partial_t u_{h\Delta t}, \psi_{h\Delta t})_\Omega + a(u_{h\Delta t}, \psi_{h\Delta t})\}\, dt + ([u_{h\Delta t}]^{n-1}, \psi^{n-1+}_{h\Delta t})_\Omega = \int_{I_n} (f(u_{h\Delta t}), \psi_{h\Delta t})_\Omega\, dt, \\ \\ u_{h\Delta t}(x, t_{n-1}) = \overline{u}^{n-1}_{h\Delta t}(x). \end{cases} \tag{3.54}$$

To obtain the numerical solution of our primal problem we have presented two different techniques in time: cG(0)dG(0) that yields a conventional Implicit Euler scheme (3.10) and cG(1)dG(0) that yields a version of the Crank-Nicolson scheme (3.9), both of them being implicit schemes. However, implicit methods require the use of robust iterative methods for nonlinear equations such as the Newton method, or some of its variants, plus a preconditioned iterative linear solver. An approach to overcome this problem, and obtain a good solution with a small cost in terms of CPU time and memory requirements, is to apply a Runge-Kutta-Chebyshev (hereafter RKC) scheme (the first version, Explicit RKC scheme [65] and more recently Implicit-Explicit RKC scheme [71]). There are RKC schemes of order 1 and 2, however, we only consider the second-order scheme as this is believed to be more efficient than the first-order one.

Historically, the principal goal when constructing Runge-Kutta schemes was to achieve the highest order possible with a given number of stages r. Stabilized methods differ in that only a few stages are used to achieve an usually low order whereas additional stages are exploited to increase the region of absolute stability. RKC schemes have been developed with the last philosophy and initially were intended for parabolic PDE problems, that have previously been spatially discretized and take the form

$$\mathbf{Y}'(t) = \mathbf{F}(t, \mathbf{Y}), \quad t > 0, \quad \mathbf{Y}(0) = \mathbf{Y}_0, \tag{3.55}$$

$\mathbf{F}(t, \mathbf{Y})$ is further decomposed as $\mathbf{F}(t, \mathbf{Y}) = \mathbf{F}_D(t, \mathbf{Y}) + \mathbf{F}_R(t, \mathbf{Y})$, where the subscripts D and R stand for semi-discrete diffusion operator and reaction terms respectively. Typically, the dimension of this ODE system is large, especially for multi-space dimensional PDEs, and usually this system is nonlinear and stiff.

Explicit Runge-Kutta-Chebyshev (EX-RKC) scheme

The EX-RKC method for reaction-diffusion equations is explicit, thus avoiding the resolution of an algebraic system. Additionally, this scheme presents a stability region that can be extended by increasing the number of stages r of the method (such number depends on the stiffness of the system) but always maintaining a temporal error of second order. In this kind of schemes the stability region is a narrow strip along the negative real axis in the complex plane, with real stability boundary $\beta \approx r^2$, where this quadratic dependency is derived from the first kind Chebyshev polynomial.

The EX-RKC scheme has the following algorithm (see [65]):

Suppose that at time t_{n-1} the numerical solution $\mathbf{Y}^{n-1}$ is known and we wish to calculate the solution $\mathbf{Y}^n$ at time t_n. We consider $\Delta t = t_n - t_{n-1}$ and $r \geq 2$, r being an integer denoting the number of stages of the method. We set

$$\begin{cases} \mathbf{W}_0 = \mathbf{Y}^{n-1}, \\ \mathbf{W}_1 = \mathbf{W}_0 + \widetilde{\mu}_1 \Delta t \mathbf{F}_0, \\ \text{for } j = 2 \text{ to } r \\ \quad \mathbf{W}_j = (1-\mu_j-\nu_j)\mathbf{W}_0 + \mu_j \mathbf{W}_{j-1} + \nu_j \mathbf{W}_{j-2} + \widetilde{\mu}_j \Delta t \mathbf{F}_{j-1} + \widetilde{\gamma}_j \Delta t \mathbf{F}_0 \\ \text{end} \\ \mathbf{Y}^n = \mathbf{W}_r \end{cases} \tag{3.56}$$

where $\mathbf{W}_0, ..., \mathbf{W}_r$ are internal vectors and $\mathbf{F}_k := \mathbf{F}(t_{n-1} + c_k \Delta t, \mathbf{W}_k)$. All coefficients in (3.56) are available in analytical form [65]. For convenience of the reader we write down the expressions of such coefficients. Let $T_j(x)$ denote the Chebyshev polynomial of first kind and degree j satisfying the recursive relation

$$T_j(x) = 2xT_{j-1}(x) - T_{j-2}(x), \quad 2 \leq j \leq r \text{ and } -1 \leq x \leq 1$$

where $T_0(x) = 1$ and $T_1(x) = x$.

$$\begin{cases} \epsilon = \dfrac{2}{13}, \quad w_0 = 1 + \dfrac{\varepsilon}{r^2}, \quad w_1 = \dfrac{T_r'(w_0)}{T_r''(w_0)}, \quad b_0 = b_1 = b_2, \quad b_j = \dfrac{T_j''(w_0)}{\left(T_j'(w_0)\right)^2} \ (2 \leq j \leq r), \quad \widetilde{\mu}_1 = b_1 w_1, \\ \qquad a_j = 1 - b_j T_j(w_0), \quad \mu_j = \dfrac{2b_j w_0}{b_{j-1}}, \quad \nu_j = -\dfrac{b_j}{b_{j-2}}, \quad \widetilde{\mu}_j = \dfrac{2b_j w_1}{b_{j-1}}, \quad \widetilde{\gamma}_j = -a_{j-1}\widetilde{\mu}_j \\ \qquad c_0 = 0, \quad c_1 = \dfrac{c_2}{T_2'(w_0)} = \dfrac{c_2}{4w_0}, \quad c_j = \dfrac{T_r'(w_0)}{T_r''(w_0)} \dfrac{T_j'(w_0)}{T_j'(w_0)} \simeq \dfrac{j^2-1}{r^2-1} \ (2 \leq j \leq r). \end{cases} \tag{3.57}$$

The parameter $\epsilon \geq 0$ is a free parameter. This parameter is called a damping parameter and usually takes the value $\epsilon = 2/13$ in second-order RKC scheme.

Implicit-Explicit Runge-Kutta-Chebyshev (IMEX-RKC) scheme

For systems from low to moderate stiffness, EX-RKC scheme is very efficient. However, in case of severe stiffness this scheme could become inefficient because a very large number of stages would be needed to achieve stability with reasonable step sizes (in general, if $r > 50$ EX-RKC scheme may lose efficiency). The new IMEX-RKC scheme has been designed to overcome this shortcoming, the diffusion term $\mathbf{F}_D(t, \mathbf{Y})$, is assumed to be moderately stiff and it is treated explicitly. The reaction term $\mathbf{F}_R(t, \mathbf{Y})$ is strongly stiff and it is treated implicitly.

The IMEX-RKC scheme follows the algorithm below (see [71]):

Suppose that at time t_{n-1} the numerical solution $\mathbf{Y}^{n-1}$ is known and we wish to calculate the solution $\mathbf{Y}^n$ at time t_n. We consider $\Delta t = t_n - t_{n-1}$ and $r \geq 2$, r being an integer denoting the number of stages of the method. We set

$$\begin{cases} \mathbf{W}_0 = \mathbf{Y}^{n-1}, \\ \mathbf{W}_1 = \mathbf{W}_0 + \widetilde{\mu}_1 \Delta t \mathbf{F}_{D,0} + \widetilde{\mu}_1 \Delta t F_{R,1}, \\ \text{for } j = 2 \text{ to } r \\ \quad \mathbf{W}_j = (1-\mu_j-\nu_j)\mathbf{W}_0 + \mu_j \mathbf{W}_{j-1} + \nu_j \mathbf{W}_{j-2} + \widetilde{\mu}_j \Delta t \mathbf{F}_{D,j-1} + \widetilde{\gamma}_j \Delta t \mathbf{F}_{D,0} + \\ \qquad + \left[\widetilde{\gamma}_j - \left(1-\mu_j-\nu_j\right)\widetilde{\mu}_1\right] \Delta t \mathbf{F}_{R,0} - \nu_j \widetilde{\mu}_1 \Delta t \mathbf{F}_{R,j-2} + \widetilde{\mu}_1 \Delta t \mathbf{F}_{R,j} \\ \text{end} \\ \mathbf{Y}^n = \mathbf{W}_r \end{cases} \tag{3.58}$$

With this scheme, we have to solve at each stage a system of nonlinear algebraic equations, which has a nice property: this system can be broken into as many subsystems as grid points in

the mesh, with the size of these systems being equal to the number of unknown functions of the parabolic problem. The form of the nonlinear systems is

$$\mathbf{W}_j - \widetilde{\mu}_1 \Delta t \mathbf{F}_R(\mathbf{W}_j, t_{n-1} + c_j \Delta t) = \mathbf{V}_j.$$

These small size systems can be dealt with by the classic Newton method because it is always possible to start the iterations with a guest solution close enough to the true one.

The coefficients in (3.58) take the same expression as in (3.57) with only one difference in the definition of the coefficients $b_{0,}$ b_1 and b_2.

$$b_0 = b_2, \quad b_1 = \frac{1}{w_0}.$$

The number of stages r

Given a time step Δt_n, we estimate the number of stages r that make the explicit part of the RKC method be absolutely stable for such Δt_n as [69]

$$r = 1 + interger\ part \left[\sqrt{1 + \frac{\rho(-\Delta t \cdot J_{F_{explicit}})}{0.653}} \right]$$

where $\rho(-\Delta t \cdot J_{F_{explicit}})$ denotes the spectral radius of the matrix $-\Delta t \cdot J_{F_{explicit}}$, here $J_{F_{explicit}}$ is the Jacobian matrix of the explicit part of the right hand side ODE system (3.55). We employ the power method to calculate the spectral radius $\rho(J_{F_{explicit}})$.

Note that in EX-RKC scheme $\mathbf{F}_{explicit}(t, \mathbf{Y}) = \mathbf{F}_D(t, \mathbf{Y}) + \mathbf{F}_R(t, \mathbf{Y})$ and in IMEX-RKC scheme $\mathbf{F}_{explicit}(t, \mathbf{Y}) = \mathbf{F}_D(t, \mathbf{Y})$. For the examples presented in this Chapter and in Chapter 5, we make a fine refinement of the spatial mesh $\mathbb{T}_h^n$ to obtain always a diffusion term with a similar stiffness to that of the reaction term, therefore we will use the EX-RKC scheme to solve the primal problem (3.54).

For the application of the DWR methodology combined with a RKC scheme to compute the numerical solution to the model problem (3.54), we shall work with the trial and test functional spaces $V_{h\Delta t}^{(1)}$ and $W_{h\Delta t}^{(0)}$, respectively. Thus, for each n, the numerical solution to the continuous problem is given by:

$$\begin{cases} u_{h\Delta t}(x,t)|_{I_n} \in V_{h\Delta t}^{(1)}, \\ \\ u_{h\Delta t}(x,t)|_{I_n} := \overline{u}_{h\Delta t}^{n-1}(x) + \dfrac{t - t_{n-1}}{\Delta t_n}(u_{h\Delta t}^n(x) - \overline{u}_{h\Delta t}^{n-1}(x)), \end{cases} \tag{3.59}$$

where $\overline{u}_{h\Delta t}^{n-1}(x)$ is the interpolatory projection of $u_{h\Delta t}^{n-1}(x) \in V_h^{n-1}$ onto the finite element space V_h^n associated to the partition $\mathbb{T}_h^n$, and $u_{h\Delta t}^n(x) := u_{h\Delta t}(x, t_n)|_{I_n} \in V_h^n$. Hence, $\overline{u}_{h\Delta t}^{n-1}(x)$ and $u_{h\Delta t}^n(x)$ as elements of V_h^n can be expressed as

$$\overline{u}_{h\Delta t}^{n-1}(x) = \sum_{j=1}^{N_h^n} \overline{U}_{h\Delta t,j}^{n-1} \varphi_j(x)$$

and

$$u_{h\Delta t}^n(x) = \sum_{j=1}^{N_h^n} U_{h\Delta t,j}^n \varphi_j(x),$$

where N_h^n denotes the number of mesh points in the partition $\mathbb{T}_h^n$ and $\left\{\varphi_j\right\}_{j=1}^{N_h^n}$ is the set of global basis functions of V_h^n.

Since in each I_n, the numerical solution, $U_{h\Delta t}(x,t) \in V_{h\Delta t}^{(1)}$, to the primal problem (3.16) is, precisely, the numerical solution $u_{h\Delta t}(x,t) \in V_{h\Delta t}^{(1)}$ to the continuous problem (3.54), then $u_{h\Delta t}(x,t)$ is determined if one calculates $u_{h\Delta t}^n(x) \in V_h^n$. We do this by solving the following problem:

For $t \in I_n$, find $u_{h\Delta t}^n(x) \in V_h^n$ such that for $\psi_{h\Delta t} \in W_{h\Delta t}^{(0)}$

$$\begin{cases} (\partial_t u_{h\Delta t}, \psi_{h\Delta t})_\Omega + a(u_{h\Delta t}, \psi_{h\Delta t}) = (f(u_{h\Delta t}), \psi_{h\Delta t})_\Omega, \\ \\ u_{h\Delta t}(x, t_{n-1}) = \overline{u}_{h\Delta t}^{n-1}(x) \in V_h^n. \end{cases} \tag{3.60}$$

However, the dual problem is stated as in (3.17); that is, for all $\varphi_{h\Delta t} \in V_{h\Delta t}^{(1)}$, $z_{h\Delta t} \in W_{h\Delta t}^{(0)}$ solves

$$J'(u_{h\Delta t})(\varphi_{h\Delta t}) - \int_{I_n} \{(-\partial_t z_{h\Delta t}, \varphi_{h\Delta t})_\Omega + a(z_{h\Delta t}, \varphi_{h\Delta t}) - (f'(u_{h\Delta t})\varphi_{h\Delta t}, z_{h\Delta t})_\Omega\}\, dt - (\varphi_{h\Delta t}^{n-}, z_{h\Delta t}^{n-})_\Omega = 0. \tag{3.61}$$

We remark that in the dual problem we shall take

$$\varphi_{h\Delta t}(x) = \frac{t - t_{n-1}}{\Delta t_n}\varphi_h(x), \quad \varphi_h(x) \in V_h^n.$$

Next, noting that (3.60) yields the system of differential equations

$$[M]\,\frac{d\mathbf{Y}}{dt} + [S]\,\mathbf{Y} = F(\mathbf{Y}), \tag{3.62}$$

where $[M]$ and $[S] \in \mathbb{R}^{N_h^n \times N_h^n}$ are the so called mass and stiffness matrices, $\mathbf{Y}$ and $F(\mathbf{Y})$ are $N_h^n - dimensional$ column vectors,

$$\mathbf{Y} = \left[U_{h\Delta t,1}(t), ..., U_{h\Delta t,N_h^n}(t)\right]^T,$$

$$F(\mathbf{Y}) = \left[F_1(\mathbf{Y}), ..., F_{N_h^n}(\mathbf{Y})\right]^T,$$

with

$$F_i(\mathbf{Y}) = \int_\Omega f(u_{h\Delta t})\varphi_i(x) d\Omega.$$

$[M]$ and $[S]$ are symmetric positive definite matrices, the entries of which are given by

$$m_{ij} = \int_\Omega \varphi_i \varphi_j d\Omega, \qquad s_{ij} = \int_\Omega \epsilon \nabla\varphi_i \nabla\varphi_j d\Omega.$$

By applying RKC formulas to (3.62) we shall obtain $u_{h\Delta t}^n(x) \in V_h^n$ as solution of the system of algebraic equations (3.56) or (3.58).

As all the second order RKC schemes maintain the temporal error independently of the number of stages r, we consider a linear evolution of the solution $u_{h\Delta t}(x,t)$ given by the formula (3.59) to calculate the a posteriori error. Then, we use the same formulas of cG(1)dG(0) (a type of Crank-Nicolson) scheme to make the post-processing procedure, substituting into (3.30a and 3.30b).

Now, we will solve the problem defined in section 3.4, namely, the propagation of laminar flames according to the model equations proposed before in (3.42). We adapt the spatial mesh and the time step using the first functional $J_1(U) = \int_\Omega \omega(U_1^n, U_2^n) d\Omega$.

In the Figure 3.20 we have plotted the number of nodes N_h and the time step size Δt for the different time discretizations (CN and RKC schemes). We can observe as the time step size in the RKC case is larger than CN one and the number of nodes too. Furthermore with the CN scheme we have needed 2981 time steps (with 31 rejects 1%) and with RKC 1850 time steps (with only 9 rejects 0.5%).

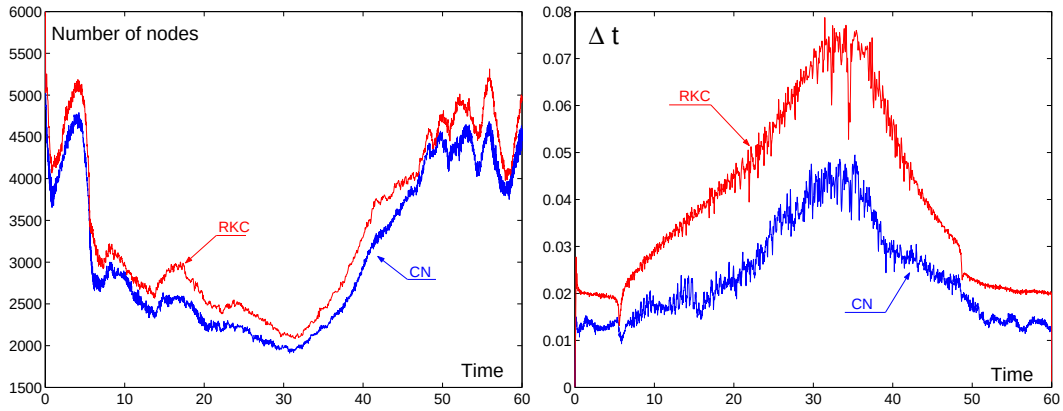

Figure 3.20: Comparative between the evolution of the number of nodes N_h (left) and the time step Δt (right) for both CN and RKC methods for the functional $J_1(U)$.

In Figure 3.21 we show the number of stages in the EX-RKC scheme. We can see that when the flame comes into the obstacle the number of stages reaches a peak.

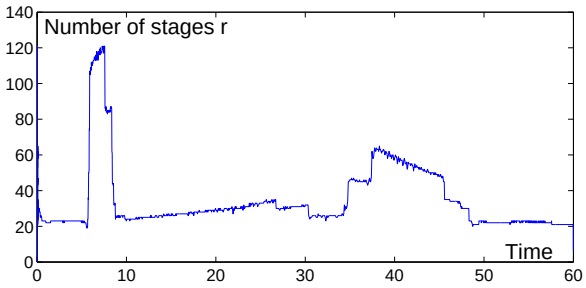

Figure 3.21: Number of stages r with the EX-RKC scheme

Now, we are interested in knowing the values of the effectivity index I_{eff} in each interval I_n. We expect that the effectivity index of the error in space $I_{s,eff}$ has a good behaviour and be close to 1, but the effectivity index in time $I_{t,eff}$ could have values away from the value 1. In fact, $I_{t,eff}$ will take values bigger than 1, so we will have an over-estimation of the true error. In each time subinterval I_n we have the numerical solutions $\overline{u}_{h\Delta t}^{n-1}$, $u_{h\Delta t}^n$ in the mesh $\mathbb{T}_h^n$ and both the temporal and spatial a posteriori error estimators that have been calculated using the post-processing procedure, see (3.30a and 3.30b).

To calculate the total effectivity index I_{eff} in each interval I_n we have to evaluate the true error $J(U) - J(u_{h\Delta t})$ and we shall take as "exact" solution $U = (U_1, U_2)$ the one computed in a

mesh $\frac{\mathbb{T}_h^n}{2}$ with a time step $\frac{\Delta t_n}{4}$. In Table 7a we show the numerical experiment carried out in the intervals I_n corresponding to the time instants of the solution snapshots shown in Figure 3.9.

I_n	Δt	N_h	$J(u_{h\Delta t})$	$\widetilde{e}_t^n$	$\widetilde{e}_s^n$	$\widetilde{e}_t^n + \widetilde{e}_s^n$	$J(U) - J(u_{h\Delta t})$	$\mathbf{I}_{eff}$
$[2.99, 3.01]$	0.019	4869	14.67	$1.45 \cdot 10^{-3}$	$4.47 \cdot 10^{-4}$	$1.90 \cdot 10^{-3}$	$5.46 \cdot 10^{-4}$	**3.48**
$[15.99, 16.03]$	0.038	2953	5.10	$7.52 \cdot 10^{-4}$	$5.90 \cdot 10^{-4}$	$1.34 \cdot 10^{-3}$	$5.24 \cdot 10^{-4}$	**2.55**
$[47.99, 48.03]$	0.031	4510	16.97	$1.37 \cdot 10^{-3}$	$4.33 \cdot 10^{-4}$	$1.80 \cdot 10^{-3}$	$5.59 \cdot 10^{-4}$	**3.22**
$[59.99, 60.01]$	0.020	4986	14.55	$1.51 \cdot 10^{-3}$	$5.39 \cdot 10^{-4}$	$2.05 \cdot 10^{-3}$	$5.91 \cdot 10^{-4}$	**3.47**

Table 7a: Errors and I_{eff} in different time intervals I_n for the functional $J_1(U)$, with the RKC method to solve the primal problem.

We can also see in Table 7a the equidistribution of the error in temporal $\widetilde{e}_t^n$ and spatial part $\widetilde{e}_s^n$. In Figure 3.13 we have plotted the effectivity index I_{eff} every tenth time intervals I_n from $t = 0$ up to $t = T = 60$. Some statistics of I_{eff} are the following: mean value $E(I_{eff})$= 4.625, median $M(I_{eff})$= 3.947 and the standard deviation $\sigma(I_{eff}) = 2.76$. This means that the error estimator is 4.625 times larger (more or less constant for all the time intervals I_n) than the real error, therefore our criteria is conservative.

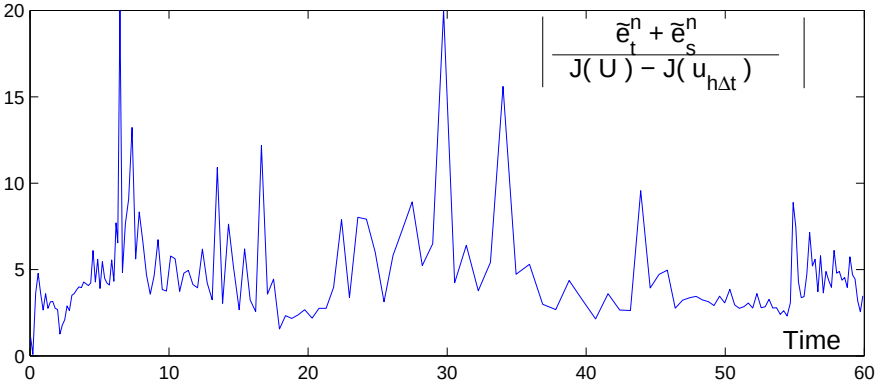

Figure 3.22: Effectivity index I_{eff} in different time intervals I_n for the functional $J_1(U)$.

Now, we are going to deal separately with the temporal and spatial error. Using Corollary 1, we can compare the a posteriori error estimator in time with the real error $J(u_h) - J(u_{h\Delta t})$ (3.24a) and the a posteriori error estimator in space with the real error $J(U) - J(u_h)$ (3.24b). The solution $u_h = (u_{1h}, u_{2h})$ is computed in the same mesh $\mathbb{T}_h^n$ that $u_{h\Delta t}^n$ with a time step $\frac{\Delta t_n}{4}$.

Temporal error:

In Table 7b we show the results for the error in time $J(u_h) - J(u_{h\Delta t})$ in the time subinterval I_n. The effectivity index $I_{t,eff}$ appears in the ninth column.

I_n	Δt	N_h	$J(u_{h\Delta t})$	$\widetilde{e}_{t,primal}^n$	$\widetilde{e}_{t,dual}^n$	$\widetilde{e}_t^n$	$J(u_h) - J(u_{h\Delta t})$	$\mathbf{I}_{t,eff}$
$[2.99, 3.01]$	0.019	4869	14.67	$6.99 \cdot 10^{-4}$	$7.48 \cdot 10^{-4}$	$1.45 \cdot 10^{-3}$	$-2.90 \cdot 10^{-4}$	**5.00**
$[15.99, 16.03]$	0.038	2953	5.10	$3.55 \cdot 10^{-4}$	$3.97 \cdot 10^{-4}$	$7.52 \cdot 10^{-4}$	$-1.52 \cdot 10^{-4}$	**4.95**
$[47.99, 48.03]$	0.031	4510	16.97	$6.55 \cdot 10^{-4}$	$7.18 \cdot 10^{-4}$	$1.37 \cdot 10^{-3}$	$-2.31 \cdot 10^{-4}$	**5.93**
$[59.99, 60.01]$	0.020	4986	14.55	$7.36 \cdot 10^{-4}$	$7.77 \cdot 10^{-4}$	$1.51 \cdot 10^{-3}$	$-3.17 \cdot 10^{-4}$	**4.76**

Table 7b: Temporal Errors and $I_{t,eff}$ in different time intervals I_n for the functional $J_1(U)$ with the EX-RKC method used to solve the primal problem.

Spatial error:

In Table 7c we show the results for the error in space $J(U) - J(u_h)$ in the time subinterval I_n. The effectivity index $I_{s,eff}$ appears in the ninth column.

I_n	Δt	N_h	$J(u_{h\Delta t})$	$\widetilde{e}^n_{s,primal}$	$\widetilde{e}^n_{s,dual}$	$\widetilde{e}^n_s$	$J(U) - J(u_h)$	$\mathbf{I}_{s,eff}$
$[2.99, 3.01]$	0.019	4869	14.67	$3.36 \cdot 10^{-4}$	$1.37 \cdot 10^{-4}$	$4.47 \cdot 10^{-4}$	$8.36 \cdot 10^{-4}$	**0.53**
$[15.99, 16.03]$	0.038	2953	5.10	$3.74 \cdot 10^{-4}$	$2.16 \cdot 10^{-4}$	$5.90 \cdot 10^{-4}$	$6.76 \cdot 10^{-4}$	**0.87**
$[47.99, 48.03]$	0.031	4510	16.97	$2.58 \cdot 10^{-4}$	$1.75 \cdot 10^{-4}$	$4.33 \cdot 10^{-4}$	$7.88 \cdot 10^{-4}$	**0.55**
$[59.99, 60.01]$	0.020	4986	14.55	$3.88 \cdot 10^{-4}$	$1.51 \cdot 10^{-4}$	$5.39 \cdot 10^{-4}$	$9.08 \cdot 10^{-4}$	**0.59**

Table 7c: Spatial errors and $I_{s,eff}$ in different time intervals I_n for the functional $J_1(U)$ with the EX-RKC method to solve the primal problem.

Chapter 4

CONVECTION REACTION-DIFFUSION PROBLEMS

This chapter is devoted to the formulation and testing of a space-time adaptive algorithm based on the DWR technique for convection-reaction-diffusion problems. The algorithm we propose in this chapter follows the spirit of the adaptive algorithm introduced in Chapter 3 for reaction-diffusion problems; that is, chosen a target functional to be calculated accurately, the algorithm solves at each time step both the discrete primal and its associated dual problems to calculate an a posteriori error estimate. As we saw in Chapter 3 this approach is efficient in computational terms, but it allows to only have local control of the error; this may be considered as a weak point of our method because, looking at the features of the behavior of the error inherent to any approximation method, we can distinguish two processes:

i) *Global error transport.* The local error committed at the element K_i of the mesh at time instant t_n is strongly affected by the residuals of a distant element K_j at time instant t_m. This is the so called *"pollution effect"*.

ii) *Interaction of the error components*. The error in one component of the solution may depend in a complicated way on the element residuals at different time instants.

An effective method for error estimation should include all these dependencies. The effect of the residuals at distant elements K_j at time instant t_m on the residuals of the the element K_i at time t_n is governed by the Green function of the continuous problem. In convection-reaction-diffusion problems the error propagation depends on the characteristics of the operator. Thus, the diffusion terms isotropically smooth out the error, but they may contribute to the global error propagation from local irregularities. The convective terms propagate the errors in the transport direction, but the errors decay exponentially in the crosswind direction. Finally, the reaction terms cause isotropic exponential decay, but if they are stiff the components of the error are coupled in a complex way.

