Hierarchical finite element methods for compressible flow problems

---

Dissertation
zur Erlangung des Grades eines
Doktors der Naturwissenschaften

Der Fakultät für Mathematik der
Technischen Universität Dortmund
vorgelegt von

Melanie Bittl

Hierarchical finite element methods for compressible flow problems
Melanie Bittl

Dissertation eingereicht am: 03. 12. 2014
Tag der mündlichen Prüfung: 26. 02. 2015

Mitglieder der Prüfungskommission:

Prof. Dr. Dmitri Kuzmin (1. Gutachter, Betreuer)
Prof. Dr. Roland Becker (2. Gutachter)
Prof. Dr. Stefan Turek
Prof. Dr. Ben Schweizer

"*When you have exhausted all possibilities, remember this - you haven't.* "

by Thomas A. Edison

Acknowledgements

I would like to thank all the people who contributed directly or indirectly to the realization of this thesis.

First and foremost, I would like to express my deepest gratitude to my advisor Dmitri Kuzmin for giving me the opportunity to write this thesis and guiding me through all those years. He gave me the opportunity to attend conferences to present our work and introduced me to a lot of researchers who are expert in the field of computational fluid dynamics. He always took his time to discuss any problems arising during my research. I could not have imagined having a better advisor.

I am also very grateful to Roland Becker who played an important role in the development of the CG1-DG2 method. Without him the analytical results described in this thesis would not have been proven.

My appreciation also goes to my former colleagues at the University of Erlangen-Nuremberg as well as to my new colleagues at the TU Dortmund for the friendly atmosphere and their support.

All numerical studies presented in this thesis haven been computed using the *hp*-FEM library Hermes. Therefore, I would like to thank Pavel Solin (Hermes group leader), who helped me to get familiar with Hermes very quickly, and Lukas Korous (main developer), who always took his time answering my questions and implementing new features into Hermes helping me with my work.

I would like to particularly thank the following people for proofreading this thesis and providing me with valuable feedback: Dmitri Kuzmin, Roland Becker, Christopher Basting and Steffen Basting.

This work was also supported by the DFG (German Research Association), which is gratefully acknowledged.

Last but not least, I would like to thank my family for their love and support throughout my life. Most of all, I am deeply grateful to Steffen for his encouragement during my years of studies. He gave me the confidence and strength to complete this thesis.

Thank you all!

Dortmund, December 2014

Melanie Bittl

Contents

1 Introduction

The study of gas dynamics is an important task in modern life. It is concerned with the behavior of compressible flows like the flow of gas in pipelines or the air around high-speed aircrafts. Depending on the ratio of the gas speed to the speed of sound, which defines the so-called Mach number, different wave phenomenon, e.g., shock waves and rarefaction waves, can occur. The flow behavior can also be influenced by other materials, which pollute the gas. A typical example is an ash cloud after a volcano eruption, which consists of air and small solid particles (ash). It is very important to understand the physical and chemical processes which take place in the flow so that an accurate mathematical model can be established.

The next challenging step is the development of numerical methods to solve this model. For that reason one usually starts with a simpler problem which does not take into account all physical and chemical processes and tries to find a numerical method which solves it accurately and efficiently. The last step is then to transfer the method to the more complex problem.

A first step toward the simplification of flow problems is to investigate the behavior of a physical quantity driven by diffusion and advection. Diffusion is the process induced by a concentration gradient where the considered quantity moves from a region of higher concentration to a region of lower concentration. For example, considering a glass of water containing a piece of sugar, the sugar will diffuse until the concentration is equally distributed. This process is modeled by the diffusion equation

$$u_t - \nabla \cdot (D \nabla u) = 0,$$

where $u = u(\mathbf{x}, t)$ denotes the concentration and $D = D(\mathbf{x}, t)$ the diffusion coefficient, both depending on space $\mathbf{x}$ and time t .

Advection is the transport of the considered quantity induced by the motion of the surrounding fluid or gas. For example, considering a flowing river with a (light) object thrown into, the object will be transported due to advection. This movement is usually modeled by the continuity equation

$$u_t + \nabla \cdot (\boldsymbol{\beta} u) = 0,$$

where $\boldsymbol{\beta}$ is the velocity field given by the motion of the fluid/gas. The movement of the fluid/gas is usually induced by physical forces such as gravity. Often advection and diffusion arise together, so that we need to deal with a convection-diffusion equation

$$u_t - \nabla \cdot (D \nabla u) + \nabla \cdot (\boldsymbol{\beta} u) = 0.$$

Hereby, the term convection is often used as synonym for advection.

One of the most popular numerical methods used to discretize convection-diffusion equations with respect to space is the continuous Galerkin finite element method. It is very well suited for solving pure diffusion problems, but has only L^2-stability in the advection case meaning that oscillations can occur even if the exact solution is smooth. Therefore, the continuous Galerkin method is usually regarded as unstable in convection-dominated regimes. Popular stabilization techniques are the streamline upwind/Petrov-Galerkin (SUPG) method [16, 18, 49] and the flux-corrected transport (FCT) algorithm [58]. The first method adds stabilizing terms to the variational formulation which gives additional control over streamline derivatives. However, it is not monotone and therefore, if sharp layers are present, additional stabilization-improvements like the SOLD-method need to be considered [47]. The FCT algorithm works on an algebraic level to keep the solution local extremum diminishing and positivity preserving. Discontinuous solutions are resolved very well by this scheme in the sense that the discontinuity is treated as a steep gradient and no oscillations occur. In [49] it was shown that the FCT scheme applied to time-dependent convection-diffusion-reaction problems is the most promising stabilization scheme concerning accuracy and efficiency.

Another popular approach to solving advection-diffusion equations is the discontinuous Galerkin (DG) method which was first introduced to solve the neutron transport equation [76]. Using upwind numerical fluxes makes this method stable in the context of advection equations [53]. In this case stability is given in a similar manner as for the SUPG method which means that besides L^2-stability one obtains control over streamline derivatives and, in contrast to continuous methods, over jumps across element boundaries. Therefore, it is superior to the unstabilized continuous version when it comes to solving convection-dominated problems. However, the downside of this method is the higher computational costs due to more degrees of freedom (DOFs). As with the SUPG-method, under- and overshoots near steep gradients can occur. To prevent those, discontinuity capturing and slope limiting techniques have been developed [22, 41, 60].

The mentioned methods can also be applied to more complex problems like the Euler equations which model compressible gas flows. An extension of the SUPG-method to solve systems can be found in [44, 52]. The FCT scheme can also handle systems of conservation laws [64, 70, 88] but was only considered for linear finite elements in these references. For the DG-method an extension to solve the Euler equations is also given, e.g. in [27, 30]. In this context the application of slope limiting and discontinuity capturing techniques have proven successful as well [23, 56].

In elements located near steep gradients or singularities, large errors usually occur. Therefore, different refinement strategies like h-, p- or hp-adaptivity can be considered to improve accuracy [83]. Thereby, h indicates the maximum mesh size and p the maximum polynomial degree. To decide which element needs to be refined, different error indicators can be used. For example, the Z^2 estimator [89] uses the element-wise error between the gradient and a reconstructed gradient as indicator. In the case of hp-adaptivity one also needs to decide, if h or p or both should be refined. One possibility is to increase p in elements where the solution is expected to be smooth and h-refine elements in non-smooth regions. A summary of different strategies can be found in [71]. The advantage of the combined hp-adaptive approach is that in theory an exponential order of convergence can be be expected at least for elliptic problems [7] .

1.1. Objectives

The primary goal of this research is to develop a hierarchical finite element method which can be applied to compressible flow problems and also be used in the context of hp-adaptivity. The type of compressible flow problems which we are interested in are modeled by the Euler equations. These equations are given as a system of conservation laws. A peculiarity of conservation laws is

the appearance of shocks, which results in discontinuous solutions traveling over time. Since such discontinuities usually appear locally, one needs to refine only a few elements to obtain a better resolution of those phenomena. If in addition element-wise changes of the polynomial degree are considered, *hp*-adaptivity is derived resulting in a higher accuracy of the solution but lower computational costs compared to uniform refinements. Since the simplest case of a conservation law is the linear advection equation, the first step of our research was the development of an *hp*-adaptive algorithm for this equation.

The solution of advection equations can be obtained by a stabilized continuous Galerkin method or the DG method. In case of the DG method different *hp*-adaptive algorithms have already been presented [10, 42, 43]. However, since solutions of conservation laws frequently comprise shocks, i.e., discontinuities, additional stabilization to prevent oscillations near discontinuities may be applied [22, 41].

Since the DG method is computationally more expensive than the continuous version, we prefer the FCT scheme which is capable of handling steep gradients very well [49, 58]. Starting from the continuous linear finite element method, which is referred to as high-order scheme in this context, a low-order scheme is derived algebraically by applying mass-lumping and introducing artificial diffusion. This gives a low-order approximation which is local-extremum diminishing and positivity preserving [58]. The difference between these two schemes defines sums of anti-diffusive fluxes. These fluxes need to be controlled since they can cause non-physical oscillations. A multidimensional limiter was introduced by Zalesak [88] which limits the fluxes in such a way that the so-derived solution fulfills discrete maximum principles and is free of oscillations.

In [72] an *h*-adaptive FCT algorithm for linear finite elements was presented. A straightforward extension of this scheme to *hp*-adaptivity is difficult, since the design of the FCT scheme for higher-order elements is complicated [59]. Applying the strategy for the derivation of a low-order scheme to higher order elements equipped with Lagrange basis functions can lead to low-order approximations which are highly oscillatory. Positivity and compliance with discrete maximum principles are not guaranteed in this case. Also the design of limiters for higher-order FCT is not clear, since besides the fluxes between vertex nodes edge-edge fluxes and edge-vertex fluxes need to be taken into account. Therefore, the FCT scheme is usually considered for linear elements only.

To avoid this complication with FCT for higher-order elements, we divide the mesh in smooth and non-smooth parts and use FCT only in non-smooth parts together with linear elements. Hereby, the smoothness refers to the regularity of the solution. Different techniques to capture discontinuities have been developed in the context of the DG method [41, 60]. In the case of continuous finite elements a parameter-free smoothness indicator based on the design of DG slope-limiters was proposed in [61]. It compares the solution with a reconstructed solution which may be discontinuous. If the continuous solution evaluated at element centers is bounded by the reconstructed approximation, the solution can be regarded as smooth. Similarly, the smoothness of higher order derivatives can be verified. We will adopt this indicator and consider an element as smooth, if the solution or all components of its gradient are smooth.

In smooth parts the solution can be approximated by higher-order elements without further stabilization. One possibility may be the use of the continuous Galerkin method in these elements. However, since the method is not stable, it cannot be guaranteed that smooth regions may stay free of oscillations during a long-time computation. Therefore, a stable method should be applied in smooth elements.

One possible approach is the use of the DG method which is stable but of higher computational costs. To reduce the costs different attempts have already been made. For example, in [46] a multiscale DG method was introduced which divides the problem in continuous coarse and discontinuous fine scales, where the continuous part has the computational costs of a continuous

finite element problem. The fine-scale part is taken into account by using an inter-scale transfer operator. Another approach was introduced in [11] where the continuous piecewise-linear finite element (CG1) space was enriched by discontinuous piecewise-constants. It was shown that this approach leads to the same convergence rate as the linear DG method but with fewer degrees of freedom.

Since we are interested in higher-order polynomials, we adopt the idea from [11] and enrich the CG1 space with discontinuous quadratic basis functions. It can be shown that on triangular meshes the resulting method, referred to as CG1-DG2 method, is stable and leads to the same convergence rate as the quadratic DG method [12].

Using the FCT scheme combined with h-adaptivity in linear elements and the CG1-DG2 method in quadratic elements, gives us a stable hp-adaptive algorithm for scalar convection-dominated problems. Since FCT is only used in non-smooth elements, the appearance of peak clipping is also prevented. This phenomenon is well known and occurs at smooth extrema which are smoothed out by the FCT limiter [88].

For h-adaptivity different error estimators have been developed. For example, there are residual-based error estimators [1, 74, 85], which use the residual of the discrete solution as error indicator, or recovery-based error estimators [1, 89, 90], which compare reconstructed solutions or solution gradients with the numerical approximation to obtain an estimate of the error. The latter type of estimators has been further developed and been applied to FCT-schemes in [72]. Another strategy mainly used in the context of hp-adaptivity is the so-called reference solution approach [24, 83], which is based on the assumption that a reference solution leads to a better approximation of the exact solution than the solution on the current (coarse) space. Hereby, the reference solution is computed on the reference space which is usually derived by h- and p-refinement of the coarse space. We will adopt this approach and derive the reference space by h-refining non-smooth elements and p-enriching smooth elements. The error between the reference solution and its projection into the coarse space is used as indicator which elements should be refined.

1.2. Outline of the thesis

This thesis is concerned with two main topics: the introduction of the CG1-DG2 method and the design of an hp-adaptive algorithm for advection-diffusion equations.

Chapters 3 and 4 give a brief introduction to the continuous and discontinuous Galerkin method. Chapter 3 is concerned with the continuous Galerkin method and presents the assumptions for existence of a solution and stability. For Poisson's equation, we show that these assumptions are fulfilled and recall a priori error estimates in the H^1- and L^2-norm. For advection equations the method is not stable in the sense that derivatives cannot be controlled and therefore, only L^2-a priori error estimates can be derived.

Chapter 4 presents the discontinuous Galerkin method. In the advection case we define the upwind numerical flux to get a stable method and present a priori error estimates. For Poisson's equation we present different flux discretizations: Symmetric interior penalty [2], non-symmetric interior penalty [77] and the Baumann-Oden method [10]. For all methods, a priori error estimates are derived.

In Chapter 5 we introduce the CG1-DG2 space. For the linear advection equation on triangular meshes we show stability and present a priori error estimates which are similar to those derived for the DG method. In the case of Poisson's equation the analytical results of the pure discontinuous method can be directly adopted for the CG1-DG2 space. The analytical results are confirmed by numerical computations in Chapter 6. Here we will present stationary and time-dependent examples. These examples show that the new method produces similar results to those computed by the DG method.

After that, in Chapter 7 we extend the new method to systems of conservation laws. We introduce the Euler equations and describe the derivation of the discrete system for continuous and discontinuous methods. Chapter 8 shows numerical results for the Euler equations which indicate that the CG1-DG2 methods lead to similar results as the DG method.

In Chapter 9 we derive the *hp*-adaptive algorithm for convection-dominated problems. At first we explain the FCT algorithm and show, how positivity of the numerical approximation can be guaranteed and which assumptions need to be satisfied to fulfill discrete maximum principles. Then we will present a regularity estimator, which determines smooth elements. The presented *hp*-adaptive algorithm is based on the reference solution approach and defines a reference space with linear non-smooth elements and smooth quadratic elements. FCT is used on linear elements and the CG1-DG2 method on quadratics. Finally, we present a constrained L^2 projection that guarantees that no new extrema are introduced.

Chapter 10 shows numerical results for the different topics discussed in the previous chapter. It also shows the advantage of the *hp*-adaptive algorithm over pure *h*-refinement.

The last chapter concludes this thesis and gives an outlook for further research.

1.3. Original publications

The analytical results and part of the numerical results concerning the scalar equations for the CG1-DG2 method have already been published or have been accepted for publication. The results can be found in [12] and [15] which are joint work with Roland Becker and the author's advisor Dmitri Kuzmin. The work concerning *hp*-adaptivity can be found in [13] and [14] and was developed with Dmitri Kuzmin. The numerical results presented in this thesis are the author's own work. The list of publications is as follows:

- [12] R. Becker, M. Bittl, and D. Kuzmin. Analysis of a combined cg1-dg2 method for the transport equation. *SIAM Journal on Numerical Analysis*, 53(1):445–463, 2015. doi: 10.1137/13093683X. URL http://dx.doi.org/10.1137/13093683X

- [13] M. Bittl and D. Kuzmin. An *hp*-adaptive flux-corrected transport algorithm for continuous finite elements. *Computing*, 95(1, suppl.):S27–S48, 2013. ISSN 0010-485X. doi: 10.1007/s00607-012-0223-y. URL http://dx.doi.org/10.1007/s00607-012-0223-y

- [14] M. Bittl and D. Kuzmin. The reference solution approach to *hp*-adaptivity in finite element flux-corrected transport algorithms. In I. Lirkov, S. Margenov, and J. Wasniewski, editors, *Large-Scale Scientific Computing*, Lecture Notes in Computer Science, pages 197–204. Springer Berlin Heidelberg, 2014. ISBN 978-3-662-43879-4. URL http://dx.doi.org/10.1007/978-3-662-43880-0_21

- [15] M. Bittl, D. Kuzmin, and R. Becker. The cg1-dg2 method for convection-diffusion equations in 2d. *J. Comput. Appl. Math.*, 270(0):21 – 31, 2014. ISSN 0377-0427. doi: http://dx.doi.org/10.1016/j.cam.2014.03.008. URL http://www.sciencedirect.com/science/article/pii/S0377042714001484. Fourth International Conference on Finite Element Methods in Engineering and Sciences (FEMTEC 2013)

2

General Notation

In this chapter we introduce the general notation used throughout the thesis. Table 2.1 provides an overview of the mathematical notation used in the scope of partial differential equations. The notation related to spatial discretization can be found in Table 2.2.

In the context of discontinuous Galerkin discretizations, we define for an interior side $S \in \mathcal{S}_h^{\text{int}}$ of an element K, i.e., $S \in S_K$, $\mathbf{x} \in S$, and the normal vector $\mathbf{n}_{S_K}$

$$u_{S_K}^+(\mathbf{x}) := \lim_{\varepsilon \searrow 0} u_h(\mathbf{x} - \varepsilon \mathbf{n}_{S_K}), \quad \text{the interior trace,}$$

and

$$u_{S_K}^-(\mathbf{x}) := \lim_{\varepsilon \searrow 0} u_h(\mathbf{x} + \varepsilon \mathbf{n}_{S_K}), \quad \text{the exterior trace,}$$

where u_h is a scalar-valued function smooth enough to admit a possibly two-valued trace. This is also illustrated in Fig. 2.1 for an element K and its neighboring element K'.

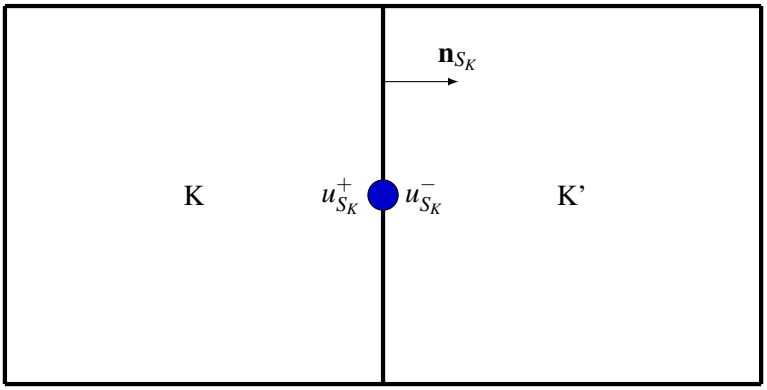


Figure 2.1: Element K and neighbor element K' with normal vector $\mathbf{n}_{S_K}$ and interior ($u_{S_K}^+$) and exterior ($u_{S_K}^-$) traces

Ω	open domain in $\mathbb{R}^{\dim}$, $\dim = 1,2,3$
dim	spatial dimension
Γ	boundary $\partial\Omega$
Γ^{D}	part of the boundary with Dirichlet boundary conditions
Γ^{N}	part of the boundary with Neumann boundary conditions
Γ_-	inflow part of the boundary,
	i.e., $\Gamma_- := \{x \in \Gamma : \boldsymbol{\beta} \cdot \mathbf{n} < 0\}$ with prescribed velocity field $\boldsymbol{\beta}$
Γ_+	outflow part of the boundary,
	i.e., $\Gamma_+ := \{x \in \Gamma : \boldsymbol{\beta} \cdot \mathbf{n} \geq 0\}$ with prescribed velocity field $\boldsymbol{\beta}$
$\mathbf{x}$	spatial variable ($\mathbf{x} = (x_1, ..., x_{\dim})^T \in \mathbb{R}^{\dim}$)
t	time variable
$\partial_\bullet$	partial derivative, e.g. $\partial_x = \partial/\partial_x$
u_t	time derivative of function u, i.e., $u_t = \partial_t u$
∇	gradient operator, i.e., $\nabla = (\partial_{x_1}, ..., \partial_{x_{\dim}})^T \in \mathbb{R}^{\dim}$
div	divergence operator, i.e., $\operatorname{div} \mathbf{u} = \nabla \cdot \mathbf{u} = \sum_{i=1}^{\dim} \partial_{x_i} \mathbf{u}_i$, where $\mathbf{u}$ is vector-valued
Δ	Laplace operator, i.e. $\Delta u = \nabla \cdot \nabla u = \partial_{x_1 x_1} u + ... + \partial_{x_{\dim} x_{\dim}} u$
C	Constant, $C > 0$
$V, V', \|.\|_V$	Hilbert-space V (i.e., a complete space with scalar product $(\cdot,\cdot)_V$),
	its dual space V' and the norm $\|u\|_V^2 = (u,u)_V$,
$L^p(\Omega)$	$L^p(\Omega) := \{u : \Omega \to \mathbb{R} \mid \text{u is Lebesgue measurable}, \|u\|_{L^p(\Omega)} \leq \infty\}$
	$\|u\|_{L^p(\Omega)} = (\int_\Omega u^p \, \mathrm{d}\mathbf{x})^{\frac{1}{p}}$, $1 \leq p < \infty$; $\|u\|_{L^\infty(\Omega)} = \operatorname{ess\,sup}_\Omega \lvert u \rvert$
$\|.\|$	L^2-norm; if not obvious, the domain is added as subscript (e.g. $\|.\|_S$)
$\|.\|_\infty, \|.\|_{\infty,K}$	L^∞-norm on Ω and domain K, respectively
$D^k u(\mathbf{x})$	total derivative of order k of a function $u(\mathbf{x})$
$C_0^1(\Omega)$	space of functions which are continuously differentiable with compact support
$W^{k,p}(\Omega)$	Sobolev space, $W^{k,p}(\Omega) := \{u \in L^p(\Omega) : D^\alpha u \in L^p(\Omega) \forall \ \lvert\alpha\rvert \leq k\}$
$H^k(\Omega)$	Sobolev space with $p = 2 : H^k(\Omega) = W^{k,2}(\Omega)$
$\mathrm{Lip}(\Omega)$	space spanned by Lipschitz continous functions
$\|.\|_l, \|.\|_{l,K}$	H^l-norm on Ω and domain K, respectively; $l = 0$ indicates the L^2-norm
$\lvert.\rvert_l, \lvert.\rvert_{l,K}$	H^l-seminorm on Ω and domain K, respectively, e.g. $\lvert u\rvert_{1,K} = \|\nabla u\|_K$.

