UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH

Monotonicity-preserving finite element methods for hyperbolic problems

Jesús Bonilla

Universitat Politècnica de Catalunya

Doctoral Thesis

Monotonicity-preserving finite element methods for hyperbolic problems

Author:
Jesús Bonilla

Supervisor:
Prof. Santiago Badia

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

Doctorat en Enginyeria Civil

Departament d'Enginyeria Civil i Ambiental

Barcelona, November 2019

Escola de Camins
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports
UPC BARCELONATECH

To my family & friends

"A thesis has to be presentable... but don't attach too much importance to it. If you do succeed in the sciences, you will do later on better things and then it will be of little moment. If you don't succeed in the sciences, it doesn't matter at all. "

Paul Ehrenfest

Abstract

This thesis covers the development of monotonicity-preserving finite element methods for hyperbolic problems. In particular, scalar convection-diffusion and Euler equations are used as model problems for the discussion in this dissertation.

A novel artificial diffusion stabilization method has been proposed for scalar problems. This technique is proved to yield monotonic solutions, to be local extremum diminishing (LED), Lipschitz continuous, and linearity preserving. These properties are satisfied in multiple dimensions and for general meshes. However, these results are limited to first order Lagrangian finite elements. A modification of this stabilization operator that is twice differentiable has been also proposed. With this regularized operator, nonlinear convergence is notably improved, while the stability properties remain unaltered (at least, in a weak sense).

An extension of this stabilization method to high-order discretizations has also been proposed. In particular, arbitrary order space-time isogeometric analysis is used for this purpose. It has been proved that this scheme yields solutions that satisfy a global space-time discrete maximum principle unconditionally. A partitioned approach has also been proposed. This strategy reduces the computational cost of the scheme, while it preserves all stability properties. A regularization of this stabilization operator has also been developed. As for the first order finite element method, it improves the nonlinear convergence without harming the stability properties.

An extension to Euler equations has also been pursued. In this case, instead of monotonicity-preserving, the developed scheme is local bounds preserving. Following the previous works, a regularized differentiable version has also been proposed. In addition, a continuation method using the parameters introduced for the regularization has been used. In this case, not only the nonlinear convergence is improved, but also the robustness of the method. However, the improvement in nonlinear convergence is limited to moderate tolerances and it is not as notable as for the scalar problem.

Finally, the stabilized schemes proposed had been adapted to adaptive mesh refinement discretizations. In particular, nonconforming hierarchical octree-based meshes have been used. Using these settings, the efficiency of solving a monotonicity-preserving high-order stiff nonlinear problem has been assessed. Given a specific accuracy, the computational time required for solving the high-order problem is compared to the one required for solving a low-order problem (easy to converge) in a much finer adapted mesh. In addition, an error estimator based on the stabilization terms has been proposed and tested.

The performance of all proposed schemes has been assessed using several numerical tests and solving various benchmark problems. The obtained results have been commented and included in the dissertation.

Resum

La present tesi tracta sobre mètodes d'elements finits que preserven la monotonia per a problemes hiperbòlics. Concretament, els problemes que s'han utilitzat com a model en el desenvolupament d'aquesta tesi són l'equació escalar de convecció–difusió–reacció i les equacions d'Euler.

Per a problemes escalars s'ha proposat un nou mètode d'estabilització mitjançant difusió artificial. S'ha provat que amb aquesta tècnica les solucions obtingudes són monòtones, l'esquema "disminueix els extrems locals", i preserva la linearitat. Aquestes propietats s'han pogut demostrar per múltiples dimensions i per malles generals. Per contra, aquests resultats només són vàlids per elements finits Lagrangians de primer ordre. També s'ha proposat una modificació de l'operador d'estabilització per tal de que aquest sigui diferenciable. Aquesta regularització ha permès millorar la convergència no-lineal notablement, mentre que les propietats d'estabilització no s'han vist alterades.

L'anterior mètode d'estabilització s'ha adaptat a discretitzacions d'alt ordre. Concretament, s'ha utilitzat anàlisi isogeomètrica en espai i temps per a aquesta tasca. S'ha provat que les solucions obtingudes mitjançant aquest mètode satisfan el principi del màxim discret de forma global. També s'ha proposat un esquema particionat. Aquesta alternativa redueix el cost computacional, mentre preserva totes les propietats d'estabilitat. En aquest cas, també s'ha realitzat una regularització de l'operador d'estabilització per tal de que sigui diferenciable. Tal i com s'ha observat en els mètodes de primer ordre, aquesta regularització permet millorar la convergència no-lineal sense perdre les propietats d'estabilització.

Posteriorment, s'ha estudiat l'adaptació dels mètodes anteriors a les equacions d'Euler. En aquest cas, en comptes de preservar la monotonia, l'esquema preserva "fites locals". Seguint els desenvolupaments anteriors, s'ha proposat una versió diferenciable de l'estabilització. En aquest cas, també s'ha desenvolupat un mètode de continuació utilitzant els paràmetres introduïts per a la regularització. En aquest cas, no només ha millorat la convergència no-lineal sinó que l'esquema també esdevé més robust. Per contra, la millora en la convergència no-lineal només s'observa per a toleràncies moderades i no és tan notable com en el cas dels problemes escalars.

Finalment, els esquemes d'estabilització proposats s'han adaptat a malles de refinament adaptatiu. Concretament, s'han utilitzat malles no-conformes basades en *octrees*. Utilitzant aquesta configuració, l'eficiència de resoldre un problema altament no-lineal ha estat avaluada de la següent forma. Donada una precisió determinada, el temps computacional requerit per resoldre el problema utilitzant un esquema d'alt ordre ha estat comparat amb el temps necessari per resoldre'l utilitzant un esquema de baix ordre en una malla adaptativa molt més refinada. Addicionalment, també s'ha proposat un estimador de l'error basat en l'operador d'estabilització.

El comportament de tots els esquemes proposats anteriorment s'ha avaluat mitjançant varis tests numèrics. Els resultats s'han compilat i comentat en la present tesi.

Acknowledgements

This thesis would have never been possible without the support, advice and guidance of Santiago Badia. I am very grateful for your patience and for always attending me the countless times that I have asked for your help. It has been a pleasure to work with you and all researchers at the Large Scale Scientific Group of CIMNE. I would like to thank Javier for his kind support and advice, and Alberto for his help and availability, always ready for sharing his deep knowledge in computer science.

I would like to especially thank all the officemates that at some point have stayed at office 214 during these years. Oriol, Alba and Marc helped me during the beginning of this thesis, and to get started with `FEMPAR`. Eric and Pere joined later, but I greatly appreciate the discussions and supportive talks I have had with all of them. I would like to thank Manuel, Francesc, Víctor, Daniel, Ramon, Hieu, Jerrad, and Àlex, for their willingness to help and discuss about our research.

A deep thanks to John Shadid and his group in Sandia. I am very grateful for the opportunity of joining his group for a while, and for all his help and support during the stay (and afterwards). I would like to especially thank Sibu and Sidafa for helping me getting started with Drekar, and for their help and fruitful discussions. I make this words extensive to all the members of the group, who always showed open for discussion and made me feel one more of the team from the very first day.

In a more personal side, I am very grateful for the unconditional support of all my hometown friends. I would like to thank them for always being there and for making life more fun. A special thanks to Gonzalo and all my former workmates at CTM. I am grateful for their guidance during my first years in research, and for making me discover and enjoy this profession. I could not forget to thank all JIPI organizers for their work, and for all the things I could learn while contributing to its organization.

Words cannot express how grateful I am to my parents and my brother. I would like to thank them for the support and confidence given during all this journey. During my whole life they have been examples of determination, hard work, and confidence. Without these values learned from them, I would have never been able to pursue this work.

Finally, I would like to gratefully acknowledge the support received from "la Caixa" Foundation through its PhD scholarship program (LCF/BQ/DE15/10360010).

I have the feeling that, in some extent, all of you have contributed to help me complete this thesis. My gratitude to all of you.

Contents

List of Figures

List of Tables

List of Abbreviations

FE	**F**inite **E**lement
dG	**d**iscontinuous **G**alerkin
cG	**c**ontinuous **G**alerkin
DOF	**D**egree **O**f **F**reedom
SSP	**S**trong **S**tability **P**reserving
RK	**R**unge **K**utta
BE	**B**ackward **E**uler
TVD	**T**otal **V**ariation **D**iminishing
DMP	**D**iscrete **M**aximum **P**rinciple
MP	**M**aximum **P**rinciple
FCT	**F**lux **C**orrected **T**ransport
LED	**L**ocal **E**xtremum **D**iminishing
CFD	**C**omputational **F**luid **D**ynamics
CSM	**C**omputational **S**olid **M**echanics
CEM	**C**omputational **E**lectro**M**agnetics
PDE	**P**artial **D**ifferential **E**quation
FDM	**F**inite **D**ifference **M**ethod
FVM	**F**inite **V**olume **M**ethod
FEM	**F**inite **E**lement **M**ethod
FE	**F**inite **E**lement
MHD	**M**agneto**H**ydro**D**ynamics
IMEX	**IM**plicit–**EX**plicit
CDR	**C**onvection–**D**iffusion–**R**eaction

AMR **A**daptive **M**esh **R**efinement

CFL **C**ourant–**F**riedrichs–**L**ewy

Chapter 1

Introduction

Computational mechanics is an interdisciplinary field of science that studies the use of computational methods to simulate complex natural phenomena. This discipline is based on three main pillars: mechanics, mathematics, and computer science. Mechanics provide the necessary mathematical models that describe the behavior of complex natural phenomena. Mathematics are in charge of developing and analyzing the necessary methods to solve these models. Finally, computational science is fundamental to perform the actual computation efficiently.

During the last decades, computational resources have continuously increased and have become more available. This has boosted the capability of computational mechanics to solve complex phenomena faster and more accurately. As a result, computational mechanics have been integrated in many engineering and scientific environments. In some fields, computational mechanics methods are well established. In these fields, they have become an essential analysis tool in design and engineering processes.

For the scientific community, computational science has become a "third approach to research" between experimentation and theory. In many fields, simulation is used in a stage prior to experimentation as a "virtual laboratory". This stage allows scientists to try their experiment design beforehand and optimize it accordingly. Therefore, it reduces the total number of required experiments, which in turn reduces the time and costs of research. Moreover, simulation can provide data which is not available through direct measurements, e.g., the distribution of stresses in the interior of a mechanical part, the temperature distribution in the interior of a bulk, or the complete velocity field around complex objects.

Computational mechanics is a vast discipline and comprises many specialties. For example, computational fluid dynamics (CFD), computational solid mechanics (CSM) or computational electromagnetics (CEM), among many others. Regardless of the specific field, most (if not all) of them follow the same strategy to perform the simulations. The first step consists in defining or selecting the appropriate model that is able to describe the phenomenon of interest. Usually, this model takes the form of a continuous partial differential equation (PDE). Then, this PDE is numerically approximated using the discretization method of choice. There exist many discretization procedures. The most known methods are the finite difference method (FDM), the finite volume method (FVM), and the finite element method (FEM). The result of any discretization process is

a system of algebraic equations. The solution to this system approximates the behavior of the natural phenomenon at hand.

The present thesis focuses on FE discretizations for CFD. In particular, it addresses the simulation of convection dominated flows. This problem is still challenging for the CFD community due to the stability issues that it may present. Namely, the numerical approximation of convection dominated flows might lead to spurious oscillations. This behavior can be suppressed (or at least limited) by the addition of stabilization terms to the original FE method. This thesis is centered in developing stabilization procedures for the FE method in order to improve its stability, efficiency, and applicability.

1.1 Motivation

Since the 1950s, it has been said that nuclear fusion energy is a safe and limitless source for electricity generation [19]. One of the main unsolved issues of this technology is handling the fuel inside the reactors, which is in the form of plasma. Predicting and controlling the behavior of plasmas is very difficult due to a number of different reasons. For example, stability simulations for magnetic confinement fusion present multiscale and multiphysic effects, and many instabilities may arise. Instabilities can deteriorate the confinement or even lead to a sudden stop of the reaction. Thus, it is crucial to design reactors able to maintain plasma stability, reducing and mitigating instabilities so the reaction can be sustained. Performing experiments with this kind of devices consume a lot of resources and time. Therefore, this field of research would clearly benefit from stable, accurate and efficient numerical discretizations to enable high-fidelity simulations.

Even though the fusion phenomenon is intrinsically a multiscale problem –time scales range from 10^{-10} seconds electron cyclotron to 10^{4} seconds discharge time scale–, aspects of the macroscopic (longer time and spatial) scales can be modeled using fluid models of plasma [19, 81]. There exist many models depending on how many physical effects are neglected. In general, any of these models can be seen as the compressible Navier-Stokes equations coupled with some reduced form of Maxwell's equations.

Traditionally, compressible flow problems have been discretized using explicit time integrators. This kind of integrators are only conditionally stable. The so-called Courant–Friedrichs–Lewy (CFL) stability condition needs to be satisfied in order to ensure stability of the scheme. However, the CFL stability condition becomes very stringent due to the viscous terms present in the formulation. This is especially evident in locally refined meshes. For example, assume a given mesh is locally refined by a factor of 10. Then, due to the diffusive terms present in the formulation, the global time step would need to be reduced 100 times to satisfy the new stability conditions. Therefore, implicit or implicit–explicit (IMEX) time integration methods are preferred in these situations [19].

Stability of the temporal integration is not the only difficulty. Solutions to compressible flow equations need to satisfy some constraints. Namely, density and internal energy are non-negative, and the entropy is non-decreasing. Discretization schemes that satisfy

those properties are of crucial importance to provide physically meaningful solutions. Satisfying these constraints is difficult for problems with discontinuous solutions. It is even more challenging in the case of requiring the usage of implicit time integration. A paradigmatic example of this situation might be the simulation of fluid models of plasma. However, there exist many applications where implicit time integration is preferred and, at the same time, their solutions need to be bounded by specific values.

The simulation of supersonic aircrafts is a classical example where compressible flow equations are used. Viscous effects are neglected many times for this application. Hence, it might seem appropriate to use an explicit integrator. However, this is not always the case. For instance, implicit time integration (or even direct to steady state) is preferred if one is only interested in the steady behavior of the flow. The analysis of shock waves and boundary layer interactions is another example. In this case, viscous effects cannot be neglected and implicit time integration might be preferred.

Many problems present in science and engineering can be modeled using convection–diffusion–reaction (CDR) equations. The concentration evolution of some pollutant in a fluid is a classical example that can be modeled with CDR equations. In this case, the problem might exhibit discontinuities or sharp layers depending on how the pollutant gets into the fluid. However, the concentration must remain positive. As in the previous example, implicit time integration schemes might be preferred depending on the particular simulation interests.

Chemical industry is another application example of CDR equations. The concentration of reactants and products in a reaction chamber can be modeled using CDR equations. Fast reactions can lead to abrupt changes in concentrations. However, one might not be interested in resolving all time scales of this reaction. In this case, using implicit methods might be more computationally efficient.

In all previous examples, positivity needs to be preserved for one or more unknowns. Satisfying this constraint is particularly difficult for these problems, which can develop shocks or present sharp layers. As mentioned above, without any special treatment, numerical schemes may present an oscillatory behavior in the vicinity of discontinuities. These oscillations might lead to violation of the physical constraints previously mentioned. Thus, the obtained solution might become physically meaningless. Furthermore, some terms of the discrete equations might become undefined, e.g., many stabilization terms for Euler equations depend on the speed of sound which is undefined for negative values of the internal energy. Hence, the stabilization term becomes undefined, and thus the numerical scheme is unable to provide any solution.

Therefore, positivity preserving solvers for implicit time integration are of special interest for the CFD community. Unfortunately, the current state-of-the-art of positivity preserving numerical methods *for implicit time integrations* of systems of equations is rather scarce. Actually, to the best of our knowledge there is no scheme able to prove these properties for implicit time integration. It is important to mention that for explicit time integration the situation is very different. There exist many explicit schemes able

to preserve these constraints [68]. However, for the reasons previously exposed we will focus on implicit time integration.

For implicit solvers, positivity, or even monotonicity preservation, has only been formally proved for scalar problems. In any case, monotonicity-preserving FE schemes might still present some limitations. For instance, most of the previous methods in the literature require additional conditions on the mesh to preserve positivity. Having to solve a very stiff nonlinear problem is another common drawback of previous schemes.

In the present work, monotonicity-preserving FE methods for the scalar CDR problem and Euler equations are analyzed and developed. More specifically, in this thesis we develop efficient monotonicity-preserving FE schemes for scalar problems. The extension to Euler equations is also tackled. Moreover, we perform the extension of these methods to numerical schemes with AMR. Briefly, AMR are schemes able to dynamically and automatically adapt the accuracy and resolution of the approximated solution to the features of the problem at hand.

1.2 Thesis objectives

Taking into account the motivation of this thesis, below we unwrap some specific goals to address the global objective.

- **Design of a monotonicity-preserving scheme for arbitrary mesh geometries.**

  To date, several monotonicity-preserving stabilization techniques have been proposed for implicit FE methods. However, many of them can only provide monotonic solution under mesh restrictions. Thus, the applicability of the resulting scheme is undermined. Therefore, we consider that it is important to overcome this restriction in order to develop a stabilization method applicable for general meshes.

- **Analysis and improvement of the nonlinear convergence behavior of monotonicity-preserving schemes.**

  As proved long ago by Godunov [36] any high-order monotonicity-preserving scheme is necessarily nonlinear. The drawback of many schemes present in the literature is the requirement of solving stiff nonlinear problems. This notably increases the computational cost of the original (unstabilized) problem. In order to improve the applicability of these methods, we will explore the enhancement of the nonlinear convergence properties.

- **Extension to high-order discretizations in space and time.**

  Most of the state-of-the-art monotonicity-preserving stabilization methods for implicit time integration are limited to first order Lagrangian FEs. Therefore, one

cannot benefit from the higher accuracy of high-order FEs, which is of special interest for problems that combine shocks and smooth regions. Moreover, in the context of *hp*-adaptive schemes, forcing $p = 1$ in the vicinity of discontinuities and shocks might become cumbersome. Moreover, many authors refer to strong stability preserving (SSP) Runge Kutta (RK) methods to achieve high-order convergence in time. However, to achieve high convergence rates these kind of methods require to satisfy a CFL-like condition [37]. The motivation of using an implicit time integrator was precisely to avoid stability conditions on the time step length. Therefore, we will explore alternatives to SSP RK methods to achieve high-order time integration.

- **Extension to first order hyperbolic systems of equations.**

  After exploring the previously mentioned goals, an important step is to start working with systems of equations. In particular, to extend at least some of the previous achievements to a problem closer to the application in the motivation. As a first step, we will consider the extension of the methods developed for scalar problems to Euler equations.

- **Extension to AMR FE schemes.**

  In the case of problems with discontinuities, the ability of automatically adapt the resolution of the mesh to the features of the problem can notably improve the convergence. Therefore, we consider important to ensure that the methods developed in this thesis are well suited to be used in this kind of discretizations. Moreover, the stabilization methods explored in this thesis are characterized by restricting its action to the vicinity of discontinuities. Hence, we will explore the possibility of using this property in the AMR process to identify which regions of the mesh need to be refined.

- **Assessment of the efficiency of high-order monotonicity-preserving schemes in AMR context.**

  As previously mentioned, using this kind of methods requires to solve a stiff nonlinear problem. Previous goals explicitly attempt to improve the nonlinear convergence. However, we find interesting to check whether it is still clearly better to use a high-order scheme with AMR. That is, for a given accuracy, we would like to test whether it is more efficient to use a high-order method (with a stiff nonlinear problem), or if it is better to use a low-order method with a much finner mesh.

- **Code development.**

  All the results in this thesis (but the ones in Chapter 2) have been implemented in the in-house FE library `FEMPAR` [9, 11]. `FEMPAR` stands for Finite Element Multiphysics PARallel solvers, and it is a finite element library that provides all the

necessary tools for computing FE approximations of PDEs, e.g. from discretization to numerical linear algebra. It is important to mention that `FEMPAR` is a collaborative software project and without the contributions of current and former developers the results in this thesis would have been impossible to achieve.

1.3 Document structure

The first chapter of this thesis contains a brief introduction, the motivations of the research developed, and the specific goals of this thesis. Chapters 2 to 5 contain the main contributions of this study. Each one of this chapters corresponds directly to the publications in the list of the next section. The chapters are self-contained, preserve the structure of the paper, and can be read independently. However, we have tried to keep the notation as homogeneous as possible.

Chapter 2 is devoted to the development of a monotonicity-preserving stabilization method for first order Lagrangian FEs in arbitrary meshes. It also contains a differentiable version that improves the computational cost. First, Sect. 2.1 contains an introduction to monotonicity-preserving stabilization methods for FEs. In Sect. 2.2, the continuous problem and its discretization using the FE method are presented. Sect. 2.3 contains the formulation of the novel nonlinear stabilization method. Sect. 2.4 is devoted to the monotonicity analysis of the proposed method. An alternative approach is presented in Sect. 2.5. Lipschitz continuity of the methods is proved in Sect. 2.6. A differentiable version the previous method is presented in Sect. 2.7. Sect. 2.8 is devoted to nonlinear solvers. Different numerical experiments are introduced in Sect. 2.9. Finally, we draw some conclusions in Sect. 2.10.

Chapter 3 contains an extension of the previous stabilization method to space–time arbitrary high-order isogeometric analysis. First, we introduce the problem, its discretization, and monotonicity properties for scalar problems in Sect. 3.2. Then, the stabilization techniques are introduced in Sect. 3.3. Sect. 3.4 is devoted to a partitioned time integration scheme. Afterwards, we introduce a regularized version of the stabilization term in Sect. 3.5. Finally, we show numerical experiments in Sect. 3.6 and draw some concluding remarks in Sect. 3.7.

In Chapter 4 the previous differentiable shock detector techniques are combined with stabilization methods for systems of equations. The resulting scheme is proved to be local bounds preserving for first order hyperbolic problems. In Sect. 4.2 we present the continuous Galerkin (cG) discretization for scalar convection and Euler equations. Sect. 4.3 is devoted to the definition of the stabilization terms. We describe the nonlinear solvers used in Sect. 4.4. Then, we present the numerical experiments performed in Sect. 4.5. Finally, we draw some conclusions in Sect. 4.6.

In Chapter 5 the performance of the methods developed in Chapter 2 and Chapter 4 are evaluated in the context of AMR. First, we introduce the problem, its discretization, and monotonicity properties for scalar problems and hyperbolic systems in Sect. 5.2.

Then, the stabilization techniques are introduced in Sect. 5.3. Sect. 5.4 is devoted to the AMR strategy. Afterwards, we introduce the nonlinear solvers used in Sect. 5.5. Finally, we show numerical experiments in Sect. 5.6 and draw some conclusions in Sect. 5.7.

Finally, Chapter 6 summarizes the conclusions and the main goals achieved in the thesis at hand. In addition, we also introduce possible future works to pursue based on the developments in this thesis.

1.4 List of publications and conference participations

The work presented in this thesis has also been published in international peer reviewed journals, as well as in international conferences. The journal articles written in the scope of this thesis are listed below.

[4] S. Badia and J. Bonilla, *Monotonicity-preserving finite element schemes based on differentiable nonlinear stabilization,* Computer Methods in Applied Mechanics and Engineering **313** (2017) 133–158.

[16] J. Bonilla and S.Badia, *Maximum principle preserving space-time isogeometric analysis*, Computer Methods in Applied Mechanics and Engineering **354** (2019) 422–440.

[6] S. Badia, J. Bonilla, S. Mabuza and J. Shadid, *Differentiable local bounds preserving stabilization for first order hyperbolic problems.* Submitted.

[17] J. Bonilla and S. Badia, *Monotonicity-preserving finite element schemes with adaptive mesh refinement for hyperbolic problems.* In preparation.

In addition, the following article was also written during this thesis. However, for the forthcoming chapters we will focus only on cG methods.

[5] S. Badia J. Bonilla, and A. Hierro, *Differentiable monotonicity-preserving schemes for discontinuous Galerkin methods on arbitrary meshes,* Computer Methods in Applied Mechanics and Engineering **320** (2017) 582–605.

Moreover the developments described in this thesis have been presented in the following international conferences.

2016 S. Badia* and J. Bonilla, *Monotonicity preserving nonlinear stabilization for hyperbolic scalar problems,* Conference on the Mathematics of Finite Elements and Applications. Uxbridge, England.

2017 S. Badia* and J. Bonilla, *Finite element methods preserving maximum principles,* Finite Elements in Fluids Conference. Rome, Italy.

2017 S. Badia and J. Bonilla*, *Monotonicity preserving finite element methods for scalar convection–diffusion problems,* European Workshop on High Order Nonlinear Numerical Methods for Evolutionary PDEs. Stuttgart, Germany.

2017 J. Bonilla* and S. Badia, *High-order monotonicity preserving finite element methods for scalar convection–diffusion problems,* European Conference on Numerical Mathematics and Advanced Applications. Voss, Norway.

2019 J. Bonilla* and S. Badia, *Monotonicity preserving stabilization for convection dominated flows,* International Congress on Industrial and Applied Mathematics. Valencia, Spain

It is worth mentioning that during the course of this doctoral studies the author performed a six-months research stay at Sandia National Laboratories. Among other tasks, the author could collaborate with Prof. Shadid and his team, which led to the following proceeding as well as the journal article [6].

2018 J. Bonilla, S. Mabuza, J.N. Shadid, and S. Badia, *On differentiable linearity and local bounds preserving stabilization methods for first order conservation law systems,* (2018).

Chapter 2

Monotonicity preserving stabilization for linear FEs

This chapter is focused on a nonlinear stabilization technique for scalar conservation laws with implicit time stepping. The method relies on an artificial diffusion method, based on a graph-Laplacian operator. It is nonlinear, since it depends on a shock detector. Further, the resulting method is linearity preserving. The same shock detector is used to gradually lump the mass matrix. The resulting method is LED, positivity preserving, and also satisfies a global DMP. Lipschitz continuity has also been proved. However, the resulting scheme is highly nonlinear, leading to very poor nonlinear convergence rates. We propose a smooth version of the scheme, which leads to twice differentiable nonlinear stabilization schemes. It allows one to straightforwardly use Newton's method and obtain quadratic convergence. In the numerical experiments, steady and transient linear transport, and transient Burgers' equation have been considered in 2D. Using the Newton method with a smooth version of the scheme we can reduce 10 to 20 times the number of iterations of Anderson acceleration with the original non-smooth scheme. In any case, these properties are only true for the converged solution, but not for iterates. In this sense, we have also proposed the concept of projected nonlinear solvers, where a projection step is performed at the end of every nonlinear iterations onto a FE space of admissible solutions. The space of admissible solutions is the one that satisfies the desired monotonic properties (maximum principle or positivity).

2.1 Introduction

Many PDEs satisfy some sort of maximum principle or positivity property. However, numerical discretizations usually violate these structural properties at the discrete level, with implications in terms of accuracy and stability, e.g., leading to non-physical local oscillations.

It is well-understood now how to build methods that satisfy some sort of DMP based on explicit time integration combined with FVM or discontinuous Galerkin (dG) schemes [27, 68]. However, implicit time integration is preferred in problems with multiple scales in time when the fastest scales are not relevant. E.g., under-resolved simulations of

multi-scale problems in time are essential in plasma physics [54]. Unfortunately, implicit DMP-preserving hyperbolic solvers are scarce and not so well developed.

In the frame of FE discretizations, the local instabilities present in the solution of hyperbolic problems have motivated the use of so-called shock capturing schemes based on artificial diffusion (see, e.g., [49]). These methods introduce nonlinear stabilization, in contrast with classical SUPG-type linear stabilization techniques [47, 48]. Since linear schemes are at most first-order accurate and highly dissipative [36], recent research on FE techniques for conservation laws has focused on the development of less dissipative nonlinear schemes. Many of these ideas come from the numerical approximation of convection dominated CDR, where one encounters similar issues. The cornerstone of these methods is the design of a nonlinear artificial diffusion that vanishes in smooth regions and works on discontinuities or sharp layers. Many residual-based diffusion methods have been considered so far (see, e.g., [32] and references therein). Most of these approaches have failed to reach DMP-preserving methods. A salient exception is the method by Burman and Ern [25], which satisfies a DMP under mesh restrictions. Recently, due to some interesting novel approaches in the field, the state-of-the-art in nonlinear stabilization has certainly advanced [7, 13, 14, 24, 26, 64, 65].

Implicit FE schemes for hyperbolic problems rely on four key ingredients:

1. The first ingredient is the definition of the *shock detector* that only activates the nonlinear diffusion around shocks/discontinuities. Recent nonlinear stabilization techniques have been developed based on shock detectors driven by gradient jumps [7, 23] or edge differences [13, 64, 65]. The use of such schemes was proposed in [23] for 1D problems and extended to multiple dimensions in [7]. A salient property of the scheme in [7] is that it is DMP-preserving, but it relies on the DMP of the Poisson operator, which is only true under stringent constraints on the mesh. Another salient feature of the gradient-jump diffusion approach in [7] is the fact that it leads to so-called linearity preserving methods, i.e., the artificial diffusion vanishes for first order polynomials. This property is related to high-order convergence on smooth regions [66]. A modification of the nonlinear diffusion in [64] that also satisfies this property is proposed in [65].

2. The second ingredient is the *amount of diffusion* to be introduced on shocks, which is the amount of diffusion introduced in a first order linear scheme. In this sense, one can consider flux-corrected transport techniques [67].

3. The third ingredient is the form of the *discrete viscous operator*. In order to keep the DMP on arbitrary meshes, Guermond and Nazarov have proposed to use graph-theoretic, instead of PDE-based, operators for the artificial diffusion terms. This approach has been used in [13, 93] (for the steady-state convection–diffusion–reaction problem) and in [39] (for linear conservation laws) combined with artificial diffusion definitions similar to the one in [38].

4. The fourth ingredient is the *perturbation of the mass matrix*, in order to satisfy a DMP. Full mass lumping is one choice, but it introduces an unacceptable phase error. For continuous FE methods, improved techniques can be found in [40]. Alternatively, limiting-type strategies are used, e.g., in [64, 65].

5. The method in [13] is Lipschitz continuous, which is needed for the well-posedness of the resulting nonlinear scheme. However, in practice, all the methods presented above are still highly nonlinear, and nonlinear convergence becomes very hard and expensive. It leads to a fifth additional ingredient that has not been considered so far in much detail. In order to reduce the computational cost of these schemes, we consider the *smoothing* of the nonlinear artificial diffusion, to make it differentiable up to some fixed order. The possibility to define smooth nonlinear schemes can improve the nonlinear convergence of the methods and make them practical for realistic applications. Further, the smoothing step enables advanced linearization strategies based on Newton's method. It also involves the development of efficient nonlinear solvers, e.g., based on the combination of Newton, line search, and/or Anderson acceleration techniques.

All the results commented above are restricted to linear (or bilinear) FEs. We are not aware of the existence of high-order implicit DMP-preserving FE schemes. For explicit time integration and limiters, second order methods can be found in [39]. The use of hp-adaptive schemes that keep first order schemes around shocks has been proposed in [44].

In this chapter, we propose a novel nonlinear stabilization method that satisfies a DMP, positivity, and LED properties at the discrete level. It combines: (1) a novel shock detector related to the one in [7], which is simple and linearity preserving; (2) the graph-Laplacian artificial viscous term proposed in [38]; (3) an edge FCT-type definition of the amount of diffusion (see [64]); (4) a novel gradual mass lumping technique that exploits the same shock detector used for the artificial diffusion. We prove that the resulting method ticks all the boxes, i.e., it is total variation diminishing (TVD), DMP, positivity-preserving, linearity preserving, Lipschitz continuous, and introduces low dissipation. With regard to the last point, we prove that the amount of diffusion is the minimum needed in our analysis to prove the DMP. Further, we consider a novel approach to design a smoothed version of the resulting scheme that is twice differentiable. We prove that linear preservation is weakly enforced in this case, but all the other properties remain unchanged. Finally, we analyze the effect of the smoothing in the computational cost, and observe a clear reduction in the CPU cost of the nonlinear solver when using the smooth version of the method proposed herein while keeping almost unchanged the sharp layers of the non-smooth version. Future work will be focused on the entropy stability analysis of these schemes for nonlinear scalar conservation laws. A partial result in this direction is the proof of entropy stability for a related method when applied to the 1D Burger's equations (see [23]).

This chapter is structured as follows. In Sect. 2.2 the continuous problem and its discretization using the FE method are presented. Sect. 2.3 contains the formulation of a novel nonlinear stabilization method. Sect. 2.4 is devoted to the monotonicity analysis of the proposed method. An alternative approach is presented in Sect. 2.5. Lipschitz continuity of the methods is proved in Sect. 2.6. A differentiable version the previous method is presented in Sect. 2.7. Sect. 2.8 is devoted to nonlinear solvers. Different numerical experiments are introduced in Sect. 2.9. Finally, in Sect. 2.10 we draw some conclusions.

2.2 Preliminaries

2.2.1 The continuous problem

Let $\Omega \subset \mathbb{R}^d$ be a bounded domain, where d is the space dimension, and $(0,T]$ the time interval. The scalar conservation equation reads: find $u(\boldsymbol{x},t)$ such that

$$\partial_t u + \boldsymbol{\nabla}\cdot\boldsymbol{f}(u) = g, \quad \text{on } \Omega\times(0,T],$$

where $\boldsymbol{f}\in \mathrm{Lip}(\mathbb{R};\mathbb{R}^d)$ is the flux. It is also subject to the initial condition $u(\boldsymbol{x},0) = u_0 \in L^\infty(\Omega)$ and boundary condition $u(\boldsymbol{x},t) = \overline{u}(\boldsymbol{x},t)$ on the inflow $\Gamma_{\text{in}} \doteq \{(\boldsymbol{x},t)\in\partial\Omega\times(0,T] \mid \boldsymbol{f}(\boldsymbol{x},t)\cdot\boldsymbol{n}(\boldsymbol{x},t) < 0\}$. There exist a unique entropy solution u of the above problem that satisfies the entropy inequalities $\partial_t E(u) + \boldsymbol{\nabla}\cdot\boldsymbol{F}(u) \le 0$ for all convex entropies $E\in\mathrm{Lip}(\mathbb{R};\mathbb{R})$ with its associated entropy fluxes $\boldsymbol{F}_i(u) = \int_0^u E'(v)\boldsymbol{f}_i'(v)\,\mathrm{d}v$, $1\le i\le d$ (see Kružkov [55]). Let us consider the weak form of this problem consists in seeking u such that $u=\overline{u}$ on $\Gamma_{\text{in}}\times(0,T]$ and

$$(\partial_t u, v) + (\boldsymbol{f}'(u)\cdot\boldsymbol{\nabla}u, v) = (g,v) \quad \forall v\in L^2(\Omega), \tag{2.1}$$

almost everywhere in $(0,T]$, with $g\in L^2(\Omega)$.

2.2.2 Finite element spaces and meshes

Let $\mathcal{T}_h$ be a conforming partition of Ω into elements, K. Elements can be triangles or quadrilaterals for $d=2$, or tetrahedrals or hexahedra for $d=3$. The set of interpolation nodes of $\mathcal{T}_h$ is represented by $\mathcal{N}_h$, whereas $\mathcal{N}_h(K)$ denotes the set of nodes belonging to element $K\in\mathcal{T}_h$. Moreover, Ω_i is the macroelement composed by the union of the elements $K\in\mathcal{T}_h$ such that $i\in\mathcal{N}_h(K)$. $\mathcal{N}_h(\Omega_i)$ denotes the set of nodes in that macroelement. The continuous linear FE space is defined as

$$V_h \doteq \left\{v_h\in\mathcal{C}^0(\Omega):\ v_h|_K\in P_k(K)\ \ \forall K\in\mathcal{T}_h\right\}$$

for triangular or tetrahedral elements (replacing $P_1(K)$ by $Q_1(K)$ for quadrilateral or hexahedral elements). $P_1(K)$ (resp., $Q_1(K)$) is the space of polynomials with total (resp.,

partial) degree less or equal to 1. The nodal basis of V_h is written $\{\varphi_i\}_{i\in\mathcal{N}_h}$, and the FE functions can be expressed as $v_h = \sum_{i\in\mathcal{N}_h} \varphi_i v_i$, where v_i is the value of v_h at node i.

2.2.3 The semi-discrete problem

The semi-discrete Galerkin FE approximation of (2.1) reads: find $u_h \in V_h$ such that $u_h(\Gamma_{\text{in}}, t) = \pi_h(\overline{u})$ and

$$(\partial_t u_h, v_h) + (\boldsymbol{f}'(u_h)\cdot\boldsymbol{\nabla} u_h, v_h) = (g, v_h) \quad \forall v_h \in V_h, \tag{2.2}$$

for $t \in (0, T]$, with initial conditions $u_h(\cdot, 0) = \pi_h(u_0)$. π_h denotes a FE interpolation, e.g., the Scott-Zhang projector [83].

Using the notation $\mathbf{M}u_h \doteq (u_h, \cdot)$ and $\mathbf{F}(w_h)u_h \doteq (\boldsymbol{f}'(w_h)\cdot\boldsymbol{\nabla} u_h, \cdot)$ we can write problem (2.2) in compact form as

$$\mathbf{M}\partial_t u_h + \mathbf{F}(u_h)u_h = g \tag{2.3}$$

in V_h', i.e., the dual space of V_h. Further, we define $\mathbf{M}_{ij} \doteq (\varphi_j, \varphi_i)$, $\mathbf{F}_{ij}(u_h) \doteq (\boldsymbol{f}'(u_h)\cdot\boldsymbol{\nabla}\varphi_j, \varphi_i)$, and $g_i \doteq (g, \varphi_i)$.

In order to carry out the time discretization of (2.3), let us consider a partition of the time domain $(0, T]$ into sub-intervals $(t^n, t^{n+1}]$, with $0 \doteq t^0 < t^1 < \ldots < t^N \doteq T$. We consider the Backward-Euler (BE) implicit time integrator to keep at the time-discrete level the monotonicity properties of the semi-discrete problem, leading to the discrete problem: given $u_h^0 \doteq \pi_h(u_0) \in V_h$, compute for $n = 1, \ldots, N-1$

$$\mathbf{M}\delta_t u_h^{n+1} + \mathbf{F}(u_h^{n+1})u_h^{n+1} = g \quad \text{in } V_h', \tag{2.4}$$

where $\delta_t u_h^{n+1} \doteq \Delta t_{n+1}^{-1}(u_h^{n+1} - u_h^n)$, and $\Delta t_{n+1} \doteq |t^{n+1} - t^n|$. Implicit strong stability preserving Runge-Kutta methods [53] also preserve the monotonic properties at the discrete level [53], under some restrictions on the time step size. For the sake of brevity we consider the BE scheme.

Systems (2.3) and (2.4) will be supplemented with additional stabilization terms to minimize the oscillations generated by the Galerkin FE approximation. Of particular interest are methods which provide solutions that satisfy the following property for all nodes, for zero forcing terms.

Definition 2.2.1 (Local DMP)**.** *A solution $u \in V_h$ satisfies the local DMP if*

$$u_i^{\min} \le u_i \le u_i^{\max}, \quad \textit{where} \;\; u_i^{\max} \doteq \max_{j\in\mathcal{N}_h(\Omega_i)\setminus\{i\}} u_j, \quad u_i^{\min} \doteq \min_{j\in\mathcal{N}_h(\Omega_i)\setminus\{i\}} u_j.$$

Actually, for steady problems, if this is satisfied for all $i \in \mathcal{N}_h$, then the extrema will be at the boundary and there exist no local extrema.

Furthermore, it is useful to define *local extremum diminishing* (LED) methods for transient problems.

Definition 2.2.2 (LED). *A method is called LED if for $g = 0$ and any time in $(0, T]$, the solution satisfies*

$$\mathrm{d}_t u_i \leq 0 \text{ if } u_i \text{ is a maximum and } \quad \mathrm{d}_t u_i \geq 0 \text{ if } u_i \text{ is a minimum.}$$

For time-discrete methods, the same definition applies, replacing d_t by δ_t.

2.3 Nonlinear stabilization

We want to design a linearity preserving LED method for stabilizing the scalar semi-discrete hyperbolic problem (2.3) (or the discrete problem (2.4)), described in the previous section. As written above, this method is based on a graph-theoretic approach. Let us consider a nonlinear stabilization operator $\mathbf{B}(u_h) : V_h \to V_h'$ and denote $\mathbf{B}_{ij}(u_h) \doteq \langle \mathbf{B}(u_h)\varphi_j, \varphi_i \rangle$. Particularly, we require that the stabilization term will satisfy the following properties (see also [38]):

1. compact support: $\mathbf{B}_{ij}(u_h) = 0$ if $j \notin \mathcal{N}_h(\Omega_i)$ for any $u_h \in V_h$,
2. symmetry: $\mathbf{B}_{ij}(u_h) = \mathbf{B}_{ji}(u_h)$ for any $u_h \in V_h$,
3. conservation: $\sum_{j \neq i} \mathbf{B}_{ij}(u_h) = -\mathbf{B}_{ii}(u_h)$ for any $u_h \in V_h$,
4. linear preservation: $\mathbf{B}(u_h) = 0$ for any $u_h \in P_1(\Omega)$.

To achieve this properties we define the nonlinear stabilization term

$$\langle \mathbf{B}(w_h)u_h, v_h \rangle \doteq \sum_{i \in \mathcal{N}_h} \sum_{j \in \mathcal{N}_h(\Omega_i)} \nu_{ij}(w_h) v_i u_j \ell(i,j), \qquad u_h,\ v_h \in V_h, \tag{2.5}$$

where the graph-theoretic Laplacian is defined as $\ell(i,j) \doteq 2\delta_{ij} - 1$, and the artificial diffusion computed as

$$\begin{aligned} \nu_{ij}(w_h) &\doteq \max\{\alpha_i(w_h)\mathbf{F}_{ij}(w_h), \alpha_j(w_h)\mathbf{F}_{ji}(w_h), 0\} \qquad \text{for } i \neq j, \\ \nu_{ii}(w_h) &\doteq \sum_{\substack{j \in \mathcal{N}_h(\Omega_i) \\ j \neq i}} \nu_{ij}(w_h), \end{aligned} \tag{2.6}$$

where $\alpha_i(\cdot)$ is the shock detector. We note that this choice leads to a symmetric stabilization operator $\mathbf{B}(w_h)$. In order to define the shock detector, let us introduce some notation. Let $i \in \mathcal{N}_h$ be a node of the mesh, $\boldsymbol{v}$ a vector field, and w a scalar field. Let $\boldsymbol{r}_{ij} = \boldsymbol{x}_j - \boldsymbol{x}_i$ be the vector pointing from nodes i to j in $\mathcal{N}_h$ and $\hat{\boldsymbol{r}}_{ij} \doteq \frac{\boldsymbol{r}_{ij}}{|\boldsymbol{r}_{ij}|}$. Let $\boldsymbol{x}_{ij}^{\mathrm{sym}}$ be the point at the intersection between the line that passes through $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and $\partial\Omega_i$ that is not $\boldsymbol{x}_j$ (see Fig. 2.1). The set of all symmetric nodes with respect to node i is represented with $\mathcal{N}_h^{\mathrm{sym}}(\Omega_i)$. We define $\boldsymbol{r}_{ij}^{\mathrm{sym}} \doteq \boldsymbol{x}_{ij}^{\mathrm{sym}} - \boldsymbol{x}_i$, and $u_j^{\mathrm{sym}} \doteq u_h(\boldsymbol{x}_{ij}^{\mathrm{sym}})$. Then, one can define the jump and the mean of the unknown gradient at node i in direction

$\boldsymbol{r}_{ij}$ as

$$[\![\boldsymbol{\nabla} u_h]\!]_{ij} \doteq \frac{u_j - u_i}{|\boldsymbol{r}_{ij}|} + \frac{u_j^{\text{sym}} - u_i}{|\boldsymbol{r}_{ij}^{\text{sym}}|},$$
$$\{\!\{|\boldsymbol{\nabla} u_h \cdot \hat{\boldsymbol{r}}_{ij}|\}\!\}_{ij} \doteq \frac{1}{2}\left(\frac{|u_j - u_i|}{|\boldsymbol{r}_{ij}|} + \frac{|u_j^{\text{sym}} - u_i|}{|\boldsymbol{r}_{ij}^{\text{sym}}|}\right).$$

We note that the symmetric nodes and their corresponding values u_j^{sym} are used in the proof of the following results, Lemma 2.3.2, and Theorem 2.6.1, but *not required in the implementation* of (2.11). For triangular or tetrahedral meshes, since $\boldsymbol{\nabla} u_h$ is constant, u_j^{sym} can be computed easily as

$$u_j^{\text{sym}} = u_h(\boldsymbol{x}_i) + \boldsymbol{\nabla} u_h(\boldsymbol{x}_i) \cdot \boldsymbol{r}_{ij}^{\text{sym}}.$$

For quadrilateral or hexahedral structured (possibly adapted and nonconforming) meshes, u_j^{sym} is also easy to obtain since j^{sym} is already in $\mathcal{N}_h(\Omega_i)$. It also applies for symmetric meshes, when a mesh is said to be symmetric with respect to its internal nodes if for any $i \in \mathcal{N}_h$ all symmetric nodes $j^{\text{sym}} \in \mathcal{N}_h^{\text{sym}}(\Omega_i)$ already belong to $\mathcal{N}_h(\Omega_i)$.

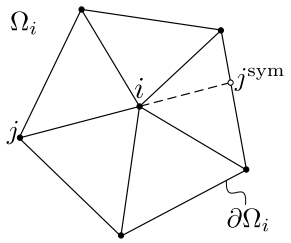


FIGURE 2.1: Representation of the symmetric node j^{sym} of j with respect to i.

Making use of these definitions, the proposed shock detector at node $i \in \mathcal{N}_h$ for a FE solution u_h reads:

$$\alpha_i(u_h) \doteq \begin{cases} \left[\dfrac{\left|\sum_{j\in\mathcal{N}_h(\Omega_i)} [\![\boldsymbol{\nabla} u_h]\!]_{ij}\right|}{\sum_{j\in\mathcal{N}_h(\Omega_i)} 2\{\!\{|\boldsymbol{\nabla} u_h \cdot \hat{\boldsymbol{r}}_{ij}|\}\!\}_{ij}}\right]^q & \text{if } \sum_{j\in\mathcal{N}_h(\Omega_i)} \{\!\{|\boldsymbol{\nabla} u_h \cdot \hat{\boldsymbol{r}}_{ij}|\}\!\}_{ij} \neq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{2.7}$$

for some $q \in \mathbb{R}^+$. We note that this shock detector is motivated from [7], where the directional nodal-wise jumps and mean values are first used for such purposes. For triangular or tetrahedral meshes, the only difference strives in the fact that the supremum over all $j \in \mathcal{N}_h(\Omega_i)$ in both the numerator and denominator was used in [7] instead of the sum. In the next lemma we show that in fact (2.7) detects extrema.

Lemma 2.3.1. *The shock detector $\alpha_i(u_h)$ defined in (2.7) is equal to 1 if u_h has an extremum at point $\boldsymbol{x}_i$. Otherwise, $\alpha_i(u_h) < 1$ in general, and $\alpha_i(u_h) = 0$ for $q = \infty$.*

Proof. Using the fact that u_h has an extremum at $\boldsymbol{x}_i$,

$$\left|\sum_{j\in\mathcal{N}_h(\Omega_i)} [\![\boldsymbol{\nabla} u_h]\!]_{ij}\right| = \left|\sum_{j\in\mathcal{N}_h(\Omega_i)} \frac{u_j-u_i}{|\boldsymbol{r}_{ij}|} + \frac{u_j^{\mathrm{sym}}-u_i}{|\boldsymbol{r}_{ij}^{\mathrm{sym}}|}\right|$$
$$= \sum_{j\in\mathcal{N}_h(\Omega_i)} \frac{|u_j-u_i|}{|\boldsymbol{r}_{ij}|} + \frac{\left|u_j^{\mathrm{sym}}-u_i\right|}{|\boldsymbol{r}_{ij}^{\mathrm{sym}}|} = \sum_{j\in\mathcal{N}_h(\Omega_i)} 2\left\{\!\!\left\{|\boldsymbol{\nabla} u_h\cdot\hat{\boldsymbol{r}}_{ij}|\right\}\!\!\right\},$$

since u_j-u_i has the same sign (or it is equal to zero) in all directions. It proves that $\alpha_i(u_h)=1$ on an extremum. In fact, if the solution does not have an extremum, these quantities neither can have the same sign nor be zero in all cases, and we only have

$$\left|\sum_{j\in\mathcal{N}_h(\Omega_i)} [\![\boldsymbol{\nabla} u_h]\!]_{ij}\right| < \sum_{j\in\mathcal{N}_h(\Omega_i)} \frac{|u_j-u_i|}{|\boldsymbol{r}_{ij}|} + \frac{\left|u_j^{\mathrm{sym}}-u_i\right|}{|\boldsymbol{r}_{ij}^{\mathrm{sym}}|} = \sum_{j\in\mathcal{N}_h(\Omega_i)} 2\left\{\!\!\left\{|\boldsymbol{\nabla} u_h\cdot\hat{\boldsymbol{r}}_{ij}|\right\}\!\!\right\}. \quad (2.8)$$

Hence, $\alpha_i(u_h)<1$ when there is no extremum at $\boldsymbol{x}_i$. Moreover, for $q=\infty$, the shock detector vanishes in all the nodes that are not extrema. □

In addition to the nonlinear stabilization term $\mathbf{B}(u_h)$, it is necessary to do a mass matrix lumping to prove that the LED property is satisfied. In the numerical analysis, it is enough to make this approximation when testing against the shape functions corresponding to nodes related to extrema, which is identified by the shock detector. Therefore, we propose the following stabilized semi-discrete version of (2.2):

$$\begin{aligned}(1-\alpha_i(u_h))(\partial_t u_h,\varphi_i) + \alpha_i(u_h)(\partial_t u_i,\varphi_i) + (\boldsymbol{f}'(u_h)\cdot\boldsymbol{\nabla} u_h,\varphi_i)\\ + \sum_{j\in\mathcal{N}_h(\Omega_i)} \nu_{ij}(u_h)v_i u_j l(i,j) = (g,\varphi_i) \quad \text{for any } i\in\mathcal{N}_h,\end{aligned} \quad (2.9)$$

with the definition of the shock detector (2.7) and the nonlinear artificial diffusion (2.6). Thus, the definition of the mass matrix is nonlinear

$$\mathbf{M}_{ij}(u_h) \doteq (1-\alpha_i(u_h))(\varphi_j,\varphi_i) + \alpha_i(u_h)(\delta_{ij},\varphi_i). \quad (2.10)$$

It can be understood as a mass matrix with gradual lumping. Full lumping is only attained at extrema. Denoting $\mathbf{K}(u_h)\doteq\mathbf{F}(u_h)+\mathbf{B}(u_h)$, the stabilized problem (2.9) can be expressed in compact form as

$$\mathbf{M}(u_h)\mathrm{d}_t u_h + \mathbf{K}(u_h)u_h = g \quad \text{in } V_h'. \quad (2.11)$$

Analogously for the discrete problem (2.4),

$$\mathbf{M}(u_h^{n+1})\delta_t u_h^{n+1} + \mathbf{K}(u_h^{n+1})u_h^{n+1} = g^{n+1} \quad \text{in } V_h'. \quad (2.12)$$

Finally, let us note that the shock detector (2.7) leads to the one of Barrenechea and co-workers [13],

$$\tilde{\alpha}_i \doteq \begin{cases} \left(\frac{\left| \sum_{j \in \mathcal{N}_h(\Omega_i)} u_i - u_j \right|}{\sum_{j \in \mathcal{N}_h(\Omega_i)} |u_i - u_j|} \right)^q & \text{if } \sum_{j \in \mathcal{N}_h(\Omega_i)} |u_i - u_j| \neq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{2.13}$$

when restricted to symmetric meshes of equilateral triangles.

Lemma 2.3.2. *For a symmetric triangular mesh where all the edges have the same length, α_i in* (2.7) *is identical to $\tilde{\alpha}_i$ in* (2.13)*.*

Proof. For symmetric meshes, for every $j \in \mathcal{N}_h(\Omega_i)$, $j^{\text{sym}} \in \mathcal{N}_h(\Omega_i)$. So, we can group nodes in $\mathcal{N}_h(\Omega_i)$ in pairs, getting

$$2 \sum_{j \in \mathcal{N}_h(\Omega_i)} (u_i - u_j) = \sum_{j \in \mathcal{N}_h(\Omega_i)} (u_i - u_j + u_i - u_{ij}^{\text{sym}}).$$

We proceed analogously for the mean value. Further, since $\boldsymbol{r}_{ij}$ is identical for all $j \in \mathcal{N}_h(\Omega_i)$ by assumption, we get

$$\frac{\left| \sum_{j \in \mathcal{N}_h(\Omega_i)} [\![\boldsymbol{\nabla} u_h]\!]_{ij} \right|}{2 \sum_{j \in \mathcal{N}_h(\Omega_i)} \{\!\{ |\boldsymbol{\nabla} u_h \cdot \hat{\boldsymbol{r}}_{ij}| \}\!\}_{ij}} = \frac{\left| \sum_{j \in \mathcal{N}_h(\Omega_i)} u_i - u_j \right|}{\sum_{j \in \mathcal{N}_h(\Omega_i)} |u_i - u_j|}.$$

□

For arbitrary symmetric meshes the methods only differ on the weights of the terms in the sums in (2.7) and all the required properties stated in (2.14) are readily satisfied for the use of the shock detector in (2.13). In general meshes, the shock detectors are different, and the one in (2.13) is not linearity preserving.

2.4 Monotonicity properties

In the sequel, we prove that the scheme (2.9) is LED. First, we define a set of necessary conditions on the nonlinear discrete operators that lead to LED schemes. They are the nonlinear extension of the ones for linear systems (see, e.g., [64]).

Theorem 2.4.1. *The semi-discrete problem* (2.11) *is LED if $g(\boldsymbol{x}) = 0$ in Ω and, for every node $i \in \mathcal{N}_h$ such that u_i is a local extremum, it holds:*

$$\begin{aligned} &\mathbf{M}_{ij}(u_h) \doteq \delta_{ij} m_i, \text{ with } m_i > 0, \\ &\mathbf{K}_{ij}(u_h) \leq 0 \ \ \forall i \neq j, \text{ and } \sum_{j \in \mathcal{N}_h(\Omega_i)} \mathbf{K}_{ij}(u_h) = 0. \end{aligned} \tag{2.14}$$

Moreover, for $g(\boldsymbol{x}) \leq 0$ (resp. $g(\boldsymbol{x}) \geq 0$) in Ω and for all $i \in \mathcal{N}_h$ such that u_i is a local maximum (resp. minimum), if (2.14) *holds the maximum (resp. minimum) is*

diminishing (resp. increasing). These results are also true for the discrete problem (2.12)*. Furthermore, the discrete problem* (2.12) *is positivity-preserving for* $g = 0$ *and* $u_0 \geq 0$*.*

Proof. Let us start proving the LED property. If u_i is a maximum, from (2.11), conditions in (2.14), and the fact that $\alpha_i(u_h) = 1$, we have:

$$g_i = m_i \mathrm{d}_t u_i + \sum_{j \in \mathcal{N}_h(\Omega_i)} \mathbf{K}_{ij}(u_h) u_j \geq m_i \mathrm{d}_t u_i + \sum_{j \in \mathcal{N}_h(\Omega_i)} \mathbf{K}_{ij}(u_h) u_i = m_i \mathrm{d}_t u_i,$$

for $m_i \doteq \int_\Omega \varphi_i \mathrm{d}\Omega$. As a result, $\mathrm{d}_t u_i \leq 0$ and thus LED. We proceed analogously for the minimum. The proof is analogous for the discrete problem with BE time integration.