4.1 The model problem

To make simple the presentation of the adaptive algorithm for convection-reaction-diffusion equations, we shall consider a model problem in a bounded domain $\Omega \subset \mathbb{R}^2$ with sufficiently

smooth boundary $\partial\Omega$,

$$\begin{cases} \dfrac{Du}{Dt} = \epsilon\Delta u + f(u,x,t) \quad \text{in } \Omega \times (0,T], \\ \\ u(x,0) = u^0(x) \text{ in } \Omega, \\ \\ u|_{\partial\Omega} = 0 \ \ t > 0. \end{cases} \tag{4.1}$$

Here, the diffusion parameter $\epsilon > 0$ is supposed to be constant, the reaction term $f : C^l(\mathbb{R}) \times \Omega \times (0,T] \to \mathbb{R}$, $l \geq 1$ integer, satisfies suitable growth conditions and the total derivative operator

$$\frac{D}{Dt} := \frac{\partial}{\partial t} + \mathbf{a}(x,t) \cdot \nabla, \tag{4.2}$$

where the vector velocity $\mathbf{a}(x,t)$ satisfies

$$\mathbf{a}(x,t) \in L^\infty(0,T; W_0^{1,\infty}(\mathbb{R}^2)^2) \text{ and } \forall t > 0 \ \mathbf{a}(x,t) \cdot \mathbf{n} \mid_{\partial\Omega} = g(x,t) \in L^2(\partial\Omega). \tag{4.3}$$

$\mathbf{n}$ being the outward unit normal vector on $\partial\Omega$. Furthermore, we shall consider a fixed open bounded domain $\Omega^* \supset \overline{\Omega}$ and assume that for all non-negative integers k and real numbers p in the range $1 \leq p \leq \infty$, there exists an extension operator

$$E : W^{k,p}(\Omega) \to W^{k,p}(\Omega^*),$$

such that $C^1(\overline{\Omega})$ functions are mapped onto $C^1(\overline{\Omega}^*)$ functions and

$$Eu|_\Omega = u.$$

Under these conditions there exists a unique weak solution to problem (4.1) $u \in L^2(0,T; H_0^1(\Omega)) \cap C((0,T; L^2(\Omega)) \cap L^p(\Omega \times (0,T))$ such that for all $v \in L^p(0,T; H_0^1(\Omega))$, $\dfrac{\partial v}{\partial t} \in L^{p'}(0,T; H^{-1}(\Omega))$ and $v(x,T) = 0$, it holds

$$-\int_0^T \int_\Omega \left(\frac{\partial v}{\partial t} + div(\mathbf{a}v) \right) u d\Omega dt + \int_0^T \int_\Omega \epsilon \nabla u \cdot \nabla v d\Omega dt = \int_0^T \int_\Omega f(u,x,t) v d\Omega dt. \tag{4.4}$$

The application of the semi-Lagrangian approach to calculate at any instant t_n an approximation to the weak solution requires: To break the interval $\overline{I} := [0,T]$ into subintervals $I_n := (t_{n-1}, t_n]$, $n = 1, 2, ..., N$, with $t_0 = 0$, $t_N = T$ and the integration of the system

$$\begin{cases} \dfrac{dX(x,t_n;t)}{dt} = \mathbf{a}(X(x,t_n;t),t), \\ \\ X(x,t_n;t_n) = x, \end{cases} \tag{4.5}$$

for $x \in \Omega$ and $t \in \overline{I}_n$. $X(x,t_n;t)$ denotes the characteristics of the operator $\dfrac{D}{Dt}$ in the time subinterval $I_n := (t_{n-1}, t_n]$, in particular, $X(x,t_n;t_{n-1})$ are the feet of the characteristics at time t_{n-1}. By virtue of the assumptions on $\mathbf{a}(x,t)$, see (4.3), there exists a unique solution of (4.5) that can be expressed as

$$X(x,t_n;t) = x - \int_t^{t_n} \mathbf{a}(X(x,t_n;\tau),\tau) d\tau. \tag{4.6}$$

In fact, from the analytical theory of ODEs we have the following results

Lemma 1. *Assume that* $\mathbf{a} \in L^\infty\ (0,T;W^{k,\infty}(\mathbb{R}^2)^2),\ k \geq 1$. *Then for any* n, $1 \leq n \leq N$, *there exists a unique solution* $t \to X(x,t_n;t)$ $(t \in [t_{n-1},t_n] \subset [0,T])$ *of (4.6) such that* $X(x,t_n;t) \in W^{1,\infty}(0,T;W^{k,\infty}(\mathbb{R}^2)^2)$. *Furthermore, let the multi-index* $\alpha \in N^2$, *then for all* α, *such that* $1 \leq \mid \alpha \mid \leq k$, $\partial_x^{\boldsymbol{\alpha}} X_i(x,t_n,t) \in C^0([0,T];L^\infty(\mathbb{R}^2 \times [0,T]))$, $1 \leq i \leq 2$.

Lemma 2. *Suppose the assumptions of Lemma 1 hold. For* $|t_n - t|$ *sufficiently small,* $x \to X(x,t_n;t)$ *defines a quasi-isometric homeomorphism of* $\Omega(t_n) \subset \Omega^*$ *onto* $\Omega(t) \subset \Omega^*$ *with Jacobian determinant* $J(x,t_n;t) \in C^0([0,T];L^\infty(\Omega(t_n) \times [0,T]))$ *satisfying*

$$\exp(-C\,|t_n - t|) \leq J(x,t_n;t) \leq \exp(C\,|t_n - t|), \tag{4.7}$$

where $C = \|\nabla \cdot \mathbf{a}\|_{L^\infty(\mathbb{R}^2 \times (0,T))}$

Moreover,

$$K^{-1} \mid x - y \mid \leq \mid X(x,t_n;t) - X(y,t_n;t) \mid \leq K \mid x - y \mid, \tag{4.8}$$

where $K = \exp(|t_n - t|\,|\nabla \cdot \mathbf{a}|_{L^\infty(0;T,(W^{1,\infty}(\mathbb{R}^2))^2)}$. $|x-y|$ *denotes the Euclidean distance between the points* $x,\ y \in \mathbb{R}^2$.

Next, for each $t \in \overline{I}_n$ and $x \in \Omega^*$ we can define $u^* : \Omega^* \times \overline{I}_n \to \mathbb{R}$ as $u^* = Eu$, and $\overline{u} : \Omega \times \overline{I}_n$ as

$$\overline{u}(x,t) = u^*(\cdot,t) \circ X(\cdot,t_n;t)(x). \tag{4.9}$$

Introducing the arclength $s(x,t)$ for the characteristics $X(x,t_n;t)$, which is defined as

$$ds = dt\left(1 + \left(\frac{dX_1(x,t_n;t)}{dt}\right)^2 + \left(\frac{dX_2(x,t_n;t)}{dt}\right)^2\right)^{1/2} = dt\zeta(x,t), \tag{4.10a}$$

where by virtue of (4.5)

$$\zeta(x,t) = \sqrt{1 + |a(X(x,t_n;t)|^2}, \tag{4.10b}$$

we have that

$$s(x,t) = \int_{t_{n-1}}^{t} \zeta(x,\tau)d\tau, \tag{4.10c}$$

with $s(x,t_{n-1}) = 0$ and $s(x,t_n)$ being the arclength from the feet $(X(x,t_n;t_{n-1}),t_{n-1})$ to the points (x,t_n). Hence, it follows that for x fixed

$$\frac{D}{Dt} = \zeta(x,t)\frac{d}{ds}$$

and, therefore, we can recast (4.1) as

$$\begin{cases} \zeta(x,t)\dfrac{du(s)}{ds} = \epsilon\Delta u(s) + f(u(s),s) \quad \text{in } S_n, \\ \\ u(s=0) = \overline{u}(x,t_{n-1}) \ \text{ in } \Omega \text{ is a datum}, \\ \\ u|_{\partial\Omega} = 0 \quad t > 0. \end{cases} \tag{4.11}$$

We must note that (4.11) is formally a reaction-diffusion problem along the characteristics; therefore, its numerical treatment must be similar to the reaction-diffusion problems in (x,t)

coordinates of Chapter 3 with the exception of the initial condition $u(s=0) = \overline{u}(x, t_{n-1})$. However, another form of writing (4.11), more convenient for the construction of the adaptive algorithm based on the DWR methodology, makes use of (4.9) and is

$$\begin{cases} \dfrac{\partial \overline{u}}{\partial t} = \epsilon \Delta \overline{u} + f(\overline{u}, X(x, t_n; t), t) \quad \text{in } S_n, \\ \\ \overline{u}(x, t_{n-1}) \text{ in } \Omega \text{ is a datum,} \\ \\ u^n|_{\partial\Omega} = 0, \end{cases} \tag{4.12a}$$

noting that

$$\overline{u}(x,t)|_{t=t_n} = u(x, t_n), \;\; x \in \Omega. \tag{4.12b}$$

To solve numerically the problem (4.12a) we shall consider for each t_n the slab $S_n := \Omega \times I_n$, and for fixed integers $r \geq 1$ and s the trial and test spaces respectively

$$V_{h\Delta t}^{(r)} = \{\varphi_{h\Delta t} : \overline{\Omega} \times \overline{I} \to \mathbb{R} : \; \forall n \text{ and } (x,t) \in S_n, \; \varphi_{h\Delta t} \in C(\overline{I}_n; V_h^n), \; \varphi_{h\Delta t}(x,0) \in V_h^0 \text{ and } \; \varphi_{h\Delta t}(x,\cdot)|_{\overline{I}_n} \; \in \; P_r\}, \tag{4.13}$$

$$W_{h\Delta t}^{(s)} = \{\psi_{h\Delta t} : \overline{\Omega} \times \overline{I} \to \mathbb{R} : \; \forall n \text{ and } (x,t) \in S_n, \; \psi_{h\Delta t} \in L^p(I_n; V_h^n), \; \psi_{h\Delta t}(x,0) \in V_h^0 \text{ and } \; \psi_{h\Delta t}(x,\cdot)|_{I_n} \; \in \; P_s\}; \tag{4.14}$$

here, P_r and P_s are the set of polynomials of degrees at most r and s respectively defined on I_n, and

$$V_h^n = \left\{ v_h : C^0(\overline{\Omega}) : \, v_h|_K \in P_m(K) \; \forall K \in \mathbb{T}_h^n \right\}$$

where P_m is the set of polynomials of degrees at most m defined on respectively defined on $K \in \mathbb{T}_h^n$. Note that for $r = 0$ the space $V_{h\Delta t}^{(0)}$ coincides with $W_{h\Delta t}^{(0)}$.

The adaptive algorithm we describe in this chapter computes the numerical solution to (4.1) in the finite dimensional spaces $V_{h\Delta t}^{(1)} \times W_{h\Delta t}^{(0)}$. A numerical solution to problem (4.12a) is then a function

$$\overline{u}_{h\Delta t}(x,t) = \overline{u}_{h\Delta t}^{n-1}(x) + \frac{t - t_{n-1}}{t_n - t_{n-1}}(u_{h\Delta t}^n - \overline{u}_{h\Delta t}^{n-1}(x)), \tag{4.15a}$$

with

$$u_{h\Delta t}^n|_{\partial\Omega} = 0, \tag{4.15b}$$

and such that for all $\psi_{h\Delta t} \in W_{h\Delta t}^{(0)}$

$$(u_{h\Delta t}^n - \overline{u}_{h\Delta t}^{n-1}, \psi_{h\Delta t})_\Omega + \frac{\Delta t_n}{2} a(u_{h\Delta t}^n + \overline{u}_{h\Delta t}^{n-1}, \psi_{h\Delta t}) = \int_{I_n} \left(f(u_{h\Delta t}, X_{h\Delta t}^{n-1}(x), t), \psi_{h\Delta t} \right)_\Omega dt. \tag{4.15c}$$

Here, the following notations are used: for all $n, \; g(x,t_n) = g^n(x), \; X_{h\Delta t}^{n-1}(x)$ is an approximation to $X(x, t_n; t_{n+1})$ and

$$\begin{aligned} a(u,v) &= \epsilon \int_\Omega \nabla u \cdot \nabla v d\Omega, \quad u, v \in H^1(\Omega), \\ (u,v)_\Omega &= \int_\Omega u \cdot v d\Omega, \;\; u, v \in L^2(\Omega). \end{aligned} \tag{4.16}$$

Thus, to calculate the solution $u_{h\Delta t}^n(x)$ for each I_n we perform the following two stages:

1. *The semi-Lagrangian stage*. In this stage we calculate for each I_n the set of departure points $\{X^{n-1}(x_i)\}$, $x_i \in \Omega$, and then obtain $\overline{u}_{h\Delta t}^{n-1}(x)$ defined on the partition $\mathbb{T}_h^n$ via quasi-monotone interpolatory projection of order 2 from the solution $u_{h\Delta t}^{n-1}(x)$ defined on the partition $\mathbb{T}_h^{n-1}$.

2. *The parabolic stage*. Here, we calculate the solution $u_{h\Delta t}^n$ by solving (4.15c).

Next, we describe in detail each stage as well as the way we apply DWR methodology to calculate a posteriori error estimates

4.2 The semi-Lagrangian stage

Here, we propose a numerical procedure to calculate $\overline{u}_{h\Delta t}(x, t_{n-1})$ that is a discrete approximation in V_h^n of the function $u_{h\Delta t}(X(x, t_n; t_{n-1}), t_{n-1})$, where $u_{h\Delta t}(x, t_{n-1})$ is the numerical solution at the previous time instant t_{n-1}. Hereafter, we use $X^{n-1}(x)$ to denote the departure points $X(x, t_n; t_{n-1})$ of the trajectories.

The numerical solution $\overline{u}_{h\Delta t}(x, t_{n-1})$ belongs to the family of conforming finite element space V_h^n associated to the triangulation $\mathbb{T}_h^n$. To define V_h^n, we consider an element of reference $\widehat{K} \in \mathbb{R}^2$ such that for each element $K \in \mathbb{T}_h^n$ we define the one-to-one mapping $F_K : \widehat{K} \to K$.

$$F_K : \widehat{K} \longrightarrow K,\ x = B_K^n \widehat{x} + b_K^n, \quad B_K^n \in L(\mathbb{R}^2) \text{ and } b_K^n \in \mathbb{R}^2,$$

with the property that if $\widehat{P}_m(\widehat{K})$ is the set of polynomials $\widehat{p}(\widehat{x})$ of degree $\leq m$ defined on $\widehat{K}$, then for each K there exists the set

$$P_m(K) = \left\{p(x),\ x \in K : p(x) = \widehat{p}((F_K)^{-1}(x))\right\}$$

so that we can now define the family of conforming finite element as

$$V_h^n = \left\{v_h \in C^0(\overline{\Omega}) : v_h|_K \in P_m(K)\ \forall K \in \mathbb{T}_h^n\right\}.$$

Thus, if N_h^n is the number of mesh points, then any element of V_h^n can be expressed as

$$v_h = \sum_{i=1}^{N_h^n} V_i \varphi_i^n(x), \tag{4.17}$$

where $V_i = v_h(x_i),\ x_i$ being the $i-th$ mesh point of $\mathbb{T}_h^n$, and $\{\varphi_i^n\}$ being the set of global nodal basis functions of V_h^n characterized by the property $\varphi_i^n(x_j) = \delta_{ij}$.

The solution $u_{h\Delta t}^{n-1}(x) \in V_h^{n-1}$, however, $u_{h\Delta t}^{n-1}(X^{n-1}(x)) \in \overline{V}_h^n$, which is the finite element space associated to the triangulation $\overline{\mathbb{T}}_h^n$ composed by curved triangles $\overline{K}$. In Figure 4.1 we see a triangle $K \in \mathbb{T}_h^n$ transported backwards in time along of the characteristic curves $X^{n-1}(x)$, thus generating a new curved triangle $\overline{K} \in \overline{\mathbb{T}}_h^n$.

Hence, we can define the finite element space $\overline{V}_h^n$ associated to the partition $\overline{\mathbb{T}}_h^n := X^{n-1}(\mathbb{T}_h^n)$ as

$$\overline{V}_h^n = \left\{\overline{v}_h \in C^0(\mathbb{R}^2) : \overline{v}_h|_{\overline{K}} \in \overline{P}_m(\overline{K})\ \forall \overline{K} \in \overline{\mathbb{T}}_h^n\right\},$$

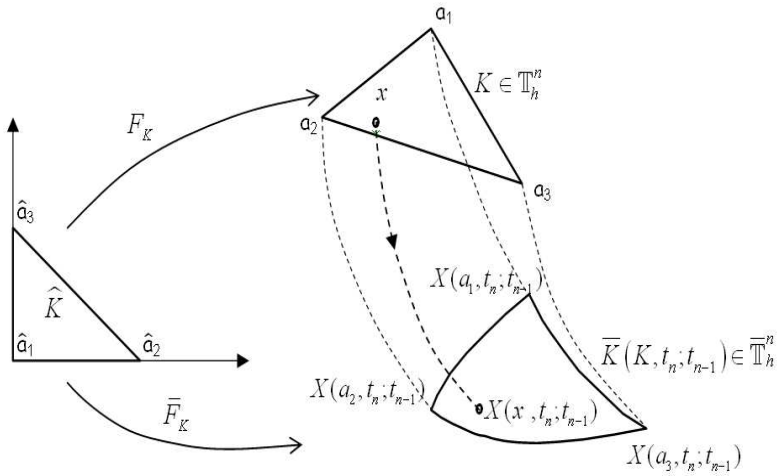

Figure 4.1: The mapping F_K from the reference element $\widehat{K}$ onto K. K is transported backwards in time along of the characteristic curves , generating a curved triangle $\overline{K}$. The mapping $\overline{F}_K$ from the reference element $\widehat{K}$ onto $\overline{K}$.

because for each finite element in the partition $\mathbb{T}_h^n$ there is one and only one curved finite element in $\overline{\mathbb{T}}_h^n$ by the mapping $\overline{F}_K : \widehat{K} \to \overline{K}$

$$\overline{F}_K : \widehat{K} \longrightarrow \overline{K},\ \overline{F}_K(\widehat{x}) = (X^{n-1} \circ F_K)(\widehat{x}),$$

satisfying the properties

$$\overline{P}_m(\overline{K}) = \left\{ p(y),\ y \in \overline{K} : p(y) = \widehat{p}\left(\overline{F}_K^{-1}(y)\right),\ \widehat{p} \in \widehat{P}_m(\widehat{K}) \right\}$$

with $\overline{P}_m(\overline{K}) \subset W^{m+1,\infty}(\overline{K})$.

4.2.1 Computation of $X_{\Delta t}(x, t_n; t_{n-1})$

A crucial step in the application of a semi-Lagrangian method is to approximate the points $X(x, t_n; t_{n-1})$ as accurate as possible. We solve the equation (4.5) by the embedded Runge-Kutta 2(3) with the following algorithm

$$\begin{cases} K_1 = \mathbf{a}(x, t_n), \\ K_2 = \mathbf{a}(x - \Delta t_n K_1, t_n - \Delta t_n), \\ K_3 = \mathbf{a}\left(x - \dfrac{\Delta t_n K_1}{4} - \dfrac{\Delta t_n K_2}{4}, t_n - \dfrac{\Delta t_n}{2}\right), \\ \\ X_{\Delta t}(x, t_n; t_{n-1}) = x - \Delta t_n \left(\frac{K_1}{2} + \frac{K_2}{2}\right). \\ \\ X^*_{\Delta t}(x, t_n; t_{n-1}) = x - \Delta t_n \left(\frac{K_1}{6} + \frac{K_2}{6} + \frac{4K_3}{6}\right). \end{cases} \tag{4.18}$$

noting that $X_{\Delta t}(x, t_n; t_{n-1})$ ($X_{\Delta t}^{n-1}(x)$) and $X^*_{\Delta t}(x, t_n; t_{n-1})$ ($X_{\Delta t}^{*n-1}(x)$) are second and third order approximations to $X(x, t_n; t_{n-1})$ ($X^{n-1}(x)$), respectively. $X_{\Delta t}^{*n-1}(x)$ will be an auxiliar solution that allow us to control the error in time.

Although in principle the velocity vector $\mathbf{a}(x,t)$ is a known datum in our problem, for the sake of generality and thinking of the Navier-Stokes equations, we shall consider that the velocity at time instants t_{n-1} and t_{n-2} is given by the functions $\mathbf{a}_h(x,t_{n-1}) \in V_h^{n-1}$ and $\mathbf{a}_h(x,t_{n-2}) \in V_h^{n-2}$ respectively. Thus, in order to know the velocity at intermediate time instants $t \in (t_{n-1},t_n]$ we shall use the second order extrapolation formula

$$\mathbf{a}_{h\Delta t}(x,t) = \left(1+\frac{t-t_{n-1}}{\Delta t_{n-1}}\right)\mathbf{a}_h(x,t_{n-1}) - \frac{t-t_{n-1}}{\Delta t_{n-1}}\mathbf{a}_h(x,t_{n-2}). \tag{4.19}$$

Moveover, we also approximate the velocity in space by interpolation, i.e. $\mathbf{a}_h(x,t) = I_h^n\mathbf{a}(x,t)$, where I_h^n is the interpolation operator defined on V_h^n as $\mathbf{a}_h(x_i,t) = I_h^n\mathbf{a}(x_i,t) = \mathbf{a}(x_i,t)$, x_i being the $i-th$ mesh point.

4.2.2 Calculation of $\overline{u}_{h\Delta t}^{n-1}(x)$

Here, we propose a numerical procedure to calculate $\overline{u}_{h\Delta t}(x,t_{n-1})$ as an approximation to $u_{h\Delta t}^{n-1}(X^{n-1}(x))$, with $u_{h\Delta t}^{n-1}(x) \in V_h^{n-1}$ and $u_{h\Delta t}^{n-1}(X^{n-1}(x)) \in \overline{V}_h^n$. Since $\overline{u}_{h\Delta t}^{n-1}(x)$ must belong to the finite element space V_h^n, it would be desirable that the feet of the characteristic curves $X^{n-1}(x)$ coincided with the nodes of the mesh $\mathbb{T}_h^{n-1}$. However, this is not generally true, and one has to devise a numerical procedure to calculate $\overline{u}_{h\Delta t}^{n-1}(x)$ as the projection of $u_{h\Delta t}^{n-1}(X^{n-1}(x))$ in V_h^n. A typical configuration can be seen in Figure 4.2

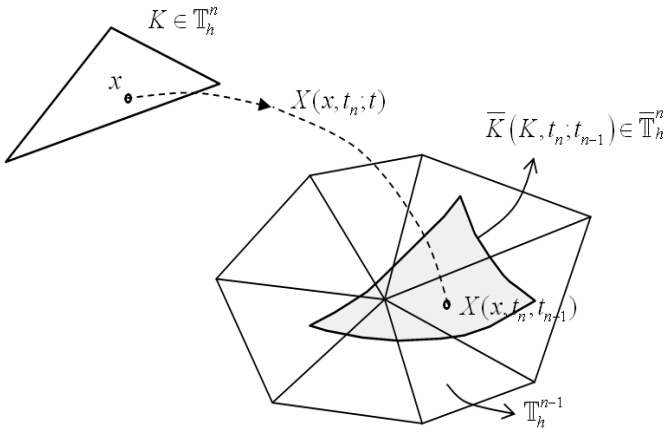

Figure 4.2: We have to transfer the information from the mesh $\mathbb{T}_h^{n-1}$ to the curved triangle $\overline{K}$ obtained by transporting backwards in time the triangle K belonging to $\mathbb{T}_h^n$.

An economical strategy from the point of view of CPU time is the quasi-monotone semi-Lagrangian interpolation of degree 2. With this procedure we only need to compute the departure points $X_{\Delta t}^{n-1}(x)$ for the grid-points $\{x_i\}$, because, when we introduce a new node in the mesh we calculate the foot of the characteristic for this point and then interpolate the solution at it. Thus, applying the finite element interpolation (4.17), to evaluate $u_{h\Delta t}^{n-1}(\cdot)$ at point $X_{\Delta t}^{n-1}(x)$ one obtains the following procedure:

1. Let $\{\varphi_i^n(x)\}$ and $\left\{\varphi_j^{n-1}(x)\right\}$ be the sets of global basis functions of V_h^n and V_h^{n-1}, respectively. Since $\overline{u}_{h\Delta t}^{n-1}(x) \in V_h^n$, then we set

$$\overline{u}_{h\Delta t}^{n-1}(x) = \sum_{i=1}^{N_h^n} \overline{U}_i^{n-1}\varphi_i^n(x), \tag{4.20a}$$

where N_h^n is the dimension of the set of mesh nodes $\{x_i\}$ of $\mathbb{T}_h^n$ and $\overline{U}_i^{n-1} = \overline{u}_{h\Delta t}^{n-1}(x_i) = u_{h\Delta t}^{n-1}(X_{\Delta t}^{n-1}(x_i))$. Moreover, given that $u_{h\Delta t}^{n-1}(x) \in V_h^{n-1}$, we calculate $\overline{U}_i^{n-1}$ as

$$\overline{U}_i^{n-1} = \sum_{j=1}^{N_h^{n-1}} U_j^{n-1} \varphi_j^{n-1}(X_{\Delta t}^{n-1}(x_i)). \tag{4.20b}$$

where N_h^{n-1} is the dimension of the set of mesh nodes $\{x_i\}$ of $\mathbb{T}_h^{n-1}$. In general, $N_h^{n-1} \neq N_h^n$ and $X_{\Delta t}^{n-1}(x_i) \notin \{x_j\}$; in fact, there is an element $K \in \mathbb{T}_h^{n-1}$ where $X_{\Delta t}^{n-1}(x_i)$ is contained. To find such an element we use the search-locate algorithm presented in [3].

2. It is known that Lagrange interpolation of degree ≥ 2 leads to a result that exhibits an oscillatory behavior and does not satisfy a discrete maximum principle. To overcome these problems we use the mesh independent limiting procedure (or monotony procedure) of [15], specifically designed for semi-Lagrangian schemes. Then,

$$\overline{U}_i^{n-1} = \begin{cases} U^{n-1+} & \text{if } \; U^{n-1+} > \overline{U}_i^{n-1}, \\ U^{n-1-} & \text{if } \; U^{n-1-} < \overline{U}_i^{n-1}, \\ \overline{U}_i^{n-1} & \text{as calculated in (4.20b) otherwise,}. \end{cases} \tag{4.20c}$$

where,

$$U^{n-1+} = \max_l \{U_l^{n-1}\}|_K \;\text{ and }\; U^{n-1-} = \min_l \{U_l^{n-1}\}|_K. \tag{4.20d}$$

Here, $K \in \mathbb{T}_h^{n-1}$ is the element that contains $X_{\Delta t}^{n-1}(x_i)$ and $1 \leq l \leq n_e$, n_e being the number of nodes of K. For triangles $n_e = (m+1)(m+2)/2$ and tetrahedra $n_e = (m+1)(m+2)(m+3)/6$, where m is the degree of the polynomials.

4.2.3 An a posteriori error estimate for the semi-Lagrangian stage

Here, we will define the a posteriori error in time and in space for the quasi-monotone interpolatory semi-Lagrangian scheme introduced above. To have a coherent representation of the error following the ideas of goal-oriented adaptation, we calculate the error in the functional $J(u_{h\Delta t}^{n-1}(X^{n-1}(x))$ as

$$e_{convect}^n = J(u_{h\Delta t}^{n-1}(X^{n-1}(x))) - J(\overline{u}_{h\Delta t}^{n-1}(x))$$

To have an error in each triangle $K \in \mathbb{T}_h^n$, the functional $J(u)$ has to be expressed as an integral over the whole domain, such that we can write $J(\cdot) = \int_\Omega j(\cdot) d\Omega$. In so doing, we can define a new mesh $\mathbb{T}_h^n$ where the initial condition $\overline{u}_{h\Delta t}^{n-1}(x)$ of the diffusion-reaction equation along the characteristic curves will be accurately reproduced in the whole domain Ω. To calculate the spatial integral that appears in the functional, we need to evaluate the feet of the characteristic curves at all the quadrature points $X^{n-1}(x_g)$ in each triangle $K \in \mathbb{T}_h^n$. But this operation has two problems: the first one is that we do not know the exact feet of the characteristic, the best approximation that we have for the exact feet of the characteristic curve is $X_{\Delta t}^{*n-1}$. The second one, is that obtaining $X^{n-1}(x_g)$ for each quadrature point (the number of quadrature points used in the numerical examples is $npg = 7$) inside of each triangle K of the mesh $\mathbb{T}_h^n$ has a high computational cost in terms of CPU time. Then, we propose the approximation

$X^{n-1}(x_g) \simeq y_g = \widetilde{X}^{*n-1}_{\Delta t}(x_g)$ to calculate the error. This means approximating the moved triangle backward in time $\overline{K}$ by $\widetilde{K}^*$, a straight triangle whose vertices are $X^{*n-1}_{\Delta t}(a_i)$, a_i being the 3 vertices of K, as we can see in Figure 4.3. With this idea in mind, we can obtain directly an approximate expression for $\widetilde{X}^{*n-1}_{\Delta t}(x_g)$, since we can define an affine mapping $\widetilde{F}^*_K$ from the reference element $\widehat{K}$ onto the straight triangle $\widetilde{K}^*$ as

$$\widetilde{F}^*_K : \widehat{K} \longrightarrow \widetilde{K}^*,\ y = \widetilde{B}^n_K \widehat{x} + \widetilde{b}^n_K, \quad \widetilde{B}^n_K \in L(\mathbb{R}^2) \text{ and } \widetilde{b}^n_K \in \mathbb{R}^2. \tag{4.21}$$

Notice that for the definition of the mapping $\widetilde{F}^*_K$, we only need to know the feet of the characteristic curves $X^{*n-1}_{\Delta t}(a_i)$ at the 3 vertices of K, that we have previously stored when we have used the quasi-monotone semi-Lagrangian interpolation to calculate $\overline{u}^{n-1}_{\Delta t}(x)$.

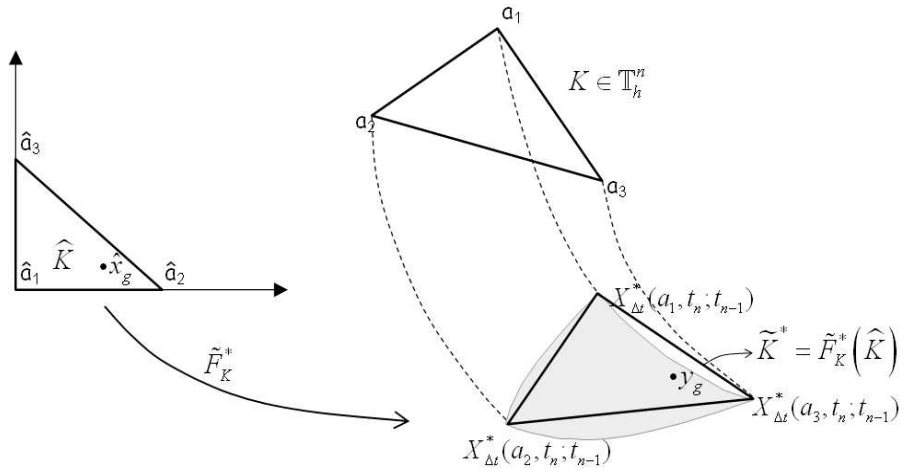

Figure 4.3: The mapping $\widetilde{F}^*_K$ from the reference element $\widehat{K}$ onto $\widetilde{K}^*$. $\widetilde{K}^*$ is the triangle whose vertices are the images of the vertices of K and have straight edges.

To sum up, the convective error can be approximated for the quasi-monotone interpolatory semi-Lagrangian method by

$$\widetilde{e}^n_{convect} = J(u^{n-1}_{h\Delta t}(y)) - J(\overline{u}^{n-1}_{h\Delta t}(x)).$$

Now, we are interested in dividing the error into spatial and temporal contributions; for this reason we introduce the term $J(\overline{u}^{*n-1}_{h\Delta t}(x))$ in the last expression

$$J(u^{n-1}_{h\Delta t}(y)) - J(\overline{u}^{n-1}_{h\Delta t}(x)) = J(u^{n-1}_{h\Delta t}(y)) - J(\overline{u}^{*n-1}_{h\Delta t}(x)) \ + \ J(\overline{u}^{*n-1}_{h\Delta t}(x)) - J(\overline{u}^{n-1}_{h\Delta t}(x)), \tag{4.22}$$

where $\overline{u}^{*n-1}_{h\Delta t}(x)$ is calculated in the same way as $\overline{u}^{n-1}_{h\Delta t}(x)$ (with equations (4.20a)-(4.20d)), but with the more accuracy feet of the characteristic curves $X^{*n-1}_{\Delta t}(x)$ instead of $X^{n-1}_{\Delta t}(x)$, both calculated with the embedded Runge-Kutta 2(3) scheme (4.18).