Table 2.1: General Notation for partial differential equations

$\mathcal{T}_h$	conforming triangulation of Ω_h
$\mathcal{S}_h$	set of element boundaries/sides (e.g. edges, faces) of triangulation $\mathcal{T}_h$
K, S	element of $\mathcal{T}_h$ and $\mathcal{S}_h$, respectively
S_K	all sides $S \in \mathcal{S}_h$ of element K ($S_K = \partial K$)
h	maximum element size $h = \max_{K \in \mathcal{T}_h} h_K$,
	as subscript: indicates a discrete quantity (e.g. V_h discrete space)
h_K, h_S	diameter of $K \in \mathcal{T}_h$ and $S \in \mathcal{S}_h$, respectively
$\|K\|$,$\|S\|$	measure of $K \in \mathcal{T}_h$ and $S \in \mathcal{S}_h$, respectively
$\hat{K}$	reference element to K
T_K	transformation from the reference element $\hat{K}$ to K
$\mathcal{S}_h^{\partial}$	set of boundary sides which cover the domain boundary $\partial\Omega$
$\mathcal{S}_h^{\text{int}}$	set of interior sides, i.e. $\mathcal{S}_h \setminus \mathcal{S}_h^{\partial}$
$\mathcal{S}_h^{\partial,D}$,$\mathcal{S}_h^{\partial,N}$	set of boundary sides which cover the boundary Γ^{D} and Γ^{N}, respectively
$\mathcal{S}_h^{\partial,-}$,$\mathcal{S}_h^{\partial,+}$	set of boundary sides which cover the boundary Γ_- and Γ_+, respectively
$\mathbf{n}_S, \mathbf{n}$	fixed unit vector normal to side $S \in \mathcal{S}_h$, $\mathbf{n} = (n_x, n_y)^T$,
	outward pointing for $S \in \mathcal{S}_h^{\partial}$
$\boldsymbol{\tau}$	tangential vector, i.e. $\boldsymbol{\tau} = (-n_y, n_x)^T$
$\mathbf{n}_{S_K}$	unit vector normal to side $S \in S_K$, outward pointing with respect to K
S_K^-, S_K^+	all sides $S \in S_K$ with $\boldsymbol{\beta} \cdot \mathbf{n}_{S_K} < 0$ and $\boldsymbol{\beta} \cdot \mathbf{n}_{S_K} \geq 0$, respectively
	and prescribed velocity field $\boldsymbol{\beta}$
$P^k(A)$	polynomials of total degree k on the set A
$Q^{p,q}(A)$,$Q^p(A)$	$Q^{p,q}(A) := \text{span}\{x^i y^j, (x,y) \in A, i = 0,...,p, j = 0,...,q\}$, $Q^p(A) := Q^{p,p}(A)$
N	number of (spatial) degrees of freedom
φ_i	basis functions of the discrete space V_h, $i = 1,...,N$
$I(= [0, t_{end}])$	time interval
n (superscript)	index for the time level t^n, e.g. $u^n = u(t^n)$
Δt	time step size (i.e. $\Delta t = t^{n+1} - t^n$)
V_h^k	$:= \left\{ v_h \in C(\bar{\Omega}_h) : \ v_h\|_K \circ T_K^{-1} \in P^k(\hat{K}) \right\}$
$V_h^{k,Q}$	$:= \left\{ v_h \in C(\bar{\Omega}_h) : \ v_h\|_K \circ T_K^{-1} \in Q^k(\hat{K}) \right\}$

Table 2.2: Notation for continuous and discontinuous finite elements

We fix the normal vector $\mathbf{n}_S$ of a side $S \in \mathcal{S}_h^{\text{int}}$ to one direction (either $\mathbf{n}_S = \mathbf{n}_{S_K}$ or $\mathbf{n}_S = \mathbf{n}_{S_{K'}}$ for $S \in S_K \cap S_{K'}$) and define

$$u^+(\mathbf{x}) := \lim_{\varepsilon \searrow 0} u_h(\mathbf{x} - \varepsilon \mathbf{n}_S), \quad u^-(\mathbf{x}) := \lim_{\varepsilon \searrow 0} u_h(\mathbf{x} + \varepsilon \mathbf{n}_S),$$

where $\mathbf{x} \in S$ and u_h is a scalar-valued function smooth enough to admit a possibly two-valued trace. This means for $S \in \mathcal{S}_h^{\text{int}}$ with $S \in S_K \cap S_{K'}$, that $u^+ = u^+_{S_K} = u^-_{S_{K'}}$, if $\mathbf{n}_S := \mathbf{n}_{S_K}$.

The jump and the mean value for an interior side $S \in \mathcal{S}_h^{\text{int}}$ and $\mathbf{x} \in S$ are defined by

$$[u](\mathbf{x}) := u^+(\mathbf{x}) - u^-(\mathbf{x}), \quad \{u\}(\mathbf{x}) := \frac{1}{2}\left(u^+(\mathbf{x}) + u^-(\mathbf{x})\right).$$

In the case of a boundary side, we define $[u] = u^+$ and $\{u\} = u^+$. For vector-valued functions jump and mean value are applied to each component separately. The jump of two functions can be written as

$$[uv] = [u]\{v\} + \{u\}[v].$$

Note that $[[u]] = 0$, $\{\{u\}\} = \{u\}$ and $[\{u\}] = 0$, $\{[u]\} = [u]$.

For simplicity we will use the following notation:

$$\int_{\mathcal{T}_h} = \sum_{K \in \mathcal{T}_h} \int_K, \qquad \int_{\mathcal{S}_h} = \sum_{S \in \mathcal{S}_h} \int_S.$$

The generalization of the integration by parts formula to the discontinuous case reads

$$\begin{aligned} \int_{\mathcal{T}_h} \nabla \cdot (\boldsymbol{\beta} u) v \, \mathrm{d}\mathbf{x} &= -\int_{\mathcal{T}_h} u \boldsymbol{\beta} \cdot \nabla v \, \mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h^{\text{int}}} \boldsymbol{\beta} \cdot \mathbf{n}_S [uv] \, \mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial}} \boldsymbol{\beta} \cdot \mathbf{n}_S uv \, \mathrm{d}\mathbf{s}, \\ &= -\int_{\mathcal{T}_h} u \boldsymbol{\beta} \cdot \nabla v \, \mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h^{\text{int}}} \boldsymbol{\beta} \cdot \mathbf{n}_S \left([u]\{v\} + \{u\}[v]\right) \mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial}} \boldsymbol{\beta} \cdot \mathbf{n}_S uv \, \mathrm{d}\mathbf{s}. \end{aligned} \tag{2.0.1}$$

We use $\lesssim$ instead of $\leq C$ for a constant C independent of h.

3

The continuous Galerkin method for scalar equations

In this chapter we give a short introduction to the continuous Galerkin method which is used to solve partial differential equations (PDEs). The idea of this method is to replace the solution space by a finite-dimensional subspace and obtain an approximate solution. This approach leads to the continuous finite element method (FEM) where the subspace is defined as a space of continuous piecewise polynomials [20].

In the following we show the derivation of the Galerkin problem for Poisson's equation and a linear advection equation. In a much more general framework and with certain assumptions we show uniqueness and existence of the solution and approximation properties of the finite dimensional subspace. In the context of finite elements, we show how boundary conditions can be imposed, namely in the weak and in the strong sense. Furthermore, we present error estimates for the discussed equations. At last we show the limitation of the continuous finite element method regarding stability when it comes to solving advection equations.

3.1. Poisson's equation

Let us consider Poisson's equation

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \Gamma^{\mathrm{D}}, \end{aligned} \tag{3.1.1}$$

where $f \in L^2(\Omega)$, $\Omega \subset \mathbb{R}^{\dim}$ is open and $\Gamma^{\mathrm{D}} = \partial\Omega$ is the Dirichlet boundary.

Multiplying (3.1.1) by a test function $\varphi \in H_0^1(\Omega)$ and integrating by parts lead to the following weak formulation:

$$\int_\Omega \nabla u \cdot \nabla \varphi \, \mathrm{d}\mathbf{x} = \int_\Omega f \varphi \, \mathrm{d}\mathbf{x} \quad \forall \varphi \in H_0^1(\Omega), \tag{3.1.2}$$

where u is called weak solution. The problem can be written in a more compact form:

$$\text{Find } u \in H_0^1(\Omega) \text{ such that } b(u,\varphi) = f(\varphi) \quad \forall \varphi \in H_0^1(\Omega) \tag{3.1.3}$$

with the bilinear form

$$b(u,\varphi) = \int_\Omega \nabla u \cdot \nabla \varphi \, \mathrm{d}\mathbf{x}$$

and the linear form

$$f(\varphi) = \int_\Omega f\varphi \, \mathrm{d}\mathbf{x}.$$

We say the problem is well-posed (according to Hadamard), if there exists one and only one solution and this solution depends continuously on the data for the problem (e.g., right-hand side).

By choosing a finite-dimensional subspace $V_h \subset H_0^1(\Omega)$ of dimension N we derive the Ritz-Galerkin approximation problem:

$$\text{Find } u_h \in V_h \text{ such that } b(u_h,\varphi_h) = f(\varphi_h) \quad \forall \varphi_h \in V_h. \tag{3.1.4}$$

The solution u_h is expressed as a linear combination of basis functions φ_i, $i = 1,..,N$ of V_h:

$$u_h = \sum_{i=1}^{N} u_i \varphi_i. \tag{3.1.5}$$

The Galerkin method is also referred to as a projection method due to the Galerkin orthogonality property:

$$b(u-u_h,\varphi_h) = 0 \quad \forall \varphi_h \in V_h. \tag{3.1.6}$$

In the context of continuous finite elements the space $V = H_0^1(\Omega)$ is approximated by the space of continuous, piecewise polynomial functions with zero boundary values

$$V \supset V_{h,0}^k := \left\{ v_h \in C(\bar{\Omega}_h) : v_h(x) = 0, \forall x \in \Gamma^{\mathrm{D}},\ v_h|_K \circ T_K^{-1} \in P^k(\hat{K}) \right\}, \tag{3.1.7}$$

where K is an element of a conforming triangulation $\mathcal{T}_h$ of Ω. The transformation between the physical element K and the reference element $\hat{K}$ is denoted by T_K^{-1}. Since the boundary conditions are directly incorporated into the space, we call them strong boundary conditions.

If we consider the space $V = H^1(\Omega)$ without imposing boundary conditions, we can approximate it by the space of continuous, piecewise polynomial functions

$$V \supset V_h^k := \left\{ v_h \in C(\bar{\Omega}_h) :\ v_h|_K \circ T_K^{-1} \in P^k(\hat{K}) \right\}. \tag{3.1.8}$$

Note that the space can also be defined for piecewise bilinear, biquadratic etc. functions, i.e., $V_h^{k,Q} := \left\{ v_h \in C(\bar{\Omega}_h) :\ v_h|_K \circ T_K^{-1} \in Q^k(\hat{K}) \right\}$. For simplicity, we will skip the superscript Q.

Definition 3.1: Conforming finite elements

Finite elements with $V_h \subset V$ are called conforming finite elements, otherwise, i.e., if $V_h \not\subset V$, non-conforming finite elements.

The continuous finite elements defining the space V_h^k or $V_{h,0}^k$ are conforming with respect to the space $H^1(\Omega)$ and $H_0^1(\Omega)$, respectively.

3.2. Existence and uniqueness

Let us now consider the variational problem

$$\text{Find } u \in V \text{ such that } b(u,\varphi) = f(\varphi) \quad \forall \varphi \in V \tag{3.2.1}$$

and its Galerkin approximation for a finite-dimensional subspace $V_h \subset V$

$$\text{Find } u_h \in V_h \text{ such that } b(u_h,\varphi_h) = f(\varphi_h) \quad \forall \varphi_h \in V_h \tag{3.2.2}$$

in a more general way than before, not necessarily based on the Poisson equation (3.1.1). To prove existence and uniqueness, the following assumptions are made:

Assumptions 1:

C.1 V is a Hilbert space,

C.2 the linear form $f : V \to \mathbb{R}$ is bounded (continuous):

$$\exists C_0 \geq 0 : |f(\varphi)| \leq C_0 \|\varphi\|_V, \quad \forall \varphi \in V,$$

C.3 the bilinear form $b : V \times V \to \mathbb{R}$ is bounded (continuous):

$$\exists C_1 \geq 0 : |b(u,\varphi)| \leq C_1 \|u\|_V \|\varphi\|_V, \quad \forall u, \varphi \in V,$$

C.4 the bilinear form $b : V \times V \to \mathbb{R}$ is V-coercive:

$$\exists C_2 > 0 : b(u,u) \geq C_2 \|u\|_V^2, \quad \forall u \in V.$$

Then the following Theorem holds:

Theorem 3.2: Lax-Milgram Theorem

If Assumptions 1 are fulfilled, then for arbitrary $f \in V'$ problem (3.2.1) has a unique solution $u \in V$ and

$$\|u\|_V \leq \frac{1}{C_2} \|f\|_{V'}, \tag{3.2.3}$$

*where C_2 is the constant from **C.4**.*

Note that the previous theorem also gives well-posedness of the problem. The proof of the Lax-Milgram theorem can be found, e.g., in [66]. In the case of the Poisson problem (3.1.1) for $V = H_0^1(\Omega)$ Assumptions 1 hold and therefore existence and uniqueness of a solution $u \in V$ follows from application of the Lax-Milgram theorem. In the context of finite elements we call the method stable, if inequality (3.2.3) holds for the finite element solution $u_h \in V_h$ and a constant C_2, which does not depend on h.

The following Lemma gives information about how well the Galerkin solution u_h of (3.2.2) approximates the solution u of (3.2.1) (see, e.g., [17]):

Lemma 3.3: Céa's Lemma

If Assumptions 1 are fulfilled and $u \in V$ solves (3.2.1), then the solution $u_h \in V_h \subset V$ of (3.2.2) satisfies

$$\|u - u_h\|_V \leq \frac{C_1}{C_2} \min_{v_h \in V_h} \|u - v_h\|_V, \tag{3.2.4}$$

*with the constants C_1 and C_2 from **C.3** and **C.4**.*

If we consider a bilinear form $a : V_1 \times V_2 \to \mathbb{R}$ mapping from two different Hilbert spaces V_1 and V_2, $V_1 \neq V_2$, e.g., in the case of non-symmetric bilinear forms, we have the variational problem:

$$\text{Find } u \in V_1 \text{ such that } a(u, \varphi) = f(\varphi) \quad \forall \varphi \in V_2. \tag{3.2.5}$$

For the finite-dimensional subspaces $V_{1,h} \subset V_1$ and $V_{2,h} \subset V_2$ we obtain the Galerkin problem:

$$\text{Find } u_h \in V_{1,h} \text{ such that } a(u_h, \varphi_h) = f(\varphi_h) \quad \forall \varphi_h \in V_{2,h}. \tag{3.2.6}$$

In this case the coercivity assumption **C.4** can be replaced by

$$\exists C_3 > 0 \ : \quad \sup_{\varphi \in V_2 \setminus \{0\}} \frac{a(u, \varphi)}{\|\varphi\|_{V_2}} \geq C_3 \|u\|_{V_1}, \quad \forall u \in V_1, \tag{3.2.7}$$

$$\forall \varphi \in V_2, \varphi \neq 0, \exists u \in V_1 \ : \quad a(u, \varphi) \neq 0. \tag{3.2.8}$$

Equation (3.2.7) is often called the inf-sup condition and is equivalent to

$$\exists C_3 > 0 \ : \quad \inf_{u \in V_1 \setminus \{0\}} \sup_{\varphi \in V_2 \setminus \{0\}} \frac{b(u, \varphi)}{\|\varphi\|_{V_2} \|u\|_{V_1}} \geq C_3. \tag{3.2.9}$$

We remark that in the finite dimensional case, if $\dim(V_1) = \dim(V_2)$, condition (3.2.7) implies (3.2.8) [33].

Using (3.2.7) and (3.2.8) instead of the coercivity assumption **C.4** gives a more general form of the Lax-Milgram theorem [6, 73]:

Theorem 3.4: Generalized Lax-Milgram Theorem

Let V_1, V_2 be Hilbert spaces and $a : V_1 \times V_2 \to \mathbb{R}$ be a bounded bilinear form, i.e.,

$$\exists C_1 \geq 0 : |a(u, \varphi)| \leq C_1 \|u\|_{V_1} \|\varphi\|_{V_2}, \quad \forall u \in V_1, \varphi \in V_2. \tag{3.2.10}$$

For arbitrary linear bounded functionals $f \in V_2'$ we consider the variational problem (3.2.5). Then the following statements are equivalent:

1. *Conditions (3.2.7) and (3.2.8) hold.*
2. *Problem (3.2.5) has a unique solution $u \in V_1$ with*

$$\|u\|_{V_1} \leq \frac{1}{C_3} \|f\|_{V_2'}. \tag{3.2.11}$$

Theorem 3.4 also holds if V_1 is a Banach space and V_2 is a reflexive Banach space. In this form, the theorem is often called the *Banach-Nečas-Babuška Theorem*.

In contrast to the coercivity assumption **C.4** which holds also on subspaces $V_h \subset V$ we need to define a discrete inf-sup condition for the Galerkin problem (3.2.6)

$$\exists C_h > 0 \ : \quad \sup_{\varphi_h \in V_{2,h} \setminus \{0\}} \frac{a(u_h, \varphi_h)}{\|\varphi_h\|_{V_2}} \geq C_h \|u_h\|_{V_1}, \quad \forall u_h \in V_{1,h}. \tag{3.2.12}$$

Using this condition Céa's lemma can also be written in a more general form (see, e.g., [33]):

Lemma 3.5: Generalized Céa's Lemma

Let V_1, V_2 be Hilbert spaces, $V_{1,h} \subset V_1$ and $V_{2,h} \subset V_2$ finite dimensional subspaces with $\dim(V_{1,h}) = \dim(V_{2,h})$ and $a : V_1 \times V_2 \to \mathbb{R}$ a bounded bilinear form. We assume that conditions (3.2.7),(3.2.8) and the discrete inf-sup condition (3.2.12) hold. Then the variational problem (3.2.5) and the Galerkin problem (3.2.6) have unique solutions $u \in V_1$ and $u_h \in V_{1,h}$ and the following estimate holds

$$\|u - u_h\|_{V_1} \leq \left(1 + \frac{C_1}{C_h}\right) \min_{v_h \in V_{1,h}} \|u - v_h\|_{V_1}, \tag{3.2.13}$$

with the constants C_1 and C_h from (3.2.10) and (3.2.12).

In the context of finite elements inequalities (3.2.4) and (3.2.13) are usually the starting point to derive a priori estimates of the form $\|u - u_h\| \leq Ch^p |u|$ where p is called the order of convergence and $\|.\|$ and $|.|$ are a norm and seminorm, respectively. In the following, we will demonstrate this approach for the Poisson equation.

3.3. A priori error estimates for Poisson's equation

Let us consider inequality (3.2.4) which holds in the case of the Poisson problem (3.1.1) and $V = H_0^1(\Omega)$, $V_h = V_{h,0}^k$. The minimizer v_h on the right-hand side can be exchanged with an arbitrary $v_h \in V_h$ and the inequality stays true. For that reason, we introduce an interpolation operator $I_h : V \to V_h$ and replace v_h by the interpolant $I_h(u)$.

Definition 3.6: Interpolation operator I_h

Let φ_i, $i = 1, ..., N$, be nodal basis functions of V_h with

$$\varphi_i(\mathbf{x}_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise,} \end{cases}$$

where $\mathbf{x}_j$, $j = 1, .., N$, are the corresponding nodal points of V_h. Then the interpolation operator $I_h : V \to V_h$ is defined as

$$I_h(u) := \sum_{i=1}^{N} u(\mathbf{x}_i)\varphi_i. \tag{3.3.1}$$

For the interpolation on V_h^k the following estimate can be derived (see, e.g., [37, 54].