Next, we prove positivity. Let us consider that at some time step m the solution becomes negative, and consider the node i in which the minimum value is attained. Using the previous result for a minimum at the discrete level, we have that $\delta_t u_i^m \geq 0$ and thus $u_i^m \geq u_i^{m-1}$. It leads to a contradiction, since $u_i^{m-1} \geq 0$. Thus, the solution must be positive at all times. □

Theorem 2.4.2 (LED)**.** *The semi-discrete (resp., discrete) problem* (2.11) *(resp.,* (2.12)*) leads to solutions* $u_h \in V_h$ *that enjoy the LED property in Def. 2.2.2 for any* $q \in \mathbb{R}^+$*.*

Proof. Assume u_h reaches an extremum on $i \in \mathcal{N}_h$. Then $\alpha_i(u_h) = 1$ and $\mathbf{M}_{ij}(u_h)\mathrm{d}_t u_j = m_i \mathrm{d}_t u_i$ with $m_i = \int_\Omega \varphi_i$. On the other hand, taking into account the definition of $\nu_{ij}(u_h)$ in (2.6), the convective term for $j \neq i$ reads

$$\mathbf{K}_{ij}(u_h) = \mathbf{F}_{ij}(u_h) - \max\{\mathbf{F}_{ij}(u_h), \alpha_j(u_h)\mathbf{F}_{ji}(u_h), 0\} \leq 0.$$

Using the fact that $\sum_{j \in \mathcal{N}_h(\Omega_i)} \mathbf{F}_{ij}(u_h) = (\boldsymbol{f}'(u_h) \cdot \boldsymbol{\nabla} 1, \varphi_i) = 0$, the definition of $\nu_{ii}(u_h)$, and (2.5), we have

$$\begin{aligned}
\mathbf{K}_{ii}(u_h) &= \mathbf{F}_{ii}(u_h) + \sum_{j \in \mathcal{N}_h(\Omega_i)\setminus\{i\}} \max\{\mathbf{F}_{ij}(u_h), \alpha_j(u_h)\mathbf{F}_{ji}(u_h), 0\} \\
&= \sum_{j \in \mathcal{N}_h(\Omega_i)\setminus\{i\}} -\mathbf{F}_{ij}(u_h) + \sum_{j \in \mathcal{N}_h(\Omega_i)\setminus\{i\}} \max\{\mathbf{F}_{ij}(u_h), \alpha_j(u_h)\mathbf{F}_{ji}(u_h), 0\} \\
&= -\sum_{j \in \mathcal{N}_h(\Omega_i)\setminus\{i\}} \mathbf{K}_{ij}(u_h).
\end{aligned}$$

Therefore it is clear that the conditions stated in Theorem 2.4.1 hold, thus the method is LED. The discrete case is proved analogously. □

Corollary 2.4.3 (DMP)**.** *The discrete problem* (2.12) *leads to solutions that satisfy the local DMP property in Def. 2.2.1 at every* t^n*, for* $n = 1, \ldots, N$*.*

Proof. If the maximum (resp., minimum) at time t^n is on a node whose value is not on the Dirichlet boundary, it is known from the LED property in Theorem 2.4.2 that it is bounded above (resp., below) by the maximum (resp., minimum) at the previous time step value. By induction, it will be bounded by the maximum (resp., minimum) at

$t = 0$. Alternatively, the maximum or minimum is on the Dirichlet boundary. It proves the result. □

Theorem 2.4.4. *The diffusion defined in* (2.6) *is the one that introduces the minimum amount of numerical dissipation* $\langle \mathbf{B}(u_h)u_h, u_h\rangle$ *required to satisfy* (2.14) *when* $q = \infty$.

Proof. Using the definition of the graph-Laplacian, the amount of dissipation introduced by the nonlinear stabilization is

$$\langle \mathbf{B}(u_h)u_h, u_h\rangle = \sum_{i\in\mathcal{N}_h}\sum_{j\in\mathcal{N}_h(\Omega)} \nu_{ij}(u_h)(u_i - u_j)^2.$$

Let us consider two connected nodes, i.e., $i, j \in \mathcal{N}_h$ and $j \in \mathcal{N}_h(\Omega_i)$. If neither i nor j are extrema, then $\alpha_i(u_h) = \alpha_j(u_h) = 0$ and $\nu_{ij} = 0$. Let us assume (without loss of generality) that u_h has an extremum at i. If $u_i = u_j$, the dissipation is independent of the expression for ν_{ij}. If $u_i > u_j$, $\alpha_j = 0$ (since $q = \infty$). Thus, $\nu_{ij} = -\max\{\mathbf{F}_{ij}(u_h), 0\}$. If $\mathbf{F}_{ij}(u_h) \le 0$, no dissipation is introduced. If $\mathbf{F}_{ij}(u_h) > 0$, then the diffusion introduced by the method is $-\mathbf{F}_{ij}(u_h)$ and $\mathbf{K}_{ij}(u_h) = 0$.

Let us assume that we have a method that is less dissipative than the one proposed herein. Based on the previous analysis, there exists a pair of connected nodes such that $u_i > u_j$ and the dissipation introduced is smaller than $-\mathbf{F}_{ij}(u_h)$, for $\mathbf{F}_{ij}(u_h) > 0$. As a result, $\mathbf{K}_{ij}(u_h) > 0$. Thus, the properties in (2.4.1) do not hold. It proves the theorem. □

Furthermore, it can be proved that the above method (2.11) (also (2.12)) is linearly preserving. In addition, using (2.13) instead, the method is still linearly preserving for symmetric meshes.

Theorem 2.4.5 (Linearity preservation)**.** *Let* u_h *be a continuous first order FE approximation of* $u \in P_1(\Omega)$*, then the semi-discrete and discrete problems* (2.11) *and* (2.12)*, respectively, are linearity preserving, in the sense that the Galerkin problem and the stabilized one are identical.*

Proof. If $u_h \in P_1(\Omega)$, then it is obvious that $\boldsymbol{\nabla} u_h$ is constant. Thus, $[\![\boldsymbol{\nabla} u_h]\!]_{ij} = 0$ for any direction $\boldsymbol{r}_{ij}$, and $\alpha_i(u_h) = 0$ for any $i \in \mathcal{N}_h$. Therefore, recalling (2.6), it is easy to see that $\nu_{ij} = 0$ for any $i, j \in \mathcal{N}_h$. Thus, the nonlinear stabilization and gradual lumping terms vanish and the Galerkin scheme is recovered. □

2.5 Symmetric mass matrix stabilization

The nonlinear mass matrix that has been considered in (2.10) is nonsymmetric by construction. In any case, we can easily consider a symmetric version of the method.

Another alternative strategy to the nonlinear mass matrix definition in (2.10) is to consider the fully discrete problem (2.12), keeping the mass matrix at the current time

step as a reaction term, leading to the following expression of the artificial diffusion

$$\begin{aligned}\tilde{\nu}_{ij}(w_h) &\doteq \nu_{ij}(w_h) + \tfrac{1}{\Delta t}\max\{\alpha_i \mathbf{M}_{ij}, 0, \alpha_j \mathbf{M}_{ji}\} \qquad \text{for } i \neq j,\\ \tilde{\nu}_{ii}(w_h) &\doteq \sum_{\substack{j\in\mathcal{N}_h(\Omega_i)\\ j\neq i}} \tilde{\nu}_{ij}.\end{aligned} \tag{2.15}$$

Let us consider another notion of DMP property.

Definition 2.5.1 (Global DMP)**.** *A solution satisfies the global DMP if given* $(\boldsymbol{x}, t)$ *in* $\Omega \times (0, T]$

$$\min_{(\boldsymbol{y},\bar{t})\in\Gamma} u(\boldsymbol{y},\bar{t}) \leq u(\boldsymbol{x}, t) \leq \max_{(\boldsymbol{y},\bar{t})\in\Gamma} u(\boldsymbol{y},\bar{t})$$

where $\Gamma \doteq \Omega \times \{0\} \cup \Gamma_{\text{in}}$.

It is easy to check that the global DMP is a consequence of the local DMP and LED properties.

It is possible to prove that the modified method with BE time integration satisfies the global DMP in Def. 2.5.1. Linear preservation can also be easily checked.

Theorem 2.5.2 (Global DMP)**.** *Let* u_h *be a continuous first order FE approximation of* u*. Then, the BE time discretization of problem* (2.2) *with* $g = 0$*, stabilized with* (2.5)*, and using* (2.15) *as artificial diffusion, satisfies the global DMP property in Def.* (2.5.1) *for any* $q \in \mathbb{R}^+$.

Proof. Let us denote by $\mathbf{K}(u)$ and $\tilde{\mathbf{K}}(u)$ the stabilized matrix with the artificial diffusion computed with (2.6) and (2.15), respectively. Assume u_h reaches a maximum on $\boldsymbol{x}_i \in \Omega\backslash\Gamma_{\text{in}}$. Then $\alpha_i = 1$, and we have:

$$\mathbf{M}_{ij}(u_h)u_j + \tilde{\mathbf{K}}_{ij}(u_h)u_j = m_i u_i + \mathbf{K}_{ij}(u_h)u_j,$$

where we have used the fact that $\max\{\alpha_i \mathbf{M}_{ij}, 0, \alpha_j \mathbf{M}_{ji}\} = \mathbf{M}_{ij}$. Thus, the equation related to the test function φ_i leads to

$$\sum_{j\in\mathcal{N}_h(\Omega_i)} \frac{\mathbf{M}_{ij}}{m_i} u_j^n = u_i^{n+1} + \sum_{j\in\mathcal{N}_h(\Omega_i)} \frac{\mathbf{K}_{ij}(u_h)}{m_i} u_j^{n+1} \geq u_i^{n+1} + \sum_{j\in\mathcal{N}_h(\Omega_i)} \frac{\mathbf{K}_{ij}(u_h)}{m_i} u_i^{n+1} = u_i^{n+1}.$$

Note that $\frac{\mathbf{M}_{ij}}{m_i} > 0$, and $\sum_{j\in\mathcal{N}_h(\Omega_i)} \frac{\mathbf{M}_{ij}}{m_i} = 1$. Hence u_i^{n+1} is smaller or equal to a convex combination of u_j^n, for $j \in \mathcal{N}_h(\Omega_i)$, and thus it is bounded above by the largest of these values. As a result, $u_h^{n+1}(\boldsymbol{x}) \leq \max\{\max_{\boldsymbol{y}\in\Omega} u_h^n(\boldsymbol{y}), \max_{(\boldsymbol{y},t^{n+1})\in\Gamma_{\text{in}}} u_D(\boldsymbol{y}, t^{n+1})\}$. Using a recursion argument, we prove the upper bound. We proceed analogously for the case lower bound. It proves the theorem. □

2.6 Lipschitz continuity

In the next, we want to prove the Lipschitz continuity of the nonlinear operator at every time step, i.e., $\mathsf{T} : V_h \to V_h'$ defined as

$$\mathsf{T}(u_h) \doteq \Delta t_{n+1}^{-1}\mathbf{M}(u_h)u_h + \mathbf{K}(u_h)u_h - g - \Delta t_{n+1}^{-1}\mathbf{M}(u_h)u_h^n.$$

In order to prove the Lipschitz continuity of $\mathsf{T}(\cdot)$, we must deal with the nonlinear stabilization and gradual mass lumping terms. The Galerkin terms can be handled using the fact that $\boldsymbol{f} \in \mathrm{Lip}(\mathbb{R};\mathbb{R}^d)$.

Let us introduce the following semi-norm generated by the graph-Laplacian operator

$$|w|_\ell \doteq \sqrt{\frac{1}{2}\sum_{i\in\mathcal{N}_h}\sum_{j\in\mathcal{N}_h(\Omega_i)}(w_i - w_j)^2}.$$

Further, we define $|\mathbf{v}|$ as the supremum of $|\boldsymbol{f}(v)|$ for $v \in V_h^{\mathrm{adm}}$, where $V_h^{\mathrm{adm}} \subset V_h$ is the subspace of functions that satisfy the global DMP in Def. 2.5.1.

Theorem 2.6.1. *Let us consider a non-degenerate partition $\mathcal{T}_h$. Given $u_h^n \in V_h$ and $g \in V_h'$, the nonlinear operators $\mathbf{B}(\cdot) : V_h \to V_h'$ and $\mathbf{M}(\cdot) : V_h \to V_h'$ are Lipschitz continuous in V_h^{adm} for $q \in \mathbb{N}^+$, since they satisfy*

$$\langle \mathbf{B}(u) - \mathbf{B}(v), z\rangle \le qh^{d-1}|\mathbf{v}||u-v|_\ell|z|_\ell, \qquad \textit{for any } z \in V_h,$$

$$\langle \mathbf{M}(u) - \mathbf{M}(v), z\rangle \le C(qh^{\frac{d}{2}}|u-v|_\ell + \|u-v\|)\|z\|, \qquad \textit{for any } z \in V_h.$$

Proof. We assume that the FE mesh is quasi-uniform in order to reduce technicalities. However, the proof for Lipschitz continuity can be extended to more general meshes. We denote $A = cB$ as $A \eqsim B$ and $A < cB$ as $A \lesssim B$, for any positive constant c that does not depend on the numerical or physical parameters.

From the definition of the nonlinear stabilization in (2.5), we get

$$\begin{aligned}|\langle \mathbf{B}(u)u, z\rangle - \langle \mathbf{B}(v)v, z\rangle| \le &\left|\sum_{i\in\mathcal{N}_h}\sum_{j\in\mathcal{N}_h(\Omega_i)}\nu_{ij}(v)\ell(i,j)(u_j - v_j)z_i\right| \\ &+ \left|\sum_{i\in\mathcal{N}_h}\sum_{j\in\mathcal{N}_h(\Omega_i)}(\nu_{ij}(u) - \nu_{ij}(v))\ell(i,j)u_j z_i\right|.\end{aligned} \tag{2.16}$$

Using the definition of $|\mathbf{v}|$, the Cauchy-Schwarz inequality, the fact that $\|\varphi_i\| \le Ch^{d/2}$, and the inverse inequality $\|\nabla v_h\| \lesssim h^{-1}\|v_h\|$ for $v_h \in V_h$ (see [21]), we get:

$$\mathbf{F}_{ij}(w) \le |\mathbf{v}|\|\boldsymbol{\nabla}\varphi_i\|_L^2\|\varphi_j\|_L^2 \lesssim h^{d-1}|\mathbf{v}|, \tag{2.17}$$

for any $w \in V_h^{\text{adm}}$. Using (2.17), the first term in the RHS of (2.16) is bounded as follows:

$$\left| \sum_{i \in \mathcal{N}_h} \sum_{j \in \mathcal{N}_h(\Omega_i)} \nu_{ij}(v) \ell(i,j)(u_j - v_j) z_i \right| \lesssim h^{d-1} |\mathbf{v}| |u - v|_\ell |z|_\ell.$$

The second term is bounded using the Cauchy-Schwarz inequality:

$$\begin{aligned} &\sum_{i \in \mathcal{N}_h} \sum_{j \in \mathcal{N}_h(\Omega_i)} (\nu_{ij}(u) - \nu_{ij}(v)) \ell(i,j) u_j z_i \qquad (2.18) \\ &\qquad \lesssim \left| \sum_{i \in \mathcal{N}_h} \sum_{j \in \mathcal{N}_h(\Omega_i)} \frac{1}{2} (\nu_{ij}(u) - \nu_{ij}(v))^2 (u_i - u_j)^2 \right|^{\frac{1}{2}} \times |z|_\ell. \end{aligned}$$

Using (2.17), we have:

$$\begin{aligned} &\nu_{ij}(u) - \nu_{ij}(v) \\ &= \max\{\alpha_i(u) \mathbf{F}_{ij}(u), \alpha_j(u) \mathbf{F}_{ji}(u), 0\} - \max\{\alpha_i(v) \mathbf{F}_{ij}(v), \alpha_j(v) \mathbf{F}_{ji}(v), 0\} \qquad (2.19) \\ &\leq \max\{(\alpha_i(u) \mathbf{F}_{ij}(u) - \alpha_i(v) \mathbf{F}_{ij}(v), \alpha_j(u) \mathbf{F}_{ji}(u) - \alpha_j(v) \mathbf{F}_{ji}(v), 0\} \\ &\lesssim h^{d-1} |\mathbf{v}| \max\{|\alpha_i(u) - \alpha_i(v)|, |\alpha_j(u) - \alpha_j(v)|\}. \end{aligned}$$

Let us assume that $\sum_{j \in \mathcal{N}_h(\Omega_i)} \{\!\{ |\boldsymbol{\nabla} u_h \cdot \boldsymbol{r}_{ij}| \}\!\}_{ij} \neq 0$. (The other case is straightforward.) On one hand, for a non-degenerate FE mesh, we have that $ch \leq \boldsymbol{r}_{ij} \leq Ch$, $j \in \mathcal{N}_h^{\text{sym}}(\Omega_i)$, for positive constants c, C that do not depend on h. Using this fact in the definition of the shock detector (2.7), we get:

$$\begin{aligned} \alpha_i(u)^{\frac{1}{q}} &= \frac{\left| \sum_{j \in \mathcal{N}_h(\Omega_i)} [\![\boldsymbol{\nabla} u_h]\!]_{ij} \right|}{\sum_{j \in \mathcal{N}_h(\Omega_i)} 2 \{\!\{ |\boldsymbol{\nabla} u_h \cdot \boldsymbol{r}_{ij}| \}\!\}_{ij}} = \frac{\left| \sum_{j \in \mathcal{N}_h(\Omega_i)} \frac{u_i - u_j}{|\boldsymbol{r}_{ij}|} + \frac{u_i - u_j^{\text{sym}}}{|\boldsymbol{r}_{ij}^{\text{sym}}|} \right|}{\sum_{j \in \mathcal{N}_h(\Omega_i)} \frac{|u_i - u_j|}{|\boldsymbol{r}_{ij}|} + \frac{|u_i - u_j^{\text{sym}}|}{|\boldsymbol{r}_{ij}^{\text{sym}}|}} \qquad (2.20) \\ &\eqsim \frac{\left| \sum_{j \in \mathcal{N}_h(\Omega_i)} (u_i - u_j) + (u_i - u_j^{\text{sym}}) \right|}{\sum_{j \in \mathcal{N}_h(\Omega_i)} |u_i - u_j| + |u_i - u_j^{\text{sym}}|}. \end{aligned}$$

Now, we use the following result for two sequences $\{a_i\}_{i=1}^n$ $\{b\}_{i=1}^n$ (see [13] for further details):

$$\begin{aligned} &\frac{|\sum_{i=1}^n a_i|}{\sum_{i=1}^n |a_i|} - \frac{|\sum_{i=1}^n b_i|}{\sum_{i=1}^n |b_i|} = \frac{|\sum_{i=1}^n a_i| - |\sum_{i=1}^n b_i|}{\sum_{i=1}^n |a_i|} + \sum_{i=1}^n |b_i| \left(\frac{1}{\sum_{i=1}^n |a_i|} - \frac{1}{\sum_{i=1}^n |b_i|} \right) \\ &\leq \frac{|\sum_{i=1}^n a_i - b_i|}{\sum_{i=1}^n |a_i|} + \frac{\sum_{i=1}^n |b_i| - \sum_{i=1}^n |a_i|}{\sum_{i=1}^n |a_i|} \leq \frac{|\sum_{i=1}^n a_i - b_i| + \sum_{i=1}^n |a_i - b_i|}{\sum_{i=1}^n |a_i|} \\ &\leq 2 \frac{\sum_{i=1}^n |a_i - b_i|}{\sum_{i=1}^n |a_i|}. \qquad (2.21) \end{aligned}$$

Using simple algebraic manipulation, we have $a^q - b^q = (a-b) \sum_{k=0}^{q-1} a^k b^{q-i} k$ for $q \in \mathbb{N}^+$. For $a, b \in [0,1]$, it leads to $|a^q - b^q| \leq q|a - b|$ (see [13]). This inequality, together with

(2.20) and (2.21), leads to:

$$\frac{1}{q}|\alpha_i(u)-\alpha_i(v)| \lesssim \frac{\left|\sum_{j\in\mathcal{N}_h(\Omega_i)}((u-v)i-(u-v)_j)+((u-v)_i-(u-v)_j^{\mathrm{sym}})\right|}{\sum_{j\in\mathcal{N}_h(\Omega_i)}|u_i-u_j|+|u_i-u_j^{\mathrm{sym}}|}. \quad (2.22)$$

On the other hand, the bounds

$$|u_i-u_j| \le \sum_{k\in\mathcal{N}_h(\Omega_i)}|u_i-u_k| \quad \text{and} \quad |u_i-u_j| \le \sum_{k\in\mathcal{N}_h(\Omega_j)}|u_j-u_k|,$$

(2.19), and (2.22), yield

$$\begin{aligned}(\nu_{ij}(u)-\nu_{ij}(v))(u_i-u_j) \lesssim & qh^{d-1}|\mathbf{v}| \sum_{k\in\mathcal{N}_h^{\mathrm{sym}}(\Omega_i)}|(u-v)_i-(u-v)_k| \\ & + qh^{d-1}|\mathbf{v}| \sum_{k\in\mathcal{N}_h^{\mathrm{sym}}(\Omega_j)}|(u-v)_j-(u-v)_k|.\end{aligned} \quad (2.23)$$

The second term is bounded by combining (2.18), (2.23), and the fact that the number of elements surrounding a node is bounded above independently of h:

$$\sum_{i\in\mathcal{N}_h}\sum_{j\in\mathcal{N}_h(\Omega_i)}(\nu_{ij}(u)-\nu_{ij}(v))\ell(i,j)u_jz_i \lesssim qh^{d-1}|\mathbf{v}||u-v|_\ell|z|_\ell.$$

Next, we have to prove that the nonlinear mass matrix is also Lipschitz continuous. First, we note that

$$\begin{aligned}&\sum_{j\in\mathcal{N}_h(\Omega_i)}(1-\alpha_i(u_h))(\varphi_j,\varphi_i)u_j+\alpha_i(u_h)(1,\varphi_i)u_i \\ &=\sum_{j\in\mathcal{N}_h(\Omega_i)}(\varphi_j,\varphi_i)u_j+\alpha_i(u_h)(\varphi_j,\varphi_i)(u_i-u_j).\end{aligned}$$

Thus

$$\begin{aligned}\langle\mathbf{M}(u)u,z\rangle-\langle\mathbf{M}(v)v,z\rangle \le & \sum_{i\in\mathcal{N}_h}\sum_{j\in\mathcal{N}_h(\Omega_i)}(\varphi_i,\varphi_j)(u_j-v_j)z_i \\ & +\sum_{i\in\mathcal{N}_h}\sum_{j\in\mathcal{N}_h(\Omega_i)}(\varphi_i,\varphi_j)(u_i-u_j)(\alpha_i(u_h)-\alpha_i(v_h))z_i \\ & +\sum_{i\in\mathcal{N}_h}\sum_{j\in\mathcal{N}_h(\Omega_i)}(\varphi_i,\varphi_j)((u+v)_i-(u+v)_j)\alpha_i(v_h)z_i.\end{aligned}$$

Bounds for the second and third term follow the same lines as above. For the second term, we proceed as in (2.18), getting:

$$\begin{aligned}
&\sum_{i\in\mathcal{N}_h}\sum_{j\in\mathcal{N}_h(\Omega_i)}(\varphi_i,\varphi_j)(u_i-u_j)(\alpha_i(u_h)-\alpha_i(v_h))z_i\\
&\lesssim\left|\sum_{i\in\mathcal{N}_h}\frac{1}{2}\sum_{j\in\mathcal{N}_h(\Omega_i)}(\varphi_i,\varphi_j)(\alpha_i(u_h)-\alpha_i(v_h))^2(u_i-u_j)^2\right|^{\frac{1}{2}}\times\|z\|\\
&\lesssim qh^{\frac{d}{2}}|u-v|_\ell\|z\|.
\end{aligned}$$

where we have used the spectral equivalence of the consistent and lumped mass matrices in the last inequality. The first and third term are easily bounded as

$$\begin{aligned}
&\sum_{i\in\mathcal{N}_h}\sum_{j\in\mathcal{N}_h(\Omega_i)}(\varphi_i,\varphi_j)(u_j-v_j)z_i\le\|u-v\|\|z\|,\\
&\sum_{i\in\mathcal{N}_h}\sum_{j\in\mathcal{N}_h(\Omega_i)}(\varphi_i,\varphi_j)((u+v)_i-(u+v)_j)\alpha_i(v_h)z_i\le qh^{\frac{d}{2}}|u-v|_\ell\|z\|.
\end{aligned}$$

It proves the theorem. □

2.7 Differentiable stabilization

The previous nonlinear system is Lipschitz continuous, which improves the convergence of the nonlinear iterations. In fact, assuming that we supplement (2.1) with a diffusive term, existence and uniqueness can be proved in the diffusive regime (see [13]). However, even using Anderson acceleration nonlinear convergence can be very hard (see [64, 65] and Sect. 2.9).

Based on these observations, we want to develop methods that lead to at least twice differentiable operators, i.e., $\frac{\partial^2\mathsf{T}(u_h)}{\partial^2 u_h}\in\mathcal{C}^0$, using the previous framework. This allows the usage of the Newton method to linearize the system, and reduces the required number of nonlinear iterations. Smoothness is achieved by substituting the non-differentiable functions of the previous formulation with smooth approximations.

In order to end up with a twice differentiable method, we propose to use the following artificial diffusion:

$$\begin{aligned}
&\nu_{ij}\doteq\max\nolimits_{\sigma_h}\left\{\max\nolimits_{\sigma_h}\left\{\alpha_{\varepsilon_h,i}(\mathbf{F}_{ij}(w_h)),\alpha_{\varepsilon_h,j}\mathbf{F}_{ji}(w_h)\right\},0\right\},\quad\text{for } i\neq j,\\
&\nu_{ii}\doteq\sum_{\substack{j\in\mathcal{N}_h(\Omega_i)\\ j\neq i}}\nu_{ij}.
\end{aligned}\tag{2.24}$$

The function $\max_{\sigma_h}(\cdot)$ is a regularized maximum function

$$\max\nolimits_{\sigma_h}\{x,y\}\doteq\frac{|x-y|_{1,\sigma}}{2}+\frac{x+y}{2},\tag{2.25}$$

where $|x|_{1,\sigma} \doteq \sqrt{x^2+\sigma}$ is a smooth approximation of the absolute value. In order to keep dimensional consistency, σ should be a small parameter of order $\mathcal{O}\left(|\mathbf{v}|^2\ell^{2(d-1)}\right)$, where ℓ is a characteristic length of the problem. Let us define the smooth limiter function $f(x) \in \mathcal{C}^2$ that will be used in the definition of α_{ε_h},

$$f(x) \doteq \begin{cases} 2x^4 - 5x^3 + 3x^2 + x & \text{if} \quad x < 1 \\ 1 & \text{if} \quad x \geq 1 \end{cases}.$$

This function is used to smoothly limit the value of x up to 1. Further, let us define another smooth approximation of the absolute value, namely

$$|x|_{2,\varepsilon} \doteq \frac{x^2}{\sqrt{x^2+\varepsilon}}.$$

Finally, the shock detector is defined as

$$\alpha_{\varepsilon_h,i}(u_h) \doteq \left[f\left(\frac{\left| \sum_{j\in\mathcal{N}_h(\Omega_i)} [\![\boldsymbol{\nabla} u_h]\!]_{ij} \right|_{1,\varepsilon_h} + \gamma}{\sum_{j\in\mathcal{N}_h(\Omega_i)} 2\left\{\!\!\left\{ |\boldsymbol{\nabla} u_h \cdot \hat{\boldsymbol{r}}_{ij}|_{2,\varepsilon_h} \right\}\!\!\right\}_{ij} + \gamma} \right) \right]^q, \tag{2.26}$$

where γ is a small parameter that prevents division by zero.

It has been proved in Lemma 2.3.1 that α_i equals 1 when i is an extremum in Ω_i. Let us prove that this is still true for $\alpha_{\varepsilon_h,i}$.

Lemma 2.7.1. *If u_h has an extremum on $i \in \mathcal{N}_h$ then $\alpha_{\varepsilon_h,i}(u_h) = 1$.*

Proof. It is clear that $f(x)$ equals 1 for $x \geq 1$, then the proof reduces to check that

$$\left| \sum_{j\in\mathcal{N}_h(\Omega_i)} [\![\boldsymbol{\nabla} u_h]\!]_{ij} \right|_{1,\varepsilon_h} + \gamma \geq \sum_{j\in\mathcal{N}_h(\Omega_i)} 2\left\{\!\!\left\{ |\boldsymbol{\nabla} u_h \cdot \hat{\boldsymbol{r}}_{ij}|_{2,\varepsilon_h} \right\}\!\!\right\}_{ij} + \gamma.$$

Taking into account that

$$\sqrt{x^2+\varepsilon} = |x|_{1,\varepsilon_h} > |x| \geq |x|_{2,\varepsilon_h} = \frac{x^2}{\sqrt{x^2+\varepsilon}},$$

and the fact that $u_j - u_i$ has the same sign (or it is equal to zero) in all directions, it is easy to see that

$$\begin{aligned}\left|\sum_{j\in\mathcal{N}_h(\Omega_i)}[\![\boldsymbol{\nabla}u_h]\!]_{ij}\right|_{1,\varepsilon_h} &= \left|\sum_{j\in\mathcal{N}_h(\Omega_i)}\frac{u_j-u_i}{|\boldsymbol{r}_{ij}|}+\frac{u_j^{\mathrm{sym}}-u_i}{|\boldsymbol{r}_{ij}^{\mathrm{sym}}|}\right|_{1,\varepsilon_h}\\ &\geq \left|\sum_{j\in\mathcal{N}_h(\Omega_i)}\frac{u_j-u_i}{|\boldsymbol{r}_{ij}|}+\frac{u_j^{\mathrm{sym}}-u_i}{|\boldsymbol{r}_{ij}^{\mathrm{sym}}|}\right|\\ &= \sum_{j\in\mathcal{N}_h(\Omega_i)}\frac{|u_j-u_i|}{|\boldsymbol{r}_{ij}|}+\frac{\left|u_j^{\mathrm{sym}}-u_i\right|}{|\boldsymbol{r}_{ij}^{\mathrm{sym}}|} \geq \sum_{j\in\mathcal{N}_h(\Omega_i)} 2\left\{\!\!\left\{|\boldsymbol{\nabla}u_h\cdot\hat{\boldsymbol{r}}_{ij}|\right\}\!\!\right\}\\ &\geq \sum_{j\in\mathcal{N}_h(\Omega_i)} 2\left\{\!\!\left\{|\boldsymbol{\nabla}u_h\cdot\hat{\boldsymbol{r}}_{ij}|_{2,\varepsilon_h}\right\}\!\!\right\}.\end{aligned}$$

It proves that $\alpha_{\varepsilon_h,i}(u_h) = 1$ on an extremum. In fact, if the solution does not have an extremum, these quantities neither can have the same sign nor be zero in all cases. Since

$$\left|\sum_{j\in\mathcal{N}_h(\Omega_i)}[\![\boldsymbol{\nabla}u_h]\!]_{ij}\right| = \lim_{\varepsilon\to 0}\left|\sum_{j\in\mathcal{N}_h(\Omega_i)}[\![\boldsymbol{\nabla}u_h]\!]_{ij}\right|_{1,\varepsilon_h}$$

and

$$\sum_{j\in\mathcal{N}_h(\Omega_i)} 2\left\{\!\!\left\{|\boldsymbol{\nabla}u_h\cdot\hat{\boldsymbol{r}}_{ij}|\right\}\!\!\right\} = \lim_{\varepsilon\to 0}\sum_{j\in\mathcal{N}_h(\Omega_i)} 2\left\{\!\!\left\{|\boldsymbol{\nabla}u_h\cdot\hat{\boldsymbol{r}}_{ij}|_{2,\varepsilon_h}\right\}\!\!\right\},$$

bound (2.8) leads to the fact that $\lim_{\varepsilon\to 0}\alpha_{\varepsilon_h,i}(u_h) < 1$ when there is no extremum on i. □

It is straightforward to check the following results.

Corollary 2.7.2. *System* (2.11) *with the definition of the shock detector* (2.26) *and artificial diffusion* (2.24) *is LED and satisfies the local DMP. The method tends to a linearly preserving scheme as* $\gamma \to 0$.

Proof. From lemma 2.7.1 and the definition of the regularized maximum (2.25) it is easy to see that artificial diffusion in (2.24) is greater or equal to the one in (2.6). Hence, Theorem 2.4.2 still holds. The linearity preservation is straighforward. □

Remark 2.7.3. *Note that the smoothed shock detector is not linearly preserving because* $\alpha_{\varepsilon_h,i}$ *will never be zero. However, for regions where* u_h *is constant the gradient is zero, thus the solution is not affected. In the case of* $u_h \in P_1(\Omega)$*, but not constant,* $\alpha_{\varepsilon_h,i}$ *goes to zero with* γ*. Values of* γ *of order* 10^{-8} *(or even smaller) have been considered in the numerical experiments section with good nonlinear convergence properties. Thus, the linearity preservation is virtually preserved in practice.*

As in the previous section, when restricted to symmetric meshes, the following approximation (similar to the one in Barrenechea et al. [13]) of (2.26) maintains the same

properties

$$\tilde{\alpha}_{\varepsilon_h,i} \doteq \left[f\left(\frac{\left|\sum_{j\in\mathcal{N}_h(\Omega_i)} u_i - u_j\right|_{1,\varepsilon^*} + \gamma^*}{\sum_{j\in\mathcal{N}_h(\Omega_i)} |u_i - u_j|_{2,\varepsilon^*} + \gamma^*} \right) \right]^q ,$$

with $\varepsilon^* \sim \mathcal{O}(h^2\varepsilon)$ and $\gamma^* \sim \mathcal{O}(h\gamma)$.

2.8 Nonlinear Solvers

In this section the methods used for solving the system of nonlinear equations resulting from the above formulation (2.12) with the artificial diffusion defined in (2.24) is discussed. Taking advantage of the differentiability of the stabilization described in Sect. 2.7, Newton's method is used for the smooth version of the method. In addition, we use fixed point iterations with Anderson acceleration to compare against Newton's method performance. In order to define the schemes, it is useful to write the time-discrete problem (2.12) as

$$\mathsf{A}(u_h^{n+1})u_h^{n+1} = \mathsf{G}$$

where G is the force vector. Let $\mathsf{J}(u_h^{n+1}) \doteq \frac{\partial \mathsf{T}(u_h^{n+1})}{\partial u_h^{n+1}}$ be the Jacobian.

Since the above problem is nonlinear we will solve it iteratively. We denote by $u_h^{k,n+1}$ the k-th iteration of u_h at time step $n+1$. Let us define some auxiliary variables used in the definition of the algorithms: m denotes the number of previous nonlinear iterations used in Anderson acceleration, s is the slope resulting form fitting the last m nonlinear errors, $s_{\min}$ is the minimum slope allowed before increasing the relaxation, ω is the relaxation parameter, $\omega_{\min}$ is its allowed minimum, $k_{\max}$ is the maximum nonlinear iterations allowed, *tol* is the nonlinear tolerance, and *nlerr* is the nonlinear error.

For the non-differentiable methods in Sect. 2.3 we use Picard linearization with Anderson acceleration (see Alg. 1). Our particular implementation also includes a simple convergence rate test, where it is decided if the relaxation parameter should be reduced or not. This improves the global convergence rate and the robustness of the method. Moreover, we add a projection onto V_h^{adm} to ensure that the global DMP in Def. 2.5.1 is satisfied at all nonlinear iterations. This step is of special interest in the case of solving for variables that cannot become negative, e.g., the density. In this case, the projection onto the space of admissible solutions is performed truncating the obtained solution. However, more sophisticated methodologies can be also applied but at a higher computational cost.

For the differentiable method, Newton's linearization is used (see Alg. 2). In addition, we supplement it with the line search method to improve robustness. We use numerical 1D minimization of the residual norm up to a tolerance of 10^{-4} for the line search method. Following the same approach in Alg. 1, a projection to the FE space of admissible solutions can be performed in Alg. 2. As said before, this step ensures that for all nonlinear iterations the solution satisfies the global DMP. The numerical experiments

in the next section show that the modified method keeps quadratic convergence, even though we do not have a theoretical analysis.

Algorithm 1: Fixed point iterations with relaxed Anderson acceleration

Input: $u_h^{0,n+1}$, m, $s_{\min}$, $\omega_{\min}$, tol, A, G, $k_{\max}$
Output: $u_h^{k,n+1}$,k
$k=1$, $nlerr^1 = tol$
while *($nlerr^k \geq tol$) and ($k < k_{\max}$)* **do**
  Set $m^k = \min(k, m)$
  Solve $\mathsf{A}(u_h^{k,n+1})\tilde{u}_h^{k,n+1} = \mathsf{G}$
  Compute $r^{k,n+1} = \tilde{u}_h^{k,n+1} - u_h^{k,n+1}$
  Minimize $\| \sum_{i=1}^{m^k} \xi_i^k r^{k-m^k+i,n+1} \|$ with respect to ξ_i^k subject to $\sum_{i=1}^{m^k} \xi_i^k = 1$
  Set $u_h^{k+1,n+1} = (1-\omega_k) \sum_{i=1}^{m^k} \xi_i^k u_h^{k-m+i,n+1} + \omega_k \sum_{i=1}^{m^k} \xi_i^k \tilde{u}_h^{k-m^k+i,n+1}$
  Project $u_h^{k+1,n+1}$ to V_h^{adm}
  Set $nlerr^k = \frac{\|u_h^{k+1,n+1} - u_h^{k,n+1}\|}{\|u_h^{k+1,n+1}\|}$
  Compute the slope (s) of $\{nlerr^i\}$ with $k \geq i \geq k - m^k$
  if *($s < s_{\min}$) and ($\omega > \omega_{\min}$)* **then**
    Set $\omega_{k+1} = \omega_k - 0.1$
  else
    Set $\omega_{k+1} = \omega_k$
  Update $k = k+1$

Algorithm 2: Newton's method + Line search

Input: $u_h^{0,n+1}$,u_h^n, tol, J, R, $k_{\max}$
Output: $u_h^{k,n+1}$,k
$k=1$, $nlerr^1 = tol$
while *($nlerr^k \geq tol$) and ($k < k_{\max}$)* **do**
  Solve $\mathsf{J}(u_h^{k,n+1})\Delta u_h^{k,n+1} = -\mathsf{T}(u_h^{k,n+1})$
  Minimize $\|\mathsf{T}(u_h^{k,n+1} + \xi^k \Delta u_h^{k,n+1})\|$ with respect to $\xi \in [0,1]$
  Set $u_h^{k+1,n+1} = u_h^{k,n+1} + \xi^k \Delta u_h^{k,n+1}$
  Project $u_h^{k+1,n+1}$ to V_h^{adm}
  Set $nlerr^k = \frac{\|\xi^k \Delta u_h^{k,n+1}\|}{\|u_h^{k+1,n+1}\|}$
  Update $k = k+1$

2.9 Numerical Experiments

2.9.1 Steady problems

First, in order to test the previous formulation, the convergence to a smooth solution is analyzed. For this purpose, the following equation is solved

$$\begin{aligned} \boldsymbol{\nabla} \cdot (\mathbf{v}u) &= 0 \quad \text{in} \quad \Omega = [0,1] \times [0,1], \\ u &= u_D \quad \text{on} \quad \Gamma_{\text{in}}, \end{aligned} \tag{2.27}$$

with $\mathbf{v}(x, y) \doteq (1, 0)$, and inflow boundary conditions $u_D = y - y^2$ on $\partial\Omega\backslash\{x = 1\}$. This problem consists in the transport of the parabolic profile along the x direction, which has the analytical solution $u(x, y) = y - y^2$.

Fig. 2.2 shows the convergence rates using the previously defined formulation ((2.12) with (2.24)), and the Galerkin formulation. To perform this test, an initial mesh of $12 \times 12\, Q_1$ has been considered, then successive refinements have been performed up to a $96 \times 96\, Q_1$ mesh. Analogous meshes has been also used for P_1 FE. Newton's method has been used with $q = 4$, $\varepsilon = 10^{-7}$, $\sigma = |\mathbf{v}|h^4 10^{-8}$ and $\gamma = 10^{-10}$. In this case, σ has been scaled as $|\mathbf{v}|^2 L^{2(d-3)} h^4$ in order to recover optimal convergence, where L denotes a characteristic length of the physical domain Ω. As desired, the convergence rates are not affected by the stabilization, while (as expected) the stabilized solutions have higher errors.

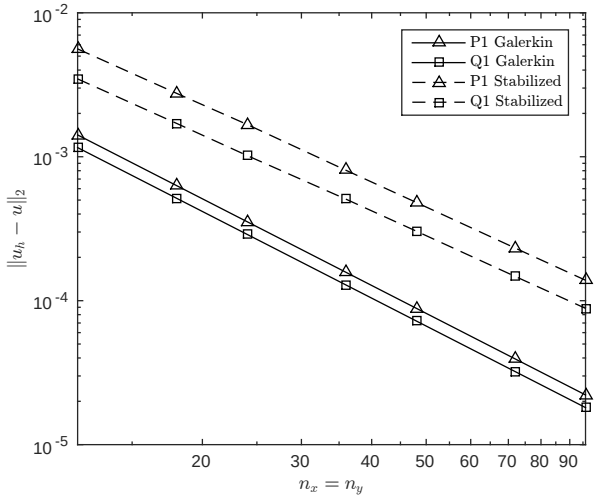


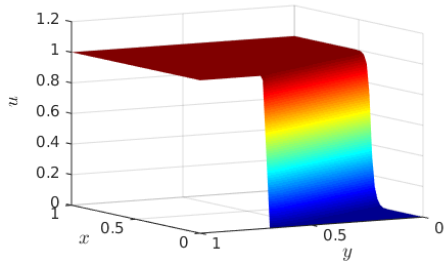
FIGURE 2.2: Convergence test, $L^2(\Omega)$ error versus size of the mesh. For P_1 and Q_1 FE meshes ranging from $h = 1/12$ to $h = 1/96$. Newton's method has been used with parameters $q = 4$, $\varepsilon = 10^{-7}$, $\sigma = |\mathbf{v}|h^4 10^{-8}$ and $\gamma = 10^{-10}$.

A typical linear test to assess the performance of a shock capturing method is the propagation of a discontinuity. Consider now the previous hyperbolic PDE (2.27) with $\mathbf{v}(x, y) \doteq (1/2, \sin -\pi/3)$, and inflow boundary conditions $u_D = 1$ on $\{x = 0\} \cap \{y > 0.7\}$ and $y = 1$, while $u_D = 0$ at the rest of the inflow boundary. This problem has the following analytical solution

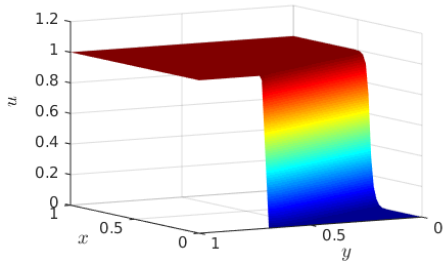
$$u(x, y) = \begin{cases} 1 & \text{if} \quad y > 0.7 + 2x \sin -\pi/3, \\ 0 & \text{otherwise.} \end{cases}$$

At Fig. 2.3(a), the numerical solution using the stabilization in (2.24) is shown. A $48 \times 48\, Q_1$ mesh have been used. The values chosen for the parameters in (2.24) are $q = 25$, $\varepsilon = 10^{-4}$, $\sigma = |\mathbf{v}|10^{-9}$, and $\gamma = 10^{-10}$. This parameter choice makes the solution at the outflow sharp while the DMP is always satisfied. Furthermore, convergence is not

jeopardized thanks to the smoothed stabilization. Particularly, it took 18 iterations for the Newton's method to converge to a nonlinear tolerance of 10^{-6}. The non-smooth version in Fig. 2.3(b) ((2.11) with (2.6)) did not converge using Anderson acceleration, adding a fixed relaxation parameter of $\omega = 0.5$ took 392 iterations, and 117 with Alg. 1. In any case, observing Fig. 2.4, where the outflow profile is depicted, no apparent improvement on accuracy is observed when using the non-smooth version.



(A) Smooth stabilization (2.26), with parameters $q = 25$, $\varepsilon = 10^{-4}$, $\sigma = |\mathbf{v}|10^{-9}$, and $\gamma = 10^{-10}$.



(B) Non-smooth version (2.7) with $q = 25$.

FIGURE 2.3: Stabilized solution of the straight propagation of a discontinuity test using the steady version of discrete problem (2.12) with two stabilization choices (2.26) or (2.7).

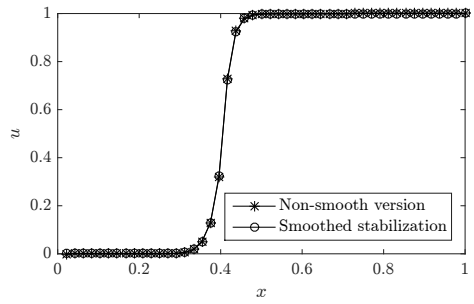


FIGURE 2.4: Stabilized solution of the straight propagation of a discontinuity test using the steady version of discrete problem (2.12) with two stabilization choices (2.26) and (2.7). The stabilization parameters used for the smoothed version are $q = 25$, $\varepsilon = 10^{-4}$, $\sigma = |\mathbf{v}|10^{-9}$, and $\gamma = 10^{-10}$.

Fig. 2.5 shows the solution for several combinations of q and ε, with $\sigma = |\mathbf{v}|\varepsilon 10^{-5}$ and $\gamma = 10^{-10}$, solved with the two nonlinear solvers presented in the previous section over a $48 \times 48\, Q_1$ mesh. Furthermore, the $\|u - u_h\|_{L^1}$ and $\|u - u_h\|$ errors, computed at the whole domain and restricted to the outflow boundary, are listed in Table 2.1. These results show that, as expected, either increasing q or reducing ε the L^1 error diminishes. Nevertheless, the computational cost also increases at a higher rate. The same can be observed for the 2 error. It is slightly reduced after increasing q or diminishing ε, while this makes nonlinear convergence much harder. Moreover, comparing both nonlinear solvers in Sect. 2.8, it is important to note that using Newton's method the number of nonlinear iterations is reduced between 10 to 15 times.

TABLE 2.1: Straight propagation test errors and iterations, using the steady version of discrete problem (2.12) and nonlinear diffusion (2.24), for different values of q and ε, $\sigma = |\mathbf{v}|\varepsilon 10^{-5}$, $\gamma = 10^{-10}$, and both nonlinear solvers in Sect. 2.8.

q	ε	Iterations				L_1 error	L_1 error at Γ_{out}	L_2 error	L_2 error at Γ_{out}
		A	Ap	N	Np				
1	10^{-1}	42	42	9	9	2.77e-02	5.57e-02	8.65e-02	1.23e-01
1	10^{-2}	43	42	8	8	2.61e-02	5.16e-02	8.40e-02	1.18e-01
1	10^{-3}	50	58	7	7	2.59e-02	5.09e-02	8.37e-02	1.17e-01
1	10^{-4}	50	57	7	7	2.58e-02	5.08e-02	8.37e-02	1.17e-01
1	0	56	47			2.59e-02	5.10e-02	8.37e-02	1.17e-01
4	10^{-1}	51	64	8	8	2.20e-02	4.43e-02	7.79e-02	1.12e-01
4	10^{-2}	58	61	11	11	1.83e-02	3.45e-02	6.97e-02	9.70e-02
4	10^{-3}	60	68	10	10	1.77e-02	3.28e-02	6.83e-02	9.44e-02
4	10^{-4}	66	85	11	11	1.76e-02	3.25e-02	6.82e-02	9.40e-02
4	0	70	73			1.76e-02	3.24e-02	6.81e-02	9.39e-02
8	10^{-1}	62	70	9	9	2.10e-02	4.27e-02	7.68e-02	1.11e-01
8	10^{-2}	71	63	11	11	1.62e-02	3.04e-02	6.63e-02	9.23e-02
8	10^{-3}	82	67	13	13	1.51e-02	2.75e-02	6.33e-02	8.74e-02
8	10^{-4}	70	77	12	12	1.49e-02	2.69e-02	6.27e-02	8.66e-02
8	0	94	60			1.48e-02	2.68e-02	6.26e-02	8.64e-02
25	10^{-1}	39	58	11	12	2.03e-02	4.18e-02	7.63e-02	1.11e-01
25	10^{-2}	57	62	19	20	1.46e-02	2.78e-02	6.39e-02	8.95e-02
25	10^{-3}	154	66	15	15	1.28e-02	2.35e-02	5.90e-02	8.24e-02
25	10^{-4}	116	82	17	18	1.25e-02	2.27e-02	5.79e-02	8.18e-02
25	0	86	163			1.23e-02	2.25e-02	5.75e-02	8.15e-02

A: Alg. 1 without projecting to V_h^{adm}, Ap: Alg. 1.
N: Alg. 2 without projecting to V_h^{adm}, Np: Alg. 2.

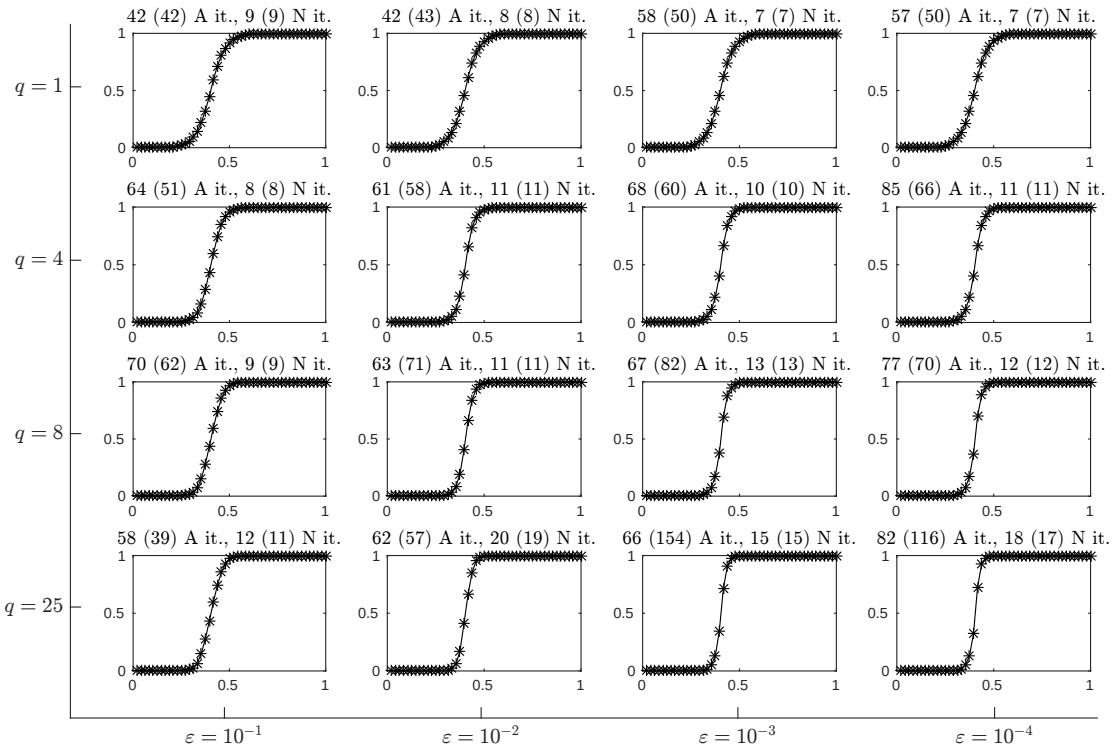
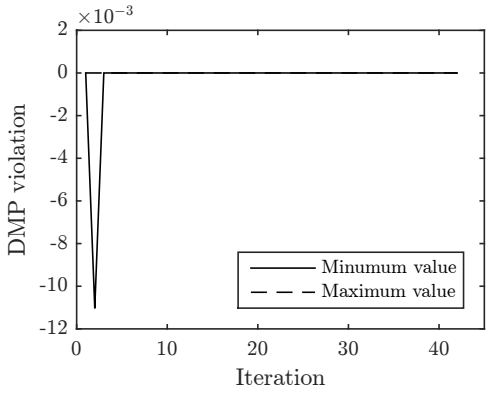
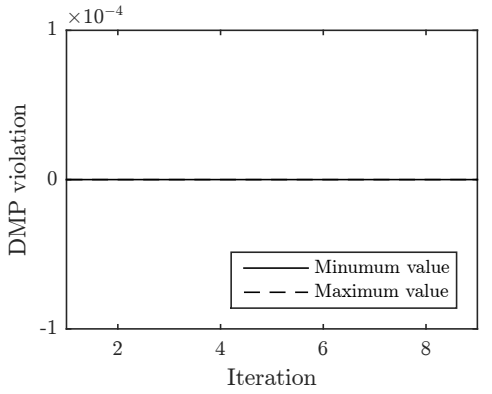


FIGURE 2.5: Straight propagation test solution at the outflow boundary $\partial\Omega\backslash\Gamma_{\text{in}}$. Using the steady version of discrete problem (2.12) and nonlinear diffusion (2.24), for different values of q and ε, $\sigma = |\mathbf{v}|\varepsilon 10^{-5}$, $\gamma = 10^{-10}$, and both nonlinear solvers in Sect. 2.8. The result in brackets shows the number of iterations if no projection to V_h^{adm} is done.

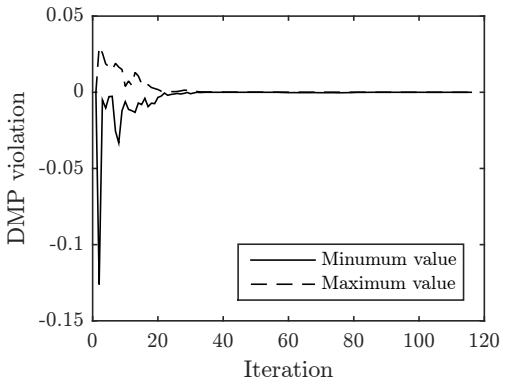
It is important to analyze the solution at each nonlinear iteration. If the projection to the space of admissible solutions is not performed, it is possible that the solution does neither satisfy the local nor the global DMP (Def. 2.2.1 or 2.5.1, resp.) at some nonlinear iterations. The DMP is only proved when convergence is attained. We denote by global DMP violation the difference between the global extremum of the analytical solution and the actual global extremum of the numerical solution. Fig. 2.6 shows the global DMP violation of the maximum and the minimum values produced at each nonlinear iteration for different values of q, ε, and σ. For $q = 25$, the global DMP is clearly not satisfied at the beginning of the iterative process. In this particular case, this does not destroy the nonlinear convergence, but this is not the case in some other problems, e.g. Euler's equations. Therefore, adding a projection step to V_h^{adm} is highly recommended. Further, it can be observed in Table 2.1 that in practice the projection step almost does not affect Newton convergence rate.



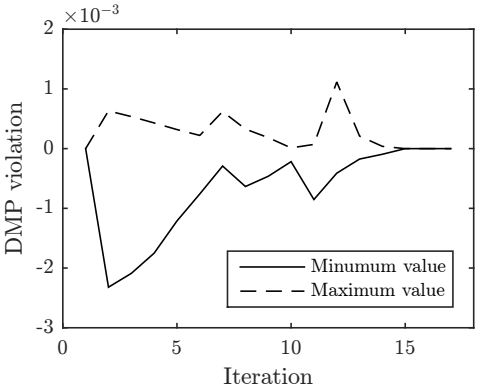
(A) Using Alg. 1 with $q = 1$, $\varepsilon = 10^{-1}$, $\sigma = |\mathbf{v}|10^{-6}$, and $\gamma = 10^{-10}$.



(B) Using Alg. 2 with $q = 1$, $\varepsilon = 10^{-1}$, $\sigma = |\mathbf{v}|10^{-6}$, and $\gamma = 10^{-10}$.



(C) Using Alg. 1 with $q = 25$, $\varepsilon = 10^{-4}$, $\sigma = |\mathbf{v}|10^{-9}$, and $\gamma = 10^{-10}$.



(D) Using Alg. 2 with $q = 25$, $\varepsilon = 10^{-4}$, $\sigma = |\mathbf{v}|10^{-9}$, and $\gamma = 10^{-10}$.

FIGURE 2.6: Evolution of global DMP violation during nonlinear iterations when avoiding the projection step in Algs. 1 and 2 for the straight propagation of a discontinuity test.

Finally, it is worth to test the nonlinear convergence of the method as the mesh is refined for a problem with a discontinuity. For this purpose, we have solved the previous

benchmark with $q = 4$, $\varepsilon = 10^{-2}$, $\sigma = |\mathbf{v}|h^4 10^{-6}$, and $\gamma = 10^{-10}$. The used meshes range form $12 \times 12\, Q_1$ to $96 \times 96\, Q_1$.

At Fig. 2.7, the number of nonlinear iterations for each mesh size is depicted. For Alg. 1 it can be observed that the number of iterations is increasing. On the contrary, this behavior is much less pronounced for Alg. 2; the number of iterations slightly increases and remains constant in the last interval.

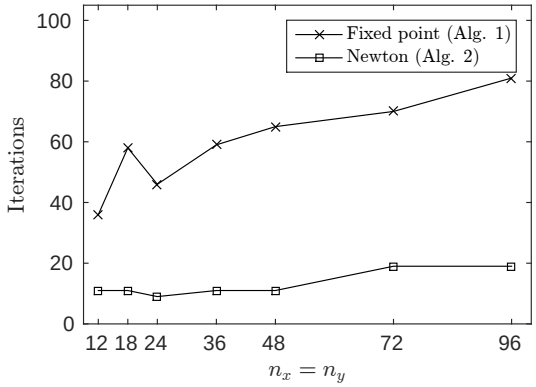


FIGURE 2.7: Straight propagation test nonlinear iterations as mesh refined from $12 \times 12\, Q_1$ to $96 \times 96\, Q_1$, for both Alg. 1 and Alg. 2. The shock capturing parameters used are $q = 4$, $\varepsilon = 10^{-2}$, $\sigma = |\mathbf{v}|h^4 10^{-6}$, and $\gamma = 10^{-10}$.