Therefore, we write the spatial and temporal errors in the following way:

$$\begin{cases} \widetilde{e}^n_{s,convect} = J(u^{n-1}_{h\Delta t}(y)) - J(\overline{u}^{*n-1}_{h\Delta t}(x)), \\ \\ \widetilde{e}^n_{t,convect} = J(\overline{u}^{*n-1}_{h\Delta t}(x)) - J(\overline{u}^{n-1}_{h\Delta t}(x)). \end{cases} \tag{4.23}$$

Now, to calculate the a posteriori error estimator, we have to divide the whole domain Ω into triangles $K \in \mathbb{T}^n_h$, because we wish to evaluate the error on the triangles belonging to the new mesh $\mathbb{T}^n_h$.

Spatial error:

We have already stated in equation (4.23) that the spatial error can be expressed as:

$$\begin{cases} \widetilde{e}^n_{s,convect} = J(u^{n-1}_{h\Delta t}(y)) - J(\overline{u}^{*n-1}_{h\Delta t}(x)) \\ \qquad\qquad = \sum\limits_{K\in\mathbb{T}^n_h} J(u^{n-1}_{h\Delta t}(y))_K - J(\overline{u}^{*n-1}_{h\Delta t}(x))_K. \end{cases} \tag{4.24}$$

The functional $J(u)$ must be written as an integral over the whole domain, such that we can write $J(\cdot) = \int_\Omega j(\cdot)d\Omega$, then we shall use a quadrature rule to evaluate these integrals at each triangle K

$$\begin{cases} \widetilde{e}^n_{sK,convect} = J(u^{n-1}_{h\Delta t}(y))_K - J(\overline{u}^{*n-1}_{h\Delta t}(x))_K = \\ \qquad\qquad = \int_K \left[j(u^{n-1}_{h\Delta t}(y)) - j(\overline{u}^{*n-1}_{h\Delta t}(x)) \right] d\Omega = \\ \qquad\qquad = \sum\limits_{g=1}^{npg} \left[j(u^{n-1}_{h\Delta t}(y_g))) - j(\overline{u}^{*n-1}_{h\Delta t}(x_g))\widehat{w}_g \cdot det(F_K) \right], \end{cases}$$

where $\widehat{w}_g$ is the weight associated to the quadrature point $\widehat{x}_g$, and $det(F_K)$ is the determinant of the mapping from the reference element $\widehat{K}$ onto the fixed element K and $y_g = \widetilde{X}^{*n-1}_{\Delta t}(x_g) = \widetilde{F}^*_K(\widehat{x}_g)$.

Temporal error:

Now, we have to measure the error committed in time. We have already stated in equation (4.23) that the temporal error can be expressed as:

$$\begin{cases} \widetilde{e}^n_{t,convect} := J(\overline{u}^{*n-1}_{h\Delta t}(x)) - J(\overline{u}^{n-1}_{h\Delta t}(x)) \\ \qquad\qquad = \sum\limits_{K\in\mathbb{T}^n_h} J(\overline{u}^{*n-1}_{h\Delta t}(x))_K - J(\overline{u}^{n-1}_{h\Delta t}(x))_K, \end{cases} \tag{4.25}$$

where $\overline{u}^{*n-1}_{h\Delta t}(x)$ and $\overline{u}^{n-1}_{h\Delta t}(x)$ are calculated following (4.20a) to (4.20d) at points $X^{*n-1}_{\Delta t}(x)$ and $X^{n-1}_{\Delta t}(x)$ respectively. Both $\overline{u}^{*n-1}_{h\Delta t}(x)$ and $\overline{u}^{n-1}_{h\Delta t}(x) \in V^n_h$, therefore, there is no problem in evaluating the expression of the error at each triangle using a quadrature rule.

$$\begin{cases} \widetilde{e}^n_{tK,convect} = J(\overline{u}^{*n-1}_{h\Delta t}(x))_K - J(\overline{u}^{n-1}_{h\Delta t}(x))_K \\ \qquad\qquad = \int_K \left[j(\overline{u}^{*n-1}_{h\Delta t}(x)) - j(\overline{u}^{n-1}_{h\Delta t}(x)) \right] d\Omega = \\ \qquad\qquad = \sum\limits_{g=1}^{npg} \left[j(\overline{u}^{*n-1}_{h\Delta t}(x_g)) - j(\overline{u}^{n-1}_{h\Delta t}(x_g))\widehat{w}_g \cdot det(F_K) \right], \end{cases}$$

where $\widehat{w}_g$ is the weight associated to the quadrature point $\widehat{x}_g$, and $det(F_K)$ is the determinant of the mapping from the reference element $\widehat{K}$ onto the fixed element K.

4.3 Diffusion-reaction stage

To apply the DWR methodology to control the local error of the numerical solution in each I_n, we follows the approach of Chapter 3 and consider (4.12a) with a perturbed initial condition

given by the semi-Lagrangian stage. Thus, we have in $S_n = \Omega \times I_n$

$$\begin{cases} \dfrac{\partial \overline{U}}{\partial t} = \epsilon \Delta \overline{U} + f(\overline{U}, X(x, t_n; t), t) \quad \text{in } S_n, \\ \overline{U}(t_{n-1}) = \overline{u}_{h\Delta t}^{n-1}(x) \text{ in } \Omega, \\ U^n|_{\partial\Omega} = 0. \end{cases} \tag{4.26}$$

For each I_n we define the semi-linear form $A : V^n \times W^n \to \mathbb{R}$ as

$$A(\overline{U})(z) = \int_{I_n} \left\{ \left\langle \partial_t \overline{U}, z \right\rangle_\Omega + a(\overline{U}, z) - \left\langle f(\overline{U}, X(x, t_n; t), t), z \right\rangle_\Omega \right\} dt + ([\overline{U}]^{n-1}, z^{n-1+})_\Omega, \tag{4.27}$$

where, $[\overline{U}]^{n-1} = \overline{U}^{n-1} - \overline{u}_{h\Delta t}^{n-1}$ is the jump of the solution at time t_{n-1}, and V^n and W^n are the local restrictions to I_n of the spaces $L^2(I_n; H_0^1(\Omega)) \cap L^p(S_n) \cap C^0(\overline{I}_n; L^2(\Omega))$ and $L^p(I_n; H_0^1(\Omega))$, respectively.

Next, we choose an output functional $J : V^n \to \mathbb{R}$, such that we can define the Lagrangian $\mathcal{L} : V^n \times W^n \to \mathbb{R}$ as

$$\mathcal{L}(\overline{U}; z) := J(\overline{U}) - A(\overline{U})(z).$$

Then, we calculate the stationary points $(\overline{U}, z) \in V^n \times W^n$ of $\mathcal{L}(\overline{U}; z)$ which are the solution of

$$\mathcal{L}'(\overline{U}; z)(\varphi, \psi) = 0;$$

that is, we have to find the pair $(\overline{U}, z) \in V^n \times W^n$ that satisfies

$$-\int_{I_n} \left\{ \left\langle \partial_t \overline{U}, \psi \right\rangle_\Omega + a(\overline{U}, \psi) - \left\langle f(\overline{U}, X(x, t_n; t), t), \psi \right\rangle_\Omega \right\} dt - ([\overline{U}]^{n-1}, \psi^{n-1+})_\Omega = 0 \ \forall \psi \in W^n \tag{4.28}$$

and

$$J'(\overline{U})(\varphi) - \int_{I_n} \left\{ - \langle \partial_t z, \varphi \rangle_\Omega + a(z, \varphi) - \left\langle f'(\overline{U}, X(x, t_n; t), t)\varphi, z \right\rangle_\Omega \right\} dt - (\varphi^{n-}, z^{n-})_\Omega = 0, \ \forall \varphi \in V^n. \tag{4.29}$$

Problems (4.28) and (4.29) are termed *primal* and *dual problems,* respectively, in the slab S_n. Note that for the primal problem we have that $\overline{U}^{n-1-} = \overline{u}_{h\Delta t}^{n-1}$, so, this problem is the weak formulation of the auxiliary problem (4.26).

The Galerkin approximation $(\overline{u}_{h\Delta t}, z_{h\Delta t}) \in V_{h\Delta t}^{(r)} \times W_{h\Delta t}^{(s)}$ to the primal and dual problems by time-space finite elements in each slab S_n satisfies for all $(\varphi_{h\Delta t}, \psi_{h\Delta t}) \in V_{h\Delta t}^{(r)} \times W_{h\Delta t}^{(s)}$ the equation

$$\mathcal{L}'(\overline{u}_{h\Delta t}; z_{h\Delta t})(\varphi_{h\Delta t}, \psi_{h\Delta t}) = 0;$$

that is, $(\overline{u}_{h\Delta t}, z_{h\Delta t})$ is the unique solution of the following problem:

For each I_n, find the pair $(\overline{u}_{h\Delta t}, z_{h\Delta t}) \in V_{h\Delta t}^{(r)} \times W_{h\Delta t}^{(s)}$ such that for all $(\varphi_{h\Delta t}, \psi_{h\Delta t}) \in V_{h\Delta t}^{(r)} \times W_{h\Delta t}^{(s)}$

$$-\int_{I_n} \{(\partial_t \overline{u}_{h\Delta t}, \psi_{h\Delta t})_\Omega + a(\overline{u}_{h\Delta t}, \psi_{h\Delta t}) - (f(\overline{u}_{h\Delta t}), \psi_{h\Delta t})_\Omega\} \, dt - ([\overline{u}_{h\Delta t}]^{n-1}, \psi_{h\Delta t}^{n-1+})_\Omega = 0, \tag{4.30}$$

and

$$J'(\overline{u}_{h\Delta t})(\varphi_{h\Delta t}) - \int_{I_n} \{(-\partial_t z_{h\Delta t}, \varphi_{h\Delta t})_\Omega + a(z_{h\Delta t}, \varphi_{h\Delta t}) - (f'(\overline{u}_{h\Delta t})\varphi_{h\Delta t}, z_{h\Delta t})_\Omega\}\, dt - (\varphi^{n-}_{h\Delta t}, z^{n-}_{h\Delta t})_\Omega = 0. \tag{4.31}$$

From (4.30) and (4.31) together with Proposition 1 of Chapter 3 is easy to obtain the following result:

Proposition 3. *For each I_n, let $(\overline{U}, z)$ and $(\overline{u}_{h\Delta t}, z_{h\Delta t})$ be solutions of ((4.28), (4.29)) and ((4.30), (4.31)) respectively. Assume that the functional $J : V^n \to \mathbb{R}$ and the semi-linear form $A : V^n \times W^n \to \mathbb{R}$ have directional derivatives up to order three. Then, we have the following error representation*

$$\begin{cases} e^n_{diff-react} = J(\overline{U}) - J(\overline{u}_{h\Delta t}). \\ \\ J(\overline{U}) - J(\overline{u}_{h\Delta t}) \approx \frac{1}{2}\rho(\overline{u}_{h\Delta t})(z - \psi_{h\Delta t}) + \frac{1}{2}\rho^*(\overline{u}_{h\Delta t}, z_{h\Delta t})(\overline{U} - \varphi_{h\Delta t}), \end{cases}$$

with $\rho(\overline{u}_{h\Delta t})(.)$ and $\rho^*(\overline{u}_{h\Delta t}, z_{h\Delta t})(.)$ primal and dual residuals respectively. The idea is that the functional $J(\cdot)$ will be equal for convection and diffusion-reaction parts.

Primal residual:

$$\begin{cases} \rho(\overline{u}_{h\Delta t})(\cdot) = \sum\limits_{K\in\mathbb{T}^n_h} \int_{I_n} \left\{(\overline{R}_{h\Delta t}, \cdot)_K + (\overline{r}_{h\Delta t}, \cdot)_{\partial K}\right\} dt - ([\overline{u}_{h\Delta t}]^{n-1}, (\cdot)^{n-1+})_K, \\ \\ \overline{R}_{h\Delta t} = \overline{f}(\overline{u}_{h\Delta t}) - \partial_t \overline{u}_{h\Delta t} + \epsilon\Delta\overline{u}_{h\Delta t} \quad \text{and} \quad \overline{r}_{h\Delta t} = \begin{cases} \frac{\epsilon}{2}\left[\partial_n \overline{u}_{h\Delta t}\right]_\Gamma \ \text{ if } \ \Gamma \subset \partial K \backslash \partial\Omega, \\ 0 \ \text{ if } \ \Gamma \subset \partial\Omega. \end{cases} \end{cases}$$

Dual residual:

$$\begin{cases} \rho^*(\overline{u}_{h\Delta t}, z_{h\Delta t})(\cdot) = \sum\limits_{K\in\mathbb{T}^n_h} \int_{I_n} \left\{(\overline{R}^*_{h\Delta t}, \cdot)_K + (\overline{r}^*_{h\Delta t}, \cdot)_{\partial K}\right\} dt - J'(\overline{u}_{h\Delta t})(\cdot)_K - (z^n_{h\Delta t}, (\cdot)^{n-})_K, \\ \\ R^*_{h\Delta t} = \overline{f}'(\overline{u}_{h\Delta t}) z_{h\Delta t} + \partial_t z_{h\Delta t} + \epsilon\Delta z_{h\Delta t} \quad \text{and} \quad r^*_{h\Delta t} = \begin{cases} \frac{\epsilon}{2}\left[\partial_n z_{h\Delta t}\right]_\Gamma \ \text{ if } \ \Gamma \subset \partial K \backslash \partial\Omega, \\ 0 \ \text{ if } \ \Gamma \subset \partial\Omega. \end{cases} \end{cases}$$

Furthermore, in order to separate the error in time and in space, we consider Proposition 2 and the post-processing procedure defined also in Chapter 3.

Error in Space:

Using equation (3.23c) in Proposition 2 we can write

$$e^n_{s,diff-react} = \frac{1}{2}\rho(\overline{u}_{h\Delta t})(z - z_h) + \frac{1}{2}\rho^*(\overline{u}_{h\Delta t}, z_{h\Delta t})(\overline{U} - \overline{u}_h)$$

and with equations (3.25) of the temporal post-processing procedure, we can calculate an approximation for the weights

$$\begin{cases} z - z_h \approx \left(I^{2m}_{2h} z_{h\Delta t} - z_{h\Delta t}\right), \\ \\ \overline{U} - \overline{u}_h \approx \left(I^{2m}_{2h} \overline{u}_{h\Delta t} - \overline{u}_{h\Delta t}\right). \end{cases}$$

Therefore, the restriction to the element $K \in \mathbb{T}^n_h$ of the time error estimator for the diffusion-reaction problem along the characteristic curves takes the following expression

$$\begin{cases} \widetilde{e}^n_{sK,diff-react} = \frac{1}{2}\rho(\overline{u}_{h\Delta t})(I^4_{2h} z_{h\Delta t} - z_{h\Delta t}) + \frac{1}{2}\rho^*(\overline{u}_{h\Delta t}, z_{h\Delta t})(I^4_{2h}\overline{u}_{h\Delta t} - \overline{u}_{h\Delta t})_K, \\ \\ \widetilde{e}^n_{s,diff-react} = \sum\limits_{K\in\mathbb{T}^n_h} \widetilde{e}^n_{sK,diff-react}. \end{cases}$$

Error in Time:

Using equation (3.23b) in Proposition 2 we can write

$$e^n_{t,diff-react} = \frac{1}{2}\rho(\overline{u}_{h\Delta t})(z_h - z_{h\Delta t}) + \frac{1}{2}\rho^*(\overline{u}_{h\Delta t}, z_{h\Delta t})(\overline{u}_h - \overline{u}_{h\Delta t})$$

and with equation (3.29b) of the temporal post-processing procedure, we can calculate the next approximation for the weights

$$\begin{cases} z_h - z_{h\Delta t} \approx \left(\widetilde{I}^{(s+1)}_{\Delta t} z_{h\Delta t} - z_{h\Delta t}\right), \\ \\ \overline{u}_h - \overline{u}_{h\Delta t} \approx \left(I^{(r+1)}_{2\Delta t}\overline{u}_{h\Delta t} - \overline{u}_{h\Delta t}\right). \end{cases}$$

Therefore, the restriction to the element $K \in \mathbb{T}^n_h$ of the time error estimator for the diffusion-reaction problem along the characteristic curves takes the following expression

$$\begin{cases} \widetilde{e}^n_{tK,diff-react} = \frac{1}{2}\rho(\overline{u}_{h\Delta t})(\widetilde{I}^1_{\Delta t} z_{h\Delta t} - z_{h\Delta t}) + \frac{1}{2}\rho^*(\overline{u}_{h\Delta t}, z_{h\Delta t})(I^2_{2\Delta t}\overline{u}_{h\Delta t} - \overline{u}_{h\Delta t})_K, \\ \\ \widetilde{e}^n_{t,diff-react} = \sum\limits_{K\in\mathbb{T}^n_h} \widetilde{e}^n_{tK,diff-react}. \end{cases}$$

Remark:

In the case of linear problems, both primal and dual residual coincide, i.e. $\rho(\overline{u}_{h\Delta t})(z - z_{h\Delta t}) = \rho^*(\overline{u}_{h\Delta t}, z_{h\Delta t})(\overline{U} - u_{h\Delta t})$, but this is no longer true in the nonlinear case. In general we can write by proposition 6.3 of [11] the following results:

$$\rho^*(\overline{u}_{h\Delta t}, z_{h\Delta t})(\overline{U} - \varphi_{h\Delta t}) = \rho(\overline{u}_{h\Delta t})(z - \psi_{h\Delta t}) + \Delta\rho$$

for any $\varphi_{h\Delta t}$, $\psi_{h\Delta t} \in V^{(r)}_{h\Delta t}$ and $W^{(s)}_{h\Delta t}$ respectively, with

$$\Delta\rho := \int_0^1 \{A''(\overline{u}_{h\Delta t} + se)(e, e, z_{h\Delta t} + se^*) - J''(\overline{u}_{h\Delta t} + se)(e, e)\}ds.$$

Then, we could only consider the primal residual in the calculation of the error in the functional $J(\overline{U})$. In this case we obtain a quadratic remainder term (4.32) instead of the cubic remainder term obtained when we consider both primal and dual residuals by Proposition 1.

$$\begin{cases} J(\overline{U}) - J(\overline{u}_{h\Delta t}) = \rho(\overline{u}_{h\Delta t})(z - \varphi_{h\Delta t}) + \mathcal{R}^{(2)}_{h\Delta t}, \\ \\ \mathcal{R}^{(2)}_{h\Delta t} := \int_0^1 \{A''(\overline{u}_{h\Delta t} + se)(e, e, z) - J''(\overline{u}_{h\Delta t} + se)(e, e)\}sds. \end{cases} \tag{4.32}$$

We also saw in the numerical examples of Chapter 3 that the dual and primal residual take similar values if the degree of nonlinearity is not large enough. This property introduces a big simplification in our problem, since for the evaluation of $I^2_{2\Delta t}\overline{u}_{h\Delta t}$ we will need to calculate $\overline{u}^{n-2}_{h\Delta t}(x)$, and that means to compute the feet of the characteristic $X(x, t_n; t_{n-2})$ and then apply the semi-Lagrangian interpolatory procedure as we employ for $\overline{u}^{n-1}_{h\Delta t}(x)$. The semi-Lagrangian step for treating the convective terms in a general convection-diffusion-reaction equation have a negligible CPU time cost $(5 - 10\%)$ compared with the resolution of the nonlinear primal problem, but to consider the dual residual increases the complexity of the code as well as the memory requirements. On the other hand, considering only the primal residual introduces a great simplification in the code; moreover, in our experience with numerical examples, we have observed no relevant differences in the mesh and time step size using this property.

4.4 Error indicators and adaptive procedure

We present in this section a fully adaptive algorithm to solve convection-diffusion-reaction time-dependent problems, so that, the temporal time step length Δt_n and the spatial triangulation $\mathbb{T}_h^n$ are adapted as time of integration progresses. To design a practical adaptive algorithm is customary to define an error indicator η^n, derived from the a posteriori error estimator e^n in each time subinterval I_n, for the convection and diffusion-reaction parts.

1. $\eta^n_{convect}$: The error indicator at time t_n defined from $e^n_{convect}$. It is the error committed in the approximation of the solution at the feet of the characteristic curves $u^{n-1}_{h\Delta t}(X^{n-1}(x))$ by $\overline{u}^{n-1}_{h\Delta t}(x)$ via the semi-Lagrangian scheme to treat the convection term. Note that if we have no convection term, i.e. the velocity field is $\mathbf{a}(x,t)=0$, the convective error in time will be zero, and the spatial error will be the error that we commit when transferring the information of the initial condition from $\mathbb{T}_h^{n-1}$ to $\mathbb{T}_h^n$.

2. $\eta^n_{diff-react}$: The error indicator at time t_n defined from $e^n_{diff-react}$. It is the error committed when we solve the diffusion-reaction equation along of the characteristic curves in $I_n := (t_{n-1}, t_n]$. This error has been obtained by the application of the local DWR technique.

With $p = s$ or t for spatial and temporal errors respectively, we define the convection $\eta^n_{convect}$ and diffusion-reaction $\eta^n_{diff-react}$ errors as the absolute value of the relative error of the a posteriori error estimator at each element $K \in \mathbb{T}_h^n$.

$$\eta^n_{pK,convect} = \frac{\left|\widetilde{e}^n_{pK,convect}\right|}{\left|J(\overline{u}^{n-1}_{h\Delta t})\right|} \text{ and } \eta^n_{pK,diff-react} = \frac{\left|\widetilde{e}^n_{pK,diff-react}\right|}{|J(u^n_{h\Delta t})|}.$$

Therefore, the error indicators in space and in time take the following expressions at time t_n

$$\begin{cases} \eta^n_s = \displaystyle\sum_{K\in\mathbb{T}_h^n} \eta^n_{sK,convect} + \sum_{K\in\mathbb{T}_h^n} \eta^n_{sK,diff-react}, \\ \\ \eta^n_t = \displaystyle\sum_{K\in\mathbb{T}_h^n} \eta^n_{tK,convect} + \sum_{K\in\mathbb{T}_h^n} \eta^n_{tK,diff-react}. \end{cases}$$

Prescribing a tolerance $TOL = Tol_s + Tol_t$, where Tol_s and Tol_t are the space and time tolerances respectively, the adaptive algorithm will accept the numerical solution $u^n_{h\Delta t}$ at time level t_n if the following criteria is satisfied

$$\eta^n_s \leq Tol_s \quad \text{and} \quad \eta^n_t \leq Tol_t.$$

To balance the space and time errors one chooses $Tol_s \approx Tol_t$.

4.4.1 Mesh adaptation: mesh-optimization strategy

The criterium to adapt the spatial mesh consists of calculating such a mesh with the minimum number of elements NE to satisfy $\eta^n_s \leq Tol_s$. This yields the optimal size h_K^{opt} of the element $K \in \mathbb{T}_h^n$ as we have seen in Chapter 2

$$h_K^{opt} = h_K \left(\frac{Tol_s}{W}\right) \left(\eta^n_K\right)^{-1/(\alpha+d)}, \tag{4.33}$$

where $W := \sum_{K \in \mathbb{T}_h^n} (\eta_K^n)^{\frac{d}{d+\alpha}} < \infty$; $\eta_K^n = \eta_s^n|_K$ is the spatial error indicator for the element K; α is the convergence rate of the spatial error. Thus, we adopt the following refining and coarsening criteria:

Refining criterion:
If $\dfrac{h_K^{opt}}{h_K} \leq 1$, mark the element K to refine n_r times,

$$n_r = Integer\ part \left[0.5 + 2 \frac{\log(h_K / h_K^{opt})}{\log 2} \right]. \tag{4.34a}$$

Coarsening criterion:
If $\dfrac{h_K^{opt}}{h_K} > 1$, mark the element K to coarsen n_c times,

$$n_c = Integer\ part \left[2 \frac{\log(h_K^{opt} / h_K)}{\log 2} \right]. \tag{4.34b}$$

4.4.2 Adaptation of the time step size

The error indicator η_t^n is used to adjust the size of the new time step as well as to ascertain whether the solution $u_{h\Delta t}^n$ is accepted or not. Following the strategy of the numerical ODE community, we adjust the time step by the formula as we have seen in Chapter 2

$$\begin{aligned} \Delta t_{new} &= \min(fac_{\max}, \max(fac_{\min}, fac)) \Delta t_{old}, \\ fac &= \begin{cases} \left(\left(\dfrac{\eta_t^{n-1}}{\eta_t^n} \right)^{1/\beta} \dfrac{\Delta t_n}{\Delta t_{n-1}} \right) \left(\dfrac{\kappa \cdot Tol_t}{\eta_t^n} \right)^{1/\beta} & \text{when } u_{h\Delta t}^n \text{ is accepted,} \\ \left(\dfrac{\kappa \cdot Tol_t}{\eta_t^n} \right)^{1/\beta} & \text{when } u_{h\Delta t}^n \text{ is rejected,} \end{cases} \end{aligned} \tag{4.35}$$

where β is the order of the time local truncation error; $fac_{\max}$ and $fac_{\min}$ are factors limiting the maximum and minimum step sizes respectively, usually $fac_{\max} = 5$ and $fac_{\min} = 0.2$; and κ is a security factor $\kappa = 0.7$.

4.4.3 Numerical Algorithm

The adaptive numerical algorithm is a generalization of the algorithm shown in Chapter 3 (3.35) and it has been tested in several conditions: pure convection equations; pure reaction-diffusion equations; convective-diffusion-reaction problem where the character of the equations change over time. We have seen the good behaviour of the algorithm in all of them, having to recalculate the solution $u_{h\Delta t}^n$ in just a few time steps either by choosing a new mesh or changing Δt_n.

A schematic presentation of the numerical algorithm to solve any convection-diffusion-reaction equation is the following:

Space – time adaptive algorithm :

Given the tolerances Tol_s, Tol_t, the macro triangulation $\mathbb{T}_{h,0}$ and the time step Δt_0 :
Obtain an initial mesh $\mathbb{T}_h^0$ by local refinement and project the initial datum u^0 in that mesh as $u_{h\Delta t}^0 \in V_h^0$.
Refine the initial mesh until $\eta_{s,convect}^0 < Tol_s$.
$n = 1$.
$\theta = 0.8$.
while $(t_{n-1} < T)$
{
$\mathbb{T}_{h,0}^n = \mathbb{T}_h^{n-1}$.
$i = 1$.
while $(k == 0)$
{

1. Adapt the mesh $\mathbb{T}_{h,i-1}^n$ to generate a new mesh $\mathbb{T}_{h,i}^n$.

2. Solve the convective part with the semi-Lagrangian scheme and obtain $\overline{u}_{h\Delta t}^{n-1}$.
   Calculate both initial errors $\eta_{s,convect}^n$ and $\eta_{t,convect}^n$.
   Use successive refinements to obtain a new mesh $\mathbb{T}_{h,i}^n$with $\eta_{s,convect}^n < \theta \cdot Tol_s$.

3. Solve the diffusion-reaction equation and obtain the solution $u_{h\Delta t,i}^n$ on the mesh $\mathbb{T}_{h,i}^n$.
   Calculate the error indicator with DWR technique in time $\eta_{t,diff-react}^n$ and in space $\eta_{s,diff-react}^n$.

4. Calculate the total error indicator: $\eta_t^n = \eta_{t,convect}^n + \eta_{t,diff-react}^n \quad \eta_s^n = \eta_{s,convect}^n + \eta_{s,diff-react}^n$
   and $\theta = \max\left(0.2, 0.8\dfrac{\eta_{s,convect}^n}{\eta_s^n}\right)$.

5. Select elements to be refined or coarsened with formulae (4.34a) and (4.34b).

If $[(\eta_s^n > Tol_s)\ and\ (\eta_t^n \leq Tol_t)]$
Reject step and maintain the time step length Δt_n. $k = 0$ and $i = i + 1$.
Else
Calculate a new time step Δt_{new} with formula (4.35).
if $(\eta_t^n > Tol_t)$
$t_n = t_{n-1} + \Delta t_{new}$. Reject step: $k = 0$ and $i = i + 1$.
else
$t_{n+1} = t_n + \Delta t_{new}$ and $k = 1$.
end if
End if
}
$u_{h\Delta t}^n = u_{h\Delta t,i}^n$.
$\mathbb{T}_h^n = \mathbb{T}_{h,i}^n$.
}

Note that θ is a parameter between 0.2 and 0.8 defined as $\theta = \max\left(0.2, 0.8\dfrac{\eta_{s,convect}^n}{\eta_s^n}\right)$. We use this parameter to refine the spatial mesh in the semi-Lagrangian stage until the criterion $\eta_{s,convect}^n < \theta \cdot Tol_s$ is satisfied. With this idea, the index i in the internal loop of the algorithm, takes the value 1 on most occasions, therefore, the number of time steps rejected is negligible.

4.5 Numerical tests

In this section we present some numerical tests to illustrate the performance of the previous goal orientated adaptive algorithm for convection-diffusion-reaction problems. The main features of these examples are the existence of moving regions of strong gradients, where an adaptive algorithm in time and in space is necessary to calculate an accurate solution. Next we show three different examples, namely, a pure convection problem, a convection-diffusion problem and a convection-diffusion-reaction problem.

- For the resolution of the convective part, we use quadratic positive interpolation and a Runge-Kutta 2(3) to find the feet of the characteristic curves, as we have described before.
- For the diffusion-reaction part of the equations, we consider quadratic finite elements in space and linear functions in time for the solution $\overline{u}_{h\Delta t}(x,t)$ (cG(1)-dG(0) scheme). We calculate the solution of nonlinear systems with a classical Newton algorithm, with the possibility of damping for those situations where the stiffness of the systems could require it. We solve linear systems with a diagonal preconditioned BICGSTAB for the nonsymmetrical systems.

4.5.1 Example 1. A pure convection problem

The first example consists of simulating the time evolution of an initial condition in a fixed rotating velocity field. The initial value problem is:

$$\begin{cases} \dfrac{Du}{Dt} := \dfrac{\partial u}{\partial t} + \mathbf{a}(x,t)\cdot\nabla u = 0 \quad \text{in } \Omega\times(0,T], \\ \\ u(x,0) = u^0(x), \end{cases} \tag{4.36}$$

where the initial condition $u^0(x)$ is defined by the slotted cylinder of radius $1/4$ and height 1, centered at $(0.5,0)$. The width of the slot is 0.1 and its depth is 0.35. The velocity field $\mathbf{a} = 2\pi(-x_2, x_1)$, so that, the period of the rotating motion is one.