Lemma 3.7: Interpolation estimate

Let $\mathcal{T}_h$ be a conforming triangulation of Ω and I_h an interpolation operator on V_h^k for $k \geq 1$. Then there exists a constant C_I , independent of h, such that for $0 \leq m \leq k+1$ we have

$$\|u - I_h(u)\|_m \leq C_I h^{k+1-m} |u|_{k+1}, \quad \forall u \in H^{k+1}(\Omega). \tag{3.3.2}$$

This leads directly to an H^1-error estimate for the Poisson problem (3.1.1). In order to obtain an estimate in the L^2-norm one usually applies a duality argument from Aubin and Nitsche (see, e.g., [54]) and derives the following a priori error estimates:

Theorem 3.8: A priori error estimates

Let $u \in H_0^{k+1}(\Omega)$ and $u_h \in V_{h,0}^k$ be the solutions to (3.1.2) and (3.1.4), respectively. Then

$$\|u - u_h\| \leq Ch^{k+1} |u|_{k+1}, \tag{3.3.3}$$

$$\|u - u_h\|_1 \leq Ch^k |u|_{k+1}. \tag{3.3.4}$$

Let us now briefly summarize the results obtained for the Poisson equation. By applying the Lax-Milgram theorem, we have seen, that there exists a unique solution of the weak problem (3.1.3). The corresponding finite element method is stable and has also a unique solution. The application of Céa's lemma gives an estimate for the error between the weak solution and the Galerkin solution. This estimate can also be applied to the finite element solution and is the starting point to derive the a priori estimates (3.3.3) and (3.3.4).

3.4. Linear advection

Let us now consider the linear advection equation

$$\nabla \cdot (\boldsymbol{\beta} u) + cu = f \quad \text{in } \Omega, \tag{3.4.1}$$

$$u = g \quad \text{on } \Gamma_-, \tag{3.4.2}$$

with $f \in L^2(\Omega)$, $\boldsymbol{\beta} \in [\mathrm{Lip}(\Omega)]^{\dim}$, $c \in L^\infty(\Omega)$ and $g \in L^2(\Gamma_-)$. Note that $\mathrm{Lip}(\Omega) \subset W^{1,\infty}(\Omega)$ (details can be found in [17]) and therefore, there exist constants $C_{\boldsymbol{\beta}_i} > 0$ such that

$$\|\nabla \boldsymbol{\beta}_i\|_{[L^\infty(\Omega)]^{\dim}} \leq C_{\boldsymbol{\beta}_i}, \quad i = 1, .., \dim. \tag{3.4.3}$$

We also assume

$$c(\mathbf{x}) + \frac{1}{2} \nabla \cdot \boldsymbol{\beta}(\mathbf{x}) \geq c_0 > 0, \quad \forall x \in \Omega \tag{3.4.4}$$

to guarantee L^2-coercivity (see Lemma 3.9), and

$$\sup_{\mathbf{x} \in \Omega} |c(\mathbf{x}) + \nabla \cdot \boldsymbol{\beta}(\mathbf{x})| =: c_1 < \infty \tag{3.4.5}$$

to obtain boundedness.

For the variational formulation we define the function space

$$H^{1,\boldsymbol{\beta}}(\Omega) = \{u \in L^2(\Omega) : \nabla \cdot (\boldsymbol{\beta} u) \in L^2(\Omega)\}, \tag{3.4.6}$$

which is called *graph space* [26] and is equipped with the scalar product

$$(u, v)_{H^{1,\boldsymbol{\beta}}(\Omega)} := (u, v)_{L^2(\Omega)} + (\boldsymbol{\beta} \cdot \nabla u, \boldsymbol{\beta} \cdot \nabla v)_{L^2(\Omega)} \tag{3.4.7}$$

and the norm $\|u\|_{1,\boldsymbol{\beta}}^2 := \|u\|_{H^{1,\boldsymbol{\beta}}(\Omega)}^2 = (u, u)_{H^{1,\boldsymbol{\beta}}(\Omega)}$.

Following [37] we will show, how boundary conditions can be weakly imposed. At first, we multiply (3.4.1) by a test function $\varphi \in H^{1,\boldsymbol{\beta}}(\Omega)$ and integrate by parts over the domain Ω

$$\int_\Omega -(\boldsymbol{\beta} u) \cdot \nabla \varphi + cu\varphi \, \mathrm{d}\mathbf{x} + \int_{\Gamma_+} \boldsymbol{\beta}_n u \varphi \, \mathrm{d}\mathbf{s} = \int_\Omega f \varphi \, \mathrm{d}\mathbf{x} - \int_{\Gamma_-} \boldsymbol{\beta}_n g \varphi \, \mathrm{d}\mathbf{s} \quad \forall \varphi \in H^{1,\boldsymbol{\beta}}(\Omega), \tag{3.4.8}$$

where $\boldsymbol{\beta}_n = \boldsymbol{\beta} \cdot \mathbf{n}$. A second integration by parts gives the variational formulation with weak boundary conditions:

$$\text{Find } u \in H^{1,\boldsymbol{\beta}}(\Omega) \text{ such that } a(u,\varphi) = f(\varphi) \quad \forall \varphi \in H^{1,\boldsymbol{\beta}}(\Omega), \tag{3.4.9}$$

where

$$a(u,\varphi) = \int_\Omega (\nabla \cdot (\boldsymbol{\beta} u) + cu)\,\varphi \,\mathrm{d}\mathbf{x} - \int_{\Gamma_-} \boldsymbol{\beta}_n u \varphi \,\mathrm{d}\mathbf{s}, \tag{3.4.10}$$

$$f(\varphi) = \int_\Omega f\varphi \,\mathrm{d}\mathbf{x} - \int_{\Gamma_-} \boldsymbol{\beta}_n g \varphi \,\mathrm{d}\mathbf{s}. \tag{3.4.11}$$

For the discretization with continuous finite elements we consider the space of continuous, piecewise polynomial functions (3.1.8). This gives the discrete problem:

$$\text{Find } u_h \in V_h^k \text{ such that } a(u_h,\varphi_h) = f(\varphi_h) \quad \forall \varphi_h \in V_h^k. \tag{3.4.12}$$

We will now check if Assumptions 1 are fulfilled. We have that $H^{1,\boldsymbol{\beta}}(\Omega)$ equipped with the scalar product $(\cdot,\cdot)_{H^{1,\boldsymbol{\beta}}(\Omega)}$ is a Hilbert space (see, e.g., [26]). It is also easy to verify that the linear form $f(\varphi)$ and the bilinear form $a(u,\varphi)$ are continuous in $H^{1,\boldsymbol{\beta}}(\Omega)$. However, the following lemma shows that we obtain coercivity only with respect to the L^2-norm but not in the $H^{1,\boldsymbol{\beta}}$-norm.

Lemma 3.9: L^2-coercivity

The bilinear form $a(\cdot,\cdot)$ as defined in (3.4.10) is coercive with respect to the L^2-norm, i.e., there exists a constant $C > 0$ such that

$$a(u,u) \geq C\|u\|^2_{L^2(\Omega)}, \quad \forall u \in H^{1,\boldsymbol{\beta}}(\Omega). \tag{3.4.13}$$

Proof. Choosing $\varphi = u$ in (3.4.10) gives

$$a(u,u) = \int_\Omega \left(\nabla \cdot (\boldsymbol{\beta} u)u + cu^2\right) \mathrm{d}\mathbf{x} - \int_{\Gamma_-} \boldsymbol{\beta}_n u^2 \,\mathrm{d}\mathbf{s}.$$

The first part of the volume integral can be simplified in the following way:

$$\begin{aligned}
\int_\Omega \nabla \cdot (\boldsymbol{\beta} u)u \,\mathrm{d}\mathbf{x} &= \int_\Omega (\nabla \cdot \boldsymbol{\beta})u^2 + (\boldsymbol{\beta} \cdot \nabla u)u \,\mathrm{d}\mathbf{x} = \int_\Omega (\nabla \cdot \boldsymbol{\beta})u^2 + \frac{1}{2}\boldsymbol{\beta} \cdot \nabla u^2 \,\mathrm{d}\mathbf{x} \\
&= \int_\Omega (\nabla \cdot \boldsymbol{\beta})u^2 + \frac{1}{2}\left(\nabla \cdot (\boldsymbol{\beta} u^2) - (\nabla \cdot \boldsymbol{\beta})u^2\right) \mathrm{d}\mathbf{x} \\
&= \int_\Omega \frac{1}{2}(\nabla \cdot \boldsymbol{\beta})u^2 + \frac{1}{2}\nabla \cdot (\boldsymbol{\beta} u^2) \,\mathrm{d}\mathbf{x}.
\end{aligned}$$

Applying the divergence theorem gives

$$\begin{aligned}
a(u,u) &= \int_\Omega \left(\frac{1}{2}(\nabla \cdot \boldsymbol{\beta}) + c\right) u^2 + \frac{1}{2}\nabla \cdot (\boldsymbol{\beta} u^2) \,\mathrm{d}\mathbf{x} - \int_{\Gamma_-} \boldsymbol{\beta}_n u^2 \,\mathrm{d}\mathbf{s} \\
&= \int_\Omega \left(\frac{1}{2}(\nabla \cdot \boldsymbol{\beta}) + c\right) u^2 \,\mathrm{d}\mathbf{x} + \frac{1}{2}\int_\Gamma \boldsymbol{\beta}_n u^2 \,\mathrm{d}\mathbf{s} - \int_{\Gamma_-} \boldsymbol{\beta}_n u^2 \,\mathrm{d}\mathbf{s} \\
&= \int_\Omega \left(\frac{1}{2}(\nabla \cdot \boldsymbol{\beta}) + c\right) u^2 \,\mathrm{d}\mathbf{x} + \frac{1}{2}\int_{\Gamma_+} \boldsymbol{\beta}_n u^2 \,\mathrm{d}\mathbf{s} - \frac{1}{2}\int_{\Gamma_-} \boldsymbol{\beta}_n u^2 \,\mathrm{d}\mathbf{s}.
\end{aligned}$$

Taking into account that $\boldsymbol{\beta}_n \geq 0$ on Γ_+, $\boldsymbol{\beta}_n < 0$ on Γ_- and assumption (3.4.4) we obtain

$$\begin{aligned} a(u,u) &\geq c_0\|u\|^2 + \frac{1}{2}\int_\Gamma |\boldsymbol{\beta}_n|\, u^2 \,\mathrm{d}\mathbf{s} \\ &\geq c_0\|u\|^2, \end{aligned}$$

which completes the proof. □

Since we do not have $H^{1,\boldsymbol{\beta}}$-coercivity, we cannot control the derivative part $\|\boldsymbol{\beta}\cdot\nabla u\|$, which directly influences the stability [37]. This will be motivated in the following example.

Example 3.10: 1D convection problem

Following [33] we will illustrate the problem concerning stability for

$$\beta u'(x) + u(x) = f(x), \quad x \in \Omega := (0,1), \tag{3.4.14}$$
$$u(0) = 0, \tag{3.4.15}$$

with constant velocity $\beta > 0$ and $f \in L^2(\Omega)$.
We define the bilinear form $a : V_1 \times V_2 \to \mathbb{R}$ by

$$a(u,\varphi) = \int_0^1 \beta u'\varphi + u\varphi \,\mathrm{d}x.$$

Let $V_1 = \{v \in H^1(\Omega) : v(0) = 0\}$ and $V_2 = L^2(\Omega)$. We obtain the problem:

$$\text{Find } u \in V_1 \text{ such that } a(u,\varphi) = f(\varphi) \quad \forall \varphi \in V_2. \tag{3.4.16}$$

By application of Theorem 3.4 with conditions (3.2.7) and (3.2.8) we obtain existence and uniqueness of a solution $u \in V_1$. For details see [33].

Now we discretize (3.4.16) by using the Galerkin method with the continuous linear finite element space $V_{h,0}^1 \subset V_1$ as defined in (3.1.7). The discretized problem is as follows

$$\text{Find } u_h \in V_{h,0}^1 \text{ such that } a(u_h,\varphi_h) = f(\varphi_h) \quad \forall \varphi_h \in V_{h,0}^1 \subset V_2. \tag{3.4.17}$$

To show existence and uniqueness of the discrete solution $u_h \in V_{h,0}^1$ the discrete inf-sup condition (3.2.12) needs to be fulfilled. In [33] it was shown that the inf-sup condition (3.2.12) holds with $C_h = ch$ where $c > 0$ is independent of h. For $h \to 0$, C_h deteriorates, which indicates the instability of the method. Note that this constant is also used in equation (3.2.11) where for $h \to 0$ the right-hand side goes to infinity and therefore the solution can blow up. In numerical experiments this instability manifests itself in oscillations, see Chapter 6.

Example 3.10 showed that Theorem 3.4 can be applied to prove the existence of a unique solution $u \in V_1$ for the 1D case, if the test functions are in $L^2(\Omega)$. An extension of this example to the multidimensional space was presented in [26], where it was proven that problem (3.4.9) is well-posed, even for test functions in $H^{1,\boldsymbol{\beta}}(\Omega)$. Similar to the 1D case, the discrete inf-sup condition (3.2.12) can be derived. However, the constant ε_h depends on h also in the multidimensional case, which means that the finite element method is not stable.

If the solution u is smooth enough, the following a priori error estimate holds:

Theorem 3.11: A priori error estimate

For $k \geq 1$ let $u \in H^{k+1}(\Omega)$ be the solution to (3.4.1) and $u_h \in V_h^k$ the solution to (3.4.12). Then

$$\|u - u_h\|_{L^2(\Omega)} \leq Ch^k \, |u|_{H^{k+1}(\Omega)} \,. \tag{3.4.18}$$

The proof can be found, e.g., in [37]. In comparison to the a priori error estimates of the Poisson equation in Theorem 3.8 we get one power of h less in the L^2-norm. Furthermore, we do not obtain any H^1-error estimate [33].

We also mention that in order to stabilize the continuous Galerkin method for convection-dominated problems and improve the order of convergence in estimate (3.4.18) the so-called streamline diffusion method can be used (see, e.g., [37]). In this method, the test function φ in (3.4.9) is replaced by $\varphi_h + \sigma h \boldsymbol{\beta} \cdot \nabla \varphi_h$ with $\sigma > 0$ on Ω and by φ_h on Γ. Assuming that $\nabla \cdot \boldsymbol{\beta} = 0$, c is constant and h small enough, it can be shown that the resulting bilinear form is coercive with respect to the $\|.\|_{\mathrm{SD}}$-norm defined by

$$\|u\|_{\mathrm{SD}}^2 := h\|\boldsymbol{\beta} \cdot \nabla u\|_{L^2(\Omega)}^2 + \|u\|_{L^2(\Omega)}^2 + \int_\Gamma |\boldsymbol{\beta}_n| \, u^2 \, \mathrm{d}\mathbf{s}, \tag{3.4.19}$$

which is a norm on $H^{1,\boldsymbol{\beta}}(\Omega)$. This leads to a stable method for which a priori estimates with half an order more than in (3.4.18) can be obtained

$$\|u - u_h\|_{\mathrm{SD}} \leq Ch^{k+\frac{1}{2}} \, |u|_{H^{k+1}(\Omega)} \,. \tag{3.4.20}$$

In summary, we have seen that the standard continuous finite element method is stable and achieves optimal L^2-and H^1-error estimates in the context of Poisson's equation. For the linear advection equation only suboptimal error estimates can be derived and stabilization to control streamline derivatives is necessary. Therefore, the standard continuous finite element method is better suited for elliptic problems (e.g. Poisson's equation) than for hyperbolic problems (e.g. linear advection).

4

The discontinuous Galerkin method for scalar equations

In this chapter we give a brief introduction to the discontinuous Galerkin (DG) method. In contrast to the continuous Galerkin method the weak formulation is derived for a mesh-dependent space, the so-called broken Sobolev space consisting of functions which may be discontinuous across element boundaries. In the context of advection equations, where solution may have jumps, the DG method can resolve discontinuities, if those are captured by an element edge. The broken Sobolev space can be approximated by discontinuous piecewise polynomials, which leads to the discontinuous finite element method (DG FEM).

In the following we explain the DG method for the same equations as in the previous chapter, i.e., for the linear advection and the Poisson equation. Since the derivation of the DG method is more intuitive for the advection case and was also first defined in this context [76], we will begin with that one. We will see that the upwind DG formulation applied to advection equations has better stability properties than the standard continuous Galerkin method in the sense that it gives not only L^2-stability but also control over the streamline derivatives and the jumps across element boundaries [51]. For the Poisson equation we present two penalty methods, namely the symmetric interior penalty Galerkin (SIPG) method [2] and the non-symmetric interior penalty Galerkin (NIPG) method [77], which enforce continuity of the solution by a penalty term. In this context another method is introduced as well, the Baumann-Oden method [8, 10], which is similar to the NIPG method but free of penalty terms.

4.1. Broken Sobolev spaces

At first, we introduce the so-called broken Sobolev space [26, 28].

Definition 4.1: Broken Sobolev space

Let $\mathcal{T}_h$ be a conforming triangulation of Ω. The broken Sobolev space is defined by

$$W^{k,p}(\mathcal{T}_h) := \{v \in L^p(\Omega) : v|_K \in W^{k,p}(K) \forall K \in \mathcal{T}_h\}, \tag{4.1.1}$$

where $k \geq 0$ is an integer and $1 \leq p \leq \infty$ a real number.

For $p = 2$, we have $H^k(\mathcal{T}_h) = W^{k,2}(\mathcal{T}_h)$ with the corresponding seminorm

$$|v|^2_{H^k(\mathcal{T}_h)} := \sum_{K \in \mathcal{T}_h} |v|^2_{H^k(K)}, \quad v \in H^k(\mathcal{T}_h), \tag{4.1.2}$$

where $|.|^2_{H^k(K)}$ is the Sobolev seminorm on K.

Furthermore, we define the broken counterpart of (3.4.6)

$$H^{1,\boldsymbol{\beta}}(\mathcal{T}_h) = \{v \in L^2(\Omega) : v|_K \in H^{1,\boldsymbol{\beta}}(K) \forall K \in \mathcal{T}_h \}. \tag{4.1.3}$$

We remark that $W^{m,p}(\Omega) \subset W^{m,p}(\mathcal{T}_h)$ and $H^{1,\boldsymbol{\beta}}(\Omega) \subset H^{1,\boldsymbol{\beta}}(\mathcal{T}_h)$.
Note that the operators on the broken Sobolev spaces should also be defined as broken operators (see, e.g., [26]).

Definition 4.2: Broken gradient

The broken gradient $\nabla_h : W^{1,p}(\mathcal{T}_h) \to [L^p(\Omega)]^{\dim}$ is defined such that

$$(\nabla_h v)|_K := \nabla(v|_K), \quad \forall K \in \mathcal{T}_h, \forall v \in W^{1,p}(\mathcal{T}_h). \tag{4.1.4}$$

Similarly, the broken Laplace operator $\Delta_h : H^2(\mathcal{T}_h) \to L^2(\mathcal{T}_h)$ can be defined.

For a proper definition of the broken divergence operator, we first introduce the space $H(\mathrm{div};\Omega)$ and its broken counterpart.

Definition 4.3: $H(\mathrm{div};\Omega)$ and $H(\mathrm{div};\mathcal{T}_h)$

We define the following function space

$$H(\mathrm{div};\Omega) := \{\boldsymbol{\tau} \in [L^2(\Omega)]^{\dim} : \nabla \cdot \boldsymbol{\tau} \in L^2(\Omega)\} \tag{4.1.5}$$

and its broken counterpart

$$H(\mathrm{div};\mathcal{T}_h) := \{\boldsymbol{\tau} \in [L^2(\Omega)]^{\dim} : \boldsymbol{\tau}|_K \in H(\mathrm{div};K) \forall K \in \mathcal{T}_h\}. \tag{4.1.6}$$

This leads to the definition of the broken divergence operator:

Definition 4.4: Broken divergence operator

The broken divergence operator $\nabla_h \cdot : [H(\mathrm{div};\mathcal{T}_h)]^{\dim} \to L^2(\Omega)$ is defined such that

$$(\nabla_h \cdot \boldsymbol{\tau})|_K := \nabla \cdot (\boldsymbol{\tau}|_K), \quad \forall K \in \mathcal{T}_h, \forall \boldsymbol{\tau} \in H(\mathrm{div};\mathcal{T}_h). \tag{4.1.7}$$

For simplicity, we skip the subscript h in the following.

In order to check, if a function of the broken space belongs to the usual Sobolev space, i.e., $W^{1,p}(\Omega)$ and $H(\mathrm{div};\Omega)$, respectively, we can apply the following lemmata [26].

Lemma 4.5: Characterization of functions in $W^{1,p}(\Omega)$

Let $1 \leq p \leq \infty$. A function $v \in W^{1,p}(\mathcal{T}_h)$ belongs to $W^{1,p}(\Omega)$, if and only if

$$[v] = 0, \quad \forall S \in \mathcal{S}_h^{\text{int}}. \tag{4.1.8}$$

Lemma 4.6: Characterization of functions in $H(\text{div};\Omega)$

A function $\boldsymbol{\tau} \in H(\text{div};\mathcal{T}_h) \cap [W^{1,1}(\mathcal{T}_h)]^{dim}$ belongs to $H(\text{div};\Omega)$, if and only if

$$[\boldsymbol{\tau}] \cdot \mathbf{n}_S = 0, \quad \forall S \in \mathcal{S}_h^{\text{int}}. \tag{4.1.9}$$

We have now defined the spaces related to the DG method and can now derive the weak formulations for the linear advection and the Poisson equation.

4.2. Linear advection

Let us consider the linear advection equation:

$$\nabla \cdot (\boldsymbol{\beta} u) + cu = f \quad \text{in } \Omega, \tag{4.2.1}$$

$$u = g \quad \text{on } \Gamma_-, \tag{4.2.2}$$

where $f \in L^2(\Omega)$, $\boldsymbol{\beta} \in [\text{Lip}(\Omega)]^{\text{dim}}$, $c \in L^\infty(\Omega)$ and $g \in L^2(\Gamma_-)$. We assume that (3.4.4) and (3.4.5) hold.