Consider now the hyperbolic PDE (2.27) on $\Omega = [0,1]\times[-1,1]$ with $\mathbf{v}(x,y) \doteq (y,-x)$, and inflow boundary conditions

$$u_D = \begin{cases} 1 & \text{if} \quad 0.35 < x < 0.65, \\ 0 & \text{otherwise.} \end{cases}$$

This particular configuration has the following analytical solution

$$u(x,y) = \begin{cases} 1 & \text{if} \quad 0.35 < \sqrt{x^2+y^2} < 0.65, \\ 0 & \text{otherwise.} \end{cases}$$

At Fig. 2.8 the solutions at the outflow boundary are depicted for several combinations of q and ε, with $\sigma = |\mathbf{v}|\varepsilon 10^{-5}$ and $\gamma = 10^{-10}$. In all cases, we have considered the two schemes presented in Sect. 2.8 using a $64 \times 128\, Q_1$ FE mesh. As for the previous numerical experiment, we collect the number of iterations and the errors in Table 2.2. We observe that it is particularly difficult to converge to the solution for $q = 1$ and small values of ε. In any case, for q equal to 4 or greater, the number of iterations increase with q, as naturally expected. We also observe in this test that the number of nonlinear iterations can be highly reduced using Newton's method. Particularly, it reduces the number of nonlinear iterations up to 20 times. 3D plots of the smoothest and the sharpest solutions in Fig. 2.8 (respectively top-left and bottom-right subfigures) are shown in Fig. 2.9 .

TABLE 2.2: Circular propagation test errors and iterations, using the steady version of discrete problem (2.12) and nonlinear diffusion (2.24), for different values of q and ε, $\sigma = |\mathbf{v}|\varepsilon 10^{-5}$, $\gamma = 10^{-10}$, and both nonlinear solvers in Sect. 2.8.

q	ε	Iterations A	Ap	N	Np	L_1 error	L_1 error at Γ_{out}	L_2 error	L_2 error at Γ_{out}
1	10^{-1}	30	30	9	9	1.42e-01	1.93e-01	2.01e-01	2.36e-01
1	10^{-2}	–	54	10	10	1.11e-01	1.50e-01	1.74e-01	2.05e-01
1	10^{-3}	–	–	11	11	1.05e-01	1.42e-01	1.68e-01	1.99e-01
1	10^{-4}	196	–	19	19	1.04e-01	1.40e-01	1.68e-01	1.98e-01
1	0	–	–			–	–	–	–
4	10^{-1}	23	23	10	10	1.33e-01	1.82e-01	1.97e-01	2.31e-01
4	10^{-2}	64	64	15	15	8.47e-02	1.15e-01	1.55e-01	1.84e-01
4	10^{-3}	105	111	22	22	6.74e-02	9.31e-02	1.34e-01	1.64e-01
4	10^{-4}	–	139	24	24	6.38e-02	8.88e-02	1.29e-01	1.60e-01
4	0	198	194			6.31e-02	8.80e-02	1.28e-01	1.59e-01
8	10^{-1}	23	22	11	11	1.32e-01	1.81e-01	1.97e-01	2.31e-01
8	10^{-2}	73	68	15	15	8.10e-02	1.10e-01	1.53e-01	1.82e-01
8	10^{-3}	95	96	19	19	5.91e-02	8.18e-02	1.28e-01	1.57e-01
8	10^{-4}	100	109	22	22	5.28e-02	7.46e-02	1.18e-01	1.50e-01
8	0	256	231			5.12e-02	7.28e-02	1.16e-01	1.48e-01
25	10^{-1}	22	22	14	14	1.32e-01	1.80e-01	1.97e-01	2.31e-01
25	10^{-2}	45	49	16	15	7.82e-02	1.07e-01	1.51e-01	1.80e-01
25	10^{-3}	77	70	20	20	5.37e-02	7.50e-02	1.24e-01	1.54e-01
25	10^{-4}	131	109	23	24	4.51e-02	6.49e-02	1.11e-01	1.44e-01
25	0	180	289			4.22e-02	6.14e-02	1.06e-01	1.39e-01

A: Alg. 1 without projecting to V_h^{adm}, Ap: Alg. 1.
N: Alg. 2 without projecting to V_h^{adm}, Np: Alg. 2.

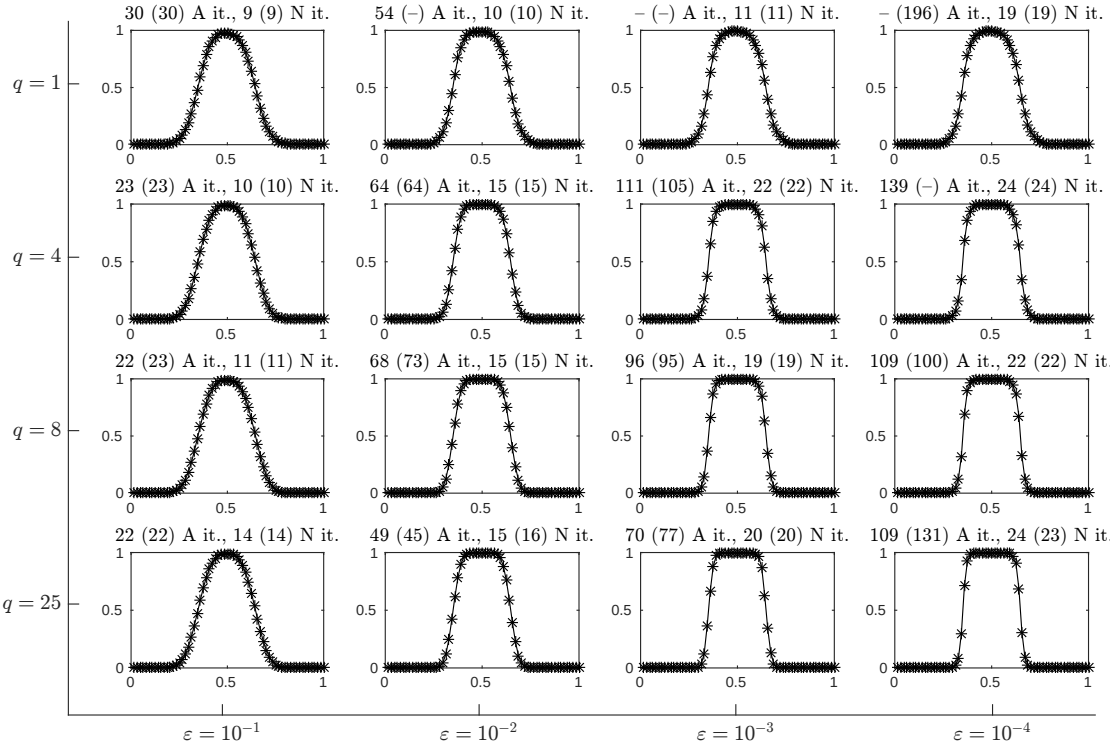
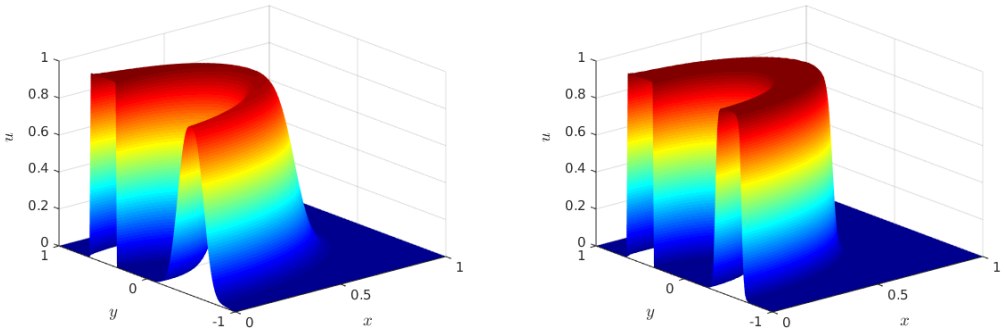


FIGURE 2.8: Circular propagation test solution at the outflow boundary $\partial\Omega\backslash\Gamma_{\text{in}}$. Using the steady version of discrete problem (2.12) and nonlinear diffusion (2.24), for different values of q and ε, $\sigma = |\mathbf{v}|\varepsilon 10^{-5}$, $\gamma = 10^{-10}$ and both nonlinear solvers in Sect. 2.8. The result in brackets shows the number of iterations if no projection to V_h^{adm} is done.

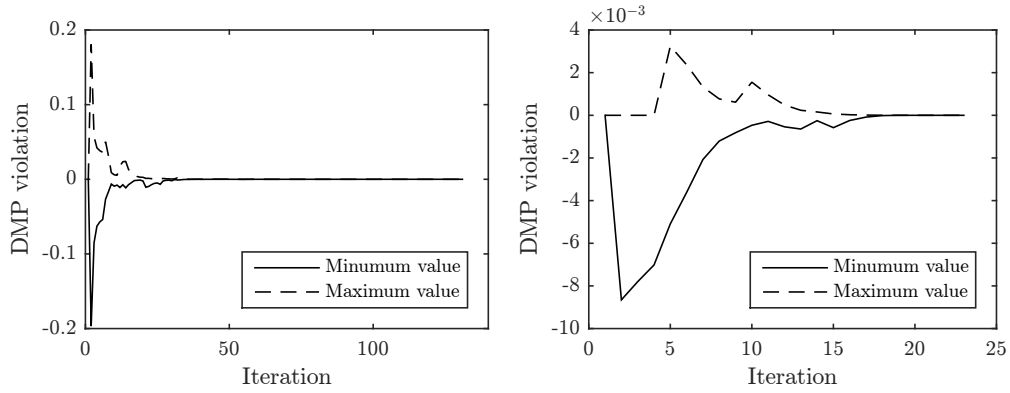


(A) Smoothest solution with parameters: $q = 1$, $\varepsilon = 10^{-1}$, $\sigma = |\mathbf{v}|10^{-6}$, and $\gamma = 10^{-10}$.

(B) Sharpest solution with parameters: $q = 25$ and $\varepsilon = 10^{-4}$, $\sigma = |\mathbf{v}|10^{-9}$, and $\gamma = 10^{-10}$.

FIGURE 2.9: Stabilized solution of the circular convection test using the steady version of the discrete problem (2.12) and the nonlinear diffusion (2.24) for two different parameter choices.

Fig. 2.10 shows that in this second test, as in the previous one, if the projection step is not performed the global DMP (Def. 2.5.1) is not satisfied at all nonlinear iterations. This is especially evident for the combination shown in the figure, i.e., high values of q and low values of ε and σ.



(A) Using Alg. 1

(B) Using Alg. 2

FIGURE 2.10: Evolution of global DMP violation during nonlinear iterations when avoiding the projection step in Algs. 1 and 2 for the circular propagation of a discontinuity. Using $q = 25$, $\varepsilon = 10^{-4}$, $\sigma = |\mathbf{v}|10^{-9}$, $\gamma = 10^{-10}$.

2.9.2 Transient transport problems

Let us test the performance of the stabilization method in Sect. 2.7 for transient problems. For this purpose we will consider the 3 body rotation benchmark that reads as:

$$\begin{aligned} \partial_t u + \boldsymbol{\nabla}\cdot(\mathbf{v}u) &= 0 \quad \text{in} \quad \Omega = [0,1]\times[0,1], \\ u &= 0 \quad \text{on} \quad \Gamma_{\text{in}}, \\ u &= u_0 \quad \text{at} \quad t = 0, \end{aligned} \tag{2.28}$$

where $\mathbf{v} \doteq (1/2 - y, x - 1/2)$ and

$$u_0(x,y) \doteq \begin{cases} \frac{1}{4} + \cos\left(\frac{\pi\sqrt{(x-0.25)^2+(y-0.5)^2}}{0.15}\right)/4 & \text{if} \quad \sqrt{(x-0.25)^2+(y-0.5)^2}/0.15 \leq 1 \\ 1 - \sqrt{(x-0.5)^2+(y-0.25)^2}/0.15 & \text{if} \quad \sqrt{(x-0.5)^2+(y-0.25)^2}/0.15 \leq 1 \\ 1 & \text{if} \quad \begin{cases} \sqrt{(x-0.5)^2+(y-0.75)^2}/0.15 \leq 1 \\ 0.55 < x < 0.45,\ y > 0.85 \end{cases} \end{cases}.$$

The above problem is solved in a 150×150 Q_1 FE mesh, with solver parameters $q = 25$, $\gamma = 10^{-8}$, $\sigma = |\mathbf{v}|10^{-10}$, and $\varepsilon = 10^{-4}$. The discretization in time is performed using the BE method with a time step of 10^{-3}. At Fig. 2.12(a), the initial solution is depicted. Figs. 2.12(b) to 2.12(d) show the solution after one revolution (at time $t = 2\pi$).

The solution obtained with the stabilization in (2.24), (2.15), and (2.6) are depicted in Figs. 2.12(b), 2.12(c), and 2.12(d), respectively. It is observed that the symmetric mass matrix method yields slightly more diffusive solutions than the LED method. This can be better observed in Fig. 2.11, where a cross-section of each of the figures rotated is depicted at $t = 0$ and after one revolution ($t = 2\pi$) for all three methods. As naturally expected, regularizing the stabilization makes the method faster to converge but the solution becomes smoother. Nevertheless, the regularization parameters (σ and ε) allow one to take the choice that better fits the requirements, either a faster but smoother method or the opposite. In any case, all schemes satisfy the DMP at all time steps.

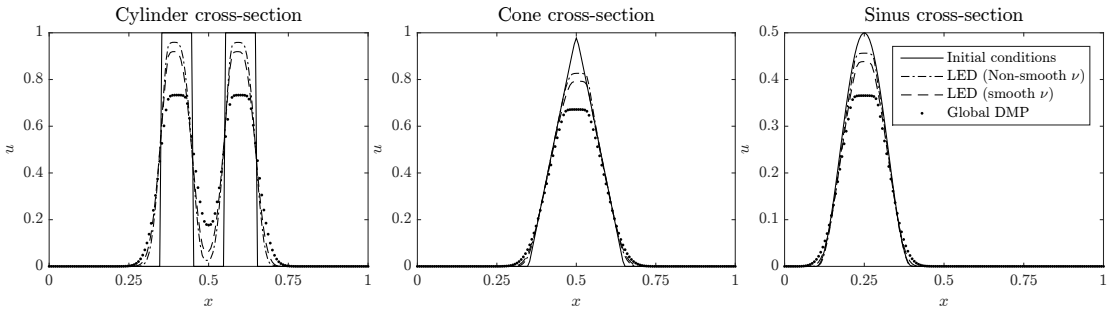
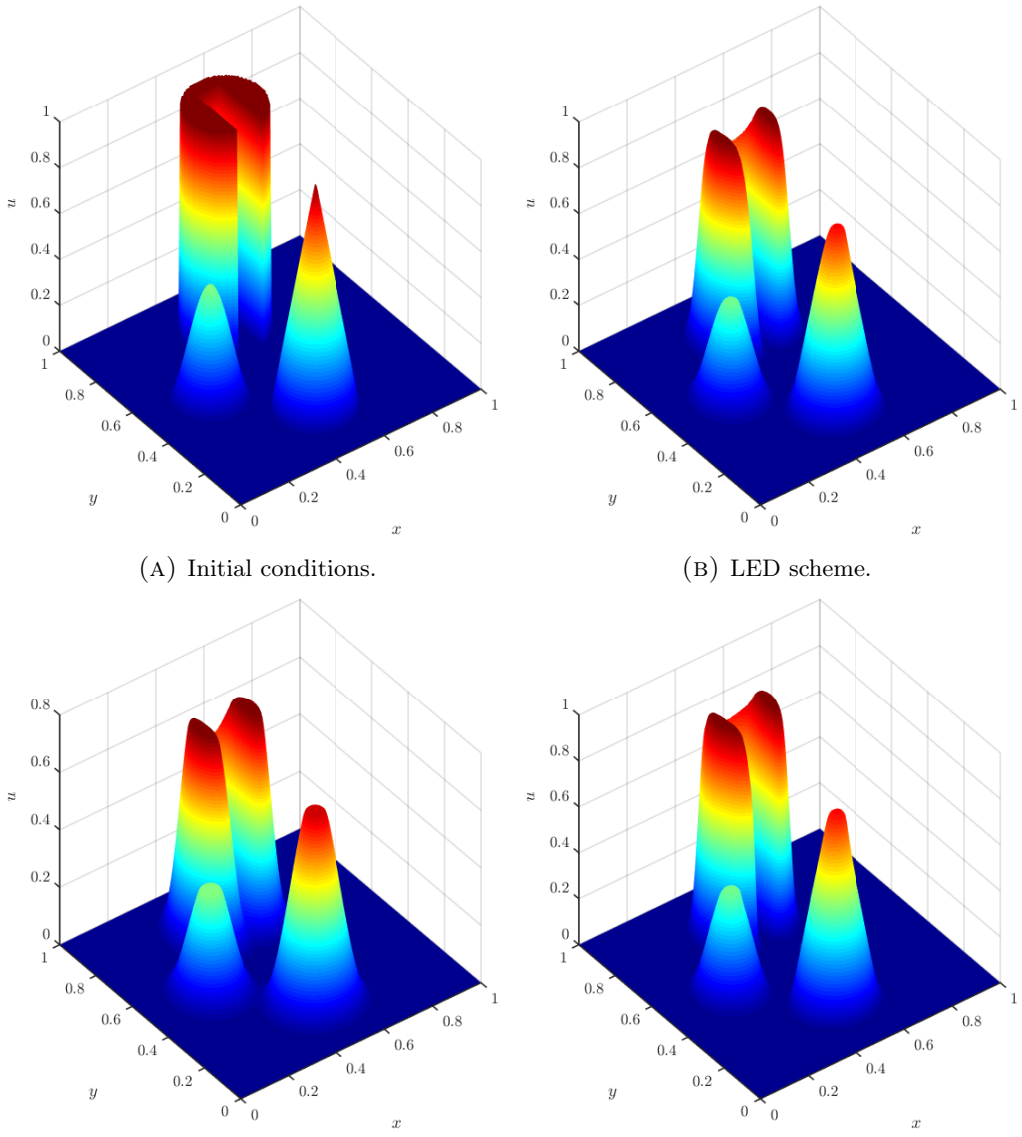


FIGURE 2.11: Cross-sections of each for the figures rotated in the three body rotation benchmark. The parameters used are $q = 25$, $\gamma = 10^{-8}$, $\sigma = |\mathbf{v}|10^{-10}$, $\varepsilon = 10^{-4}$, and $\Delta t = 10^{-3}$, in a 150×150 Q_1 element mesh. The discrete problem (2.12) is used in combination with three different artificial diffusions (2.24) and (2.6) leading to a LED scheme, and (2.15) leading to a global DMP scheme.



(A) Initial conditions.

(B) LED scheme.

(C) Global DMP scheme.

(D) LED DMP nonsmooth stabilization.

FIGURE 2.12: 3 Body rotation test results using discrete problem (2.12) and two different artificial diffusions ((2.24) leading an LED scheme, and (2.15) with (2.26) leading a global DMP scheme). Using a 150×150 Q_1 element mesh, and parameters: $q = 25$, $\gamma = 10^{-8}$, $\sigma = |\mathbf{v}|10^{-10}$, $\varepsilon = 10^{-4}$, and $\Delta t = 10^{-3}$.

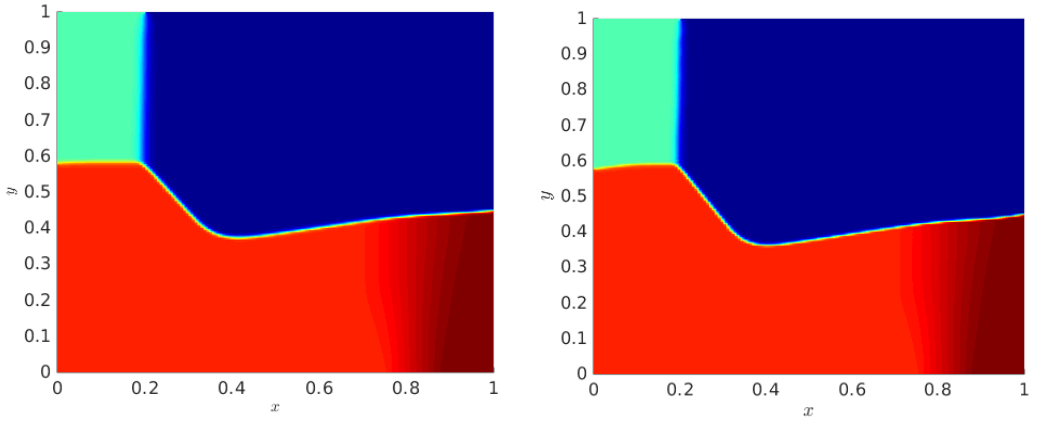
2.9.3 Burgers' equation

Finally, let us test our stabilization with a nonlinear transient problem. Particularly the 2D Burgers' equation, i.e. equation (2.28) with $\mathbf{v} \doteq (1,1)^u/_2$, is solved on $\Omega = [0,1]\times[0,1]$ using a $150 \times 150\, Q_1$ mesh. The discretization in time is performed using the BE method

with a time step of 10^{-2}. The initial conditions at $t = 0$ are

$$u_0(x,y) \doteq \begin{cases} -0.2 & \text{if} \quad x < 0.5 \quad \text{and} \quad y > 0.5 \\ -1 & \text{if} \quad x > 0.5 \quad \text{and} \quad y > 0.5 \\ 0.5 & \text{if} \quad x < 0.5 \quad \text{and} \quad y < 0.5 \\ 0.8 & \text{if} \quad x > 0.5 \quad \text{and} \quad y < 0.5 \end{cases},$$

and the solution is advanced until $t = 0.5$.



(A) Solution for:$q = 1$, $\varepsilon = 10^{-2}$, $\sigma = |\mathbf{v}|10^{-6}$, and $\gamma = 10^{-8}$.

(B) Solution for: $q = 4$, $\varepsilon = 10^{-3}$, $\sigma = |\mathbf{v}|10^{-7}$, and $\gamma = 10^{-8}$.

FIGURE 2.13: Burger's equation solutions at $t = 0.5$ using discrete problem (2.12) and (2.6) with (2.24). Using a 150×150 Q_1 element mesh, $\Delta t = 10^{-2}$, and two sets of parameters q, γ, σ, and ε.

The following stabilization parameters have been used for obtaining the results in Fig. 2.13(a): $q = 1$, $\varepsilon = 10^{-3}$, $\sigma = |\mathbf{v}|10^{-6}$, and $\gamma = 10^{-8}$. Although the parameters used are not enforcing a particularly sharp solution (see Figs. 2.5 and 2.8), Fig. 2.13(a) shows properly transported and minimally smeared shocks. Only in the lower right region the method appears to be more diffusive than desired. Notice that in that region the gradient in the x direction spreads as y increases, while it should not. Nevertheless, in Fig. 2.13(b), that shows the solution for $q = 4$, $\varepsilon = 10^{-4}$, $\sigma = |\mathbf{v}|10^{-7}$, and $\gamma = 10^{-8}$. the method is less diffusive and the obtained shocks are even sharper. In any case, both choices satisfy the DMP for all time steps.

2.10 Conclusions

In this chapter, we have considered a nonlinear stabilization technique for the FE approximation of scalar conservation laws with implicit time stepping. The method relies on an artificial diffusion method, based on a graph-Laplacian operator. The artificial diffusion is judiciously chosen in order to satisfy a local DMP for steady problems. It is nonlinear, since it depends on a shock detector. Further, the resulting method is linearity preserving. The same shock detector is used to gradually lump the mass matrix.

The resulting method is LED, positivity preserving, and also satisfies a global DMP. Lipschitz continuity has also been proved.

However, the resulting scheme is highly nonlinear, leading to very poor nonlinear convergence rates, even using Anderson acceleration techniques. It is due to the fact that the nonlinear operator to be inverted at every time step is non-differentiable. The critical problem of nonlinear convergence of implicit monotonic methods based on nonlinear artificial diffusion have already been previously reported in the literature (see [57]). As a result, we propose a smooth version of the scheme. It leads to twice differentiable nonlinear stabilization schemes, which allows one to straightforwardly use Newton's method using the exact Jacobian. Twice differentiability ensures quadratic convergence.

We have considered two nonlinear solvers, namely Anderson acceleration and Newton's method. We have observed numerically that the effect of the smoothness has a positive impact in the reduction of the computational cost. The impact of using Newton's method versus Anderson acceleration is also very positive. In general, using the Newton method with a smooth version of the method we can reduce 10 to 20 times the number of iterations of Anderson acceleration with the original non-smooth algorithms.

All the monotonic properties are satisfied (as theoretically proved) in the numerical experiments. Steady and transient linear transport, and transient Burgers' equation have been considered in 2D. In any case, these properties are only true for the converged solution, but not for iterates. In this sense, we have also proposed the concept of projected nonlinear solvers, where a projection step is performed at the end of every nonlinear iterations onto a FE space of admissible solutions. The space of admissible solutions is the one that satisfies the desired monotonic properties (maximum principle or positivity). The projection has no effect on the quality of the nonlinear convergence.

Future work should tackle the entropy stability analysis of the resulting schemes when applied to nonlinear problems. Some initial results in this direction can be found in [23]. The extension to systems of conservation laws and higher order methods in space and time is another interesting line of research.

Chapter 3

Arbitrary order space–time monotonicity preserving scheme

This chapter is devoted to a nonlinear stabilization technique for convection–diffusion–reaction and pure transport problems discretized with space–time isogeometric analysis. The stabilization is based on a graph-theoretic artificial diffusion operator and a novel shock detector for isogeometric analysis. Stabilization in time and space directions are performed similarly, which allow us to use high-order discretizations in time without any CFL-like condition. The method is proved to yield solutions that satisfy the discrete maximum principle (DMP) unconditionally for arbitrary order. In addition, the stabilization is linearity preserving in a space–time sense. Moreover, the scheme is proved to be Lipschitz continuous ensuring that the nonlinear problem is well-posed. Solving large problems using a space–time discretization can become highly costly. Therefore, we also propose a partitioned space–time scheme that allow us to select the length of every time slab, and solve sequentially for every subdomain. As a result, the computational cost is reduced while the stability and convergence properties of the scheme remain unaltered. In addition, we propose a twice differentiable version of the stabilization scheme, which enjoys the same stability properties while the nonlinear convergence is significantly improved. Finally, the proposed schemes are assessed with numerical experiments. In particular, we considered steady and transient pure convection and convection–diffusion problems in one and two dimensions.

3.1 Introduction

Many different applications in science and industry require solving problems satisfying some sort of positivity or maximum principle (MP) property. These include scalar transport problems, compressible flows, or fluid-based MHD simulations, among others. These problems are of particular interest in a variety of industries and scientific research areas, such as the chemical industry, aviation, aerospace, or nuclear fusion research, just to cite few examples.

Some of these problems exhibit a multiscale nature in time. In those cases, explicit methods are not suitable, since the smallest time scales pose very stringent stability conditions to the time step length, i.e., fully resolved time simulations are required.

Thus, implicit methods are favored in applications where the smallest time scales are not of scientific or engineering interest.

As a result, schemes that preserve monotonicity (or at least positivity) for implicit time integration are of special interest. The standard technique to attain such schemes is adding nonlinear artificial diffusion (usually called shock capturing). The common ingredients of a shock capturing or nonlinear stabilization method are the following. The first ingredient is the *artificial diffusion*, which needs to be sufficient to eliminate non-physical oscillations. The schemes in [7, 8, 23, 25] use an element-based artificial diffusion with a standard PDE-based diffusion operator. The drawback of this choice is the fact that the DMP only holds under unpractical mesh restrictions. This problem has been solved by Guermond and Nazarov in [38, 41] by replacing the PDE-based diffusion operator by an edge or graph-theoretic diffusion operator; see [4, 5, 65, 71, 75] for schemes that preserve the DMP on arbitrary meshes using a graph-Laplacian. The second ingredient is a *shock detector*, which is the term responsible of deactivating the artificial diffusion in smooth regions. A good shock detector is of vital importance for minimizing the numerical diffusion while satisfying a DMP. One example of shock detector is the one developed in 1D by Burman in [23] and later extended to multiple dimensions by Badia and Hierro [7]. The last ingredient consists on perturbing the mass matrix. One option is a full lumping of the mass matrix, but it can lead to unacceptable phase errors. Instead, a nonlinear lumping is used, e.g., in [4, 5], using the same shock detector to lump the mass matrix. Other alternatives can be found in [40, 65]. It is worth mentioning that all previous stabilization methods yield a very stiff nonlinear system of equations. In fact, some of the methods proposed in the literature are not even Lipschitz continuous and thus ill-posed (see [13]). In practice, the nonlinear convergence of these methods is unacceptably slow, making hard its practical use. To solve this problem, Badia et al. [4, 5] have designed differentiable nonlinear stabilization terms, noticeably improving the nonlinear convergence.

The methods commented above have an algebraic nature and provide some type of DMP for the nodal values. The monotonicity of the nodal values only translates into monotonic solutions if the FE space satisfies the convex hull property, which is only true in the first order case. As a result, using the ideas above it does not seem possible to design monotonic second or higher order methods. Recently, Kuzmin and coworkers [2, 71], have proposed instead the usage of Bernstein–Bèzier FEs, since they satisfy the convex hull for high-order. However, the temporal dimension is discretized using Backward Euler or SSP RK methods (see [53]). In the first case, the problem is first order in time, whereas in the second case, a CFL-like condition arises [67], since high-order SSP methods pose a restriction on the time step size similar to the ones in explicit methods [53].

The main contribution present in this chapter is the development of a high-order (both in space and time) and DMP-preserving discretization for the convection–diffusion–reaction and pure transport problems. This is achieved by combining the nonlinear

stabilization techniques in the previous chapter and [5] with a *new shock detector* for arbitrary order space–time isogeometric analysis. Another novelty introduced in this chapter is the stabilization in the time direction, which is performed in a similar manner as in space. This results in an unconditionally stable high-order method in time (and space). However, the space–time method requires to solve the whole space–time problem at once, which increases the computational cost. Hence, we also propose a partitioned approach in the temporal direction, where one can determine the width of the time slab to be computed every time. This strategy allows us to maintain a *reasonable computational cost while having a high-order scheme in space and time, as well as satisfying the DMP without any CFL-like condition.* Finally, we also propose a *differentiable version* of the above scheme. This allow us to use Newton's method, which improves nonlinear convergence significantly.

This chapter is structured as follows. First, we introduce the problem, its discretization, and monotonicity properties for scalar problems in Sect. 3.2. Then, the stabilization techniques are introduced in Sect. 3.3. Sect. 3.4 is devoted to the partitioned time integration scheme. Afterwards, we introduce a regularized version of the stabilization term in Sect. 3.5. Finally, we show numerical experiments in Sect. 3.6 and draw some concluding remarks in Sect. 3.7.

3.2 Preliminaries

3.2.1 Convection–Diffusion problem

We consider a transient convection–diffusion problem with Dirichlet boundary conditions. Let $\Omega \times (0,T) \doteq \prod_{\alpha=1}^{d+1}(0,L_\alpha)$ be a $(d+1)$-cube, where d is the number of spatial dimensions. Then, the problem reads:

$$\begin{cases} \partial_t u + \boldsymbol{\nabla}\cdot(\mathbf{v}u) - \boldsymbol{\nabla}\cdot(\mu\boldsymbol{\nabla}u) = g & \text{in } \Omega\times(0,T], \\ u(x,t) = \overline{u}(x,t) & \text{on } \partial\Omega\times(0,T], \\ u(x,0) = u_0(x) & x\in\Omega, \end{cases} \tag{3.1}$$

where $\mathbf{v}$ is a divergence-free convection velocity, $\mu \geq 0$ is a scalar constant diffusion, and $g(\boldsymbol{x},t)$ is the body force. In the case of pure convection ($\mu = 0$), boundary conditions are only imposed at the inflow $\Gamma_{\text{in}} \doteq \{\boldsymbol{x} \in \partial\Omega : \mathbf{v}\cdot\boldsymbol{n}_{\partial\Omega} < 0\}$, where $\boldsymbol{n}_{\partial\Omega}$ is a unit vector outward-pointing normal to the boundary. We also define the outflow boundary as $\Gamma_{\text{out}} \doteq \partial\Omega\backslash\Gamma_{\text{in}}$. Moreover, we will also consider the steady problem, which is obtained by dropping the time derivative term and the initial condition. It is important to mention that a reaction term can be included without harming any of the properties satisfied by the schemes introduced below. However, a convection–diffusion–reaction problem only satisfies a MP if the minimum is negative and the maximum positive (analogously for its proposed discretizations). In other words, it only satisfy a weak MP, see [13]. In order

to simplify the discussion below, we will limit the present chapter to pure convection and convection–diffusion problems.

In order to avoid technicalities and facilitate the exposition of the stabilization method, we restrict this chapter to cubic domains. However, it is possible to work with complex geometries using standard procedures from isogeometric analysis [29]. E.g., a complex geometry would be divided in several parts, which would be mapped to multiple d-cubic patches. The stabilization method presented in this chapter is independent from this procedure.

3.2.2 Discretization

In this chapter, we consider a standard B-spline discretization with interpolative boundaries (see [29]). A spline of order p in the variable x is a piecewise polynomial function in x of degree p. The values of x in which different polynomials meet are call *knots*. Knots might be placed at the same location, i.e. can be repeated. When the knots are not repeated, the first $p-1$ derivatives of the spline are continuous. When a knot is repeated r times, only the first $p-r$ derivatives are continuous across that knot. Knots are sorted in increasing order and collected in the so called *knot vector* $\{\xi_1, \xi_2, \ldots\}$. Given a knot vector, B-splines of order p are basis functions for spline functions of the same order. B-splines are constructed in a recursive way using the Cox-de Boor formula:

$$B_i^0(x) \doteq \begin{cases} 1 & \text{if } \xi_i \leq \xi < \xi_{i+1} \\ 0 & \text{otherwise} \end{cases},$$

$$B_i^k(x) \doteq \frac{x - \xi_i}{\xi_{i+k} - \xi_i} B_i^{k-1}(x) + \frac{\xi_{i+k+1} - x}{\xi_{i+k+1} - \xi_{i+1}} B_{i+1}^{k-1}(x),$$

for $k = 1, ..., p$. By construction, $B_i^p(x)$ has compact support, is non-negative, and non-zero in $[\xi_i, \xi_{i+p+1}]$. Notice that its support increases with the degree of the polynomial.

Let us consider the domain $[0, L]$ and the uniform partition into m sub-intervals of size $h = L/m$. The *open* knot vector $\{\xi_1, \ldots, \xi_{m+2p+1}\}$ is defined as follows. The first $p+1$ knots are located at zero, i.e., $\xi_1 = \ldots = \xi_{p+1} = 0$. The last $p+1$ knots are located at L, i.e., $\xi_{m+p+1} = \ldots = \xi_{m+2p+1} = L$. The interior points are equi-distributed, with $\xi_i = (i - p - 1)h$, for $i = p+1, \ldots, m+p+1$. It leads to a basis $B_i^p(x)$ (for $i = 1, \ldots, m+p$) for a space of splines in $[0, L]$ and a partition of unity, i.e., $\sum_{i=1}^m B_i^p(x) = 1$ for $x \in [0, L]$. Any spline $v(x)$ of order p in $[0, L]$ can uniquely be defined by the *control points* $(v_1, \ldots, v_{m+p}) \in \mathbb{R}^{m+d}$ as the linear combination of B-splines $v(x) = \sum_{i=1}^{m+p} B_i^p(x) v_i$. In one dimension, the basis functions obtained from an open knot vector are interpolatory at the extremes, i.e., $v(0) = v_1$ and $v(L) = v_{m+p+1}$ (see Fig. 3.1). For a first order polynomial v in $[0, L]$, it holds $v(x) = \sum_{i=1}^m B_i^p(x) v(x_i)$, where $x_i \doteq (\xi_{i+1} + \ldots + \xi_{i+p})/p$ are called the *Greville abscissae* [30, 76].

Let us consider the number of partitions per dimension with m_α, for $\alpha = 1, \ldots, d+1$. We represent with $\mathcal{N}_h$ the set of multi-indices $\boldsymbol{i} \doteq (i_1, ..., i_{d+1}) \in \mathbb{Z}^{d+1}$ with $i_\alpha \in$

$\{1, \dots, m_\alpha + p\}$. Every $\boldsymbol{i} \in \mathcal{N}_h$ can be expressed as $(\boldsymbol{i}_{\boldsymbol{x}}, i_t)$, where $\boldsymbol{i}_{\boldsymbol{x}}$ is the spatial index and i_t is the temporal index. The $(d+1)$-dimensional B-spline is defined as the tensor product of $d+1$ unidimensional B-splines $B^p_{\boldsymbol{i}}(\boldsymbol{x}) \doteq B^p_{i_1}(x_1) \times \cdots \times B^p_{i_{d+1}}(x_{d+1})$. Notice that a Greville abscissa in the case of a multidimensional spline reads $\boldsymbol{x}_{\boldsymbol{i}} = (x_{\boldsymbol{i}_1}, ..., x_{\boldsymbol{i}_{d+1}})$.

We define the space of splines $V_h \doteq \text{span}\{B^p_{\boldsymbol{i}}(\boldsymbol{x}) \ : \ \boldsymbol{i} \in \mathcal{N}_h\}$. We use the notation $\varphi_{\boldsymbol{i}} \equiv B^p_{\boldsymbol{i}}$. The order is omitted since it is assumed to be fixed. Thus, every spline $v_h \in V_h$ can be written as $v_h = \sum_{\boldsymbol{i} \in \mathcal{N}_h} \varphi_{\boldsymbol{i}} v_{\boldsymbol{i}}$. Furthermore, we define the following sets of indices, which are useful for the definition of the forthcoming schemes. The set of neighbors of $\boldsymbol{i}$ is defined as $\mathcal{N}^{\boldsymbol{i}}_h \doteq \{\boldsymbol{j} \in \mathcal{N}_h : |\boldsymbol{i} - \boldsymbol{j}|_\infty \leq 1\}$. We define as $\mathcal{S}^{\boldsymbol{i}}_h \doteq \{\boldsymbol{j} \in \mathcal{N}_h : |\boldsymbol{i} - \boldsymbol{j}|_\infty \leq p\}$ the set of indices whose associated shape functions intersect with the support of $\varphi_{\boldsymbol{i}}$.

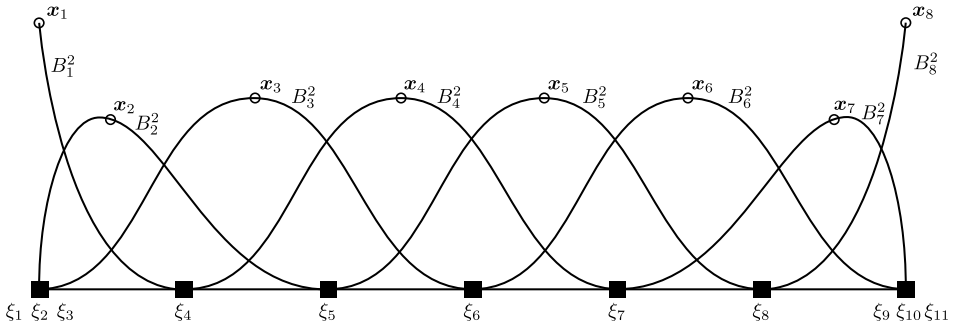


FIGURE 3.1: Representation of the basis functions of V^2_h in one dimension, with its associated Greville abscissae.

We use standard notation for Sobolev spaces. The $L^2(\omega)$ scalar product is denoted by $(\cdot, \cdot)_\omega$ for $\omega \subset \Omega$. However, we omit the subscript for $\omega \equiv \Omega$. The $L^2(\Omega)$ norm is denoted by $\|\cdot\|$.

3.2.3 Discrete problem

The weak form of (3.1) using the Galerkin method reads: find $u_h \in V_h$ such that $u_h(\boldsymbol{x}, t) = \overline{u}_h(\boldsymbol{x}, t)$ on $\partial\Omega \times (0, T]$, $u_h(\boldsymbol{x}, 0) = u_{0h}(\boldsymbol{x})$ on $\Omega \times \{0\}$, and

$$(\partial_t u_h, v_h) + (\mathbf{v} \cdot \boldsymbol{\nabla} u_h, v_h) + \mu(\boldsymbol{\nabla} u_h, \boldsymbol{\nabla} v_h) = (g, v_h), \qquad \forall v_h \in V_h, \tag{3.2}$$

where $\overline{u}_h(t)$ and u_{0h} are projections of $\overline{u}(t)$ and u_0 to V_h, respectively, such that the local DMP is satisfied (see Def. 3.2.2). Furthermore, we can rewrite the previous discrete problem in matrix form as $\mathbf{K}_{\boldsymbol{ij}} u_{\boldsymbol{j}} = \mathbf{F}_{\boldsymbol{i}}$, where $\mathbf{K}_{\boldsymbol{ij}} \doteq (\partial_t \varphi_{\boldsymbol{j}}, \varphi_{\boldsymbol{i}}) + (\mathbf{v} \cdot \boldsymbol{\nabla} \varphi_{\boldsymbol{j}}, \varphi_{\boldsymbol{i}}) + \mu(\boldsymbol{\nabla} \varphi_{\boldsymbol{j}}, \boldsymbol{\nabla} \varphi_{\boldsymbol{i}})$, and $\mathbf{F}_{\boldsymbol{i}} \doteq (g, \varphi_{\boldsymbol{i}})$ for $\boldsymbol{i}, \boldsymbol{j} \in \mathcal{N}_h$. Notice that we have not applied the boundary conditions yet. To apply boundary conditions the space of test functions is restricted to $v_h \in V_{h0}$, and the force vector is redefined as $\mathbf{F}_{\boldsymbol{i}} \doteq (g, \varphi_{\boldsymbol{i}}) - (\partial_t \overline{u}_h, \varphi_{\boldsymbol{i}}) - (\mathbf{v} \cdot \boldsymbol{\nabla} \overline{u}_h, \varphi_{\boldsymbol{i}}) - \mu(\boldsymbol{\nabla} \overline{u}_h, \boldsymbol{\nabla} \varphi_{\boldsymbol{i}})$.

3.2.4 Monotonicity properties

In this section we define all the properties that we demand our scheme to fulfill. In this case, since we are using a space–time discretization, it becomes more useful to define

these properties in a space–time sense. This means that the variation of u_h in the temporal direction will also be taken into account to define an extremum. Hence, we define the concept of a local discrete extremum as follows.

Definition 3.2.1 (Local Discrete Extremum)**.** *The function $u_h \in V_h$ has a local discrete minimum (resp. maximum) on $\boldsymbol{i} \in \mathcal{N}_h$, if $u_{\boldsymbol{i}} \leq u_{\boldsymbol{j}}$ (resp. $u_{\boldsymbol{i}} \geq u_{\boldsymbol{j}}$) $\forall \boldsymbol{j} \in \mathcal{N}_h^{\boldsymbol{i}}$.*

For problems that satisfy a maximum principle, e.g., problem (3.2) with $g = 0$, it is also important to define the concepts of local and global space–time DMP. The latter is a slightly weaker property than the former, but it is more useful for the discussion in this chapter. A local DMP is a stronger property because it implies that no oscillations can appear, while the global DMP only implies that the global extrema are located at the boundary conditions.

Definition 3.2.2 (Local space–time DMP)**.** *A solution $u_h \in V_h$ satisfies the local discrete maximum principle if for every $\boldsymbol{i} \in \mathcal{N}_h$*

$$\min_{\boldsymbol{j} \in \mathcal{N}_h^{\boldsymbol{i}} \setminus \{\boldsymbol{i}\}} u_{\boldsymbol{j}} \leq u_{\boldsymbol{i}} \leq \max_{\boldsymbol{j} \in \mathcal{N}_h^{\boldsymbol{i}} \setminus \{\boldsymbol{i}\}} u_{\boldsymbol{j}}.$$

Definition 3.2.3 (Global space–time DMP)**.** *A solution $u_h \in V_h$ satisfies the global discrete maximum principle if the global extrema are located at boundary conditions, i.e., for every $\boldsymbol{i} \in \mathcal{N}_h$*

$$\min\left(\min_{\substack{\boldsymbol{x}\in\partial\Omega,\\ t\in[0,T)}} \overline{u}_h(\boldsymbol{x},t),\ \min_{\boldsymbol{x}\in\Omega} u_{h0}(\boldsymbol{x})\right) \leq u_{\boldsymbol{i}} \leq \max\left(\max_{\substack{\boldsymbol{x}\in\partial\Omega,\\ t\in[0,T)}} \overline{u}_h(\boldsymbol{x},t),\ \max_{\boldsymbol{x}\in\Omega} u_{h0}(\boldsymbol{x})\right).$$

Finally, let us recall the definition of linearity-preservation, which is a desired property to achieve high-order convergence in smooth regions (see [66]).

Definition 3.2.4 (Linearity-preservation)**.** *A stabilization term, $\mathbf{B}_{\boldsymbol{ij}}(u_h)$, is said to be linearity-preserving if, for a solution that is linear in all directions in the neighborhood of $\boldsymbol{x}_{\boldsymbol{i}}$, then the stabilization term becomes null, i.e., $\mathbf{B}_{\boldsymbol{ij}}(u_h) = 0$ if $u_h(\boldsymbol{x}) \in \mathcal{P}_1(\Omega_{\boldsymbol{i}})$ where $\Omega_{\boldsymbol{i}}$ is the convex hull defined by the set of neighboring Greville abscissae $\{\boldsymbol{x}_{\boldsymbol{j}}\}_{\boldsymbol{j}\in\mathcal{N}_h^{\boldsymbol{k}},\ \boldsymbol{k}\in\mathcal{S}_h^{\boldsymbol{i}}}$.*

3.3 Lipschitz-continuous nonlinear stabilization

In this section we define a nonlinear stabilization operator, $B_h(w_h; u_h, v_h)$, to be added to the discrete problem (3.2), such that it satisfies at least the global DMP in Def. 3.2.3. Let us define $\mathbf{B}_{\boldsymbol{ij}}(u_h) \doteq B_h(u_h; \varphi_{\boldsymbol{j}}, \varphi_{\boldsymbol{i}})$. We also enforce that, for any $u_h \in V_h$, $\mathbf{B}_{\boldsymbol{ij}}(u_h)$

1. has compact support: $\mathbf{B}_{\boldsymbol{ij}}(u_h) = 0$ if $\boldsymbol{j} \notin \mathcal{S}_h^{\boldsymbol{i}}$,
2. is symmetric: $\mathbf{B}_{\boldsymbol{ij}}(u_h) = \mathbf{B}_{\boldsymbol{ji}}(u_h)$,
3. is conservative: $\sum_{\boldsymbol{j}\in\mathcal{S}_h^{\boldsymbol{i}}\setminus\{\boldsymbol{i}\}} \mathbf{B}_{\boldsymbol{ij}}(u_h) = -\mathbf{B}_{\boldsymbol{ii}}(u_h)$.

In order to achieve these requirements, we recall the stabilization term in chapter 2, which is defined as

$$B_h(w_h; u_h, v_h) \doteq \sum_{\boldsymbol{i}\in\mathcal{N}_h} \sum_{\boldsymbol{j}\in\mathcal{S}_h^{\boldsymbol{i}}} \nu_{\boldsymbol{ij}}(w_h) v_{\boldsymbol{i}} u_{\boldsymbol{j}} \ell(\boldsymbol{i},\boldsymbol{j}), \tag{3.3}$$

for any $w_h,\ u_h,\ v_h \in V_h$. Here, ℓ is the graph-Laplacian operator defined as $\ell(\boldsymbol{i},\boldsymbol{j}) = 2\delta_{\boldsymbol{ij}} - 1$ (see Chapter 2 or [39]), and $\nu_{\boldsymbol{ij}}(w_h)$ is the artificial diffusion defined as

$$\begin{aligned} \nu_{\boldsymbol{ij}}(w_h) &\doteq \max\{\alpha_{\boldsymbol{i}}(w_h)\mathbf{K}_{\boldsymbol{ij}}, 0, \alpha_{\boldsymbol{j}}(w_h)\mathbf{K}_{\boldsymbol{ji}}\} \quad \text{for} \quad \boldsymbol{j} \in \mathcal{S}_h^{\boldsymbol{i}}\backslash\{\boldsymbol{i}\}, \\ \nu_{\boldsymbol{ii}}(w_h) &\doteq \sum_{\boldsymbol{j}\in\mathcal{S}_h^{\boldsymbol{i}}\backslash\{\boldsymbol{i}\}} \nu_{\boldsymbol{ij}}(w_h). \end{aligned} \tag{3.4}$$

We denote by $\alpha(w_h)$ the shock detector used for computing the artificial diffusion parameter. The idea behind the definition of this detector is to ensure that the global DMP defined in Def. 3.2.3 is satisfied using a minimal amount of artificial diffusion, i.e., the lower admissible value of $\nu_{\boldsymbol{ij}}$. A shock detector must be a positive real number, which takes value 1 when $u_h(\boldsymbol{x}_{\boldsymbol{i}})$ is an inadmissible value of u_h (i.e., local discrete extremum) and smaller than 1 otherwise; to have linearity preservation (see Def. 3.2.4), it must be equal to 0 for $u_h \in \mathcal{P}_1(\Omega \times (0,T])$. In this section, we propose an isotropic approach for $\alpha_{\boldsymbol{i}}(w_h)$, which consists in using the shock detector in the previous chapter (see Sect. 2.3) in all directions (including time).

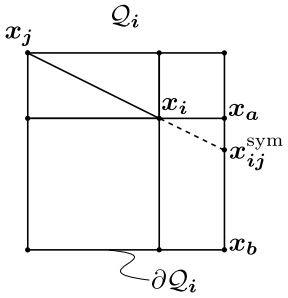


FIGURE 3.2: Representation of the polytope $\mathcal{Q}_i$ in two dimensions, the symmetric node $\boldsymbol{x}_{\boldsymbol{ij}}^{\text{sym}}$ of $\boldsymbol{x}_{\boldsymbol{j}}$ with respect to $\boldsymbol{x}_{\boldsymbol{i}}$, $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$.

In order to introduce the shock detector, let us recall some useful notation from the previous chapter. Let $\boldsymbol{r}_{\boldsymbol{ij}} = \boldsymbol{x}_{\boldsymbol{j}} - \boldsymbol{x}_{\boldsymbol{i}}$ be the vector pointing from Greville abscissae $\boldsymbol{x}_{\boldsymbol{i}}$ to $\boldsymbol{x}_{\boldsymbol{j}}$ with $\boldsymbol{i},\boldsymbol{j} \in \mathcal{N}_h$ and $\hat{\boldsymbol{r}}_{\boldsymbol{ij}} \doteq \frac{\boldsymbol{r}_{\boldsymbol{ij}}}{|\boldsymbol{r}_{\boldsymbol{ij}}|}$. Let us take the set of Greville abscissae $\boldsymbol{x}_{\boldsymbol{j}}$ for $\boldsymbol{j} \in \mathcal{N}_h^i\backslash\{\boldsymbol{i}\}$ as vertices of a polytope in $d+1$ dimensions. In particular, let us name this polytope $\mathcal{Q}_{\boldsymbol{i}}$. Let $\boldsymbol{x}_{\boldsymbol{ij}}^{\text{sym}}$ be the point at the intersection between $\partial\mathcal{Q}_{\boldsymbol{i}}$ and the line that passes through $\boldsymbol{x}_{\boldsymbol{i}}$ and $\boldsymbol{x}_{\boldsymbol{j}}$ that is not $\boldsymbol{x}_{\boldsymbol{j}}$ (see Fig. 3.2). The set of all $\boldsymbol{x}_{\boldsymbol{ij}}^{\text{sym}}$ for all $\boldsymbol{j} \in \mathcal{N}_h^{\boldsymbol{i}}\backslash\{\boldsymbol{i}\}$ is represented with $\mathcal{N}_h^{\boldsymbol{i},\text{sym}}$. We define $\boldsymbol{r}_{\boldsymbol{ij}}^{\text{sym}} \doteq \boldsymbol{x}_{\boldsymbol{ij}}^{\text{sym}} - \boldsymbol{x}_{\boldsymbol{i}}$. Given $\boldsymbol{x}_{\boldsymbol{ij}}^{\text{sym}}$ in two dimensions, let as call $\boldsymbol{a}$ and $\boldsymbol{b}$ the indices of the vertices such that they define the edge in $\partial\mathcal{Q}_{\boldsymbol{i}}$ that contains $\boldsymbol{x}_{\boldsymbol{ij}}^{\text{sym}}$. We define $u_{\boldsymbol{j}}^{\text{sym}}$ as the linear interpolation of $u_{\boldsymbol{a}}$ and $u_{\boldsymbol{b}}$ at $\boldsymbol{x}_{\boldsymbol{ij}}$, i.e. $u_{\boldsymbol{ij}}^{\text{sym}} \doteq u_{\boldsymbol{a}} \frac{\boldsymbol{x}_{\boldsymbol{b}} - \boldsymbol{x}_{\boldsymbol{ij}}^{\text{sym}}}{\boldsymbol{x}_{\boldsymbol{b}} - \boldsymbol{x}_{\boldsymbol{a}}} + u_{\boldsymbol{b}} \frac{\boldsymbol{x}_{\boldsymbol{ij}}^{\text{sym}} - \boldsymbol{x}_{\boldsymbol{a}}}{\boldsymbol{x}_{\boldsymbol{b}} - \boldsymbol{x}_{\boldsymbol{a}}}$. For higher dimensions, $u_{\boldsymbol{ij}}^{\text{sym}}$ is defined

analogously. Given the facet of $\partial\mathcal{Q}_i$ where $\boldsymbol{x}_{ij}^{\text{sym}}$ lies, u_{ij}^{sym} is the linear interpolation at $\boldsymbol{x}_{ij}^{\text{sym}}$ of the control points whose Greville abscissae are at the same facet.

Notice that it is essential to use Greville abscissae since they satisfy that for a linear function $u_h \in \mathcal{P}_1$, $u_h(\boldsymbol{x}_i) = u_i$. Therefore, one can construct easily linear approximations of the unknown gradients that are exact for $u_h \in \mathcal{P}_1$. Furthermore, one can define the jump and the mean of a linear approximation of the unknown gradient at Greville abscissa $\boldsymbol{x}_i$ in direction $\boldsymbol{r}_{ij}$ as

$$\llbracket \boldsymbol{\nabla} u_h \rrbracket_{ij} \doteq \frac{u_j - u_i}{|\boldsymbol{r}_{ij}|} + \frac{u_j^{\text{sym}} - u_i}{|\boldsymbol{r}_{ij}^{\text{sym}}|},$$

$$\{\!\{ |\boldsymbol{\nabla} u_h \cdot \hat{\boldsymbol{r}}_{ij}| \}\!\}_{ij} \doteq \frac{1}{2}\left(\frac{|u_j - u_i|}{|\boldsymbol{r}_{ij}|} + \frac{|u_j^{\text{sym}} - u_i|}{|\boldsymbol{r}_{ij}^{\text{sym}}|} \right).$$

In this chapter we will use the same shock detector developed in Chapter 2, which reads

$$\alpha_i(u_h) \doteq \begin{cases} \left[\dfrac{\left| \sum_{j \in \mathcal{N}_h^i} \llbracket \boldsymbol{\nabla} u_h \rrbracket_{ij} \right|}{\sum_{j \in \mathcal{N}_h^i} 2 \{\!\{ |\boldsymbol{\nabla} u_h \cdot \hat{\boldsymbol{r}}_{ij}| \}\!\}_{ij}} \right]^q & \text{if} \quad \sum_{j \in \mathcal{N}_h^i} \{\!\{ |\boldsymbol{\nabla} \cdot \hat{\boldsymbol{r}}_{ij}| \}\!\}_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases}. \tag{3.5}$$

From Lm. 2.3.1 we know that (3.5) is valued between 0 and 1, and it is only equal to one if $u_h(\boldsymbol{x}_i)$ is a local discrete extremum (in a space–time sense as in Def. 3.2.1). Since the linear approximations of the unknown gradients are exact for $u_h \in \mathcal{P}_1$, the shock detector vanishes when the solution is linear in all dimensions (including time). This result follows directly from Th. 2.4.5.

Supplementing the discrete problem (3.2) with the above stabilization term, the stabilized problem reads: Find $u_h \in V_h$ such that $u_h = \overline{u}_h$ on $\partial\Omega$, $u_h = u_{0h}$ at $t = 0$, and

$$(\partial_t u_h, v_h) + (\mathbf{v} \cdot \boldsymbol{\nabla} u_h, v_h) + \mu(\boldsymbol{\nabla} u_h, \boldsymbol{\nabla} v_h) + B_h(u_h; u_h, v_h) = (g, v_h), \quad \forall v_h \in V_h, \tag{3.6}$$

which in turn can be expressed in matrix form as

$$\tilde{\mathbf{K}}_{ij}(u_h) u_j = \mathbf{F}_i, \tag{3.7}$$

where $\tilde{\mathbf{K}}_{ij}(u_h) \doteq \mathbf{K}_{ij} + \mathbf{B}_{ij}(u_h)$ for $i, j \in \mathcal{N}_h$.