Note that the exact solution of this problem is the initial condition transported by the velocity field, along the characteristic curves of the total derivative operator, which are circumferences

$$\begin{cases} x_1 = x_1^0\cos(2\pi t) - x_2^0\sin(2\pi t), \\ \\ x_2 = x_1^0\sin(2\pi t) + x_2^0\cos(2\pi t). \end{cases}$$

This example is a pure convection problem and its numerical solution is obtained applying the quasi-monotone interpolatory semi-Lagrangian scheme described in section 4.2.

$$u^n_{h\Delta t}(x) = \overline{u}^{n-1}_{h\Delta t}(x).$$

We will consider a circular domain Ω of radius 1 centered at the origin, and $T = 15$ (it means 15 revolutions) for the numerical integration of equation (4.36), with a tolerance $Tol_s = Tol_t = 7.5\cdot 10^{-4}$. We will control the mesh and the time step size using the following functionals:

$$J_1(u^{n-1}_{h\Delta t}(X)) = \int_\Omega u^{n-1}_{h\Delta t}(X)d\Omega \quad \text{and} \quad J_2(u^{n-1}_{h\Delta t}(X)) = \int_\Omega \left(u^{n-1}_{h\Delta t}(X)\right)^2 d\Omega. \tag{4.37}$$

With this example we will see the good behaviour of the error calculated in the way shown in Section 4.2 of this chapter. The number of rejected steps has been 1 at the beginning of the integration, and the number of time steps to complete the 15 revolutions has been 1008 (with a CPU_{time} of 362 $seg.$) and 1017 (with a CPU_{time} of 353 $seg.$) for the first and second functionals, respectively.

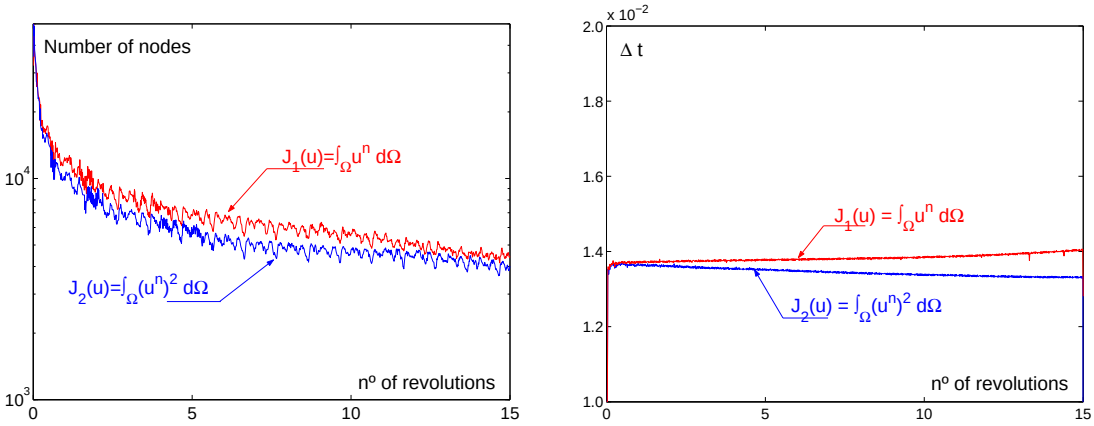

Figure 4.4: History of nodes (left) and time step size (right) for both functionals plotted against revolutions of the cylinder.

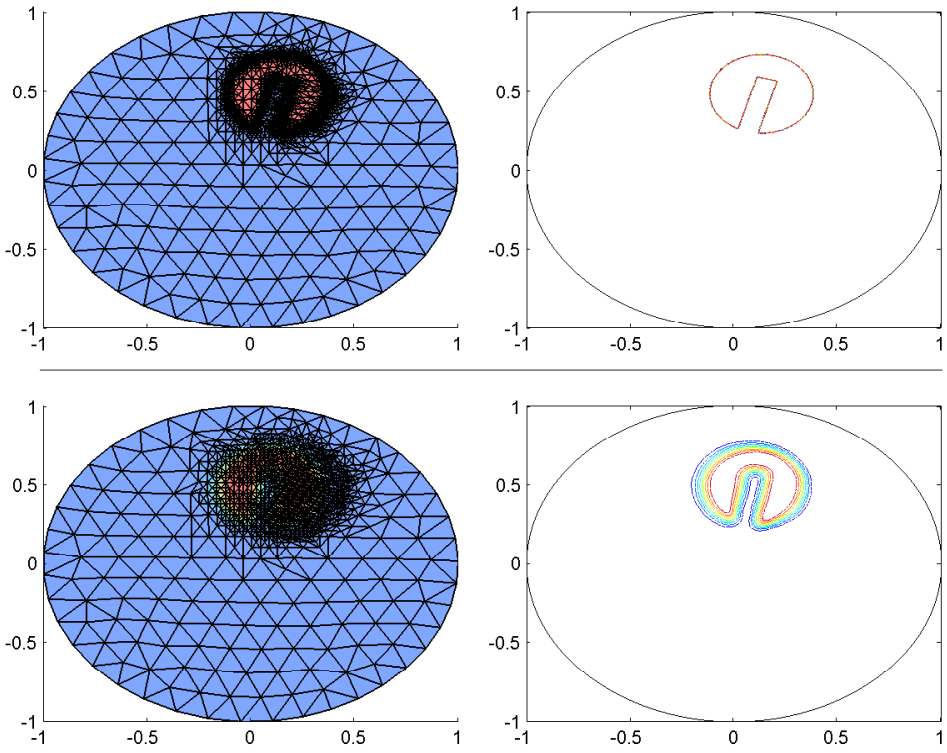

Figure 4.5: On the left, mesh in the first (upper panel) and in the last (bottom panel) revolution for the functional J_1. On the right, isolines of the solution from $u = 0.1$ to $u = 0.9$ at 0.1 intervals.

The evolution of the number of nodes and the time step size for both functionals is shown

in Figure 4.4. In this figure we see that the time step size is practically constant, as we could predict, but the number of elements decreases with time.

In Figure 4.5 we show on the top the mesh at a given instant during the first revolution with 10959 elements and 21960 nodes; and at the bottom, the mesh during the last revolution at the same position with 2213 elements and 4468 nodes. On the right side we have plotted the isolines of the solution from 0.1 to 0.9 at intervals 0.1, in order to better notice the effect of the numerical diffusion introduced by the semi-Lagrangian scheme. At the beginning of the integration, we need to reproduce the discontinuity of the solution to avoid a fast increase in the error, but when the numerical diffusion makes the solution smoother, the number of elements required to obtain the same error is smaller.

To see the performance of the error estimator in the convection stage of the equation, we calculate the effectivity index in each time interval I_n and plot it in Figure 4.6 for the first and second functionals.

$$I_{eff} := \frac{|\widetilde{e}_{s,convect} + \widetilde{e}_{t,convect}|}{\left|J(u_{h\Delta t}^{n-1}(X)) - J(\overline{u}_{h\Delta t}^{n-1}(x))\right|}.$$

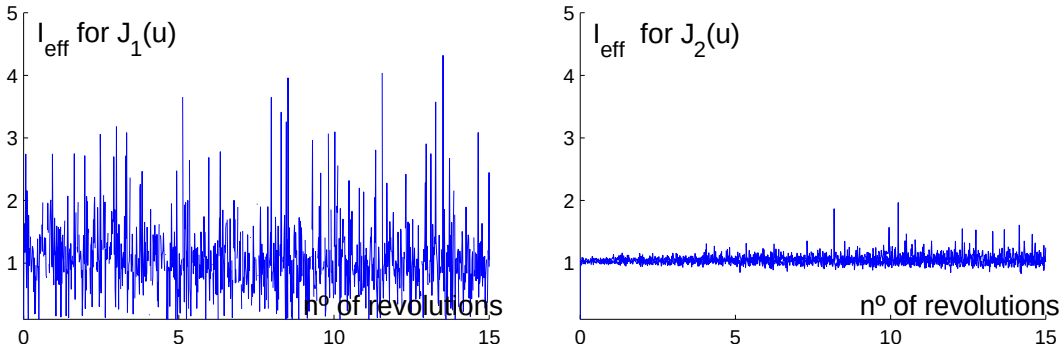

Figure 4.6: Effectivity index for the two functionals J_1 and J_2 versus the number of revolutions.

We show in Figure 4.7, the mesh obtained at the last time step (a mesh with 2016 elements and 4067 nodes for the functional J_1 and a mesh with 1935 elements and 3912 nodes for the functional J_2) and the isolines of the solution from 0.1 to 0.9 at intervals 0.1. Although the refinement zone is around the discontinuity, we can see a slight difference. When we use the first functional J_1, we refine uniformly throughout the discontinuity, and the numerical diffusion is similar at the bottom and on the top of the cylinder; however, when we use the functional J_2, we make a special control of the higher values along the discontinuity, and we have more numerical diffusion at the bottom than on the top of the cylinder.

In Figure 4.8 we have plotted the error of the numerical solution in time. We have represented the error for the first functional $\int u d\Omega - \int u_{h\Delta t} d\Omega$, for the second functional $\int u^2 d\Omega - \int u_{h\Delta t}^2 d\Omega$ and for the L^2-norm $\left[\int \left(u - u_{h\Delta t}\right)^2 d\Omega\right]^{1/2}$. We have plotted the curves of the error for both numerical experiments (for J_1 and J_2 adaptation).

4.5.2 Example 2. A convection-diffusion problem

We consider a convection-diffusion problem taken from [46]. The equation to be solved is the following

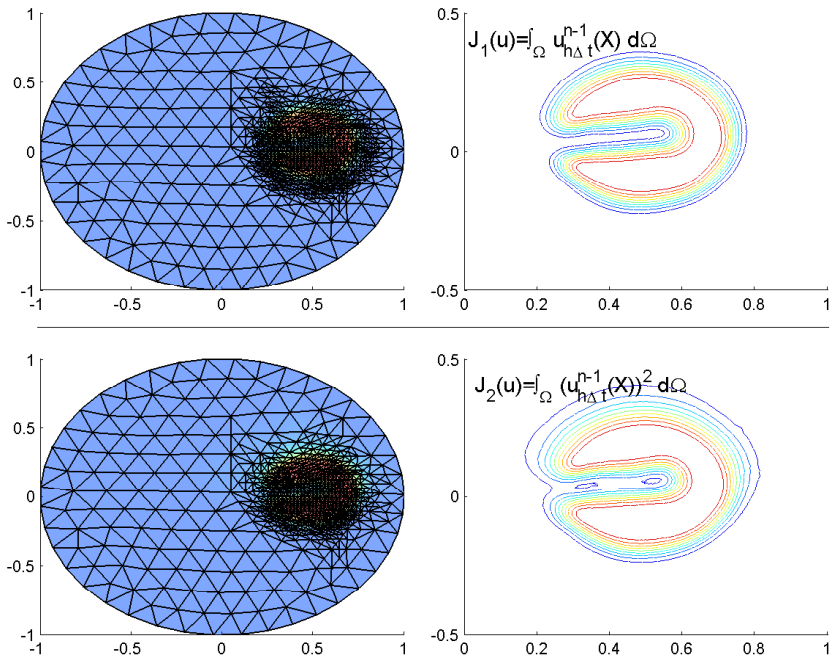

Figure 4.7: Mesh (left) and cross section of the solution $(x_2 = 0)$ (right) for the functionals: J_1 (above) and J_2 (below).

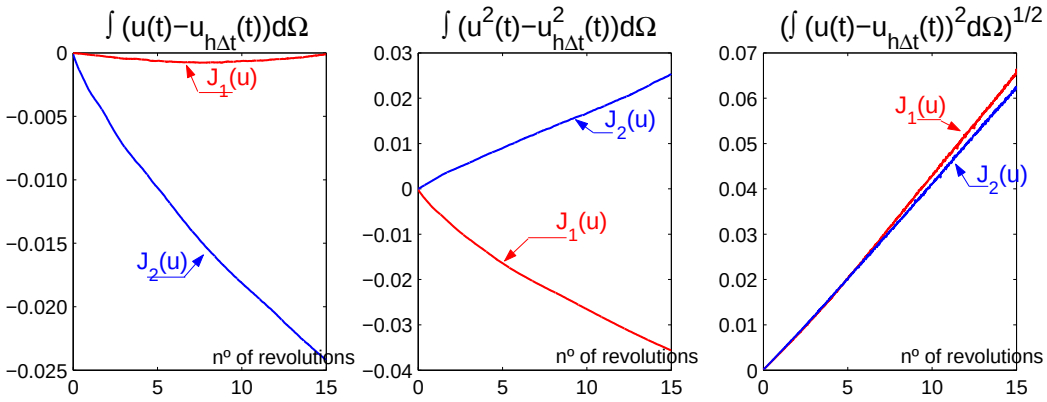

Figure 4.8: Error in the mass (left) in the energy (middle) and in the L^2 norm (right) of the solution for the two numerical experiments (for J_1 and J_2 adaptation).

$$\begin{cases} \dfrac{\partial u}{\partial t} + \mathbf{a}(x,t)\cdot\nabla u = k\Delta u \quad \text{in } \Omega\times(0,T], \\ u(x,t) = g(x,t) \quad \text{on } \partial\Omega, \\ u(x,0) = u^0(x). \end{cases} \tag{4.38}$$

With the following considerations:

- The domain $\Omega \in \mathbb{R}^2$ is $\Omega = (0,1)^2$ and $T = 0.55$.
- The initial condition, is defined as $u^0(x) = 0$ for $x = (x_1, x_2) \in \Omega_\delta = (\delta, 1)\times(0, 1-\delta)$.

For $x \in \Omega \backslash \Omega_\delta$, $u^0(x)$ is defined to be the linear function which satisfies the boundary conditions.

- The Dirichlet boundary conditions are:

$$u(x,t) = g(x,t) = \begin{cases} 1 & \text{for } x_1 = 0,\ 0 \le x_2 \le 1, \\ 1 & \text{for } 0 \le x_1 \le 1,\ x_2 = 1, \\ \dfrac{(\delta - x_1)^+}{\delta} & \text{for } 0 \le x_1 \le 1,\ x_2 = 0, \\ \dfrac{(x_2 - 1 + \delta)^+}{\delta} & \text{for } x_1 = 1,\ 0 \le x_2 \le 1, \end{cases}$$

  where $(a)^+ = \max(0, a)$ and $\delta = 7.8125 \times 10^{-3}$.

- The diffusion coefficient $k = 10^{-3}$.

- The velocity field $\mathbf{a}(x,t) = [2,\ 1]^T$.

In the example we consider as output functional to control the errors

$$J(U^n) = \int_\Omega (1 - U^n) d\Omega$$

and the tolerances $Tol_s = Tol_t = 2 \cdot 10^{-4}$.

We note that (for δ small enough) initially the solution to this problem has boundary layers along $x_1 = 0$ and $x_2 = 1$. The boundary layer along $x_1 = 0$ propagates into the domain Ω and interacts with the outflow boundary at $x_1 = 1$, where a new boundary layer develops at time $t \sim 0.5$. The combination of both internal and boundary layers makes it a challenging model problem.

We will consider for the numerical integration a background 32×32 triangular mesh shown in Figure 4.9 (left). This mesh is initially refined in order to resolve the boundary layers along $x_1 = 0$ and $x_2 = 1$ at time $t = 0$ (see Figure 4.9 (right)) using the functional $J(u)$ and the marked tolerance.

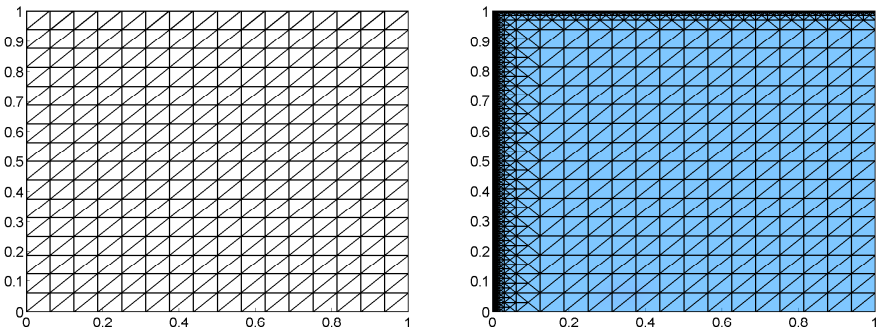

Figure 4.9: Macro triangulation (left) with 512 elements and 1089 nodes. Initial mesh (right) adapted to solve the initial condition with 2600 elements and 5617 nodes.

In Figure 4.10 we plot both the spatial and temporal error indicators. We further divide the error indicators into two components, namely, the convection error and the diffusion-reaction error, with their corresponding spatial and temporal error indicators; we have then $\eta_s^n = \eta_{s,convect}^n + \eta_{s,diff-react}^n$ and $\eta_t^n = \eta_{t,convect}^n + \eta_{t,diff-react}^n$.

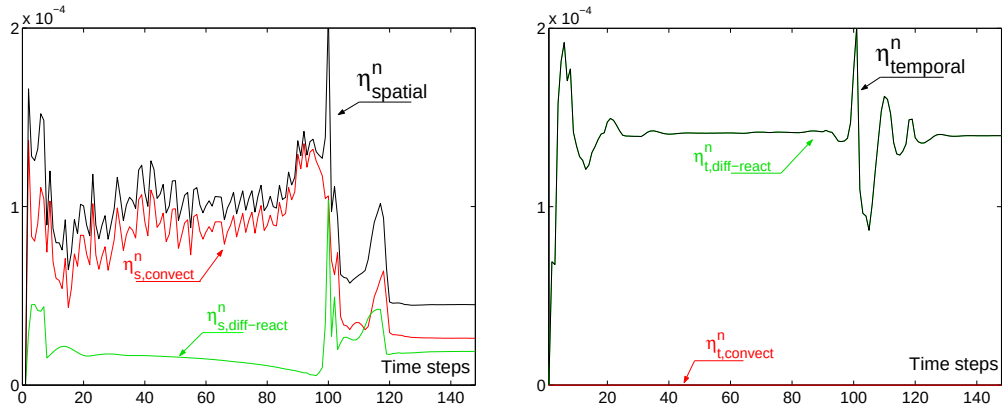

Figure 4.10: Spatial (left) and temporal (right) error indicators and their respective convection $\eta^n_{convect}$ and diffusion-reaction $\eta^n_{diff-react}$ errors plotted against the number of time steps.

The convection error is due to the semi-Lagrangian scheme and we note, at least in this example, that its spatial component is much larger than the spatial component of the diffusion-reaction error; however, the time error is due to the diffusion-reaction component. An explanation for the behaviour of the time error in this example could be that $\eta^n_{t,convect}$ is zero because the feet of the characteristics are calculated exactly when the velocity is a constant.

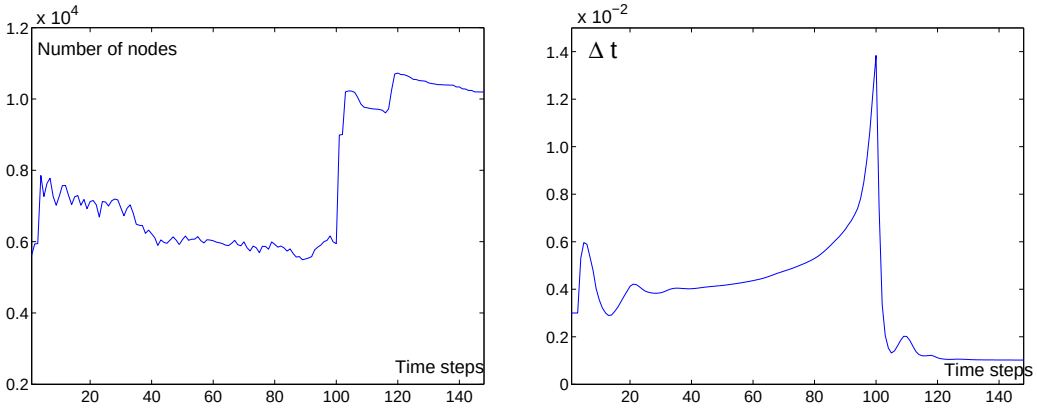

Figure 4.11: History of nodes (left) and time step size (right) plotted against the number of time steps.

In Figure 4.11 we represent a history of the number of nodes and the time step size against the number of time steps. Initially, and in order to resolve the internal and boundary layers, there is a large number of nodes; however, as time progresses, the layers become smoother through the diffusive process and the number of nodes gradually decreases to a minimum before the appearance of the boundary layer at $x = 1$, when the number of nodes increases again. A reversal trend is followed by the size of the time step as shown in the left panel of the figure.

Numerical results are presented in Figures 4.12 and 4.13 at times $t = 0.12$ and $t = T = 0.55$ respectively. These figures show that the spatial mesh is concentrated in the internal and boundary layers of the solution. In particular, we see that the boundary conditions are nearly discontinuous. Further, the mesh emanating from the bottom left-hand corner (i.e. at $x_1 = 0$ and $x_2 = 0$) is finer on the top and at the bottom of the internal layer due to the viscosity. In the

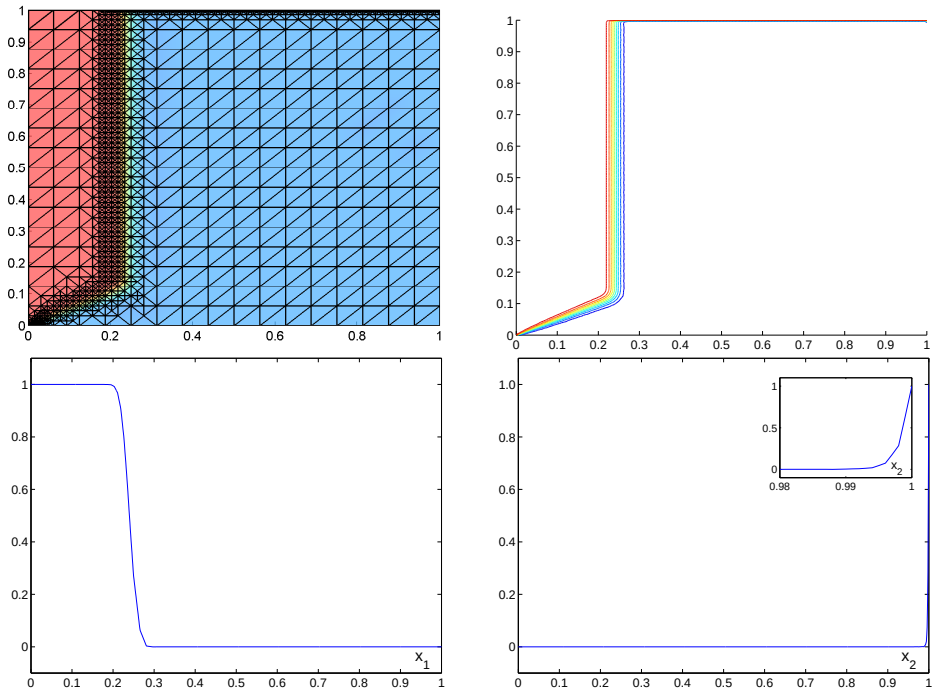

Figure 4.12: Solution at time $t = 0.12$. Top-left: the mesh with 3328 elements and 6931 nodes; Top-right: isolines from $u = 0.1$ to $u = 0.9$ at intervals $\Delta u = 0.1$. Bottom-left: cross-section at $x_2 = 0.75$; Bottom-right: cross-section at $x_1 = 0.5$.

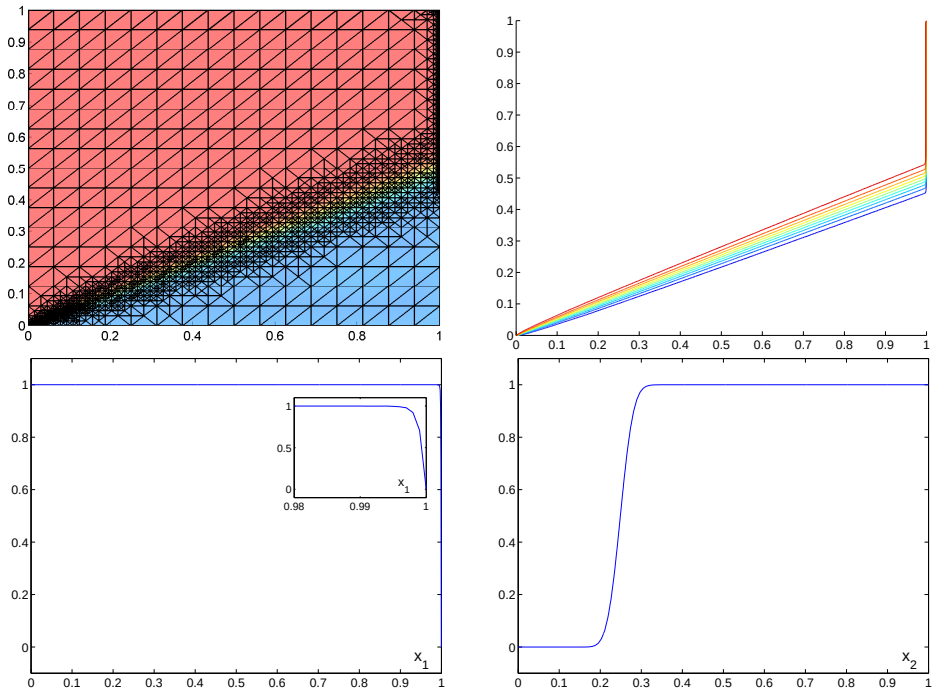

Figure 4.13: Solution at time $t = 0.55$. Top-left: the mesh with 4924 elements and 10196 nodes; Top-right: isolines from $u = 0.1$ to $u = 0.9$ at intervals $\Delta u = 0.1$. Bottom-left: cross-section at $x_2 = 0.75$; Bottom-right: cross-section at $x_1 = 0.5$.

lower panels we plot cross sections of the solution at $x_2 = 0.75$ and $0 \leq x_1 \leq 1$, and at $x_1 = 0.5$ and $0 \leq x_2 \leq 1$; we can see that the implementation of the adaptive mesh algorithm gives rise to smooth approximations of the very steep features of the solution, showing no oscillation in the zoom (a problem that is shown in [46]) at the outflow boundary layer.

Another characteristic of the algorithm is that the number of time steps to complete the temporal integration has been 148, with a total of 4 time steps rejected at the beginning of the integration and when the boundary layer appears, at time $t \sim 0.5$, causing a sudden reduction in the time step size. We have needed $CPU_{time} = 147\ seg.$ to solve the problem and the distribution of time in each part of the algorithm is shown in Figure 4.14.

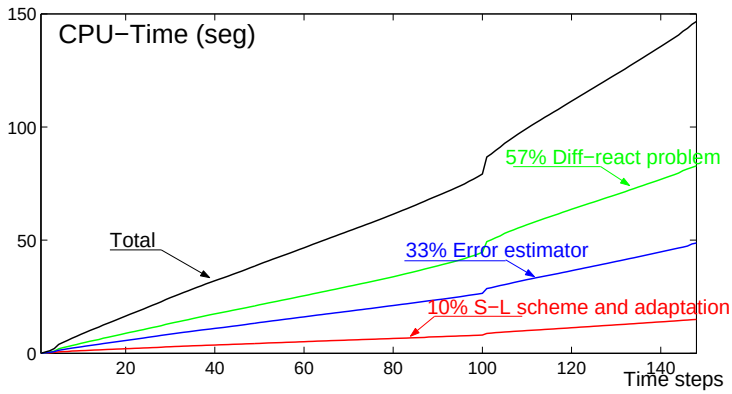

Figure 4.14: Distribution of the CPU time required in each part of the algorithm to calculate the solution plotted against the number of time steps. The semi-Lagrangian stage with the mesh and time adaptation represents 10%, the resolution of the primal diffusion-reaction problem 57% and to calculate the a posteriori error estimator 33% of the CPU time is needed.

4.5.3 Example 3. A 2D convection-diffusion-reaction problem

We will test our numerical adaptive procedure with 2D convection-reaction-diffusion system that models the transport of two chemical species in a plane shear flow. The model equation for $u(x,t) = [u_1(x,t), u_2(x,t)]^T$ is:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \mathbf{a}(x,t) \cdot \nabla u = k\Delta u + f(u) \quad \text{in } \Omega \times (0,T], \\ \\ u(x,t) = 0 \quad \text{on } \partial\Omega, \\ \\ u(x,0) = u^0(x). \end{cases} \tag{4.39}$$

- The initial condition $u^0(x) = [u_1^0,\, u_2^0]^T$ is

$$u_1^0(x) = \exp\left(-30\left[(x_1 - 2.5)^2 + {x_2}^2\right]\right) \quad \text{and} \quad u_2^0(x) = 0.$$

- The reaction term $f(u) = [f_1,\, f_2]^T$ has as components

$$f_1(u) = -c_1 u_1^2 + c_2 u_2 u_1 \quad \text{and} \quad f_2(u) = -c_2 u_1 u_2 + k_1 u_1^2.$$

where c_1 and c_2 are given positive constants that in the experiment take the value $c = c_1 = c_2 = 10^3$.

- The diffusion coefficients $k = diag(k_1,\ k_2)$. In the experiment $k_1 = k_2 = 10^{-3}$.
- The velocity $\mathbf{a}(x,t) = [a_1,\ a_2]^T$ with

$$a_1(x,t) = 2\,(1 + 5\sin(2\pi t)\cdot x_2) \quad \text{and} \quad a_2(x,t) = 0,$$

defining thus a solenoidal velocity field that produces stretching in the solution with period 1.

We consider in this example tolerances $Tol_s = Tol_t = 2\cdot 10^{-4}$ to control the mass of one of the reactants (a good functional in combustion problems). Then, we define the following functional

$$J(U^n) = \int_\Omega U^n d\Omega.$$

We define $\Omega = (0,7)\times(-1,1)$ with a background mesh of 126 elements and 285 nodes, and a time integration interval $(0,1]$. We will finish the temporal integration when the center of the solution is at $(x_1,x_2) = (4.5,\ 0)$ and it returns close to its initial shape after the stretching.

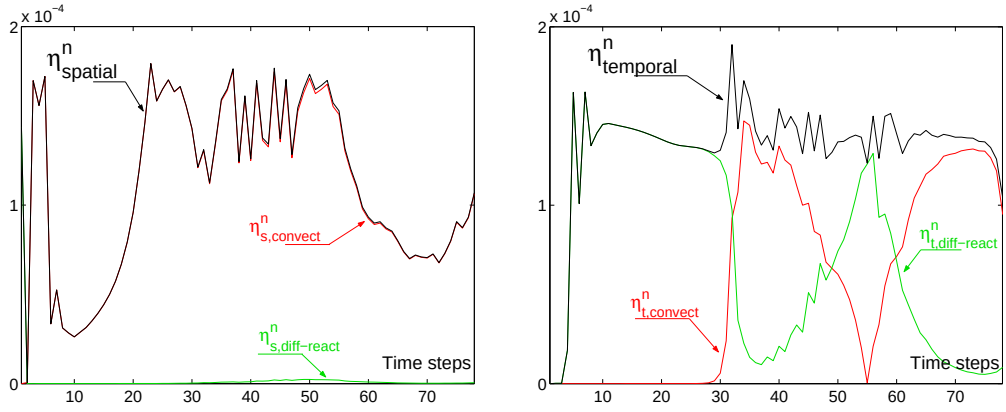

Figure 4.15: Spatial (left) and temporal (right) error indicators and their respective convection $\eta^n_{convect}$ and diffusion-reaction $\eta^n_{diff-react}$ errors plotted against the number of time steps.