4.2.1. The upwind DG formulation

In the following, we will show, how the upwind DG formulation is derived. Let K be an element of $\mathcal{T}_h$. We multiply (4.2.1) by a test function $\varphi \in H^{1,\boldsymbol{\beta}}(\mathcal{T}_h)$ and integrate over K:

$$\int_K (\nabla \cdot (\boldsymbol{\beta} u) + cu)\, \varphi \, \mathrm{d}\mathbf{x} = \int_K f \varphi \, \mathrm{d}\mathbf{x}. \tag{4.2.3}$$

Integration by parts leads to

$$\int_K -(\boldsymbol{\beta} u) \cdot \nabla \varphi + cu\varphi \, \mathrm{d}\mathbf{x} + \int_{S_K} \boldsymbol{\beta}_{n_K} u \varphi \, \mathrm{d}\mathbf{s} = \int_K f \varphi \, \mathrm{d}\mathbf{x}, \tag{4.2.4}$$

where $\boldsymbol{\beta}_{n_K} = \boldsymbol{\beta} \cdot \mathbf{n}_{S_K}$.

The crucial point in DG methods is the treatment of $\int_{S_K} \boldsymbol{\beta}_{n_K} u \varphi \, \mathrm{d}\mathbf{s}$, since the solution is, in general, discontinuous across the edges. One possible way is the upwind formulation. Hereby, we impose the external trace $u_{S_K}^-$ on the inflow part of S_K

$$\int_K -(\boldsymbol{\beta} u) \cdot \nabla \varphi + cu\varphi \, \mathrm{d}\mathbf{x} + \int_{S_K^+} \boldsymbol{\beta}_{n_K} u_{S_K}^+ \varphi_{S_K}^+ \, \mathrm{d}\mathbf{s} + \int_{S_K^-} \boldsymbol{\beta}_{n_K} u_{S_K}^- \varphi_{S_K}^+ \, \mathrm{d}\mathbf{s} = \int_K f \varphi \, \mathrm{d}\mathbf{x}. \tag{4.2.5}$$

For simplicity, we will skip the subscript S_K of the external and internal traces. Now we sum over all elements $K \in \mathcal{T}_h$:

$$\int_{\mathcal{T}_h} -(\boldsymbol{\beta} u) \cdot \nabla \varphi + cu\varphi \, \mathrm{d}\mathbf{x} + \sum_{K \in \mathcal{T}_h} \left(\int_{S_K^+ \setminus \Gamma} \boldsymbol{\beta}_{n_K} u^+ \varphi^+ \, \mathrm{d}\mathbf{s} + \int_{S_K^- \setminus \Gamma} \boldsymbol{\beta}_{n_K} u^- \varphi^+ \, \mathrm{d}\mathbf{s} \right) + \int_{\mathcal{S}_h^{\partial,+}} \boldsymbol{\beta}_n u \varphi \, \mathrm{d}\mathbf{s}$$

$$= \int_{\mathcal{T}_h} f \varphi \, \mathrm{d}\mathbf{x} - \int_{\mathcal{S}_h^{\partial,-}} \boldsymbol{\beta}_n g \varphi \, \mathrm{d}\mathbf{s}. \tag{4.2.6}$$

Introducing the upwind value

$$\hat{u}(\mathbf{x}) = \begin{cases} u^-(\mathbf{x}) & \text{if } \boldsymbol{\beta}(\mathbf{x})\cdot\mathbf{n}_S < 0 \wedge \mathbf{x} \notin \partial\Omega, \\ g(\mathbf{x}) & \text{if } \boldsymbol{\beta}(\mathbf{x})\cdot\mathbf{n}_S < 0 \wedge \mathbf{x} \in \Gamma_-, \\ u^+(\mathbf{x}) & \text{otherwise,} \end{cases} \tag{4.2.7}$$

which gives

$$\sum_{K\in\mathcal{T}_h}\left(\int_{S_K^+\setminus\Gamma}\boldsymbol{\beta}_{n_K}u^+\varphi^+\,\mathrm{d}\mathbf{s}+\int_{S_K^-\setminus\Gamma}\boldsymbol{\beta}_{n_K}u^-\varphi^+\,\mathrm{d}\mathbf{s}\right)=\sum_{K\in\mathcal{T}_h}\int_{S_K\setminus\Gamma}\boldsymbol{\beta}\cdot\mathbf{n}_S\hat{u}\varphi^+\,\mathrm{d}\mathbf{s}, \tag{4.2.8}$$

the problem related to (4.2.6) can be written in compact form:

$$\text{Find } u\in H^{1,\boldsymbol{\beta}}(\mathcal{T}_h) \text{ such that } a_{\mathrm{DG}}(u,\varphi)=f_{\mathrm{DG}}(\varphi)\quad \forall\varphi\in H^{1,\boldsymbol{\beta}}(\mathcal{T}_h), \tag{4.2.9}$$

where

$$a_{\mathrm{DG}}(u,\varphi)=\int_{\mathcal{T}_h}-(\boldsymbol{\beta}u)\cdot\nabla\varphi+cu\varphi\,\mathrm{d}\mathbf{x}+\sum_{K\in\mathcal{T}_h}\int_{S_K\setminus\Gamma}\boldsymbol{\beta}\cdot\mathbf{n}_S\hat{u}\varphi^+\,\mathrm{d}\mathbf{s}+\int_{\mathcal{S}_h^{\partial,+}}\boldsymbol{\beta}_n u\varphi\,\mathrm{d}\mathbf{s}, \tag{4.2.10}$$

$$f_{\mathrm{DG}}(\varphi)=\int_{\mathcal{T}_h}f\varphi\,\mathrm{d}\mathbf{x}-\int_{\mathcal{S}_h^{\partial,-}}\boldsymbol{\beta}_n g\varphi\,\mathrm{d}\mathbf{s}. \tag{4.2.11}$$

In the context of finite elements the broken Sobolev space is approximated by the discontinuous finite element space

$$V_{DG,h}^k:=\left\{v_h\in L^2(\Omega_h):\ v_h|_K\circ T_K^{-1}\in P^k(\hat{K})\right\}. \tag{4.2.12}$$

As in the continuous case, a similar definition can be derived for bilinear, biquadratic, etc. functions. For simplicity we use $V_{DG,h}^k$ to indicate both spaces. Furthermore, we introduce the DG-norm

$$\|u_h\|_{\mathrm{DG}}:=\sqrt{c_0\|u_h\|^2+\frac{1}{2}\int_{\mathcal{S}_h^{\partial}}|\boldsymbol{\beta}_n|u_h^2\,\mathrm{d}\mathbf{s}+\frac{1}{2}\int_{\mathcal{S}_h^{\mathrm{int}}}|\boldsymbol{\beta}_n|[u_h]^2\,\mathrm{d}\mathbf{s}}. \tag{4.2.13}$$

Note that $V_{DG,h}^k\subset H^m(\mathcal{T}_h)\subset H^1(\mathcal{T}_h)\subset H^{1,\boldsymbol{\beta}}(\mathcal{T}_h), m>1$ but $V_{DG,h}^k\not\subset H^{1,\boldsymbol{\beta}}(\Omega)$ [37]. Therefore discontinuous finite elements are non-conforming. Hence Céa's lemma cannot be applied.

The discretized finite element problem is given by:

$$\text{Find } u_h\in V_{DG,h}^k \text{ such that } a_{\mathrm{DG}}(u_h,\varphi_h)=f_{\mathrm{DG}}(\varphi_h)\quad \forall\varphi_h\in V_{DG,h}^k. \tag{4.2.14}$$

The DG solution u_h is piecewise polynomial like the solution of the continuous FEM . However, the difference is that u_h may be discontinuous across element boundaries.

We will now come back to the treatment of the boundary integral $\int_{S_K}\boldsymbol{\beta}_{n_K}u\varphi\,\mathrm{d}\mathbf{s}$.

4.2.2. Numerical fluxes

The expression $\boldsymbol{\beta}\cdot\mathbf{n}_S\hat{u}$ is called upwind flux and defines the numerical flux

$$H_{up}(u^+,u^-,\mathbf{n}_S):=\boldsymbol{\beta}\cdot\mathbf{n}_S\hat{u}(\mathbf{x}). \tag{4.2.15}$$

The upwind flux $H_{up}(u^+,u^-,\mathbf{n})$ can also be replaced by some other numerical flux $H(u^+,u^-,\mathbf{n})$ to approximate $\boldsymbol{\beta}\cdot\mathbf{n}_S u$ at $S\in\mathcal{S}_h^{\mathrm{int}}$. For example, there is the mean value flux $H_{mv}(u^+,u^-,\mathbf{n}):=$

$\boldsymbol{\beta}\cdot\mathbf{n}\{u\}$, also called centered or central flux (see, e.g., [26, 37, 39]). This leads to a more general weak formulation:

$$\text{Find } u \in H^{1,\boldsymbol{\beta}}(\mathcal{T}_h) \text{ such that } a_{\mathrm{DG}}(u,\varphi) = f_{\mathrm{DG}}(\varphi) \quad \forall \varphi \in H^{1,\boldsymbol{\beta}}(\mathcal{T}_h), \tag{4.2.16}$$

where

$$a_{\mathrm{DG}}(u,\varphi) = \int_{\mathcal{T}_h} -(\boldsymbol{\beta}u)\cdot\nabla\varphi + cu\varphi \,\mathrm{d}\mathbf{x} + \sum_{K\in\mathcal{T}_h}\int_{S_K\setminus\Gamma} H(u^+,u^-,\mathbf{n}_{S_K})\varphi^+ \,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial,+}} \boldsymbol{\beta}_n u\varphi \,\mathrm{d}\mathbf{s}, \tag{4.2.17}$$

$$f_{\mathrm{DG}}(\varphi) = \int_{\mathcal{T}_h} f\varphi \,\mathrm{d}\mathbf{x} - \int_{\mathcal{S}_h^{\partial,-}} \boldsymbol{\beta}_n g\varphi \,\mathrm{d}\mathbf{s}, \tag{4.2.18}$$

for an arbitrary numerical flux $H(u^+,u^-,\mathbf{n})$.

We assume that the numerical flux $H(u^+,u^-,\mathbf{n})$ is consistent and conservative.

Definition 4.7: Consistent and conservative flux

The numerical flux $H(u^+,u^-,\mathbf{n})$ is

1. **consistent**, if
$$H(u,u,\mathbf{n}) = \boldsymbol{\beta}\cdot\mathbf{n}u. \tag{4.2.19}$$
2. **conservative**, if
$$H(u_1,u_2,\mathbf{n}) = -H(u_2,u_1,-\mathbf{n}). \tag{4.2.20}$$

The upwind flux $H_{up}(u^+,u^-,\mathbf{n})$ and the mean value flux $H_{mv}(u^+,u^-,\mathbf{n})$ are conservative and consistent. However, the use of upwind fluxes leads to improved stability properties compared to the use of mean value fluxes. For details see, e.g., [37].

Since $\mathbf{n}_{S_K} = -\mathbf{n}_{S_{K'}}$ for a common edge $S \in \partial K \cap \partial K'$, the sum $\sum_{K\in\mathcal{T}_h}\int_{S_K\setminus\Gamma}$ can be simplified for conservative fluxes:

$$\sum_{K\in\mathcal{T}_h}\int_{S_K\setminus\Gamma} H(u^+,u^-,\mathbf{n}_{S_K})\varphi^+ \,\mathrm{d}\mathbf{s} = \int_{\mathcal{S}_h^{\mathrm{int}}} H(u^+,u^-,\mathbf{n}_S)[\varphi] \,\mathrm{d}\mathbf{s}, \tag{4.2.21}$$

where $\mathbf{n}_S$ is a fixed normal vector to side $S \in \mathcal{S}_h^{\mathrm{int}}$. Note that for a side $S = \partial K_2 \cap \partial K_1$ the fixed normal vector $\mathbf{n}_S$ is either $\mathbf{n}_S = \mathbf{n}_{S_{K_1}}$ or $\mathbf{n}_S = \mathbf{n}_{S_{K_2}}$ ($\mathbf{n}_{S_{K_1}} = -\mathbf{n}_{S_{K_2}}$) depending on how the normal vector was fixed.

The following lemmata (see, e.g., [37]) show that a consistent numerical flux leads to a consistent discretization and a conservative flux leads to a conservative discretization.

Lemma 4.8: Consistency

The discretization (4.2.17) - (4.2.18) is consistent, i.e., the exact solution $u_{ex} \in H^{1,\boldsymbol{\beta}}(\Omega)$ of (4.2.1) satisfies

$$a_{\mathrm{DG}}(u_{ex},\varphi) = f_{\mathrm{DG}}(\varphi) \quad \forall \varphi \in H^{1,\boldsymbol{\beta}}(\mathcal{T}_h),$$

if and only if (4.2.19) is fulfilled.

We remark that consistency leads to Galerkin orthogonality

$$a_{\mathrm{DG}}(u_{ex}-u,\varphi) = 0 \quad \forall \varphi \in H^{1,\boldsymbol{\beta}}(\mathcal{T}_h), \tag{4.2.22}$$

where u_{ex} is the exact solution of (4.2.1) and u the weak solution of (4.2.16).

Lemma 4.9: Conservation

The discretization (4.2.17) - (4.2.18) with $c = 0$ is conservative, i.e.,

$$\int_{\mathcal{S}_h^{\partial,+}} \boldsymbol{\beta}_{n_S} u \,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial,-}} \boldsymbol{\beta}_{n_S} g \,\mathrm{d}\mathbf{s} = \int_{\mathcal{T}_h} f \,\mathrm{d}\mathbf{x},$$

if and only if (4.2.20) is fulfilled.

The application of these lemmata shows that the DG discretization with upwind or mean value flux is consistent and conservative. Note that the continuous Galerkin discretization is also consistent and conservative.

4.2.3. Coercivity and a priori error estimates for the upwind DG formulation

In the following we assume $H(u^+, u^-, \mathbf{n}) = H_{up}(u^+, u^-, \mathbf{n})$. The following coercivity result indicates the stability of the upwind formulation:

Lemma 4.10: Coercivity

The bilinear form (4.2.17) is coercive with respect to the DG-norm:

$$a_{\mathrm{DG}}(v, v) \geq \|v\|_{\mathrm{DG}}^2 \quad \forall v \in H^{1,\boldsymbol{\beta}}(\mathcal{T}_h). \tag{4.2.23}$$

Proof. Applying the discontinuous integration by parts formula (2.0.1) to equation (4.2.10) and taking into account (4.2.21) give:

$$\int_{\mathcal{T}_h} -(\boldsymbol{\beta} v)\cdot\nabla\varphi + cv\varphi \,\mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h^{\mathrm{int}}} \boldsymbol{\beta}_{n_S} \hat{v}[\varphi] \,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial,+}} \boldsymbol{\beta}_{n_S} v\varphi \,\mathrm{d}\mathbf{s} \tag{4.2.24}$$

$$= \int_{\mathcal{T}_h} \nabla\cdot(\boldsymbol{\beta} v)\varphi + cv\varphi \,\mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h^{\mathrm{int}}} \boldsymbol{\beta}_{n_S} \left(\hat{v}[\varphi] - [v\varphi]\right) \mathrm{d}\mathbf{s} - \int_{\mathcal{S}_h^{\partial,-}} \boldsymbol{\beta}_{n_S} v\varphi \,\mathrm{d}\mathbf{s}. \tag{4.2.25}$$

We sum up (4.2.24) and (4.2.25), choose $\varphi = v$ and obtain:

$$2a_{\mathrm{DG}}(v, v) = \int_{\mathcal{T}_h} -(\boldsymbol{\beta} v)\cdot\nabla v + \nabla\cdot(\boldsymbol{\beta} v)v + 2cv^2 \,\mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h^{\mathrm{int}}} \boldsymbol{\beta}_{n_S} \left(2\hat{v}[v] - [v^2]\right) \mathrm{d}\mathbf{s}$$
$$+ \int_{\mathcal{S}_h^{\partial,+}} \boldsymbol{\beta}_{n_S} v^2 \,\mathrm{d}\mathbf{s} - \int_{\mathcal{S}_h^{\partial,-}} \boldsymbol{\beta}_{n_S} v^2 \,\mathrm{d}\mathbf{s}.$$

Let us now consider the different terms. For the volume integral we have:

$$\int_{\mathcal{T}_h} -(\boldsymbol{\beta} v)\cdot\nabla v + \nabla\cdot(\boldsymbol{\beta} v)v \,\mathrm{d}\mathbf{x} = \int_{\mathcal{T}_h} -(\boldsymbol{\beta} v)\cdot\nabla v + (\nabla\cdot\boldsymbol{\beta})v^2 + (\boldsymbol{\beta}\cdot\nabla v)v \,\mathrm{d}\mathbf{x}$$
$$= \int_{\mathcal{T}_h} (\nabla\cdot\boldsymbol{\beta})v^2 \,\mathrm{d}\mathbf{x}.$$

Using $\boldsymbol{\beta}_{n_S} < 0$ on Γ_- gives for the boundary integrals:

$$\int_{\mathcal{S}_h^{\partial,+}} \boldsymbol{\beta}_{n_S} v^2 \,\mathrm{d}\mathbf{s} - \int_{\mathcal{S}_h^{\partial,-}} \boldsymbol{\beta}_{n_S} v^2 \,\mathrm{d}\mathbf{s} = \int_{\mathcal{S}_h^{\partial,+}} \left|\boldsymbol{\beta}_{n_S}\right| v^2 \,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial,-}} \left|\boldsymbol{\beta}_{n_S}\right| v^2 \,\mathrm{d}\mathbf{s}$$
$$= \int_{\mathcal{S}_h^{\partial}} |\boldsymbol{\beta}_n| v^2 \,\mathrm{d}\mathbf{s}.$$

For the edge integrals we have to make the following distinction:

- For $\boldsymbol{\beta}_{n_S} < 0$ we have $\hat{v} = v^-$, which yields

$$\int_{\mathcal{S}_h^{\text{int}}} \boldsymbol{\beta}_{n_S} \left(2v^-[v] - [v^2]\right) \,\mathrm{d}\mathbf{s} = \int_{\mathcal{S}_h^{\text{int}}} \boldsymbol{\beta}_{n_S} \left(2v^-(v^+ - v^-) - (v^+v^+ - v^-v^-)\right) \,\mathrm{d}\mathbf{s}$$
$$= -\int_{\mathcal{S}_h^{\text{int}}} \boldsymbol{\beta}_{n_S} [v]^2 \,\mathrm{d}\mathbf{s} = \int_{\mathcal{S}_h^{\text{int}}} \left|\boldsymbol{\beta}_{n_S}\right| [v]^2 \,\mathrm{d}\mathbf{s}.$$

- For $\boldsymbol{\beta}_{n_S} \geq 0$ we have $\hat{v} = v^+$ and obtain

$$\int_{\mathcal{S}_h^{\text{int}}} \boldsymbol{\beta}_{n_S} \left(2v^+[v] - [v^2]\right) \,\mathrm{d}\mathbf{s} = \int_{\mathcal{S}_h^{\text{int}}} \boldsymbol{\beta}_{n_S} [v]^2 \,\mathrm{d}\mathbf{s} = \int_{\mathcal{S}_h^{\text{int}}} \left|\boldsymbol{\beta}_{n_S}\right| [v]^2 \,\mathrm{d}\mathbf{s}.$$

Overall, we find that

$$2a_{\text{DG}}(v,v) = \int_{\mathcal{T}_h} (2c + \nabla \cdot \boldsymbol{\beta}) v^2 \,\mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h^{\partial}} |\boldsymbol{\beta}_n| v^2 \,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\text{int}}} |\boldsymbol{\beta}_n| [v]^2 \,\mathrm{d}\mathbf{s}.$$

Applying (3.4.4) completes the proof. □

Coercivity leads to well-posedness of the discrete problem (4.2.14). Inserting the discrete solution $u_h \in V_{DG,h}^k$ into (4.2.23) yields

$$\|u_h\|_{\text{DG}}^2 \leq a_{\text{DG}}(u_h, u_h) = f_{\text{DG}}(u_h) \leq C_f \|u_h\|_{\text{DG}}. \tag{4.2.26}$$

This directly leads to the stability estimate

$$\|u_h\|_{\text{DG}} \leq C_f, \tag{4.2.27}$$

where C_f is a constant which is independent of h.

We can even obtain a stronger stability result by introducing the norm

$$|||u|||_{\text{DG}}^2 := \|u\|_{\text{DG}}^2 + \sum_{K \in \mathcal{T}_h} \delta_K \|\boldsymbol{\beta} \cdot \nabla u\|_K^2, \quad \delta_K = \frac{h_K}{\|\boldsymbol{\beta}\|_\infty}, \tag{4.2.28}$$

which gives control over the streamline derivatives. Following [26], we assume that

$$h \leq \frac{\|\boldsymbol{\beta}\|_\infty}{\max(\|c\|_\infty, C_{\boldsymbol{\beta}})}, \tag{4.2.29}$$

where $C_{\boldsymbol{\beta}} := \max_{1 \leq i \leq \dim} C_{\boldsymbol{\beta}_i}$ and $C_{\boldsymbol{\beta}_i}$ defined by (3.4.3). For this norm a discrete inf-sup condition can be derived:

Lemma 4.11: Discrete inf-sup condition

There exists a mesh-independent constant $\gamma > 0$ such that

$$\sup_{v_h \in V_{DG,h}^k \setminus \{0\}} \frac{a_{\mathrm{DG}}(u_h, \varphi_h)}{|||\varphi_h|||_{\mathrm{DG}}} \geq \gamma |||u_h|||_{\mathrm{DG}}, \quad \forall u_h \in V_{DG,h}^k. \tag{4.2.30}$$

This condition leads directly to stability in the $|||.|||_{\mathrm{DG}}$-norm.