Theorem 3.3.1 (DMP)**.** *The solution of the discrete problem* (3.7) *using the shock detector* (3.5) *satisfies the global DMP in Def. 3.2.3 if $g = 0$ and, for every control point $i \in \mathcal{N}_h$ such that u_i is a local discrete extremum, it holds:*

$$\tilde{\mathbf{K}}_{ij}(u_h) \leq 0, \, \forall j \in \mathcal{S}_h^i \backslash \{i\}, \qquad \sum_{j \in \mathcal{S}_h^i} \tilde{\mathbf{K}}_{ij}(u_h) = 0. \tag{3.8}$$

Moreover, the resulting scheme is linearity-preserving as defined in Def. 3.2.4, i.e. $\mathbf{B}_{ij}(u_h) = 0$ for $u_h \in \mathcal{P}_1$.

Proof. Let us assume that $u_{\boldsymbol{i}}$ is a discrete maximum. Then, (3.7), for $g = 0$ and before applying boundary conditions, reads

$$\sum_{\boldsymbol{j}\in\mathcal{S}_h^{\boldsymbol{i}}} \tilde{\mathbf{K}}_{\boldsymbol{ij}}(u_h)u_{\boldsymbol{j}} = \mathbf{0}.$$

Therefore, $u_{\boldsymbol{i}}$ can be computed as

$$u_{\boldsymbol{i}} = \frac{\sum_{\boldsymbol{j}\in\mathcal{S}_h^{\boldsymbol{i}}\setminus\{\boldsymbol{i}\}} \tilde{\mathbf{K}}_{\boldsymbol{ij}}(u_h)u_{\boldsymbol{j}}}{\tilde{\mathbf{K}}_{\boldsymbol{ii}}(u_h)}.$$

Since $u_{\boldsymbol{i}}$ is an extremum, which implies $\alpha_{\boldsymbol{i}} = 1$, the stabilization term ensures (3.8) by construction. Hence, the coefficients that multiply $u_{\boldsymbol{j}}$ are in $[0, 1]$, and the sum of all these coefficients add up to one. Therefore, $u_{\boldsymbol{i}}$ is a convex combination of its neighbors (including boundary conditions $\overline{u}$). Since $u_{\boldsymbol{i}}$ is a maximum and a convex combination of its neighbors, then $u_{\boldsymbol{k}} = u_{\boldsymbol{i}}$ for some $\boldsymbol{k} \in \mathcal{S}_h^{\boldsymbol{i}}$. In that case, we can write

$$u_{\boldsymbol{i}} = \frac{\tilde{\mathbf{K}}_{\boldsymbol{ik}}(u_h)u_{\boldsymbol{k}}}{\tilde{\mathbf{K}}_{\boldsymbol{ii}}(u_h)} + \frac{\sum_{\boldsymbol{j}\in\mathcal{S}_h^{\boldsymbol{i}}\setminus\{\boldsymbol{i},\boldsymbol{k}\}} \tilde{\mathbf{K}}_{\boldsymbol{ij}}(u_h)u_{\boldsymbol{j}}}{\tilde{\mathbf{K}}_{\boldsymbol{ii}}(u_h)}.$$

Since $u_{\boldsymbol{k}} = u_{\boldsymbol{i}}$,

$$\left(1 - \frac{\tilde{\mathbf{K}}_{\boldsymbol{ik}}(u_h)}{\tilde{\mathbf{K}}_{\boldsymbol{ii}}(u_h)}\right) u_{\boldsymbol{i}} = \frac{\sum_{\boldsymbol{j}\in\mathcal{S}_h^{\boldsymbol{i}}\setminus\{\boldsymbol{i},\boldsymbol{k}\}} \tilde{\mathbf{K}}_{\boldsymbol{ij}}(u_h)u_{\boldsymbol{j}}}{\tilde{\mathbf{K}}_{\boldsymbol{ii}}(u_h)}.$$

Therefore, it can also be proved that $u_{\boldsymbol{i}}$ is a convex combination of all its neighbors *but* $u_{\boldsymbol{k}}$,

$$u_{\boldsymbol{i}} = \frac{\sum_{\boldsymbol{j}\in\mathcal{S}_h^{\boldsymbol{i}}\setminus\{\boldsymbol{i},\boldsymbol{k}\}} \tilde{\mathbf{K}}_{\boldsymbol{ij}}(u_h)u_{\boldsymbol{j}}}{\left(1 - \frac{\tilde{\mathbf{K}}_{\boldsymbol{ik}}(u_h)}{\tilde{\mathbf{K}}_{\boldsymbol{ii}}(u_h)}\right)\tilde{\mathbf{K}}_{\boldsymbol{ii}}(u_h)} = \frac{\sum_{\boldsymbol{j}\in\mathcal{S}_h^{\boldsymbol{i}}\setminus\{\boldsymbol{i},\boldsymbol{k}\}} \tilde{\mathbf{K}}_{\boldsymbol{ij}}(u_h)u_{\boldsymbol{j}}}{\tilde{\mathbf{K}}_{\boldsymbol{ii}}(u_h) - \tilde{\mathbf{K}}_{\boldsymbol{ik}}(u_h)}.$$

Proceeding analogously, one can also prove that $u_{\boldsymbol{k}}$ is a convex combination of all its neighbors *but* $u_{\boldsymbol{i}}$. Hence, we know that the value of $u_{\boldsymbol{i}} = u_{\boldsymbol{k}}$ is bounded by all their neighbors. At this point, the same reasoning can be applied to any of their neighbors. Thus, by induction, we know that extrema at any control point are bounded by the boundary conditions. Thus, the global DMP is satisfied.

From Th. 2.4.5, $\alpha_{\boldsymbol{j}} = 0$ for any $\boldsymbol{j} \in \mathcal{S}_h^{\boldsymbol{i}}$ if $u_h \in \mathcal{P}_1(\Omega_{\boldsymbol{i}})$ where $\Omega_{\boldsymbol{i}}$ is the convex hull defined by the set of neighboring Greville abscissae $\{\boldsymbol{x_j}\}_{\boldsymbol{j}\in\mathcal{N}_h^{\boldsymbol{k}},\ \boldsymbol{k}\in\mathcal{S}_h^{\boldsymbol{i}}}$. By definition, the stabilization term also vanishes if $\alpha_{\boldsymbol{j}} = 0$ for $\boldsymbol{j} \in \mathcal{S}_h^{\boldsymbol{i}}$ (see (3.3) and (3.4)). Therefore, the scheme is linearity-preserving as defined in Def. 3.2.4. □

Theorem 3.3.2. *The diffusion defined in* (3.4) *introduces the minimal amount of artificial dissipation such that condition* (3.8) *is satisfied when* $q \to \infty$.

Proof. The proof follows the same lines as Th. 2.4.4. We do not include it for the sake of conciseness. □

Finally, we proof Lipschitz continuity of the stabilization term. In this particular case, the proof follows the same reasoning as in Th. 2.6.1. Let us recall the definition of the semi-norm generated by the graph-Laplatian required to show Lipschitz continuity,

$$|w|_\ell \doteq \sqrt{\frac{1}{2}\sum_{\boldsymbol{i}\in\mathcal{N}_h}\sum_{j\in\mathcal{S}_h^{\boldsymbol{i}}}(w_{\boldsymbol{i}}-w_{\boldsymbol{j}})^2}.$$

In addition, we will also need the $L^2(\Omega)$ norm denoted as $\|\cdot\|$ and the $L^\infty(\Omega)$ expressed as $\|\cdot\|_\infty$. We do not include all details for the sake of conciseness and refer the reader to the previously cited work.

Theorem 3.3.3. *Let $V_h^{\text{adm}} \subset V_h$ be the subspace of functions that satisfy the global DMP in Def. 3.2.3, then $\mathbf{B}(\cdot)$ with the shock detector (3.5) is Lipschitz continuous in V_h^{adm} for $u_h \in V_h$ and bounded q, since*

$$(\mathbf{B}(u)-\mathbf{B}(v),z) \leq Cq(h^d+\|\mathbf{v}\|_\infty h^{d-1}\delta t+\mu h^{d-2}\delta t)|u-v|_\ell|z|_\ell$$

is satisfied.

Proof. The proof is analogous to the one in Th. 2.6.1. The only difference arises from the bound for

$$\mathbf{K}_{\boldsymbol{ij}} = (\partial_t\varphi_{\boldsymbol{j}},\varphi_{\boldsymbol{i}})+(\mathbf{v}\cdot\boldsymbol{\nabla}\varphi_{\boldsymbol{j}},\varphi_{\boldsymbol{i}})+\mu(\boldsymbol{\nabla}\varphi_{\boldsymbol{j}},\boldsymbol{\nabla}\varphi_{\boldsymbol{i}}).$$

In this case, using Cauchy–Schwarz inequality, the inverse inequality $\|\boldsymbol{\nabla}v_h\| \leq Ch^{-1}\|v_h\|$ for $v_h \in V_h$, and $\|\varphi_i\| \leq Ch^{d/2}$, we get

$$\begin{aligned}\mathbf{K}_{\boldsymbol{ij}} &\leq \|\partial_t\varphi_{\boldsymbol{j}}\|\|\varphi_{\boldsymbol{i}}\|+\|\mathbf{v}\|_\infty\|\boldsymbol{\nabla}\varphi_{\boldsymbol{j}}\|\|\varphi_{\boldsymbol{i}}\|+\mu\|\boldsymbol{\nabla}\varphi_{\boldsymbol{j}}\|\|\boldsymbol{\nabla}\varphi_{\boldsymbol{i}}\|\\ &\leq C_1h^d+\|\mathbf{v}\|_\infty C_2h^{d-1}\delta t+\mu C_3h^{d-2}\delta t,\end{aligned}$$

where d is the number of spatial dimensions, h is the distance between knots for the spatial directions and δt is the distance between knots for the time direction. □

Notice that whereas C is uniform with respect to the mesh size, it can depend on the polynomial order of the discretization.

3.4 Time partitioned scheme

Hitherto, we have only considered the solution of the whole space-time problem at once. In order to substantially reduce the computational cost, we propose the division of the time integration in several time subdomains, considering the proposed space-time formulation at every subdomain. Namely, the problem (3.1) set in $\Omega\times(0,T]$, will be decomposed in $\Omega\times(t_l,t_{l+1}]$ for $0\leq l\leq n_t-1$, with $t_0=0$ and $t_{n_t}=T$. We define the length of the subdomain as $\Delta t \doteq t_{l+1}-t_l$, and restrict its possible values to $\Delta t = n\,p\,\delta t$ for some $n\in\mathbb{N}$, where p is the order of the spline space, and δt is the distance between

knots in the temporal direction. Notice that we are only using discretizations formed from the tensor product of discretizations in 1D. Therefore, with the particular choice of Δt, t_l will always be the temporal coordinate of a layer of knots. Hence, performing this kind of partitions is straightforward. Other partitions might be considered, however we choose the previous one because it is particularly simple to use it in our implementation.

The approximation space of splines for every subdomain is obtained as follows. Given the complete domain $\Omega \times (0, T]$, we discretize it as described in Sect. 3.2.2, resulting in a spline space V_h. Then in order to reduce the coupling between partitions, we insert p knots at t_l. The resulting spaces at each subdomain, say V_h^l, are fully decoupled. However, due to causality in time there exists a sequential coupling between subdomains, i.e. the information travels in the positive direction. In other words, the solution at subdomain l will affect the solution at $l+1$, but not the opposite. Therefore, we impose that the initial conditions at subdomain $l+1$ are equal to the solution at the final time of subdomain l, i.e. $u_h^{l+1}(t_l) \doteq u_h^l(t_l)$. After imposing this restriction, the complete approximation space, $\tilde{V}_h$, is C^0, and coupled sequentially. Hence, each subdomain can be solved sequentially, and thus the computational cost is significantly reduced.

The partitioned space–time scheme with nonlinear stabilization reads as follows. For $l = 1, ..., n_t$; find $u_h^l \in V_h^l$ such that $u_h^l = \overline{u}_h$ on $\partial\Omega$, $u_h^l(\boldsymbol{x}, t_l) = u_h^{l-1}(\boldsymbol{x}, t_l)$ with $u_h^0(\boldsymbol{x}, t_0) = u_{0h}$, and

$$(\partial_t u_h^l, v_h) + (\mathbf{v} \cdot \boldsymbol{\nabla} u_h^l, v_h) + \mu(\boldsymbol{\nabla} u_h^l, \boldsymbol{\nabla} v_h) + B_h(u_h^l; u_h^l, v_h) = (g, v_h), \quad \forall v_h \in V_h^l, \tag{3.9}$$

Due to the partition u_h will only be piecewise continuous in time. Let us proof now that the scheme still satisfies the global DMP.

Lemma 3.4.1. *The solution of problem* (3.9)*, using the shock detector defined in* (3.5)*, satisfies the global DMP (Def. 3.2.3) if $g = 0$ in $\Omega \times (0, T]$ and, for every control point $\boldsymbol{i} \in \mathcal{N}_h$ such that $u_{\boldsymbol{i}}$ is a local discrete extremum, conditions* (3.8) *hold.*

Proof. From Th. 3.3.1 it is easy to see that conditions (3.8) hold for the first subdomain. Hence, the solution at the first subdomain at time t_1, $u_h^1(\boldsymbol{x}, t_1)$, is bounded by the initial and boundary conditions. Since $u_h^1(\boldsymbol{x}, t_1)$ are the initial conditions for the second subdomain, then again from Th. 3.3.1 it is known that the solution in the second subdomain is bounded by $u_h^1(\boldsymbol{x}, t_1)$, and thus by the initial and boundary conditions. Therefore, by induction, we conclude that the global DMP is satisfied in the whole domain. □

3.5 Differentiable stabilization

In this section, we introduce a different version of the previous operators. As exposed in [4, 5], the regularization of all non-differentiable operators in the stabilization term improves the nonlinear convergence, and allows us to use Newton's method. We use the same strategy introduced in Chapter 2. Then, the shock detector reads

$$\alpha_{\varepsilon_h,i}(u_h) \doteq \left[Z\left(\frac{\left| \sum_{\boldsymbol{j}\in\mathcal{N}_h^{\boldsymbol{i}}} [\![\boldsymbol{\nabla} u_h]\!]_{\boldsymbol{ij}} \right|_{1,\varepsilon_h} + \gamma_h}{\sum_{\boldsymbol{j}\in\mathcal{N}_h^{\boldsymbol{i}}} 2 \left\{\!\!\left\{ |\boldsymbol{\nabla} u_h \cdot \hat{\boldsymbol{r}}_{\boldsymbol{ij}}|_{2,\varepsilon_h} \right\}\!\!\right\}_{\boldsymbol{ij}} + \gamma_h} \right) \right]^q , \tag{3.10}$$

where $\gamma_h > 0$ is a parameter to prevent division by zero, and the regularized absolute values by

$$|x|_{1,\varepsilon_h} = \sqrt{x^2 + \varepsilon_h}, \qquad |x|_{2,\varepsilon_h} = \frac{x^2}{\sqrt{x^2 + \varepsilon_h}}.$$

Notice that $|x|_{2,\varepsilon_h} \leq |x| \leq |x|_{1,\varepsilon_h}$. With this regularization, the quotient in the shock detector might become greater than one, thus we need to smoothly limit its value to one. To this end we recall $Z(x)$, which reads

$$Z(x) \doteq \begin{cases} 2x^4 - 5x^3 + 3x^2 + x & x < 1 \\ 1 & x \geq 1 \end{cases} ,$$

and clearly is twice differentiable and bounded above by 1. The differentiable version still satisfies the requirements for a shock detector, i.e., it is a real value in $[0, 1]$ and it is equal to 1 if $u_{\boldsymbol{i}}$ is a local extrema. This result follows directly from Lm. 2.7.1.

Furthermore, for the definition of the artificial diffusion we need to regularize the maximum function. We choose again the same strategy as in the previous chapter, and define $\max_{\sigma_h}\{\cdot,\cdot\}$ as

$$\max_{\sigma_h}\{x, y\} \doteq \frac{|x - y|_{1,\sigma_h}}{2} + \frac{x + y}{2}.$$

Finally, we can define the twice differentiable artificial diffusion parameter as

$$\tilde{\nu}_{\boldsymbol{ij}}(w_h) \doteq \begin{cases} \max_{\sigma_h}\left\{\max_{\sigma_h}\left\{\alpha_{\varepsilon_h,\boldsymbol{i}}(w_h)\mathbf{K}_{\boldsymbol{ij}}, \alpha_{\varepsilon_h,\boldsymbol{j}}(w_h)\mathbf{K}_{\boldsymbol{ji}}\right\}, 0\right\} & \text{for} \quad \boldsymbol{j} \neq \boldsymbol{i} \\ \displaystyle\sum_{\boldsymbol{j}\in\mathcal{S}_h^{\boldsymbol{i}}\setminus\{\boldsymbol{i}\}} \tilde{\nu}_{\boldsymbol{ij}}(w_h) & \end{cases} ,$$

and the stabilization operator reads

$$\tilde{B}_h(w_h; u_h, v_h) \doteq \sum_{\boldsymbol{i}\in\mathcal{N}_h} \sum_{\boldsymbol{j}\in\mathcal{S}_h^{\boldsymbol{i}}} \tilde{\nu}_{ij}(w_h) v_{\boldsymbol{i}} u_{\boldsymbol{j}} \ell(\boldsymbol{i}, \boldsymbol{j}).$$

In order to obtain a differentiable operator, we have added a set of regularizations that rely on different parameters, e.g., σ_h, ε_h, γ_h. Giving a proper scaling of these parameters is essential to recover theoretic convergence rates. In particular, we use the following relations

$$\sigma_h = \sigma|\mathbf{v}|^2 L^{2(d-3)} h^{2(p+1)}, \quad \varepsilon_h = \varepsilon L^{-4} h^2, \quad \gamma_h = L^{-1}\gamma,$$

where d is the spatial dimension of the problem, L is a characteristic length, and σ, ε, and γ are of the order of the unknown.

3.6 Numerical experiments

In this section we present some numerical experiments showing the behavior of the scheme previously introduced. First, a convergence analysis is performed in order to assess the correctness of the proposed scheme and its implementation. Then, we assess the performance of the proposed stabilization method for high-order discretizations, including a brief analysis of the effect of the regularization.

3.6.1 1D Transient Diffusion

The purpose of this test is assessing the partitioned time integration scheme in Sect. 3.4. To this end, we solve the following problem for $t \in (0, 1]$ and $x \in \Omega \doteq (0, 1)$,

$$\begin{cases} \partial_t u + \partial_{xx} u = f & \text{in } \Omega \times (0, 1] \\ u = 0 & \text{at } \partial\Omega \end{cases} , \tag{3.11}$$

where $f \doteq 2(6x^2 - 6x + 1)(t(t-1))^2 + 2t(t-1)(2t-1)(x(x-1))^2$. This problem has $u = (x(x-1))^2\ (t(t-1))^2$ as exact solution. We perform a convergence analysis where the mesh is successively refined in the time direction for first, second, and third order discretizations. In particular, the distance between knots in the temporal direction is refined as $\delta t = \{0.2, 0.1, 0.05, 0.025, 0.0125\}$ for the first order discretization. In spatial directions, the distance is small enough ($h = 1/400$) to prevent that spatial discretization errors affect the analysis. Second and third order discretizations are obtained using the following k-refinement (see [29] for more details). We refine the discretization such that the number of control points increase at the same rate as a Lagrangian FE discretization does when its order is increased. Fig. 3.3 shows the result of k-refinements to $p = 2$ and $p = 3$ discretizations, for an interior subset of the discretization. Henceforth, we will use this kind of k-refinement in order to increase the discretization order.

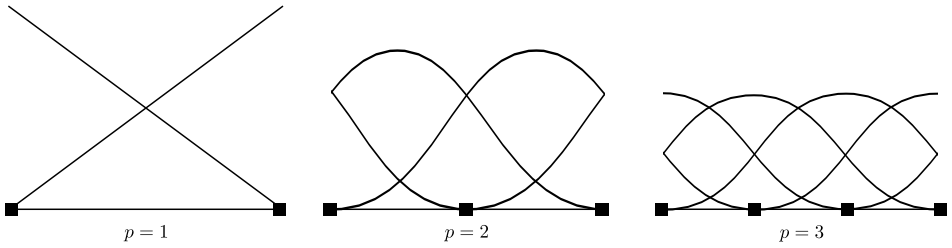
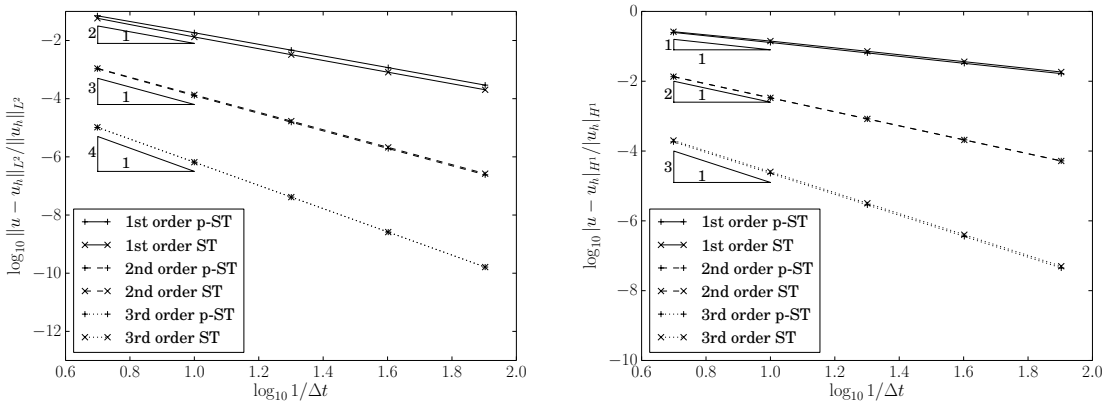


FIGURE 3.3: Second and third order discretizations obtained from the k-refinement of an initial first order discretization. Notice that shape functions are depicted for interior knots, at boundary knots shape functions become interpolatory, see Fig. 3.1.

We measure the relative L^2 norm and H^1 semi-norm of error in the whole space–time domain, and compute the resulting convergence rate. Errors in L^2 norm and H^1 semi-norm are depicted in Fig. 3.4 (a) and (b), respectively. In Table 3.1 the measured convergence rates are shown for the original non-partitioned scheme and the proposed in



(A) Relative L^2 norm of the error. (B) Relative H^1 norm of the error.

FIGURE 3.4: Convergence in time results for problem (3.11), using standard and partitioned space-time schemes.

Sect. 3.4. We observe a slight increase in the error for the partitioned scheme. However, the obtained results show optimal convergence rates for both schemes introduced above.

TABLE 3.1: Measured convergence rates in L^2 norm and H^1 semi-norm, for problem (3.11).

Order	Method	L^2 convergence	H^1 convergence
1	p-ST	-1.98	-0.98
1	ST	-2.04	-0.96
2	p-ST	-3.03	-2.00
2	ST	-3.00	-2.01
3	p-ST	-3.99	-3.00
3	ST	-3.99	-2.99

p-ST: Partitioned space-time, ST: space-time.

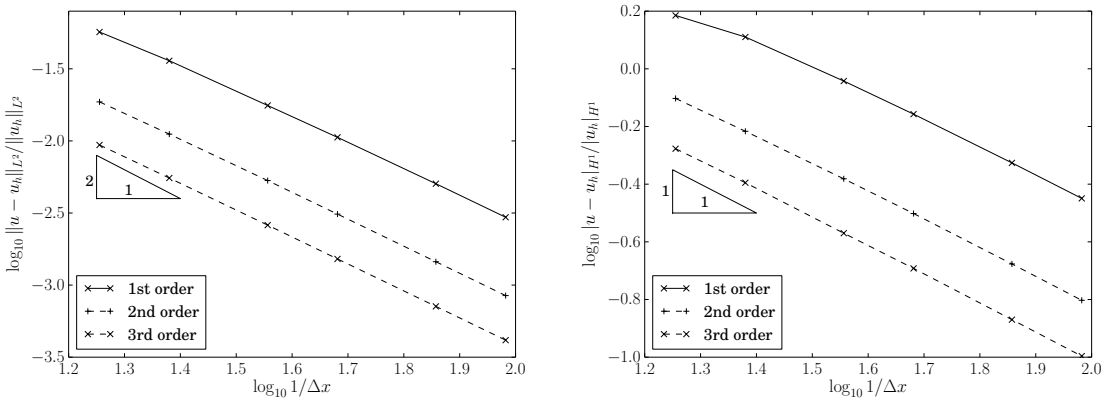
3.6.2 Steady convection

In this experiment we assess the convergence of the stabilized schemes introduced in Sect. 3.5. We use a steady pure convection problem with a non-monotonic smooth solution. In particular, we solve the following problem for $x \in \Omega \doteq [0,1]^2$,

$$\begin{cases} \mathbf{v} \cdot \boldsymbol{\nabla} u = 0 & \text{in } \Omega \\ u = u_D & \text{at } \Gamma_{\text{in}} \end{cases}, \tag{3.12}$$

where $u_D = \sin\left(2\pi\left(x - \frac{y}{\tan\theta}\right)\right)$, $\mathbf{v} = (\cos\theta, \sin\theta)$, and $\theta = \pi/3$. The analytical solution of the above problem reads $u = \sin\left(2\pi\left(x - \frac{y}{\tan\pi/3}\right)\right)$. The convergence analysis is performed for first, second, and third order discretizations, i.e. $p = \{1, 2, 3\}$. We use a standard nonlinear solver (see [5] for details), with a nonlinear tolerance $\frac{u^{k+1} - u^k}{u^k} < 10^{-6}$. The following stabilization parameters have been used: $q = 10$, $\varepsilon = 10^{-5}$, $\sigma = 10^{-6}$, $\gamma = 10^{-10}$. The selection of these parameters is based on the outcome of previous works [4, 5] and Sect. 3.6.3.

Convergence plots are shown in Fig. 3.5 and the corresponding convergence rates in Table 3.2. As expected, it is observed that the scheme recovers second order convergence in the L^2 error norm and first in the H^1 error semi-norm. It is known that the stabilized scheme should recover second order convergence for $p = 1$. However, due to peak clipping errors, higher convergence rates are not expected even if a higher order discretization is used [58]. In any case, we do observe that the error diminishes as the discretization order is increased using the k-refinement previously defined.



(A) Relative L^2 norm of the error. (B) Relative H^1 semi-norm of the error.

FIGURE 3.5: Convergence in space results for problem (3.12).

TABLE 3.2: Measured convergence rates in L^2 and H^1 norms, for problem (3.12).

Order	L^2 convergence	H^1 convergence
1	1.77	0.88
2	1.85	0.96
3	1.86	0.99

3.6.3 Nonlinear convergence

In the current test, we aim to briefly analyze the effect of the stabilization parameters on the nonlinear convergence of the method. To this end, we solve the following 1D pure convection problem with discontinuous initial conditions.

$$\begin{cases} \partial_t u + \mathbf{v} \cdot \boldsymbol{\nabla} u = 0 & \text{in } \Omega \times [0, T) \\ u = u_0 & \text{at } t = 0 \\ u = u_D & \text{at } \partial\Omega \end{cases}, \tag{3.13}$$

where $\mathbf{v} \doteq 1$, $\Omega \doteq (0, 1]$, $T = 0.5$, and $u_0 \doteq 1 - H_0(x - 0.25)$, where H_0 is the well-known zero-centered Heaviside function. First and second discretization orders are used in a coarse mesh of 25×25 control points. To obtain the second order mesh, we perform the k-refinement as in the previous experiment.

We refer the reader to [4, 5] for a deeper analysis on the effect of each regularization parameter. Therein, the same family of shock detectors is used in the context of first order cG and dG Lagrangian FEs. In this chapter, we analyze the effect of the regularization globally using a fixed relation between the different parameters. In particular, we use the following parameters: $\gamma = 10^{-10}$, $\sigma = \zeta$, $\varepsilon = \zeta^2$, where $\zeta = \{10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. Furthermore, the effect is also compared as q is incremented, particularly for $q = \{1, 2, 5, 10\}$. In addition, the non-regularized version is also used to show the improvement in the nonlinear convergence. The relaxed Picard and hybrid nonlinear solvers presented in [5] are used, and the nonlinear tolerance is set to 10^{-5}.

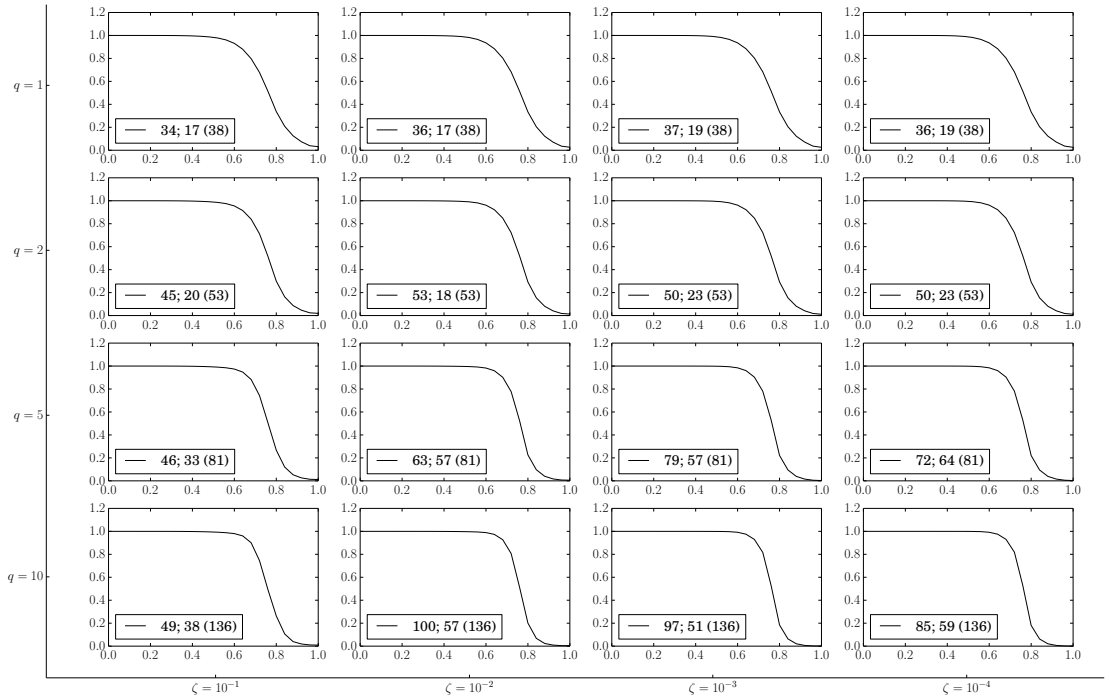


FIGURE 3.6: Effect of the regularization parameters for first order discretizations. The numbers in legends are the number of nonlinear iterations performed. First number is for relaxed Picard and the next for hybrid scheme, both for the regularized stabilization. The number in brackets is the number of iterations required to converge the non-differentiable method using relaxed Picard scheme.

Fig. 3.6 and 3.7 show the results for first and second order discretizations, respectively. In general terms, as q is increased or ζ is decreased, sharper solutions are observed. However, nonlinear iterations increase. As expected, the hybrid method outperforms the relaxed Picard method. Even though it requires more nonlinear iterations, the non-regularized detector might be a simpler (parameter-free) alternative to the regularized one. Finally, it is worth mentioning that a slight increase in the required number of iterations is observed as the discretization order is increased. However, the obtained results are more accurate, i.e., the discontinuity becomes sharper.

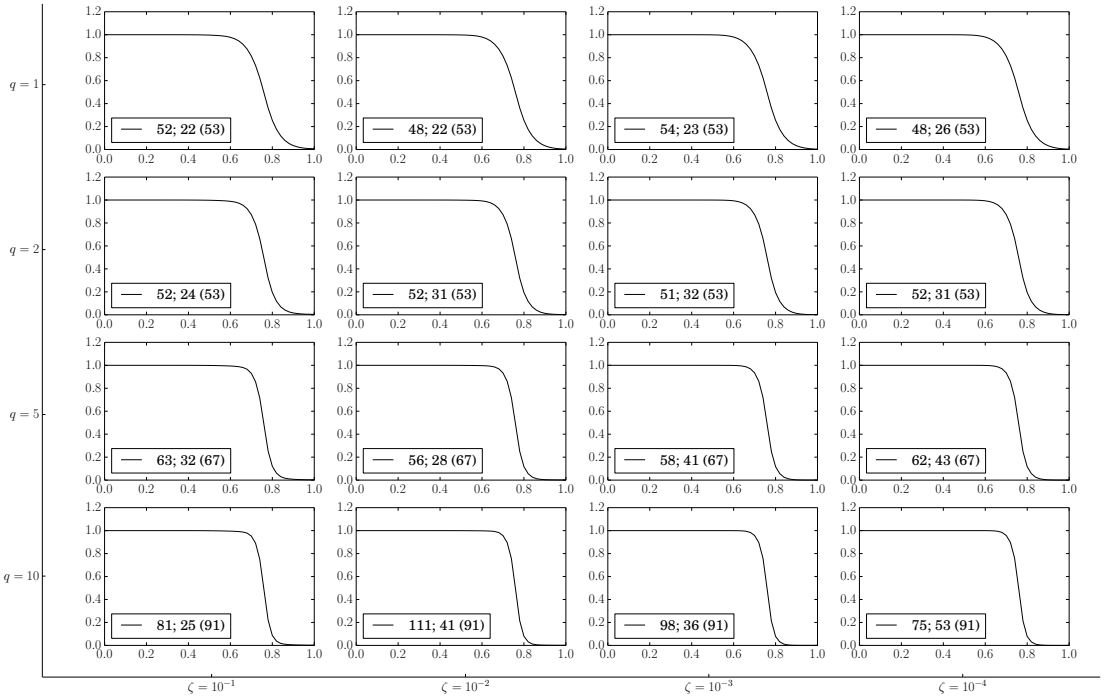


FIGURE 3.7: Effect of the regularization parameters for second order discretization. The numbers in legends are the number of nonlinear iterations performed. First number is for relaxed Picard and the next for hybrid scheme, both for the regularized stabilization. The number in brackets is the number of iterations required to converge the non-differentiable method using relaxed Picard scheme.

3.6.4 1D Sharp layer propagation

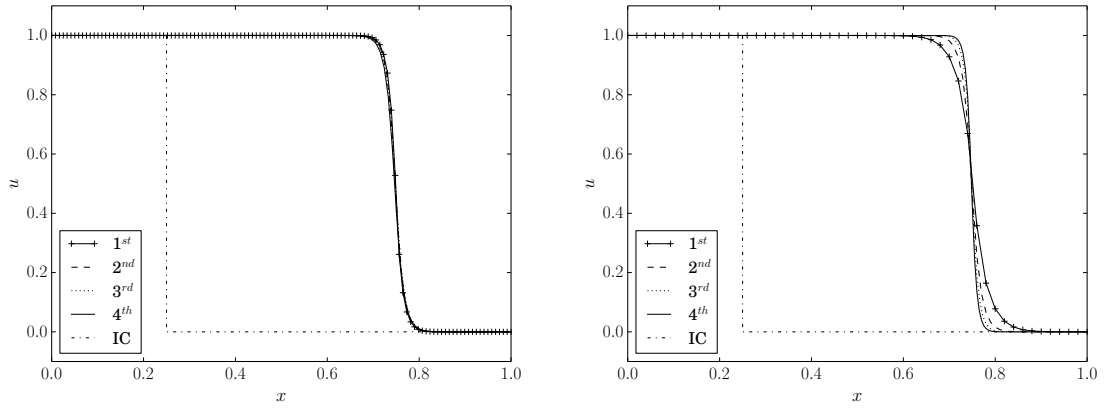
The performance of the stabilization schemes is analyzed as the discretization order is increased. To this end, we use again the previous problem (3.13). The regularization parameters are kept fixed, while the discretization is modified both in terms of the order of accuracy and the number of control points.

We use a nonlinear tolerance of 10^{-5}. The regularization parameters used are $q = 10$, $\varepsilon = 10^{-8}$, $\sigma = 10^{-6}$, and $\gamma = 10^{-10}$. With this setting, we solve the above problem using a discretization that keeps the number of control points fixed as the order is increased, and another one using the k-refinement defined in the previous experiment. For the former, we use a discretization of 120 by 60 control points. For the latter, we start with a first order discretization of 120 by 60 control points and refine as previously explained.

In Fig. 3.8(a) the solution at $t = 0.5$ is shown for different orders and fixed number of control points, and using the k-refinement in Fig. 3.8(b). We observe that for non-smooth solutions, fixing the number of control points and increasing the order does not improve the results. This is a consequence of the underlying discretization properties. The support of the shape functions becomes larger as the order is increased. Therefore, nonsmooth solutions become slightly more smeared. In the case of Fig. 3.8(b), as expected, we observe better approximations as the order is increased using the k-refinement.

Hence, better results might be expected as the order is increased for problems that combine discontinuities and smooth profiles.

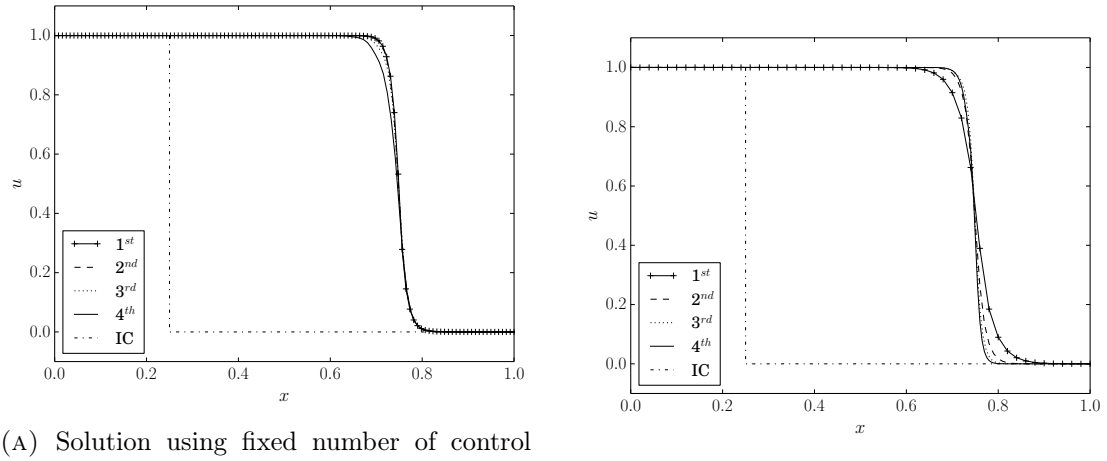
In Fig. 3.9, similar results are shown when using the time integration scheme proposed in Sect. 3.4. A small degradation of the results can be seen in Fig. 3.9(a) as we increase the discretization order. In a similar trend, we observe less improvement in Fig. 3.9(b) than in Fig. 3.8(b). We attribute this degradation to the time partitions, which becomes more evident as the subdomains are smaller. In particular, at the boundary of each partition the method might slightly increase the amount of diffusion introduced. At these boundaries, the shock detector rely on a smaller domain to determine if the DMP is satisfied. Therefore, it is more likely to introduce more diffusion. On the other hand, the partition itself modifies the scheme introducing some error as shown in Sect. 3.6.1.



(A) Solutions increasing the order while keeping fixed the number of control points.

(B) Solution increasing the order using the k-refinement process described above.

FIGURE 3.8: Solution of problem (3.13) at $t = 0.5$ for first to fourth order discretizations.



(A) Solution using fixed number of control points, and time integration defined in Sect. 3.4 with 5 partitions.

(B) Solution using k-refinement, and time integration defined in Sect. 3.4 with 5 partitions.

FIGURE 3.9: Solution of problem (3.13) at $t = 0.5$ for first to fourth order discretizations.

3.6.5 Boundary layer

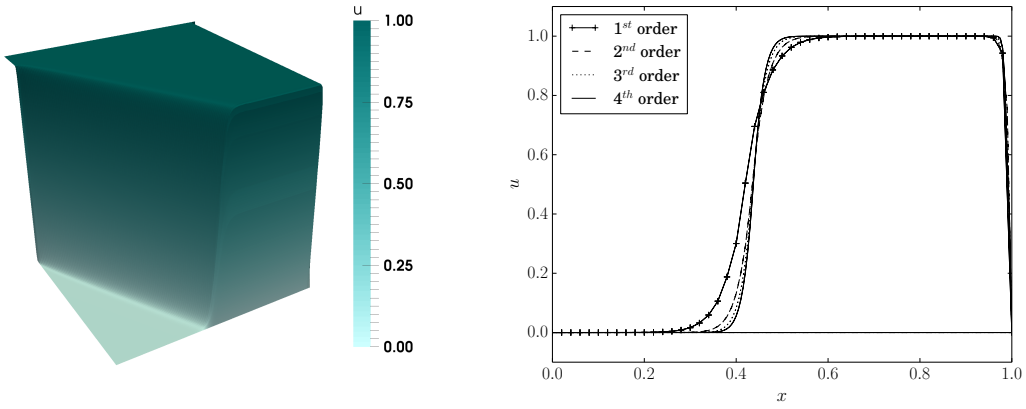
In this section the effect of the discretization order in a convection–diffusion problem is analyzed. To this end, we solve a problem with the propagation of a sharp layer and a boundary layer. In particular, we solve for $\Omega \doteq [0,1]^2$

$$\begin{cases} -10^{-4}\Delta u + \mathbf{v}\cdot\boldsymbol{\nabla} u = 0 & \text{in } \Omega \\ u = u_D & \text{at } \partial\Omega \end{cases}, \tag{3.14}$$

where $\mathbf{v} = (\cos\theta, \sin\theta)$, $\theta = -\pi/3$, and the boundary conditions are defined as

$$u_D = \begin{cases} \frac{1}{2} + \frac{1}{\pi}\arctan\left(10^{-4}\left(y - 5/6\right)\right) & \text{if } y = 0 \\ 0 & \text{otherwise} \end{cases}.$$

For this test, we use the following settings: a nonlinear tolerance of 10^{-8}, $q = 2$, $\varepsilon = 10^{-8}$, $\sigma = 10^{-6}$, and $\gamma = 10^{-10}$. In Fig. 3.10(a), the solution for $p = 4$ is depicted. The converged solution does not exhibit any oscillation. Very sharp layers are obtained for this parameter setting. In Fig. 3.10(b), we show the profile of the solution at $y = 0.1$ for different orders. In this case, we start with a discretization of 50 control points per direction. Then, we increase the order using the k-refinement used previously. As previously observed for transient problems, we observe an improvement of the solution as the order is increased.



(A) 3D representation of the solution. (B) Profiles at $y = 0.1$.

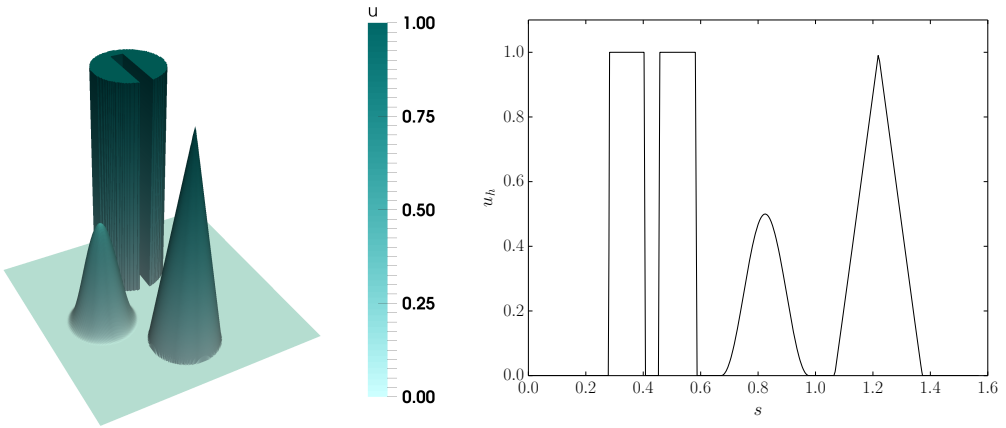
FIGURE 3.10: Solution of problem (3.14) using scheme (3.6), and different discretization orders.

3.6.6 Three Body rotation

Finally, we solve the transient pure convection problem (3.13) in $\Omega \times (0,1]$ for $\Omega = [0,1]^2$, with $\mathbf{v} = (-2\pi(y-0.5), 2\pi(x-0.5))$. Initial conditions are given in [56]. Its interpolation in a first order 200×200 control point mesh is depicted in Fig. 3.11(a). The analytical solution of this problem is simply the translation of the profiles in the direction of the

convection. In particular, for $t = 1$, one revolution is completed and the solution is equal to the initial conditions. The purpose of this test is to evaluate how diffusive is the proposed scheme. We perform this evaluation evolving the solution until $t = 1$ and comparing the results with the initial conditions.

The solution is computed using scheme (3.6) in combination with the shock detector in (3.10). We use the following parameters for the stabilization: $q = 10$, $\sigma = 10^{-6}$, $\varepsilon = 10^{-8}$, and $\gamma = 10^{-10}$. Different meshes, time partitions, and discretization orders are used in this experiment. We start with a linear discretization of 100×100 control points in space, and 500 in time divided in 125 subdomains. Then, we increase the discretization order to $p = 2$ using the k-refinement. In order to compare first and second order discretizations, but using a similar number of control points we use a discretization with 200×200 control points in space, and 1000 in time divided in 250 subdomains. Finally, we assess the effect of the partitions in the temporal direction. We compare the previous discretization of $100 \times 100 \times 500$ control points divided in 125 subdomains, with the same discretization divided in 250 subdomains. We do the same comparison for the second order discretization using 125 subdomains and when it is divided in 250 subdomains.

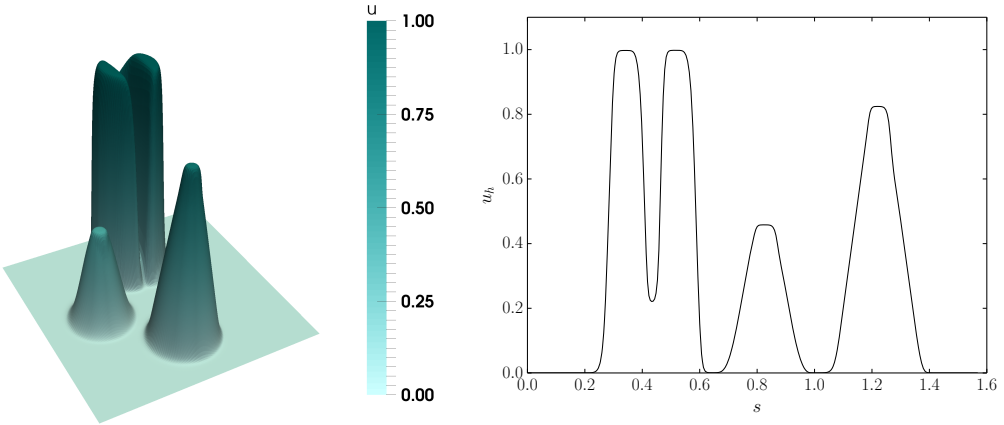


(A) Initial conditions of the 3 body rotation. (B) Profile at $\sqrt{(x-0.5)^2 + (y-0.5)^2} = 0.25$.

FIGURE 3.11: Three body rotation test initial conditions.

Fig. 3.13 shows the solutions for 100×100 meshes, and 125 subdomains in time, whereas Fig. 3.14 show the ones for 250 subdomains. In both cases, a great improvement can be observed as we increase the discretization order. However, the computational cost is also increased. It is interesting to compare the solutions for first and second order discretizations using meshes with similar amount of control points, namely solutions at Fig. 3.12 and 3.13(b). For this particular problem, using a higher order discretization with similar number of control points does not improve the solution, which it is actually slightly more diffusive for $p = 2$. It is also worth mentioning that increasing the discretization order does not modify the behavior of the solution in terms of clipping or

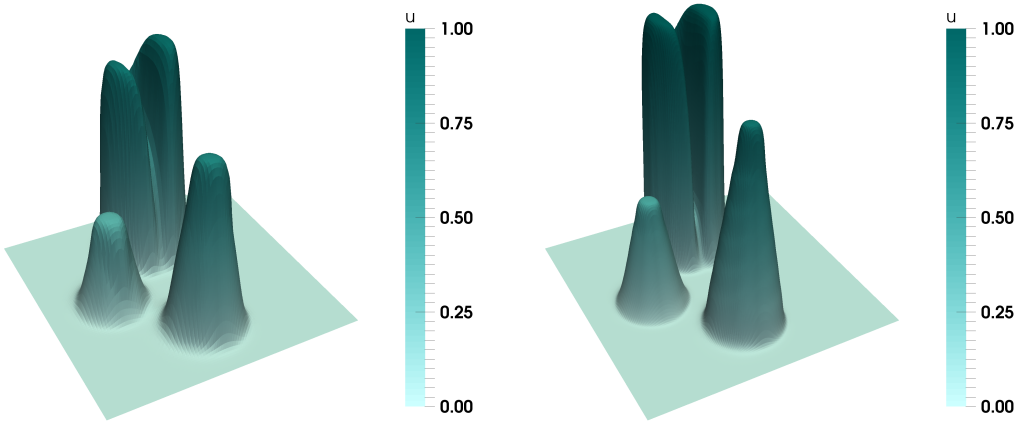
terracing. Comparing Figs. 3.14(a) and 3.13(a), we observe that the scheme becomes more dissipative as the number of partitions is increased. This is even clearer in Fig. 3.15, where the profile of the solution at $s \doteq \{(x, y) : \sqrt{(x-0.5)^2 + (y-0.5)^2} = 0.25\}$ is depicted.



(A) 3D view.

(B) Profile at $\sqrt{(x-0.5)^2 + (y-0.5)^2} = 0.25$.

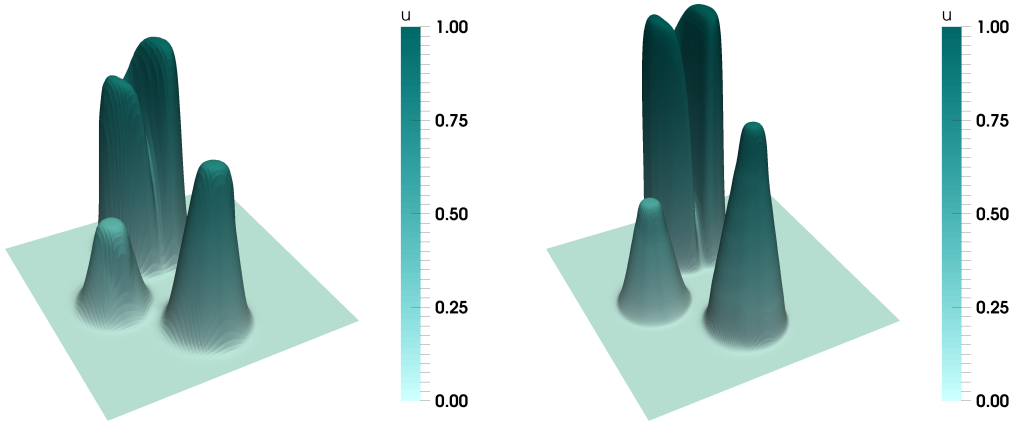
FIGURE 3.12: Three body rotation test results at $t = 1$ using scheme (3.6), $q = 10$, $\sigma = 10^{-6}$, $\varepsilon = 10^{-8}$, and $\gamma = 10^{-10}$. A first order discretization of $200 \times 200 \times 1000$ control points is used with 250 subdomains in the temporal direction.



(A) Solution for a first order discretization.

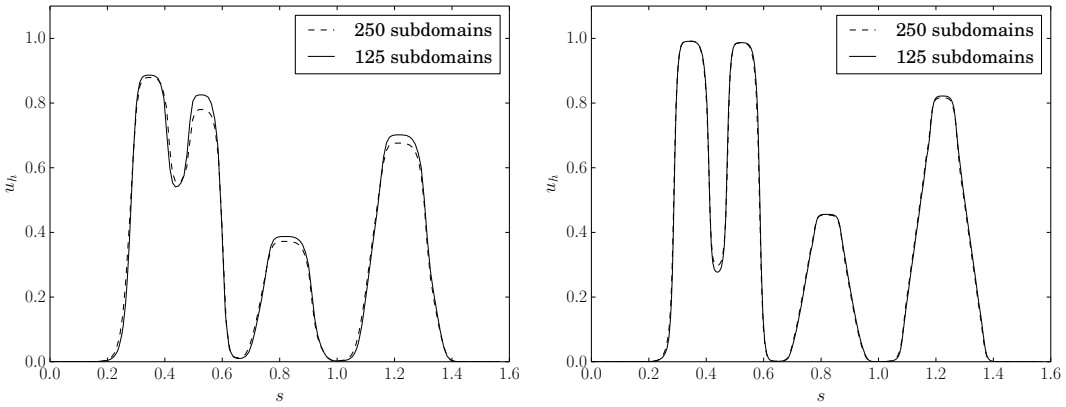
(B) Solution after one k-refinement.

FIGURE 3.13: Three body rotation test results at $t = 1$ using scheme (3.6), $q = 10$, $\sigma = 10^{-6}$, $\varepsilon = 10^{-8}$, and $\gamma = 10^{-10}$. A first order discretization of $100 \times 100 \times 500$ control points is used. The second order discretization is obtained using k-refinement. 125 subdomains in the temporal direction have been used.



(A) Solution for first order discretization. (B) Solution after one k-refinement.

FIGURE 3.14: Three body rotation test results at $t = 1$ using scheme (3.6), $q = 10$, $\sigma = 10^{-6}$, $\varepsilon = 10^{-8}$, and $\gamma = 10^{-10}$. A first order discretization of $100 \times 100 \times 500$ control points is used. The second order discretization is obtained using k-refinement. 250 subdomains in the temporal direction have been used.



(A) Solution for first order discretization. (B) Solution after one k-refinement.

FIGURE 3.15: Three body rotation test profiles for $t = 1$ at $\sqrt{(x-0.5)^2 + (y-0.5)^2} = 0.25$ using scheme (3.6), $q = 10$, $\sigma = 10^{-6}$, $\varepsilon = 10^{-8}$, and $\gamma = 10^{-10}$. A first order discretization of $100 \times 100 \times 500$ control points is used. The second order discretization is obtained using k-refinement. 125 and 250 subdomains in the temporal direction have been used.

3.7 Conclusions

In this chapter an extension of the stabilization in Chapter 2 to isogeometric analysis methods have been developed. The proposed method is unconditionally DMP preserving for arbitrary high-order discretizations in space and time without any CFL-like condition. Furthermore, it is shown to be linearity-preserving in a space–time sense. Moreover, the regularized version is shown to yield better convergence behavior, especially when for the hybrid Picard–Newton method.

Moreover, the numerical experiments show that increasing the discretization order yield much better solutions. However, as the order is increased the number of control points and the computational cost is also increased. On the contrary, if the order is increased while the number of control points is fixed, then similar or even slightly more diffusive results are obtained for non-smooth solutions. Hence, for problems with regions of smooth and non-smooth solutions a high-order is expected to outperform linear discretizations with similar amount of control points. Furthermore, a method capable of providing solutions that satisfy the DMP for high-order discretizations is of special interest in *hp*-adaptive schemes, since the usage of first order discretizations in shocks is not required.

In addition, a partitioned scheme that does not harm any monotonicity property is presented. This scheme reduces significantly the computational cost of the original space–time scheme. It is important to mention, that this partitioning slightly increases the error. However, this approach allows finer meshes, and thus, in practice better solutions can be obtained.

Chapter 4

Local bounds preserving FEs for first order conservation laws

This chapter is focused on the design of nonlinear stabilization techniques for the finite element approximation of the Euler equations in steady form and the implicit time integration of the transient form. A differentiable local bounds preserving method has been developed, which combines a Rusanov artificial diffusion operator and a differentiable shock detector. Nonlinear stabilization schemes are usually stiff and highly nonlinear. We attempt to mitigate this issue by the differentiability properties of the proposed method. Moreover, in order to further improve the nonlinear convergence, we also propose a continuation method for a subset of the stabilization parameters. The resulting method has been successfully applied to steady and transient problems with complex shock patterns. Numerical experiments show that it is able to provide sharp and well resolved shocks. The importance of the differentiability is assessed by comparing the developed scheme with its non-differentiable counterpart. Numerical experiments suggest that, up to moderate nonlinear tolerances, the method exhibits improved robustness and nonlinear convergence behavior for steady problems. In the case of transient problem, we also observe a reduction in the computational cost.