With this example we want to analyze a problem whose character changes as time progresses and several temporal scales appear. The behaviour of the solution in time is the following: since the stiffness parameter is $c = 10^3$, the component $u_1(x,t)$ decreases very rapidly from $u_1^0(x,t)$ to $u_1^0(x,t)/2$ and $u_2(x,t)$ increases very rapidly from 0 to $u_1^0(x,t)/2$. After the transient state, the reaction term becomes negligible and the convection term is dominant. Then, the solution is stretched by the velocity field and the diffusion term becomes important (around $t = 0.5$, the stretching is maximum). This behavior is shown in Figure 4.15, where the spatial and the temporal error indicators are plotted against time steps with their convection and diffusion-reaction errors. On one hand, the spatial error is composed almost exclusively of the convection error, and on the other hand, the temporal error is composed of diffusion-reaction error due to the chemical reaction, at its earlier stages; it reaches the chemical equilibrium, the time step size grows and the convective terms become important. We also note that when the solution has its highest elongation, the diffusion-reaction error will be important, and this error decreases when the solution goes back to its original shape at time $t = T = 1$.

In Figure 4.16 we show the evolution of the number of nodes and the time step size against the number of time steps. The spatial discretization at the beginning is only necessary to maintain the shape of the initial condition, and although the spatial error is smaller we need to take a

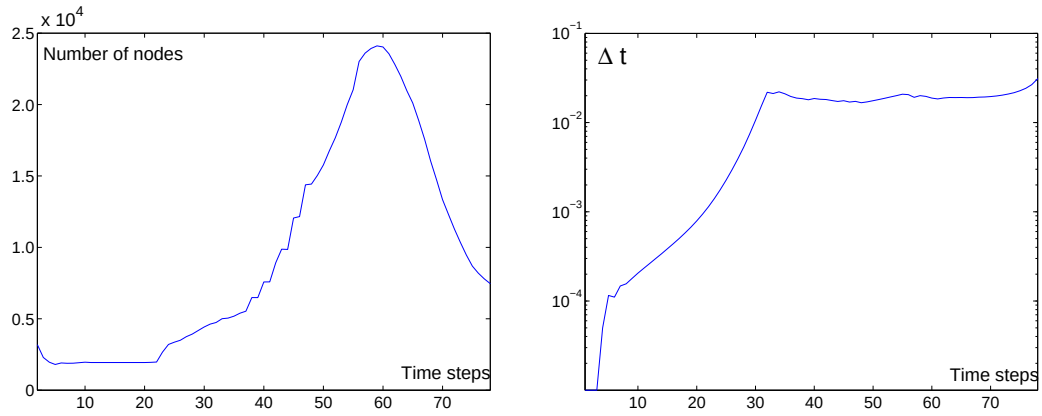

Figure 4.16: History of nodes (left) and time step size (right) plot against the number of time steps.

number of nodes large enough to reproduce the shape of the solution. Then, the convective term becomes important and the number of nodes increases and reaches a peak when the stretching produced by the fluid is maximum at $t = 0.5$. Finally, the velocity field moves the solution back to its circular shape and the numbers of elements decrease. Regarding the time step size, we can see how it grows as time goes by and the chemical reaction reaches the equilibrium, then the time step size is governed by the convective term and part of it by the diffusion terms.

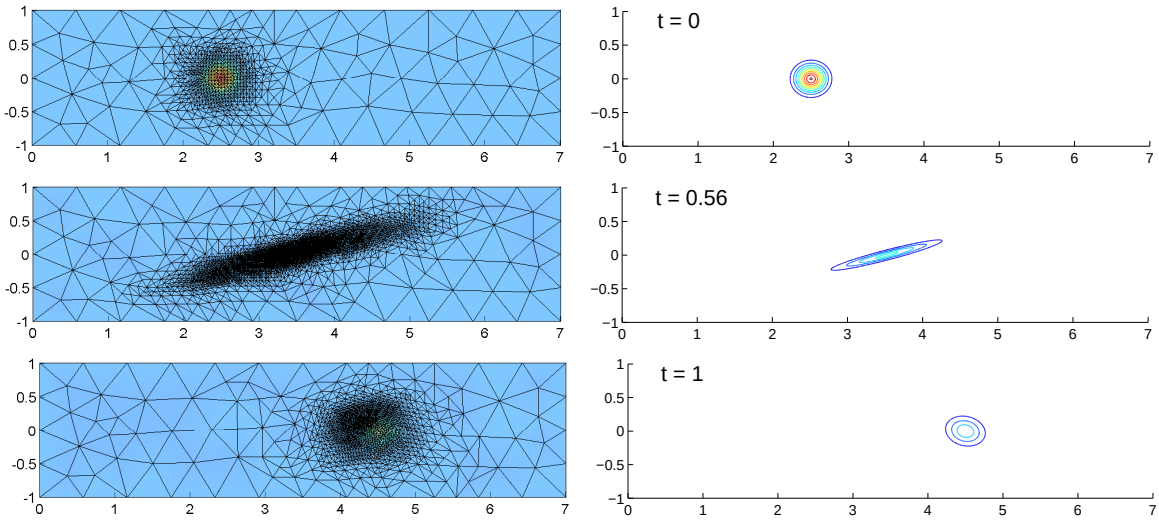

Figure 4.17: Mesh (left) and isolines (right) for the solution u_1 at time t=0, t=0.56 and t=1.

In Figure 4.17 we present the numerical solution u_1 at different time instants: $t = 0$, $t = 0.56$ and $t = T = 1$. Here, we can see the isolines of the solution and the adaptive mesh refined around the numerical solution. The mesh has been refined as one would expect, being noticeable the small refined region at the trail of the solution where the error of the convective terms accumulate.

At $t = 0$ we have a mesh with 1574 elements and 3185 nodes, where we have plotted the isolines of the solution from $u_1 = 0.1$ to $u_1 = 0.9$ at intervals $\Delta u_1 = 0.1$. At $t = 0.56$ we have a mesh with the maximum number of elements and nodes, 12033 elements and 24108

nodes, and we have plotted the isolines of the solution $u_1 = 0.1,\ 0.2,\ 0.3$ and 0.4. At time $t = T = 1$ we have a mesh with 3542 and 7121 nodes and the isolines plotted are $u_1 = 0.1$, 0.2 and 0.3. Note that the deformation of the solution at the end is caused by the diffusion term around $t = 0.5$, (where we have the maximal deformation of the Gaussian hill). We have run this example without diffusion and the final solution is a Gaussian hill of value $u_1^0/2$ with the center at point $(4.5, 0)$, as we expected.

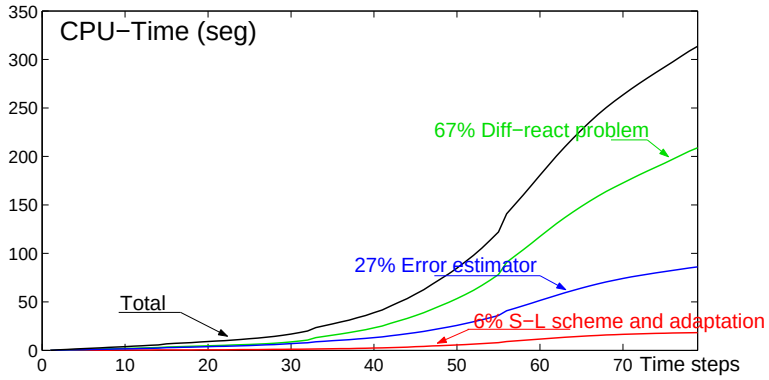

Figure 4.18: Distribution of the CPU time required by each part of the algorithm plotted against the number of time steps. The semi-Lagrangian stage with the mesh and time adaptation represents 6%, the resolution of the diffusion-reaction problem 67% and to calculate the a posteriori error estimator 27% of the CPU time is needed.

The number of time steps to complete the temporal integration has been 79, with 1 time step rejected at the beginning of the integration. We have needed a CPU_{time} of 313 $seg.$ to solve the problem and the distribution of time in each part of the algorithm is shown in Figure 4.18.

Chapter 5

LIFTED FLAMES PROBLEM

The main characteristic of lifted flames is that the flame front is separated from the injector a distance x_f, such that the flame front in non-premixed systems presents a typical triple flame structure. Triple flames, also known as tribrachical flames, play an important role in the lift-off and blow-off as well as in ignition of diffusion flames by localized heat sources. Such flames have been studied experimentally, theoretically and numerically, due to the role they play as transient laminar flamelets in partially premixed systems.

The first analysis of the two-dimensional structure of a triple flame is due to Dold (1989) [30], who used a constant density model with a single step chemical reaction with large activation energy. The analysis of triple flames was later extended to non-unity Lewis numbers by Daou and Liñán (1998) [28], still using the constant density approximation. Using a different approach, Buckmaster (1996) [23] developed a simplified one-dimensional model containing some of the key physical ingredients of the problem and despite its simplicity, the one-dimensional model predicts flames with advancing, retreating, or stationary edges depending on the Damköhler number $D_m = (\delta_m/\delta_L)^2$ determined by the ratio of the characteristic thickness, δ_m, of the non-reacting mixing layer at the flame front position and the laminar flame thickness, δ_L. The effects of thermal expansion on the propagation velocity of triple flames were first studied by Ruetsch et al. (1995) [60]. For further references, an extensive review of the work made in this field can be found in the paper of Buckmaster (2002) [24].

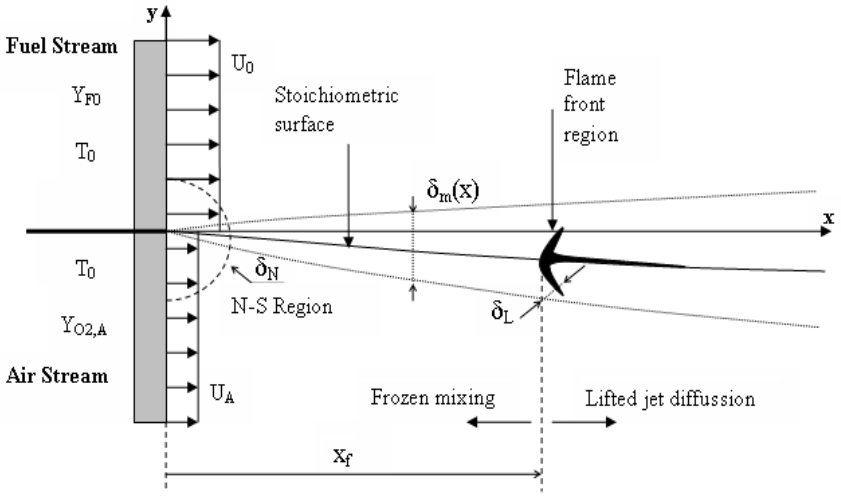

Figure 5.1: A sketch of the mixing layer configuration, showing the typical scales of the problem.

The main characteristics of the diffusion flames are located in the mixing layer that is developed in the presence of two parallel streams of fuel and air. Figure 5.1 shows a schematic representation of the problem, where the inflow velocities of the streams are U_0 for the fuel and $U_A \leq U_0$ for the air coming out normal from the porous wall with uniform temperature T_0. $Y_{F,0}$ and $Y_{O_2,A}$ denote the mass fraction of fuel and oxygen (diluted with Nitrogen) in the feed streams respectively.

In Figure 5.1 we note that the Navier-Stokes region, where the mixing layers originate, has a length scale $\delta_N = D_{TA}/U_0$, D_{TA} being the thermal diffusivity of the air stream at temperature T_0. Further downstream, at distances $x >> \delta_N$, the Navier-Stokes region evolves to a slender mixing layer with a thickness of order $\delta_m = (D_{TA}x_f/U_0)^{1/2} << x_f$. We may provoke the ignition of a flame in the mixing layer by an external localized energy source such that the flame front becomes rapidly elongated by the flow and a quasi-planar premixed flame is formed with a characteristic thickness $\delta_L = \frac{D_{TA}}{S_L}$ ($\delta_L \simeq 10^{-4}m$ in typical hydrocarbons). The premixed flame moves upstream and downstream along the stoichiometric surface and exhibits a characteristic structure with both lean and rich branches. On the lean side the premixed flame consumes all of the available fuel, leaving behind oxygen that reacts in a trailing diffusion flame with the fuel left behind by the rich branch.

An important kinetic parameter that plays an essential role in combustion processes is the ratio U_0/S_L of the fuel stream velocity to the planar flame velocity S_L of the stoichiometric mixture. In the mixing layer example if the ratio U_0/S_L is smaller than a critical value, the diffusion flame will be anchored in the Navier-Stokes region near the injector wall $x_f \sim \delta_N$. As U_0/S_L grows, the flame will be lifted to a length x_f, in this regime the flame front structure is that of a curved premixed flame with a radius of curvature of order δ_m, large compared with the flame front thickness of order δ_L. Correspondingly, the curved flame front propagates against the incoming flow as a quasi-planar premixed flame with a maximal velocity S_L in the stoichiometric surface, and decays rapidly as the mixture becomes either leaner or richer. Finally, if the ratio U_0/S_L is larger than the critical value $(U_0/S_L)_b$ the flame will be blown-off.

Clearly, as x_f/δ_N grows the numerical description of the flame region becomes increasingly difficult because the computational domain must now be extended further and further downstream if we wish to solve the flame front region. This suggests that an adaptive algorithm would be efficient and accurate to obtain the numerical solution when the flame front is located far away from the porous wall. Thus, we present an adaptive numerical method based on the philosophy of the DWR method in the framework of finite elements to simulate the behavior of lifted flames for a large range of Reynolds numbers (we exclude the turbulent limit). A significant property of the method is its ability to adapt both the spatial mesh and the time step size following the features of the solution.

5.1 Physical model and governing equations

In this section, we present the equations governing chemical reactive flows in the low-Mach-number limit. They are based on the compressible formulation of the Navier-Stokes equations with additional convection-diffusion-reaction equations for the temperature and the chemical species. A good explanation of the equations can be seen in the thesis of M. Braack [20]. The variables of the problem are ρ for density, p_{total} for pressure, $\mathbf{u}$ for velocity, T for temperature and Y_α for species mass fractions of the reactants and products.

The compressible form of the Navier-Stokes equations allows two kinds of propagation speeds: the velocity of the flow and the speed of sound. In many reactive flow problems, e.g., burners and reaction chambers, the flow is slow in comparison to the speed of sound. This is the case of low Mach number where we can consider that the sound waves will propagate infinitely fast, and the gas becomes hydrodynamically incompressible. For low Mach number, only thermodynamical effects (e.g., heat release) lead to compression; in other words, the compression due to temperature variation is significantly more important than that due to pressure variations.

The first part of this section describes the compressible Navier-Stokes equations with the simplification of low Mach number. Then, we deduce the convection-diffusion-reaction equations for the temperature and the chemical species. At the end of the section we focus on the modeling of the chemical source terms and the coefficients of the equations.

5.1.1 Compressible Navier-Stokes equations

Here we briefly describes the low-Mach-number model of Navier-Stokes equations for compressible gases. We discuss separately the continuity equation and the momentum equation. In the compressible formulation, the equation for the density ρ is the continuity equation, which describes the conservation of mass

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0. \tag{5.1}$$

We can manipulate (5.1) to obtain an expression with the total or material derivative. Assuming that the density is always a positive real number, then we can write

$$\frac{1}{\rho}\frac{\partial \rho}{\partial t} + \frac{1}{\rho}\nabla \cdot (\rho \mathbf{u}) = 0, \quad \frac{1}{\rho}\frac{\partial \rho}{\partial t} + \mathbf{u} \cdot \frac{1}{\rho}\nabla \rho + \nabla \cdot \mathbf{u} = 0, \quad \frac{\partial \ln \rho}{\partial t} + \mathbf{u} \cdot \nabla \ln \rho + \nabla \cdot \mathbf{u} = 0,$$

and finally

$$\frac{D \ln \rho}{Dt} + \nabla \cdot \mathbf{u} = 0. \tag{5.2}$$

The expression (5.2) will be more useful than the continuity expression (5.1) because we can integrate in time this equation taking advantage of the semi-Lagrangian framework.

The vector-valued momentum equation insures the conservation of the momentum $\rho\mathbf{u}$,

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p_{total} + \nabla \tau + \rho \mathbf{f}_e. \tag{5.3}$$

It consists of the temporal and convective part of the momentum, the pressure gradient, the viscous stress tensor and external forces, which can be, e.g., gravitational effects (although in our problem we shall consider that gravitational effects are negligible in gas streams). With the exception of very high temperatures or pressures, the viscous stress tensor for Newtonian fluids has the form

$$\tau = \mu \left[\left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) + \left(\mu_v - \frac{2}{3}\mu \right) (\nabla \cdot \mathbf{u}) \, I \right],$$

where μ is the dynamic viscosity, μ_v is the volumetric viscosity of the fluid and I is the second order unit tensor. The expression $\nabla \cdot \tau$ is therefore

$$\nabla \cdot \tau = \nabla \cdot (\mu \nabla \mathbf{u}) + \nabla \left(\mu \nabla \cdot \mathbf{u} \right) + \nabla \left[\left(\mu_v - \frac{2}{3}\mu \right) (\nabla \cdot \mathbf{u}) \right].$$

In the incompressible case ($\nabla \cdot \mathbf{u} = 0$), the divergence of the viscous stress tensor simplifies to

$$\nabla \cdot \tau = \nabla \cdot (\mu \nabla \mathbf{u}) .$$

But we can make a further simplification of the momentum equation for low-Mach-number flow because the fully compressible Navier-Stokes equations becomes ill-conditioned for low Mach numbers. There are many publications devoted to numerical methods to treat low-Mach-number flows (for instance [49]). A common option to solve them is to split the pressure field into two parts, namely, thermodynamical part and hydrodynamical part, such that

$$p_{total}(x,t) = P_{th}(t) + p_{hyd}(x,t).$$

The thermodynamical part P_{th} is uniform in space and does not appear in the momentum equation (5.3). P_{th} may be determined as the average pressure in Ω. It is only variable with time, but in the case of an open boundary vessel this term is also constant in time (the hypothesis in our problem). On other hand, the hydrodynamic part p_{hyd} varies in both time and space, and is far smaller than P_{th}. This is important since the influence of p_{hyd} is neglected in the gas law and therefore the density only varies in space due to gradients of temperature and mean molar mass (see 5.10). With this splitting of the pressure we can write the divergence of the whole stress tensor as

$$\begin{cases} \nabla \cdot \tau - \nabla p_{total} = \nabla \cdot (\mu \nabla \mathbf{u}) + \nabla \left[\left(\mu_v + \frac{1}{3}\mu \right) (\nabla \cdot \mathbf{u}) \right] - \nabla p_{hyd} = \nabla \cdot (\mu \nabla \mathbf{u}) - \nabla p, \\ \\ p = p_{hyd} - \left(\mu_v + \frac{1}{3}\mu \right) (\nabla \cdot \mathbf{u}) , \end{cases}$$

where we have defined a new variable p that allows us to simplify the momentum equation (5.3) as

$$\rho \frac{D\mathbf{u}}{Dt} = \nabla \cdot (\mu \nabla \mathbf{u}) - \nabla p. \tag{5.4}$$

Note that in the quasi-isobaric limit the momentum conservation equation takes the same form as the incompressible Navier-Stokes equation, the only difference is that p is not the real pressure of the fluid. The function p is fundamental in establishing the fluid motion in order to satisfy the equation of continuity but it has not physical meaning.

5.1.2 Temperature and species conservation equations

To describe the evolution of the temperature T, we use the second law of thermodynamics, the entropy equation

$$\rho T \left(\frac{\partial s}{\partial t} + \mathbf{u} \cdot \nabla s \right) = \Phi_v - \nabla \cdot \mathbf{q}_c + Q_F + Q_r, \tag{5.5}$$

where:

- Φ_v represents the heat due to viscous dissipation which will be neglected because is not important compared with reaction heat.
- Q_r represents the radiation effects which we will omit because only are important for high temperatures of flame.
- $-\nabla \cdot \mathbf{q}_c$ represents a diffusive part of heat conduction. The mathematical expression of this terms is giving by the Fourier law $\mathbf{q}_c = -k\nabla T$. Then this term takes the form $-\nabla \cdot \mathbf{q}_c = \nabla \cdot (k \nabla T)$, where k is the heat conductivity.

- Q_F represents the heat due to chemical reactions. This term is modelled by the product of the specific heat of the chemical reaction H_F and the consumed fuel mass fraction by the chemical reaction ω_F. Therefore $Q_F = -H_F \cdot \omega_F$.

- We have also omitted in this equation the Dufour effect, which describes heat transport due to diffusion of chemical species with different enthalpies, and is needed only in special situations.

- s is the thermodynamic variable named entropy. By the second Gibbs equation we can write s as function of the enthalpy h and the pressure p

$$T \cdot ds = dh - \frac{1}{\rho} dp.$$

Since the Mach number is small compared with unity, then we can consider that the pressure variations are negligible with respect to the value of pressure. Then $Tds = dh$ and if we consider an ideal gas mixture, the enthalpy is $dh = c_p dT$, with c_p being the specific heat capacity for constant pressure. We can write for the temperature a convection-diffusion-reaction equation of the form

$$\rho c_p \frac{DT}{Dt} = \nabla \cdot (k \nabla T) - H_F \omega_F. \tag{5.6}$$

And if we assume that the spatial variation of c_p is negligible compared with its mean value, we can introduce the thermal diffusion coefficient D_T as:

$$\rho D_T = k/c_p,$$

and this allows us to write the equation of the conservation of the energy as

$$\rho \frac{DT}{Dt} = \nabla \cdot (\rho D_T \nabla T) - \frac{H_F}{c_p} \omega_F. \tag{5.7}$$

For multicomponent fluids, additional equations for each involved species have to be added. The species will be represented by their mass fractions and each species can be described by its own characteristic diffusion behavior $\nabla \cdot \mathbf{j}_\alpha$ and the reaction rate ω_α that depends on the mass fractions and the temperature. For a mixture of n_s chemical species, the corresponding equations are:

$$\rho \left(\frac{\partial Y_\alpha}{\partial t} + \mathbf{u} \cdot \nabla Y_\alpha \right) = -\nabla \cdot \mathbf{j}_\alpha + \omega_\alpha, \qquad \alpha = 1, ...n_s. \tag{5.8}$$

This system of equations has to be solved under the compatibility conditions

$$\sum_{\alpha=1}^{n_s} Y_\alpha = 1 \text{ and } \sum_{\alpha=1}^{n_s} \mathbf{j}_\alpha = 0.$$

To model the diffusive flux terms $\mathbf{j}_\alpha$ we use the Fick law $\mathbf{j}_\alpha = -\rho D_\alpha \nabla Y_\alpha$, where D_α are the mass diffusion coefficients. This approximation is true if there are only two species in the mixture or one of the species is skipped (case of an abundant inert gas is the mixture).

We suppose that in the mixture the specie Y_{n_s} is abundant and inert, therefore the governing equations for the chemical species take the expression.

$$\begin{cases} \rho\left(\dfrac{\partial Y_\alpha}{\partial t} + \mathbf{u}\cdot\nabla Y_\alpha\right) = \nabla\cdot(\rho D_\alpha \nabla Y_\alpha) + \omega_\alpha, \qquad \alpha = 1, ...n_s - 1. \\ Y_{n_s} = 1 - \sum\limits_{\alpha=1}^{n_s-1} Y_\alpha. \end{cases} \tag{5.9}$$

These equations have to be closed by an algebraic relation between pressure, temperature and density, namely, the equation of state. In our examples we consider the ideal gas law

$$p_{total} = \frac{\rho R T}{M}, \tag{5.10}$$

where R is the universal gas constant and M is the mean molar mass of the mixture. In multicomponent fluid, where the species have different mole masses, the mean molar mass M depends on the chemical composition according to the relation

$$\frac{1}{M} = \sum_{\alpha=1}^{n_s} \frac{Y_\alpha}{M_\alpha},$$

where M_α is molar mass of species α. As we write above, the pressure is split into two parts, where the thermodynamic part P_{th}, which is uniform in space, is several orders of magnitude bigger than p_{hyd}, which is variable in space and in time. Then, $p_{total} \simeq P_{th}$ and if P_{th} is constant in time the equation of state will be written in the form

$$\rho = \frac{P_0}{R}\frac{M}{T}. \tag{5.11}$$

5.1.3 Physical parameters and constants

In the previous part of this section we have formulated the mathematical model that describes the behavior of chemical reactive fluid in the low-Mach-number limit. Thus, the model consists of the compressible Navier-Stokes equations at small Mach number (5.2) and (5.4), plus convection-diffusion-reaction equations for the temperature (5.7) and chemical species (5.9), with strong nonlinearity and stiff source terms characteristics in combustion problems, plus the equation of state (5.10) for the fluid.

In the practical resolution of the equations we consider a 2D planar model, where the fuel stream will be methane CH_4 diluted in Nitrogen N_2 that reacts with an air stream (with 77% of N_2 and 23% of O_2 in mass). The analysis takes into account the effects of thermal expansion and assumes a one-step overall Arrhenius reaction, thus CH_4 reacts with the oxygen of the air O_2 to generate the combustion products that are CO_2 and H_2O. As an additional hypotesis, the considered Reynolds numbers Re are small or relatively large, but always far of the turbulent limit.

We consider that $\Omega \subset \mathbb{R}^2$ is a domain with appropriately smooth boundary $\partial\Omega := \Gamma^D \cup \Gamma^N$, $\Gamma^D \cap \Gamma^N = \emptyset$, where Γ^D and Γ^N are the pieces of $\partial\Omega$ for Dirichlet and Neumann boundary conditions respectively. The system of equations of the model are:

- Navier-Stokes equations:

$$\begin{cases} \dfrac{D\ln(\rho)}{Dt} + \nabla\cdot\mathbf{u} = 0, \\ \rho\left(\dfrac{\partial\mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = \nabla\cdot(\mu\nabla\mathbf{u}) - \nabla p, \end{cases} \tag{5.12a}$$

with Dirichlet and Neumann boundary conditions

$$\mathbf{u} = \mathbf{u}^D \ \ on\ \Gamma^D_{\mathbf{u}}, \qquad\qquad \mu\frac{\partial\mathbf{u}}{\partial n} - p\mathbf{n} = 0 \ \ \ on\ \Gamma^N_{\mathbf{u}} = \partial\Omega\backslash\Gamma^D_{\mathbf{u}},$$

where the unit outer normal vector to the boundary is denoted by $\mathbf{n}$.

An initial condition only for the velocity $\mathbf{u}(t=0)$ is needed.

- Combustion equations:

$$\rho\left(\frac{\partial T}{\partial t} + \mathbf{u}\cdot\nabla T\right) = \nabla\cdot(\rho D_T\nabla T) - \frac{H_F}{c_p}\omega_{CH_4}, \tag{5.12b}$$

with Dirichlet and Neumann boundary conditions

$$T = T^D \ \ \ on\ \Gamma^D_T, \qquad\qquad \rho D_T\frac{\partial T}{\partial n} = 0 \ \ \ on\ \Gamma^N_T = \partial\Omega\backslash\Gamma^D_T,$$

and initial conditions for the temperature $T(t=0)$.

$$\begin{cases} \rho\left(\dfrac{\partial Y_{CH_4}}{\partial t} + \mathbf{u}\cdot\nabla Y_{CH_4}\right) = \nabla\cdot(\rho D_{CH_4}\nabla Y_{CH_4}) + \omega_{CH_4}, \\ \rho\left(\dfrac{\partial Y_{O_2}}{\partial t} + \mathbf{u}\cdot\nabla Y_{O_2}\right) = \nabla\cdot(\rho D_{O_2}\nabla Y_{O_2}) + \omega_{O_2}, \\ \rho\left(\dfrac{\partial Y_{CO_2}}{\partial t} + \mathbf{u}\cdot\nabla Y_{CO_2}\right) = \nabla\cdot(\rho D_{CO_2}\nabla Y_{CO_2}) + \omega_{CO_2}, \\ \rho\left(\dfrac{\partial Y_{H_2O}}{\partial t} + \mathbf{u}\cdot\nabla Y_{H_2O}\right) = \nabla\cdot(\rho D_{H_2O}\nabla Y_{H_2O}) + \omega_{H_2O}, \\ \\ Y_{N_2} = 1 - Y_{CH_4} - Y_{O_2} - Y_{CO_2} - Y_{H_2O}, \end{cases} \tag{5.12c}$$

with Dirichlet and Neumann boundary conditions

$$Y_\alpha = Y^D_\alpha \ \ \ on\ \Gamma^D_Y, \qquad\qquad \rho D_\alpha\frac{\partial Y_\alpha}{\partial n} = 0 = 0 \ \ \ \Gamma^N_Y = \partial\Omega\backslash\Gamma^D_Y,$$

and initial conditions for the mass fractions $Y_\alpha(t=0)$.