Furthermore, one can also derive a priori error estimates (see, e.g., [26] or in the case of constant velocity field $\boldsymbol{\beta}$ [50, 51]):

Theorem 4.12: A priori error estimate

Let $u \in H^{k+1}(\Omega)$ be the solution to (3.4.9) and $u_h \in V_{DG,h}^k$ the solution to (4.2.14) with upwind flux $H_{up}(u^+, u^-, \mathbf{n}_S)$. Then the following a priori estimates hold

$$\|u - u_h\|_{\mathrm{DG}} \leq C' h^{k+\frac{1}{2}} \|u\|_{H^{k+1}}, \tag{4.2.31}$$

$$|||u - u_h|||_{\mathrm{DG}} \leq C'' h^{k+\frac{1}{2}} \|u\|_{H^{k+1}}, \tag{4.2.32}$$

where C' and C'' are independent of h.

Estimate (4.2.31) also holds in the L^2-norm. Therefore, we gain a half power of h in comparison to the estimate of Theorem 3.11 for the continuous Galerkin method. Note that estimate (4.2.32) is equivalent to (3.4.20), which holds for the streamline diffusion method.

We have now seen how the linear advection equation can be discretized by the upwind DG method and how well the DG solution can approximate the exact solution.

In the following we will show, how diffusive terms can be discretized by the DG method.

4.3. Poisson's equation

Let us consider Poisson's equation

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{4.3.1}$$

where $f \in L^2(\Omega)$.

For a proper analysis we introduce the space for the traces of functions in $H^1(\mathcal{T}_h)$ [4]:

$$T(\mathcal{S}_h) := \prod_{K \in \mathcal{T}_h} L^2(S_K).$$

Note that functions $g \in T(\mathcal{S}_h)$ have two values on $S \in \mathcal{S}_h^{\mathrm{int}}$, namely g^+ and g^-, and one value on $S \in \mathcal{S}_h^{\partial}$.

Following [4], we write the Poisson problem as a first-order system

$$\left.\begin{aligned} \mathbf{b} &= \nabla u \\ -\nabla \cdot \mathbf{b} &= f \end{aligned}\right\} \quad \text{in } \Omega, \tag{4.3.2}$$

$$u = 0 \quad \text{on } \partial\Omega, \tag{4.3.3}$$

where u is called potential and ∇u diffusive flux. Multiplying by test functions $\varphi \in H^2(\mathcal{T}_h)$ and $\boldsymbol{\tau} \in H(\mathrm{div};\mathcal{T}_h)$, respectively, and integrating by parts over $K \in \mathcal{T}_h$ gives

$$\int_K \mathbf{b}\cdot\boldsymbol{\tau}\,\mathrm{d}\mathbf{x} = -\int_K u\nabla\cdot\boldsymbol{\tau}\,\mathrm{d}\mathbf{x} + \int_{S_K} u\mathbf{n}_{S_K}\cdot\boldsymbol{\tau}\,\mathrm{d}\mathbf{s}, \tag{4.3.4}$$

$$\int_K \mathbf{b}\cdot\nabla\varphi\,\mathrm{d}\mathbf{x} = \int_K f\varphi\,\mathrm{d}\mathbf{x} + \int_{S_K} \mathbf{b}\cdot\mathbf{n}_{S_K}\varphi\,\mathrm{d}\mathbf{s}. \tag{4.3.5}$$

Similarly to the advection case, the treatment of the edge integrals $\int_{S_K} u\mathbf{n}_{S_K}\cdot\boldsymbol{\tau}\,\mathrm{d}\mathbf{s}$ and $\int_{S_K}\mathbf{b}\cdot\mathbf{n}_{S_K}\varphi\,\mathrm{d}\mathbf{s}$ is a crucial point, since u and $\mathbf{b}$ are, in general, discontinuous across element boundaries. Therefore, we replace u and $\mathbf{b}$ in the edge integrals by numerical flux functions

$$\hat{u} : H^1(\mathcal{T}_h) \to T(\mathcal{S}_h)$$

and

$$\hat{\mathbf{b}} : H^2(\mathcal{T}_h)\times H(\mathrm{div};\mathcal{T}_h) \to [T(\mathcal{S}_h)]^{\dim}.$$

Possible definitions of those functions will be discussed later.

If we sum over all elements $K \in \mathcal{T}_h$, we obtain

$$\int_{\mathcal{T}_h} \mathbf{b}\cdot\boldsymbol{\tau}\,\mathrm{d}\mathbf{x} = -\int_{\mathcal{T}_h} u\nabla\cdot\boldsymbol{\tau}\,\mathrm{d}\mathbf{x} + \sum_{K\in\mathcal{T}_h}\int_{S_K} \hat{u}\mathbf{n}_{S_K}\cdot\boldsymbol{\tau}^+\,\mathrm{d}\mathbf{s}, \tag{4.3.6}$$

$$\int_{\mathcal{T}_h} \mathbf{b}\cdot\nabla\varphi\,\mathrm{d}\mathbf{x} = \int_{\mathcal{T}_h} f\varphi\,\mathrm{d}\mathbf{x} + \sum_{K\in\mathcal{T}_h}\int_{S_K} \hat{\mathbf{b}}\cdot\mathbf{n}_{S_K}\varphi^+\,\mathrm{d}\mathbf{s}. \tag{4.3.7}$$

As in the advection case we assume that the numerical fluxes are consistent and conservative.

Definition 4.13: Consistent and conservative fluxes

The numerical fluxes $\hat{u}$ and $\hat{\mathbf{b}}$ are

1. **consistent**, if
$$\hat{u}(v) = v|_S, \quad \hat{\mathbf{b}}(v,\nabla v) = \nabla v|_S \quad \forall S \in \mathcal{S}_h,$$
for functions $v \in H_0^1(\Omega)\cap W^{2,1}(\Omega)$.
2. **conservative**, if they are single-valued on S, $\forall S \in \mathcal{S}_h$.

Note that functions $v \in H_0^1(\Omega)\cap W^{2,1}(\Omega)$ satisfy [26]

$$[v] = 0 \quad \text{and} \quad [\nabla v]\cdot\mathbf{n}_S = 0, \quad \forall S \in \mathcal{S}_h^{\mathrm{int}}. \tag{4.3.8}$$

In the following, we will make use of

$$\sum_{K\in\mathcal{T}_h}\int_{S_K} q\mathbf{v}\cdot\mathbf{n}_{S_K}\,\mathrm{d}\mathbf{s} = \int_{\mathcal{S}_h}[q]\mathbf{n}_S\cdot\{\mathbf{v}\}\,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\mathrm{int}}}\{q\}\mathbf{n}_S\cdot[\mathbf{v}]\,\mathrm{d}\mathbf{s}, \tag{4.3.9}$$

where $q \in T(\mathcal{S}_h)$ and $\mathbf{v} \in [T(\mathcal{S}_h)]^2$.

Inserting this relation in (4.3.6) - (4.3.7) yields

$$\int_{\mathcal{T}_h} \mathbf{b}\cdot\boldsymbol{\tau}\,\mathrm{d}\mathbf{x} = -\int_{\mathcal{T}_h} u\nabla\cdot\boldsymbol{\tau}\,\mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h}[\hat{u}]\mathbf{n}_S\cdot\{\boldsymbol{\tau}\}\,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\mathrm{int}}}\{\hat{u}\}\mathbf{n}_S\cdot[\boldsymbol{\tau}]\,\mathrm{d}\mathbf{s}, \tag{4.3.10}$$

$$\int_{\mathcal{T}_h} \mathbf{b}\cdot\nabla\varphi\,\mathrm{d}\mathbf{x} = \int_{\mathcal{T}_h} f\varphi\,\mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h}\{\hat{\mathbf{b}}\}\cdot\mathbf{n}_S[\varphi]\,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\mathrm{int}}}[\hat{\mathbf{b}}]\cdot\mathbf{n}_S\{\varphi\}\,\mathrm{d}\mathbf{s}. \tag{4.3.11}$$

We now replace $\boldsymbol{\tau}$ by $\nabla\varphi$ in (4.3.10) and integrate by parts on the right-hand side:

$$\int_{\mathcal{T}_h} \mathbf{b}\cdot\nabla\varphi\,\mathrm{d}\mathbf{x} = \int_{\mathcal{T}_h} \nabla u\cdot\nabla\varphi\,\mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h} [\hat{u}-u]\mathbf{n}_S\cdot\{\nabla\varphi\}\,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\mathrm{int}}} \{\hat{u}-u\}\mathbf{n}_S\cdot[\nabla\varphi]\,\mathrm{d}\mathbf{s}. \tag{4.3.12}$$

Substituting (4.3.12) into (4.3.11) yields the primal flux formulation:

$$\text{Find } u\in H^2(\mathcal{T}_h): \quad b_{\mathrm{DG}}(u,\varphi) = \int_\Omega f\varphi\,\mathrm{d}\mathbf{x} \quad \forall\,\varphi\in H^2(\mathcal{T}_h), \tag{4.3.13}$$

where $b_{\mathrm{DG}}(\cdot,\cdot): H^2(\mathcal{T}_h)\times H^2(\mathcal{T}_h)$ is defined by

$$\begin{aligned} b_{\mathrm{DG}}(u,\varphi) := &\int_{\mathcal{T}_h} \nabla u\cdot\nabla\varphi\,\mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h} [\hat{u}-u]\mathbf{n}_S\cdot\{\nabla\varphi\} - \{\hat{\mathbf{b}}\}\cdot\mathbf{n}_S[\varphi]\,\mathrm{d}\mathbf{s} \\ &+ \int_{\mathcal{S}_h^{\mathrm{int}}} \{\hat{u}-u\}\mathbf{n}_S\cdot[\nabla\varphi] - [\hat{\mathbf{b}}]\cdot\mathbf{n}_S\{\varphi\}\,\mathrm{d}\mathbf{s} \end{aligned} \tag{4.3.14}$$

Consistency of the numerical fluxes leads to consistency of the primal flux formulation (see, e.g., [37]):

Lemma 4.14: Consistency

The discretization (4.3.13) of (4.3.1) is consistent, i.e., the exact solution $u\in H^2(\Omega)$ of (4.3.1) satisfies

$$b_{\mathrm{DG}}(u,\varphi) = \int_{\mathcal{T}_h} f\varphi\,\mathrm{d}\mathbf{x} \quad \forall\,\varphi\in H^2(\mathcal{T}_h), \tag{4.3.15}$$

if and only if the numerical fluxes $\hat{u}$ and $\hat{\mathbf{b}}$ are consistent.

There are different possibilities to define numerical fluxes $\hat{u}$ and $\hat{\mathbf{b}}$. In Table 4.1 some definitions for consistent and conservative fluxes are listed. Note that $u=0$ on $\partial\Omega$, which simplifies the definition of the fluxes on the boundary.

Method	$\hat{u}(u)$	$\hat{\mathbf{b}}(u,\nabla u)$	additional term $\alpha(u,\varphi)$
Symmetric interior penalty (SIPG)	$\{u\}$	$\{\nabla u\}$	$\int_{\mathcal{S}_h}\mu[u][\varphi]$
Non-symmetric interior penalty (NIPG)	$\{u\}+[u]$	$\{\nabla u\}$	$\int_{\mathcal{S}_h}\mu[u][\varphi]$
Baumann-Oden (BO)	$\{u\}+[u]$	$\{\nabla u\}$	-

Table 4.1: Numerical fluxes for the Poisson equation

Using these definitions the following bilinear forms are obtained

$$b_{SIPG}(u,\varphi) := \int_{\mathcal{T}_h} \nabla u\cdot\nabla\varphi\,\mathrm{d}\mathbf{x} - \int_{\mathcal{S}_h} ([u]\mathbf{n}_S\cdot\{\nabla\varphi\} + \{\nabla u\}\cdot\mathbf{n}_S[\varphi])\,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h}\mu[u][\varphi]\,\mathrm{d}\mathbf{s}, \tag{4.3.16}$$

$$b_{NIPG}(u,\varphi) := \int_{\mathcal{T}_h} \nabla u\cdot\nabla\varphi\,\mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h} [u]\mathbf{n}_S\cdot\{\nabla\varphi\} - \{\nabla u\}\cdot\mathbf{n}_S[\varphi]\,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h}\mu[u][\varphi]\,\mathrm{d}\mathbf{s}, \tag{4.3.17}$$

$$b_{BO}(u,\varphi) := \int_{\mathcal{T}_h} \nabla u\cdot\nabla\varphi\,\mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h} [u]\mathbf{n}_S\cdot\{\nabla\varphi\} - \{\nabla u\}\cdot\mathbf{n}_S[\varphi]\,\mathrm{d}\mathbf{s}, \tag{4.3.18}$$

where

$$\mu : S_h \to \mathbb{R} \text{ is a penalty weighting function defined by } \mu = \frac{\eta_S}{h_S}, \forall S \in S_h, \eta_S > 0. \tag{4.3.19}$$

Note that the penalty term $\int_{\mathcal{S}_h} \mu[u][\varphi]$ enforces continuity of the solution. Consistency of the presented methods follows directly from Lemma 4.14.

The discretization of (4.3.13) by discontinuous finite elements leads to the following problem

$$\text{Find } u_h \in V_{DG,h}^k \text{ such that } b_{\mathrm{DG}}(u_h, \varphi_h) = \int_{\mathcal{T}_h} f\varphi_h \,\mathrm{d}\mathbf{x} \quad \forall \varphi_h \in V_{DG,h}^k. \tag{4.3.20}$$

We remark that the Baumann-Oden method is not stable for $k \leq 1$. However, numerical results indicate, that stability is given for $k \geq 2$ [10].

For all presented methods the same optimal H^1-error estimate can be derived. For the L^2-error we obtain optimal rates only for the SIPG method. The estimates are summarized in the following theorem (see, e.g., [4, 75, 77]):

Theorem 4.15: A priori error estimates

Let $u \in H^{k+1}(\Omega)$ be the exact solution of (4.3.1) and $u_h \in V_{DG,h}^k$ the solution of (4.3.20). Then

1. *For $b_{\mathrm{DG}}(\cdot,\cdot) = b_{SIPG}(\cdot,\cdot)$:*

$$\|u - u_h\| \leq Ch^{k+1} |u|_{k+1}, \tag{4.3.21}$$
$$\|u - u_h\|_1 \leq Ch^{k} |u|_{k+1}, \tag{4.3.22}$$

2. *For $b_{\mathrm{DG}}(\cdot,\cdot) = b_{NIPG}(\cdot,\cdot)$:*

$$\|u - u_h\| \leq Ch^{k} |u|_{k+1}, \tag{4.3.23}$$
$$\|u - u_h\|_1 \leq Ch^{k} |u|_{k+1}, \tag{4.3.24}$$

3. *For $b_{\mathrm{DG}}(\cdot,\cdot) = b_{BO}(\cdot,\cdot)$ and $k \geq 2$:*

$$\|u - u_h\| \leq Ch^{k} |u|_{k+1}, \tag{4.3.25}$$
$$\|u - u_h\|_1 \leq Ch^{k} |u|_{k+1}. \tag{4.3.26}$$

To prove this theorem we define the following norms:

$$\|v\|_\mu^2 = |v|_{H^1(\mathcal{T}_h)}^2 + \int_{\mathcal{S}_h} \tilde{\mu}[v]^2 \,\mathrm{d}\mathbf{s}, \tag{4.3.27}$$

$$|||v|||_\mu^2 = \|v\|_\mu^2 + \int_{\mathcal{S}_h} \tilde{\mu}^{-1} (\{\nabla v\} \cdot \mathbf{n}_S)^2 \,\mathrm{d}\mathbf{s}, \tag{4.3.28}$$

where

$$\tilde{\mu} = \begin{cases} \mu & \text{for } NIPG, SIPG, \\ 1 & \text{for } BO. \end{cases} \tag{4.3.29}$$

In the continuous case we made use of the interpolation operator I_h (see Def. 3.6) to derive a priori error estimates. However, this operator is only well-defined for $u \in C(\bar{\Omega})$, so that we cannot use it in the discontinuous case. Therefore we introduce the L^2 projection for functions $u \in L^2(\Omega)$:

Definition 4.16: L^2 projection

Let $u \in L^2(\Omega)$ and $k \geq 0$. Then the L^2 projection onto $V_{DG,h}^k$ is defined by

$$\int_{\mathcal{T}_h} \pi_h^k u \, v_h \, \mathrm{d}\mathbf{x} = \int_{\mathcal{T}_h} u \, v_h \, \mathrm{d}\mathbf{x} \quad \forall v_h \in V_{DG,h}^k, \tag{4.3.30}$$

where π_h^k is the L^2 projection operator. Furthermore, it satisfies the local projection property:

$$\int_K \pi_h^k u \, v_h \, \mathrm{d}\mathbf{x} = \int_K u \, v_h \, \mathrm{d}\mathbf{x} \quad \forall v_h \in V_{DG,h}^k \text{ and } \forall K \in \mathcal{T}_h. \tag{4.3.31}$$

For simplicity we write π_h neglecting the superscript k, if it is clear onto which space we project.

Furthermore, the following estimates hold (see, e.g., [37, 51]):

Lemma 4.17: Local approximation estimates for the L^2 projection

Let $k \geq 0$ and $\pi_h u$ be the L^2 projection onto $V_{DG,h}^k$ as defined in (4.3.30). For $u_{|_K} \in H^{r_K+1}(K)$, $r_k \geq 0$ for $K \in \mathcal{T}_h$ we have

$$\|u - \pi_h u\|_{m,K} \leq C h_K^{l_k+1-m} \, |u|_{l_k+1,K}, \tag{4.3.32}$$

where $l_k = \min(r_K, k)$ and $m \leq l_k + 1$. Furthermore, for $u \in H^{k+1}(K)$ the following estimate holds

$$\|u - \pi_h u\|_{\partial K} \leq C h^{k+\frac{1}{2}} \, |u|_{k+1,K}. \tag{4.3.33}$$

We will now prove Theorem 4.15.

Proof. The proof for the Baumann-Oden method can be found in [77]. We will focus on the proof of the L^2-error estimate for the SIPG method following [4]. At first, we prove

$$|||u - u_h|||_\mu \leq C h^k \, |u|_{H^{k+1}(\mathcal{T}_h)}, \tag{4.3.34}$$

which is satisfied by the SIPG and the NIPG method.

Let π_h be the L^2 projection operator as defined in (4.3.30). Then we have

$$|||u - u_h|||_\mu \leq |||u - \pi_h u|||_\mu + |||\pi_h u - u_h|||_\mu. \tag{4.3.35}$$

Using the trace inequalities as defined in Theorem A.8 we obtain the following estimate for the first term with $\mu = \frac{\eta}{h}$ as defined in (4.3.19)

$$\begin{aligned}
|||u - \pi_h u|||_\mu^2 &= |u - \pi_h u|_{1,\mathcal{T}_h}^2 + \int_{\mathcal{S}_h} \mu [u - \pi_h u]^2 \, \mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h} \mu^{-1} (\{\nabla (u - \pi_h u)\} \cdot \mathbf{n}_S)^2 \\
&\lesssim |u - \pi_h u|_{1,\mathcal{T}_h}^2 + \sum_{K \in \mathcal{T}_h} \mu \left(\frac{1}{h} \|u - \pi_h u\|_K^2 + h \, |u - \pi_h u|_{1,K}^2 \right) \\
&\quad + \sum_{K \in \mathcal{T}_h} \mu^{-1} \left(\frac{1}{h} |u - \pi_h u|_{1,K}^2 + h \, |u - \pi_h u|_{2,K}^2 \right) \\
&\lesssim |u - \pi_h u|_{1,\mathcal{T}_h}^2 + \frac{1}{h^2} \|u - \pi_h u\|_{\mathcal{T}_h}^2 + h^2 \, |u - \pi_h u|_{2,\mathcal{T}_h}^2.
\end{aligned}$$

Applying the local error estimates of the L^2 projection as in (4.3.32) yields

$$|||u-\pi_h u|||_\mu \leq Ch^k\,|u|_{k+1,\mathcal{T}_h}\,. \tag{4.3.36}$$

The bilinear form $b_{\mathrm{DG}}(\cdot,\cdot)$ is bounded (cf. Lemma 5.12) and in the case of the SIPG and NIPG method coercive (cf. Lemma 5.13). If we also take into account Galerkin orthogonality which is based on the consistency of the method (cf. Lemma 4.14), we can approximate the second term in (4.3.35) as follows:

$$\begin{aligned} C|||\pi_h u-u_h|||_\mu^2 &\leq b_{\mathrm{DG}}(\pi_h u-u_h,\pi_h u-u_h)\\ &= b_{\mathrm{DG}}(\pi_h u-u,\pi_h u-u_h)+\underbrace{b_{\mathrm{DG}}(u-u_h,\pi_h u-u_h)}_{=0}\\ &\lesssim |||\pi_h u-u|||_\mu|||\pi_h u-u_h|||_\mu\\ &\lesssim h^k\,|u|_{k+1,\mathcal{T}_h}\,|||\pi_h u-u_h|||_\mu. \end{aligned}$$

This gives (4.3.34) for the SIPG and the NIPG method.