4.1 Introduction

The solution of many hyperbolic conservation laws satisfy a number of mathematical and physics constraints. These can include for example maximum principles, positivity and monotonicity preservation. A classical example are the Euler equations, where positivity must be preserved for the density, internal energy, and therefore also the pressure. In general, discretizations can yield non-physical solutions that violate these properties, leading to nonlinear instabilities. This is a well known issue. In the context of explicit finite volume schemes or dG FE methods, several stabilized schemes have already been developed [27, 68, 91]. However, explicit time integrators need to resolve all time scales for stability reasons. For some multiple-time-scale problems, this can often imply very stringent stability conditions on the time-step size. If the fastest time scales are critical to the dynamics, and therefore of scientific or engineering interest, then explicit time integrators are well suited. On the contrary, implicit time integration is favored when

the smallest time-scales are not relevant to the dynamics of interest. For example, the ability to integrate accurately and efficiently for longer time-scale simulations can be essential in some plasma physics applications [54]. Moreover, clearly explicit schemes become inefficient for steady problems because one is forced to solve all the hydrodynamic evolution until the steady state is reached. Therefor the design of implicit *stabilized* schemes that preserve the previously mentioned structure continues to be an important challenge.

In this chapter, we focus on implicit cG FE approximations of steady and transient shock hydrodynamics problems. It is well known that the Galerkin method (without any modification) is generally unstable for hyperbolic problems and yields solutions with spurious oscillations. Therefore, FE schemes are usually supplemented with additional artificial diffusion terms. Those terms are designed such that the resulting scheme satisfies the properties of the continuous problem. E.g., positive density and internal energy or non-decreasing entropy.

Developing a discretization scheme able to preserve all the properties of the continuous problem can be very challenging. This becomes especially complex for nonlinear hyperbolic systems of equations. Only a few methods that preserve the continuous problem properties have been proposed. For example, for explicit finite difference and finite volume methods, schemes in [35, 45, 46] are able to preserve these properties for Euler and the p-system. Recently, Guermond and Popov [42] have extended these works to explicit cG FE schemes, and it is applicable to any first order hyperbolic system with bounded wave propagation speed. An alternative is to try to impose conditions based on the diagonalization of the problem. Since it is a hyperbolic system of equations, then there exist a set of *characteristic* variables for which the system can be diagonalized and written as a set of independent transport problems. At this point, one can use techniques developed for scalar problems. Hence, the stabilization methods are based on adapting the scalar techniques for characteristic variables to the system written in the original set of variables. Following this strategy, some progress has been recently made in stabilized FE schemes by making use of flux corrected transport (FCT) algorithms [61, 70, 74, 75]. The schemes proposed therein are based in two main ingredients. On the one hand, a diffusive term able to minimize or eliminate any oscillatory behavior. On the other hand, a limiter, or shock detector, to modulate the stabilization term and restrict its action to the vicinity of shocks.

Since we focus on fully implicit problems, it is important to consider the nonlinear convergence behavior of these schemes. It is known that for certain limiter choices the convergence of the nonlinear solver might be remarkably hard [57]. Recent progress has been made in this direction for scalar convection–diffusion problems [4, 5, 16]. In these studies, the authors are able to improve the nonlinear convergence by proposing differentiable stabilization terms to improve convergence rates of the Newton's iterative nonlinear solver.

In the current chapter, we extend the differentiable nonlinear stabilization in Chapter

2 to the Euler equations using the ideas from [61, 75] to define the artificial diffusion operators for hyperbolic systems of equations. The new method is applied to the steady and transient Euler equations, and its nonlinear convergence is assessed.

The remainder of this chapter is structured as follows. In Sect. 4.2 we present the CG discretization for scalar convection and Euler equations. Sect. 4.3 is devoted to the definition of the stabilization terms. We describe the nonlinear solvers used in Sect. 4.4. Then, in Sect. 4.5 we present the numerical experiments performed. Finally, we draw some conclusions in Sect. 4.6.

4.2 Preliminaries

In this section, we introduce the problem of interest and its FE discretization. At the end of the section, we also revisit a few properties usually requested to numerical schemes for solving hyperbolic problems.

4.2.1 Continuous problem

Let us consider an open bounded and connected domain, $\Omega \in \mathbb{R}^d$, where d is the number of spatial dimensions. Let $\partial\Omega$ be the Lipschitz continuous boundary of Ω. A first order hyperbolic problem can be written in conservative form as

$$\begin{cases} \partial_t \boldsymbol{u} - \boldsymbol{\nabla} \cdot \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{g}, & \text{in } \Omega \times (0,T], \\ u^\beta(x,t) = \bar{u}^\beta(x,t), & \text{on } \Gamma_{\text{in}}^\beta \times (0,T],\ \beta = 1,...,m, \\ \boldsymbol{u}(x,0) = \boldsymbol{u}_0(x), & x \in \Omega, \end{cases} \tag{4.1}$$

where $\boldsymbol{u} = \{u^\beta\}_{\beta=1}^m$ are $m \geq 1$ conserved variables, $\boldsymbol{f}$ is the physical flux, $\bar{u}^\beta(x,t)$ are the boundary values for the βth-component of $\boldsymbol{u}$, $\boldsymbol{u}_0(x)$ are the initial conditions, and $\boldsymbol{g}(x,t)$ is a function defining the body forces. Note that the flux, $\boldsymbol{f} : \mathbb{R}^m \to \mathbb{R}^{m\times d}$, is composed as $\boldsymbol{f} = \{\boldsymbol{f}_i\}_{i=1}^d$, where $\boldsymbol{f}_i : \mathbb{R}^m \to \mathbb{R}^m$ is the flux in the ith spatial direction. We denote by $\boldsymbol{f}' : \mathbb{R}^m \to \mathbb{R}^{m\times m\times d}$ the flux Jacobian. Let $\boldsymbol{n} \in \mathbb{R}^d$ be any direction vector. Since the system is hyperbolic the flux Jacobian in any direction is diagonalizable and has only real eigenvalues, i.e. $\boldsymbol{f}'(\boldsymbol{u}) \cdot \boldsymbol{n} = \sum_{i=1}^d \boldsymbol{f}_i'(\boldsymbol{u}) n_i$ is diagonalizable with real eigenvalues, say $\{\lambda_\beta\}_{\beta=1}^m$. These eigenvalues might have different multiplicities, and different signs. Hence, for a given direction, $\boldsymbol{n}$, each characteristic variable might be convected forward (along $\boldsymbol{n}$) or backwards (along $-\boldsymbol{n}$). Therefore, it is convenient to define inflow and outflow boundaries for each component. The inflow boundary for component β is defined as $\Gamma_{\text{in}}^\beta \doteq \{\boldsymbol{x} \in \partial\Omega \ : \ \lambda_\beta(\boldsymbol{f}'(\boldsymbol{u}) \cdot \boldsymbol{n}_{\partial\Omega}) \leq 0\}$, where $\boldsymbol{n}_{\partial\Omega}$ is the unit outward normal to the boundary, and λ_β is the βth-eigenvalue of the flux Jacobian. We define the outflow boundary as $\Gamma_{\text{out}}^\beta \doteq \partial\Omega \backslash \Gamma_{\text{in}}^\beta$. We refer the reader to [34, 43, 91] for a detailed discussion on boundary conditions for hyperbolic problems. We will also consider the steady counterpart of (4.1), which is obtained by dropping the time derivative term and the initial conditions.

Note that if $m = 1$, and $\boldsymbol{f}(u) \doteq \mathbf{v}u$ with $\mathbf{v}$ a divergence-free convection field, we recover the well known scalar convection problem. However, in this chapter we focus on Euler equations for ideal gases, which are recovered by defining

$$\boldsymbol{u} \doteq \begin{pmatrix} \rho \\ \boldsymbol{m} \\ \rho E \end{pmatrix}, \quad \boldsymbol{f} \doteq \begin{pmatrix} \boldsymbol{m} \\ \boldsymbol{m} \otimes \mathbf{v} + p\mathbb{I} \\ \mathbf{v}(\rho E + p) \end{pmatrix}, \text{ and } \boldsymbol{g} \doteq \begin{pmatrix} 0 \\ \boldsymbol{b} \\ \boldsymbol{b} \cdot \mathbf{v} + r \end{pmatrix},$$

where ρ is the density, E is the total energy, p is the pressure, $\boldsymbol{m} = \{m_1, \dots, m_d\}$, where $m_i = \rho v_i$, is the momentum, $\mathbf{v} = \{v_1, \dots, v_d\}$ is the velocity, $\boldsymbol{b} = \{b_1, \dots, b_d\}$ are the body forces, r is an energy source term per unit mass, and $\mathbb{I}$ an identity matrix of dimension d. In addition, the system is equipped with the ideal gas equation of state $p = (\gamma - 1)\rho\imath$, where $\imath = E - \frac{1}{2}\|\mathbf{v}\|^2$ is the internal energy, and γ is the adiabatic index.

4.2.2 Discretization

Let $\mathcal{T}_h$ be a conforming partition of Ω. The set of interpolation of the mesh $\mathcal{T}_h$ is represented with $\mathcal{N}_h$. For every node $i \in \mathcal{N}_h$, the node coordinates are represented with $\boldsymbol{x}_i$. We denote by $N = \text{card}(\mathcal{N}_h)$ the total number of nodes. The set of nodes belonging to a particular element $K \in \mathcal{T}_h$ is defined as $\mathcal{N}_h(K) \doteq \{i \in \mathcal{N}_h : \boldsymbol{x}_i \in K\}$. Moreover, Ω_i is the macroelement composed by the union of elements that contain node i, i.e., $\Omega_i \doteq \bigcup_{K \in \mathcal{T}_h,\, \boldsymbol{x}_i \in K} K$. To simplify the discussion below, we abuse notation and use i for both the node and its associated index.

We restrict the discussion in this chapter to first order FEs and define the FE space as follows. For simplicial meshes of Ω, we define $\boldsymbol{V}_h \doteq \left\{\boldsymbol{v}_h \in (\mathcal{C}^0(\Omega))^m : \boldsymbol{v}_h|_K \in (\mathcal{P}_1(K))^m \ \forall K \in \mathcal{T}_h\right\}$, where m is the number of components of $\boldsymbol{u}$, and $\mathcal{P}_1(\Omega)$ is the space of polynomials of total degree less than or equal to one. For d-cube partitions, we define $\boldsymbol{V}_h \doteq \left\{\boldsymbol{v}_h \in (\mathcal{C}^0(\Omega))^m : \boldsymbol{v}_h|_K \in (Q_1(K))^m, \forall K \in \mathcal{T}_h\right\}$, where $Q_1(K)$ is space of polynomials of partial degree less than or equal to one. Furthermore, we define the space $\boldsymbol{V}_{h0} \doteq \{\boldsymbol{v}_h \in \boldsymbol{V}_h \ : \ \boldsymbol{v}_h(\boldsymbol{x}) = 0 \ \forall \boldsymbol{x} \in \Gamma_{\text{in}}\}$. The functions $\boldsymbol{v}_h \in \boldsymbol{V}_h$ can be constructed as a linear combination of the basis $\{\boldsymbol{\varphi}_i^\beta\}_{i \in \tilde{\mathcal{N}}_h}^{1 \le \beta \le m}$ and nodal values $\boldsymbol{v}_i$, where $\boldsymbol{\varphi}_i^\beta = \varphi_i\{\delta_{1\beta}, \dots, \delta_{m\beta}\}$ is the shape function associated to the component β of node i, and $\delta_{\alpha\beta}$ is the Kronecker delta. Thus, only component β of $\boldsymbol{\varphi}_i^\beta$ is different from zero, which takes the value of the classical scalar shape function used in Lagrangian FE, φ_i. Hence, $\boldsymbol{v}_h = \sum_{i \in \tilde{\mathcal{N}}_h, 1 \le \beta \le m} \boldsymbol{\varphi}_i^\beta v_i^\beta = \sum_{i \in \tilde{\mathcal{N}}_h} \varphi_i \boldsymbol{v}_i$.

We use standard notation for Sobolev spaces. The $L^2(\omega)$ scalar product is denoted by $(\cdot, \cdot)_\omega$ for $\omega \subset \Omega$. However, we omit the subscript for $\omega \equiv \Omega$. The L^2 norm is denoted by $\|\cdot\|$. Using this notation, the weak form of problem (4.1) reads as follows. Find $\boldsymbol{u} \in L^2(\Omega)$ such that $u^\beta = \bar{u}^\beta$ on $\Gamma_{\text{in}}^\beta \times (0, T]$ for $\beta = 1, ..., m$ and

$$(\partial_t \boldsymbol{u}, \boldsymbol{v}) + (\boldsymbol{u}, \boldsymbol{f}'(\boldsymbol{u}) : \boldsymbol{\nabla v}) - (\boldsymbol{u}, \boldsymbol{n}_{\Gamma_{\text{out}}} \cdot \boldsymbol{f}'(\boldsymbol{u})\boldsymbol{v})_{\Gamma_{\text{out}}} = (\boldsymbol{g}, \boldsymbol{v}), \ \forall \boldsymbol{v} \in L_0^2(\Omega), \tag{4.2}$$

subject to appropriate initial conditions $\boldsymbol{u}(x,0)=\boldsymbol{u}_0(x)$. Note that the double contraction is applied as $\boldsymbol{f}'(\boldsymbol{u}):\boldsymbol{\nabla v}=\sum_{k,\gamma}\boldsymbol{f}'_k(\boldsymbol{u})^{\beta\gamma}\ \boldsymbol{v}_{\gamma,\beta}$.

In combination with the FE spaces described above for the spatial discretization the method of lines is being applied. The solution is approximated using $\boldsymbol{u}\approx\boldsymbol{u}_h=\sum_{i\in\tilde{\mathcal{N}}_h,\,1\le\beta\le m}\boldsymbol{\varphi}_i^{\beta}u_i^{\beta}=\sum_{i\in\tilde{\mathcal{N}}_h}\varphi_i\boldsymbol{u}_i$. In a similar manner, the fluxes are approximated as $\boldsymbol{f}\approx\boldsymbol{f}_h=\sum_{i\in\tilde{\mathcal{N}}_h,\,1\le\beta\le m}\boldsymbol{\varphi}_i^{\beta}\boldsymbol{f}(\boldsymbol{u}_i)^{\beta}=\sum_{i\in\tilde{\mathcal{N}}_h}\varphi_i\boldsymbol{f}(\boldsymbol{u}_i)$. For the sake of brevity, we use Backward Euler (BE) for the temporal discretization. Higher order time discretizations can be achieved using SSP RK methods (see [37]). In the latter case, a CFL-like condition arises to ensure that monotonicity properties in Sect. 4.2.3 are satisfied [60, 67].

The semi-discrete Galerkin FE approximation of problem (4.2) reads: find $\boldsymbol{u}_h\in\boldsymbol{V}_h$ such that $u_h^{\beta}=\bar{u}_h^{\beta}$ on $\Gamma_{\text{in}}^{\beta}$, $\boldsymbol{u}_h=\boldsymbol{u}_{0h}$ at $t=0$, and

$$(\partial_t\boldsymbol{u}_h,\boldsymbol{v}_h)+(\boldsymbol{u}_h,\boldsymbol{f}'_h(\boldsymbol{u}_h):\boldsymbol{\nabla v}_h)-(\boldsymbol{u}_h,\boldsymbol{n}_{\Gamma_{\text{out}}}\cdot\boldsymbol{f}'_h(\boldsymbol{u}_h)\boldsymbol{v}_h)_{\Gamma_{\text{out}}}=(\boldsymbol{g},\boldsymbol{v}_h),\ \forall\boldsymbol{v}_h\in\boldsymbol{V}_{h0},$$

where $\overline{u}_h^{\beta}$ and $\boldsymbol{u}_{0h}$ are admissible FE approximations of $\overline{u}^{\beta}$ and $\boldsymbol{u}_0$. In this context, we consider admissible any approximation that satisfies the maximum principle, i.e. it does not introduce new extrema. To obtain the fully discrete problem, we consider a partition of the time domain $(0,T]$ into n^{ts} sub-intervals of length $(t^n,t^{n+1}]$. Then, at every time step $n=0,\dots,n^{ts}-1$, the discrete problem consists in solving

$$\mathbf{M}\delta_t\mathbf{U}^{n+1}+\mathbf{K}\mathbf{U}^{n+1}=\mathbf{G},$$

where $\mathbf{U}^{n+1}\doteq[\boldsymbol{u}_1^{n+1},...,\boldsymbol{u}_N^{n+1}]^T$ is the vector of nodal values at time t^{n+1}, $\delta_t(\mathbf{U})\doteq\Delta t_{n+1}^{-1}(\mathbf{U}^{n+1}-\mathbf{U}^n)$, and $\Delta t_{n+1}\doteq(t^{n+1}-t^n)$. The $m\times m$-matrices relating nodes $i,j\in\mathcal{N}_h$ are given by

$$\begin{aligned}\mathbf{M}_{ij}^{\beta\gamma}&\doteq(\varphi_j,\varphi_i)\delta_{\beta\gamma},\\ \mathbf{K}_{ij}^{\beta\gamma}&\doteq(\varphi_j\delta_{\beta\xi},\boldsymbol{f}'_k(\boldsymbol{u}_j^{n+1})^{\xi\eta}\cdot\partial_k\varphi_i\delta_{\eta\gamma})-(\varphi_j\delta_{\beta\xi},n_k\cdot\boldsymbol{f}'_k(\boldsymbol{u}_j^{n+1})^{\xi\eta}\varphi_i\delta_{\eta\gamma})_{\Gamma_{\text{out}}},\\ \mathbf{G}_i^{\beta}&\doteq(\boldsymbol{g}^{\beta},\varphi_i),\end{aligned}$$

where Einstein summation applies, $\beta,\gamma,\xi,\eta\in\{1,\dots,m\}$ are the component indices, and $\delta_{\beta\gamma}$ is the Kronecker delta.

4.2.3 Stabilization properties

In this section, we introduce some concepts required for discussing the stabilization method presented in subsequent sections. In the case of hyperbolic systems of equations, some stabilization methods are based on schemes developed for scalar equations. Let us recall some definitions used for scalar problems.

Definition 4.2.1 (Local Discrete Extremum)**.** *The function $v_h\in V_h$ has a local discrete minimum (resp. maximum) on $i\in\mathcal{N}_h$ if $u_i\le u_j$ (resp. $u_i\ge u_j$) $\forall j\in\mathcal{N}_h(\Omega_i)$.*

Definition 4.2.2 (Local DMP). *A solution $u_h \in V_h$ satisfies the local discrete maximum principle if for every $i \in \mathcal{N}_h$*

$$\min_{j \in \mathcal{N}_h(\Omega_i)\setminus\{i\}} u_j \leq u_i \leq \max_{j \in \mathcal{N}_h(\Omega_i)\setminus\{i\}} u_j.$$

Definition 4.2.3 (LED). *A scheme is local extremum diminishing if, for every u_i that is a local discrete maximum (resp. minimum),*

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} \leq 0, \qquad \left(resp.\ \frac{\mathrm{d}u_i}{\mathrm{d}t} \geq 0\right),$$

is satisfied.

One possible strategy to satisfy the above properties consist on designing a scheme that yields a positive diagonal mass matrix and a stiffness matrix that satisfies

$$\sum_j \mathsf{A}_{ij} = 0, \quad \text{and} \quad \mathsf{A}_{ij} \leq 0 \ i \neq j. \tag{4.3}$$

In this case, it is possible to rewrite the system as

$$m_i \frac{du_i}{dt} + \sum_{j \in \mathcal{N}_h(\Omega_i)\setminus\{i\}} \mathsf{A}_{ij}(u_j - u_i) = 0, \quad \forall\, i \in \mathcal{N}_h. \tag{4.4}$$

As shown in [28] and [67], such a scheme satisfies the local DMP for steady problems and it is also LED when applied to transient problems.

The extension of these properties to hyperbolic systems is based on analyzing them in characteristic variables. Let us consider a one-dimensional linear hyperbolic system with a constant Jacobian flux, $\boldsymbol{f}'$. In this particular case, the continuous system can be diagonalized. Thus it is possible to discretize and solve for the characteristic variables. For example, for the set of characteristics variables, say W, the continuous system reads:

$$\partial_t W + \Lambda \partial_x W = 0, \tag{4.5}$$

where $\Lambda = diag(\lambda_1, ..., \lambda_m)$ is a diagonal m by m matrix. At this point, one can see the system as a set of independent scalar transport problems. Thus, it leads to a system with diagonal blocks after discretizing it with FEs.

Assuming conditions (4.3) are satisfied for every component of problem (4.5), then the scheme will be LED for each characteristic variable. Notice that this is equivalent to forcing the original (coupled) FE approximation to have negative semi-definite off-diagonal blocks. That is, the FE discretization of the problem in characteristic variables reads

$$(\varphi_j, \varphi_i)\partial_t W_j + \Lambda(\partial_x \varphi_j, \varphi_i) W_j = 0. \tag{4.6}$$

Since in this case it is a one dimensional linear problem, we can recover the original

problem using the fact that $W = R^{-1}U$, and $\boldsymbol{f}' = R\Lambda R^{-1}$. Multiplying (4.6) at the left by R,

$$(\varphi_j, \varphi_i)\partial_t R R^{-1} U_j + R\Lambda(\partial_x \varphi_j, \varphi_i) R^{-1} U_j = 0.$$

In this case, $(\partial_x \varphi_j, \varphi_i)$ is simply a scalar value. Hence, we are able to recover the original (coupled) problem FE discretization.

$$(\varphi_j, \varphi_i)\partial_t U_j + \boldsymbol{f}'(\partial_x \varphi_j, \varphi_i) U_j = 0.$$

Thus, if $\boldsymbol{f}'(\partial_x \varphi_j, \varphi_i)$ is negative semi-definite for $j \neq i$, then the problem in characteristic variables will satisfy conditions (4.3) for each variable.

In the case of more general multidimensional problems (e.g. Euler equations), this would *only* imply that the scheme is LED for a certain set of *local* characteristic variables. Furthermore, if the flux Jacobian $\boldsymbol{f}'$ is not linear, then even the definition of the matrix A_{ij} (relating nodes i and j) is not trivial. Let us recall the definition of these blocks for Euler equations

$$\mathsf{K}_{ij} \doteq (\varphi_j, \boldsymbol{f}_k'(\boldsymbol{u}_j) \cdot \partial_k \varphi_i) - (\varphi_j, n_k \cdot \boldsymbol{f}_k'(\boldsymbol{u}_j)\varphi_i)_{\Gamma_{\text{out}}} = -\boldsymbol{f}_k'(\boldsymbol{u}_j) \cdot (\partial_k \varphi_j, \varphi_i),$$

where we have undone integration by parts. It is easy to check that $\sum_j (\partial_k \varphi_j, \varphi_i) = 0$. Hence, we can write

$$\sum_{j \in \mathcal{N}_h(\Omega_i)} \mathsf{K}_{ij} \boldsymbol{u}_j = \sum_{j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}} -(\partial_k \varphi_j, \varphi_i)(\boldsymbol{f}_k'(\boldsymbol{u}_j) \cdot \boldsymbol{u}_j - \boldsymbol{f}_k'(\boldsymbol{u}_i) \cdot \boldsymbol{u}_i).$$

As previously stated, it is not straightforward in the case of Euler equations to rewrite the discrete problem in the form of (4.4). However, making use of special density-averaged variables it is possible to rewrite the previous expression as

$$\sum_{j \in \mathcal{N}_h(\Omega_i)} \mathsf{K}_{ij} \boldsymbol{u}_j = \sum_{j \in \mathcal{N}_h(\Omega_i) \setminus \{i\}} -\boldsymbol{f}_k'(\boldsymbol{u}_{ij}) \cdot (\partial_k \varphi_j, \varphi_i)(\boldsymbol{u}_j - \boldsymbol{u}_i),$$

where $\boldsymbol{u}_{ij}$ are the Roe mean values [82]. For an ideal gas, these are defined as

$$\rho_{ij} = \sqrt{\rho_i \rho_j}, \quad \boldsymbol{m}_{ij} = \frac{\boldsymbol{m}_i \sqrt{\rho_j} + \boldsymbol{m}_j \sqrt{\rho_i}}{\sqrt{\rho_i} + \sqrt{\rho_j}}, \quad (\rho E)_{ij} = \frac{1}{2 - \gamma} \left(\rho_{ij} H_{ij} - \frac{|\boldsymbol{m}_{ij}|^2}{2\rho_{ij}} \right),$$

where H_{ij} is the average enthalpy

$$H_{ij} = \frac{H_i \sqrt{\rho_i} + H_j \sqrt{\rho_j}}{\sqrt{\rho_i} + \sqrt{\rho_j}}, \quad \text{and} \quad H_i = E - \frac{p_i}{\rho_i}.$$

Therefore, using this density-averaged variables it is possible to rewrite Euler problem in the form of (4.4). Hence, if $-\boldsymbol{f}_k'(\boldsymbol{u}_{ij}) \cdot (\partial_k \varphi_j, \varphi_i)$ has non-positive eigenvalues, then the scheme will be LED for a certain set of *local* characteristic variables. Schemes that satisfy this property are named *local bounds preserving schemes* in the literature

[75]. This reasoning above motivated the definition of the LED *principle* for hyperbolic systems of equations by Kuzmin [59] and coworkers. Adapted from this principle, we define local bounds preserving schemes as follows.

Definition 4.2.4. *The semi-discrete scheme*

$$\sum_j \mathbf{M}_{ij}\partial_t \boldsymbol{u}_j + \sum_{j\neq i} \mathsf{A}_{ij}(\boldsymbol{u}_j - \boldsymbol{u}_i) = \mathbf{0}$$

is said to be local bounds preserving if $\mathbf{M}$ *is diagonal with positive entries (i.e.* $\mathbf{M}_{ij} = m_i\delta_{ij}I_{m\times m}$*),* A_{ij} *has non-positive eigenvalues for every* $j \neq i$*, and* $\sum_j \mathsf{A}_{ij} = \mathbf{0}$*.*

Unfortunately, to the best of our knowledge, satisfying this definition does not ensure positivity of density, internal energy, or non-decreasing entropy. In any case, numerical schemes based on this definition have shown good numerical behavior [59, 62, 70, 75].

Several stabilization strategies have been defined based on the previous ideas. One of the most simple strategies consists on adding a scalar artificial diffusion term proportional to the spectral radius of A_{ij} [59, 73]. Sometimes, this strategy is named Rusanov artificial diffusion, since for linear FEs in one dimension the scheme results in the Rusanov Riemann solver [59, 91]. Without any special treatment, the resulting scheme is only first order accurate. The key to recovering high-order convergence is to modulate the action of the artificial diffusion term, and restrict its action to the vicinity of discontinuities. We base our stabilization term in Rusanov artificial diffusion and a differentiable shock detector recently developed for scalar problems in Chapters 2 and 3.

4.3 Nonlinear stabilization

As previously discussed, the Galerkin FE discretization yields oscillatory solutions in regions around discontinuities. We supplement the original scheme with an artificial diffusion term to stabilize it and mitigate these oscillations. The proposed stabilization term is given by

$$B_h(\boldsymbol{w}_h;\boldsymbol{u}_h,\boldsymbol{v}_h) \doteq \sum_{K_e\in\mathcal{T}_h} \sum_{\substack{i,j\in\mathcal{N}_h(K_e)\\ 1\leq\beta,\gamma\leq m}} \nu^e_{ij}(\boldsymbol{w}_h)\ell(i,j)\, v_i^\beta\cdot\delta_{\beta\gamma}u_j^\gamma, \tag{4.7}$$

for any $\boldsymbol{u}_h \in V_h$ and $\boldsymbol{v}_h \in V_{h0}$. Here, $\ell(i,j) \doteq 2\delta_{ij} - 1$ is a graph Laplacian operator defined in Chapter 2, and $\nu^e_{ij}(\boldsymbol{w}_h)$ is the element-wise artificial diffusion defined as

$$\begin{aligned} \nu^e_{ij}(\boldsymbol{w}_h) &\doteq \max\left(\boldsymbol{\alpha}_i(\boldsymbol{w}_h)\lambda^{\max}_{ij}, \boldsymbol{\alpha}_j(\boldsymbol{w}_h)\lambda^{\max}_{ji}\right), \quad \text{for } j\in\mathcal{N}_h(\Omega_i)\backslash\{i\},\\ \nu^e_{ii}(\boldsymbol{w}_h) &\doteq \sum_{j\in\mathcal{N}_h(\Omega_i)\backslash\{i\}} \nu^e_{ij}(\boldsymbol{w}_h), \end{aligned} \tag{4.8}$$

where $\lambda^{\max}_{ij}$ is the spectral radius of the elemental convection matrix relating nodes $i,j\in\mathcal{N}_h$, i.e. $\rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij})\cdot(\boldsymbol{\nabla}\varphi_j,\varphi_i)_{K_e}\right)$. As previously introduced, this artificial diffusion

term is based on Rusanov scalar diffusion [61]. It is important to mention that the eigenvalues of these matrices can be easily computed as

$$\lambda_{1,..,d} = \mathbf{v}_{ij} \cdot \boldsymbol{c}_{ij}^e, \quad \lambda_{d+1} = \mathbf{v}_{ij} \cdot \boldsymbol{c}_{ij}^e - c\|\boldsymbol{c}_{ij}^e\|, \quad \lambda_{d+2} = \mathbf{v}_{ij} \cdot \boldsymbol{c}_{ij}^e + c\|\boldsymbol{c}_{ij}^e\| \tag{4.9}$$

where

$$\boldsymbol{c}_{ij}^e = (\boldsymbol{\nabla}\varphi_j, \varphi_i)_{K_e}, \quad \text{and} \quad c = \sqrt{(\gamma - 1)\left(H_{ij} - \frac{\|m_{ij}\|^2}{2\rho_{ij}^2}\right)}.$$

We denote by $\boldsymbol{\alpha}_i(\boldsymbol{w}_h)$ the shock detector used for modulating the action of the artificial diffusion term. The idea behind the definition of this detector is minimizing the amount of artificial diffusion introduced while stabilizing any oscillatory behavior. In regions where the local DMP (see Def. 4.2.2) is not satisfied for any chosen set of components, we ensure that Def. 4.2.4 is satisfied. $\boldsymbol{\alpha}_i(\boldsymbol{w}_h)$ must be a positive real number which takes value 1 when $u_h(\boldsymbol{x}_i)$ is an inadmissible value of $\boldsymbol{u}_h$, and smaller than 1 otherwise. To this end, we define

$$\boldsymbol{\alpha}_i(\boldsymbol{u}_h) \doteq \max\{\alpha_i(u_h^\beta)\}_{\beta \in C}, \tag{4.10}$$

where C is the set of components that are used to detect inadmissible values of $\boldsymbol{u}_h$, e.g. density and total energy in the case of Euler equations. For simplicity, we restrict ourselves to the components of $\boldsymbol{u}_h$. However, derived quantities such as the pressure or internal energy can be also used.

In order to introduce the shock detector, let us recall some useful notation from Chapter 2. Let $\boldsymbol{r}_{ij} = \boldsymbol{x}_j - \boldsymbol{x}_i$ be the vector pointing from node $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ with $i, j \in \mathcal{N}_h$ and $\hat{\boldsymbol{r}}_{ij} \doteq \frac{\boldsymbol{r}_{ij}}{|\boldsymbol{r}_{ij}|}$. Recall that the set of points $\boldsymbol{x}_j$ for $j \in \mathcal{N}_h(\Omega_i)\backslash\{i\}$ define the macroelement Ω_i around node $\boldsymbol{x}_i$. Let $\boldsymbol{x}_{ij}^{\text{sym}}$ be the point at the intersection between $\partial\Omega_i$ and the line that passes through $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ that is not $\boldsymbol{x}_j$ (see Fig. 4.1). The set of all $\boldsymbol{x}_{ij}^{\text{sym}}$ for all $j \in \mathcal{N}_h(\Omega_i)\backslash\{i\}$ is represented with $\mathcal{N}_h^{\text{sym}}(\Omega_i)$. We define $\boldsymbol{r}_{ij}^{\text{sym}} \doteq \boldsymbol{x}_{ij}^{\text{sym}} - \boldsymbol{x}_i$. Given $\boldsymbol{x}_{ij}^{\text{sym}}$ in two dimensions, let as call a and b the indices of the vertices such that they define the edge in $\partial\Omega_{\boldsymbol{i}}$ that contains $\boldsymbol{x}_{ij}^{\text{sym}}$. We define $\boldsymbol{u}_j^{\text{sym}}$ as the value of $\boldsymbol{u}_h$ at $\boldsymbol{x}_{ij}^{\text{sym}}$, i.e. $\boldsymbol{u}_h(\boldsymbol{x}_{ij}^{\text{sym}})$.

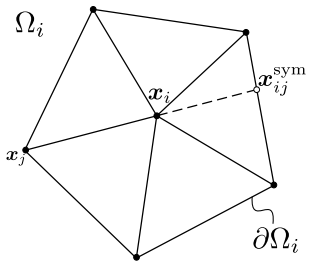


FIGURE 4.1: u^{sym} drawing.

Both $\boldsymbol{u}_{ij}^{\text{sym}}$ and $\boldsymbol{x}_{ij}^{\text{sym}}$ are only required to construct a linearity preserving shock detector. Let us define the jump and the mean of a linear approximation of component β

of the unknown gradient at node $\boldsymbol{x}_i$ in direction $\boldsymbol{r}_{ij}$ as

$$\left[\!\left[\boldsymbol{\nabla} u_h^\beta\right]\!\right]_{ij} \doteq \frac{u_j^\beta - u_i^\beta}{|\boldsymbol{r}_{ij}|} + \frac{u_j^{\mathrm{sym},\beta} - u_i^\beta}{|\boldsymbol{r}_{ij}^{\mathrm{sym}}|},$$
$$\left\{\!\!\left\{|\boldsymbol{\nabla} u_h^\beta \cdot \hat{\boldsymbol{r}}_{ij}|\right\}\!\!\right\}_{ij} \doteq \frac{1}{2}\left(\frac{|u_j^\beta - u_i^\beta|}{|\boldsymbol{r}_{ij}|} + \frac{|u_j^{\mathrm{sym},\beta} - u_i^\beta|}{|\boldsymbol{r}_{ij}^{\mathrm{sym}}|}\right).$$

For each component in C, we use the same shock detector developed in Chapter 2. Let us recall its definition

$$\alpha_i(u_h^\beta) \doteq \begin{cases} \left[\dfrac{\left|\sum_{j\in\mathcal{N}_h(\Omega_i)} \left[\!\left[\boldsymbol{\nabla} u_h^\beta\right]\!\right]_{ij}\right|}{\sum_{j\in\mathcal{N}_h(\Omega_i)} 2\left\{\!\!\left\{\left|\boldsymbol{\nabla} u_h^\beta \cdot \hat{\boldsymbol{r}}_{ij}\right|\right\}\!\!\right\}_{ij}}\right]^q & \text{if} \quad \sum_{j\in\mathcal{N}_h(\Omega_i)} \left\{\!\!\left\{|\boldsymbol{\nabla}\cdot\hat{\boldsymbol{r}}_{ij}|\right\}\!\!\right\}_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases}. \tag{4.11}$$

From Lm. 2.3.1 we know that (4.11) is valued between 0 and 1, and it is only equal to one if $u_h^\beta(\boldsymbol{x}_i)$ is a local discrete extremum (in a space–time sense as in Def. 4.2.1). Since the linear approximations of the unknown gradients are exact for $u_h^\beta \in \mathcal{P}_1$, the shock detector vanishes when the solution is linear. Thus, it is also linearly preserving for every component in C. This result follows directly from Th. 2.4.5.

The final stabilized problem in matrix form reads as follows. Find $\boldsymbol{u}_h \in \boldsymbol{V}_h$ such that $\boldsymbol{u}_h = \overline{\boldsymbol{u}}_h$ on $\partial\Omega$, $\boldsymbol{u}_h = \boldsymbol{u}_{0h}$ at $t = 0$, and

$$\overline{\mathbf{M}}(\boldsymbol{u}_h^{n+1})\delta_t \mathbf{U}^{n+1} + \overline{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{n+1})\mathbf{U}^{n+1} = \mathbf{G} \tag{4.12}$$

for $n = 1, ..., n^{ts}$, where

$$\overline{\mathbf{M}}_{ij}(\boldsymbol{u}_h^{n+1}) \doteq [1 - \max(\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j)]\,(\varphi_j, \varphi_i) I_{m\times m} + \max(\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j)\,(\delta_{ij}, \varphi_i) I_{m\times m},$$

$\overline{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{n+1}) \doteq \mathbf{K}_{ij} + \mathbf{B}_{ij}$, and $\mathbf{B}_{ij}(\boldsymbol{u}_h) = B_h(\boldsymbol{u}_h; \varphi_j, \varphi_i)$, for $i, j \in \mathcal{N}_h$.

Lemma 4.3.1 (Local bounds preservation)**.** *Consider $\boldsymbol{u}_h \in \boldsymbol{V}_h$ with component β in the set of tracked variables C. The stabilized problem* (4.12) *is local bounds preserving as defined in Def. 4.2.4 at any region where u_h^β has extreme values.*

Proof. If component $\beta \in C$ of $\boldsymbol{u}_h$ has an extremum at $\boldsymbol{x}_i$, then from Lm. 2.3.1 we know that $\alpha_i(u_h^\beta) = 1$. Moreover, from (4.10) is easy to see that $\boldsymbol{\alpha}_i(\boldsymbol{u}_h) = 1$. In this case, $\overline{\mathbf{M}}_{ij}(u_h) = (\delta_{ij}, \varphi_i) I_{m\times m}$. Hence, $\overline{\mathbf{M}}_{ij}(u_h) = 0$ for $j \neq i$ and $\overline{\mathbf{M}}_{ii}(u_h) = m_i$. Therefore, we can rewrite the system as follows

$$m_i \partial_t \boldsymbol{u}_i + \sum_{j\in\mathcal{N}_h(\Omega_i)\setminus\{i\}} \overline{\mathbf{K}}_{ij}(\boldsymbol{u}_{ij})(\boldsymbol{u}_j - \boldsymbol{u}_i) =$$
$$m_i \partial_t \boldsymbol{u}_i + \sum_{j\in\mathcal{N}_h(\Omega_i)\setminus\{i\}} \sum_{K_e\in\mathcal{T}_h} \left(\boldsymbol{f}'(\boldsymbol{u}_{ij}) \cdot (\boldsymbol{\nabla}\varphi_j, \varphi_i)_{K_e} - \nu_{ij}^e I_{m\times m}\right)(\boldsymbol{u}_j - \boldsymbol{u}_i) = \mathbf{0}.$$

We need to prove that the eigenvalues of $\overline{\mathbf{K}}_{ij}(\boldsymbol{u}_{ij})$ are non-positive. To this end, let us show the following inequality holds

$$\sum_{K_e \in \mathcal{T}_h} \rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij}) \cdot (\boldsymbol{\nabla}\varphi_j, \varphi_i)_{K_e}\right) \geq \rho(\boldsymbol{f}'(\boldsymbol{u}_{ij}) \cdot (\boldsymbol{\nabla}\varphi_j, \varphi_i)).$$

From (4.9), it is easy to check that $\rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij}) \cdot (\boldsymbol{\nabla}\varphi_j, \varphi_i)_{K_e}\right) = |\mathbf{v}_{ij} \cdot \boldsymbol{c}_{ij}^e| + c_{ij}\|\boldsymbol{c}_{ij}^e\|$. Since $\boldsymbol{c}_{ij} = \sum_{K_e \in \mathcal{T}_h} \boldsymbol{c}_{ij}^e$, we have that

$$\sum_{K_e \in \mathcal{T}_h} \left|\mathbf{v}_{ij} \cdot \boldsymbol{c}_{ij}^e\right| \geq \left|\mathbf{v}_{ij} \cdot \boldsymbol{c}_{ij}\right|, \quad \text{and} \quad \sum_{K_e \in \mathcal{T}_h} c_{ij}\|\boldsymbol{c}_{ij}^e\| \geq c_{ij}\|\boldsymbol{c}_{ij}\|.$$

Hence, $\sum_e \rho(\mathbf{K}_{ij}^e(\boldsymbol{u}_{ij})) \geq \rho(\mathbf{K}_{ij}(\boldsymbol{u}_{ij}))$. Moreover, by definition (see (4.8)),

$$\nu_{ij}^e \geq \rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij}) \cdot (\boldsymbol{\nabla}\varphi_j, \varphi_i)_{K_e}\right) \quad \text{for} \quad j \neq i.$$

Furthermore, from (4.7), is easy to see that $\rho(\mathbf{B}_{ij}^e(\boldsymbol{u}_{ij})) \geq \rho(\mathbf{K}_{ij}^e(\boldsymbol{u}_{ij}))$. Therefore, $\rho(\mathbf{B}_{ij}(\boldsymbol{u}_{ij})) \geq \rho(\mathbf{K}_{ij}(\boldsymbol{u}_{ij}))$. Finally, since $\overline{\mathbf{K}}_{ij} = \mathbf{K}_{ij} + \mathbf{B}_{ij}$ and $\mathbf{B}_{ij} = \sum_e \mathbf{B}_{ij}^e = \sum_e -\nu_{ij}^e I_{m\times m}$ for all $j \neq i$, then the maximum eigenvalue of $\overline{\mathbf{K}}_{ij}(\boldsymbol{u}_{ij})$ is non-positive, which completes the proof. □

4.3.1 Differentiability

In the case of steady, or implicit time integration, differentiability plays a role in the convergence behavior of the nonlinear solver. This is especially important if one wants to use Newton's method. In the case of scalar problems it has been shown in the previous chapters and [5] that convergence is greatly improved after few modifications to make a scheme twice-differentiable. In this section, we introduce a set of regularizations applied to all non-differentiable functions present in the stabilized scheme introduced above. In order to regularize these functions, we follow a similar strategy as in Chapter 2. Absolute values are substituted by

$$|x|_{1,\varepsilon_h} = \sqrt{x^2 + \varepsilon_h}, \qquad |x|_{2,\varepsilon_h} = \frac{x^2}{\sqrt{x^2 + \varepsilon_h}}.$$

Note that $|x|_{2,\varepsilon_h} \leq |x| \leq |x|_{1,\varepsilon_h}$. Next, we also use a smooth maximum function, $\max_{\sigma_h}(\cdot)$, as

$$\max_{\sigma_h}(x, y) \doteq \frac{|x - y|_{1,\sigma_h}}{2} + \frac{x + y}{2} \geq \max(x, y). \tag{4.13}$$

In addition, we need a smooth function to limit the value of any given quantity to one. To this end, we use

$$Z(x) \doteq \begin{cases} 2x^4 - 5x^3 + 3x^2 + x, & x < 1, \\ 1, & x \geq 1. \end{cases}$$

The set of twice-differentiable functions defined above allow us to redefine the stabilization term introduced in Sect. 4.3. In particular, we define

$$\tilde{B}_h(\boldsymbol{w}_h;\boldsymbol{u}_h,\boldsymbol{v}_h) \doteq \sum_{K_e\in\mathcal{T}_h}\sum_{\substack{i,j\in\mathcal{N}_h(K_e)\\ 1\le\beta,\gamma\le m}} \tilde{\nu}^e_{ij}(\boldsymbol{w}_h)\ell(i,j)\, v_i^\beta\cdot\delta_{\beta\gamma}u_j^\gamma,$$

where

$$\begin{aligned}\tilde{\nu}^e_{ij}(\boldsymbol{w}_h) &\doteq \max\nolimits_{\sigma_h}\left(\boldsymbol{\alpha}_{\varepsilon_h,i}(\boldsymbol{w}_h)\lambda^{\max}_{ij},\boldsymbol{\alpha}_{\varepsilon_h,j}(\boldsymbol{w}_h)\lambda^{\max}_{ji}\right), \quad \text{for } j\in\mathcal{N}_h(\Omega_i)\backslash\{i\},\\ \tilde{\nu}^e_{ii}(\boldsymbol{w}_h) &\doteq \sum_{j\in\mathcal{N}_h(\Omega_i)\backslash\{i\}}\tilde{\nu}^e_{ij}(\boldsymbol{w}_h).\end{aligned} \tag{4.14}$$

Let us note that $\lambda^{\max}_{ij}$ needs to be regularized as $\lambda^{\max}_{ij} = \left|\mathbf{v}_{ij}\boldsymbol{c}^e_{ij}\right|_{1,\varepsilon_h} + c\|\boldsymbol{c}^e_{ij}\|$. The shock detector is also redefined to use the regularized version of the shock detector, which reads

$$\boldsymbol{\alpha}_{\varepsilon_h,i}(\boldsymbol{u}_h) \doteq \max\nolimits_{\sigma_h}\{\alpha_{\varepsilon_h,i}(u_h^\beta)\}_{\beta\in C}.$$

In the case of the component shock detector we recall the definition in Chapter 2

$$\alpha_{\varepsilon_h,i}(u_h^\beta) \doteq \left[Z\left(\frac{\left|\sum_{j\in\mathcal{N}_h(\Omega_i)}\left[\!\left[\boldsymbol{\nabla}u_h^\beta\right]\!\right]_{ij}\right|_{1,\varepsilon_h}+\zeta_h}{\sum_{j\in\mathcal{N}_h(\Omega_i)}2\left\{\!\!\left\{\left|\boldsymbol{\nabla}u_h^\beta\cdot\hat{\boldsymbol{r}}_{ij}\right|_{2,\varepsilon_h}\right\}\!\!\right\}_{ij}+\zeta_h}\right)\right]^q, \tag{4.15}$$

where ζ_h is a small value for preventing division by zero. Finally, the twice-differentiable stabilized scheme reads:
Find $\boldsymbol{u}_h\in\boldsymbol{V}_h$ such that $\boldsymbol{u}_h=\overline{\boldsymbol{u}}_h$ on $\partial\Omega$, $\boldsymbol{u}_h=\boldsymbol{u}_{0h}$ at $t=0$, and

$$\tilde{\mathbf{M}}(\boldsymbol{u}_h^{n+1})\delta_t\mathbf{U}^{n+1}+\tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{n+1})\mathbf{U}^{n+1}=\mathbf{G} \quad \text{for } n=1,...,n^{ts}, \tag{4.16}$$

where

$$\begin{aligned}\tilde{\mathbf{M}}_{ij}(\boldsymbol{u}_h^{n+1}) &\doteq \left[1-\max\nolimits_{\sigma_h}\left(\boldsymbol{\alpha}_{\varepsilon_h,i},\boldsymbol{\alpha}_{\varepsilon_h,j}\right)\right](\varphi_j,\varphi_i)I_{m\times m}\\ &\quad+\max\nolimits_{\sigma_h}\left(\boldsymbol{\alpha}_{\varepsilon_h,i},\boldsymbol{\alpha}_{\varepsilon_h,j}\right)(\delta_{ij},\varphi_i)I_{m\times m},\\ \tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{n+1}) &\doteq \mathbf{K}_{ij}(\boldsymbol{u}_h^{n+1})+\tilde{\mathbf{B}}_{ij}(\boldsymbol{u}_h^{n+1}),\end{aligned}$$

and $\tilde{\mathbf{B}}_{ij}(\boldsymbol{u}_h)=\tilde{B}_h(\boldsymbol{u}_h;\varphi_j,\varphi_i)$, for $i,j\in\mathcal{N}_h$.

Corollary 4.3.2. *The differentiable scheme in Eq.* (4.14) *is local bounds preserving, as defined in Def. 4.2.4, at any region where u_h^β has extreme values for every β in C.*

Proof. For an extreme value of u_h^β, since $|x|_{2,\varepsilon_h}\le|x|\le|x|_{1,\varepsilon_h}$ the quotient of (4.15) is larger than one. Hence, by definition of $Z(x)$, $\alpha_{\varepsilon_h,i}$ is equal to 1. At this point, it is easy to check that $\tilde{\nu}^e_{ij}\ge\nu^e_{ij}$ in virtue of the definition of $\max_{\sigma_h}$. Therefore, $\rho(\tilde{\mathbf{B}}^e_{ij}(\boldsymbol{u}_h))\ge\rho(\mathbf{B}^e_{ij}(\boldsymbol{u}_h))$, completing the proof. □

Moreover, it is important to mention that the differentiable shock detector is weakly linearly-preserving as ζ_h tends to zero. This result follows directly from Sect. 2.7. In order to obtain a differentiable operator, we have added a set of regularizations that rely on different parameters, e.g., σ_h, ε_h, ζ_h. Giving a proper scaling of these parameters is essential to recover theoretic convergence rates. In particular, we use the following relations

$$\sigma_h = \sigma|\lambda^{\max}|^2 L^{2(d-3)} h^4, \quad \varepsilon_h = \varepsilon L^{-4} h^2, \quad \zeta_h = L^{-1}\zeta, \tag{4.17}$$

where d is the spatial dimension of the problem, L is a characteristic length, and σ, ε, and ζ are of the order of the unknown.

4.4 Nonlinear solver

In this section, we describe the method used for solving the nonlinear system of equations arising from the scheme introduced above. In particular, we use a hybrid Picard–Newton approach in order to increase the robustness of the nonlinear solver. Moreover, for the differentiable version we also use a continuation method to improve the nonlinear convergence.

We represent the residual of the equation (4.16) at the k-th iteration by $\mathbf{R}(\boldsymbol{u}_h^{k,n+1})$, i.e.,

$$\mathbf{R}(\boldsymbol{u}_h^{k,n+1}) \doteq \tilde{\mathbf{M}}(\boldsymbol{u}_h^{k,n+1})\delta_t \mathbf{U}^{k,n+1} + \tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{k,n+1})\mathbf{U}^{k,n+1} - \mathbf{G}. \tag{4.18}$$

Hence, the Jacobian is defined as

$$\begin{aligned}\mathbf{J}(\boldsymbol{u}_h^{k,n+1}) &\doteq \frac{\partial \mathbf{R}(\boldsymbol{u}_h^{k,n+1})}{\partial \mathbf{U}^{k,n+1}} \\ &= \tilde{\mathbf{M}}(\boldsymbol{u}_h^{k,n+1}) + \tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{k,n+1}) + \frac{\partial \tilde{\mathbf{M}}(\boldsymbol{u}_h^{k,n+1})}{\partial \mathbf{U}^{k,n+1}}\delta_t \mathbf{U}^{k,n+1} + \frac{\partial \tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{k,n+1})}{\partial \mathbf{U}^{k,n+1}}\mathbf{U}^{k,n+1}.\end{aligned} \tag{4.19}$$

Therefore, Newton method consist on solving $\mathbf{J}(\boldsymbol{u}_h^{k,n+1})\Delta\mathbf{U}^{k+1,n+1} = -\mathbf{R}(\boldsymbol{u}_h^{k,n+1})$. However, it is well known that Newton method can diverge if the initial guess of the solution $\boldsymbol{u}_h^{0,n+1}$ is not close enough to the solution. In order to improve the robustness, we introduce the following modifications. We use a line–search method to update the solution at every time step. Thus, the new approximation is computed as $\mathbf{U}^{k+1,n+1} = \mathbf{U}^{k,n+1} + \lambda\Delta\mathbf{U}^{k+1,n+1}$, where λ is computed (approximately) such that it minimizes $\|\mathbf{R}(\boldsymbol{u}_h^{k+1,n+1})\|$.

As introduced at the beginning of the section, we also use a hybrid approach combining Newton method with Picard linearization. Picard nonlinear iterator can be obtained removing the last two terms of (4.19), i.e.,

$$\left(\tilde{\mathbf{M}}(\boldsymbol{u}_h^{k,n+1}) + \tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{k,n+1})\right)\Delta\mathbf{U}^{k+1,n+1} = -\mathbf{R}(\boldsymbol{u}_h^{k,n+1}). \tag{4.20}$$

Clearly, it is equivalent to

$$\left(\tilde{\mathbf{M}}(\boldsymbol{u}_h^{k,n+1}) + \tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{k,n+1})\right)\mathbf{U}^{k+1,n+1} = \tilde{\mathbf{M}}(\boldsymbol{u}_h^{k,n+1})\mathbf{U}^n + \mathbf{G}.$$

Moreover, we modify the definition of left hand side terms in (4.20) to enhance the robustness of the method. In particular, we use $\boldsymbol{\alpha}_i = 1$ for computing these terms while we use the value obtained from (4.10) for the residual. Using this strategy the solution remains unaltered, but the obtained approximations $\boldsymbol{u}_h^{k,n+1}$ for intermediate values of k are more diffusive. Even though this modification slows the nonlinear convergence, it is essential at the initial iterations. Otherwise, the robustness of the method might be jeopardized.

The resulting iterative nonlinear solver consists in the following. We iterate using Picard method in (4.20), with the modification described above, until the L^2 norm of the residual is smaller than a given tolerance. In this chapter, we use tolerances close to 10^{-2}. Afterwards, Newton method with the exact Jacobian in (4.19) is used until the desired nonlinear convergence criteria is satisfied.

For the differentiable stabilization, we also equip the above scheme with a continuation method on the regularization parameters. In order to accelerate the convergence of the method, we use high values for the parameters during the first iterations. This results in a more diffusive solution, but nonlinear convergence is accelerated. As the nonlinear approximation is closer to the solution, we diminish the value of the parameters to avoid introducing excessive artificial diffusion to the system. This process is preformed gradually as a function of the residual in (4.18). In particular, we use the following relation

$$\varepsilon^k = \tilde{\varepsilon}\,\frac{\|\mathbf{R}(\boldsymbol{u}_h^{k,n+1})\|}{\|\mathbf{R}(\boldsymbol{u}_h^{0,n+1})\|},$$

where ε^k is the effective parameter used in relations 4.17, and $\tilde{\varepsilon}$ is parameter defined by the user. We summarize the nonlinear solver introduced above in Alg. 3.

4.5 Numerical experiments

In this section, we perform several numerical experiments to assess the numerical scheme introduced in the previous sections. First, we perform a convergence analysis to assess its implementation. Then, we use a steady benchmark test to analyze the effectiveness of the regularization parameters. We also analyze their effectiveness in the case of a transient problem. Finally, we solve a slightly more challenging steady benchmark test.

In all experiments below we assume that the ideal gas state equation applies, and we use an adiabatic index of $\gamma = 1.4$. From previous experience [4, 5, 16, 18], the effects of parameters σ and ε to the nonlinear convergence and numerical error are analogous. Hence, we consider $\varepsilon = 10^{-2}\sigma$. In addition, for all the tests below, the density is

Algorithm 3: Hybrid Picard–Newton scheme with the continuation method.

Input: $\mathbf{U}^{0,n+1}$, tol_1, tol_2, ε, Continuation
Output: $\mathbf{U}^{k,n+1}$, k
$k = 1$, $\varepsilon^1 = \varepsilon$
while $\|\mathbf{R}(\mathbf{U}^{k,n+1})\|/\|\mathbf{R}(\mathbf{U}^{0,n+1})\| \geq \mathrm{tol}_1$ **do**
  Compute $\boldsymbol{\alpha}_i(\mathbf{U}^{k,n+1})$ using (4.10)
  Compute $\Delta\mathbf{U}^{k+1,n+1}$ using (4.20)
  Minimize $\|\mathbf{R}(\mathbf{U}^{k+1,n+1})\|$, where $\mathbf{U}^{k+1,n+1} = \lambda\Delta\mathbf{U}^{k+1,n+1} + \mathbf{U}^{k,n+1}$, with respect to λ
  Set $\mathbf{U}^{k+1,n+1} = \lambda\Delta\mathbf{U}^{k+1,n+1} + \mathbf{U}^{k,n+1}$
  if Continuation **then**
    Set $\varepsilon^k = \tilde{\varepsilon}\frac{\|\mathbf{R}(\mathbf{U}^{k+1,n+1})\|}{\|\mathbf{R}(\mathbf{U}^{0,n+1})\|}$
  else
    Set $\varepsilon^k = \varepsilon$
  Set $\sigma^k = 10^2\,\varepsilon^k$
  Update $k = k + 1$
while $\|\mathbf{R}(\mathbf{U}^{k,n+1})\|/\|\mathbf{R}(\mathbf{U}^{0,n+1})\| \geq \mathrm{tol}_2$ **do**
  Compute $\boldsymbol{\alpha}_i(\mathbf{U}^{k,n+1})$ using (4.10)
  Solve $\mathbf{J}(\mathbf{U}^{k,n+1})\Delta\mathbf{U}^{k+1,n+1} = -\mathbf{R}(\mathbf{U}^{k,n+1})$ with $\mathbf{J}$ in (4.19)
  Minimize $\|\mathbf{R}(\mathbf{U}^{k+1,n+1})\|$, where $\mathbf{U}^{k+1,n+1} = \lambda\Delta\mathbf{U}^{k,n+1} + \mathbf{U}^{k,n+1}$, with respect to λ
  Set $\mathbf{U}^{k+1,n+1} = \lambda\Delta\mathbf{U}^{k,n+1} + \mathbf{U}^{k,n+1}$
  if Continuation **then**
    Set $\varepsilon^k = \tilde{\varepsilon}\frac{\|\mathbf{R}(\mathbf{U}^{k+1,n+1})\|}{\|\mathbf{R}(\mathbf{U}^{0,n+1})\|}$
  else
    Set $\varepsilon^k = \varepsilon$
  Set $\sigma^k = 10^2\,\varepsilon^k$
  Update $k = k + 1$

discontinuous at all shocks. Therefore, we use $C = \{1\}$ in (4.10), i.e. the shock detector is based on the density behavior.