- State equation for the fluid

$$\rho = \frac{M}{T}\frac{P_0}{R}, \tag{5.12d}$$

where $1/M = \sum_\alpha Y_\alpha/M_\alpha$ is the mean molecular mass, $P_0 = 101325\ (Pa)$ in the near isobaric limit is the ambient pressure and $R = 8.314\ (J/molK)$ is the universal gas constant.

$$\frac{1}{M} = \left[\frac{Y_{CH_4}}{16} + \frac{Y_{O_2}}{32} + \frac{Y_{CO_2}}{44} + \frac{Y_{H_2O}}{18} + \frac{1 - Y_{CH_4} - Y_{O_2} - Y_{CO_2} - Y_{H_2O}}{28}\right]\times 10^3 mol/Kg.$$

To make the model more realistic, the multi-component transport properties of methane-air mixtures are evaluated using the model proposed by Smooke & Giovangigli [64], where the Prandtl $\left(Pr = \dfrac{\mu}{\rho D_T}\right)$ and Lewis $\left(Le_\alpha = \dfrac{D_T}{D_\alpha}\right)$ numbers are assumed to be constant: $Pr = 0.75$, $Le_{O_2} = 1.11$, $Le_{CH_4} = 0.97$, $Le_{H_2O} = 0.83$, $Le_{CO_2} = 1.39$, and the dynamic diffusion coefficients (the viscosity μ, the thermal diffusivity ρD_T, the mass diffusivity of each species ρD_{CH_4}, ρD_{O_2}, ρD_{CO_2} and ρD_{H_2O}) depend upon the temperature according to the power law

$$\frac{\mu}{\mu_0} = \frac{\rho D_i}{(\rho D_i)_0} = \left(\frac{T}{T_0}\right)^\sigma \qquad \text{with } \sigma = 0.7, \tag{5.13}$$

where μ_0, $(\rho D_i)_0$ and T_0 denote a reference or initial value for the dynamic diffusion coefficients in Ω.

$$\rho_A D_{TA} = 2.58 \times 10^{-5} \frac{Kg}{m \cdot s} \quad \text{with } T_0 = 300K.$$

A burning process of typical hydrocarbon in air involves dozens of chemical species and hundreds of elementary chemical reactions. A detailed account of the chemistry is necessary to describe the production of pollutants such as carbon monoxide and oxides of nitrogen; however, many aspects of the combustion process can be understood by assuming that chemical reactions between the fuel and the oxygen of the air takes place in a single overall step. The latter is the approach adopted in this work. Thus, we consider that the fuel CH_4 reacts with the oxygen of the air O_2 to generate combustion products according to the irreversible global reaction

$$CH_4 + 2O_2 \rightarrow CO_2 + 2H_2O + (q) \ \text{ (molar equation)}, \tag{5.14}$$

with a global rate of the form shown recently in [40]

$$\omega = BC_{CH_4}C_{O_2}e^{-Ta/T},$$

where C_α is the molar concentration of the species α. If we write the reaction rate using the mass fraction of the species Y_α, then we will have the following formula

$$\omega = \rho^2 B \frac{Y_{CH_4}}{M_{CH_4}} \frac{Y_{O_2}}{M_{O_2}} e^{-Ta/T}. \tag{5.15}$$

The amount of mass fraction of the species α per unit of time ω_α that appears in equations (5.12b) and (5.12c) can be written as

$$\omega_\alpha = \nu_\alpha M_\alpha \omega = \nu_\alpha M_\alpha \rho^2 B \frac{Y_{CH_4}}{M_{CH_4}} \frac{Y_{O_2}}{M_{O_2}} e^{-Ta/T}, \tag{5.16}$$

where ν_α is the molar stoichiometic coefficient for the species α in the global reaction (5.14). Thus the relation between the reaction rates for the different species is:

$$\begin{cases} \dfrac{\omega_{CH_4}}{M_{CH_4}} = \dfrac{\omega_{O_2}}{2 \cdot M_{O_2}} = -\dfrac{\omega_{CO_2}}{M_{CO_2}} = -\dfrac{\omega_{H_2O}}{2 \cdot M_{H_2O}}, \\ \omega_{CH_4} = \dfrac{16}{64}\omega_{O_2} = -\dfrac{16}{44}\omega_{CO_2} = -\dfrac{16}{36}\omega_{H_2O}. \end{cases}$$

Nonunity reaction orders, proposed for instance in [72], are known to add numerical stiffness to the computation [56], so that the present development assumes unity reaction orders for the fuel and oxidizer. With these assumptions, three different parameters are left in the model, namely, the preexponential factor B, the activation temperature Ta, and the amount of heat released per mole of fuel consumed q. It is seen in [40] that q and Ta have to be selected as functions of the local equivalence ratio ϕ to reproduce accurately propagation velocities of diluted and undiluted flames over the whole flammability limit.

1. B : We take the preexponential factor B as a constant parameter

$$B = 8.4 \times 10^8 m^3/(mol \cdot s).$$

2. $Ta(\phi)$: The model employs a variable activation temperature $Ta(\phi)$ to mimic changes in the underlying chemistry in rich and very lean flames.

$$\frac{Ta}{Ta_0} = \begin{cases} 1 + 8.250(0.64 - \phi)^2 & \phi < 0.64, \\ 1 & 0.64 \leq \phi \leq 1.07, \\ 1 + 1.443(\phi - 1.07)^2 & \phi > 1.07, \end{cases} \tag{5.17}$$

with $Ta_0 = 15900K$.

3. $q(\phi)$: Changes of q with ϕ need to be introduced in fuel rich combustion to describe the effect of partial fuel oxidation on the amount of heat released

$$q/q_0 = \begin{cases} 1 & \phi \leq 1, \\ 1 - 0.21(\phi - 1) & \phi < 1, \end{cases} \tag{5.18}$$

with q_0 being the molar heat of reaction that depends on the temperature. Here, the formation enthalpy $h_{f,\alpha}$ and heat capacity $c_{p,\alpha}$ of the species α, which are needed to calculate q_0 and c_p, are obtained in terms of the temperature T using NASA polynomial formulae [63]

$$\begin{cases} c_p = \sum_\alpha Y_\alpha c_{p,\alpha}, \\ q_0 = \sum_\alpha \nu_\alpha h_f. \end{cases}$$

In the temperature equation (5.12b) H_F represents the amount of heat released per unit mass of fuel consumed. It can be written in terms of the molar heat of reaction q and the molecular mass of methane M_{CH_4} as

$$H_F = q/M_{CH_4}.$$

The local equivalence ratio is defined in terms of the mass fractions of methane and oxygen in the upstream fresh mixture as $\phi = sY_{F,u}/Y_{O_2,u}$, where s is the mass of oxygen burnt per unit of fuel ($s = 4$ for the methane). When the Lewis numbers for fuel and oxidant are equal, the mixture fraction $Z = (SY_F/Y_{F,0} - Y_{O2}/Y_{O2,A} + 1)/(1 + S)$ is a conserved scalar and its value does not vary along the characteristic curves and across the flame. Note that the Lewis numbers of fuel and oxidizer in hydrocarbons-air mixtures are very close to each other, so we can consider that the equivalence ratio takes the simplified expression given by (5.19).

$$\begin{cases} \phi = S\dfrac{Z}{1 - Z}, \\ Z = \dfrac{SY_F/Y_{F,0} - Y_{O2}/Y_{O2,A} + 1}{1 + S}. \end{cases} \tag{5.19}$$

More thermochemistry parameters that play an important role in combustion are:

- The planar flame velocity S_L of the stoichiometric mixture. Experimentally, for methane CH_4 diluted with nitrogen and air the propagation velocity S_L is a function of the mass fraction $Y_{CH_4,0}$ of the feed stream. The Table 6.1 can be seen in [39]

$Y_{CH_4,0}$	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
$S_L(m/s)$	0.40	0.39	0.38	0.37	0.36	0.34	0.31	0.27	0.21	0.12

Table 6.1: Propagation velocity of a planar stoichiometric premixed flame as function of $Y_{CH_4,0}$.

- The overall air to fuel stoichiometric ratio $S = \dfrac{sY_{F,0}}{Y_{O_2,A}}$, defined in terms of the mass fractions of fuel $Y_{F,0}$ and oxygen $Y_{O_2,A}$ in the air feed streams. S is the mass of air that one needs to mix with the unit mass of fuel stream to generate a stoichiometric mixture.

- The stoichiometric flame temperature $T_s = T_0 + \dfrac{H_F}{c_p}\dfrac{Y_{F,0}}{1+S}$.

- The Zeldovich number $\beta = \dfrac{Ta_0(T_s - T_0)}{T_s^2}$, which is moderately large in combustion process.

To solve numerically the conservation equations (5.12a-5.12d) and draw conclusions of them is important to know the scales of the physical phenomena. Therefore, it is very useful to write the equations in non-dimensional form using the following scales:

- $\delta_L = D_{TA}/S_L$ as scale for the lengths.

- S_L as scale for the velocity.

- Therefore, the time scale will be D_{TA}/S_L^2.

- The density will be measured with the density of the air stream ρ_A.

- The concentration of fuel will be measured with $Y_{CH_4,0}$ and the concentration of Oxygen with $Y_{O_2,A}$.

- T_s as scale for the temperature of the fluid.

We do not write here the equations (5.12a-5.12d) in a dimensionless form because they can be obtain in a straightforward manner. We only point out the important change in the form of the dimensionless reaction rate given by the following expression

$$\widehat{\omega}_{CH_4} = \frac{\rho_A D_{TA}}{S_L^2 M_{O_2}} Y_{CH_4,0} Y_{O_2,A} \cdot B\widehat{\rho}^2 \widehat{Y}_{CH_4} \widehat{Y}_{O_2} e^{-Ta \cdot T_s/\widehat{T}}. \tag{5.20}$$

5.2 Numerical Method

The calculation of the numerical solution of (5.12a-5.12d) is a challenging problem for conventional numerical methods because in addition to the compressible Navier-Stokes equations, one has to deal with strong non linear convection-reaction-diffusion equations yielding phenomenon that exhibit a large disparity of both temporal and spatial scales**,** and a free surface represented by the flame front. Therefore, an efficient numerical approach to the problem should make use of adaptive methods both in time and space to properly capture the scales and the evolution of the flame fronts. We propose to solve the problem with the fully adaptive procedure designed in this Thesis based on the Dual Weighted Residual (DWR) technique, in particular, we use the theory presented in Chapter 4 for convection-diffusion-reaction equations.

Our method considers on one side the numerical solution of the Navier-Stokes equations and on the other side the solution of the equations for temperature and species, both equations will be discretizated with schemes that have a second order error in space and second order error in time.

- For the Navier-Stokes equations (5.12a) we use the semi-Lagrangian finite element method of [2] with the modification of the positive interpolation introduced in [15] to calculate the variables at the foot of the characteristics. Such method employs Taylor-Hood P_2/P_1 elements for space discretization and at each time step calculates the solution of the discrete Stokes problems by a Preconditioned Conjugate Gradient-Uzawa algorithm that was initially proposed by Cahouet and Charbard [26] and then developed by Glowinski and coworkers [22] and [29].

- We use convection-diffusion-reaction equations to model the evolution of the temperature and the chemical species (5.12b-5.12c) and with these equations will formulate our adaptive procedure and calculate the error in the form indicated in Chapter 4.

$$\begin{cases} \rho\left(\dfrac{\partial C}{\partial t} + \mathbf{u}\cdot\nabla C\right) = \nabla\cdot(\rho D_C \nabla C) + \rho f(C) \qquad in\ \Omega\times(t_{n-1},t_n], \\ \\ C(x,t_{n-1}) = c^{n-1}_{h\Delta t}(x) \qquad in\ \Omega, \\ \\ boundary\ conditions, \end{cases} \tag{5.21}$$

  for $C = T$, Y_{CH_4}, Y_{O_2}, Y_{H_2O} and Y_{CO_2}.

  The application of the semi-Lagrangian scheme to treat the convective terms leads to integrate along of the characteristic curves and calculate $\overline{c}^{n-1}_{h\Delta t}(x)$ and the convective error indicator $\eta^n_{convect}$. Then, the equation (5.21) is transformed in the following diffusion-reaction equation.

$$\begin{cases} \overline{\rho}\dfrac{\partial \overline{C}}{\partial t} = \nabla\cdot\left(\overline{\rho D_C}\nabla\overline{C}\right) + \overline{\rho} f(\overline{C}) \qquad in\ \Omega\times(t_{n-1},t_n], \\ \\ \overline{C}(x,t_{n-1}) = \overline{c}^{n-1}_{h\Delta t}(x) \qquad in\ \Omega, \\ \\ boundary\ conditions, \end{cases} \tag{5.22}$$

  and with this formulation we can calculate the diffusion-reaction error indicator $\eta^n_{diff-react}$.

  About the resolution of the primal problem in time we can comment two options:

1. The first one is to use the derived discretization obtained by applying the DWR method with cG(1)-dG(0) finite elements. We have already seen that this discretization leads to a Crank-Nicolson (CN) scheme in time. When equations have high stiffness the Newton solver for nonlinear equations included in the CN scheme may not reach the convergency easily and the time step size would be controlled by it. A property is not at all interesting.

2. Another possibility is to use Runge-Kutta-Chebychev (RKC) schemes (Explitic version [65] or a more recently Implicit-Explicit one [71]). RKC methods are designed for parabolic equations but we can see an extended explanation of RKC schemes in combination with the semi-Lagrangian method in the recent article of Bermejo and Carpio [17].

In our case, the resolution of the primal problem with the RKC scheme yields a solution with better features than those calculated with the CN scheme, furthermore, RKC is faster and more robust than the classical CN, it means that with RKC scheme we are able to use time step sizes bigger than with the CN scheme. In this example we have seen that the number of stages with the Explicit RKC scheme is always smaller than 50, this means that is not necessary to use the IMEX-RKC in this problem. This fact is due to the high refinement of the mesh that yields similar stiffness for the diffusion and the reaction terms, the use of IMEX-RKC does not improve the efficiency of the algorithm.

In all of the numerical examples we have to choose a functional controlling the error, so that we get a solution which can provide a faithful capture of the most relevant features of lifted flames. We have to reproduce with enough accuracy the flame front and in this special case of undiluted lifted flames the mixing layer upstream of the front flame. For these reasons, a good functional for this problem would be the product of the concentration of Fuel and Oxygen, that offers a proper mixing layer and gets a good resolution of the flame front, because the form of the functional is like the global reaction rate of our problem (see 5.15).

$$J(C) = \int_{\Omega} j(C^n) d\Omega = \int_{\Omega} Y^n_{CH_4} \cdot Y^n_{O_2} d\Omega$$

$J(C)$ is an integral over the whole domain, therefore the convection and diffusion-reaction functionals will have the same structure. Finally, the tolerance that we choose in the examples is $Tol_t = Tol_s = 1 \cdot 10^{-4}$.

It is important to comment that in the numerical tests we have performed the number of rejected time steps has been always smaller than 0.5% of the total, the most part of them being in the ignition instants. Another good property of the algorithm is that in the semi-Lagrangian part of it is very fast and consume few requirements of CPU time (see Figure 5.4). These features lead to have a fast and efficient fully adaptive algorithm.

A schematic presentation of the numerical algorithm to solve the problem of the lifted flame is the following

Space – time adaptive algorithm :

Given the tolerances Tol_s, Tol_t, the macro-triangulation $\mathbb{T}_{h,0}$ and the time step Δt_0.

Obtain an initial mesh $\mathbb{T}_h^0$ by local refinement and project the initial datum c^0 in that mesh as $c_{h\Delta t}^0 \in V_h^0$, where $c = T,\ Y_{CH_4},\ Y_{O_2},\ Y_{H_2O}$ and Y_{CO_2}.

Refine the initial mesh until $\eta_{s,convect}^0 < Tol_s$.

$n = 1$.

$\theta = 0.8$.
while $(t_{n-1} < T)$
{
$\mathbb{T}_{h,0}^n = \mathbb{T}_h^{n-1}$

$i = 1$.
while $(k == 0)$
{

1. Adapt the mesh $\mathbb{T}_{h,i-1}^n$ to generate a new mesh $\mathbb{T}_{h,i}^n$.
2. Solve the convective part with the semi-Lagrangian scheme and obtain $\overline{c}_{h\Delta t}^{n-1}$.
   Calculate both initial errors $\eta_{s,convect}^n$ and $\eta_{t,convect}^n$.
   Using successive refinements to obtain a new mesh $\mathbb{T}_{h,i}^n$ with $\eta_{s,convect}^n < \theta \cdot Tol_s$.
3. Solve the diffusion-reaction equation and obtain the solution $c_{h\Delta t,i}^n$ on the mesh $\mathbb{T}_{h,i}^n$.
   Calculate the error indicator with DWR technique in time $\eta_{t,diff-react}^n$ and in space $\eta_{s,diff-react}^n$.
4. Calculate the total error indicator: $\eta_t^n = \eta_{t,convect}^n + \eta_{t,diff-react}^n \quad \eta_s^n = \eta_{s,convect}^n + \eta_{s,diff-react}^n$.
   and $\theta = \max\left(0.2, 0.8\dfrac{\eta_{s,convect}^n}{\eta_s^n}\right)$.
5. Select elements to be refined or coarsen with formulae (2.14a) and (2.14b).

If $[(\eta_s^n > Tol_s)\ and\ (\eta_t^n \leq Tol_t)]$
Reject step and maintain the time step length Δt_n. $k = 0$ and $i = i + 1$.
Else
Calculate a new time step Δt_{new} with formula (2.15b).
if $(\eta_t^n > Tol_t)$
$t_n = t_{n-1} + \Delta t_{new}$. Reject step: $k = 0$ and $i = i + 1$.
else
$t_{n+1} = t_n + \Delta t_{new}$ and $k = 1$.
end if
End if

$c_{h\Delta t}^n = c_{h\Delta t,i}^n$.

Calculate $\rho^n(c_{h\Delta t}^n)$ by using the state equation (5.12d).

Calculate the velocity of the compressible flow $\mathbf{u}_{h\Delta t}^n$ at low Mach numbers (5.12a).

$\mathbb{T}_h^n = \mathbb{T}_{h,i}^n$.
}

Note that θ is a parameter between 0.2 and 0.8 defined as $\theta = \max\left(0.2, 0.8\dfrac{\eta_{s,convect}^n}{\eta_s^n}\right)$. We use this parameter to refine the spatial mesh in the semi-Lagrangian stage until the criterion

$\eta^n_{s,convect} < \theta \cdot Tol_s$ is satisfied. With this idea, the index i in the internal loop of the algorithm, takes the value 1 on most occasions, therefore, the number of time steps rejected is negligible.

5.3 Numerical experiments

In this section we shall discuss the numerical results obtained by the integration of the general equations (5.12a-5.12d). We shall consider the ambient temperature $T_0 = 300K$ as the initial temperature in the domain; in this case, the spontaneous ignition time is several times larger than the characteristic residence time and therefore, the spontaneous flame ignition phenomenon will not likely play a significant role in the structure and dynamics of attached/lifted flames. As far as our model is concerned, the ignition will occur by means of an external source located anywhere within the mixing layer. The so-generated flame, which at early times may be spherical, will quickly grow elongated by the flow.

The examples shown below are given for methane CH_4, although they are expected to be similar for other hydrocarbons: we only need to change physical coefficients and the chemical reaction as explained in [40]. In the examples we consider a mass fraction of feed air stream $Y_{O_2,A} = 0.23$ and different concentrations of the feed fuel stream $Y_{CH_4,0}$ diluted with nitrogen. For the concentrations $Y_{CH_4,0} = 0.1$, 0.2 and 1 we can calculate the stoichiometric ratio $S = \dfrac{4Y_{CH_4,0}}{Y_{O_2,A}}$, the adiabatic flame temperature $T_s = T_0 + \dfrac{H_f}{c_p}\dfrac{Y_{CH_4,0}}{1+S}$ and the Zeldovich number $\beta = \dfrac{Ta_0(T_s - T_0)}{T_s^2}$.

	S	$T_s(K)$	β
$Y_{CH_4,0} = 0.1$	1.739	1797	7.37
$Y_{CH_4,0} = 0.2$	3.478	2131	6.41
$Y_{CH_4,0} = 1$	17.39	2530	5.54

Next, we shall show typical examples, namely the mixing layer and the planar jet, so as to evaluate the reliability of the numerical scheme.

- **Mixing layer:**

  We shall assume that the feed streams enter in the computational domain normal to the porous wall: above the separation line $(y = 0)$, the fuel stream flows with uniform velocity U_0 and below, the air stream also has uniform velocity U_A. The wall, located at $x = 0$, extends infinitely in the vertical direction, and the two streams are separated at $x < 0$ by an infinitely thin splitter plate located at $y = 0$. The feed streams are considered to possess fixed velocity, temperature and chemical composition, whereas the other boundaries of the computational domain are located far away from the front flame; for $x \longrightarrow \infty$ and $|y| \longrightarrow \infty$ we assume free stress $\mu\frac{\partial \mathbf{u}}{\partial n} - p\mathbf{n} = \mathbf{0}$ and zero heat and mass flux diffusion $\left(\rho D_T \frac{\partial T}{\partial n}=0 \text{ and } \rho D_\alpha \frac{Y_\alpha}{\partial n}=0\right)$. We consider here two different cases depending on the velocity feed streams, the first situation being when the velocities of both fuel and air are equal $U_0 = U_A = U$. This situation has been thoroughly studied and will allow us to compare our results with those obtained by other authors, e.g. [39]. As for the more interesting second problem, the different velocities $U_0 > U_A$, will likely produce instabilities in the flow, so that an adaptive temporal algorithm to solve efficiently the problem is required. A schematic illustration of the two-dimensional reacting mixing layer is shown on the left side panel of Figure 5.2.

- **Planar jet:**

  We shall assume that the fuel feed stream enters in the computational domain normal to an injector of width $2a$, with uniform velocity U_0; the air streams emerge from porous walls located above and below the injector, with uniform velocity U_A. Notice that we are dealing with the 2D or planar configuration, thus any coaxial or axilsymmetric considerations are not relevant in this case. The boundary conditions are identical to those considered in the mixing layer problem: fixed velocity, temperature and chemical composition on the porous wall and injector; free stress and negligible heat and mass diffusion flux for the other boundaries sufficiently apart from the fuel feed stream. Likewise, the two situations to be studied next are: equal velocities $U_0 = U_A = U$ and different velocities $U_0 > U_A$ for the fuel and air feed streams. Since neither case has been numerically solved as yet, it is a good opportunity to illustrate these phenomena with some significant data. A sketch of this configuration can be observed on the right panel of Figure 5.2.

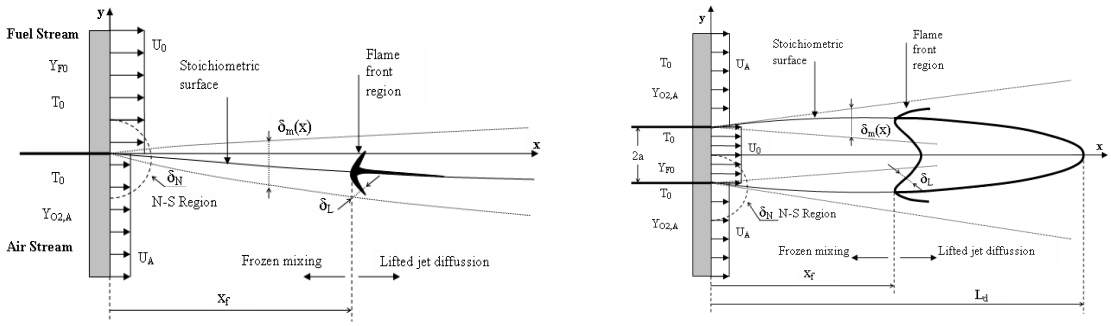

Figure 5.2: Mixing layer and planar jet configuration, showing the length scales and the upstream boundary conditions.

5.3.1 Mixing layer for $U_0 = U_A = U$

We shall consider in this example equal velocities for both feed streams $U_0 = U_A = U$. All results will be compared with those provided by the literature on lifted flames and will serve to highlight good properties of the proposed numerical algorithm.

In Figure 5.3 we show the main features of the steady solution obtained for $Y_{CH4,0} = 0.2$ and velocity $U/S_L = 2$. The upper left panel depicts the final mesh and a detail near the flame front. The upper right panel shows the stoichiometric curve and the isolines of the reaction rate; the attached zoom of the flame front presents the temperature T in thick solid line, the oxygen mass fraction Y_{O_2} in normal-size solid line and the flow streamlines in dashed lines. The lower left figure plots the temperature of the flame. Finally, the lower right panel displays the module of the velocity along the stoichiometric curve.

The key feature in these pictures is that the maximum temperature is reached along the stoichiometric curve, very close to T_s. The heat-releasing effect in the flame front causes the flow to be redirected and therefore the triple-flame front adopts a curved shape. The module of the velocity along the stoichiometric line decreases from its maximum value far upstream from the triple flame, to a minimum value close to that of the planar laminar flame speed when we are very near upstream from the flame. This has proved to be in agreement with the observations [57]. According to this last characteristic, we obtain lifted flame regimes for values of the feed stream velocities U_A and U_0 above the stoichiometric flame velocity S_L, as noted earlier.

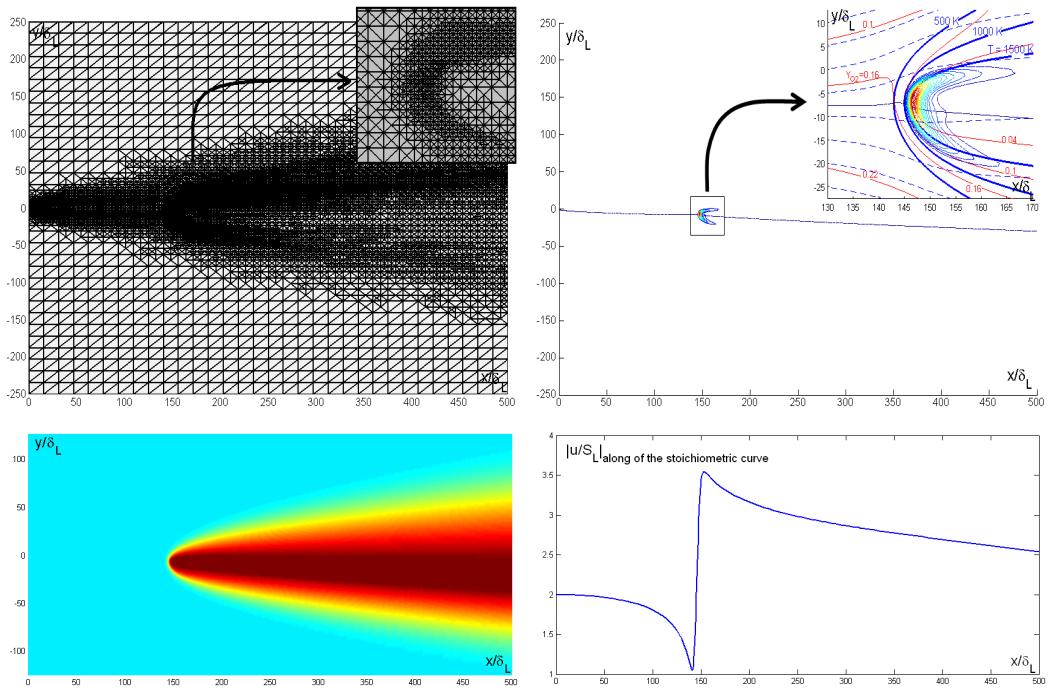

Figure 5.3: Figures of the steady solution obtained for $Y_{CH_4,0} = 0.2$ and velocity $U/S_L = 2$.

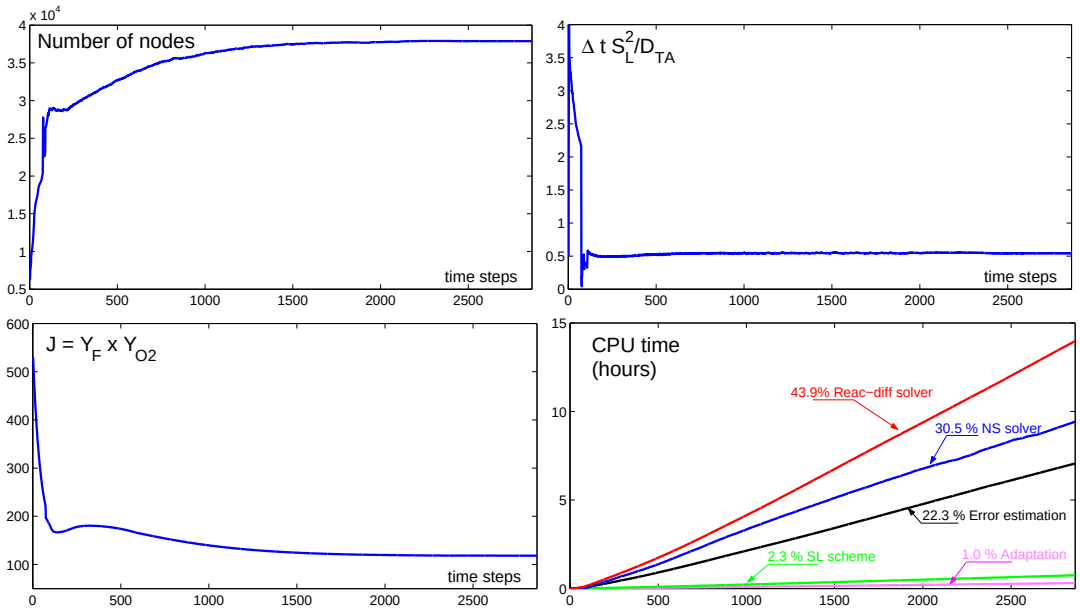

Figure 5.4: Figures of the behabiour of our fully adaptive procedure for $Y_{CH_4,0} = 0.2$ and velocity $U/S_L = 2$.

As for the numerical scheme, we show in Figure 5.4 some of the parameters involved in the numerical algorithm, such as the number of nodes, the time step size, the value of the functional J and the CPU time. We can observe that as the ignition is sparked off, the time step size experiences an abrupt drop and the number of nodes suddenly increases. On the other hand, is interesting to note that both the number of nodes and the time step size are stabilized when the value of the functional reaches a stationary state, after approximately 2000 time steps.

One significant plot could be that showing the CPU time required in each part of the algorithm: as one can see, the adaptation procedure and the semi-Lagrangian part of the scheme only need 1% and 2.3% of the total time, respectively; the computation of the temporal and spatial error estimator, which means to solve the dual linear problem and then to perform the post-processing of the numerical solution takes a 22.3% of the total; a 30.5% is employed in the solution of the NS equations; and finally the solution of the reaction-diffusion equations for the species and temperature consumes the remaining 43.9%.

In order to validate the solution calculated by our numerical method, we have compared our results with those provided by Fernández-Tarrazo, Vera and Liñan [39], in which asymptotic techniques and a non-adaptive finite difference scheme have been used. As a side note, we can remark that the reaction rate employed in [39] is somewhat different to that proposed in a more recent paper [40], the latter being also the one used in the present work. A comparison is shown in Figures 5.5 and 5.6.

The triple flame configuration moves upstream and downstream along the stoichiometric surface and takes different positions in the combustion chamber as a function of the injection velocities and concentration of the fuel feed stream. When the ratio U/S_L is small, the diffusion flame is anchored in the Navier-Stokes region close to the injector; as the ratio U/S_L grows, the flame is lifted to a distance x_f, being eventually blown off and leaving the domain for values of U/S_L exceeding of some critical value. This behaviour is shown in the Figure 5.5, where we have plotted the variation of the dimensionless lift-off distance x_f/δ_L with the injection velocity ratio U/S_L for different values of the diluted fuel ($Y_{CH_4,0} = 0.1$, $Y_{CH_4,0} = 0.2$ and $Y_{CH_4,0} = 1$). Our numerical examples have been represented with circles, and with solid line the results obtained by [39]: we can observe that there is a good agreement for values of $Y_{CH_4,0} \geqslant 0.2$, although remarkable differences arise for $Y_{CH_4,0} = 0.1$.