To obtain the L^2-error estimate for the SIPG method, we make use of adjoint consistency. **Adjoint consistency** means that the following equality holds

$$b_{\mathrm{DG}}(\varphi_h,\Psi)=\int_{\mathcal{T}_h}\varphi_h g\,\mathrm{d}\mathbf{x},\quad \forall\,\varphi_h\in H^2(\mathcal{T}_h) \tag{4.3.37}$$

for the solution $\Psi\in H^2(\Omega)$ of the adjoint problem

$$-\Delta\Psi=g \text{ in } \Omega,\quad \Psi=0 \text{ on } \partial\Omega, \tag{4.3.38}$$

where $g\in L^2(\Omega)$.

The SIPG method satisfies this equality, whereas for the NIPG method we obtain

$$b_{NIPG}(\varphi_h,\Psi)=\int_{\mathcal{T}_h}\varphi_h g\,\mathrm{d}\mathbf{x}+2\int_{\mathcal{S}_h}\{\nabla\Psi\}\cdot\mathbf{n}_S[\varphi_h]\,\mathrm{d}\mathbf{s}. \tag{4.3.39}$$

Therefore, the L^2-error estimate cannot be improved for the NIPG method. In the case of the SIPG method we can set $g=u-u_h$ and obtain

$$b_{SIPG}(\varphi_h,\Psi)=\int_{\mathcal{T}_h}\varphi_h(u-u_h)\,\mathrm{d}\mathbf{x}\quad\forall\varphi_h\in H^2(\mathcal{T}_h). \tag{4.3.40}$$

Choosing $\varphi_h=(u-u_h)$ yields

$$b_{SIPG}(u-u_h,\Psi)=\int_{\mathcal{T}_h}(u-u_h)^2\,\mathrm{d}\mathbf{x}=\|u-u_h\|_{\mathcal{T}_h}^2. \tag{4.3.41}$$

Since $\pi_h\Psi\in V_{DG,h}^k$ and the bilinear form $b_{SIPG}(\cdot,\cdot)$ is bounded, we obtain

$$\begin{aligned} \|u-u_h\|_{\mathcal{T}_h}^2 &= b_{SIPG}(u-u_h,\Psi)=b_{SIPG}(u-u_h,\Psi-\pi_h\Psi)\\ &\leq C|||u-u_h|||_\mu|||\Psi-\pi_h\Psi|||_\mu\leq Ch\,|\Psi|_{2,\mathcal{T}_h}\,|||u-u_h|||_\mu. \end{aligned}$$

If Ω is convex, the following inequality holds

$$|\Psi|_{2,\mathcal{T}_h}\leq C\|g\|_{L^2(\mathcal{T}_h)}=C\|u-u_h\|_{\mathcal{T}_h}$$

and we derive the optimal estimate

$$\|u-u_h\|_{\mathcal{T}_h}\leq Ch^{k+1}\,|u|_{k+1,\mathcal{T}_h}\,. \tag{4.3.42}$$

□

The proof showed that adjoint consistency is necessary to prove optimal L^2-convergence rates. This is similar to the continuous case, where a duality argument from Aubin and Nitsche leads to optimal rates. The only adjoint consistent method presented here is the SIPG method.

In summary, we have seen, how advection and diffusion terms can be discretized by the DG method. A proper definition of numerical fluxes leads to stable methods for advection and diffusion equations.

5

The CG1-DG2 method for scalar equations in 2D

In this chapter, we introduce a new method which combines the continuous and the discontinuous Galerkin method and is called the CG1-DG2 method. The motivation for this method is the fact that the continuous Galerkin method is unstable for convection-dominated problems (see section 3.4) whereas the discontinuous Galerkin method with upwind numerical fluxes is stable (see section 4.2) but has higher computational costs. The key idea is to use linear continuous finite elements and enrich them with quadratic basis functions which are discontinuous across element edges but vanish at mesh vertices. The resulting space is a subspace of the discontinuous quadratic space (DG2) therefore containing the space of continuous quadratic elements (CG2). On triangular meshes, it can be shown that the method is stable for advection equations (unlike CG2) and converges with the same rate as DG2 but has lower computational costs than DG2 due to fewer degrees of freedom. For Poisson's equation, the analysis of the DG method is directly applicable to the CG1-DG2 method so that we derive the same a priori error estimates.

In the following we introduce the method for scalar equations and follow our work already presented in [12] and [15].

5.1. The CG1-DG2 space

In this section, we define the CG1-DG2 space for triangular and quadrilateral meshes. For the latter we also take into account spaces consisting of so-called serendipity elements [3].

5.1.1. Triangular mesh

At first, we define the space of discontinuous quadratic basis functions. Let $\mathcal{T}_h$ be a shape-regular triangulation of Ω and K an element of $\mathcal{T}_h$, which is a triangle. There are 3 nodes $\mathbf{x}_i$, $i = 1,2,3$, of K ordered in a counter-clockwise sense such that edge e_i joins nodes $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ (modulo 3). Each node corresponds to a linear Lagrange basis function ϕ_K^i. For each side e_i we set $\psi_K^i := \phi_K^i \phi_K^{i+1}$ and define

$$D_K := \operatorname{span}\{\psi_K^i : \ 1 \leq i \leq 3\}.$$

Note that $\psi_K^i(\mathbf{x}_j) = 0, \forall i,\ j = 1,2,3$.

The space of discontinuous quadratic basis functions is defined by

$$D_h := \left\{ v_h \in L^2(\Omega) :\ v_h|_K \in D_K \right\}.$$

Adding linear continuous basis functions to D_h gives the CG1-DG2 space:

$$V_{tri,h}^{1,2} := V_h^1 \oplus D_h. \tag{5.1.1}$$

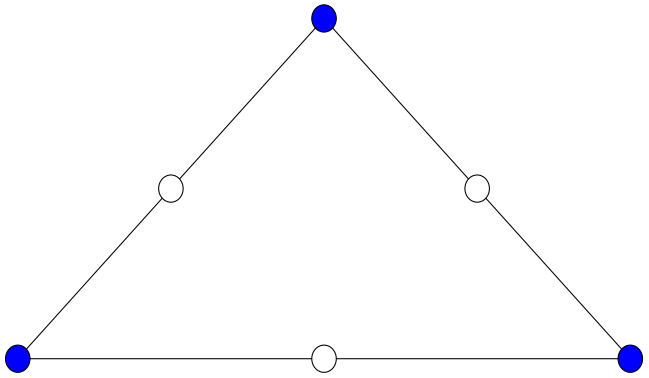


Figure 5.1: CG1-DG2 finite element with discontinuous DOFs (white) and continuous DOFs (blue)

In Fig. 5.1 an element of the CG1-DG2 space with continuous and discontinuous degrees of freedom is displayed. This distribution of continuous and discontinuous degrees of freedom leads to functions $v \in V_{tri,h}^{1,2}$ which are continuous at the vertices of mesh elements but may have jumps across the edges.

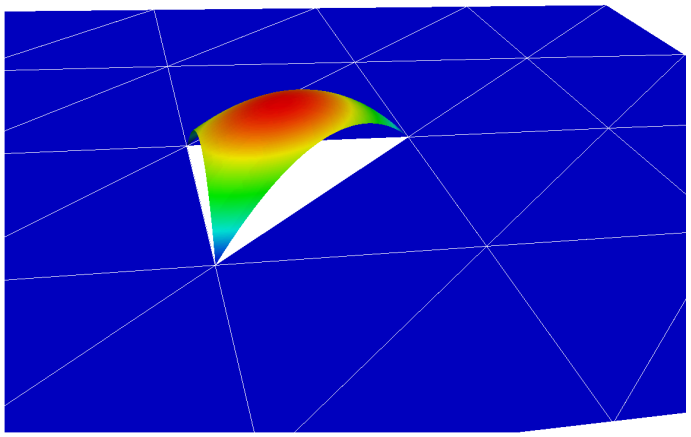
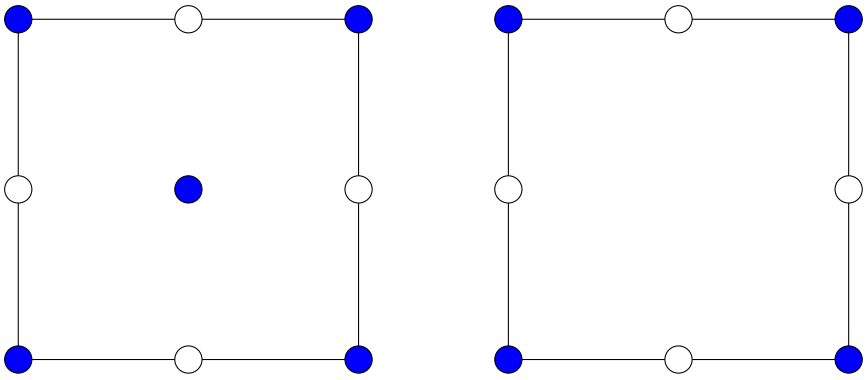


Figure 5.2: CG1-DG2 function

In Fig. 5.2 an example of a function $v \in V_{tri,h}^{1,2}$ is shown. This CG1-DG2 function is constant in all elements but one. Furthermore, it can be seen that the function is continuous at the vertices but discontinuous across the edges.

5.1.2. Quadrilateral mesh

In the case of quadrilateral meshes, we distinguish between Q2 elements (cf. Fig. 5.3a) and (quadratic) serendipity elements [3] (cf. Fig. 5.3b).



(a) Q2 element (b) Serendipity element

Figure 5.3: Quadrilateral elements with continuous DOFs (blue) and discontinuous DOFs (white)

In the following example we will show how basis functions can be defined on these elements.

Example 5.1: Linear and quadratic basis functions for Q2- and serendipity elements

In this example, we consider hierarchical basis functions based on Lobatto shape functions [83] defined by

$$l_0(x) = \frac{1}{2}(1-x), \quad l_1(x) = \frac{1}{2}(x+1), \quad l_2(x) = \sqrt{\frac{3}{8}}(x^2-1).$$

Let us consider the reference quadrilateral $\{\mathbf{x} = (\xi,\eta) | -1 \leq \xi,\eta \leq 1\}$ with nodes

$$\mathbf{x}_1 = (-1,-1), \quad \mathbf{x}_2 = (1,-1), \quad \mathbf{x}_3 = (1,1), \quad \mathbf{x}_4 = (-1,1).$$

At first, we define vertex basis functions

$$\begin{aligned} \phi_1(\mathbf{x}) &= l_0(\xi)l_0(\eta), \quad \phi_2(\mathbf{x}) = l_1(\xi)l_0(\eta), \\ \phi_3(\mathbf{x}) &= l_1(\xi)l_1(\eta), \quad \phi_4(\mathbf{x}) = l_0(\xi)l_1(\eta). \end{aligned} \tag{5.1.2}$$

Each vertex function ϕ_i corresponds to a vertex $\mathbf{x}_i$ in such a way that

$$\phi_i(\mathbf{x}_j) = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{otherwise.} \end{cases}$$

The edge e_i joins vertices $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ (modulo 4). For each edge e_i we define an edge basis function ψ_i by

$$\begin{aligned} \psi_1(\mathbf{x}) &= l_2(\xi)l_0(\eta), \quad \psi_2(\mathbf{x}) = l_1(\xi)l_2(\eta), \\ \psi_3(\mathbf{x}) &= l_2(\xi)l_1(\eta), \quad \psi_4(\mathbf{x}) = l_0(\xi)l_2(\eta), \end{aligned} \tag{5.1.3}$$

where $\psi_i(\mathbf{x}_j) = 0, \forall 1 \leq i,j \leq 4$ and $\psi_i(\mathbf{x}) = 0, \forall \mathbf{x} \in e_j, i \neq j$.

The vertex and edge basis functions define the basis for the serendipity element. This gives the reduced biquadratic space

$$S^2 := \text{span}\{1, x, y, xy, x^2, y^2, x^2y, xy^2\},$$

where $\dim(S^2) = 8$. For the Q2-element an additional bubble basis function is defined by

$$\psi_5(\mathbf{x}) = l_2(\xi)l_2(\eta), \tag{5.1.4}$$

which vanishes on the element boundary. This leads to the biquadratic space

$$Q^2 := \text{span}\{1, x, y, xy, x^2, y^2, x^2y, xy^2, x^2y^2\},$$

where $\dim(Q^2) = 9$. These spaces satisfy

$$P^2 \subset S^2 \subset Q^2,$$

where $P^2 = \text{span}\{1, x, y, xy, x^2, y^2\}$ is the quadratic space corresponding to triangles. Furthermore, we have $S^2 \subset P^3$ [3].

For the definition of CG1-DG2 spaces we introduce the spaces of the quadratic basis functions corresponding to S^2 and Q^2. Let $\mathcal{T}_h$ be a regular triangulation of Ω and K a rectangular element of $\mathcal{T}_h$. There are four nodes $\mathbf{x}_i$, $i = 1,...4$, of K ordered in a counter-clockwise sense such that edge e_i joins nodes $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ (modulo 4). Each node corresponds to a vertex basis function ϕ_K^i, see (5.1.2). For each side e_i we have the edge function ψ_K^i, see (5.1.3). In addition, we have a bubble function ψ_K^5, see (5.1.4). We define

$$D_K^l := \text{span}\{\psi_K^i : \ 1 \le i \le 3 + l\}, \ l = 1, 2.$$

Now, we have two spaces of discontinuous quadratic basis functions defined by

$$D_h^l := \left\{ v_h \in L^2(\Omega) : \ v_h|_K \in D_K^l \right\}, \ l = 1, 2,$$

corresponding to the serendipity element for $l = 1$ and to the Q2 element for $l = 2$. Adding bilinear continuous basis functions to D_h^l yields the two CG1-DG2 spaces:

$$V_{ser,h}^{1,2} := V_h^{1,Q} \oplus D_h^1 \quad \text{and} \quad V_{Q2,h}^{1,2} := V_h^{1,Q} \oplus D_h^2. \tag{5.1.5}$$

We will call $V_{ser,h}^{1,2}$ the serendipity CG1-DG2 space and $V_{Q2,h}^{1,2}$ the Q2-CG1-DG2 space.

Remark 5.2: Implementation

The different CG1-DG2 spaces have been implemented in the open-source C++ library HERMES [82] which provides continuous and discontinuous Galerkin methods as well as hierarchical basis functions. For the implementation of the CG1-DG2 space, we started with the continuous quadratic finite element space and changed all quadratic edge functions to be discontinuous. Since HERMES distinguishes between vertex, edge and bubble functions where bubble functions are functions which are only defined element-wise, switching from a continuous to a discontinuous representation can be done just by changing the function type from edge to bubble functions.

In the following, we will show how advection and diffusion equations can be discretized using the CG1-DG2 method. To simplify notation we will skip the subscripts tri, $Q2$ and ser and use $V_h^{1,2}$ if it is clear which mesh type we consider. To simplify the distinction between the triangular and the quadrilateral case, we call the method on triangular meshes the triangular CG1-DG2 method.

5.2. The CG1-DG2 method for the advection equation

As in Chapter 4 we will start with the discretization of the linear advection equation

$$\nabla \cdot (\boldsymbol{\beta} u) + cu = f \quad \text{in } \Omega, \tag{5.2.1}$$

$$u = g \quad \text{on } \Gamma_-, \tag{5.2.2}$$

where $c \in L^\infty(\Omega)$ and $\beta_i \in W^{1,\infty}(\Omega)$ for each component $\beta_i, i = 1,2$ of $\boldsymbol{\beta}$.

We require

$$c(\mathbf{x}) + \frac{1}{2}\nabla \cdot \boldsymbol{\beta}(\mathbf{x}) \geq c_0 > 0, \quad \forall x \in \Omega \tag{5.2.3}$$

to ensure L^2-coercivity, and

$$\sup_{\mathbf{x}\in\Omega} |c(\mathbf{x}) + \nabla \cdot \boldsymbol{\beta}(\mathbf{x})| =: c_1 < \infty \tag{5.2.4}$$

to obtain boundedness of the bilinear form. Following [5], we assume that

$$\boldsymbol{\beta} \text{ has no closed curves and } |\boldsymbol{\beta}(\mathbf{x})| \neq 0 \; \forall \mathbf{x} \in \Omega. \tag{5.2.5}$$

Discretization in the sense of discontinuous Galerkin with upwinding gives:

$$\int_{\mathcal{T}_h} -(\boldsymbol{\beta} u_h) \cdot \nabla \varphi_h + c u_h \varphi_h \, \mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h^{\mathrm{int}}} \boldsymbol{\beta}_{n_S} \hat{u}_h [\varphi_h] \, \mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial,+}} \boldsymbol{\beta}_{n_S} u_h \varphi_h \, \mathrm{d}\mathbf{s} = \int_{\mathcal{T}_h} f \varphi_h \, \mathrm{d}\mathbf{x} - \int_{\mathcal{S}_h^{\partial,-}} \boldsymbol{\beta}_{n_S} g \varphi_h \, \mathrm{d}\mathbf{s}. \tag{5.2.6}$$

This leads to the bilinear and linear forms of the CG1-DG2 discretization,

$$a_{CD}(u_h, \varphi_h) = \int_{\mathcal{T}_h} -(\boldsymbol{\beta} u_h) \cdot \nabla \varphi_h + c u_h \varphi_h \, \mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h^{\mathrm{int}}} \boldsymbol{\beta}_{n_S} \hat{u} [\varphi_h] \, \mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial,+}} \boldsymbol{\beta}_{n_S} u_h \varphi_h \, \mathrm{d}\mathbf{s}, \tag{5.2.7}$$

$$f(\varphi_h) = \int_{\mathcal{T}_h} f \varphi_h \, \mathrm{d}\mathbf{x} - \int_{\mathcal{S}_h^{\partial,-}} \boldsymbol{\beta}_{n_S} g \varphi_h \, \mathrm{d}\mathbf{s}. \tag{5.2.8}$$

If we integrate (5.2.7) by parts, we obtain

$$a_{CD}(u_h, \varphi_h) = \int_{\mathcal{T}_h} ((\nabla \cdot \boldsymbol{\beta} + c) u_h + \boldsymbol{\beta} \cdot \nabla u_h) \, \varphi_h \, \mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h^{\mathrm{int}}} \boldsymbol{\beta}_{n_S} \left(\hat{u}[\varphi_h] - [u_h \varphi_h] \right) \mathrm{d}\mathbf{s} - \int_{\mathcal{S}_h^{\partial,-}} \boldsymbol{\beta}_{n_S} u_h \varphi_h \, \mathrm{d}\mathbf{s}.$$

The integral over the inner sides can be simplified if we define the downwind value

$$\widetilde{u}(\mathbf{x}) = \begin{cases} u^+(\mathbf{x}) & \text{if } \boldsymbol{\beta}(\mathbf{x}) \cdot \mathbf{n}_S < 0, \\ u^-(\mathbf{x}) & \text{otherwise.} \end{cases} \tag{5.2.9}$$

This yields

$$a_{CD}(u_h,\varphi_h) = \int_{\mathcal{T}_h} ((\nabla\cdot\boldsymbol{\beta}+c)u_h+\boldsymbol{\beta}\cdot\nabla u_h)\,\varphi_h\,\mathrm{d}\mathbf{x} - \int_{\mathcal{S}_h^{\mathrm{int}}} \boldsymbol{\beta}_{n_S}[u_h]\widetilde{\varphi}_h\,\mathrm{d}\mathbf{s} - \int_{\mathcal{S}_h^{\partial,-}} \boldsymbol{\beta}_{n_S}u_h\varphi_h\,\mathrm{d}\mathbf{s}. \tag{5.2.10}$$

Since the bilinear form $a_{CD} = a_{\mathrm{DG}}$ and $V_h^{1,2} \subset V_{DG,h}^2 \subset H^{1,\boldsymbol{\beta}}(\mathcal{T}_h)$, we obtain the same coercivity result as for the DG method (see Lemma 4.10):

Lemma 5.3: Coercivity

The bilinear form (5.2.7) is coercive with respect to the DG-norm:

$$a_{CD}(u_h,u_h) \geq \|u_h\|_{\mathrm{DG}}^2, \quad \forall u_h \in V_h^{1,2} \tag{5.2.11}$$

where $\|.\|_{\mathrm{DG}}$ *is defined by (4.2.13).*

This result is similar to the continuous case, where we only obtain coercivity with respect to the L^2-norm but do not control $\boldsymbol{\beta}\cdot\nabla u$.
In the following we will restrict ourselves to triangular meshes and show as in the DG case (cf. Lemma 4.11) how to improve this result to derive stability in a stronger norm.

Stability on triangular meshes

In the following, we will use the projection operators:

1. π_K^1: the $L^2(K)$ projection on $P^1(K)$ defined by

$$\int_K (v-\pi_K^1 v)w = 0 \quad \forall w \in P^1(K),$$

   where $v \in L^2(K)$.

2. π_K^D: the $L^2(K)$ projection on D_K defined by

$$\int_K (v-\pi_K^D v)w = 0 \quad \forall w \in D_K,$$

   where $v \in L^2(K)$.

3. π_h^D: L^2 projection on D_h defined by

$$\pi_h^D = \sum_K \pi_K^D.$$

At first, we need to define a norm which controls the term $\boldsymbol{\beta}\cdot\nabla u$. Therefore, we introduce the following seminorm which gives control over the weighted streamline derivatives

$$|||u_h|||_{\boldsymbol{\beta}} := \sqrt{\sum_{K\in\mathcal{T}_h} \delta_K \|\pi_K^1\boldsymbol{\beta}\cdot\nabla u_h\|_K^2}, \quad \delta_K = \frac{h_K}{\|\boldsymbol{\beta}\|_{\infty,K}}. \tag{5.2.12}$$

This seminorm together with the DG-norm defines the augmented DG-norm

$$|||u_h||| := \sqrt{\|u_h\|_{\mathrm{DG}}^2 + |||u_h|||_{\boldsymbol{\beta}}^2}, \tag{5.2.13}$$

which controls also the streamline derivative part. Note that this norm is similar to the $|||.|||_{\mathrm{DG}}$-norm defined by (4.2.28).