4.5.1 Convergence test

We use two different problems to assess the convergence rate of the scheme. One has a smooth solution, whereas in the other there is a shock. The smooth problem is simply the translation of a sinusoidal perturbation in the density, with constant pressure and velocity. In particular, the solution for $r = \sqrt{(0.5 + t - x)^2 + (0.5 - y)^2} < 0.5$ is

$$\boldsymbol{u} = \begin{bmatrix} \rho \\ v_x \\ v_y \\ p \end{bmatrix} = \begin{bmatrix} 1 + 0.9999\cos(2\pi r) \\ 1 \\ 0 \\ 1 \end{bmatrix},$$

and $\boldsymbol{u} = [0.0001, 1, 0, 1]^t$ otherwise.

The non-smooth problem is the well known compression corner test [3, 61], also known as oblique shock test [84, 88]. This benchmark consists in a supersonic flow impinging to a wall at an angle. We use a $[0,1]^2$ domain with a $M = 2$ flow at $10°$ with respect to the wall. This leads to two flow regions separated by an oblique shock at $29.3°$, see the scheme in Fig. 4.2.

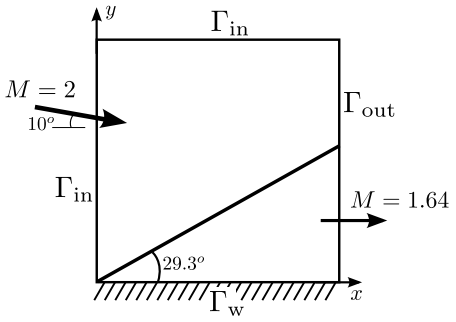
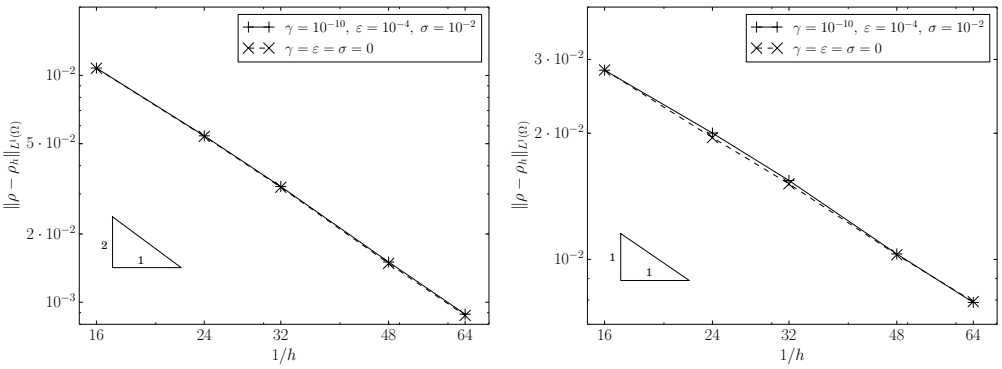


FIGURE 4.2: Compression corner scheme.

For both tests, we compare the convergence rates for the differentiable and the non-differentiable schemes. q is set to 10 and the regularization parameters are $\gamma = 10^{-10}$, $\varepsilon = 10^{-4}$, and $\sigma = 10^{-2}$ in the differentiable version.

In Fig. 4.3, the L^1 error is depicted for different mesh sizes, and in Tab. 4.1 we collect the measured convergence rates. It can be observed that for a smooth problem both settings recover second order convergence, whereas for non-smooth problems the expected first order convergence rates are obtained. For this particular choice of regularization parameters, we observe that the errors are slightly higher. However, the convergence rates are not affected by the regularization described in Sect. 4.3.1.



(A) Sinusoidal translation test. (B) Compression corner test.

FIGURE 4.3: Density convergence for successive mesh refinements.

4.5.2 Reflected Shock

In this test, we compare the nonlinear convergence behavior of the method for different regularization parameters. This benchmark consists in two flow streams colliding at different angles. The domain has dimensions $[0.0, 1.0] \times [0.0, 4.1]$ and a solid wall at its

TABLE 4.1: Experimental convergence rates for both problems.

Test	L_1 error
Sinusoidal translation (differentiable)	1.8099
Sinusoidal translation (non-differentiable)	1.8190
Compression corner (differentiable)	0.9278
Compression corner (non-differentiable)	0.9207

lower boundary. This configuration leads to a steady shock separating both flow regimes, which in turn, is reflected at the wall producing a third different flow state behind it. A sketch of this benchmark test is given in Fig. 4.4. The flow states at each region have been collected in Tab. 4.2.

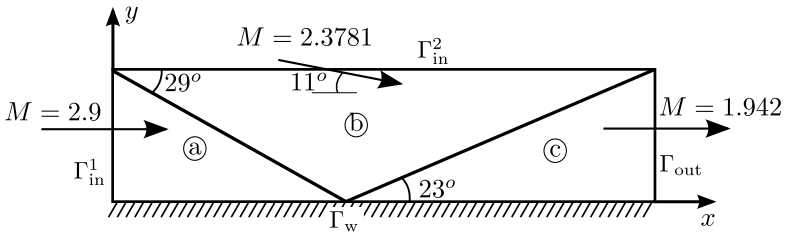


FIGURE 4.4: Reflected shock scheme.

TABLE 4.2: Reflected shock solution values at every region.

Region	Density [Kg m^{-3}]	Velocity [m s^{-1}]	Total energy [J]
ⓐ	1.0	(2.9, 0.0)	5.99075
ⓑ	1.7	(2.62, -0.506)	5.8046
ⓒ	2.687	(2.401, 0.0)	5.6122

We use a 60×20 structured $\mathcal{Q}_1$ mesh. The problem is solved directly to steady state using the hybrid method and the continuation scheme described in Sect. 4.4. The tolerance used for switching from Picard to Newton linearization is 10^{-2}. We compare the convergence behavior for $q = \{1, 2, 5, 10\}$. For the differentiable stabilization we use the following values for $\tilde{\varepsilon} = \{10^{-4}, 10^{-2}, 1\}$. We consider $\varepsilon^k = \sigma^k\, 10^{-2}$. The value of γ is 10^{-10}.

In Figs. 4.5–4.8 for every nonlinear iteration we depict (from left to right) the relative residual, the relative Galerkin residual, and the relative solution variation between iterations. The Galerkin residual is simply the residual in (4.18) minus the stabilization terms, i.e.,

$$\mathbf{R}^*(\boldsymbol{u}_h^k) \doteq \mathbf{K}_{ij}(\boldsymbol{u}_h^k)\mathbf{U}^k - \mathbf{G}.$$

We depict this value relative to the Galerkin residual of the non-differentiable scheme. This value gives a sense of how close is the computed approximation to the solution of the original problem. However, since it omits the stabilization terms present in the system solved, it will stagnate at some point.

In general, we can observe in Figs. 4.5–4.8 that as q is increased the scheme needs more iterations to converge. In addition, we observe a 15% to 35% reduction in the number of iterations when the differentiable scheme is used. Another interesting observation is about the behavior of the Galerkin residual during the first iterations. At this initial stage, the differentiable scheme is able to provide solutions closer to the solution of the original problem. However, as expected, both schemes stagnate after a number of iterations. Figs. 4.7–4.8 also show an improvement of the residual convergence rate once the complete Jacobian is used, i.e. after $\|\mathbf{R}(\boldsymbol{u}_h^{k,n+1})\|/\|\mathbf{R}(\boldsymbol{u}_h^{0,n+1})\| < 10^{-2}$.

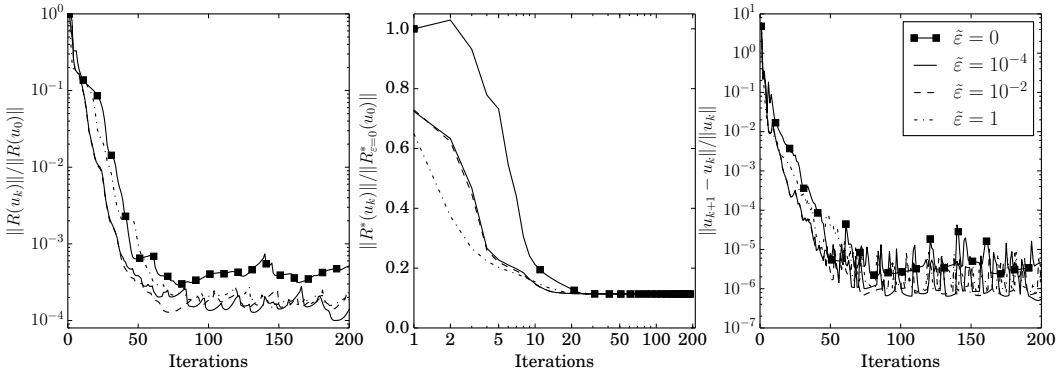


FIGURE 4.5: Reflected shock convergence history for $q = 1$.

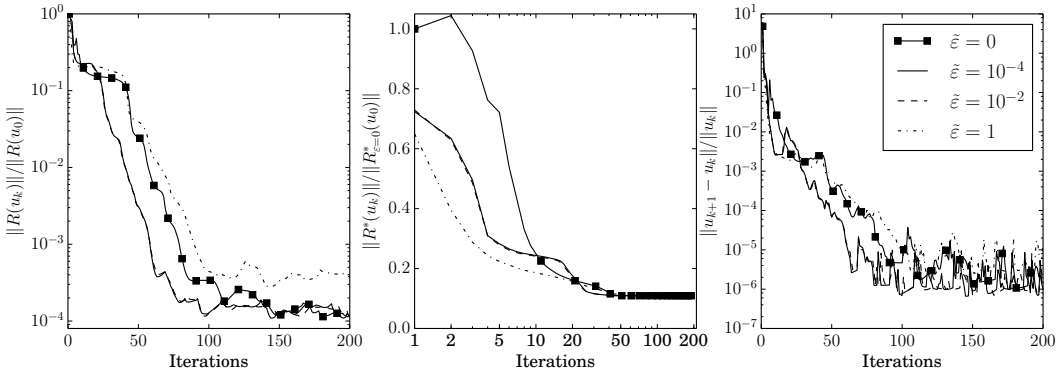


FIGURE 4.6: Reflected shock convergence history for $q = 2$.

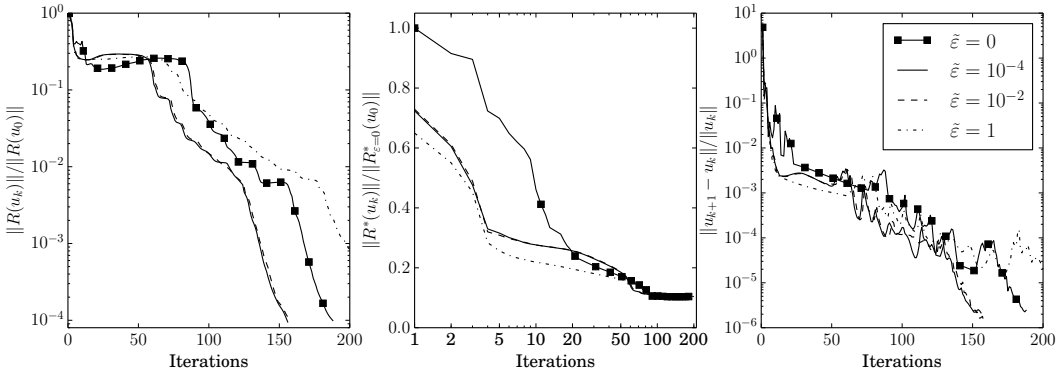


FIGURE 4.8: Reflected shock convergence history for $q = 10$.

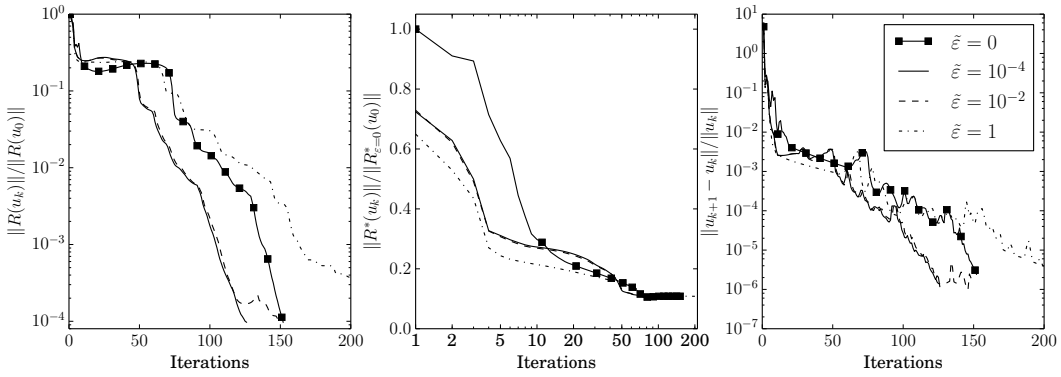
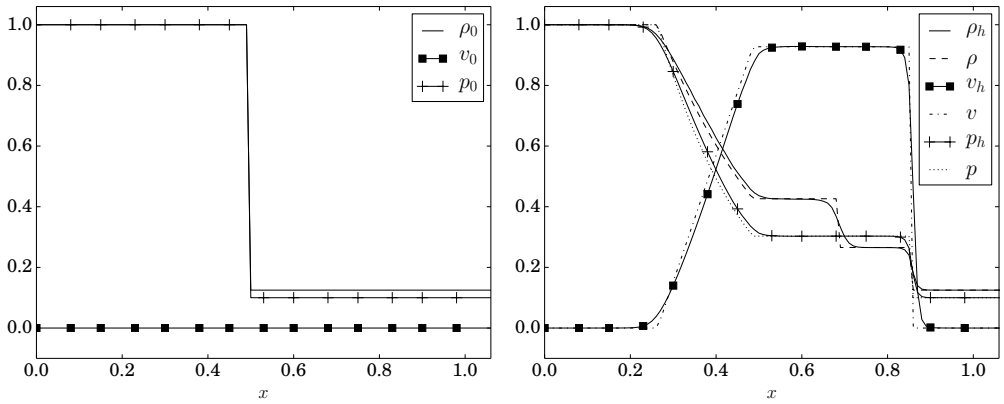


FIGURE 4.7: Reflected shock convergence history for $q = 5$.

4.5.3 Sod's Shock Tube

In this section, we evaluate the effect of the differentiability in the case of a transient problem. To this end, we solve the well known Sod's shock tube test. This is a one dimensional problem that assesses the evolution of a fluid initially at rest with a discontinuity in density and pressure. The discontinuity is initially placed at $x = 0.5$. Even though it is a 1D test, we consider a narrow 2D strip of dimensions $[0, 1] \times [0, 0.01]$ and we let the problem evolve until $t = 0.2$. We use a $\mathcal{Q}_1$ FE mesh of size $\Delta x = 0.01$ and a time step length of $\Delta t = 0.001$. Initial conditions at the left of the discontinuity are $\boldsymbol{u}_0 = (1, 0, 0, 2.5)$ and at the right $\boldsymbol{u}_0 = (0.125, 0, 0, 0.25)$. See the initial condition depicted in Fig. 4.9(a).

In this case, the hybrid nonlinear solver described in Sect. 4.4 is used directly without the continuation scheme. The tolerance used for switching from the Picard to Newton linearization is $5 \cdot 10^{-3}$. We set the nonlinear convergence criteria in terms of the relative residual, namely $\frac{\|\mathbf{R}(\boldsymbol{u}_h^{k,n+1})\|}{\|\mathbf{R}(\boldsymbol{u}_h^{0,n+1})\|} < 10^{-6}$. We use $\gamma = 10^{-10}$, $\varepsilon = \{10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}\}$, and $\varepsilon = \sigma\, 10^{-2}$ for the differentiable stabilization. We also use different values of q for this comparison, namely $q = \{1, 2, 4, 6, 8, 10, 12\}$.

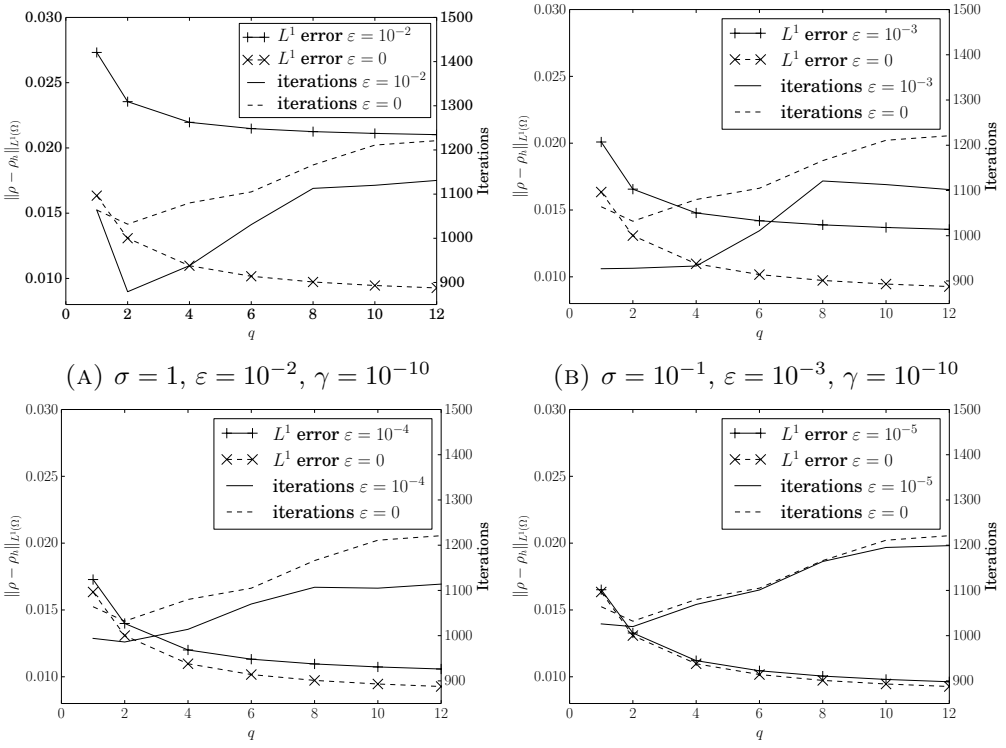


(A) Initial condition. (B) Solution a $t = 0.2$.

FIGURE 4.9: Sod shock initial condition and solution for the differentiable scheme using parameters $q = 10$, $\sigma = 10^{-3}$, $\varepsilon = 10^{-5}$, and $\gamma = 10^{-10}$.

Fig. 4.9(b) shows a comparison at $t = 0.2$ of the exact solution from ExactPack [85] against the obtained solution for $q = 10$, $\sigma = 10^{-3}$, $\varepsilon = 10^{-5}$, and $\gamma = 10^{-10}$. In this case, we observe a good agreement of the obtained solution despite the rather coarse mesh being used.

In Fig. 4.10, for different regularization values, we depict the total number of non-linear iterations required to reach $t = 0.2$, and the density error L^1 norm, as a function of the value of q. For each chart, we compare the results for the differentiable and non-differentiable stabilization. Analyzing these figures, several general observations can be made. One recovers the behavior of the non-differentiable scheme as the parameters used in the differentiable scheme become smaller. Using large values for the regularization parameters improves the computational cost required at the expense of higher numerical errors. It can also be seen that for transient problems the benefits of differentiability are not as evident as for problems solved directly to steady state.



(A) $\sigma = 1$, $\varepsilon = 10^{-2}$, $\gamma = 10^{-10}$

(B) $\sigma = 10^{-1}$, $\varepsilon = 10^{-3}$, $\gamma = 10^{-10}$

(C) $\sigma = 10^{-2}$, $\varepsilon = 10^{-4}$, $\gamma = 10^{-10}$

(D) $\sigma = 10^{-3}$, $\varepsilon = 10^{-5}$, $\gamma = 10^{-10}$

FIGURE 4.10: Comparison of L^1 error and computational cost (total number of iterations) for different regularization parameters choices at the Sod's shock test.

Another interesting observation can be made when moderate values for the parameters are used. Namely, the differentiable scheme is able to yield results with a similar accuracy while requiring a lower computational cost. For example, the error in Fig. 4.10(b) of the differentiable scheme for $q = 2$ is similar to the one obtained for $q = 1$ and the non-differentiable scheme. However, the computational cost is higher for the non-differentiable scheme. The same can be seen for $q = 4$, or for moderate values of

q in Fig. 4.10(c). Therefore, one can come to the conclusion that in order to achieve a given accuracy it is preferable to use the differentiable scheme with a slightly larger value of q rather than the non-differentiable scheme and a low value for q.

4.5.4 Scramjet

Finally, we solve a problem with a supersonic flow that develops a complex shock pattern. This test consists of a $M = 3$ channel that narrows along the streamline and has two internal obstacles. In particular, Fig. 4.11 is an illustration of the domain and Tab. 4.3 lists the coordinates of the points defining the domain. The problem is solved directly to steady state, and two different meshes have been used. The coarsest mesh used has 18476 $\mathcal{Q}_1$ elements and the finest mesh has 63695 $\mathcal{Q}_1$ elements.

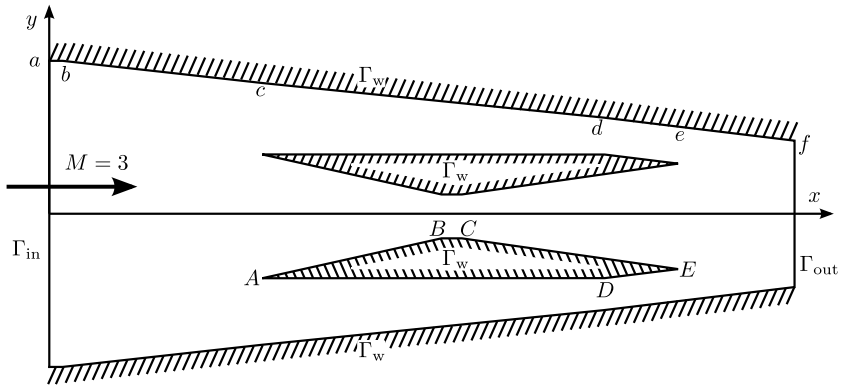


FIGURE 4.11: Scramjet test scheme.

TABLE 4.3: Domain coordinates for the scramjet test.

Wall	a	b	c	d	e	f
x_i	0.0	0.4	4.9	12.6	14.25	16.9
y_i	3.5	3.5	2.9	2.12	1.92	1.7
Interior obstacle		A	B	C	D	E
x_i		4.9	8.9	9.4	12.6	14.25
y_i		-1.4	-0.5	-0.5	-1.4	-1.2

In order to solve this problem, the hybrid nonlinear solver described in Sect. 4.4 is used with the help of the continuation scheme. The tolerance for switching from the Picard to Newton linearization is set to $5 \cdot 10^{-2}$. The nonlinear convergence criteria for this benchmark is $\frac{\|\Delta(\boldsymbol{u}_h^{k+1})\|}{\|\boldsymbol{u}_h^k\|} < 10^{-6}$. We also set a maximum number of iterations of 500. In this test, we use $q = \{2, 5\}$, $\gamma = 10^{-10}$, $\tilde{\varepsilon} = \{1, 10^{-2}, 10^{-4}\}$, and $\varepsilon^k = \sigma^k\, 10^{-2}$. Even though $\sigma = 10^2$ might seem a high value, we recall that it is used in the context of a continuation method. Therefore, the effective value of σ^k is lower than 1 for the converged solution. Moreover, the actual value used in (4.13) is computed using the relations in (4.17).

Figs. 4.12–4.13 show, respectively, the Mach and density contours for the fine mesh, $q = 5$, $\tilde{\varepsilon} = 1$, $\tilde{\varepsilon} = 10^2$, and $\gamma = 10^{-10}$. The nonlinear convergence history for this

configuration is depicted in Figs. 4.17(c)–4.17(d). The obtained values for the Mach number and the density are comparable to those in [61, 75]. The shocks are well resolved. Even when using $q = 2$ the shocks are properly resolved and only slightly more smeared than for $q = 5$, see Fig. 4.14. If instead, the coarse mesh is used (see Fig. 4.15), the solution is more dissipative. However, the scheme is able to capture most of the features present in the solution.

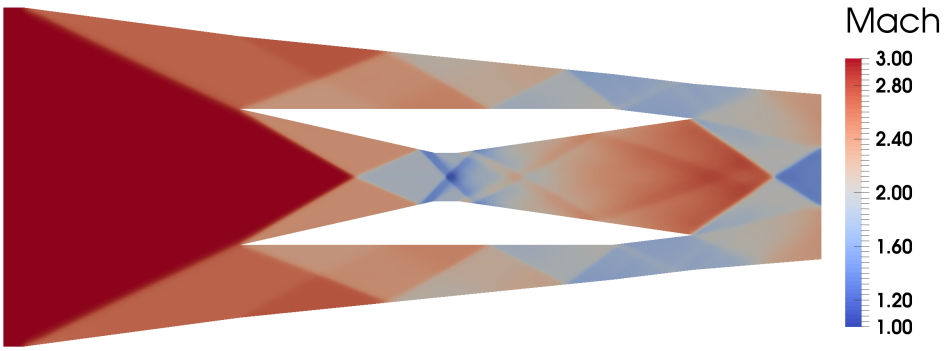


FIGURE 4.12: Scramjet Mach contours when a mesh of 63695 $\mathcal{Q}_1$ elements is used, with parameters $q = 5$, $\gamma = 10^{-10}$, and $\tilde{\varepsilon} = 1$.

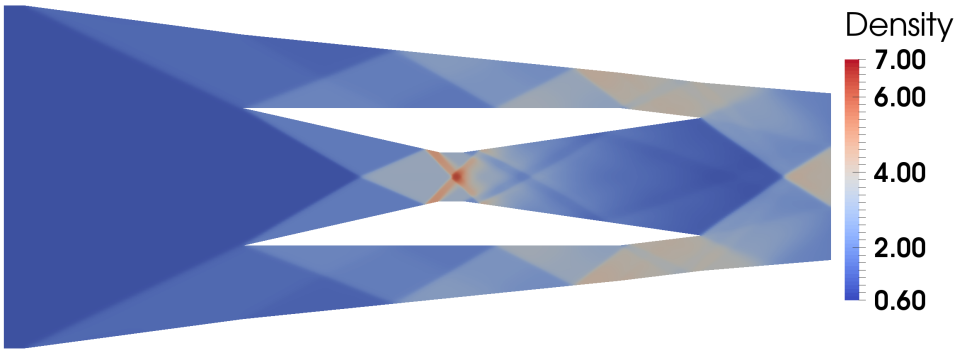


FIGURE 4.13: Scramjet Mach contours when a mesh of 63695 $\mathcal{Q}_1$ elements is used, with parameters $q = 5$, $\gamma = 10^{-10}$, and $\tilde{\varepsilon} = 1$.

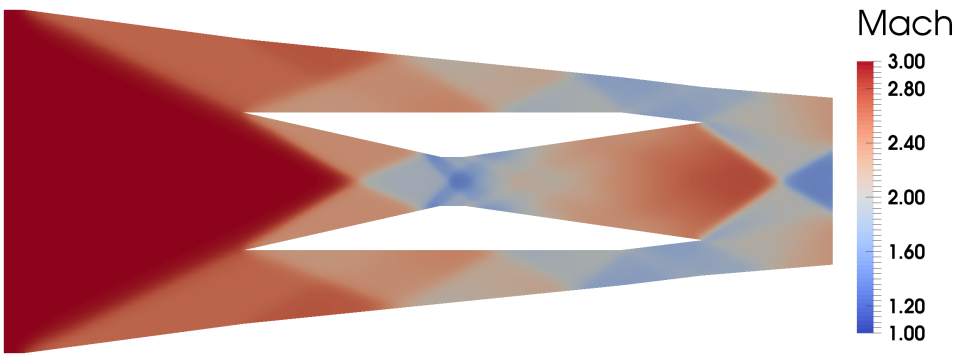


FIGURE 4.15: Scramjet Mach contours when a mesh of 18476 $\mathcal{Q}_1$ elements is used, with parameters $q = 2$, $\gamma = 10^{-10}$, and $\tilde{\varepsilon} = 1$.

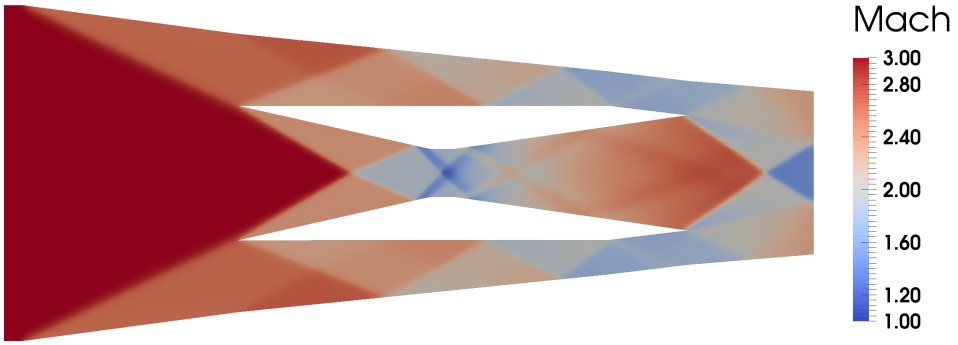
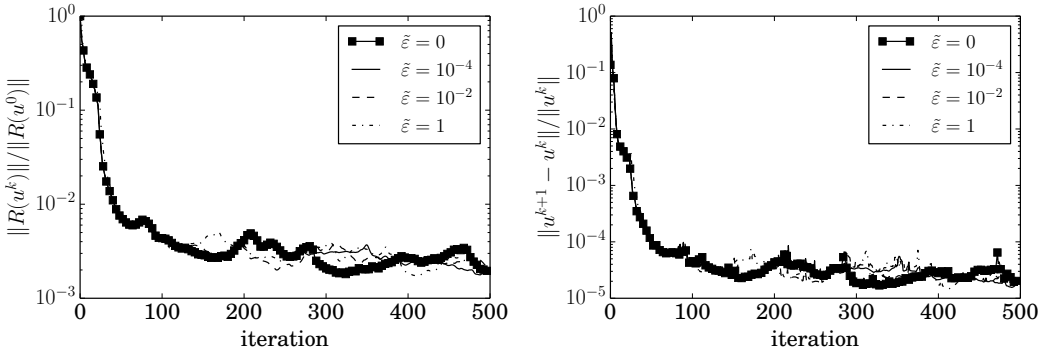


FIGURE 4.14: Scramjet Mach contours when a mesh of 63695 $\mathcal{Q}_1$ elements is used, with parameters $q = 2$, $\gamma = 10^{-10}$, and $\tilde{\varepsilon} = 1$.

Figs. 4.16–4.17 show the nonlinear convergence history in terms of the relative residual reduction and the relative solution increment between iterations. We can observe that the convergence is not ensured for an arbitrary choice of the regularization parameters. In fact, only the tests that use $\tilde{\varepsilon} = 1$ do not diverge for $q = 5$, regardless of the mesh used. Therefore, we can see that increasing the values of the regularization parameters not only improves the convergence, but also the robustness of the method.

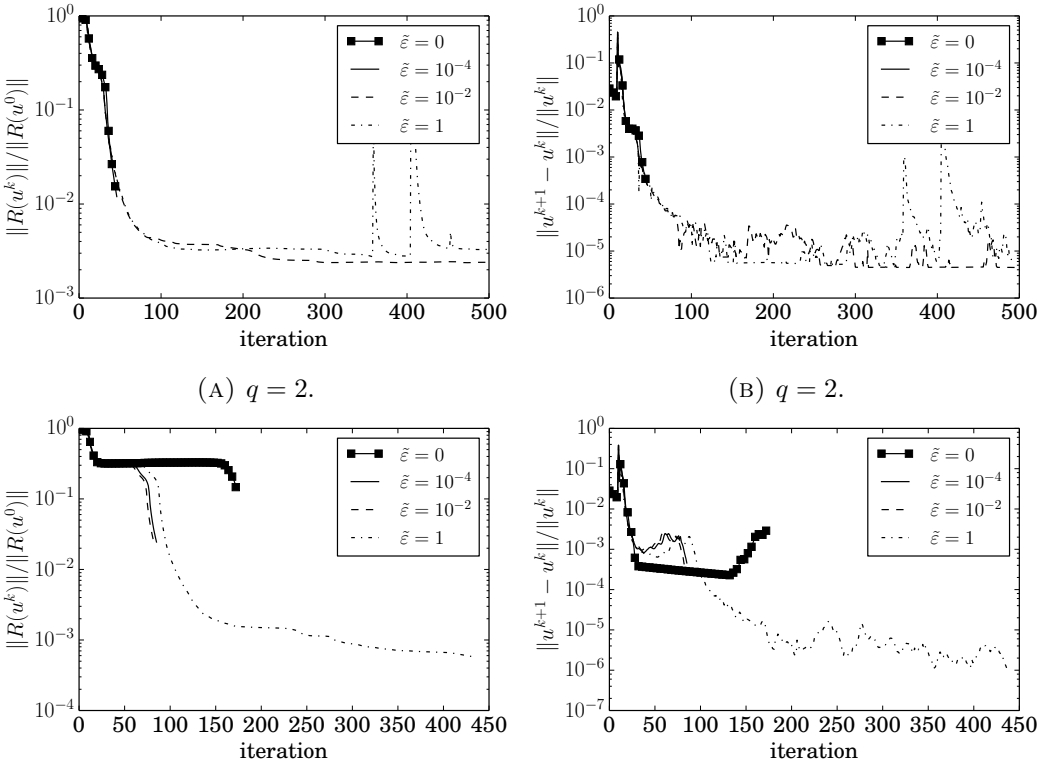


(A) $q = 2$. (B) $q = 2$.

(C) $q = 5$. (D) $q = 5$.

FIGURE 4.16: Comparison of the convergence behavior for the Scramjet test and different regularization parameters choices. A coarse mesh of 18476 $\mathcal{Q}_1$ elements is used.

However, it is important to mention that even if we can improve the convergence behavior of these types of methods, this is not enough for directly solving to steady state problems with complex shock patterns. For instance, even if the solution Fig. 4.15 seems to be correct, the scheme was unable to converge to the desired tolerance (see Figs. 4.16(c) and 4.16(d)). However the ability to introduce differentiability into the definition of the shock detector, for robustness and increased nonlinear convergence rates, could be coupled with popular pseudo-time stepping approaches [51, 86] to pursue improved methods for complex shock type systems.



(A) $q = 2$. (B) $q = 2$.

(C) $q = 5$. (D) $q = 5$.

FIGURE 4.17: Comparison of the convergence behavior for the Scramjet test and different regularization parameters choices. A fine mesh of 63695 $\mathcal{Q}_1$ elements is used.

4.6 Conclusions

In this chapter, a differentiable *local bounds preserving* stabilization method for Euler equations has been developed. This stabilization is based on the combination of a differentiable shock detector, a partially lumped mass matrix, and Rusanov artificial diffusion operator.

The resulting scheme has been successfully tested for steady and transient benchmark problems. Numerical results show that the proposed method exhibits good stability

properties. Furthermore, it is able to provide well resolved sharp shocks in both steady and transient problems.

In addition, to improve nonlinear convergence, a continuation method for the regularization parameters present in the differentiable stabilization has also been proposed. Nonlinear convergence of the scheme has been analyzed for the differentiable version and compared with its non-regularized counterpart. In general terms, the differentiable stabilization shows better convergence, especially when the hybrid Picard–Newton method is used. For small steady problems, the scheme is able to converge directly to the steady state solution without making use of pseudo-transient time stepping. However, for problems with complex shock patterns the scheme only converges to moderate tolerances. Numerical results also show that differentiability not only can improve nonlinear convergence, but it also improves the robustness of the method.

In the case of transient problems, some improvement in the computational cost is observed. However, since the non-differentiable method already exhibits good nonlinear convergence, there is not much room for improvement. Nevertheless, it is possible to show that the differentiable stabilization can achieve a similar accuracy while requiring a lower computational cost.

Chapter 5

Monotonicity-preserving FE schemes with AMR for hyperbolic problems

This chapter is focused on the extension and assessment of the monotonicity-preserving scheme in Chapter 2 and the local bounds preserving scheme in Chapter 4 to hierarchical octree AMR. Whereas the former can readily be used on this kind of meshes, the latter requires some modifications. A key question that we want to answer in this chapter is whether to move from a linear to a nonlinear stabilization mechanism pays the price when combined with shock-adapted meshes. Whereas nonlinear (or shock-capturing) stabilization leads to improved accuracy compared to linear schemes, it also negatively hinders nonlinear convergence, increasing computational cost. We compare linear and nonlinear schemes in terms of the required computational time versus accuracy for several steady benchmark problems. Numerical results indicate that, in general, nonlinear schemes can be cost-effective for sufficiently refined meshes. Besides, it is also observed that it is better to refine further around shocks rather than using sharper shock capturing terms, which usually yield stiffer nonlinear problems. In addition, a new refinement criterion has been proposed. The proposed criterion is based on the graph Laplacian used in the definition of the stabilization method. Numerical results show that this shock detector performs better than the well-known Kelly estimator for problems with shocks or discontinuities.

5.1 Introduction

Natural phenomena can develop shock waves in different scenarios. A classical example is the shock wave generated by an object traveling faster than sound. The numerical modeling of problems with shocks is still a challenge, especially when the admissible physical solution has some physical constraints, e.g., positivity or non-negativity, that must be preserved at the discrete level to have well-posedness; E.g., the fluid density and temperature are positive quantities in a compressible flow.

Several numerical schemes have been proposed so far to approximate this kind of problems by combining FVM or dG FEs for space discretization with explicit time integrators (see [27, 34, 68, 91]). Explicit time integrators are only stable under a CFL restriction over the time step size, which implies to capture all time scales. Thus, explicit methods are not suitable for problems in which the smallest time scales are not of interest. For instance, the fastest time scales at a confined plasma in a nuclear fusion reactor are not of engineering interest whereas explicit time integration is unaffordable in practical simulations [54].

Implicit monotonicity-preserving (or at least positivity-preserving) methods are still scarce. As proved by Godunov [36], linear monotonicity-preserving schemes can be at most first-order accurate. For scalar problems (and under some mesh restrictions), Burman and Ern [25], Barrenechea and co-workers [13, 14], Kuzmin and co-workers [58, 61, 71], and Badia and Hierro [7, 8] have proposed nonlinear schemes that preserve monotonicity and can presumably attain higher order accuracy.[1] However, these properties comes at the cost of solving a very stiff nonlinear problem [57]. The authors [4, 5] have proposed differentiable schemes that improve the nonlinear convergence behavior of previous methods.

For hyperbolic systems of equations, numerical methods are even less well developed. For explicit time integration, Guermond and Popov [42] have recently proposed a cG FE scheme that preserves positivity of density and energy under certain CFL-like condition. Unfortunately, these ideas cannot be easily extended to implicit time integration and we are not aware of any implicit method that theoretically satisfies such properties. Kuzmin and co-workers [61, 69, 74, 75] have proposed various schemes based on FCT [72] that are experimentally robust, but lack of a theoretical analysis. Besides, this strategy also yields very stiff nonlinear problems. Differentiable schemes for compressible flows have been proposed in Chapter 4 to alleviate (but not eliminate) this problem.

Shocks are non-smooth and localized and thus suitable for AMR [31, 92]. AMR allows one to increase the mesh resolution only in the vicinity of shocks or discontinuities. In brief, the AMR process can be divided into two main ingredients. On the one hand, to estimate the error at each element. On the other hand, to decide which elements need to be refined or coarsened. This iterative process provides a mesh locally adapted to the features of the problem at hand. As a result, it is a nonlinear approximation scheme which tries to minimize the error for a target computational cost. If performed optimally, AMR exhibits exponential convergence even for solutions with limited regularity [31].

In this context, a key question is whether it is computationally more effective to consider a nonlinear high-order scheme (with the nonlinear convergence issues) or a cheaper linear (first-order) scheme in a much refined mesh. The motivation of this chapter is to shed light on this issue. First, we adapt the schemes developed in Chapters 2 and 4 to hierarchical octree AMR [10, 89]. Next, we propose a refinement criterion that

[1] In this chapter, schemes with nonlinear stabilization are also referred to as high-order and linear stabilization schemes as low or first-order.

relies on information already present in the stabilization technique; nonlinear stabilization methods include a shock detector to activate the artificial diffusion only close to discontinuities. We propose to use a modification of the shock detector in Chapter 2 to drive the AMR process.

This chapter is structured as follows. First, we introduce the problem, its discretization, and monotonicity properties for scalar problems and hyperbolic systems in Sect. 5.2. Then, the stabilization techniques are introduced in Sect. 5.3. Sect. 5.4 is devoted to the AMR strategy. We introduce the nonlinear solvers in Sect. 5.5. Finally, we show numerical experiments in Sect. 5.6 and draw some conclusions in Sect. 5.7.

5.2 Preliminaries

5.2.1 Continuous problem

Let us consider an open bounded and connected domain, $\Omega \in \mathbb{R}^d$, where d is the number of spatial dimensions. Let $\partial\Omega$ be the Lipschitz continuous boundary of Ω. The conservative form of a first order hyperbolic problem reads

$$\begin{cases} \partial_t \boldsymbol{u} - \boldsymbol{\nabla} \cdot \boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{g}, & \text{in } \Omega \times (0,T], \\ u^\beta(x,t) = \bar{u}^\beta(x,t), \text{ on } \Gamma^\beta_{\text{in}} \times (0,T], \ \beta = 1, ..., m, \\ \boldsymbol{u}(x,0) = \boldsymbol{u}_0(x), \quad x \in \Omega, \end{cases} \tag{5.1}$$

where $\boldsymbol{u} = \{u^\beta\}_{\beta=1}^m$ are $m \geq 1$ conserved variables, $\boldsymbol{f}$ is the physical flux, $\bar{u}^\beta(x,t)$ are the boundary values for the βth-component of $\boldsymbol{u}$, $\boldsymbol{u}_0(x)$ are the initial conditions, and $\boldsymbol{g}(x,t)$ is a function defining the body forces. Note that the flux, $\boldsymbol{f} : \mathbb{R}^m \to \mathbb{R}^{m\times d}$, is composed of $\boldsymbol{f} = \{\boldsymbol{f}_i\}_{i=1}^d$, where $\boldsymbol{f}_i : \mathbb{R}^m \to \mathbb{R}^m$ is the flux in the ith spatial direction. We denote by $\boldsymbol{f}' : \mathbb{R}^m \to \mathbb{R}^{m\times m\times d}$ the flux Jacobian. Let $\boldsymbol{n} \in \mathbb{R}^d$ be any direction vector. Since the system is hyperbolic, the flux Jacobian in any direction is diagonalizable and has only real eigenvalues, i.e., $\boldsymbol{f}'(\boldsymbol{u}) \cdot \boldsymbol{n} = \sum_{i=1}^d \boldsymbol{f}_i'(\boldsymbol{u}) n_i$ is diagonalizable with real eigenvalues $\{\lambda_\beta\}_{\beta=1}^m$. These eigenvalues might have different multiplicities and different signs. Hence, for a given direction $\boldsymbol{n}$, each characteristic variable might be convected forward (along $\boldsymbol{n}$) or backwards (along $-\boldsymbol{n}$). Therefore, it is convenient to define inflow and outflow boundaries for each component. The inflow boundary for component β is defined as $\Gamma^\beta_{\text{in}} \doteq \{\boldsymbol{x} \in \partial\Omega \ : \ \lambda_\beta(\boldsymbol{f}'(\boldsymbol{u}) \cdot \boldsymbol{n}_{\partial\Omega}) \leq 0\}$, where $\boldsymbol{n}_{\partial\Omega}$ is the unit outward normal to the boundary and λ_β is the βth-eigenvalue of the flux Jacobian. We define the outflow boundary as $\Gamma^\beta_{\text{out}} \doteq \partial\Omega \backslash \Gamma^\beta_{\text{in}}$. We refer the reader to [34, 43, 91] for a detailed discussion on boundary conditions for hyperbolic problems. In the present study, we will also consider the steady counterpart of (5.1), which is obtained by dropping the time derivative term and the initial conditions.

In this chapter, we work with both scalar convection equations and Euler equations. Taking $m = 1$ and $\boldsymbol{f}(u) \doteq \mathbf{v}u$ with $\mathbf{v}$ a divergence-free convection field, we recover the well known scalar transport problem. On the other hand, Euler equations for ideal gases

are recovered by defining $m = d + 2$ and

$$\boldsymbol{u} \doteq \begin{pmatrix} \rho \\ \boldsymbol{m} \\ \rho E \end{pmatrix}, \quad \boldsymbol{f} \doteq \begin{pmatrix} \boldsymbol{m} \\ \boldsymbol{m} \otimes \mathbf{v} + p\mathbb{I} \\ \mathbf{v}(\rho E + p) \end{pmatrix}, \text{ and } \boldsymbol{g} \doteq \begin{pmatrix} 0 \\ \boldsymbol{b} \\ \boldsymbol{b} \cdot \mathbf{v} + r \end{pmatrix},$$

where ρ is the density, E is the total energy, p is the pressure, $\boldsymbol{m} = \{m_1, \ldots, m_d\}$, where $m_i = \rho v_i$, is the momentum, $\mathbf{v} = \{v_1, \ldots, v_d\}$ is the velocity, $\boldsymbol{b} = \{b_1, \ldots, b_d\}$ are the body forces, r is an energy source term per unit mass, and $\mathbb{I}$ is the identity matrix of dimension $d \times d$. In addition, the system is equipped with the ideal gas equation of state $p = (\gamma - 1)\rho\imath$, where $\imath = E - \frac{1}{2}\|\mathbf{v}\|^2$ is the internal energy and γ is the adiabatic index.

5.2.2 Discretization

The discretization used in this chapter is able to adapt its local size to the features of the problem at hand. In particular, it is a hierarchically refined octree-based hexahedral mesh [89]. This type of discretizations are constructed hierarchically. At every step of the refinement process, marked cells are refined into four (eight) cells in 2D (3D). The adaptation of the mesh to the problem at hand is achieved by only marking for refining a targeted amount of cells. This results in a mesh with different refinement levels at different regions. *Hanging* nodes appear at the interface between cells at different refinement levels. These are nodes that only belong to the cells at a higher refinement level (see Fig. 5.1). In our case, the meshes used are *2:1 balanced*. This restriction implies that there can only be a difference of one refinement level between neighboring cells. This restriction is a trade-off between implementation complexity and performance gain that has been adopted by many AMR codes [89].

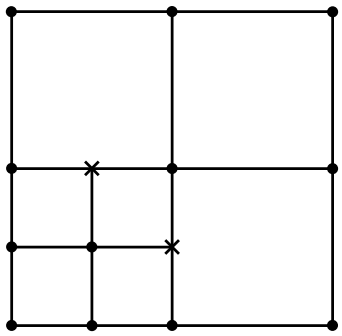


FIGURE 5.1: Example of a mesh with *hanging* nodes.

Hanging nodes need to be treated carefully in the case of working with conforming FE discretizations. Otherwise, associating a regular degree of freedom (DOF) to a hanging node may lead to discontinuities in the approximated solution. To preserve continuity of the FE space, *hanging DOFs* values are not included in the assembled system of equations but obtained by interpolating the values of the neighboring *regular* DOFs. For more details in the definitions of these constraints we refer the reader to [9–11].

Let $\mathcal{T}_h$ be a hierarchical octree-based partition of Ω. Consider a Lagrangian (nodal) FE space on top of this mesh. The set of all nodes in the FE space is represented with $\tilde{\mathcal{N}}_h$. For every node $i \in \tilde{\mathcal{N}}_h$, $\boldsymbol{x}_i$ stands for the node coordinates. We can split $\tilde{\mathcal{N}}_h$ into two subsets, namely the set of hanging nodes $\mathcal{N}_h^{hg}$ and the set of conforming nodes $\mathcal{N}_h \doteq \tilde{\mathcal{N}}_h \backslash \mathcal{N}_h^{hg}$. We denote by $N \doteq \text{card}(\mathcal{N}_h)$ the total number of conforming nodes. The set of nodes belonging to a particular element $K \in \mathcal{T}_h$ is defined as $\mathcal{N}_h(K) \doteq \{i \in \mathcal{N}_h : \boldsymbol{x}_i \in K\}$. Moreover, Ω_i is the macroelement composed by the union of elements that contain node i, i.e., $\Omega_i \doteq \bigcup_{K \in \mathcal{T}_h,\, \boldsymbol{x}_i \in K} K$. To simplify the discussion below, we abuse notation and use i for both the node and its associated index.

We restrict the present study to first order FEs and define the FE space as follows. We define $\boldsymbol{V}_h \doteq \left\{\boldsymbol{v}_h \in (\mathcal{C}^0(\Omega))^m : \boldsymbol{v}_h|_K \in (Q_1(K))^m \forall K \in \mathcal{T}_h\right\}$, where $Q_1(K)$ is the space of polynomials of partial degree less than or equal to one. Furthermore, we define the space $\boldsymbol{V}_{h0} \doteq \{\boldsymbol{v}_h \in \boldsymbol{V}_h \ : \ \boldsymbol{v}_h(\boldsymbol{x}) = 0 \ \forall \boldsymbol{x} \in \Gamma_{\text{in}}\}$. The functions $\boldsymbol{v}_h \in \boldsymbol{V}_h$ can be constructed as a linear combination of the basis $\{\boldsymbol{\varphi}_i^\beta\}_{i \in \tilde{\mathcal{N}}_h}^{1 \le \beta \le m}$ and nodal values $\boldsymbol{v}_i$, where $\boldsymbol{\varphi}_i^\beta = \varphi_i\{\delta_{1\beta}, \ldots, \delta_{m\beta}\}$ is the shape function associated to the component β of node i, and $\delta_{\alpha\beta}$ is the Kronecker delta. Thus, only component β of $\boldsymbol{\varphi}_i^\beta$ is different from zero, which takes the value of the classical scalar shape function used in Lagrangian FE, φ_i. Hence, $\boldsymbol{v}_h = \sum_{i \in \tilde{\mathcal{N}}_h,\, 1 \le \beta \le m} \boldsymbol{\varphi}_i^\beta v_i^\beta = \sum_{i \in \tilde{\mathcal{N}}_h} \varphi_i \boldsymbol{v}_i$.

We use standard notation for Sobolev spaces. The $L^2(\omega)$ scalar product is denoted by $(\cdot,\cdot)_\omega$ for $\omega \subset \Omega$. However, we omit the subscript for $\omega \equiv \Omega$. The L^2 norm is denoted by $\|\cdot\|$. Using this notation, the weak form of problem (5.1) reads as follows. Find $\boldsymbol{u} \in L^2(\Omega)$ such that $u^\beta = \bar{u}^\beta$ on $\Gamma_{\text{in}}^\beta \times (0,T]$ for $\beta = 1, ..., m$ and

$$(\partial_t \boldsymbol{u}, \boldsymbol{v}) + (\boldsymbol{u}, \boldsymbol{f}'(\boldsymbol{u}) : \boldsymbol{\nabla v}) - (\boldsymbol{u}, \boldsymbol{n}_{\Gamma_{\text{out}}} \cdot \boldsymbol{f}'(\boldsymbol{u})\boldsymbol{v})_{\Gamma_{\text{out}}} = (\boldsymbol{g}, \boldsymbol{v}), \ \forall \boldsymbol{v} \in L_0^2(\Omega), \tag{5.2}$$

subject to appropriate initial conditions $\boldsymbol{u}(x,0) = \boldsymbol{u}_0(x)$. Note that the double contraction is applied as $\boldsymbol{f}'(\boldsymbol{u}) : \boldsymbol{\nabla v} = \sum_{k,\gamma} \boldsymbol{f}'_k(\boldsymbol{u})^{\beta\gamma}\ \boldsymbol{v}_{\gamma,\beta}$.

The method of lines is applied in combination with the FE spaces described above for the spatial discretization. We make the approximation $\boldsymbol{u} \approx \boldsymbol{u}_h = \sum_{i \in \tilde{\mathcal{N}}_h,\, 1 \le \beta \le m} \boldsymbol{\varphi}_i^\beta u_i^\beta = \sum_{i \in \tilde{\mathcal{N}}_h} \varphi_i \boldsymbol{u}_i$ for the unknown. Likewise, the fluxes are approximated as $\boldsymbol{f} \approx \boldsymbol{f}_h = \sum_{i \in \tilde{\mathcal{N}}_h,\, 1 \le \beta \le m} \boldsymbol{\varphi}_i^\beta \boldsymbol{f}(\boldsymbol{u}_i)^\beta = \sum_{i \in \tilde{\mathcal{N}}_h} \varphi_i \boldsymbol{f}(\boldsymbol{u}_i)$. For simplicity in the exposition, we use the BE scheme for the time discretization; higher order time discretizations can be achieved using SSP–RK methods (see [37]). In the latter case, a CFL-like condition must be satisfied to enjoy the monotonicity properties in Sect. 5.2.3 (see [60, 67]).

The semi-discrete Galerkin FE approximation (5.2) reads: find $\boldsymbol{u}_h \in \boldsymbol{V}_h$ such that $u_h^\beta = \bar{u}_h^\beta$ on Γ_{in}^β, $\boldsymbol{u}_h = \boldsymbol{u}_{0h}$ at $t = 0$, and

$$(\partial_t \boldsymbol{u}_h, \boldsymbol{v}_h) + (\boldsymbol{u}_h, \boldsymbol{f}'_h(\boldsymbol{u}_h) : \boldsymbol{\nabla v}_h) - (\boldsymbol{u}_h, \boldsymbol{n}_{\Gamma_{\text{out}}} \cdot \boldsymbol{f}'_h(\boldsymbol{u}_h)\boldsymbol{v}_h)_{\Gamma_{\text{out}}} = (\boldsymbol{g}, \boldsymbol{v}_h), \ \forall \boldsymbol{v}_h \in \boldsymbol{V}_{h0},$$

where $\bar{u}_h^\beta$ and $\boldsymbol{u}_{0h}$ are admissible FE approximations of $\bar{u}^\beta$ and $\boldsymbol{u}_0$. In this context, we consider admissible any approximation that satisfies the maximum principle, i.e., it does

not introduce new extrema.

As commented above, we need to apply constraints to all hanging DOFs to keep conformity. The value of the hanging DOF needs to be equal to the value of the interpolation of the unknown at the neighboring coarser elements. That is, given $i \in \mathcal{N}_h^{hg}$ and its neighboring (coarse) FE $K \in \mathcal{T}_h$, $\boldsymbol{v}_i = \sum_{j\in\mathcal{N}_h(K)} \varphi_j(\boldsymbol{x}_i)\boldsymbol{v}_j$. In general, we will represent this constraint with $\boldsymbol{v}_i = \sum_{j\in\mathcal{N}_h} C_{ij}\boldsymbol{v}_j$. For details of the implementation of this kind of constraints see [9–11].

Finally, to obtain the fully discrete problem, we consider a partition of the time domain $(0,T]$ into n^{ts} sub-intervals of length $(t^n, t^{n+1}]$. Then, at every time step $n = 0,\dots,n^{ts}-1$, the discrete problem consists in solving

$$\mathbf{M}\delta_t\mathbf{U}^{n+1} + \mathbf{K}\mathbf{U}^{n+1} = \mathbf{G}, \tag{5.3}$$

where $\mathbf{U}^{n+1} \doteq [\boldsymbol{u}_1^{n+1},...,\boldsymbol{u}_N^{n+1}]^T$ is the vector of nodal values at time t^{n+1}, $\delta_t(\mathbf{U}) \doteq \Delta t_{n+1}^{-1}(\mathbf{U}^{n+1} - \mathbf{U}^n)$, and $\Delta t_{n+1} \doteq (t^{n+1} - t^n)$. The $m \times m$-matrices relating nodes $i,j \in \mathcal{N}_h$ are given by

$$\begin{aligned}
\mathbf{M}_{ij}^{\beta\gamma} &\doteq (\varphi_j,\varphi_i)\delta_{\beta\gamma} + \boldsymbol{\mathcal{M}}_{ij}^{\beta\gamma},\\
\mathbf{K}_{ij}^{\beta\gamma} &\doteq (\varphi_j\delta_{\beta\xi}, \boldsymbol{f}_k'(\boldsymbol{u}_j^{n+1})^{\xi\eta}\cdot\partial_k\varphi_i\delta_{\eta\gamma}) - (\varphi_j\delta_{\beta\xi}, n_k\cdot\boldsymbol{f}_k'(\boldsymbol{u}_j^{n+1})^{\xi\eta}\varphi_i\delta_{\eta\gamma})_{\Gamma_{\text{out}}} + \boldsymbol{\mathcal{K}}_{ij}^{\beta\gamma},\\
\mathbf{G}_i^{\beta} &\doteq (\boldsymbol{g}^{\beta},\varphi_i) + \boldsymbol{\mathcal{G}}_i^{\beta},
\end{aligned}$$

where Einstein summation applies, $\beta,\gamma,\xi,\eta \in \{1,\dots,m\}$ are the component indices, and $\boldsymbol{\mathcal{M}}$, $\boldsymbol{\mathcal{K}}$, and $\boldsymbol{\mathcal{G}}$ are the terms arising from applying the constraints in the mass, flux and body forces terms.