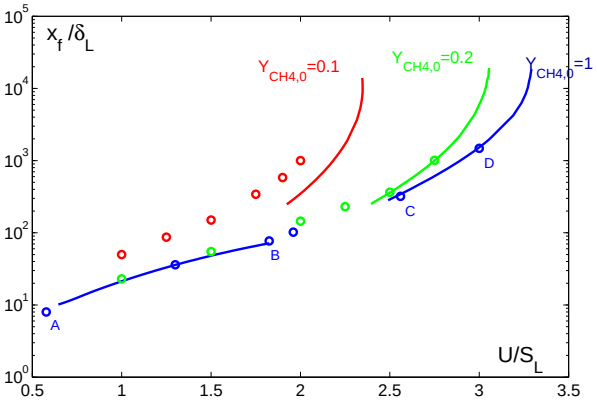

Figure 5.5: Lifted length x_f/δ_L versus velocity U/S_L. 'o' results obtained with our adaptive method and continuous line results of [39].

Figure 5.6 shows the flame front structure obtained for undiluted feed fuel stream ($Y_{CH_4,0} = 1$) and different values of the velocity ratio $U/S_L$and their related Damköhler number $D_m = (\delta_m/\delta_L)^2$. The plots correspond to the points A, B, C and D of Figure 5.5. The different pictures of Figure 5.6 illustrate the evolution of the flame front, ranging from an edge flame for sufficiently small values of D_m to a thin premixed flame front with trailing diffusion for $D_m \longrightarrow \infty$. In Figure 5.6 the coloured images have been created by Fernandez-Tarrazo, Vera and Liñan in [39], whereas the background mesh images belong to our numerical method. Again, the similarities are remarkable.

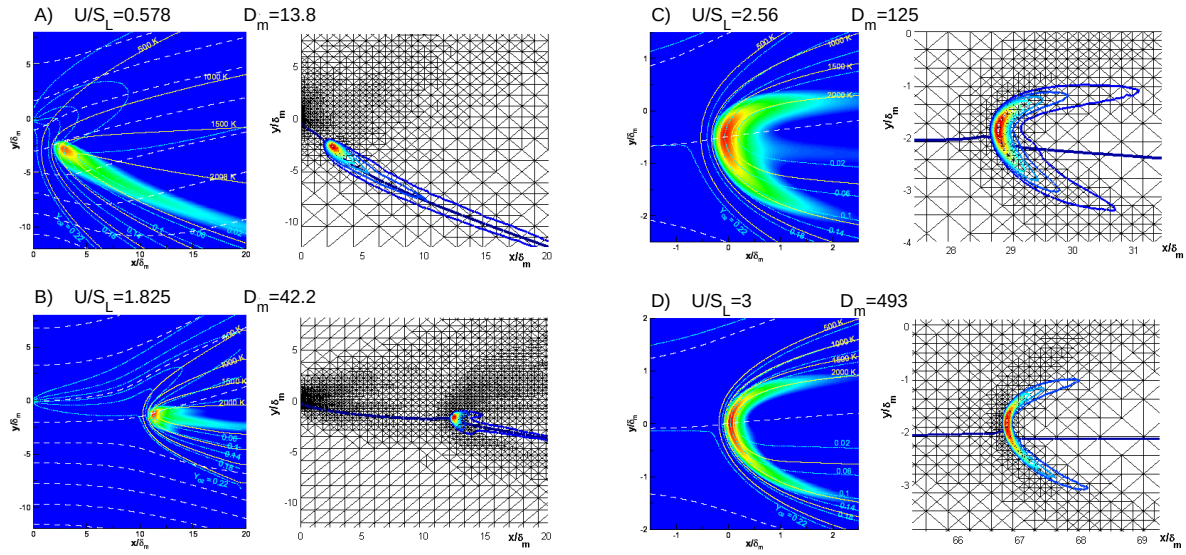

Figure 5.6: Flame front for different velocities U/S_L. Background mesh images obtained with our adaptive method and colored images by [39].

For $\beta >> 1$, the transition from edge flames to triple flames starts when the Damköhler number based on the effective size δ_m/β of the flame front region goes beyond 1. Notice that $\delta_m/(\beta\delta_L)$ is also the ratio of the burning rate per unit flame surface in the premixed and diffusion flames, so that for large values of the Damköhler number, the contribution of the burning rate in the diffusion flame ceases to be visible. In the case of undiluted methane CH_4, the transition from edge flame to triple flame is expected to occur around $D_m = (\delta_m/\delta_L)^2 \sim \beta^2 = 31$, situation which can be observed in Figure 5.6.

5.3.2 Mixing layer for $U_0 > U_A$

In the previous example, both feed streams (fuel and air) shared the same velocity $U_A = U_0$. Now, we shall consider two parallel streams with different velocities $U_0 > U_A$, separated by a thin plate, which come together when the plate ends. This situation is the so-called 'shear layer', which is technologically relevant because of two main reasons: first, the shear layer produces noise and dissipation of energy and, thus, amounts to a drag source; second, when two different fluids are present in the streams (as it happens in our example), the mixing of these fluids takes place in the shear layer. Since shear layers appear also in well known flows, such as jets and separation regions, most industrial combustors and many chemical mixers are designed with them in mind.

The main feature of this new configuration for large Reynolds numbers is that the mixing layer becomes a thin, two-dimensional vorticity layer linearly unstable, showing Kelvin-Helmholtz instabilities, for values of the ratio U_0/U_A exceeding of a certain critical value. The resulting mixing layers have adopted a corrugated shape due to the surrounding flow, which further enhances the mixing process. However, a stable configuration is attainable when a flame is present in the mixing layer, and this is so because of the expansion of the fluid and the increase of the diffusion coefficients caused by temperature. Figure 5.7 reflects this situation, with velocities $U_0/S_L = 10$ and $U_A/S_L = 1$, and the concentration of fuel is $Y_{CH_4} = 0.2$.

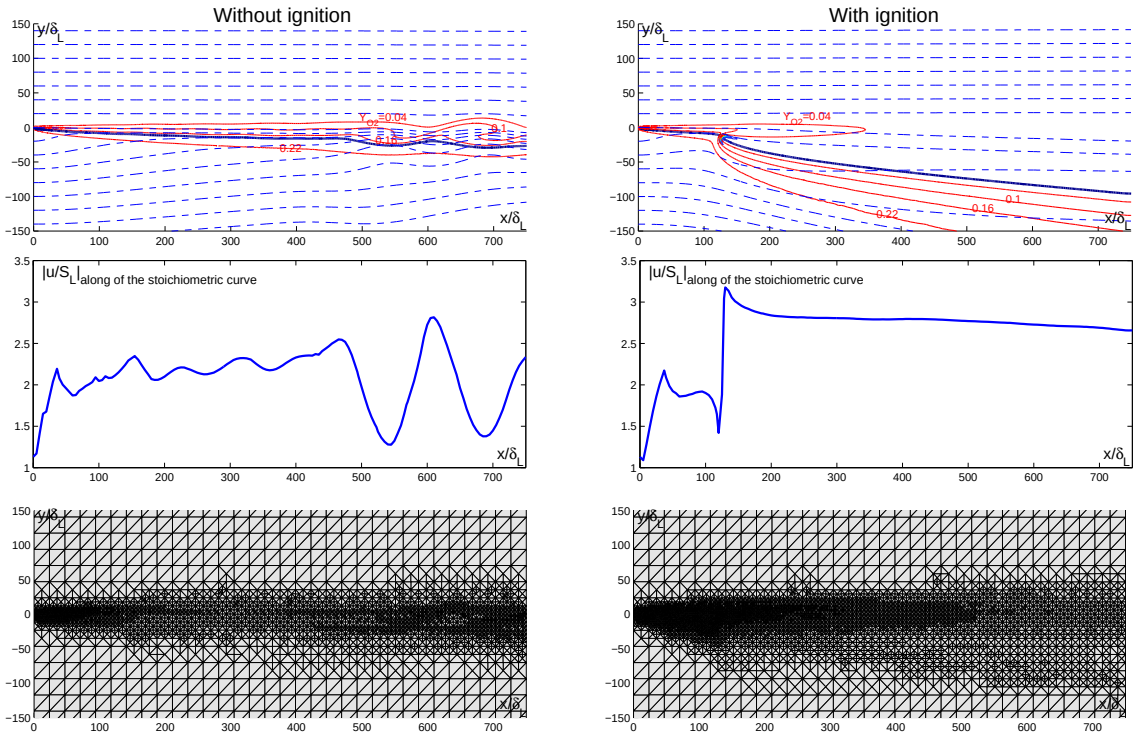

Figure 5.7: Mixing layer with a fuel stream $U_0/S_L = 10$ and air stream $U_A/S_L = 1$ for a concentration $Y_{CH_4} = 0.2$. We have on left panel figures without ignition and on the right panel figures with ignition.

The left side of Figure 5.7 shows pictures of a mixing layer where the ignition is not provoked, whereas the right side of the same figure reveals a lifted flame. For both sides, the topmost pictures show, in dashed lines, the streamlines of the flow; in thick solid line, the stoichiometric curve; and in thin solid lines, the isolines of the concentration of oxygen. The middle row plots the module of the velocity along the stoichiometric curve. Finally, the bottom pictures present the refined mesh of the numerical experiments. The solution on the right side, obtained with ignition, is a steady configuration. However, the solution without ignition on the left side of the picture has an oscillatory behaviour which can be easily observed in the corrugated form of the isolines of Y_{O_2} as well as in the oscillatory character of the velocity along of stoichiometric curve. For the example, without ignition the oscillation of the velocity close to the wall grows along the stoichiometric curve, whereas in the case with ignition the initial oscillations of the velocity drastically drop behind the front flame, so that the flame delays the oscillatory character of the solution.

In these examples the triple-flame propagation depends also on the velocity ratio U_0/U_A. Due to the velocity ratio the flame front is moved to the slower zone (in our case, the air zone) and therefore we can obtain a lifted flame for velocities of the feed fuel stream higher than the cases with $U_0/U_A = 1$. We have also seen that an expansion of the fluid and an increase of the diffusion coefficients by the heat releasing contribute to the stabilization of the mixing layer and, therefore, steady flames with high ratio U_0/U_A are obtained. However, for higher values of the ratio U_0/U_A, instabilities appear in the flow and the flame presents an oscillatory behaviour, Furthermore, if the instabilities of the flow are important just upstream of the flame front, the flame will be blown-off. In Figure 5.8 we have a steady flame on the left side with $U_0/U_A = 10$ and, on the right side, an oscillatory (near of the blown-off) flame with $U_0/U_A = 12$. In these

examples we have considered a concentration $Y_{CH_4} = 0.2$ and a ratio of velocities $U_A/S_L = 1$. In Figure 5.8 we have plotted the temperature and a detail of the front flame (with isolines for temperature, concentration of oxygen and streamlines of the flow).

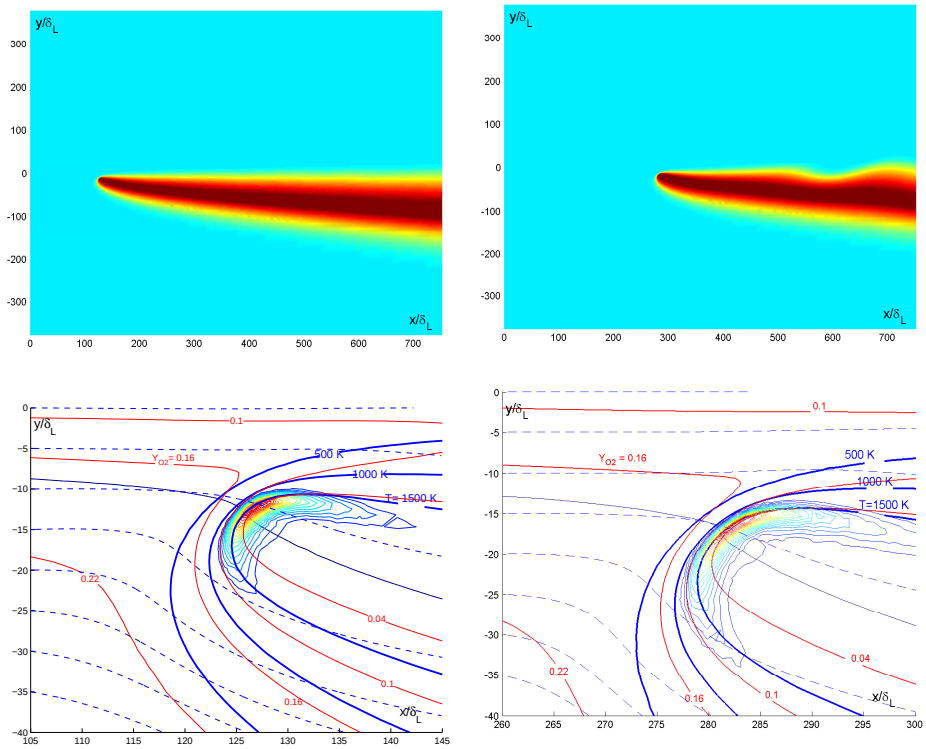

Figure 5.8: Temperature (top) and a detail of the front flame (bottom) for a mixing layer with $Y_{CH_4} = 0.2$ and $U_A/S_L = 1$. On the left panel $U_0/U_A = 10$ and on the right panel $U_0/U_A = 12$.

Due to the observed flow instabilities arising from this particular example of the mixing layer with different velocities $U_0 > U_A$, no stationary equations can properly describe the phenomenon. Thus, a time-dependent scheme such as the one proposed in this work is required to obtain an accurate description of the real solution. In Figure 5.9, we plot the evolution in number of nodes and time step size for the two different flame configurations displayed in Figure 5.8.

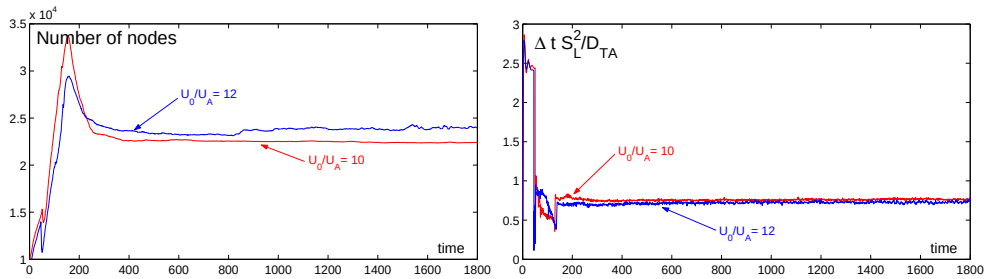

Figure 5.9: Evolution of the number of nodes and the time step size in a mixing layer with $Y_{CH_4} = 0.2$ for different gradients of velocities $U_0/U_A = 10$ and $U_0/U_A = 12$.

5.3.3 Planar jet for $U_0 = U_A = U$

In a planar jet the fuel feed stream enters in the computational domain normal to an injector of width $2a$, with uniform velocity U_0; the air streams emerge from porous walls located above and below the injector with uniform velocity U_A. In these examples we consider that both the fuel and air velocities on the porous walls are equal, that is, $U_0 = U_A = U$. The aim of this example is to observe the diffusion flame behaviour when there are two mixing layers and, therefore, the two flame fronts interfere.

When the Reynolds number $Re_a = aU/D_{TA} = (a/\delta_L \cdot U/S_L)$ and the stoichiometric ratio S are large compared with unity, the jet is slender with a developing length $L_d \sim SaRe_a$ and the lift-off of the flame must take place for distances $x_f < L_d$, otherwise the flame will be blown-off. When we provoke the ignition, symmetrically in both mixing layers, the flame fronts move together and reach an apparent symmetric steady solution that can be observed on the left part of Figure 5.10. But this situation is not stable and the interaction of the flames breaks the symmetry of the problem and an asymmetric steady solution is reached. We can observed this phenomena on the right side of Figure 5.10 with its respective refined mesh. We obtain two different asymmetric (but with mirror symmetry on the x-axis) situations that depend on the character of the perturbations as time integration progresses.

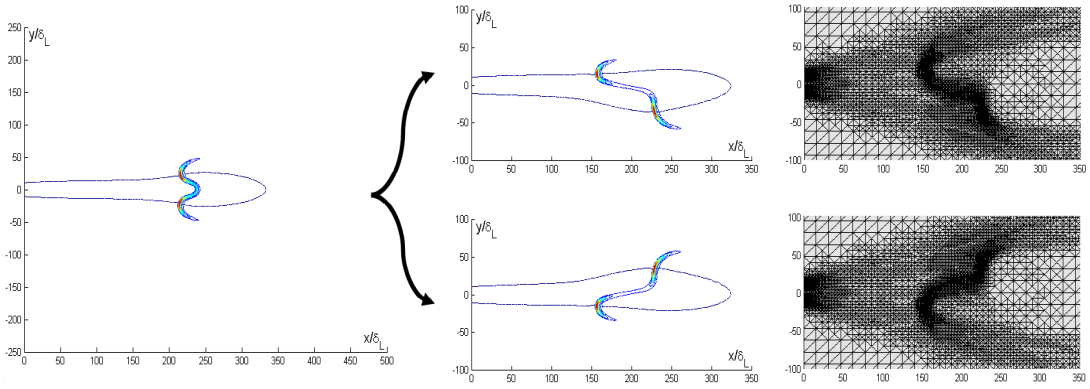

Figure 5.10: Symmetry breaking of two stable solutions with $Y_{CH_4} = 0.1$ and $U/S_L = 1.5$.

Figure 5.11: Experimental photograph of a coaxial flame, by Pablo Martínez and Jean-Marie Truffaut.

In the example of Figure 5.10 we have used diluted methane CH_4 ($Y_{CH_4} = 0.1$), an injector with semi-width $a = 10\delta_L$ and velocity $U/S_L = 1.5$, but we have similar situations with different concentrations, velocities and semi-widths of injector as long as we will have a lifted jet flame configuration with the front flames sufficiently close to each other. This phenomena has been observed in the laboratory and in Figure 5.11 we show an experimental photograph

taken by Pablo Martínez Legazpi and Jean-Marie Truffaut for a coaxial jet of methane. We have not experimental results for planar jet since its configuration is more difficult to obtain in a laboratory, but we expect similar experimental results in planar jet flames as we have observed in coaxial jet configurations.

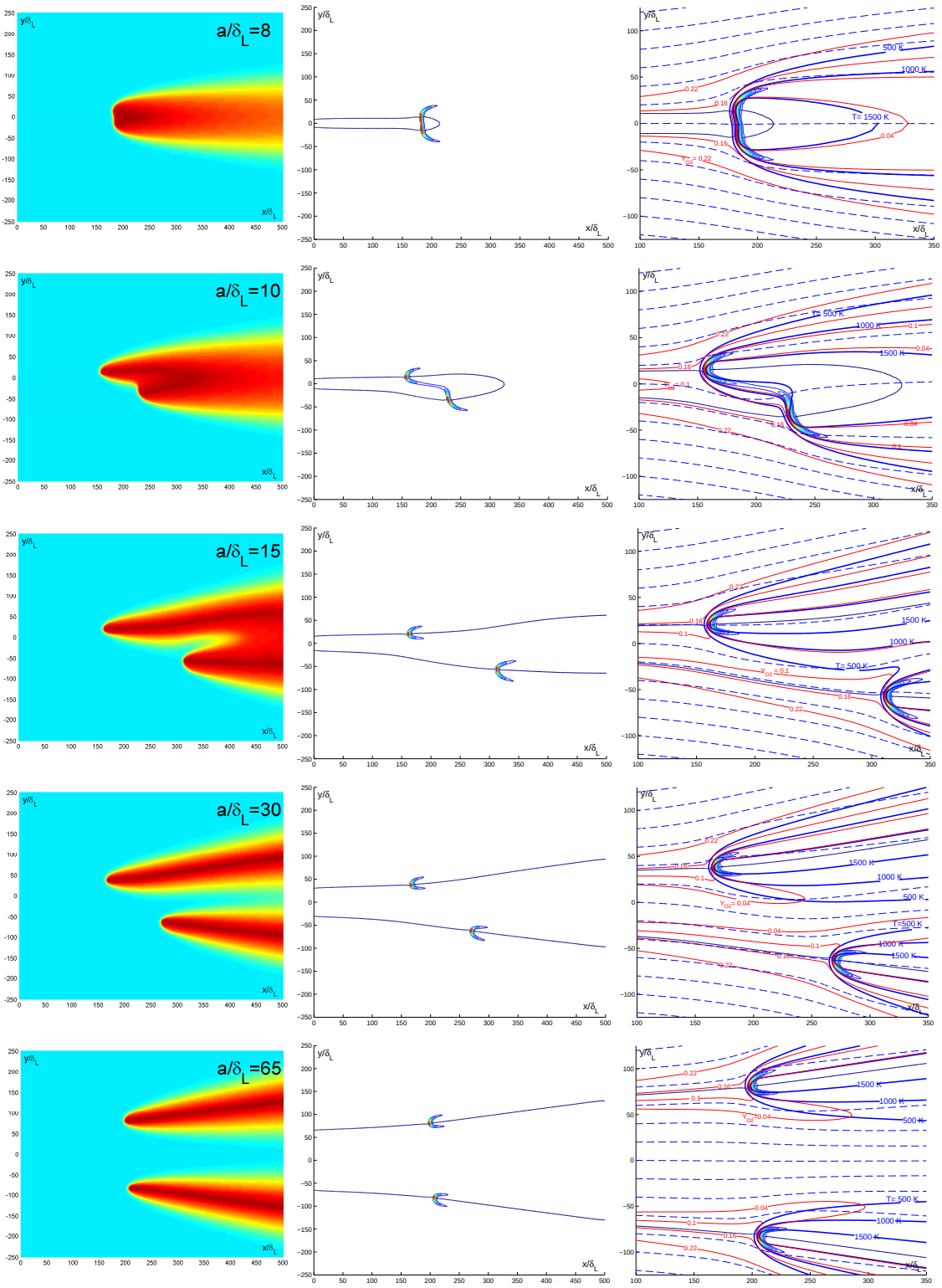

Figure 5.12: Lifted flames with $Y_{CH_4} = 0.1$ and $U/S_L = 1.5$ for different values of $a = [8, 10, 15, 30, 65]\delta_L$.

As we have said above in the mixing layer example, the effect of heat release in the flame front causes the flow redirection, therefore a redirection of the flow caused by the first flame front provokes an alteration of the flow just upstream of the second flame front. This effect produces that the first front goes forward and the second one backward until they reach a steady asymmetric solution. In Figure 5.12 we maintain constant both the concentration of fuel Y_{CH_4} and the velocity $U/S_L = 1.5$; then, we run several numerical examples for different values of the semi-width of the injector a. The lower limiting value for a is that of the example in which there is only a flame front at the end of the stoichiometric line ($x_f \lesssim L_d$). By successive increments in a, the flame fronts in both mixing layers get separated until the symmetric configuration is retrieved again for values of a large enough.

The most relevant aspects in these pictures are the following. In the limit of $x_f \lesssim L_d$ we have an almost symmetric configuration, because the zone of interaction of the flame fronts is planar and it does not produce the redirection of the flow. When the semi-width of the injector a increases, the steady solution of the problem takes an asymmetric configuration and the separation of the fronts $|x_{f1} - x_{f2}|$ reaches a maximum. If we continue increasing the length a the symmetric steady solution is recovered, because the interference between both flames is negligible. The effect of the redirection of the flow is negligible as can be observed in the last picture, and we obtain the configuration of two almost independent mixing layers. Figure 5.13 shows the lifted length x_{f1} (with circles) and x_{f2} (with asterisk) of the flame fronts as function of the dimensionless parameter a/δ_L.

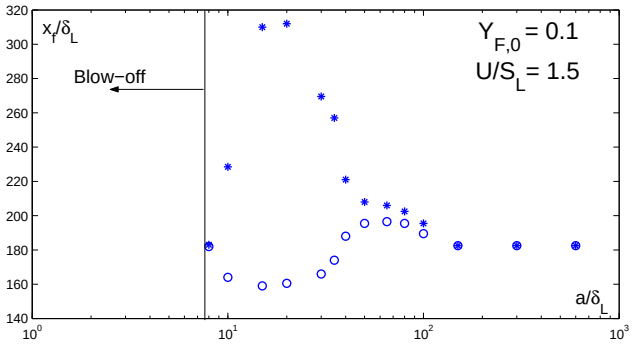

Figure 5.13: Lifted length of both flame fronts x_{f1} and x_{f2} as function of the a/δ_L.

To obtain this kind of asymmetric solution is necessary to calculate the time-dependent solution of the general equations. Should we wanted to obtain the steady solution by using the stationary equations of the problem, we would obtain only the symmetric solution, which proves to be unstable. Again, our time-space adaptive numerical algorithm shows its capabilities to provide a valid solution of the complex lifted flame problem.

5.3.4 Planar jet for $U_0 > U_A$

We shall consider in this example a planar jet with different velocities of the fuel U_0 and air feed streams U_A, with $U_0/U_A > 1$. Therefore, in this example we have two shear layers that interfere, a situation very common in technological applications. In this case, we hope to observe similar features as in the mixing layer with $U_0 > U_A$; we are also interesting in its symmetry properties.

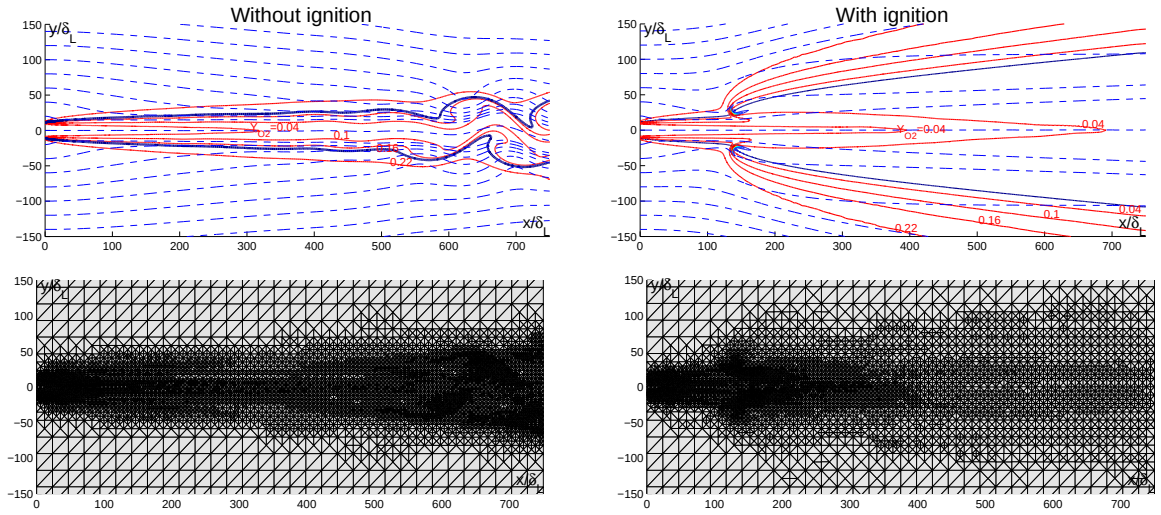

Figure 5.14: Planar jet with a fuel stream $U_0/S_L = 10$ and air stream $U_A/S_L = 1$ for a concentration $Y_{CH_4} = 0.2$. We have on left panel figures without ignition and on the right panel figures with ignition.

Again, we see in Figure 5.14 as the effect of the heat release makes the flame configuration stable. For velocities $U_0 = 10$ and $U_A = 1$ we have plotted the solution without ignition on the left side and the solution with ignition on the right side. We observe the same properties as in Figure 5.7 for the mixing layer. In the case without ignition, the resulting mixing layers are corrugated by the flow. However, when we provoke the ignition, the flame gets the stabilization of the flow downstream of the flame front and a steady solution for the streamlines, the isolines of oxygen concentration and the temperature in the flow is obtained.

Once more we obtain lifted flames with high values for the velocity of the feed fuel stream U_0 if the velocity of the air U_A is small because the front flame is moved to the slower zone. We have seen that the temperature produces the stabilization of the fluid downstream of the flame front. Another important feature is that the symmetry of the flames is maintained because the velocity in the fuel stream is too high and the redirection of the velocity caused by the flame front is negligible as compared with the unperturbed fuel stream velocity. If we continue increasing the ratio U_0/U_A, instabilities behind the flame front may appear, where we will find an oscillatory solution and flame fronts that advance and recede from the injector rim. When the instabilities are high just upstream of the flame fronts, the flame will be blown-off. In Figure 5.15 we have a steady flame on the left panel with $U_0/U_A = 10$ and an oscillatory and almost blown-off flame on the right panel with $U_0/U_A = 12$, always with concentration $Y_{CH_4} = 0.2$ and $U_A = 1$. In this figure we have plotted the temperature and a detail of the front flame (with isolines of temperature, concentration of oxygen and streamlines of the flow). The flame on the left is stable but on the right it shows an oscillatory character that changes with time.

In the global map of the temperature in Figure 5.15, for the case $U_0/U_A = 12$ appears the typical instability observed in the planar jet without ignition, and if we perform a long integration in time, the oscillation grows, the positions of the flame fronts travel far from the injector rim and the highly corrugated flame tends to leave the domain. In Figure 5.16 we plot the evolution in number of nodes and time step size for the two different flames displayed in Figure 5.15. We can notice the highly oscillatory character of the solution when $U_0/U_A = 12$ in the size of the time step, as well as the extremely corrugated shape of the mixing layer in the

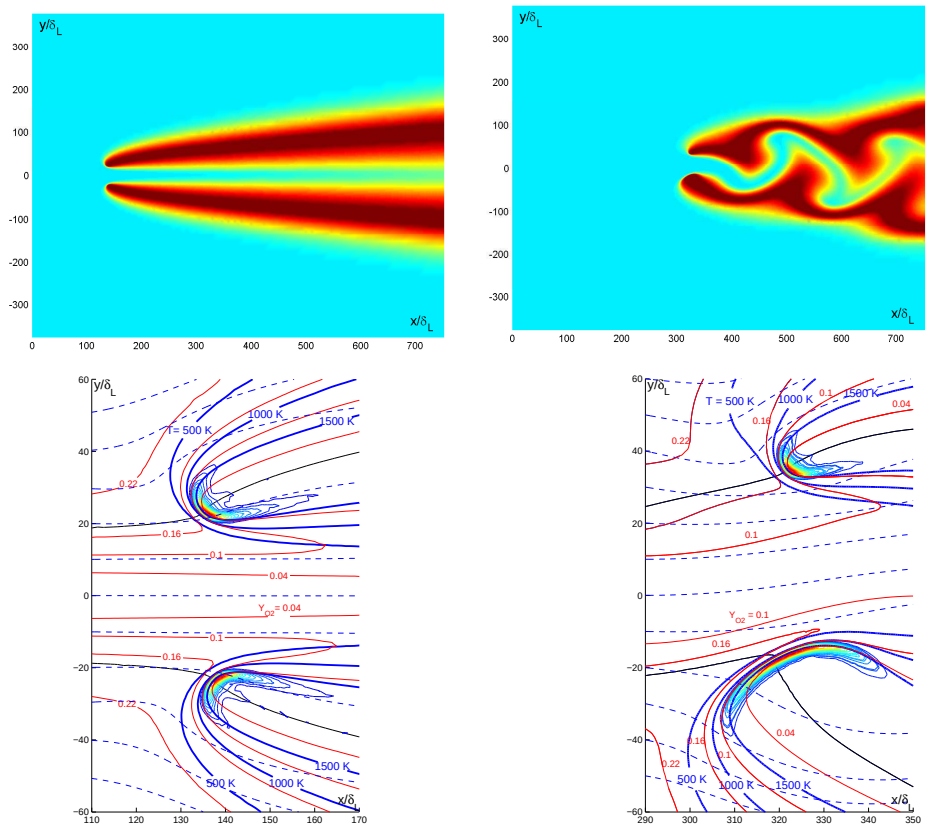

Figure 5.15: Temperature (top) and a detail of the front flame (bottom) for a planar jet with $Y_{CH_4} = 0.2$ and $U_A/S_L = 1$. On the left panel $U_0/U_A = 10$ and on the right panel $U_0/U_A = 12$.

increase of the number of nodes.