To prove stability with respect to the $|||.|||$-norm, we need the following lemma, which is only applicable to triangular elements.

Lemma 5.4

For any $p \in P^1(K)$ there exists a mesh-independent constant $1 \geq c_p > 0$ such that

$$\sup_{q \in D_K \setminus \{0\}} \frac{\int_K pq \,\mathrm{d}\mathbf{x}}{\|q\|_{L^2(K)}} \geq c_p \|p\|_{L^2(K)}. \tag{5.2.14}$$

Proof. Without loss of generality we assume $\|p\|_{L^2(K)} > 0$.
On a reference triangle $\hat{K}$ we consider linear Lagrange basis functions $\hat{\phi}_i$ and quadratic basis functions $\hat{\psi}_i := \hat{\phi}_i \hat{\phi}_{i+1}$ with $i = 1,2,3$. By straightforward computation we find that the matrix M with $M_{ij} := \int_{\hat{K}} \hat{\phi}_i \hat{\psi}_j \,\mathrm{d}\mathbf{x}$ has full rank.
This means that for $p \in P^1(\hat{K})$ there exists $q \in D_{\hat{K}}$ such that

$$\int_{\hat{K}} pq \,\mathrm{d}\mathbf{x} \geq c_p \|p\|_{L^2(\hat{K})}^2, \quad \|q\|_{L^2(\hat{K})} \leq \|p\|_{L^2(\hat{K})}.$$

This becomes clear, if we choose $p = \sum_{i=1}^{3} b_i \hat{\phi}_i$ and $q = \sum_{i=1}^{3} d_i \hat{\psi}_i$ and derive

$$\int_{\hat{K}} pq \,\mathrm{d}\mathbf{x} = b^T M d \geq c_p b^T M_p b,$$

where $d = c_p M^{-1} M_p b$ and $(M_p)_{ij} := \int_{\hat{K}} \hat{\phi}_i \hat{\phi}_j \,\mathrm{d}\mathbf{x}$.
Since $\|q\|_{L^2(\hat{K})} \leq \|p\|_{L^2(\hat{K})}$, we can choose

$$c_p \leq \frac{\|p\|_{L^2(\hat{K})}}{\|\sum_{i=1}^{3} (M^{-1} M_p b)_i \psi_i\|_{L^2(\hat{K})}}.$$

Note that we have $c_p > 0$ due to the assumption $\|p\|_{L^2(K)} > 0$.
Transformation to the physical element K completes the proof. □

An extension of Lemma 5.4 to quadrilateral elements would require $p \in Q^1(K)$. However, the proof cannot be directly transferred to quadrilateral elements since the matrix M is not regular in the quadrilateral case.

We also will make use of the following inverse inequalities [66, 75] :

Lemma 5.5: Inverse estimates

Let $K \in \mathcal{T}_h$ and $S \subset S_K$. Then, for all $v_h \in V_h^{1,2}$ there exists a mesh-independent constant $C_I > 0$ such that

$$\|v_h\|_{L^2(S)} \leq C_I h_K^{-\frac{1}{2}} \|v_h\|_{L^2(K)}, \tag{5.2.15}$$

$$\|\nabla v_h\|_{L^2(K)} \leq C_I h_K^{-1} \|v_h\|_{L^2(K)}. \tag{5.2.16}$$

Furthermore, we need the following estimate:

Lemma 5.6

There exists a mesh-independent constant $\tilde{c}_p > 0$ such that

$$\sum_{K\in\mathcal{T}_h} \delta_K \|\pi_K^1 \boldsymbol{\beta} \cdot \nabla v_h\|_K^2 \leq \tilde{c}_p^{-1} \sum_{K\in\mathcal{T}_h} \delta_K \|\pi_K^D \pi_K^1 \boldsymbol{\beta} \cdot \nabla v_h\|_K^2, \quad \forall\, v_h \in V_h^{1,2}. \tag{5.2.17}$$

Proof. We set $p = \pi_K^1 \boldsymbol{\beta} \cdot \nabla v_h \in P^1(K)$. Applying Lemma 5.4 and the definition of π_K^D gives

$$\|p\|_K \leq c_p^{-1} \sup_{q\in D_K\setminus\{0\}} \frac{\int_K pq\,\mathrm{d}\mathbf{x}}{\|q\|_K} = c_p^{-1} \sup_{q\in D_K\setminus\{0\}} \frac{\int_K (\pi_K^D p)q\,\mathrm{d}\mathbf{x}}{\|q\|_K} \leq c_p^{-1} \|\pi_K^D p\|_K.$$

Using the squared norm, multiplying by δ_K and summing up completes the proof. □

The stability of the method can now be derived by the following inf-sup condition:

Lemma 5.7

There exists a mesh-independent constant $\gamma > 0$ such that

$$\sup_{v_h\in V_h^{1,2}\setminus\{0\}} \frac{a_{CD}(u_h, v_h)}{|||v_h|||} \geq \gamma |||u_h|||, \quad \forall\, u_h \in V_h^{1,2}. \tag{5.2.18}$$

Proof. We will now show that for a given $u_h \in V_h^{1,2}$ there exists $v_h \in V_h^{1,2}$ such that

$$a_{CD}(u_h, v_h) \geq \gamma' |||u_h|||^2 \quad \text{and} \quad |||v_h||| \leq C|||u_h|||. \tag{5.2.19}$$

Choosing $\gamma = \gamma'/C$ yields (5.2.18).

Let $w_h := \sum_K \delta_K \pi_K^D \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h \in D_h$. We choose $\varphi_h = w_h$ in (5.2.10) and obtain

$$a_{CD}(u_h, w_h) = \int_{\mathcal{T}_h} (c + \nabla\cdot\boldsymbol{\beta}) u_h w_h\,\mathrm{d}\mathbf{x} + \int_{\mathcal{T}_h} \boldsymbol{\beta}\cdot\nabla u_h w_h\,\mathrm{d}\mathbf{x} - \int_{\mathcal{S}_h^{\mathrm{int}}} \boldsymbol{\beta}_{n_S} [u_h] \widetilde{w}_h\,\mathrm{d}\mathbf{s}$$
$$- \int_{\mathcal{S}_h^{\partial,-}} \boldsymbol{\beta}_{n_S} u_h w_h.$$

Adding $0 = \int_{\mathcal{T}_h} \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h w_h \,\mathrm{d}\mathbf{x} - \int_{\mathcal{T}_h} \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h w_h \,\mathrm{d}\mathbf{x}$ leads to

$$\begin{aligned} a_{CD}(u_h, w_h) = &\int_{\mathcal{T}_h} (c + \nabla \cdot \boldsymbol{\beta}) u_h w_h \,\mathrm{d}\mathbf{x} + \int_{\mathcal{T}_h} (\boldsymbol{\beta} \cdot \nabla u_h - \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h) w_h \,\mathrm{d}\mathbf{x} \\ &- \int_{\mathcal{S}_h^{\mathrm{int}}} \boldsymbol{\beta}_{n_S} [u_h] \widetilde{w}_h \,\mathrm{d}\mathbf{s} - \int_{\mathcal{S}_h^{\partial,-}} \boldsymbol{\beta}_{n_S} u_h w_h \,\mathrm{d}\mathbf{s} + \int_{\mathcal{T}_h} \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h w_h \,\mathrm{d}\mathbf{x}. \end{aligned} \tag{5.2.20}$$

If we now use the definition of π_h^D, we have

$$\sum_{K \in \mathcal{T}_h} \delta_K \|\pi_K^D \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h\|_K^2 = \int_{\mathcal{T}_h} \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h w_h \,\mathrm{d}\mathbf{x}. \tag{5.2.21}$$

Combining equations (5.2.20) and (5.2.21) gives

$$\begin{aligned} \sum_{K \in \mathcal{T}_h} \delta_K \|\pi_K^D \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h\|_K^2 = &a_{CD}(u_h, w_h) - \int_{\mathcal{T}_h} (c + \nabla \cdot \boldsymbol{\beta}) u_h w_h \,\mathrm{d}\mathbf{x} - \int_{\mathcal{T}_h} (\boldsymbol{\beta} \cdot \nabla u_h - \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h) w_h \,\mathrm{d}\mathbf{x} \\ &+ \int_{\mathcal{S}_h^{\mathrm{int}}} \boldsymbol{\beta}_{n_S} [u_h] \widetilde{w}_h \,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial,-}} \boldsymbol{\beta}_{n_S} u_h w_h \,\mathrm{d}\mathbf{s}. \end{aligned}$$

Now we use (5.2.4), Hölder's inequality and Theorem A.4 and derive

$$\begin{aligned} &\sum_{K \in \mathcal{T}_h} \delta_K \|\pi_K^D \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h\|_K^2 \\ &\quad \le a_{CD}(u_h, w_h) + \|\boldsymbol{\beta} \cdot \nabla u_h - \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h\| \|w_h\| + c_1 \|u_h\| \|w_h\| \\ &\qquad + \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} [u_h] \|_{\mathcal{S}_h^{\mathrm{int}}} \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} \widetilde{w}_h \|_{\mathcal{S}_h^{\mathrm{int}}} + \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} u_h \|_{\mathcal{S}_h^{\partial}} \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} w_h \|_{\mathcal{S}_h^{\partial}} \\ &\quad \le a_{CD}(u_h, w_h) + \|\boldsymbol{\beta} \cdot \nabla u_h - \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h\| \|w_h\| \\ &\qquad + \left(c_0 \|u_h\|^2 + \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} [u_h] \|_{\mathcal{S}_h^{\mathrm{int}}}^2 + \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} u_h \|_{\mathcal{S}_h^{\partial}}^2 \right)^{\frac{1}{2}} \left(\frac{c_1^2}{c_0} \|w_h\|^2 + \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} \widetilde{w}_h \|_{\mathcal{S}_h^{\mathrm{int}}}^2 + \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} w_h \|_{\mathcal{S}_h^{\partial}}^2 \right)^{\frac{1}{2}} \\ &\quad \le a_{CD}(u_h, w_h) + \|\boldsymbol{\beta} \cdot \nabla u_h - \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h\| \|w_h\| \\ &\qquad + \sqrt{2} \|u_h\|_{\mathrm{DG}} \left(\frac{c_1^2}{c_0} \|w_h\|^2 + \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} \widetilde{w}_h \|_{\mathcal{S}_h^{\mathrm{int}}}^2 + \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} w_h \|_{\mathcal{S}_h^{\partial}}^2 \right)^{\frac{1}{2}} \\ &\quad \le a_{CD}(u_h, w_h) + C \left(\|u_h\|_{\mathrm{DG}}^2 + \|\boldsymbol{\beta} \cdot \nabla u_h - \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h\|^2 \right)^{\frac{1}{2}} \\ &\qquad \cdot \left(\frac{c_1^2 + c_0}{c_0} \|w_h\|^2 + \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} \widetilde{w}_h \|_{\mathcal{S}_h^{\mathrm{int}}}^2 + \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} w_h \|_{\mathcal{S}_h^{\partial}}^2 \right)^{\frac{1}{2}}. \end{aligned} \tag{5.2.22}$$

The boundary and edge integrals can be estimated using the inverse estimate (5.2.15) as follows:

$$\begin{aligned} \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} \widetilde{w}_h \|_{\mathcal{S}_h^{\mathrm{int}}}^2 + \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} w_h \|_{\mathcal{S}_h^{\partial}}^2 &\le C_I^2 \sum_{K \in \mathcal{T}_h} h_K^{-1} \|\boldsymbol{\beta}\|_{\infty,K} \|\delta_K \pi_K^D \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h\|_K^2 \\ &= C_I^2 \sum_{K \in \mathcal{T}_h} \delta_K \|\pi_K^D \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h\|_K^2. \end{aligned} \tag{5.2.23}$$

Furthermore, due to the stability of the projection $\|\pi_K u\| \leq \|u\|$ and the inverse estimate (5.2.16) we obtain

$$\begin{aligned}\|w_h\|^2 &= \sum_{K\in\mathcal{T}_h} \delta_K^2 \|\pi_K^D \pi_K^1 \boldsymbol{\beta}\cdot\nabla u_h\|_K^2 \leq \sum_{K\in\mathcal{T}_h} \delta_K^2 \|\boldsymbol{\beta}\cdot\nabla u_h\|_K^2 \\ &\leq \sum_{K\in\mathcal{T}_h} \delta_K^2 \|\boldsymbol{\beta}\|_{\infty,K}^2 \|\nabla u_h\|_K^2 \leq C_I^2 \sum_{K\in\mathcal{T}_h} \delta_K^2 \|\boldsymbol{\beta}\|_{\infty,K}^2 h_K^{-2} \|u_h\|_K^2 \\ &= C_I^2 \|u_h\|^2. \end{aligned} \tag{5.2.24}$$

Combining inequalities (5.2.22)-(5.2.24) and applying Young's inequality yield

$$\begin{aligned}\sum_{K\in\mathcal{T}_h} \delta_K \|\pi_K^D \pi_K^1 \boldsymbol{\beta}\cdot\nabla u_h\|_K^2 &\leq a_{CD}(u_h, w_h) + \tilde{C}\left(\|u_h\|_{\mathrm{DG}}^2 + \|\boldsymbol{\beta}\cdot\nabla u_h - \pi_K^1 \boldsymbol{\beta}\cdot\nabla u_h\|^2\right)^{\frac{1}{2}} \\ &\quad \cdot \left(\|u_h\|^2 + \sum_{K\in\mathcal{T}_h} \delta_K \|\pi_K^D \pi_K^1 \boldsymbol{\beta}\cdot\nabla u_h\|_K^2\right)^{\frac{1}{2}} \\ &\leq a_{CD}(u_h, w_h) + \frac{\tilde{C}^2}{2}\left(\|u_h\|_{\mathrm{DG}}^2 + \|\boldsymbol{\beta}\cdot\nabla u_h - \pi_K^1 \boldsymbol{\beta}\cdot\nabla u_h\|^2\right) \\ &\quad + \frac{1}{2}\|u_h\|^2 + \frac{1}{2}\sum_{K\in\mathcal{T}_h} \delta_K \|\pi_K^D \pi_K^1 \boldsymbol{\beta}\cdot\nabla u_h\|_K^2.\end{aligned}$$

This leads to

$$\frac{1}{2}\sum_{K\in\mathcal{T}_h} \delta_K \|\pi_K^D \pi_K^1 \boldsymbol{\beta}\cdot\nabla u_h\|_K^2 \leq a_{CD}(u_h, w_h) + \frac{\tilde{C}^2}{2}\left(\|u_h\|_{\mathrm{DG}}^2 + \|\boldsymbol{\beta}\cdot\nabla u_h - \pi_K^1 \boldsymbol{\beta}\cdot\nabla u_h\|^2\right) + \frac{1}{2}\|u_h\|^2.$$

Next we estimate $\|\boldsymbol{\beta}\cdot\nabla u_h - \pi_K^1 \boldsymbol{\beta}\cdot\nabla u_h\|$. On each element K we have

$$\boldsymbol{\beta}\cdot\nabla u_h - \pi_K^1(\boldsymbol{\beta}\cdot\nabla u_h) = (\boldsymbol{\beta} - \boldsymbol{\beta}_K)\cdot\nabla u_h + \boldsymbol{\beta}_K\cdot\nabla u_h - \pi_K^1(\boldsymbol{\beta}\cdot\nabla u_h), \tag{5.2.25}$$

where $\boldsymbol{\beta}_K := \boldsymbol{\beta}(\mathbf{x}_K)$, $\mathbf{x}_K$ denotes the centroid of K. Since $\boldsymbol{\beta}_K$ is constant on K and $(\nabla u_h)|_K \in P^1(K)$, $\boldsymbol{\beta}_K\cdot\nabla u_h = \pi_K^1(\boldsymbol{\beta}_K\cdot\nabla u_h)$ holds.
Using the stability of the projection, the inequality $\|\boldsymbol{\beta} - \boldsymbol{\beta}_K\|_K \lesssim h_K|\boldsymbol{\beta}|_{1,K}$ [17] and the inverse estimate (5.2.15) yield

$$\begin{aligned}\|\boldsymbol{\beta}\cdot\nabla u_h - \pi_K^1(\boldsymbol{\beta}\cdot\nabla u_h)\|_K &= \|(I - \pi_K^1)((\boldsymbol{\beta} - \boldsymbol{\beta}_K)\cdot\nabla u_h)\|_K \\ &\leq \|\boldsymbol{\beta} - \boldsymbol{\beta}_K\|_K \|\nabla u_h\|_K \leq C_I|\boldsymbol{\beta}|_{1,K}\|u_h\|_K.\end{aligned} \tag{5.2.26}$$

It follows that

$$\frac{1}{2}\sum_{K\in\mathcal{T}_h} \delta_K \|\pi_K^D \pi_K^1 \boldsymbol{\beta}\cdot\nabla u_h\|_K^2 \leq a_{CD}(u_h, w_h) + C'\|u_h\|_{\mathrm{DG}}^2.$$

Applying Lemma 5.6 we obtain

$$\frac{\tilde{c}_p}{2}\sum_{K\in\mathcal{T}_h} \delta_K \|\pi_K^1 \boldsymbol{\beta}\cdot\nabla u_h\|_K^2 \leq a_{CD}(u_h, w_h) + C'\|u_h\|_{\mathrm{DG}}^2.$$

Now we choose $v_h := u_h + \varepsilon w_h$ with $\varepsilon > 0$ sufficiently small.

Taking into account coercivity with respect to the DG-norm gives

$$\begin{aligned} a_{CD}(u_h, v_h) &= a_{CD}(u_h, u_h) + \varepsilon a_{CD}(u_h, w_h) \\ &\geq \|u_h\|_{\mathrm{DG}}^2 + \varepsilon \left(\frac{\tilde{c}_p}{2} \sum_{K \in \mathcal{T}_h} \delta_K \|\pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h\|_K^2 - C' \|u_h\|_{\mathrm{DG}}^2 \right) \\ &= (1 - \varepsilon C') \|u_h\|_{\mathrm{DG}}^2 + \varepsilon \frac{\tilde{c}_p}{2} \sum_{K \in \mathcal{T}_h} \delta_K \|\pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h\|_K^2 \\ &\geq \gamma' |||u_h|||^2, \end{aligned}$$

where $\gamma' = \tilde{c}_p/(\tilde{c}_p + 2C')$ for $\varepsilon = 2/(\tilde{c}_p + 2C')$.
The last step is to check if

$$|||w_h||| = \sqrt{\|w_h\|_{\mathrm{DG}}^2 + |||w_h|||_{\boldsymbol{\beta}}^2} \leq C |||u_h|||$$

is fulfilled.
By applying (5.2.23) and (5.2.24) and taking into account the stability of the projection, it is easy to verify that $\|w_h\|_{\mathrm{DG}} \leq C|||u_h|||$. For the streamline part we obtain

$$\begin{aligned} |||w_h|||_{\boldsymbol{\beta}}^2 = \sum_{K \in \mathcal{T}_h} \delta_K \|\pi_K^1 \boldsymbol{\beta} \cdot \nabla w_h\|_K^2 &\leq \sum_{K \in \mathcal{T}_h} \delta_K \|\boldsymbol{\beta} \cdot \nabla w_h\|_K^2 \leq \sum_{K \in \mathcal{T}_h} \delta_K \|\boldsymbol{\beta}\|_{\infty,K}^2 \|\nabla w_h\|_K^2 \\ &\leq C_I^2 \sum_{K \in \mathcal{T}_h} \delta_K \|\boldsymbol{\beta}\|_{\infty,K}^2 h_K^{-2} \|w_h\|_K^2 = C_I^2 \sum_{K \in \mathcal{T}_h} \delta_K^{-1} \|w_h\|_K^2 \\ &= C_I^2 \sum_{K \in \mathcal{T}_h} \delta_K \|\pi_K^D \pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h\|_K^2 \\ &\leq C_I^2 \sum_{K \in \mathcal{T}_h} \delta_K \|\pi_K^1 \boldsymbol{\beta} \cdot \nabla u_h\|_K^2 = C_I^2 |||u_h|||_{\boldsymbol{\beta}}^2, \end{aligned}$$

which leads to (5.2.19). □

The inf-sup condition leads directly to the stability of the method. The next step is to derive a priori error estimates.

A priori error analysis for triangular meshes

In the following we will make use of the Clément operator [21] based on local L^2 projections on element patches. The exact definition of this operator can be found in the Appendix (Def. A.10). In our case it maps onto the space V_h^2. In Fig. 5.4 element patches with respect to an element K and an edge S are displayed. Furthermore, we will need the following interpolation error estimates (cf. Appendix, Lemma A.12):

Lemma 5.8

Let $0 \leq l \leq 1$ and $0 \leq k \leq 2$. Then the Clément operator $\mathcal{C}_h$ has the following local interpolation properties

$$\|\nabla^l (u - \mathcal{C}_h u)\|_K \leq C h_K^{k+1-l} \|u\|_{k+1,\omega_K}, \quad \forall u \in H^{k+1}(\omega_K), \tag{5.2.27}$$

$$\|u - \mathcal{C}_h u\|_S \leq C h_K^{k+\frac{1}{2}} \|u\|_{k+1,\omega_S}, \quad \forall u \in H^{k+1}(\omega_S), \tag{5.2.28}$$

where ω_K and ω_S denote the patches of cells around K and S, respectively.