5.2.3 Stability properties

Finally, let us review some concepts required for discussing the stabilization method used in the subsequent sections. Let us recall some definitions used for scalar problems.

Definition 5.2.1 (Local discrete extremum)**.** *The function $v_h \in V_h$ has a local discrete minimum (resp. maximum) on $i \in \mathcal{N}_h$ if $u_i \le u_j$ (resp. $u_i \ge u_j$) $\forall j \in \mathcal{N}_h(\Omega_i)$.*

Definition 5.2.2 (Local DMP)**.** *A solution $u_h \in V_h$ satisfies the local discrete maximum principle if for every $i \in \mathcal{N}_h$*

$$\min_{j\in\mathcal{N}_h(\Omega_i)\setminus\{i\}} u_j \le u_i \le \max_{j\in\mathcal{N}_h(\Omega_i)\setminus\{i\}} u_j.$$

Definition 5.2.3 (LED)**.** *A scheme is local extremum diminishing if, for every u_i that is a local discrete maximum (resp. minimum),*

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} \le 0, \qquad \left(\textit{resp. } \frac{\mathrm{d}u_i}{\mathrm{d}t} \ge 0\right),$$

is satisfied.

One possible strategy to satisfy the above properties consists in designing a scheme that yields a positive diagonal mass matrix and a stiffness matrix that satisfies

$$\sum_j \mathsf{A}_{ij} = 0, \quad \text{and} \quad \mathsf{A}_{ij} \leq 0,\ i \neq j.$$

In this case, it is possible to rewrite the system as

$$m_i \frac{du_i}{dt} + \sum_{j \in \mathcal{N}_h(\Omega_i)\backslash\{i\}} \mathsf{A}_{ij}(u_j - u_i) = 0, \quad \forall\, i \in \mathcal{N}_h.$$

As shown in [28, 67], such a scheme satisfies the local DMP for steady problems and it is also LED when applied to transient problems.

Stability properties for hyperbolic systems can be based on the extension of the above to hyperbolic systems in characteristic variables. In this direction, we define local bounds preserving schemes as follows.

Definition 5.2.4. *The semi-discrete scheme*

$$\sum_j \mathbf{M}_{ij} \partial_t \boldsymbol{u}_j + \sum_{j \neq i} \mathsf{A}_{ij}(\boldsymbol{u}_j - \boldsymbol{u}_i) = \mathbf{0}$$

is said to be local bounds preserving if $\mathbf{M}$ *is diagonal with positive entries (i.e.,* $\mathbf{M}_{ij} = m_i \delta_{ij} I_{m\times m}$*),* A_{ij} *has non-positive eigenvalues for every* $j \neq i$*, and* $\sum_j \mathsf{A}_{ij} = \mathbf{0}$*.*

Unfortunately, to the best of our knowledge, satisfying this definition does not ensure positivity of density, internal energy, or non-decreasing entropy. In any case, numerical schemes based on this definition have shown good numerical behavior [59, 62, 70, 75].

Several stabilization strategies have been defined based on the previous ideas. One of the most simple strategies consists in adding a scalar artificial diffusion term proportional to the spectral radius of A_{ij} [59, 73]. This strategy is usually called Rusanov artificial diffusion, since the scheme results in the Rusanov Riemann solver for linear FEs in one dimension [59, 91]. Without any special treatment, the resulting scheme is only first order accurate. The key for recovering high-order convergence is to modulate the action of the artificial diffusion term, and restrict its action to the vicinity of discontinuities. In the present chapter, our stabilization term for systems of equations is based on Rusanov artificial diffusion and a differentiable shock detector recently developed for scalar problems in Chapters 2 and 3.

Finally, it is also important to define the concept of linearity preservation.

Definition 5.2.5. *Given* $\boldsymbol{u}_h \in \boldsymbol{V}_h$*, a stabilization scheme is said to be linearity preserving if at any region such that* $u_h^\beta \in P_1(\Omega)\, \forall\, \beta \in C$*, then* $\mathbf{B}(\boldsymbol{u}_h) = 0$*.*

5.3 Nonlinear stabilization

In this section, we describe the additional terms used for the stabilized problem (5.3). In particular, we use the stabilization terms defined in Chapter 2 for the scalar problem and Chapter 4 for Euler:

$$B_h(\boldsymbol{w}_h;\boldsymbol{u}_h,\boldsymbol{v}_h) \doteq \begin{cases} \sum_{i\in\mathcal{N}_h}\sum_{j\in\mathcal{N}_h(\Omega_i)} \nu_{ij}(w_h) v_i u_j \ell(i,j), & \text{for } m=1, \\ \sum_{K_e\in\mathcal{T}_h}\sum_{\substack{i,j\in\mathcal{N}_h(K_e)\\ 1\le\beta,\gamma\le m}} \nu^e_{ij}(\boldsymbol{w}_h)\ell(i,j)\, v_i^\beta\cdot\delta_{\beta\gamma}u_j^\gamma, & \text{for } m>1, \end{cases} \tag{5.4}$$

for any $\boldsymbol{w}_h,\ \boldsymbol{u}_h,\ \boldsymbol{v}_h \in \boldsymbol{V}_h$. Here, ℓ is the graph-Laplacian operator defined as $\ell(i,j) \doteq 2\delta_{ij}-1$ (see [39]). In the case of a scalar problem, $m=1$, the nodal artificial diffusion $\nu_{ij}(w_h)$ is defined as

$$\begin{aligned} \nu_{ij}(w_h) &\doteq \max\{\alpha_i(w_h)\mathbf{K}_{ij}, 0, \alpha_j(w_h)\mathbf{K}_{ji}\} \quad \text{for} \quad j\in\mathcal{N}_h(\Omega_i)\backslash\{i\}, \\ \nu_{ii}(w_h) &\doteq \sum_{j\in\mathcal{N}_h(\Omega_i)\backslash\{i\}} \nu_{ij}(w_h). \end{aligned}$$

We denote by $\alpha(w_h)$ the scalar shock detector used for computing the artificial diffusion parameter. In the case of the Euler equations, the element-wise artificial diffusion $\nu^e_{ij}(\boldsymbol{w}_h)$ is defined as

$$\begin{aligned} \nu^e_{ij}(\boldsymbol{w}_h) \doteq{}& \max\left(\boldsymbol{\alpha}_i(\boldsymbol{w}_h)\lambda^{\max}_{ij}, \boldsymbol{\alpha}_j(\boldsymbol{w}_h)\lambda^{\max}_{ji}\right) \\ &+ \sum_{k\in\mathcal{M}(i)} C_{ki}\max\left(\boldsymbol{\alpha}_k(\boldsymbol{w}_h)\lambda^{\max}_{kj}, \boldsymbol{\alpha}_j(\boldsymbol{w}_h)\lambda^{\max}_{jk}\right) \\ &+ \sum_{k\in\mathcal{M}(j)} C_{kj}\max\left(\boldsymbol{\alpha}_i(\boldsymbol{w}_h)\lambda^{\max}_{ik}, \boldsymbol{\alpha}_k(\boldsymbol{w}_h)\lambda^{\max}_{ki}\right) \\ &+ \sum_{k\in\mathcal{M}(i)\cap\mathcal{M}(j)} C_{ki}C_{kj}\boldsymbol{\alpha}_k(\boldsymbol{w}_h)\lambda^{\max}_{kk}, \quad \text{for } j\in\mathcal{N}_h(\Omega_i)\backslash\{i\}, \\ \nu^e_{ii}(\boldsymbol{w}_h) \doteq{}& \sum_{j\in\mathcal{N}_h(\Omega_i)\backslash\{i\}} \nu^e_{ij}(\boldsymbol{w}_h), \end{aligned} \tag{5.5}$$

where $\lambda^{\max}_{ij}$ is the spectral radius of the elemental convection matrix relating nodes $i,j\in\mathcal{N}_h$, i.e., $\rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij})\cdot(\boldsymbol{\nabla}\varphi_j,\varphi_i)_{K_e}\right)$, where $\boldsymbol{u}_{ij}$ is the Roe average between $\boldsymbol{u}_i$ and $\boldsymbol{u}_j$ (see (5.7)). Notice that due to the usage of the FE approximation of the flux, $\lambda^{\max}_{ik}$ is the spectral radius of $\rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij})\cdot(\boldsymbol{\nabla}\varphi_k,\varphi_i)_{K_e}\right)$.

As previously introduced, this artificial diffusion term is based on Rusanov scalar diffusion [61]. It is important to mention that the eigenvalues of these matrices can be easily computed as

$$\lambda_{1,..,d} = \mathbf{v}_{ij}\cdot\boldsymbol{c}^e_{ij}, \quad \lambda_{d+1} = \mathbf{v}_{ij}\cdot\boldsymbol{c}^e_{ij} - c\|\boldsymbol{c}^e_{ij}\|, \quad \lambda_{d+2} = \mathbf{v}_{ij}\cdot\boldsymbol{c}^e_{ij} + c\|\boldsymbol{c}^e_{ij}\|, \tag{5.6}$$

where the velocity $\mathbf{v}_{ij}$ and sound speed c are computed using the Roe averaged values

$$\begin{aligned} \boldsymbol{c}_{ij}^{e} &= (\boldsymbol{\nabla}\varphi_j, \varphi_i)_{K_e}, \quad c = \sqrt{(\gamma-1)\left(H_{ij} - \frac{\|\boldsymbol{m}_{ij}\|^2}{2\rho_{ij}^2}\right)}, \\ H_{ij} &= \frac{H_i\sqrt{\rho_i} + H_j\sqrt{\rho_j}}{\sqrt{\rho_i}+\sqrt{\rho_j}}, \quad H_i = E - \frac{p_i}{\rho_i}, \quad \rho_{ij} = \sqrt{\rho_i\rho_j}, \\ \boldsymbol{m}_{ij} &= \frac{\boldsymbol{m}_i\sqrt{\rho_j} + \boldsymbol{m}_j\sqrt{\rho_i}}{\sqrt{\rho_i}+\sqrt{\rho_j}}, \quad \text{and} \quad (\rho E)_{ij} = \frac{1}{2-\gamma}\left(\rho_{ij}H_{ij} - \frac{|\boldsymbol{m}_{ij}|^2}{2\rho_{ij}}\right). \end{aligned} \tag{5.7}$$

We denote by $\boldsymbol{\alpha}_i(\boldsymbol{w}_h)$ the shock detector used for modulating the action of the artificial diffusion term. The idea behind the definition of this detector is to minimize the amount of artificial diffusion introduced while stabilizing any oscillatory behavior. We ensure that Def. 5.2.4 is satisfied in regions where the local DMP is violated (see Def. 5.2.2) for any set of components. $\boldsymbol{\alpha}_i(\boldsymbol{w}_h)$ must be a positive real number that takes value 1 when $\boldsymbol{u}_h(\boldsymbol{x}_i)$ is an inadmissible value of $\boldsymbol{u}_h$, and smaller than 1 otherwise. To this end, we define

$$\boldsymbol{\alpha}_i(\boldsymbol{u}_h) \doteq \max\{\alpha_i(u_h^\beta)\}_{\beta\in C}, \quad \forall i \in \mathcal{N}_h \tag{5.8}$$

where C is the set of components that are used to detect inadmissible values of $\boldsymbol{u}_h$, e.g. density and total energy in the case of Euler equations. For simplicity, we restrict ourselves to the components of $\boldsymbol{u}_h$. However, derived quantities such as the pressure or internal energy can be also used. For scalar equations, since the stabilization is defined for the assembled system, the shock detector $\boldsymbol{\alpha}_i$ only needs to be defined for $i \in \mathcal{N}_h$. However, for the elemental definition used for Euler equations, it is also required for $i \in \mathcal{N}_h^{hg}$. In that case, we use the maximum of its constraining nodes, i.e.,

$$\boldsymbol{\alpha}_k(\boldsymbol{u}_h) \doteq \max_{\{j\in\mathcal{N}_h \,:\, C_j\neq 0\}} \boldsymbol{\alpha}_j(\boldsymbol{u}_h).$$

Let us recall some useful notation from Chapter 2 to introduce the scalar shock detector $\alpha_i(w_h)$. Let $\boldsymbol{r}_{ij} = \boldsymbol{x}_j - \boldsymbol{x}_i$ be the vector pointing from node $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ with $i,j \in \mathcal{N}_h$ and $\hat{\boldsymbol{r}}_{ij} \doteq \frac{\boldsymbol{r}_{ij}}{|\boldsymbol{r}_{ij}|}$. Recall that the set of points $\boldsymbol{x}_j$ for $j \in \mathcal{N}_h(\Omega_i)\backslash\{i\}$ define the macroelement Ω_i around node $\boldsymbol{x}_i$. Let $\boldsymbol{x}_{ij}^{\text{sym}}$ be the point at the intersection between $\partial\Omega_i$ and the line that passes through $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ that is not $\boldsymbol{x}_j$ (see Fig. 5.2). The set of all $\boldsymbol{x}_{ij}^{\text{sym}}$ for all $j \in \mathcal{N}_h(\Omega_i)\backslash\{i\}$ is represented with $\mathcal{N}_h^{\text{sym}}(\Omega_i)$. We define $\boldsymbol{r}_{ij}^{\text{sym}} \doteq \boldsymbol{x}_{ij}^{\text{sym}} - \boldsymbol{x}_i$. Given $\boldsymbol{x}_{ij}^{\text{sym}}$ in two dimensions, let as call a and b the indices of the vertices such that they define the edge in $\partial\Omega_{\boldsymbol{i}}$ that contains $\boldsymbol{x}_{ij}^{\text{sym}}$. We define $\boldsymbol{u}_j^{\text{sym}}$ as the value of $\boldsymbol{u}_h$ at $\boldsymbol{x}_{ij}^{\text{sym}}$, i.e., $\boldsymbol{u}_h(\boldsymbol{x}_{ij}^{\text{sym}})$.

Both $\boldsymbol{u}_{ij}^{\text{sym}}$ and $\boldsymbol{x}_{ij}^{\text{sym}}$ are only required to construct a linearity preserving shock detector. Let us define the jump and the mean of a linear approximation of component β

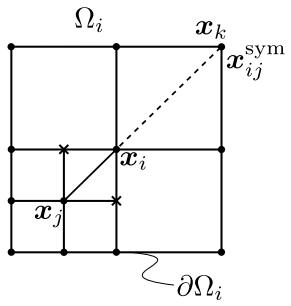


FIGURE 5.2: u^{sym} drawing

of the unknown gradient at node $\boldsymbol{x}_i$ in direction $\boldsymbol{r}_{ij}$ as

$$\left[\!\left[\boldsymbol{\nabla} u_h^\beta\right]\!\right]_{ij} \doteq \frac{u_j^\beta - u_i^\beta}{|\boldsymbol{r}_{ij}|} + \frac{u_j^{\mathrm{sym},\beta} - u_i^\beta}{|\boldsymbol{r}_{ij}^{\mathrm{sym}}|},$$

$$\left\{\!\!\left\{|\boldsymbol{\nabla} u_h^\beta \cdot \hat{\boldsymbol{r}}_{ij}|\right\}\!\!\right\}_{ij} \doteq \frac{1}{2}\left(\frac{|u_j^\beta - u_i^\beta|}{|\boldsymbol{r}_{ij}|} + \frac{|u_j^{\mathrm{sym},\beta} - u_i^\beta|}{|\boldsymbol{r}_{ij}^{\mathrm{sym}}|}\right).$$

For each component in C, we use the same shock detector developed in Chapter 2. Let us recall its definition

$$\alpha_i(u_h^\beta) \doteq \begin{cases} \left[\dfrac{\left|\sum_{j\in\mathcal{N}_h(\Omega_i)} \left[\!\left[\boldsymbol{\nabla} u_h^\beta\right]\!\right]_{ij}\right|}{\sum_{j\in\mathcal{N}_h(\Omega_i)} 2\left\{\!\!\left\{\left|\boldsymbol{\nabla} u_h^\beta \cdot \hat{\boldsymbol{r}}_{ij}\right|\right\}\!\!\right\}_{ij}}\right]^q & \text{if} \quad \sum_{j\in\mathcal{N}_h(\Omega_i)} \left\{\!\!\left\{|\boldsymbol{\nabla}\cdot\hat{\boldsymbol{r}}_{ij}|\right\}\!\!\right\}_{ij} \neq 0 \\ 0 & \text{otherwise} \end{cases}. \tag{5.9}$$

We know from Lm. 2.3.1 that (5.9) gets values between 0 and 1, and it is only equal to one if $u_h^\beta(\boldsymbol{x}_i)$ is a local discrete extremum (in a space–time sense as in Def. 5.2.1). Since the linear approximations of the unknown gradients are exact for $u_h^\beta \in \mathcal{P}_1$, the shock detector vanishes when the solution is linear. Thus, it is also linearly preserving for every component in C. This result follows directly from Th. 2.4.5.

The final stabilized problem in matrix form reads as follows. Find $\boldsymbol{u}_h \in \boldsymbol{V}_h$ such that $\boldsymbol{u}_h = \overline{\boldsymbol{u}}_h$ on $\partial\Omega$, $\boldsymbol{u}_h = \boldsymbol{u}_{0h}$ at $t = 0$, and

$$\overline{\mathbf{M}}(\boldsymbol{u}_h^{n+1})\delta_t \mathbf{U}^{n+1} + \overline{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{n+1})\mathbf{U}^{n+1} = \mathbf{G} \tag{5.10}$$

for $n = 1, ..., n^{ts}$, where

$$\overline{\mathbf{M}}_{ij}(\boldsymbol{u}_h^{n+1}) \doteq [1 - \max(\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j)]\,\mathbf{M}_{ij} + \max(\boldsymbol{\alpha}_i, \boldsymbol{\alpha}_j)\,\delta_{ij}\sum_j \mathbf{M}_{ij},$$

and $\overline{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{n+1}) \doteq \mathbf{K}_{ij} + \mathbf{B}_{ij}$, where $\mathbf{B}_{ij} \doteq B_h(u_h; \varphi_j, \varphi_i)$ is the stabilization matrix.

Let us show that adapted non-conforming meshes do not jeopardize any of the stability properties defined in Sect. 5.2.3 and proved for conforming meshes in [5, 9].

Corollary 5.3.1 (DMP)**.** *The solution of the discrete problem* (5.10) *with* $m = 1$ *and using the shock detector* (5.9) *satisfies the local DMP in Def. 5.2.2 if* $g = 0$ *and, for every control point* $i \in \mathcal{N}_h$ *such that* u_i *is a local discrete extremum, it holds:*

$$\overline{\mathbf{K}}_{ij}(u_h) \leq 0, \, \forall \, j \in \mathcal{N}_h(\Omega_i) \backslash \{i\}, \qquad \sum_{j \in \mathcal{N}_h(\Omega_i)} \overline{\mathbf{K}}_{ij}(u_h) = 0.$$

Moreover, the resulting scheme is linearity-preserving as defined in Def. 5.2.5, i.e., $\mathbf{B}_{ij}(u_h) = 0$ *for* $u_h \in \mathcal{P}_1(\Omega_i)$*.*

Proof. The stabilization scheme for scalar problems is defined on the assembled system. Hence, the modifications introduced in the assembly procedure do not affect the reasoning in the proof of [5, Thm. 5.2]. □

Lemma 5.3.2 (Local bounds preservation)**.** *Consider* $\boldsymbol{u}_h \in \boldsymbol{V}_h$ *with component* β *in the set of tracked variables* C*. The stabilized problem* (5.10) *is local bounds preserving as defined in Def. 5.2.4 at any region where* u_h^β *has extreme values.*

Proof. If component $\beta \in C$ of $\boldsymbol{u}_h$ has an extremum at $\boldsymbol{x}_i$, we know from Lm. 2.3.1 that $\alpha_i(u_h^\beta) = 1$. Moreover, it is easy to see from (5.8) that $\boldsymbol{\alpha}_i(\boldsymbol{u}_h) = 1$. In this case, $\overline{\mathbf{M}}_{ij}(\boldsymbol{u}_h) = \delta_{ij} \sum_j \mathbf{M}_{ij}$. Hence, $\overline{\mathbf{M}}_{ij}(\boldsymbol{u}_h) = 0$ for $j \neq i$ and $\overline{\mathbf{M}}_{ii}(\boldsymbol{u}_h) = m_i$. Therefore, we can rewrite the system as follows

$$\begin{aligned}
&m_i \partial_t \boldsymbol{u}_i + \sum_{j \in \mathcal{N}_h(\Omega_i)\backslash\{i\}} \overline{\mathbf{K}}_{ij}(\boldsymbol{u}_{ij})(\boldsymbol{u}_j - \boldsymbol{u}_i) = \\
&m_i \partial_t \boldsymbol{u}_i + \sum_{j \in \mathcal{N}_h(\Omega_i)\backslash\{i\}} \sum_{K_e \in \mathcal{T}_h} (\boldsymbol{f}'(\boldsymbol{u}_{ij})) \cdot \Big((\boldsymbol{\nabla}\varphi_j, \varphi_i)_{K_e} \\
&\qquad + \sum_{k \in \mathcal{M}(j)} C_{kj} (\boldsymbol{\nabla}\varphi_k, \varphi_i)_{K_e} \\
&\qquad + \sum_{k \in \mathcal{M}(i)} C_{ki} (\boldsymbol{\nabla}\varphi_j, \varphi_k)_{K_e} \\
&\qquad + \sum_{k \in \mathcal{M}(i) \cap \mathcal{M}(j)} C_{ki} C_{kj} (\boldsymbol{\nabla}\varphi_k, \varphi_k)_{K_e} \\
&\qquad - \nu_{ij}^e I_{m \times m} \Big) (\boldsymbol{u}_j - \boldsymbol{u}_i) = \mathbf{0}.
\end{aligned}$$

We need to prove that the eigenvalues of $\overline{\mathbf{K}}_{ij}(\boldsymbol{u}_{ij})$ are non-positive. To this end, let us show that the following inequality holds

$$\begin{aligned}\sum_{K_e\in\mathcal{T}_h}\Big(&\rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij})\cdot(\boldsymbol{\nabla}\varphi_j,\varphi_i)_{K_e}\right)\\&+\sum_{k\in\mathcal{M}(j)}C_{kj}\ \rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij})\cdot(\boldsymbol{\nabla}\varphi_k,\varphi_i)_{K_e}\right)\\&+\sum_{k\in\mathcal{M}(i)}C_{ki}\ \rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij})\cdot(\boldsymbol{\nabla}\varphi_j,\varphi_k)_{K_e}\right)\\&+\sum_{k\in\mathcal{M}(i)\cap\mathcal{M}(j)}C_{ki}C_{kj}\ \rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij})\cdot(\boldsymbol{\nabla}\varphi_k,\varphi_k)_{K_e}\right)\Big)\geq\rho(\boldsymbol{f}'(\boldsymbol{u}_{ij})\cdot(\boldsymbol{\nabla}\varphi_j,\varphi_i)).\end{aligned}$$

From (5.6), it is easy to check that $\rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij})\cdot(\boldsymbol{\nabla}\varphi_j,\varphi_i)_{K_e}\right)=|\mathbf{v}_{ij}\cdot\boldsymbol{c}_{ij}^e|+c_{ij}\|\boldsymbol{c}_{ij}^e\|$. We have that

$$\begin{aligned}\boldsymbol{c}_{ij}=(\boldsymbol{\nabla}\varphi_j,\varphi_i)=\sum_{K_e\in\mathcal{T}_h}\Big(&\boldsymbol{c}_{ij}^e+\sum_{k\in\mathcal{M}(j)}C_{kj}\boldsymbol{c}_{ik}^e\\&+\sum_{k\in\mathcal{M}(i)}C_{ki}\boldsymbol{c}_{kj}^e+\sum_{k\in\mathcal{M}(i)\cap\mathcal{M}(j)}C_{ki}C_{kj}\boldsymbol{c}_{kk}^e\Big),\end{aligned}$$

where $\boldsymbol{c}_{ij}^e=(\boldsymbol{\nabla}\varphi_j,\varphi_i)_{K_e}$. Thus,

$$\begin{aligned}\sum_{K_e\in\mathcal{T}_h}\Big(&\left|\mathbf{v}_{ij}\cdot\boldsymbol{c}_{ij}^e\right|+\sum_{k\in\mathcal{M}(j)}C_{kj}\left|\mathbf{v}_{ij}\cdot\boldsymbol{c}_{ik}^e\right|\\&+\sum_{k\in\mathcal{M}(i)}C_{ki}\left|\mathbf{v}_{ij}\cdot\boldsymbol{c}_{kj}^e\right|\\&+\sum_{k\in\mathcal{M}(i)\cap\mathcal{M}(j)}C_{ki}C_{kj}\left|\mathbf{v}_{ij}\cdot\boldsymbol{c}_{kk}^e\right|\Big)\geq\left|\mathbf{v}_{ij}\cdot\boldsymbol{c}_{ij}\right|,\end{aligned}$$

and

$$\begin{aligned}\sum_{K_e\in\mathcal{T}_h}\Big(&c_{ij}\|\boldsymbol{c}_{ij}^e\|+\sum_{k\in\mathcal{M}(j)}C_{kj}\left|\mathbf{v}_{ij}\cdot c_{ij}\right|\|\boldsymbol{c}_{ik}^e\|\\&+\sum_{k\in\mathcal{M}(i)}C_{ki}\left|\mathbf{v}_{ij}\cdot c_{ij}\right|\|\boldsymbol{c}_{kj}^e\|\\&+\sum_{k\in\mathcal{M}(i)\cap\mathcal{M}(j)}C_{ki}C_{kj}\left|\mathbf{v}_{ij}\cdot c_{ij}\right|\|\boldsymbol{c}_{kk}^e\|\Big)\geq c_{ij}\|\boldsymbol{c}_{ij}\|.\end{aligned}$$

Therefore, $\sum_e \rho(\mathbf{K}_{ij}^e(\boldsymbol{u}_{ij})) \geq \rho(\mathbf{K}_{ij}(\boldsymbol{u}_{ij}))$. Moreover, by definition (see (5.5)),

$$\begin{aligned}\nu_{ij}^e \geq\ & \rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij})\cdot(\boldsymbol{\nabla}\varphi_j,\varphi_i)_{K_e}\right)\\ &+\sum_{k\in\mathcal{M}(j)} C_{kj}\ \rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij})\cdot(\boldsymbol{\nabla}\varphi_k,\varphi_i)_{K_e}\right)\\ &+\sum_{k\in\mathcal{M}(i)} C_{ki}\ \rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij})\cdot(\boldsymbol{\nabla}\varphi_j,\varphi_k)_{K_e}\right)\\ &+\sum_{k\in\mathcal{M}(i)\cap\mathcal{M}(j)} C_{ki}C_{kj}\ \rho\left(\boldsymbol{f}'(\boldsymbol{u}_{ij})\cdot(\boldsymbol{\nabla}\varphi_k,\varphi_k)_{K_e}\right)\quad \text{for}\quad j\neq i.\end{aligned}$$

Furthermore, it is easy to infer from (5.4) that $\rho(\mathbf{B}_{ij}^e(\boldsymbol{u}_{ij})) \geq \rho(\mathbf{K}_{ij}^e(\boldsymbol{u}_{ij}))$. Hence, $\rho(\mathbf{B}_{ij}(\boldsymbol{u}_{ij})) \geq \rho(\mathbf{K}_{ij}(\boldsymbol{u}_{ij}))$. Finally, since $\overline{\mathbf{K}}_{ij} = \mathbf{K}_{ij}+\mathbf{B}_{ij}$ and $\mathbf{B}_{ij} = \sum_e \mathbf{B}_{ij}^e = \sum_e -\nu_{ij}^e I_{m\times m}$ for all $j \neq i$. Then, the maximum eigenvalue of $\overline{\mathbf{K}}_{ij}(\boldsymbol{u}_{ij})$ is non-positive, which completes the proof. □

Notice that it is essential to apply properly the constraints at the flux FE approximation, i.e., $\boldsymbol{f}'(\boldsymbol{u}_k) = \sum_{\{i\in\mathcal{N}_h\,:\,C_{ki}\neq 0\}} C_{ki}\boldsymbol{f}'(\boldsymbol{u}_i)$. Otherwise, it is not possible to formally prove local bound preservation. However, experimental results in the present chapter show that using $\boldsymbol{f}'(\boldsymbol{u}_k)$ does not affect the overall performance of the scheme.

5.3.1 Differentiable stabilization

In the case of steady, or implicit time integration, differentiability plays a role in the convergence behavior of the nonlinear solver. This is especially important if one wants to use Newton's method. In the previous chapters we showed that nonlinear convergence can be improved after few modifications to make the scheme twice-differentiable. In this section, we introduce a set of regularizations applied to all non-differentiable functions present in the stabilized scheme introduced above. In order to regularize these functions, we follow the same strategy as in the previous chapters. Absolute values are replaced by

$$|x|_{1,\varepsilon_h} = \sqrt{x^2+\varepsilon_h},\qquad |x|_{2,\varepsilon_h} = \frac{x^2}{\sqrt{x^2+\varepsilon_h}}.$$

Note that $|x|_{2,\varepsilon_h} \leq |x| \leq |x|_{1,\varepsilon_h}$. Next, we also use the smooth maximum function

$$\max{}_{\sigma_h}(x,y) \doteq \frac{|x-y|_{1,\sigma_h}}{2}+\frac{x+y}{2} \geq \max(x,y).$$

In addition, we need a smooth function to limit the value of any given quantity to one. To this end, we use

$$Z(x) \doteq \begin{cases} 2x^4-5x^3+3x^2+x, & x<1,\\ 1, & x\geq 1.\end{cases}$$

The set of twice-differentiable functions defined above allows us to redefine the stabilization term introduced in Sect. 5.3. In particular, we define

$$\tilde{B}_h(\boldsymbol{w}_h;\boldsymbol{u}_h,\boldsymbol{v}_h) \doteq \begin{cases} \sum_{i\in\mathcal{N}_h}\sum_{j\in\mathcal{N}_h(\Omega_i)} \tilde{\nu}_{ij}(w_h)v_iu_j\ell(i,j), & \text{for } m=1,\\ \sum_{K_e\in\mathcal{T}_h}\sum_{\substack{i,j\in\mathcal{N}_h(K_e)\\ 1\le\beta,\gamma\le m}} \tilde{\nu}^e_{ij}(\boldsymbol{w}_h)\ell(i,j)\, v_i^\beta\cdot\delta_{\beta\gamma}u_j^\gamma, & \text{for } m>1, \end{cases} \tag{5.11}$$

where

$$\begin{aligned} \tilde{\nu}_{ij}(w_h) &\doteq \max\nolimits_{\sigma_h}\{\alpha_{\varepsilon_h,i}(w_h)\mathbf{K}_{ij},0,\alpha_{\varepsilon_h,j}(w_h)\mathbf{K}_{ji}\} \quad \text{for} \quad j\in\mathcal{N}_h(\Omega_i)\backslash\{i\},\\ \tilde{\nu}_{ii}(w_h) &\doteq \sum_{j\in\mathcal{N}_h(\Omega_i)\backslash\{i\}} \tilde{\nu}_{ij}(w_h), \end{aligned}$$

and

$$\begin{aligned} \tilde{\nu}^e_{ij}(\boldsymbol{w}_h) \doteq{}& \max\nolimits_{\sigma_h}\left(\boldsymbol{\alpha}_{\varepsilon_h,i}(\boldsymbol{w}_h)\lambda^{\max}_{ij},\boldsymbol{\alpha}_{\varepsilon_h,j}(\boldsymbol{w}_h)\lambda^{\max}_{ji}\right)\\ &+\sum_{k\in\mathcal{M}(i)} C_{ki}\max\nolimits_{\sigma_h}\left(\boldsymbol{\alpha}_{\varepsilon_h,k}(\boldsymbol{w}_h)\lambda^{\max}_{kj},\boldsymbol{\alpha}_{\varepsilon_h,j}(\boldsymbol{w}_h)\lambda^{\max}_{jk}\right)\\ &+\sum_{k\in\mathcal{M}(j)} C_{kj}\max\nolimits_{\sigma_h}\left(\boldsymbol{\alpha}_{\varepsilon_h,i}(\boldsymbol{w}_h)\lambda^{\max}_{ik},\boldsymbol{\alpha}_{\varepsilon_h,k}(\boldsymbol{w}_h)\lambda^{\max}_{ki}\right)\\ &+\sum_{k\in\mathcal{M}(i)\cap\mathcal{M}(j)} C_{ki}C_{kj}\boldsymbol{\alpha}_{\varepsilon_h,k}(\boldsymbol{w}_h)\lambda^{\max}_{kk}, \quad \text{for } j\in\mathcal{N}_h(\Omega_i)\backslash\{i\},\\ \tilde{\nu}^e_{ii}(\boldsymbol{w}_h) \doteq{}& \sum_{j\in\mathcal{N}_h(\Omega_i)\backslash\{i\}} \tilde{\nu}^e_{ij}(\boldsymbol{w}_h). \end{aligned}$$

Let us note that $\lambda^{\max}_{ij}$ needs to be regularized as $\lambda^{\max}_{ij} = \left|\mathbf{v}_{ij}\boldsymbol{c}^e_{ij}\right|_{1,\varepsilon_h} + c\|\boldsymbol{c}^e_{ij}\|$. The shock detector is also regularized as follows:

$$\boldsymbol{\alpha}_{\varepsilon_h,i}(\boldsymbol{u}_h) \doteq \max\nolimits_{\sigma_h}\{\alpha_{\varepsilon_h,i}(u_h^\beta)\}_{\beta\in C}.$$

In the case of the component shock detector we recall the definition in Chapter 2

$$\alpha_{\varepsilon_h,i}(u_h^\beta) \doteq \left[Z\left(\frac{\left|\sum_{j\in\mathcal{N}_h(\Omega_i)}\left[\!\left[\boldsymbol{\nabla}u_h^\beta\right]\!\right]_{ij}\right|_{1,\varepsilon_h} + \zeta_h}{\sum_{j\in\mathcal{N}_h(\Omega_i)} 2\left\{\!\!\left\{\left|\boldsymbol{\nabla}u_h^\beta\cdot\hat{\boldsymbol{r}}_{ij}\right|_{2,\varepsilon_h}\right\}\!\!\right\}_{ij} + \zeta_h} \right)\right]^q, \tag{5.12}$$

where ζ_h is a small value for preventing division by zero. Finally, the twice-differentiable stabilized scheme reads: find $\boldsymbol{u}_h\in\boldsymbol{V}_h$ such that $\boldsymbol{u}_h=\overline{\boldsymbol{u}}_h$ on $\partial\Omega$, $\boldsymbol{u}_h=\boldsymbol{u}_{0h}$ at $t=0$, and

$$\tilde{\mathbf{M}}(\boldsymbol{u}_h^{n+1})\delta_t\mathbf{U}^{n+1} + \tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{n+1})\mathbf{U}^{n+1} = \mathbf{G} \quad \text{for } n=1,...,n^{ts}, \tag{5.13}$$

where

$$\tilde{\mathbf{M}}_{ij}(\boldsymbol{u}_h^{n+1}) \doteq [1 - \max{}_{\sigma_h}(\boldsymbol{\alpha}_{\varepsilon_h,i}, \boldsymbol{\alpha}_{\varepsilon_h,j})]\,\mathbf{M}_{ij} + \max{}_{\sigma_h}(\boldsymbol{\alpha}_{\varepsilon_h,i}, \boldsymbol{\alpha}_{\varepsilon_h,j})\,\delta_{ij}\sum_j \mathbf{M}_{ij},$$

$$\tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{n+1}) \doteq \mathbf{K}_{ij}(\boldsymbol{u}_h^{n+1}) + \tilde{\mathbf{B}}_{ij}(\boldsymbol{u}_h^{n+1}),$$

and $\tilde{\mathbf{B}}_{ij}(\boldsymbol{u}_h) = \tilde{B}_h(\boldsymbol{u}_h; \varphi_j, \varphi_i)$, for $i, j \in \mathcal{N}_h$.

Corollary 5.3.3. *The scheme in* (5.3) *with the differentiable stabilization in* (5.11) *is local bounds preserving, as defined in Def. 5.2.4, at any region where* u_h^β *has extreme values for every* β *in* C.

Proof. For an extreme value of u_h^β, since $|x|_{2,\varepsilon_h} \leq |x| \leq |x|_{1,\varepsilon_h}$ the quotient of (5.12) is larger than one. Hence, by definition of $Z(x)$, $\alpha_{\varepsilon_h,i}$ is equal to 1. At this point, it is easy to check that $\tilde{\nu}_{ij}^e \geq \nu_{ij}^e$ in virtue of the definition of $\max_{\sigma_h}$. Therefore, $\rho(\tilde{\mathbf{B}}_{ij}^e(\boldsymbol{u}_h)) \geq \rho(\mathbf{B}_{ij}^e(\boldsymbol{u}_h))$, completing the proof. □

Moreover, it is important to mention that the differentiable shock detector is weakly linearly-preserving as ζ_h tends to zero. This result follows directly from Sect. 2.7. In order to obtain a differentiable operator, we have added a set of regularizations that rely on different parameters, e.g., σ_h, ε_h, ζ_h. Giving a proper scaling of these parameters is essential to recover theoretic convergence rates. In particular, we use the following relations

$$\sigma_h = \sigma|\lambda^{\max}|^2 L^{2(d-3)} h^4, \quad \varepsilon_h = \varepsilon L^{-4} h^2, \quad \zeta_h = L^{-1}\zeta,$$

where for the scalar problem $\lambda^{\max}$ is simply $\|\mathbf{v}\|$, d is the spatial dimension of the problem, and L is a characteristic length.

5.4 Adaptive mesh refinement

The motivation of an adaptive FE method is to solve (5.10) up to a certain tolerance (or resolution) using the *minimum* number of DOFs. To this end, the solution error ($\boldsymbol{e}_h = \boldsymbol{u} - \boldsymbol{u}_h$) is estimated at each element. With this information at hand, it is possible to iteratively adapt the resolution of the mesh at certain regions. This process can be divided into two parts: estimating the error at every cell, and deciding which and how many cells need to be refined or coarsened. This procedure is performed iteratively until a desired tolerance is achieved or, alternatively, a number of elements is reached. In the present chapter, we start with a rather coarse mesh and perform the following steps till reaching a stopping criterion:

1. Compute solution $\boldsymbol{u}_h$;
2. Estimate the error $\boldsymbol{e}_h$;
3. Select all cells that need to be refinement or coarsened;

4. Update the mesh, and project the solution to the new mesh.

In some cases, the refinement might be driven by features of the solution instead of a classical error estimator. For instance, one may decide to refine the regions around discontinuities. In this scenario, one could use a expression that does not estimate the error, but it allows to concentrate the elements around discontinuities.

5.4.1 Error estimators

One of the keys of AMR is the ability to provide a good estimation of the error. Several error estimators have been proposed to date [1, 33, 50, 52, 94]. These can be classified, at least, in two main types. Some authors [33, 50, 78, 79, 87] try to compute an upper bound of the error for every cell. Then, provided a user defined tolerance, one can decide to refine or coarsen each cell. However, an adjoint problem needs to be solved in order to compute this upper bound [22, 50]. It is possible to approximate the error bounds without solving an adjoint problem only for simple cases, see [50]. Therefore, this kind of error estimators increases the computational cost substantially. Alternatively, one can simply determine the distribution of the error in the mesh and use this information to drive an adaptivity algorithm. In this scenario, some authors [1, 15, 63, 77, 94, 95] drive the adaptivity process with the solution gradient. In this case, explicit expressions of the estimated error are possible, requiring less computational resources than the previous option.

In general, the adaptive procedure can be described as follows. Given a finite element solution $\boldsymbol{u}_h$, the error $\boldsymbol{e}_h$ is approximated as $\boldsymbol{e}_h \approx \boldsymbol{\nabla u} - \boldsymbol{\nabla u}_h$. Then, the reconstruction is used as an approximation of the exact gradient. This strategy is based on *super-convergence* of special recovery techniques (see [95] and refs. in [1, 77]). Kuzmin and co-workers [15, 77] follow [94] to reconstruct an approximation of the exact gradient. Kelly et al. [52] proposed a well-known estimator based on gradient recovery:

$$\eta_K^2 \doteq \frac{h_K}{24} \int_{\partial K} \left[\!\!\left[\frac{\partial \boldsymbol{u}_h}{\partial n} \right]\!\!\right]^2 d\Gamma,$$

where η_K is the estimated error at every element, K. The main advantages of this estimator are its simplicity and its low computational cost. For these reasons, this estimator is used in the present chapter.

It is worth mentioning that our problems of interest are characterized by exhibiting discontinuities, where the error concentrates. These regions are susceptible to develop instabilities, and thus, these are the regions in which the shock capturing is activated. Therefore, it is natural to use the shock capturing to drive the adaptivity procedure. We propose an estimator based on the graph Laplacian $\ell(i,j)$ present in the stabilization term (5.4). This way, we reuse available information and reduce the computational

overhead associated with error estimation. The indicator reads:

$$\tilde{\eta}_K^2 \doteq h_K^{d-2}\ell(\boldsymbol{u}_h^\beta, \boldsymbol{u}_h^\beta)_K = h_K^{d-2}\sum_{i\in\mathcal{N}_h(K)}\sum_{j\in\mathcal{N}_h(\Omega_i)}(\boldsymbol{u}_i^\beta - \boldsymbol{u}_j^\beta)^2,$$

where $\beta \in C$ is the index of the specific component analyzed. This estimator is expected to yield high values around shocks and low values in smooth regions.

5.4.2 Refinement strategy

After the error has been estimated for every element, one needs to decide which element needs to be refined and which one coarsened. If an upper bound of the error is computed, then one may use a given tolerance to make this decision. However, in the present case this is not available. A classical alternative is to refine/coarsen a fixed amount of elements at every iteration [10, 12]. In the present study, a 30% of the elements with higher error estimates are refined whereas a 10% of the elements with lower error estimates are coarsened. This percentages are arbitrary and other choices are valid. Notice that using this setting in two dimensions the number of elements is almost doubled at every iteration. We make use of the parallel nth element algorithm [10, 90] to efficiently determine the error estimator thresholds for refining or coarsening the elements.

5.5 Nonlinear solver

In this section, we describe the method used for solving the nonlinear system of equations arising from the scheme introduced above. In particular, we use a hybrid Picard–Newton approach in order to increase the robustness of the nonlinear solver. Moreover, we also make use of a line-search method to improve the nonlinear convergence.

We define the residual of the equation (5.13) at the k-th iteration as

$$\mathbf{R}(\boldsymbol{u}_h^{k,n+1}) \doteq \tilde{\mathbf{M}}(\boldsymbol{u}_h^{k,n+1})\delta_t\mathbf{U}^{k,n+1} + \tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{k,n+1})\mathbf{U}^{k,n+1} - \mathbf{G}.$$

Hence, the Jacobian is defined as

$$\begin{aligned}\mathbf{J}(\boldsymbol{u}_h^{k,n+1}) &\doteq \frac{\partial\mathbf{R}(\boldsymbol{u}_h^{k,n+1})}{\partial\mathbf{U}^{k,n+1}} \\ &= \tilde{\mathbf{M}}(\boldsymbol{u}_h^{k,n+1}) + \tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{k,n+1}) + \frac{\partial\tilde{\mathbf{M}}(\boldsymbol{u}_h^{k,n+1})}{\partial\mathbf{U}^{k,n+1}}\delta_t\mathbf{U}^{k,n+1} + \frac{\partial\tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{k,n+1})}{\partial\mathbf{U}^{k,n+1}}\mathbf{U}^{k,n+1}.\end{aligned} \tag{5.14}$$

Therefore, Newton method consists in solving $\mathbf{J}(\boldsymbol{u}_h^{k,n+1})\Delta\mathbf{U}^{k+1,n+1} = -\mathbf{R}(\boldsymbol{u}_h^{k,n+1})$. It is well known that Newton method can diverge if the initial guess of the solution $\boldsymbol{u}_h^{0,n+1}$ is not close enough to the solution. In order to improve robustness, we use a line-search method to update the solution at every time step. The new approximation is computed as $\mathbf{U}^{k+1,n+1} = \mathbf{U}^{k,n+1} + \lambda\Delta\mathbf{U}^{k+1,n+1}$, where λ is obtained using a standard cubic backtracking algorithm.

As introduced at the beginning of the section, we also use a hybrid approach combining Newton method with Picard linearization. Picard nonlinear iterator can be obtained removing the last two terms of (5.14), i.e.,

$$\left(\tilde{\mathbf{M}}(\boldsymbol{u}_h^{k,n+1}) + \tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{k,n+1})\right) \Delta \mathbf{U}^{k+1,n+1} = -\mathbf{R}(\boldsymbol{u}_h^{k,n+1}). \tag{5.15}$$

Clearly, it is equivalent to

$$\left(\tilde{\mathbf{M}}(\boldsymbol{u}_h^{k,n+1}) + \tilde{\mathbf{K}}_{ij}(\boldsymbol{u}_h^{k,n+1})\right) \mathbf{U}^{k+1,n+1} = \tilde{\mathbf{M}}(\boldsymbol{u}_h^{k,n+1})\mathbf{U}^n + \mathbf{G}.$$

Moreover, we modify the left hand side terms in (5.15); we use $\boldsymbol{\alpha}_i = 1$ for computing these terms while we use the value obtained from (5.8) for the residual. Using this strategy, the solution remains unaltered but the obtained approximations $\boldsymbol{u}_h^{k,n+1}$ for intermediate values of k are more diffusive. Even though this modification slows the nonlinear convergence, it is essential at the first iterations. Otherwise, the robustness of the method might be jeopardized.

The resulting iterative nonlinear solver consists in the following steps. We iterate Picard method in (5.15), with the modification described above, until the L^2 norm of the residual is smaller than a given tolerance. In the present chapter, we use a tolerance of 10^{-2}. Afterwards, Newton method with the exact Jacobian in (5.14) is used until the desired nonlinear convergence criteria is satisfied. We summarize the nonlinear solver introduced above in Alg. 4.

Algorithm 4: Hybrid Picard–Newton method.

Input: $\mathbf{U}^{0,n+1}$, tol_1, tol_2, ε
Output: $\mathbf{U}^{k,n+1}$, k
$k = 1$, $\varepsilon^1 = \varepsilon$
while $\|\mathbf{R}(\mathbf{U}^{k,n+1})\|/\|\mathbf{R}(\mathbf{U}^{0,n+1})\| \geq \text{tol}_1$ **do**
  Compute $\boldsymbol{\alpha}_i(\mathbf{U}^{k,n+1})$ using (5.8)
  Compute $\Delta\mathbf{U}^{k+1,n+1}$ using (5.15)
  Minimize $\|\mathbf{R}(\mathbf{U}^{k+1,n+1})\|$, where $\mathbf{U}^{k+1,n+1} = \lambda\Delta\mathbf{U}^{k+1,n+1} + \mathbf{U}^{k,n+1}$, with respect to λ
  Set $\mathbf{U}^{k+1,n+1} = \lambda\Delta\mathbf{U}^{k+1,n+1} + \mathbf{U}^{k,n+1}$
  Update $k = k + 1$
while $\|\mathbf{R}(\mathbf{U}^{k,n+1})\|/\|\mathbf{R}(\mathbf{U}^{0,n+1})\| \geq \text{tol}_2$ **do**
  Compute $\boldsymbol{\alpha}_i(\mathbf{U}^{k,n+1})$ using (5.8)
  Solve $\mathbf{J}(\mathbf{U}^{k,n+1})\Delta\mathbf{U}^{k+1,n+1} = -\mathbf{R}(\mathbf{U}^{k,n+1})$ with $\mathbf{J}$ in (5.14)
  Minimize $\|\mathbf{R}(\mathbf{U}^{k+1,n+1})\|$, where $\mathbf{U}^{k+1,n+1} = \lambda\Delta\mathbf{U}^{k,n+1} + \mathbf{U}^{k,n+1}$, with respect to λ
  Set $\mathbf{U}^{k+1,n+1} = \lambda\Delta\mathbf{U}^{k,n+1} + \mathbf{U}^{k,n+1}$
  Update $k = k + 1$

5.6 Numerical results

In this section, we perform several numerical experiments to assess the numerical scheme introduced in the previous sections. First, we perform a convergence analysis to assess its implementation. Then, we use steady benchmark tests to analyze the effectiveness of the high-order scheme in the context of AMR. In particular, we compare the nonlinear scheme in (5.13) with its linear (first order) counterpart, i.e., using $\boldsymbol{\alpha}_{\varepsilon_h,i}(\boldsymbol{u}_h) \equiv 1$.

From the experience in the numerical experiments of the previous chapters, we choose the following regularization parameters: $\sigma = 10^{-2}$, $\varepsilon = 10^{-4}$, and $\gamma = 10^{-10}$. In addition, for all Euler tests below, the density is discontinuous at all shocks. Therefore, we use $C = \{1\}$ in (5.8), i.e., the shock detector is based on the density behavior.

5.6.1 Convergence

First, the convergence to a discontinuous solution is analyzed. To this end, we solve two different problems. On the one hand, the following scalar problem is solved

$$\begin{aligned} \boldsymbol{\nabla} \cdot (\mathbf{v}u) &= 0 \quad \text{in} \quad \Omega = [0,1] \times [0,1], \\ u &= u_D \quad \text{on} \quad \Gamma_{\text{in}}, \end{aligned} \tag{5.16}$$

where $\mathbf{v}(x,y) \doteq (1/2, \sin -\pi/3)$, and inflow boundary conditions $u_D = 1$ on $\{x = 0\} \cap \{y > 0.7\}$ and $y = 1$, while $u_D = 0$ at the rest of the inflow boundary. This problem has the following analytical solution

$$u(x,y) = \begin{cases} 1 & \text{if} \quad y > 0.7 + 2x \sin -\pi/3, \\ 0 & \text{otherwise.} \end{cases}$$

For the Euler equations, the problem is the well known compression corner test [3, 61], also known as oblique shock test [84, 88]. This benchmark consists in a supersonic flow impinging to a wall at an angle. We use a $[0,1]^2$ domain with a $M = 2$ flow at 10° with respect to the wall. This leads to two flow regions separated by an oblique shock at 29.3°, see Fig. 5.3.

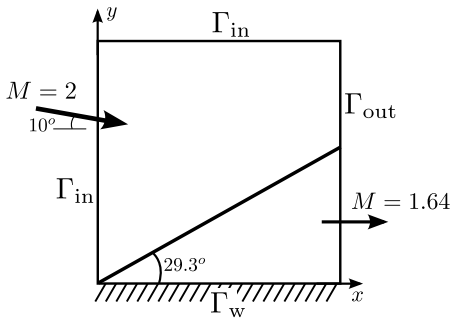
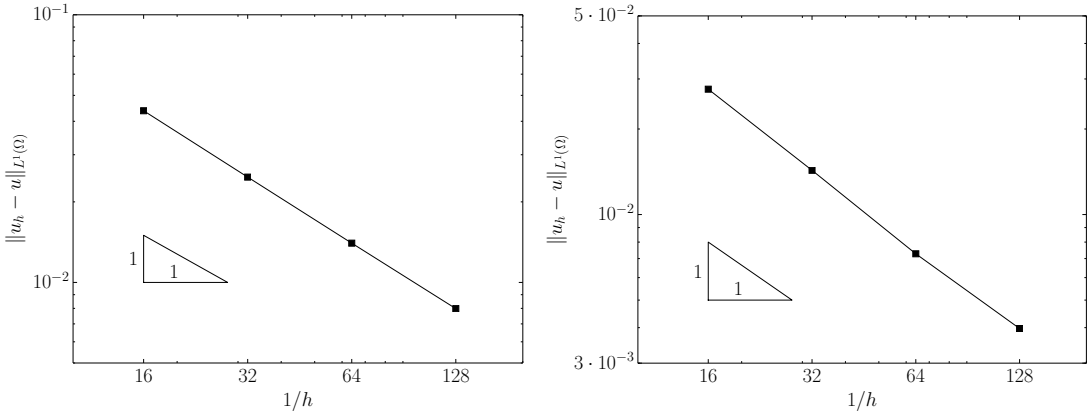


FIGURE 5.3: Compression corner scheme.

Since the solution is not smooth, we expect linear convergence rates in the L^1-norm. Fig. 5.4 shows the convergence behavior of both problems with uniform mesh refinements.



(A) Scalar transport problem. (B) Euler equations.

FIGURE 5.4: Convergence of $\|u - u_h\|_{L^1(\Omega)}$ to a solution with a discontinuity.

The experimental convergence rate measured for the scalar transport problem is 0.82, whereas the convergence rate measured is 0.94 for the compression corner test. Therefore, both tests exhibit the expected convergence behavior.

5.6.2 Linear discontinuity

For this test, we use again the problem in (5.16). The purpose of this test is twofold. On the one hand, we analyze the effectiveness of the proposed estimator. On the other hand, we compare the effectiveness of the linear and nonlinear stabilization methods. Specifically, this effectiveness is measured as follows. For a given error, we consider a method more effective if it requires less computational time, independently of the number of elements required. In addition, we also solve the problem for successive uniformly refined meshes in order to evaluate the effect of AMR.

For all comparisons, we start with a coarse mesh of 16×16 elements, and proceed adapting the mesh up to a maximum number of elements. For the nonlinear stabilization, we set a maximum of 10^4 elements. The maximum number of elements for the low-order method is 10^5. The uniform mesh is refined up to a 256×256 mesh. We use a nonlinear tolerance of $\|\Delta \boldsymbol{u}_h\|/\|\boldsymbol{u}_h\| < 10^{-4}$, and a maximum of 500 iterations.

Fig. 5.5 shows the evolution of the AMR algorithm for both estimators. The results shown in this picture have been obtained using the linear stabilization, and the leftmost column using the nonlinear one. It can be observed that both Kelly (η_K) and graph Laplacian ($\tilde{\eta}_K$) estimators refine in the vicinity of the shock. However, the graph Laplacian operator clearly outperforms Kelly estimator.

Figs. 5.6–5.8 compare the effectiveness of the low-order and the high-order stabilization schemes. The results are obtained for the stabilization parameter $q = 1$, $q = 2$, and $q = 10$, respectively.

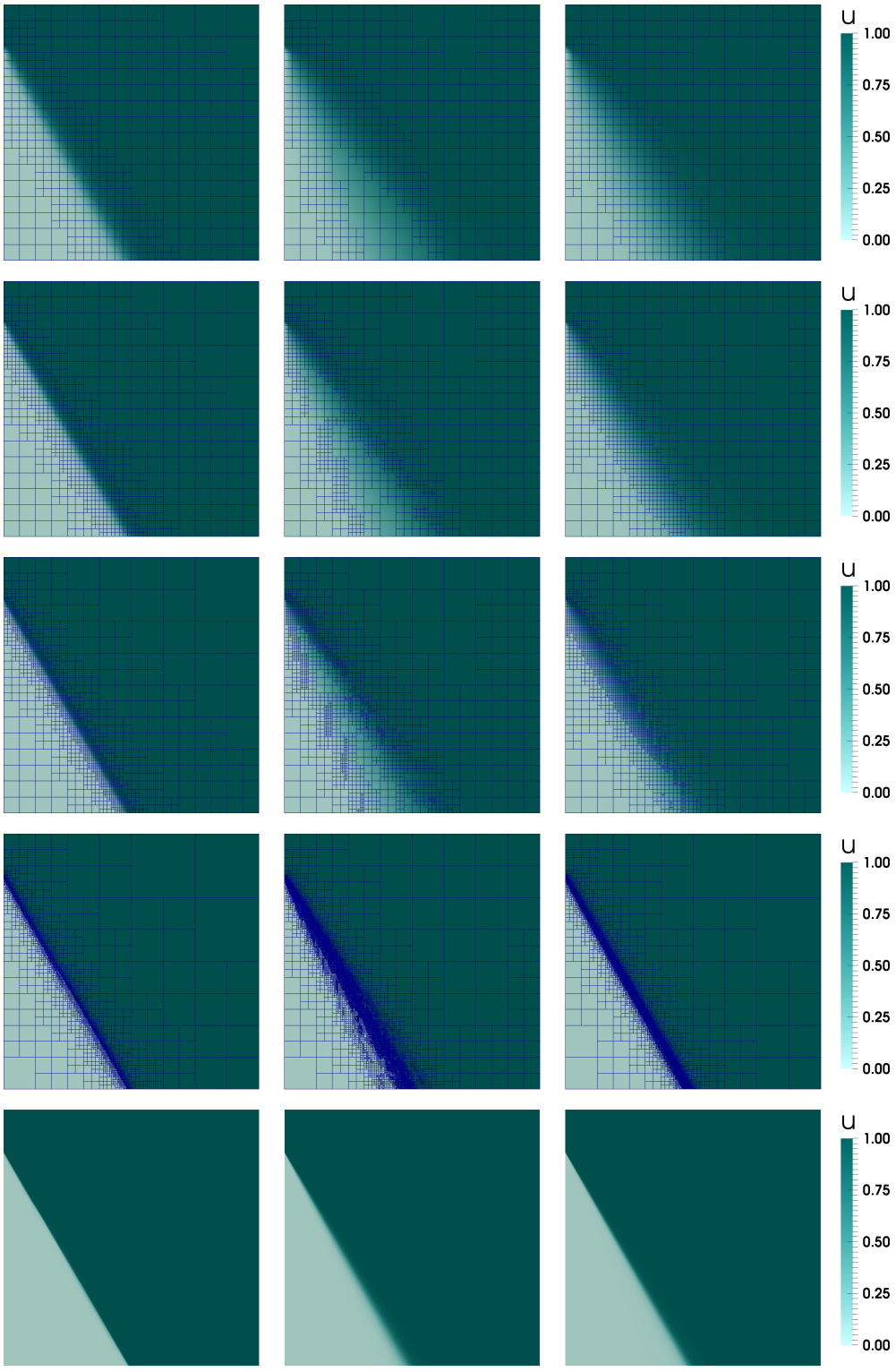


FIGURE 5.5: Evolution of the mesh refinement process. $\tilde{\eta}_K$ with high-order scheme is used in the left column. Low-order scheme with Kelly estimator is used in the central column. $\tilde{\eta}_K$ with low-order scheme is used in the right column. For the low-order scheme from top to bottom results have been obtained at refinement step 1, 2, 3, 9, and 9. For the high-order with $q = 10$, the refinement steps are 1, 2, 3, 5, and 5.