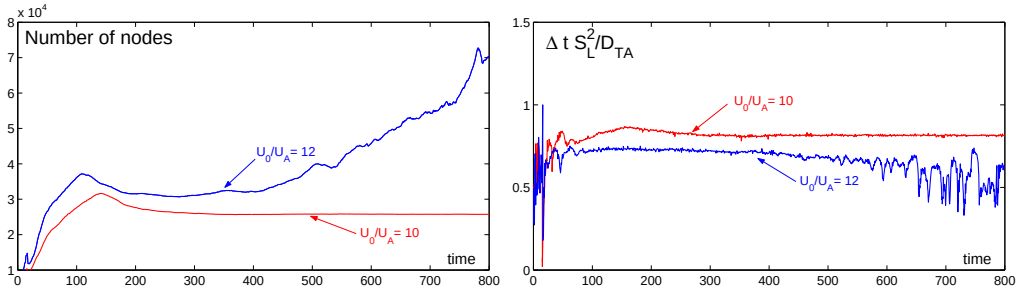

Figure 5.16: Evolution of the number of nodes and the time step size in a planar jet with $Y_{CH_4} = 0.2$ for different gradients of velocities $U_0/U_A = 10$ and $U_0/U_A = 12$.

Chapter 6

CONCLUSION AND FUTURE WORK

In this thesis we have presented a space-time adaptive method to calculate, in an efficient manner, the numerical solution of PDE's specially in time-dependent convection-diffusion-reaction equations. Our method is formulated in the context of finite elements, on unstructured 2D triangular meshes and the calculation of the a posteriori error is based on the philosophy of the *Dual Weighted Residual* (DWR) methodology for goal-oriented adaptation in arbitrary quantities of physical interest $J(u)$ of the solution.

The DWR technique yields efficient adaptive finite element algorithms for stationary problems. In our work, we have further developed this technique extending the ideas of the space post-processing to unstructured triangular meshes. However, in time-dependent problems, the application of the DWR method would require a large amount of computational resources. To palliate this issue, we have proposed in this thesis a fully adaptive algorithm that is based on the DWR idea, but applying the methodology locally in each time subinterval $I_n := (t_{n-1}, t_n]$ to the original problem with a perturbed initial condition given by the numerical solution at time t_{n-1}. Thus, rather than controlling the error for the whole interval $[0, T]$ what we have now is a good local control of the error in each interval I_n. Nevertheless, a weak point of this approach is that having a good local error control does not guarantee that the global error will be bounded. One future efforts will try to improve this shortcoming.

We have successfully combined the error committed in the discretization of the convective terms within the semi-Lagrangian framework with the DWR method in the reaction-diffusion stage of the equation. In doing so, we have defined a robust algorithm to numerically solve all kind of time-dependent convection-reaction-diffusion equations. The algorithm presented in this thesis it is self-sufficient in providing a precise criterium for the adaptation of both the time step Δt and the mesh size h, so that in each time subinterval I_n, they are automatically chosen according to a prescribed tolerance following an optimization criterium.

As an engineering application of the fully numerical adaptive procedure, we control the error for functionals of the solution to adapt the spatial mesh and the time step size in the context of reactive flows. The work presented in this thesis applies the low-Mach-number formulation of the Navier-Stokes equation for chemically reactive flows to describe the presence of lifted flames. The solutions obtained with our numerical scheme have been compared with those provided by other authors, and both being in good agreement (see Figure 6.1 and 6.2).

Furthermore, our adaptive procedure has been used in more several situations of lifted flames, with more general boundary conditions (planar jets and mixing layers for different velocities of both air and fuel feed streams). In planar jet problems our adaptive algorithm allows

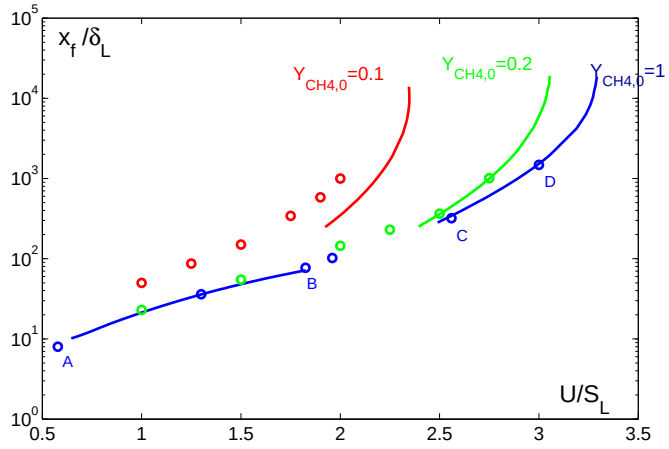

Figure 6.1: Lifted length x_f/δ_L versus velocity U/S_L. 'o' results obtained with our adaptive method and continuous line results of [39].

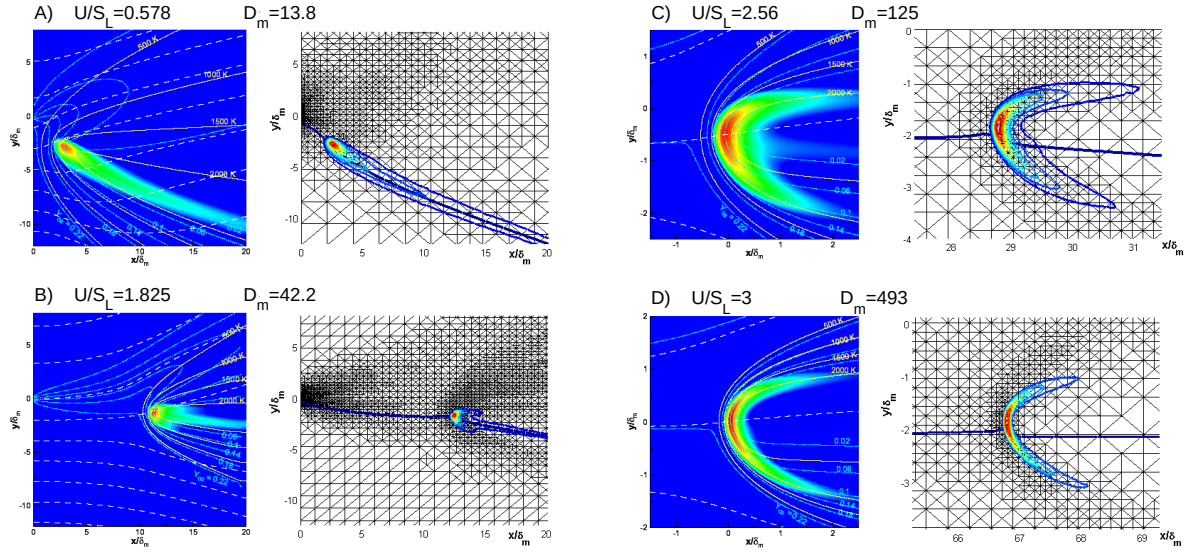

Figure 6.2: Flame front for different velocities U/S_L. Background mesh images obtained with our adaptive method and colored images by [39].

us to observe an interesting phenomenon: the interaction of the two front flames breaks the symmetry of the problem and an asymmetric steady solution is reached (see Figure 6.3). We have not found references of this behaviour in the scientific literature but a similar phenomenon has been observed in the laboratory for coaxial jets of methane.

Our future work will try to extend the ideas presented in this thesis to 3D configurations, in which the goal oriented adaptivity method will be even more efficient in comparison to classical algorithms, since we can treat more realistic chemical problems with a complex geometry as well as real applications in the engineering field; more specifically we will be interested in solving the problem of lifted flames in 3D.

The 3D extension requires a special care in setting up the data structure for mesh organization, for its implementation we will have to:

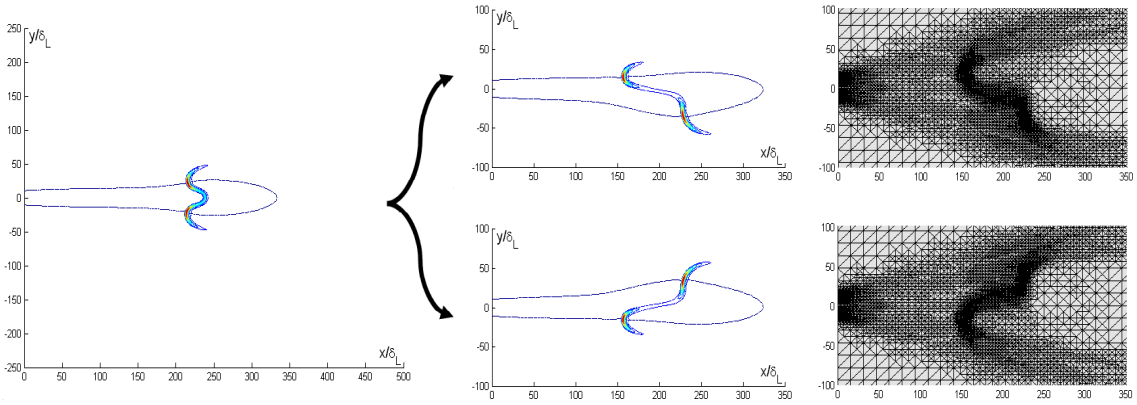

Figure 6.3: Symmetry breaking of two stable solutions with $Y_{CH_4} = 0.1$ and $U/S_L = 1.5$.

- describe a basic algorithms for the local refinement and coarsening of unstructured tetrahedra in 3D. We can take advantage of the work of several authors in this field to design a valid code for our data structure. The idea will be to devise a recursive refinement algorithm by a bisection procedure, which will lead us to nested meshes with hierarchical structure of binary trees (similar to the structure used in 2D).

- define a good spatial post-processing tool for unstructured tetrahedra.

APPENDIX

Proposition of R. Becker and R. Rannacher [11]

Let X be a function space and $L(\cdot)$ a differentiable functional on X. Its derivatives are denoted by $L'(\cdot)(\cdot)$, $L''(\cdot)(\cdot;\cdot)$ and $L'''(\cdot)(\cdot;\cdot;\cdot)$. Here, we use the convention that in these forms the dependence on the first argument in parentheses may be nonlinear while the dependence on all further arguments in the second set of parentheses is linear. We seek a stationary point x of $L(\cdot)$ on X, *i.e.,*

$$L'(x)(y) = 0 \quad \forall y \in X. \tag{A1}$$

This equation is approximated by a Galerkin method using a finite-dimensional subspace $X_h \subset X$, where $h \in R^+$ is a discretization parameter. The discrete problem seeks $x_h \in X_h$ satisfying

$$L'(x_h)(y_h) = 0 \quad \forall y_h \in X_h. \tag{A2}$$

Then, there holds the error representation

$$L(x) - L(x_h) = \frac{1}{2}L'(x_h)(x - y_h) + \mathcal{R}_h, \quad y_h \in X_h, \tag{A3}$$

with a remainder term $\mathcal{R}_h$ which is cubic in the error $e^x := x - x_h$,

$$\mathcal{R}_h := \frac{1}{2}\int_0^1 L'''(x_h + se^x)(e^x, e^x, e^x)s(s-1)ds. \tag{A4}$$

Proof: We can write

$$L(x) - L(x_h) = \int_0^1 L'(x_h + se^x)(e^x)ds.$$

The integral term can be calculated by the trapezoidal rule. For this, we have the well-know error representation

$$\int_0^1 f(t)dt = \frac{1}{2}\{f(0) + f(1)\} + \frac{1}{2}\int_0^1 f''(s)s(s-1)ds,$$

and then, we have the expression

$$L(x) - L(x_h) = \frac{1}{2}\{L'(x_h)(e^x) + L'(x)(e^x)\} + \frac{1}{2}\int_0^1 L'''(x_h + se^x)(e^x, e^x, e^x)s(s-1)ds.$$

Making use of equation (A1) $L'(x)(e^x) = 0$, and the equation (A2) $L'(x_h)(y_h) = 0$ for all $y_h \in X_h$ we have that

$$\frac{1}{2}\{L'(x_h)(e^x) + L'(x)(e^x)\} = \frac{1}{2}L'(x_h)(x - x_h) = \frac{1}{2}L'(x_h)(x - y_h), \quad y_h \in X_h.$$

This completes the proof obtaining the desired error representation.

■

Lifted flames symbols

Ω	Spatial domain.
Γ^D and $\Gamma^N \subset \partial\Omega$	Dirichlet and Neumann boundary of the domain.
$x - y$	Cartesian coordinates.
x_f	Lifted length of the flame front.
D_{TA}	Kinetic thermal diffusivity of the air at temperature T_0.
$Y_{F,0}$ and $Y_{O_2,A}$	Mass fraction of fuel (methane) and O_2 in the feed streams.
S_L	Planar flame velocity of the stoichiometric mixture.
$\delta_N = D_{TA}/U_0$	Characteristic Navier-Stokes length.
$\delta_L = D_{TA}/S_L$	Characteristic laminar flame thickness.
$\delta_m = (D_{TA}x_f/U_0)^{1/2}$	Characteristic mixing layer thickness at the flame front.
$D_m = (\delta_m/\delta_L)^2$	Damköhler number.
$U_0 \; and \; U_A$	Inflow velocities of the feed fuel and air streams respectively.
U	The velocity of the feed streams when $U_0 = U_A$.
a	Width of the injector in planar jets.
$Re_a = aU/D_{TA}$	Reynolds number of planar jets.
$\rho(x,t) \; and \; \rho_A$	Density of the fluid and density of the air at temperature T_0.
$\mathbf{u} = (u_x, u_y)$	Velocity of the fluid.
$p = p_{hyd} + (\mu_v + \frac{1}{3}\mu)(\nabla \cdot \mathbf{u})$	Hydrodynamic correction of the pressure.
T	Temperature of the fluid.
$\alpha = CH_4, O_2, CO_2 \; and \; H_2O$	Chemical species in the problem.
Y_α	Mass fraction of the species α.
$c_p = \sum_\alpha Y_\alpha c_{p,\alpha}$	Specific heat coefficient.
$M = \dfrac{1}{\sum_\alpha Y_\alpha/M_\alpha}$	Mean molecular mass.
μ	Dynamic viscosity of the fluid.
ρD_T	Dynamic thermal diffusion coefficient.
ρD_α	Dynamic mass diffusion coefficients.
$Pr = \mu/(\rho D_T)$	Prandtl number.
$Le_\alpha = D_T/D_\alpha$	Lewis number.
ω_α	Reaction rate of the species α.
B	Preexponencial factor.
$Ta(\phi)$	Activation temperature.
$q(\phi)$	Molar heat of reaction.
$H_F = q/M_{CH_4}$	Heat released per unit mass of fuel consumed.

s	Mass of oxygen burnt per unit mass of fuel consumed.
$S = sY_{F,0}/Y_{O_2,A}$	Stoichiometric ratio.
$Z = \dfrac{SY_F/Y_{F,0} - Y_{O_2}/Y_{O_2,A} + 1}{1+S}$	Mixture fraction.
$T_s = T_0 + \dfrac{H_F}{c_p} \cdot \dfrac{Y_{F,0}}{Y_{O_2,A}}$	Stoichiometric flame temperature.
$\beta = Ta_0(T_s - T_0)/T_s^2$	Zeldovich number.

Bibliography

[1] M. Ainsworth and J. T. Oden. 'A posteriori error estimation in finite element analysis'. Comput. Meth. Appl. Mech. Eng. Vol. 142 (1997), pp.1-88.

[2] A. Allievi and R. Bermejo. 'Finite element modified method of characteristics for the Navier-Stokes equations'. Int. J. Numer. Meth. Fluids, Vol. 32 (2000) pp. 439-464.

[3] A. Allievi and R. Bermejo. 'A Generalized Particle Search-Locate Algorithm for Arbitrary Grid'. J. Comp. Physics, Vol. 132 (1997), pp. 157-166.

[4] I. Babuska and W. Rheinboldt. 'Error estimates for adaptive finite element computations'. SIAM J. Numer. Anal. Vol. 15 (1978), pp. 736-754.

[5] I. Babuska and W. Rheinboldt. 'A posteriori error estimates for the finite element method'. Int. J. Numer. Meth. Eng. Vol. 12 (1978), pp. 1597-1615.

[6] I. Babuska and A.D. Miller. 'A feedback finite element method with a posteriori error estimation'. Comput. Meth. Appl. Mech. Eng. Vol. 61 (1987), pp. 1-40.

[7] I. Babuska and A.D. Miller. 'The post-processing approach in the finite element method, I: Calculations of displacements, stresses and other higher derivatives of the displacements'. Int. J. Numer. Meth. Eng. Vol. 20 (1984), pp. 1085-1109.

[8] I. Babuska and A.D. Miller. 'The post-processing approach in the finite element method, II: The calculation of stress intensity factors'. Int. J. Numer. Meth. Eng. Vol. 20 (1984), pp. 1111-1129.

[9] I. Babuska and A.D. Miller. 'The post-processing approach in the finite element method, III: A posteriori error estimation and adaptive mesh selection'. Int. J. Numer. Meth. Eng. Vol. 20 (1984), pp. 2311-2324.

[10] W. Bangerth and R. Rannacher. 'Finite element approximation of the aoustic wave equation: Error control and mesh adaptation'. East-West J. Numer. Math. Vol. 7 (1999), pp. 263-282.

[11] W. Bangerth and R. Rannacher. 'Adaptive Finite Element Methods for Differential Equations'. Birkhäuser, Basel, (2003).

[12] R.E. Bank and A. Weiser. 'Some a posteriori error estimators for elliptic partial differential equations'. Math. Comp. Vol. 44 (1985), pp. 283-301.

[13] R. Becker and R. Rannacher. 'A feed back approach to error control in finite element methods: Basic analysis and examples'. East-West J. Numer. Math. Vol. 4 (1996), pp. 237-264.

[14] R. Becker. 'Mesh adaptation for stationary flow control'. J. Math. Fluid. Mech. Vol. 3 (2001) pp. 317-341.

[15] R. Bermejo and A. Staniforth: 'The conversion of semi-Lagrangian advection schemes to quasi-monotone schemes' Month Weath. Rev., 120 (1992) 2622-2632.

[16] R. Bermejo: 'A Galerkin-characteristic algorithm for transport-diffusion equations'. SIAM J. Numer. Anal. Vol. 32 (1995), pp. 425

[17] R. Bermejo and J. Carpio. 'An adaptive finite element semi-Lagrangian implicit-explicit Runge-Kutta-Chebyshev method for convection dominated reaction-diffusion problems'. Applied Numerical Mathematic Vol. 58 (2008). pp 16-39.

[18] R. Bermejo and J. Carpio. 'A semi-Lagrangian-Galerkin projection scheme for convection equations'. IAM Journal of Numerical Analysis. Accepted. (2008).

[19] C. Bernardi. 'Optimal finite-element interpolation on curved domains'. SIAM J. Numer. Anal., Vol. 36 (1989), pp. 1212-1240.

[20] M. Braack. 'An adaptive finite element method for reactive flow problems'. Doktorarbeit, Institut für Angewandte Mathematik, Universität Heidelberg (1998).

[21] S.C. Brenner and L.R. Scott. 'The Mathematical Theory of Finite Element Methods'. Springer, NewYork (1994).

[22] M. O. Bristeau, R. Glowinsky and J. Periaux. 'Numerical Methods for the Navier-Stokes equation'. Computer Methods in Applied Mechanics and Engineering, Vol. 32 (1982), pp. 199-259.

[23] J. Buckmaster. 'Edge-Flames and Their Stability'. Combust. Sci. Tech. Vol. 115 (1996), pp. 41-68.

[24] J. Buckmaster. 'Edge-flames'. Prog. Energ. Combust. Sci. Vol. 28 (2002), pp. 435-475.

[25] J.C. Butcher. 'Numerical Methods for Ordinary Differential Equations'. John Wiley. Chichester. England. (2003).

[26] J. Cahouet and J.P. Chabard. 'Some fast 3D solvers for the generalized Stokes problem'. Inter. J. Numer. Meth. Fluids. Vol. 8 (1988), pp. 269-295.

[27] R. Courant. 'Variational method for the solution of problems of equilibrium and vibrations' Bulletin of American Mathematical Society'. Vol. 49 (1943), pp. 1-43.

[28] J. Daou and A. Liñán. 'The role of unequal diffusivities in ignition and extinction fronts in strained mixing layers'. Combust. Theory Modelling Vol. 2 (1998), pp. 449-477.

[29] E.J. Dean and R. Glowinsky. 'On some finite elements methods for the numerical simulation of incompressible viscous flow'. Incompressible computational fluid Dynamics. (M.D. Gunzburger and R.A. Nicolaides eds). Cambridge: Cambridge University Press. (1993).

[30] J.W. Dold. 'Flame propagation in a nonuniform mixture: Analysis of a slowly varying Triple Flame'. Combust. Flame Vol.76 (1989), pp. 71-88.

[31] K. Eriksson and C. Johnson. 'An adaptive finite element method for linear elliptic problems'. Math. Comp. Vol. 50 (1988), pp. 361-383.

[32] K. Eriksson and C. Johnson. 'Adaptive finite element methods for parabolic problems, I: A linear model problem'. SIAM J. Numer. Anal. Vol. 28 (1991), pp. 43-77.

[33] K. Eriksson and C. Johnson. 'Adaptive finite element methods for parabolic problems, II: Optimal error estimates in $l_\infty l_2$ and $l_\infty l_\infty$'. SIAM J. Numer. Anal. Vol. 32 (1995), pp. 706-740.

[34] K. Eriksson and C. Johnson. 'Adaptive finite element methods for parabolic problems, IV: Nonlinear problems'. SIAM J. Numer. Anal. Vol. 32 (1995), pp. 1729-1749.

[35] K. Eriksson and C. Johnson. 'Adaptive finite element methods for parabolic problems, V: Log-time integration'. SIAM J. Numer. Anal. Vol. 32 (1995), pp.1750-1763.

[36] K. Eriksson, D. Estep, P. Hansbo and C. Johnson. 'Introduction to adaptive methods for differential equations'. In Acta Numerica (A. Iserles, ed.), Vol. 4 (1995), Cambridge University Press, pp. 105-158.

[37] K. Eriksson and C. Johnson. 'Adaptive finite element methods for parabolic problems, VI: Analytic semigroups'. SIAM J. Numer. Anal. Vol. 35 (1998), pp. 1315-1325.

[38] D. Estep and S. Larsson. 'The discontinuous Galerkin method for semilinear parabolic problems'. Modél. Math. Anal. Numér. Vol. 27 (1993), pp.611-643.

[39] E. Fernandez-Tarrazo, M. Vera and A. Liñan. 'Lift-off and blow off of a diffusion flame between parallel streams of fuel and air'. Combust. Flame Vol. 144 (2006), pp. 261-276.

[40] E. Fernandez-Tarrazo, A.L. Sánchez, A. Liñan and F.A. Williams. 'A Simple One-Step Chemistry Model for Partially Premixed Hydrocarbon Combustion'. Combust. Flame Vol. 147 (2006), n^o 1-2, pp. 32-38.

[41] C. Foias, C. Guillopé and R. Temam. 'Lagrangian representation of the flow'. J. Diff. Eqns. Vol. 57 (1985), pp. 440-449.

[42] V. Girault and P.A. Raviart. 'Finite element methods for Navier-Stokes equation: theory and algorithms', in Finite Element for Navier-Stokes Equations, Springer, Berlin, (1986).

[43] P.M. Gresho. 'Some current CFD issues relevant to the incompressible Navier-Stokes equations' Comput. Methods Appl. Mech. Eng. Vol. 87 (1991), pp. 201-252.

[44] K. Gustafsson, M. Lundh and G. Söderlind. 'A PI step size control for the numerical solution of ordinary differential equations' BIT Vol. 28 (1988), pp. 270-287.

[45] R. Hartmann. 'A posteriori Fehlerschätzung und adaptive Schrittweiten- und Ortsgittersteuerung bei Galerkin- Verfahren für die Wärmeleitungsgleichung'. Diploma thesis, Institute of Applied Mathematics, University of Heidelberg, (1998).

[46] P. Houston and E. Süli. 'Adaptive Lagrange-Galerkin methods for unsteady convection-diffusion problems' Math. of Computation, Vol. 70 (2000), pp. 77-106.

[47] C. Johnson. 'Numerical Solutions of Partial Differential Equations by the Finite Element Method'. Cambrige University Press (1987).

[48] C. Johnson, 'A new paradigm for adaptive finite element methods'. In Proc. MAFELAP 93 (J. Whiteman, ed.), Wiley (1994), pp. 105-120.

[49] R. Klein. 'Semi-implicit extension of a Gudonov-type scheme based on low Mach number asymptotics I: One-dimensional flow'. J. Comp. Phys. Vol. 121 (1995) pp. 213-237.

[50] I. Kossaczký. 'A recursive approach to local mesh refinement in two and three dimensions'. J. Comput. Appl. Math. Vol. 55 (1994), pp. 275-288.

[51] P. Ladeveze and D. Leguillon. 'Error estimate procedure in the finite element method and applications'. SIAM J. Numer. Anal. Vol. 20 (1983), pp. 485-509.

[52] J. Lang. 'Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems' Lecture Notes in Computational Science and Engineering, Springer Vol. 16 (2001), pp. 83-88.

[53] A. Liñán, E. Fernández-Tarrazo, M. Vera and A. L. Sánchez. 'Lifted laminar jet diffusion flames'. Combust. Sci. Tech. Vol. 177 (2005), pp. 933–953.

[54] W. Mitchell. 'A comparison of adaptive refinement techniques for elliptic problems'. ACM Trans. Math. Softw. Vol. 15 (1989), pp. 326-347.

[55] K.W. Morton, A. Priestley and E. Süli. 'Stability of the Lagrange-Galerkin method with non-exact integration'. SIAM Mathematical Modelling and Numerical Analysis, Vol. 22 (1988), pp. 625-653.

[56] T. Poinsot and D. Veynante. Theoretical and numerical combustion. Edwards Phyladelphia, PA, 2001, p. 54.

[57] H. Phillips. 'Flame in a buoyant methane layer'. Proc. Combust. Inst. Vol. 10 (1965), pp. 1277-1283.

[58] R. Rannacher. 'A posteriori error estimation in least-squares stabilized finite element schemes'. Comput. Meth. Appl. Mech. Eng. Vol. 166 (1998), pp. 99-114.

[59] R. Rannacher. 'Duality techniques for error estimation and mesh adaptation in finite element methods, in Adaptive Finite Elements in Linear and Nonlinear Solid and Structural Mechanics' E. Stein, ed. Vol. 416 (2000)of CISM Courses and Lectures, Springer.

[60] G. R. Ruetsch, L. Vervisch and A. Liñán. 'Effects of heat release on triple flames'. Phys. Fluids Vol. 7 (1995), pp. 1447-1454.

[61] M. Schmich and B. Vexler. 'Adaptivity with dynamic meshes for space-time finite element discretizations of parabolic equations'. SIAM J. Sci. Comput. Vol. 30(1), (2008) pp. 369-393.

[62] A. Schmidt and K. G. Siebert. 'Designed of Adaptive Finite Element Software. The Finite Element Toolbox ALBERTA'. Springer Lectures Notes in Computational Science and Engineering, Springer Berlin (2005).

[63] G.P. Smith, D.M. Golden, M. Frenklach, N.W. Moriarty, B. Eiteneer, M. Goldenberg, C.T. Bowman, R.K. Hanson, S. Song, W.C. Gardiner, V.V. Lissianski and Z. Quin. Gas Research Institute Mechanism. http://www.me.berkeley.edu/grid_mech/

[64] M.D. Smooke and V. Giovangigli. 'Formulation of the premixed and nonpremixed test problems'. In Reduced Kinetic Mechanisms and Asymptotic Approximations for Methane-Air Flames, Ed. Smooke, M.D. Springer-Verlag (1990) pp. 1-28.

[65] B.P. Sommeijer, L.F. Shampine and J.G. Verwer. 'RKC: An Explicit solver for Parabolic PDEs'. Modelling Analysis and Simulation (MAS) MAS-R9715 June 30, (1997).

[66] R. Temam. 'Infinite-Dimensional Dynamical Systems in Mechanics and Physics. Springer-Verlag. New York (1988).

[67] M.J. Turner, R.W. Clough, H.C. Martin and L.C. Topp. 'Stiffness and deflection analysis of complex structures' Journal of Aeronautical Sciences, Vol. 23 (1956), pp. 805-824.sität Heidelberg (1998).

[68] R. Verfürth. 'A Review of A Posteriori Error Estimation and Adaptative Mesh-Refinement Techniques'. Wiley/Teubner, New York/Stuttgart (1996).

[69] J.G. Verwer, W. Hundsdorfer and B.P. Sommeijer. 'Convergence properties of the Runge-Kutta-Chebyshev method'. Numer. Math., Vol. 57 (1990), pp. 157-178.

[70] J.G. Verwer, B.P. Sommeijer and W. Hundsdorfer. 'RCK time-stepping for advection-diffusion-reaction problems'. Journal of computational Physics, Vol. 201 (2004), pp. 61-79.

[71] J.G. Verwer and B.P. Sommeijer. 'An implicit-explicit Runge-Kutta-Chebyshev scheme for diffusion-reaction equations'. SIAM J. Sci. Comput. Vol. 25 (2004) pp.1824-1835.

[72] C. Westbrook and F. Dryer. 'Simplified Reaction Mechanisms for the Oxidation of Hydrocarbon Fuels in FlamesCombust'. Sci. Tech. 27 (1981) 31-43.

[73] S. Ziukas and N.E. Wiberg. 'Adaptive procedure with superconvergent patch recovery for linear parabolic problems, in Finite Element Methods: Superconvergence, Post-Processing, and A Posteriori Estimates'. M. Krizek et al., eds. Vol. 196 (1998) of Inc. Lect. Notes Pure Appl. Math., Marcel Dekker, pp. 303-314.