At Fig. 5.7, the nonlinear stabilization is able to converge the nonlinear problem efficiently and the overhead of solving a nonlinear problem does not strongly affect the overall performance. We note that for the linear scheme the problem is linear. It can be observed that the convergence rate (against time) is much higher for the nonlinear scheme. The linear scheme requires less computational time for coarser meshes but the nonlinear scheme is more effective for tighter accuracies.

We can observe in Fig. 5.8 the convergence problems of the nonlinear stabilization at some steps of the refinement procedure. Even though using $q = 10$ improves the accuracy of the method, it also increases the computational cost since the nonlinear problem is harder to solve. This results in an efficiency slightly lower than the linear stabilization.

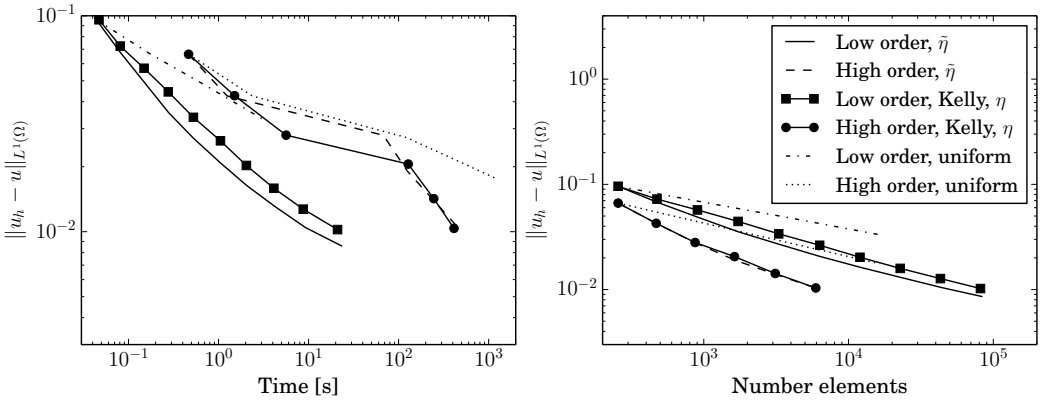


FIGURE 5.6: Time and elements convergence comparison for the transport problem with a linear discontinuity, $q = 1$.

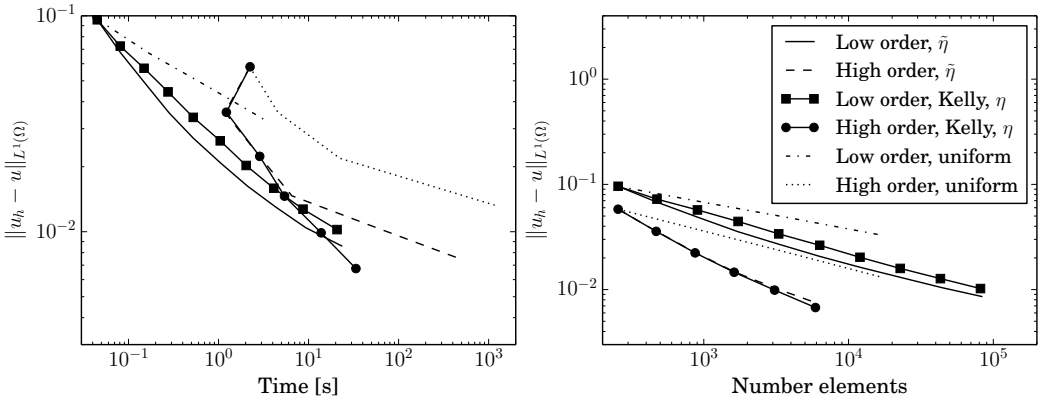


FIGURE 5.7: Time and elements convergence comparison for the transport problem with a linear discontinuity, $q = 2$.

5.6.3 Circular discontinuity

We analyze again the effectiveness of the proposed estimator and the effectiveness of the linear and nonlinear stabilization methods for a slightly more complicated convective

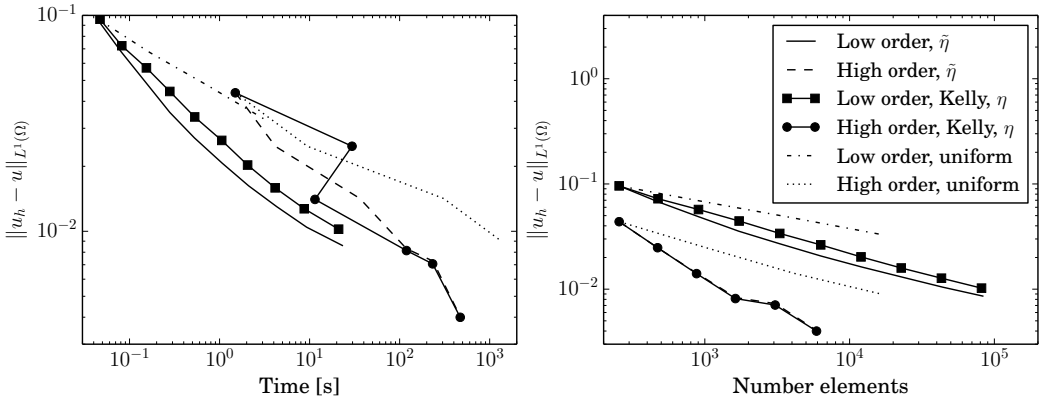


FIGURE 5.8: Time and elements convergence comparison for the transport problem with a linear discontinuity, $q = 10$.

field. For this test, we use (5.16) with $\mathbf{v}(x, y) \doteq (y, -x)$, and inflow boundary conditions

$$\overline{u}(0, y) = \begin{cases} 1 & y \in [0.15, 0.45], \\ \cos^2\left(\frac{10}{3}\pi(y - 0.4)\right) & y \in [0.55, 0.85], \\ 0 & \text{elsewhere.} \end{cases}$$

The analytical solution of this particular configuration consists in the transport of the inflow profile in the direction of the convection. As a result, the solution at the outflow boundary, corresponding to $y = 0$, is $u(x, 0) = \overline{u}(0, x)$. We start with a coarse mesh of 16×16 elements in all cases, and proceed adapting the mesh up to a maximum number of elements. For the nonlinear stabilization, we set a maximum of $5 \cdot 10^3$ elements. The maximum number of elements for the linear stabilization is $5 \cdot 10^4$. We use a nonlinear tolerance of $\|\Delta \boldsymbol{u}_h\|/\|\boldsymbol{u}_h\| < 10^{-4}$, and a maximum of 500 iterations.

Fig. 5.9 shows the evolution of the AMR algorithm for both η_K and $\tilde{\eta}_K$ error estimators with the linear stabilization and $\tilde{\eta}_K$ with the nonlinear one. It can be observed that both Kelly (η_K) and graph Laplacian indicators detect the regions that require more resolution. In any case, as in the previous example, the graph Laplacian operator ($\tilde{\eta}_K$) performs slightly better.

Figs. 5.10–5.12 compare the effectiveness of the linear and nonlinear stabilization. These results use the stabilization parameter $q = 1$, $q = 2$, and $q = 10$, respectively. In Fig. 5.11, the high-order scheme is able to converge efficiently and the overhead of solving a nonlinear problem does not strongly affect the overall performance. Nevertheless, the low-order scheme requires less computational time for any given error. However, it can be observed that the convergence rate (in time) is much higher for the high-order scheme. Therefore, it is expected to outperform the low-order scheme for more refined meshes.

In contrast, we do not observe the significant convergence problems in Fig. 5.12 even though the linear stabilization is slightly more efficient. Again, the convergence rate (versus time) is higher for the nonlinear stabilization and it will be better for more refined meshes.

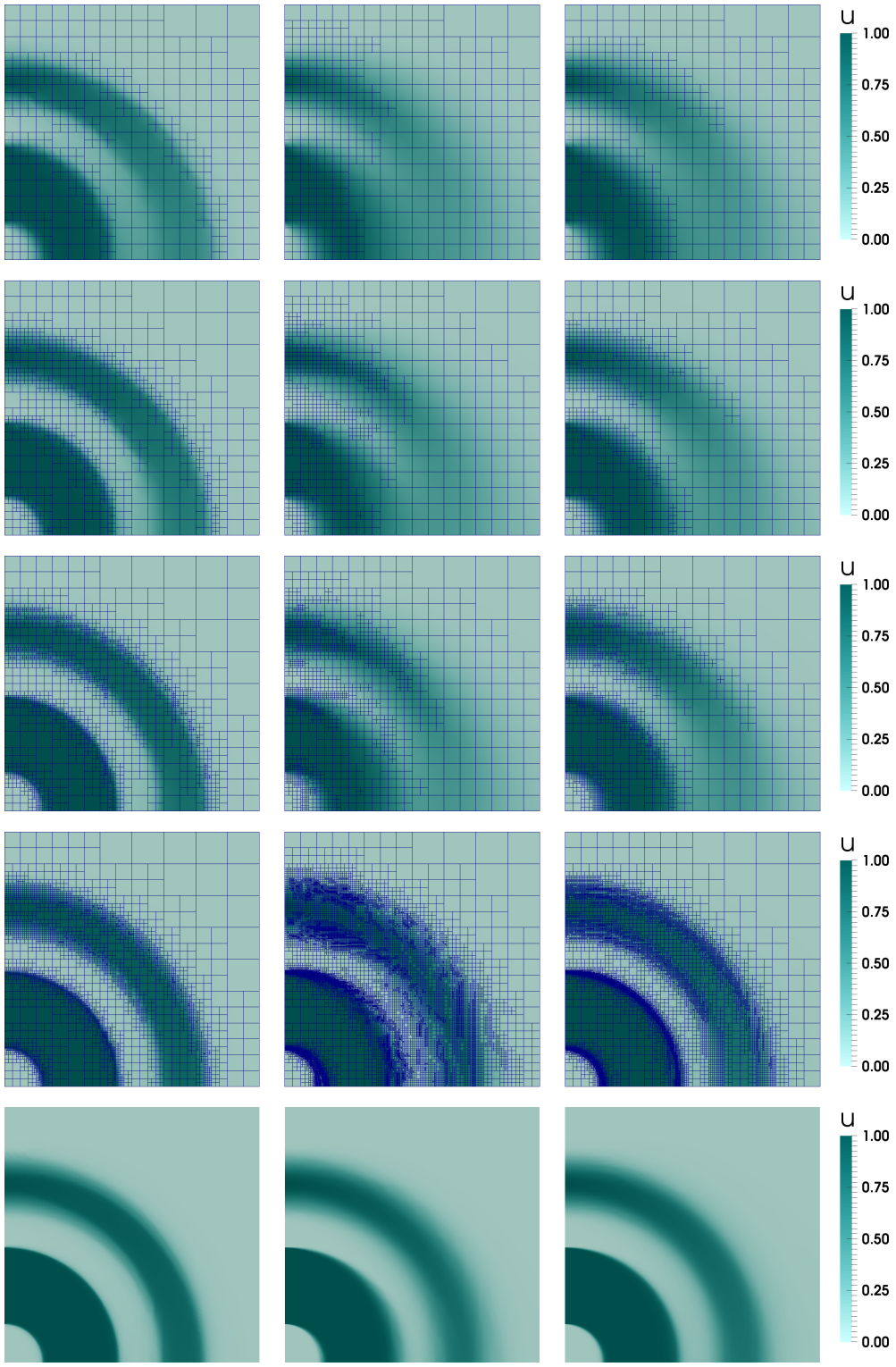


FIGURE 5.9: Evolution of the mesh refinement process. $\tilde{\eta}_K$ with high-order scheme is used in the left column. Low-order scheme with Kelly estimator is used in the central column. $\tilde{\eta}_K$ with low-order scheme is used in the right column. For the low-order scheme from top to bottom results have been obtained at refinement step 1, 2, 3, 7, and 7. For the high-order with $q = 10$, the refinement steps are 1, 2, 3, 4, and 4.

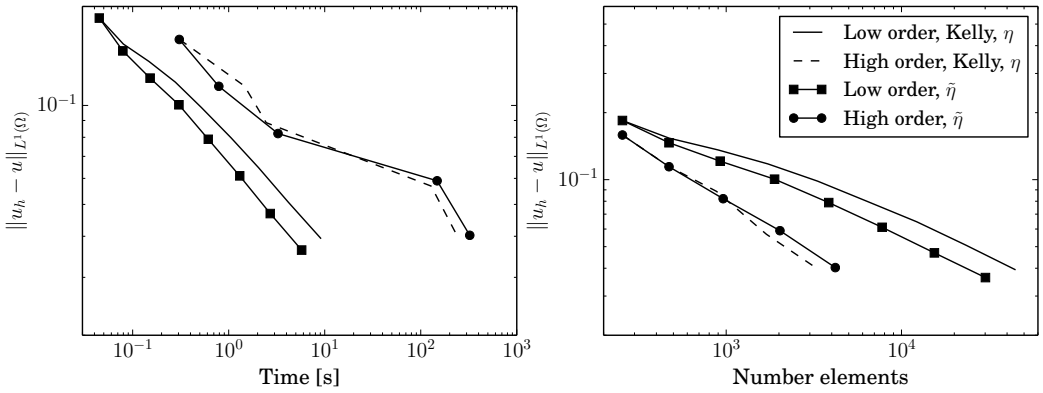


FIGURE 5.10: Time and elements convergence comparison for the transport problem with a circular convection field, $q = 1$.

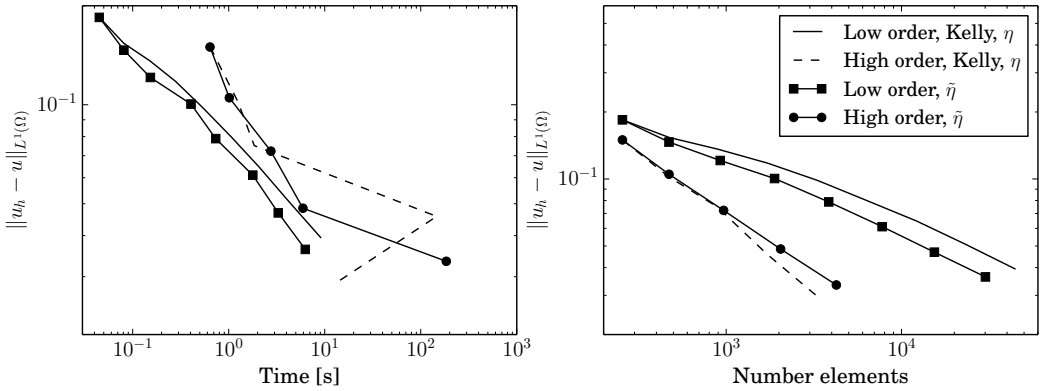


FIGURE 5.11: Time and elements convergence comparison for the transport problem with a circular convection field, $q = 2$.

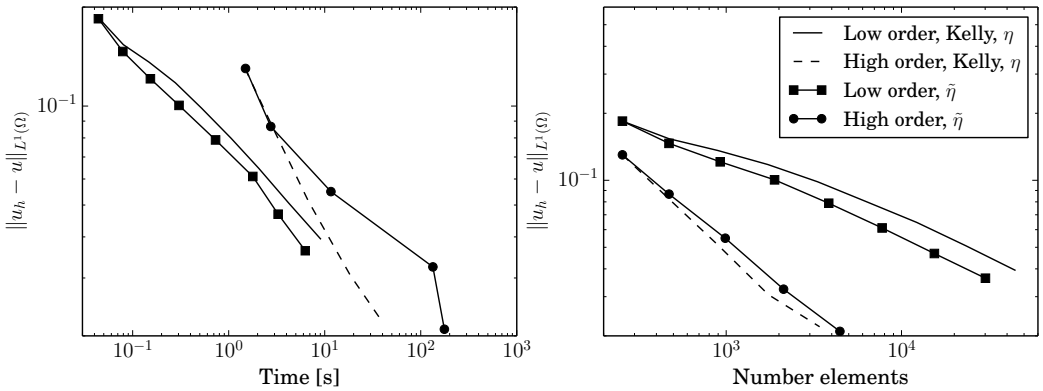


FIGURE 5.12: Time and elements convergence comparison for the transport problem with a circular convection field, $q = 10$.

5.6.4 Compression corner

Let us consider now the Euler equations. We start with the compression corner test (see Fig. 5.3). We analyze the effectiveness of the high-order scheme, and evaluate the

performance of the graph Laplacian estimator. We start with a coarse mesh of 16×16 elements, and adapt it up to a maximum number of elements. For the high-order method, we set a maximum of $5 \cdot 10^3$ elements. The maximum number of elements for the low-order method is $5 \cdot 10^4$. We use a nonlinear tolerance of $\|\Delta \boldsymbol{u}_h\|/\|\boldsymbol{u}_h\| < 10^{-4}$ and a maximum of 500 iterations.

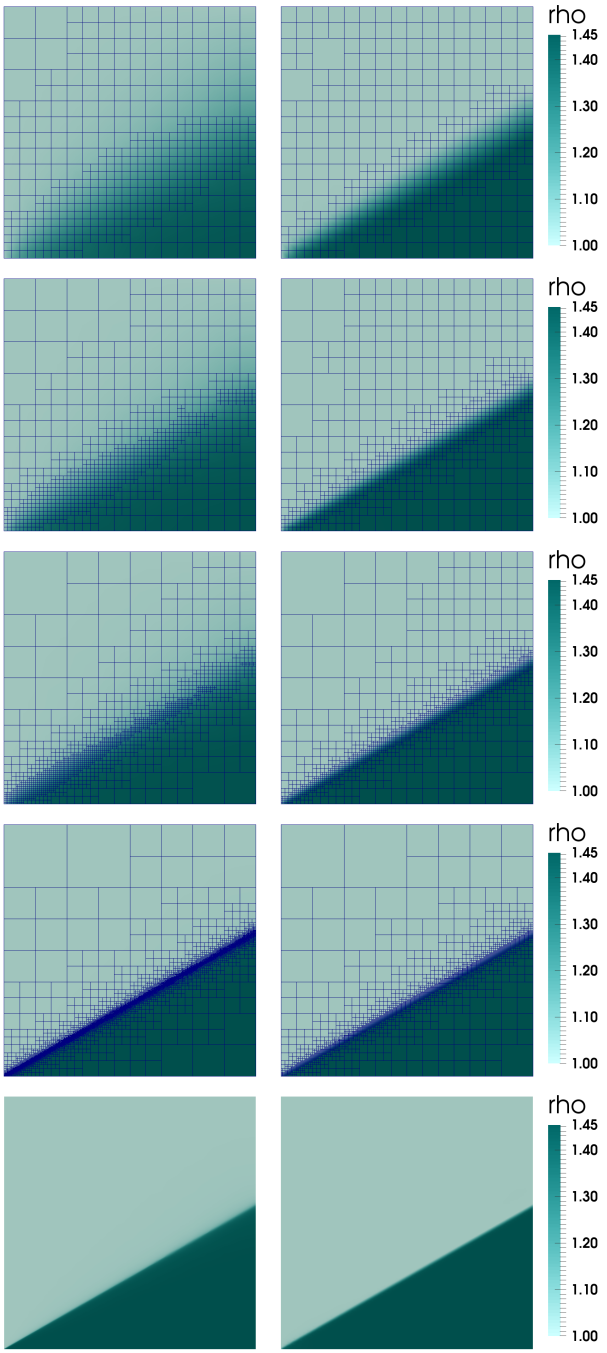


FIGURE 5.13: Evolution of the mesh refinement process. $\tilde{\eta}_K$ with high-order (right) and low-order (left) schemes are used. For the low-order scheme from top to bottom results have been obtained at refinement step 1, 2, 3, 8, and 8. For the high-order with $q = 10$, the refinement steps are 1, 2, 3, 4, and 4.

In Fig. 5.13, we depict the refinement evolution for the graph Laplacian estimator ($\tilde{\eta}_K$) for linear and nonlinear stabilization. As expected, we can observe that for the high-order method the scheme is able to resolve the shock with less refinement steps. The linear stabilization is able to provide well-resolved shocks at the final refinement step.

Fig. 5.14 compares the effectiveness of the low-order and the high-order stabilization schemes for different values of q. The high-order scheme is able to converge efficiently and the overhead of solving a nonlinear problem does not affect the overall performance. In this case, the low-order and the high-order schemes require similar computational time for any given error. Actually, for the finer meshes, the high-order scheme with either $q = 1$ or $q = 2$ already performs better than the low-order scheme. However, for some meshes the high-order scheme exhibits convergence problems. In the case of $q = 10$, the cost of converging the nonlinear problem does not compensate the increase in computational cost.

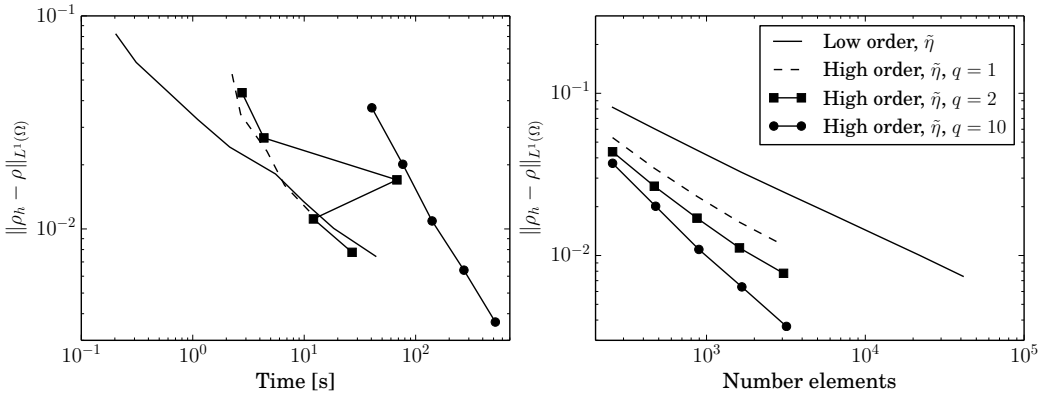


FIGURE 5.14: Time and elements convergence comparison for the compression corner problem.

5.6.5 Reflected shock

This benchmark consists in two flow streams colliding at different angles. The domain has dimensions $[0.0, 1.0] \times [0.0, 4.1]$ and a solid wall at its lower boundary. This configuration leads to a steady shock separating both flow regimes that is reflected at the wall producing a third different flow state behind it. A sketch of this benchmark test is given in Fig. 5.15. The flow states at each region have been collected in Tab. 5.1.

TABLE 5.1: Reflected shock solution values at every region.

Region	Density [$\mathrm{Kg\,m^{-3}}$]	Velocity [$\mathrm{m\,s^{-1}}$]	Total energy [J]
ⓐ	1.0	(2.9, 0.0)	5.99075
ⓑ	1.7	(2.62, -0.506)	5.8046
ⓒ	2.687	(2.401, 0.0)	5.6122

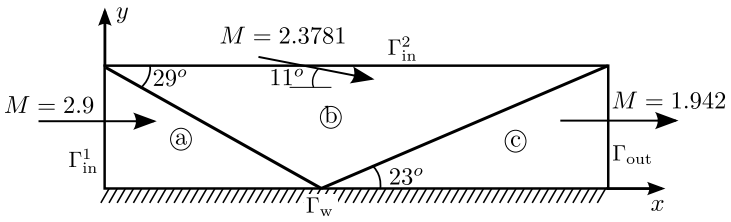


FIGURE 5.15: Reflected shock scheme.

We analyze the effectiveness of the high-order scheme, and evaluate the performance of the graph Laplacian estimator. We start with a coarse mesh of 16×64 elements and adapt the mesh till a certain number of elements is reached. For the high-order method, we set a maximum of 10^4 elements. The maximum number of elements for the low-order method is $3 \cdot 10^5$. We use a nonlinear tolerance of $\|\Delta \boldsymbol{u}_h\|/\|\boldsymbol{u}_h\| < 10^{-4}$ and a maximum of 500 iterations.

Fig. 5.16 compares the effectiveness of the low-order and the high-order stabilization schemes for different values of q. The high-order scheme converges efficiently and the overhead of solving a nonlinear problem does not affect the overall performance. Actually, for the most refined meshes the high-order method is more efficient than the low-order one. As for the previous problem, Fig. 5.16 shows that the high-order scheme can present nonlinear convergence problems at some steps of the refinement process. However, as the mesh becomes more adapted to the problem this issues is reduced.

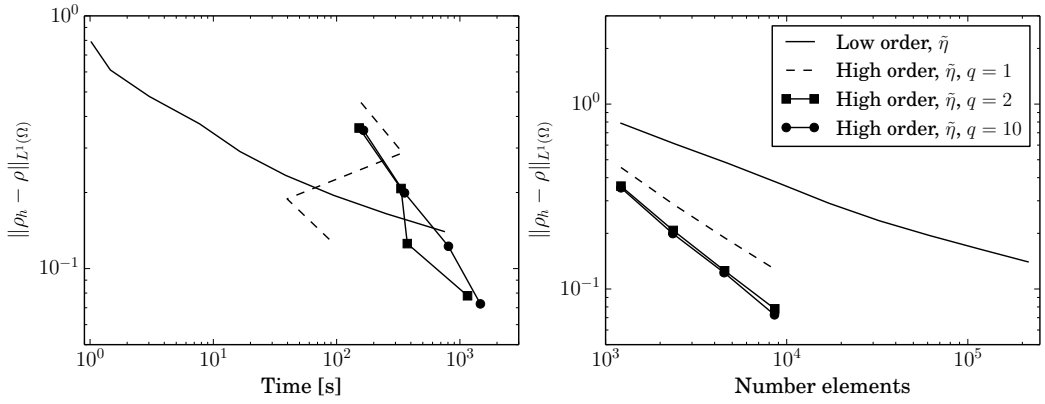


FIGURE 5.16: Time and elements convergence comparison for the reflected shock problem.

In Fig. 5.17 we depict the refinement evolution for the graph Laplacian estimator ($\tilde{\eta}_K$) for the low-order scheme. In these figures it can be observed how the graph Laplacian estimator is able to concentrate all the resolution at the shock location. Finally, we can conclude from the lower two figures that both schemes resolve the shocks properly after the mesh has been refined enough.

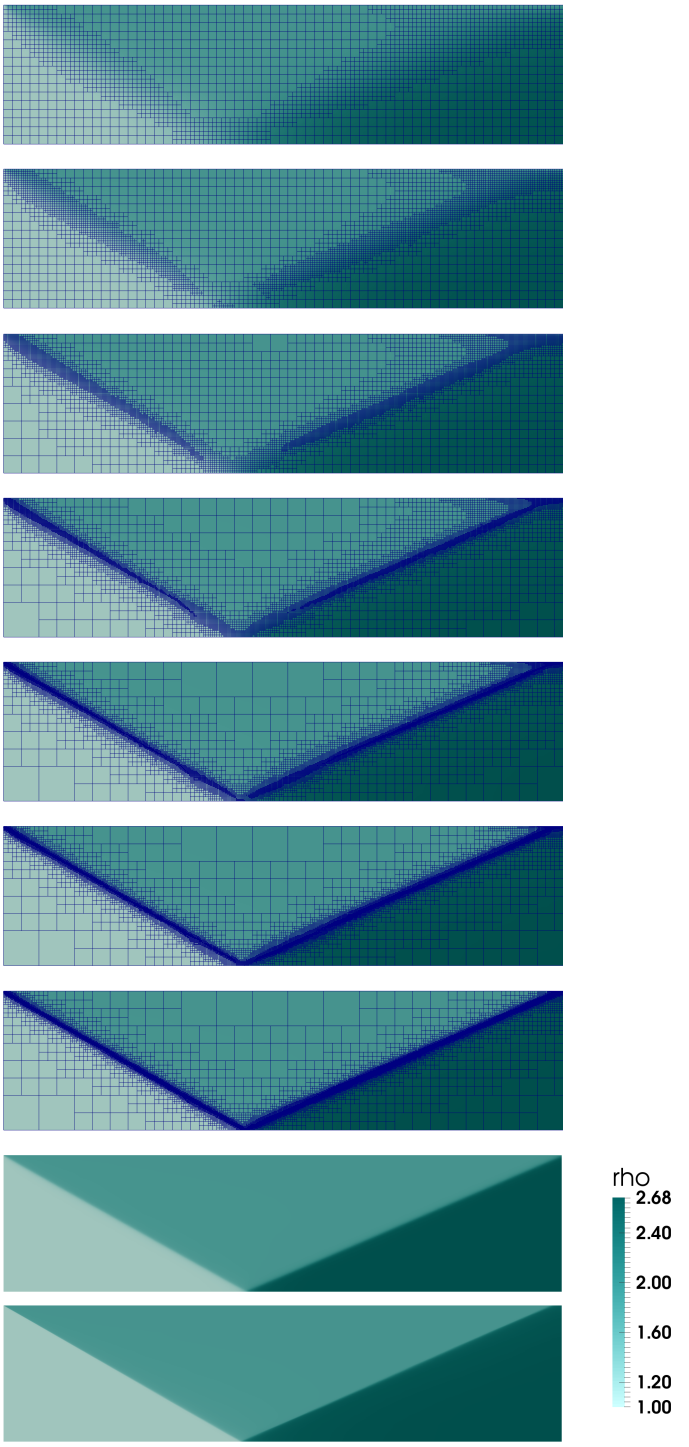


FIGURE 5.17: Evolution of the mesh refinement process. $\tilde{\eta}_K$ with low-order scheme is used. For the low-order scheme from top to bottom results have been obtained at refinement step 1, 2, 3, 4, 5, 6, and 7. The lower two figures are the high-order with $q = 2$ (top) and low-order (bottom) results at their last refinement step.

5.7 Conclusions

The stabilization schemes in Chapters 2 and 4 have been extended and assessed in the AMR context for nonconforming hierarchical octree meshes. The chapter focuses in assessing the effectiveness of linear (first-order) and nonlinear (higher-order) stabilization. We focus the comparison in terms of accuracy versus computational time.

The results indicate that linear stabilization is more effective for coarse meshes. In this case, the computational cost required to solve the stiff nonlinear problem due to the nonlinear stabilization does not compensate the improvement in the accuracy. This is especially evident for linear systems of PDEs. On the contrary, as the mesh is refined and properly adapted to the shocks, nonlinear stabilization pays the price. Even though increasing the value of q in the nonlinear stabilization (a parameter that makes shocks sharper but hinders nonlinear convergence) improves accuracy, it turns to be far more effective to refine the mesh further for low values of q. Nevertheless, it is worth mentioning that high-order method might exhibit nonlinear convergence problems for some meshes.

In addition, a new refinement criterion have been proposed. The proposed estimator is based on the graph Laplacian used in the definition of the stabilization method. Numerical results show that this shock detector is able to perform better that the well known Kelly estimator for problems with shocks or discontinuities.

Chapter 6

Conclusions and future work

6.1 Conclusions

In this thesis, the development of monotonicity-preserving FE methods has been explored. Since the main chapters of this dissertation are self-contained and preserve the structure of a paper, each one contains their own detailed conclusions. In this chapter, we present a more general overview. To this end, let us recall the list of goals set in Sect. 1.2.

- **Design of a monotonicity-preserving scheme for arbitrary mesh geometries.**

  In Chapter 2 we consider a nonlinear stabilization technique for the FE approximation of scalar conservation laws with implicit time stepping. The method relies on an artificial diffusion method, based on a graph-Laplacian operator. The artificial diffusion term is based on a graph-Laplacian artificial diffusion operator, instead of a PDE-based one. This removes any requirements on the mesh. In addition, this strategy is also used in subsequent chapters. The stabilization method proposed in Chapter 2 satisfies the local DMP, and thus preserves monotonicity. Furthermore, all numerical results in Sect. 2.9 exhibit this property.

- **Analysis and improvement of the nonlinear convergence behavior of monotonicity-preserving schemes.**

  The scheme proposed in Chapter 2 is proved to be Lipschitz continuous. This property leads to well-posedness of the nonlinear problem. However, the resulting scheme is highly nonlinear, leading to very poor nonlinear convergence rates. Therefore, we also propose a regularized version of the scheme that is twice differentiable. This allowed us to use Newton's method with the exact Jacobian. Numerical experiments in Sect. 2.9 show a reduction of 10 to 20 times in the number of iterations with respect to the original non-differentiable algorithms.

- **Extension to high-order discretizations in space and time.**

  In Chapter 3, the stabilization method in Chapter 2 is extended to isogeometric analysis. The proposed method is DMP-preserving for arbitrary high-order discretizations in space and time without any CFL-like condition. Moreover, in

order to reduce the computational cost of the space–time method, we propose a partitioned scheme in Sect. 3.4. This alternative scheme is also proved to be unconditionally DMP-preserving. In addition, all numerical in results Sect. 3.6 exhibit this property.

- **Extension to first order hyperbolic systems of equations.**

  Chapter 4 is devoted to this goal. Extension of monotonicity preservation to systems of equations is not straightforward. Actually, the continuous problem does not necessarily need to have monotonic solutions. In this case, previous works in literature resort to proving local bounds preservation. This could be seen as an heuristic extension of the properties required to prove the LED property for scalar problems (see Th. 2.4.1 and Sect. 4.2.3). In Chapter 4, a differentiable *local bounds preserving* stabilization method for Euler equations is developed.

  In addition, a continuation method for the regularization parameters present in the differentiable stabilization is proposed to improve nonlinear convergence. Numerical results show that differentiability not only can improve nonlinear convergence, but it also improves the robustness of the method. However, the improvement in the nonlinear convergence is restricted to moderate tolerances, and is not as significant as in Chapter 2.

- **Extension to AMR FE schemes.**

  In Chapter 5, the stabilization schemes in Chapter 2 and Chapter 4 are extended and assessed in the AMR context. In particular, we use nonconforming hierarchical octree meshes. The stabilization method for scalar problems is defined using the assembled matrix. Thus, it can work directly with this kind of meshes without any modification. Instead, the stabilization method for systems of equations uses elemental values to define the artificial diffusion. Hence, minor modifications are introduced to adapt the scheme. In any case, with these modifications, the scheme preserves all the properties of the original method (see 5.3.3).

  Moreover, a new refinement criterion is proposed in Sect. 5.4. The proposed estimator is based on the graph Laplacian used in the definition of the stabilization method. Numerical results show that this shock detector is able to perform better than the well known Kelly estimator for problems with shocks or discontinuities.

- **Assessment of the efficiency of high-order monotonicity-preserving schemes in AMR context.**

  The results of Chapter 5 indicate that the high-order scheme only becomes superior for a sufficiently refined mesh. The low-order method might perform better for coarse meshes. However, the high-order scheme has a higher convergence rate. Thus, it outperforms the low-order scheme once the mesh is refined enough.

In addition, it can be observed in Sect. 5.6 that it is more efficient to refine the mesh than to improve accuracy by using high values of q. Let us recall that q is a parameter in the stabilization that allows one to modulate the amount of artificial diffusion introduced. The higher it is, the less diffusive the presented method is.

Nevertheless, it is worth mentioning that the high-order method might exhibit nonlinear convergence problems for some meshes. In this scenario, the low-order scheme clearly outperforms the high-order method.

6.2 Future work

Research never comes to an end, it is simply bounded by time. In this section, we proceed to describe a few ideas that arise as possible continuation of the developments in this dissertation.

- **Smoothness indicator for isogeometric analysis**

  The high-order method developed in Chapter 3 yields solutions that satisfy the global DMP for arbitrary order discretizations. However, only for monotonic solutions the method is able to recover the high-order convergence rates. This is a direct consequence of the stabilization method introduced, which is only second order accurate. Lohmann et al. [71] proposed to use smoothness indicators based on the behavior of second order derivatives. This prevents to formally proof monotonicity preservation, but numerical experiments show an improved behavior and high-order convergence rates can be recovered. An interesting work could be to develop such a smoothness indicator for the particular case of isogeometric analysis. In this case, one could take advantage of the higher continuity of this kind of discretizations.

- **Parallelization of the implementation**

  The methods presented in this dissertation have only been tested in serial experiments. In any case, all methods are local and nothing prevents its parallelization. However, the domain of dependence is slightly larger than the one for regular FE methods. Therefore, the parallel implementation should be adapted to support at least one layer of ghost elements at subdomain interfaces. This is especially important if one is willing to compute the exact Jacobian. Otherwise, if an inexact Jacobian is sufficient, as performed in [18], then the parallel implementation does not require any special requirement.

- **AMR for high-order methods**

  In Chapter 5, we have adapted the methods in Chapter 2 and 4 to adaptive meshes. However, this is not the case for the method in Chapter 3 due to the higher coupling of isogeometric analysis basis functions. Bornemann and Cirak [20] use hierarchical B-splines to achieve an AMR discretization. We consider interesting to

combine this kind of B-spline discretizations with the AMR methods in Chapter 5. Furthermore, developing an *hp*-adaptive method using this strategy could lead to an improved behavior for problems that combine shocks with regions where the solution is smooth.

- **Extension to other problems**

  The most complex problem solved in this thesis are the Euler equations. As motivated in Chapter 1, we would like to eventually extend these methods to enhance plasma simulations. Therefore, the immediate development to be performed is the extensions to compressible Navier-Stokes, and ideal magnetohydrodynamics (MHD). In a latter stage, extensions to resistive MHD, and multi-fluid plasma equations should also be performed.

- **Extension to compatible discretizations**

  In combination with the previous point, we consider interesting to explore extensions to compatible discretizations. The magnetic field in MHD formulations is solenoidal. Several strategies have been developed to deal with this constraint. A common approach is to use Nédélec FEs [80] to discretize the magnetic field. Therefore, we consider that extending the methods presented in this dissertation to compatible discretizations could increase their applicability.

Bibliography

[1] M. Ainsworth and J. Tinsley Oden, *A posteriori error estimation in finite element analysis*, Computer Methods in Applied Mechanics and Engineering, 142 (1997), pp. 1–88.

[2] R. Anderson, V. Dobrev, T. Kolev, D. Kuzmin, M. Quezada de Luna, R. Rieben, and V. Tomov, *High-order local maximum principle preserving (MPP) discontinuous Galerkin finite element method for the transport equation*, Journal of Computational Physics, 334 (2017), pp. 102–124.

[3] J. D. Anderson Jr., *Modern Compressible Flow*, McGraw-Hill, 2nd ed., 1990.

[4] S. Badia and J. Bonilla, *Monotonicity-preserving finite element schemes based on differentiable nonlinear stabilization*, Computer Methods in Applied Mechanics and Engineering, 313 (2017), pp. 133–158.

[5] S. Badia, J. Bonilla, and A. Hierro, *Differentiable monotonicity-preserving schemes for discontinuous Galerkin methods on arbitrary meshes*, Computer Methods in Applied Mechanics and Engineering, 320 (2017), pp. 582–605.

[6] S. Badia, J. Bonilla, S. Mabuza, and J. N. Shadid, *Differentiable local bounds preserving stabilization for first order hyperbolic problems*, Submitted, (2019).

[7] S. Badia and A. Hierro, *On Monotonicity-Preserving Stabilized Finite Element Approximations of Transport Problems*, SIAM Journal on Scientific Computing, 36 (2014), pp. A2673–A2697.

[8] S. Badia and A. Hierro, *On discrete maximum principles for discontinuous Galerkin methods*, Computer Methods in Applied Mechanics and Engineering, 286 (2015), pp. 107–122.

[9] S. Badia and A. F. Martín, *A tutorial-driven introduction to the parallel finite element library FEMPAR v1.0.0*, (2019).

[10] S. Badia, A. F. Martín, E. Neiva, and F. Verdugo, *A generic finite element framework on parallel tree-based adaptive meshes*, Submitted, (2019).

[11] S. Badia, A. F. Martín, and J. Principe, *FEMPAR: An Object-Oriented Parallel Finite Element Framework*, Archives of Computational Methods in Engineering, 25 (2018), pp. 195–271.

[12] W. Bangerth, C. Burstedde, T. Heister, and M. Kronbichler, *Algorithms and data structures for massively parallel generic adaptive finite element codes*, ACM Trans. Math. Softw., 38 (2012), pp. 14:1–14:28.

[13] G. R. Barrenechea, E. Burman, and F. Karakatsani, *Edge-based nonlinear diffusion for finite element approximations of convection-diffusion equations and its relation to algebraic flux-correction schemes*, Numerische Mathematik, (2016), pp. 1–25.

[14] G. R. Barrenechea, V. John, and P. Knobloch, *Analysis of Algebraic Flux Correction Schemes*, SIAM Journal on Numerical Analysis, 54 (2016), pp. 2427–2451.

[15] M. Bittl and D. Kuzmin, *An hp-adaptive flux-corrected transport algorithm for continuous finite elements*, Computing, 95 (2013), pp. 27–48.

[16] J. Bonilla and S. Badia, *Maximum-principle preserving space-time isogeometric analysis*, Computer Methods in Applied Mechanics and Engineering, 354 (2019), pp. 422–440.

[17] ——, *Monotonicity-preserving finite element schemes with adaptive mesh refinement for hyperbolic problems*, In preparation, (2019).

[18] J. Bonilla, S. Mabuza, J. N. Shadid, and S. Badia, *On Differentiable Linearity and Local Bounds Preserving Stabilization Methods for First Order Conservation Law Systems*, in Center for Computing Research Summer Proceedings 2018, A. Cangi and M. L. Parks, eds., Sandia National Laboratories, 2018, pp. 107–119.

[19] P. Bonoli and L. C. McInnes, *Report of the Workshop on Integrated Simulations for Magnetic Fusion Energy Sciences*, tech. report, 2015.

[20] P. B. Bornemann and F. Cirak, *A subdivision-based implementation of the hierarchical b-spline finite element method*, Computer Methods in Applied Mechanics and Engineering, 253 (2013), pp. 584–598.

[21] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, vol. 15 of Texts in Applied Mathematics, Springer New York, New York, NY, softcover ed., nov 2008.

[22] E. Burman, *Adaptive finite element methods for compressible flow*, Computer Methods in Applied Mechanics and Engineering, 190 (2000), pp. 1137–1162.

[23] ——, *On nonlinear artificial viscosity, discrete maximum principle and hyperbolic conservation laws*, BIT Numerical Mathematics, 47 (2007), pp. 715–733.

[24] ——, *A monotonicity preserving, nonlinear, finite element upwind method for the transport equation*, Applied Mathematics Letters, 49 (2015), pp. 141–146.

[25] E. Burman and A. Ern, *Nonlinear diffusion and discrete maximum principle for stabilized Galerkin approximations of the convection-diffusion-reaction equation*, Computer Methods in Applied Mechanics and Engineering, 191 (2002), pp. 3833–3855.

[26] ——, *Stabilized Galerkin approximation of convection-diffusion-reaction equations: discrete maximum principle and convergence*, Mathematics of Computation, 74 (2005), pp. 1637–1652.

[27] B. Cockburn and C.-W. Shu, *Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems*, Journal of Scientific Computing, 16 (2001), pp. 173–261.

[28] R. Codina, *A discontinuity-capturing crosswind-dissipation for the finite element solution of the convection-diffusion equation*, Computer Methods in Applied Mechanics and Engineering, 110 (1993), pp. 325–342.

[29] J. A. Cottrell, T. J. R. Hughes, and Y. Bazilevs, *Isogeometric analysis: toward integration of CAD and FEA*, Wiley, 2009.

[30] C. De Boor, *B(asic)-Spline Basics*, tech. report, Madison mathematics research center, Winsconsin University., 1986.

[31] L. Demkowicz, *Computing with hp-ADAPTIVE FINITE ELEMENTS: One and Two Dimensional Elliptic and Maxwell Problems*, vol. 1, CRC Press, oct 2006.

[32] J. Donea and A. Huerta, *Finite Element Methods for Flow Problems*, John Wiley & Sons, Ltd, Chichester, UK, apr 2003.

[33] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Introduction to Adaptive Methods for Differential Equations*, Acta Numerica, 4 (1995), pp. 105–158.

[34] M. M. Feistauer, J. J. Felcman, and I. I. Straškraba, *Mathematical and computational methods for compressible flow*, Oxford University Press, 2003.

[35] H. Frid, *Maps of Convex Sets and Invariant Regions for Finite-Difference Systems of Conservation Laws*, Archive for Rational Mechanics and Analysis, 160 (2001), pp. 245–269.

[36] S. Godunov, *Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics*, Matematicheskii Sbornik, Steklov Mathematical Institute of Russian Academy of Sciences, 47(89) (1959), pp. 271–306.

[37] S. Gottlieb, C.-W. Shu, and E. Tadmor, *Strong Stability-Preserving High-Order Time Discretization Methods*, SIAM Review, 43 (2001), pp. 89–112.

[38] J.-L. Guermond and M. Nazarov, *A maximum-principle preserving finite element method for scalar conservation equations*, Computer Methods in Applied Mechanics and Engineering, 272 (2014), pp. 198–213.

[39] J.-L. Guermond, M. Nazarov, B. Popov, and Y. Yang, *A Second-Order Maximum Principle Preserving Lagrange Finite Element Technique for Nonlinear Scalar Conservation Equations*, SIAM Journal on Numerical Analysis, 52 (2014), pp. 2163–2182.

[40] J.-L. Guermond and R. Pasquetti, *A correction technique for the dispersive effects of mass lumping for transport problems*, Computer Methods in Applied Mechanics and Engineering, 253 (2013), pp. 186–198.

[41] J.-L. Guermond, R. Pasquetti, and B. Popov, *Entropy viscosity method for nonlinear conservation laws*, Journal of Computational Physics, 230 (2011), pp. 4248–4267.

[42] J.-L. Guermond and B. Popov, *Invariant domains and first-order continuous finite element approximation for hyperbolic systems*, (2015), pp. 1–22.

[43] M. Gurris, *Implicit finite element schemes for compressible gas and particle-laden gas flows*, PhD thesis, Technische Universität Dortmund, 2009.

[44] A. Hierro, S. Badia, and P. Kus, *Shock capturing techniques for hp-adaptive finite elements*, Computer Methods in Applied Mechanics and Engineering, 309 (2016), pp. 532–553.

[45] D. Hoff, *A finite difference scheme for a system of two conservation laws with artificial viscosity*, Mathematics of Computation, 33 (1979), pp. 1171–1171.

[46] ——, *Invariant regions for systems of conservation laws*, Transactions of the American Mathematical Society, 289 (1985), pp. 591–591.

[47] T. J. Hughes and A. Brooks, *A multi-dimensioal upwind scheme with no crosswind diffusion.*, in: T.J.R. Hughes ed. Finite Element Methods for Convection Dominated Flows, (ASME, New York), 34 (1979), pp. 19–35.

[48] T. J. Hughes, L. P. Franca, and G. M. Hulbert, *A new finite element formulation for computational fluid dynamics: VIII. The galerkin/least-squares method for advective-diffusive equations*, Computer Methods in Applied Mechanics and Engineering, 73 (1989), pp. 173–189.

[49] T. J. Hughes, M. Mallet, and M. Akira, *A new finite element formulation for computational fluid dynamics: II. Beyond SUPG*, Computer Methods in Applied Mechanics and Engineering, 54 (1986), pp. 341–355.

[50] C. Johnson and A. Szepessy, *Adaptive finite element methods for conservation laws based on a posteriori error estimates*, Communications on Pure and Applied Mathematics, 48 (1995), pp. 199–234.

[51] C. T. Kelley and D. E. Keyes, *Convergence Analysis of Pseudo-Transient Continuation*, SIAM Journal on Numerical Analysis, 35 (1998), pp. 508–523.

[52] D. W. Kelly, J. P. De S. R. Gago, O. C. Zienkiewicz, and I. Babuska, *A posteriori error analysis and adaptive processes in the finite element method: Part I-error analysis*, International Journal for Numerical Methods in Engineering, 19 (1983), pp. 1593–1619.

[53] D. I. Ketcheson, C. B. Macdonald, and S. Gottlieb, *Optimal implicit strong stability preserving Runge-Kutta methods*, Applied Numerical Mathematics, 59 (2009), pp. 373–392.

[54] A. Kritz and D. Keyes, *Fusion Simulation Project Workshop Report*, Journal of Fusion Energy, 28 (2009), pp. 1–59.

[55] S. N. Kružkov, *First order quasilinear equations in several independent variables*, Mathematics of the USSR-Sbornik, 10 (1970), pp. 217–243.

[56] D. Kuzmin, *A Guide to Numerical Methods for Transport Equations*, Friedrich-Alexander-Universität, 2010.

[57] ——, *Linearity-preserving flux correction and convergence acceleration for constrained Galerkin schemes*, Journal of Computational and Applied Mathematics, 236 (2012), pp. 2317–2337.

[58] D. Kuzmin, S. Basting, and J. N. Shadid, *Linearity-preserving monotone local projection stabilization schemes for continuous finite elements*, Computer Methods in Applied Mechanics and Engineering, 322 (2017), pp. 23–41.

[59] D. Kuzmin, R. Löhner, and S. Turek, *Flux-corrected transport*, Springer, 2005.

[60] D. Kuzmin and M. Möller, *Algebraic Flux Correction I. Scalar Conservation Laws*, in Flux-Corrected Transport, D. D. Kuzmin, P. R. Löhner, and P. D. S. Turek, eds., Scientific Computation, Springer Berlin Heidelberg, jan 2005, pp. 155–206.

[61] D. Kuzmin, M. Möller, and M. Gurris, *Algebraic Flux Correction II. Compressible flows*, in Flux-corrected Transport: Principles, Algorithms, and Applications, 2012, pp. 193–238.

[62] D. Kuzmin, M. Möller, and S. Turek, *Multidimensional FEM-FCT schemes for arbitrary time stepping*, International Journal for Numerical Methods in Fluids, 42 (2003), pp. 265–295.

[63] D. Kuzmin, M. Quezada De Luna, C. E. Kees, D. Kuzmin, M. Quezada De Luna, and C. E. Kees, *A partition of unity approach to adaptivity and limiting in continuous finite element methods*, (2018).

[64] D. Kuzmin and J. N. Shadid, *A new approach to enforcing discrete maximum principles in continuous Galerkin methods for convection-dominated transport equations*, Journal of Computational Physics, (2015).

[65] D. Kuzmin and J. N. Shadid, *Gradient-based nodal limiters for artificial diffusion operators in finite element schemes for transport equations*, International Journal for Numerical Methods in Fluids, (2017), pp. 675–695.

[66] D. Kuzmin, M. J. Shashkov, and D. Svyatskiy, *A constrained finite element method satisfying the discrete maximum principle for anisotropic diffusion problems*, Journal of Computational Physics, 228 (2009), pp. 3448–3463.

[67] D. Kuzmin and S. Turek, *Flux Correction Tools for Finite Elements*, Journal of Computational Physics, 175 (2002), pp. 525–558.

[68] R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, Cambridge, 2002.

[69] C. Lohmann, *Algebraic flux correction schemes preserving the eigenvalue range of symmetric tensor fields*, Mathematical Modelling and Numerical Analysis, (2018).

[70] C. Lohmann and D. Kuzmin, *Synchronized flux limiting for gas dynamics variables*, Journal of Computational Physics, 326 (2016), pp. 973–990.

[71] C. Lohmann, D. Kuzmin, J. N. Shadid, and S. Mabuza, *Flux-corrected transport algorithms for continuous Galerkin methods based on high order Bernstein finite elements*, Journal of Computational Physics, 344 (2017), pp. 151–186.

[72] R. Löhner, *An adaptive finite element scheme for transient problems in CFD*, Computer Methods in Applied Mechanics and Engineering, 61 (1987), pp. 323–338.

[73] R. Lohner, *Applied Computational Fluid Dynamics Techniques: An Introduction Based on Finite Element Methods*, vol. 508, 2004.

[74] S. Mabuza, J. N. Shadid, E. C. Cyr, R. P. Pawlowski, and D. Kuzmin, *A positivity and linearity preserving nodal variation limiting algorithm for continuous Galerkin discretization of ideal MHD equations*, Submitted, (2019).

[75] S. Mabuza, J. N. Shadid, and D. Kuzmin, *Local bounds preserving stabilization for continuous Galerkin discretization of hyperbolic systems*, Journal of Computational Physics, 361 (2018), pp. 82–110.

[76] M. J. Marsden, *An identity for spline functions with applications to variation-diminishing spline approximation*, Journal of Approximation Theory, 3 (1970), pp. 7–49.

[77] M. Möller and D. Kuzmin, *Adaptive mesh refinement for high-resolution finite element schemes*, International Journal for Numerical Methods in Fluids, 52 (2006), pp. 545–569.

[78] M. Nazarov, J.-L. Guermond, and B. Popov, *A posteriori error estimation for the compressible Euler equations using entropy viscosity*, tech. report, 2011.

[79] M. Nazarov and J. Hoffman, *An adaptive finite element method for inviscid compressible flow*, International Journal for Numerical Methods in Fluids, 64 (2010), pp. 1102–1128.

[80] J. C. Nedelec, *Mixed finite elements in R3*, Numerische Mathematik, 35 (1980), pp. 315–341.

[81] D. E. Post, D. B. Batchelor, R. B. Bramley, J. R. Cary, R. H. Cohen, P. Colella, and S. C. Jardin, *Report of the Fusion Simulation Project Steering Committee*, Journal of Fusion Energy, 23 (2004), pp. 1–26.

[82] P. Roe, *Approximate Riemann solvers, parameter vectors, and difference schemes*, Journal of Computational Physics, 43 (1981), pp. 357–372.

[83] L. R. Scott and S. Zhang, *Finite Element Interpolation of Nonsmooth Functions Satisfying Boundary Conditions*, Mathematics of Computation, 54 (1990), pp. 483–493.

[84] F. Shakib, T. J. R. Hughes, and Z. Johan, *A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations*, Computer Methods in Applied Mechanics and Engineering, 89 (1991), pp. 141–219.

[85] R. J. Singleton, D. M. Israel, S. W. Doebling, C. N. Woods, A. Kaul, J. W. J. Walter, and M. L. Rogers, *ExactPack Documentation*, tech. report, Los Alamos National Laboratory, 2017.

[86] T. Smith, R. Hooper, C. Ober, A. Lorber, and J. Shadid, *Comparison of Operators for Newton-Krylov Method for Solving Compressible Flows on Unstructured Meshes*, in 42nd AIAA Aerospace Sciences Meeting and Exhibit, vol. 87, Reston, Virigina, 2004, American Institute of Aeronautics and Astronautics.

[87] E. Süli, *A Posteriori Error Analysis and Adaptivity for Finite Element Approximations of Hyperbolic Problems*, (1999), pp. 123–194.

[88] T. E. Tezduyar and M. Senga, *Stabilization and shock-capturing parameters in SUPG formulation of compressible flows*, Computer Methods in Applied Mechanics and Engineering, 195 (2006), pp. 1621–1632.

[89] T. Tiankai Tu, D. O'Hallaron, and O. Ghattas, *Scalable Parallel Octree Meshing for TeraScale Applications*, in ACM/IEEE SC 2005 Conference (SC'05), IEEE, 2005, pp. 4–4.

[90] A. Tikhonova, G. Tanase, O. Tkachyshyn, N. M. Amato, and L. Rauchwerger, *Parallel Algorithms in STAPL: Sorting and the Selection Problem*, tech. report, 2005.

[91] E. F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 3rd ed., 2009.

[92] R. Verfurth, *A Posteriori Error Estimation Techniques for Finite Element Methods*, Oxford University Press, 2013.

[93] J. Xu and L. Zikatanov, *A Monotone Finite Element Scheme for Convection-Diffusion Equations*, Mathematics of Computation, 68 (1999), pp. 1429–1446.

[94] O. C. Zienkiewicz and J. Z. Zhu, *A simple error estimator and adaptive procedure for practical engineerng analysis*, International Journal for Numerical Methods in Engineering, 24 (1987), pp. 337–357.

[95] O. C. Zienkiewicz and J. Z. Zhu, *The superconvergent patch recovery anda posteriori error estimates. Part 1: The recovery technique*, International Journal for Numerical Methods in Engineering, 33 (1992), pp. 1331–1364.