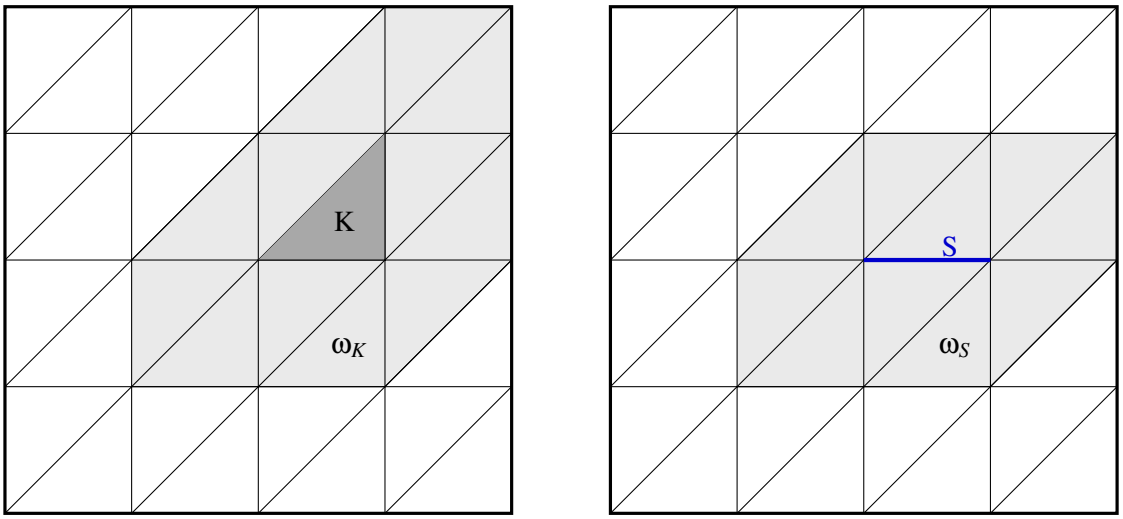


Figure 5.4: Element patches

For the proof of the a priori error estimate, we will need the following estimate of the consistency error:

Lemma 5.9

Let $0 \leq k \leq 2$, $u \in H^{k+1}(\Omega)$ *and* $h := \max_{K \in \mathcal{T}_h} h_K$. *Then the consistency error is bounded by*

$$\sup_{v_h \in V_h^{1,2} \setminus \{0\}} \frac{a_{CD}(\mathcal{C}_h u - u, v_h)}{|||v_h|||} \leq C h^{k+\frac{1}{2}} \|u\|_{k+1}. \tag{5.2.29}$$

Proof. The bilinear form is given as

$$\begin{aligned} a_{CD}(\mathcal{C}_h u - u, v_h) = &\int_{\mathcal{T}_h} -(\mathcal{C}_h u - u)\boldsymbol{\beta} \cdot \nabla v_h + c(\mathcal{C}_h u - u) v_h \, \mathrm{d}\mathbf{x} \\ &+ \int_{\mathcal{S}_h^{\mathrm{int}}} \boldsymbol{\beta}_{n_S} (\widehat{\mathcal{C}_h u - u})[v_h] \, \mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial,+}} \boldsymbol{\beta}_{n_S} (\mathcal{C}_h u - u) v_h \, \mathrm{d}\mathbf{s}. \end{aligned}$$

Now we will bound the different terms. At first we obtain for the boundary and side integrals

$$\begin{aligned} &\int_{\mathcal{S}_h^{\mathrm{int}}} \boldsymbol{\beta}_{n_S} (\widehat{\mathcal{C}_h u - u})[v_h] \, \mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial,+}} \boldsymbol{\beta}_{n_S} (\mathcal{C}_h u - u) v_h \, \mathrm{d}\mathbf{s} \\ &\quad \leq \int_{\mathcal{S}_h^{\mathrm{int}}} |\boldsymbol{\beta}_{n_S}| \, |(\mathcal{C}_h u - u)| \, |[v_h]| \, \mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial,+}} |\boldsymbol{\beta}_{n_S}| \, |(\mathcal{C}_h u - u)| \, |v_h| \, \mathrm{d}\mathbf{s} \\ &\quad \leq \sum_{S \in \mathcal{S}_h^{\mathrm{int}}} \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} (\mathcal{C}_h u - u) \|_S \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} [v_h] \|_S + \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} (\mathcal{C}_h u - u) \|_{\partial\Omega} \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} v_h \|_{\partial\Omega} \\ &\quad \leq C \|\boldsymbol{\beta}\|_\infty^{\frac{1}{2}} h^{k+\frac{1}{2}} \|u\|_{k+1} \left(\left(\int_{\mathcal{S}_h^{\mathrm{int}}} |\boldsymbol{\beta}_n| [v_h]^2 \right)^{\frac{1}{2}} + \| |\boldsymbol{\beta}_n|^{\frac{1}{2}} v_h \|_{\partial\Omega} \right). \end{aligned}$$

Next we have

$$\int_{\mathcal{T}_h} c(\mathcal{C}_h u - u) v_h \,\mathrm{d}\mathbf{x} \leq c_{max} \|(\mathcal{C}_h u - u)\| \|v_h\| \leq C h^{k+1} \|u\|_{k+1} \|v_h\|,$$

where $c_{max} := \max_{\mathbf{x}\in\Omega} c(\mathbf{x})$.

The last term to bound can be written as

$$\int_{\mathcal{T}_h} (u - \mathcal{C}_h u)\boldsymbol{\beta}\cdot\nabla v_h \,\mathrm{d}\mathbf{x} = \int_{\mathcal{T}_h} (u - \mathcal{C}_h u)\pi_K^1\boldsymbol{\beta}\cdot\nabla v_h \,\mathrm{d}\mathbf{x} + \int_{\mathcal{T}_h} (u - \mathcal{C}_h u)\left(\boldsymbol{\beta}\cdot\nabla v_h - \pi_K^1\boldsymbol{\beta}\cdot\nabla v_h\right) \mathrm{d}\mathbf{x}.$$

The first part can be easily estimated by

$$\begin{aligned}\int_{\mathcal{T}_h} (u - \mathcal{C}_h u)\pi_K^1\boldsymbol{\beta}\cdot\nabla v_h \,\mathrm{d}\mathbf{x} &\leq \left(\sum_{K\in\mathcal{T}_h} \delta_K^{-1} \|(u - \mathcal{C}_h u)\|_K^2\right)^{\frac{1}{2}} \left(\sum_{K\in\mathcal{T}_h} \delta_K \|\pi_K^1\boldsymbol{\beta}\cdot\nabla v_h\|_K^2\right)^{\frac{1}{2}} \\ &\leq C\|\boldsymbol{\beta}\|_\infty^{\frac{1}{2}} h^{k+\frac{1}{2}} \|u\|_{k+1} |||v_h|||_{\boldsymbol{\beta}}.\end{aligned}$$

For the second part we use (5.2.26) and obtain

$$\begin{aligned}\int_{\mathcal{T}_h} (u - \mathcal{C}_h u)\left(\boldsymbol{\beta}\cdot\nabla v_h - \pi_K^1\boldsymbol{\beta}\cdot\nabla v_h\right) \mathrm{d}\mathbf{x} &\leq \|(u - \mathcal{C}_h u)\| \|\boldsymbol{\beta}\cdot\nabla v_h - \pi_K^1\boldsymbol{\beta}\cdot\nabla v_h\| \\ &\leq C h^{k+1} \|u\|_{k+1} |\boldsymbol{\beta}|_{1,K} \|v_h\|_K.\end{aligned}$$

In summary, we have

$$a_{CD}(\mathcal{C}_h u - u, v_h) \leq C h^{k+\frac{1}{2}} \|u\|_{k+1} |||v_h|||,$$

which completes the proof. □

We obtain the following a priori error estimate:

Theorem 5.10

Let $0 \leq k \leq 2$ and $u \in H^{k+1}(\Omega)$. Then we have the a priori error estimate

$$|||u - u_h||| \leq C h^{k+\frac{1}{2}} \|u\|_{H^{k+1}(\Omega)}. \tag{5.2.30}$$

Proof. By the triangle inequality we have

$$|||u - u_h||| \leq |||u - \mathcal{C}_h u||| + |||\mathcal{C}_h u - u_h|||.$$

The first term can be estimated by using Lemma 5.8 as follows:

$$\begin{aligned}|||u - \mathcal{C}_h u||| \leq & c_0 \|u - \mathcal{C}_h u\| + \sqrt{\frac{1}{2}\int_{\mathcal{S}_h^\partial} |\boldsymbol{\beta}_n| (u - \mathcal{C}_h u)^2 + \frac{1}{2}\int_{\mathcal{S}_h^{\mathrm{int}}} |\boldsymbol{\beta}_n| [u - \mathcal{C}_h u]^2} \\ & + \sum_{K\in\mathcal{T}_h} \delta_K^{\frac{1}{2}} \|\pi_K^1\boldsymbol{\beta}\cdot\nabla (u - \mathcal{C}_h u)\|_K \\ & \leq C_1 h^{k+1} \|u\|_{k+1} + C_2 h^{k+\frac{1}{2}} \|u\|_{k+1} + C_3 h^{k+\frac{1}{2}} \|u\|_{k+1} \leq C h^{k+\frac{1}{2}} \|u\|_{k+1}.\end{aligned}$$

Applying Lemma 5.7 to the second term yields

$$|||\mathcal{C}_h u - u_h||| \leq \frac{1}{\gamma} \sup_{v_h \in V_h^{1,2}\setminus\{0\}} \frac{a_{CD}(\mathcal{C}_h u - u_h, v_h)}{|||v_h|||}.$$

Using Galerkin orthogonality and the consistency error estimate (5.2.29) gives

$$\sup_{v_h \in V_h^{1,2}\setminus\{0\}} \frac{a_{CD}(\mathcal{C}_h u - u_h, v_h)}{|||v_h|||} = \sup_{v_h \in V_h^{1,2}\setminus\{0\}} \frac{a_{CD}(\mathcal{C}_h u - u, v_h)}{|||v_h|||} \leq C h^{k+\frac{1}{2}} \|u\|_{H^{k+1}(\Omega)},$$

which completes the proof. □

We have shown that the CG1-DG2 method is stable in the context of advection equations on triangular meshes and converges with the same rate as the DG method or the streamline diffusion method. In the following we will extend the method to solve diffusion problems.

5.3. The CG1-DG2 method for Poisson's equation

The following results are valid for triangular and quadrilateral meshes. Let us consider Poisson's equation

$$\begin{aligned} -\Delta u &= f \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{5.3.1}$$

where $f \in L^2(\Omega)$. Following the primal flux formulation of the DG method (4.3.13), we discretize the numerical fluxes in the sense of the symmetric interior penalty Galerkin (SIPG), the non-symmetric interior penalty Galerkin (NIPG) and the Baumann-Oden (BO) method (see Table 4.1) and define the following forms

$$\begin{aligned} b_{CD}(u,\varphi) &= \int_{\mathcal{T}_h} \nabla u \cdot \nabla \varphi \, d\mathbf{x} - \int_{\mathcal{S}_h} \{\nabla u\} \cdot \mathbf{n}_S[\varphi] \, d\mathbf{s} - s \int_{\mathcal{S}_h} \{\nabla \varphi\} \cdot \mathbf{n}_S[u] \, d\mathbf{s}, \\ J_h^\mu(u,\varphi) &= \int_{\mathcal{S}_h} \mu[u][\varphi] \, d\mathbf{s}, \\ B_{CD}(u,\varphi) &:= b_{CD}(u,\varphi) + J_h^\mu(u,\varphi), \end{aligned}$$

where

$$s = \begin{cases} -1 & \text{for } NIPG, BO, \\ +1 & \text{for } SIPG, \end{cases} \quad \text{and } \mu = \begin{cases} \frac{\eta_S}{h_S} & \text{for } NIPG, SIPG, \\ 0 & \text{for } BO, \end{cases}$$

with $\eta_S > 0$ as defined in (4.3.19).
The discretized CG1-DG2 problem is then given by

$$\text{Find } u_h \in V_h^{1,2}: \quad B_{CD}(u_h, \varphi_h) = \int_{\mathcal{T}_h} f \varphi_h \, d\mathbf{x} \quad \forall \varphi_h \in V_h^{1,2}. \tag{5.3.2}$$

The norms induced by this problem are $\|.\|_\mu$ and $|||.|||_\mu$ as defined in (4.3.27). Note that $|||.|||_\mu$ is a norm on $V_h^{1,2}$, not just a seminorm.

In the following we will show consistency, boundedness and coercivity. We remark that the results can easily be extended to the DG method.

Lemma 5.11: Consistency

The discretization (5.3.2) of (5.3.1) is consistent, i.e., the exact solution $u \in H^2(\Omega)$ of (5.3.1) satisfies

$$B_{CD}(u,\varphi_h) = \int_{\mathcal{T}_h} f\varphi_h \,\mathrm{d}\mathbf{x} \quad \forall\, \varphi_h \in V_h^{1,2}. \tag{5.3.3}$$

Proof. If we integrate by parts in the first integral of the bilinear form $b_{CD}(u,\varphi_h)$, we obtain

$$\int_{\mathcal{T}_h} \nabla u \cdot \nabla \varphi_h \,\mathrm{d}\mathbf{x} = -\int_{\mathcal{T}_h} \Delta u \varphi_h \,\mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h^{\text{int}}} \{\nabla u\} \cdot \mathbf{n}_S [\varphi_h] \,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\text{int}}} [\nabla u] \cdot \mathbf{n}_S \{\varphi_h\} \,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial}} \nabla u \cdot \mathbf{n}_\Omega \varphi_h \,\mathrm{d}\mathbf{s},$$

for any $\varphi_h \in V_h^{1,2}$. Inserting this into the bilinear form we derive

$$\begin{aligned} b_{CD}(u,\varphi_h) = &-\int_{\mathcal{T}_h} \Delta u \varphi_h \,\mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h^{\text{int}}} [\nabla u] \cdot \mathbf{n}_S \{\varphi_h\} \,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial}} \nabla u \cdot \mathbf{n}_\Omega \varphi_h \,\mathrm{d}\mathbf{s} \\ &- s\int_{\mathcal{S}_h} \{\nabla \varphi_h\} \cdot \mathbf{n}_S [u] \,\mathrm{d}\mathbf{s} - \int_{\mathcal{S}_h^{\partial}} \{\nabla u\} \cdot \mathbf{n}_S [\varphi_h] \,\mathrm{d}\mathbf{s}. \end{aligned}$$

Using $[u] = 0$, $\{\nabla u\} = \nabla u$ and $[\nabla u] \cdot \mathbf{n}_S = 0$ gives

$$\begin{aligned} b_{CD}(u,\varphi_h) &= -\int_{\mathcal{T}_h} \Delta u \varphi_h \,\mathrm{d}\mathbf{x}, \\ J_h^\mu(u,\varphi_h) &= 0, \end{aligned}$$

independently of s and μ. Applying (5.3.1) leads to (5.3.3). □

Note that consistency also follows directly from Lemma 4.14, since $V_h^{1,2} \subset V_{DG,h}^2$.

Lemma 5.12: Boundedness

The bilinear form $B_{CD}(\cdot,\cdot)$ can be bounded as follows

$$|B_{CD}(v,\varphi)| \le 2|||v|||_\mu |||\varphi|||_\mu, \quad \forall v,\varphi \in H^2(\mathcal{T}_h). \tag{5.3.4}$$

Note that $B_{CD}(\cdot,\cdot) = b_{CD}(\cdot,\cdot)$ in the case of the Baumann-Oden method.

Proof. We follow [75] and obtain

$$|B_{CD}(v,\varphi)| \le |b_{CD}(v,\varphi)| + \left|J_h^\mu(v,\varphi)\right|,$$

where $v,\varphi \in H^2(\mathcal{T}_h)$. The first term gives

$$\begin{aligned} |b_{CD}(v,\varphi)| &= \left| \int_{\mathcal{T}_h} \nabla v \cdot \nabla \varphi \,\mathrm{d}\mathbf{x} - \int_{\mathcal{S}_h} \{\nabla v\} \cdot \mathbf{n}_S [\varphi] \,\mathrm{d}\mathbf{s} - s\int_{\mathcal{S}_h} \{\nabla \varphi\} \cdot \mathbf{n}_S [v] \,\mathrm{d}\mathbf{s} \right| \\ &\le \left| \int_{\mathcal{T}_h} \nabla v \cdot \nabla \varphi \,\mathrm{d}\mathbf{x} \right| + \left| \int_{\mathcal{S}_h} \{\nabla v\} \cdot \mathbf{n}_S [\varphi] \,\mathrm{d}\mathbf{s} \right| + \left| \int_{\mathcal{S}_h} \{\nabla \varphi\} \cdot \mathbf{n}_S [v] \,\mathrm{d}\mathbf{s} \right| \\ &\le |||v|||_\mu |||\varphi|||_\mu. \end{aligned}$$

In the case of the SIPG and NIPG method we also have to bound the second term

$$\left|J_h^\mu(v,\varphi)\right| = \int_{\mathcal{S}_h} \mu[v][\varphi]\,\mathrm{d}\mathbf{s} \leq \sqrt{\int_{\mathcal{S}_h} \mu[v]^2\,\mathrm{d}\mathbf{s}} \sqrt{\int_{\mathcal{S}_h} \mu[\varphi]^2\,\mathrm{d}\mathbf{s}} \leq |||v|||_\mu |||\varphi|||_\mu,$$

where $\mu > 0$. This completes the proof. □

Details of the proof can be found in [15].

Lemma 5.13: Coercivity - NIPG/SIPG

Let $B_{CD}(\cdot,\cdot)$ be the bilinear form from (5.3.2). Then for $s = \pm 1$ and $\mu > 0$ large enough there exists a constant $C > 0$ such that

$$B_{CD}(v_h, v_h) \geq C|||v_h|||_\mu^2, \quad \forall v_h \in V_h^{1,2}. \tag{5.3.5}$$

Coercivity for the SIPG and NIPG method can also be extended to the discontinuous space $V_{DG,h}^k$ (see, e.g., [75]).

In the case of the Baumann-Oden method we only have weak stability [4]:

$$b_{CD}(v,v) = |v|_{H^1(\mathcal{T}_h)}^2 \quad \forall v \in V_h^{1,2}. \tag{5.3.6}$$

Since numerical results indicate stability of the Baumann-Oden method for the DG space $V_{DG,h}^k$ with $k \geq 2$ [10], we assume that the Baumann-Oden method is also stable in the context of the CG1-DG2 space.

The a priori error estimates from Theorem 4.15 are also satisfied by the CG1-DG2 solution $u_h \in V_h^{1,2}$ and are summarized in Table 5.1. In contrast to the DG method, we need to restrict $0 \leq k \leq 2$.

Method	order of L^2-error	order of H^1-error
SIPG	$k+1$	k
NIPG	k	k
BO, $k \geq 2$	k	k

Table 5.1: Order of convergence in the L^2- and H^1-norms

This section showed that the analytical results obtained for the DG method can be directly transferred to the CG1-DG2 method if diffusion problems are considered.

6

Numerical results for scalar equations

In this chapter we will verify the analytical results presented in the previous chapters by numerical studies. We apply the CG1-DG2 method to advection-dominated problems as well as diffusion problems and compare the numerical solutions with those obtained by the following standard finite element methods

CG1	continuous Galerkin, linear elements;
CG2	continuous Galerkin, quadratic elements;
DG1	discontinuous Galerkin, linear elements;
DG2	discontinuous Galerkin, quadratic elements.

In the case of convection-dominated problems we will also compare the numerical results with those obtained by the streamline upwind Petrov-Galerkin (SUPG) method [18] which stabilizes CG1/CG2 by using the modified test functions

$$\tilde{v}_h = v_h + \tau\boldsymbol{\beta}\cdot\nabla v_h, \tag{6.0.1}$$

where τ is a free parameter. For our computations we choose

$$\tau = \frac{h}{2\|\boldsymbol{\beta}\|_K}.$$

A priori error estimates for the SUPG method result in the same order of convergence as for the DG method (see (4.2.32)).

In order to compare these methods in a quantitative way, we will estimate the expected order of convergence by the formula [69]

$$EOC = \log_2\left(\frac{E(2h)}{E(h)}\right), \tag{6.0.2}$$

where $E(h) = \|u - u_h\|$ is the error in a specific norm between the exact solution u and the numerical approximation u_h and h denoting the mesh size.

h	CG1	CG2	CG1-DG2	DG1	DG2
1/32	1089	4225	7233	6144	12288
1/64	4225	16641	28801	24576	49152
1/128	16641	66049	114945	98304	196608
1/256	66049	263169	459265	393216	786432

Table 6.1: Numbers of DOFs on triangular meshes

h	CG2	serendipity CG1-DG2	Q2-CG1-DG2	DG2
1/32	4225	5185	6209	9216
1/64	16641	20609	24705	36864
1/128	66049	82177	98561	147456
1/256	263169	328193	393729	589824

Table 6.2: Numbers of DOFs on quadrilateral meshes

In Tables 6.1 and 6.2 the numbers of degrees of freedom (DOFs) are displayed for $\Omega = (0,1) \times (0,1)$ using different mesh sizes. The number of DOFs for the CG1-DG2 method lies between the numbers for the DG2 method and the CG2 method. Due to the lack of the bubble functions the serendipity CG1-DG2 space has fewer DOFs than the Q2-CG1-DG2 space.

In the following numerical studies we will illustrate that the CG1-DG2 method converges with the same rates as the DG method, and has improved stability properties compared to the CG method, in the context of triangular and serendipity elements. All computations were carried out on uniform meshes.

The presented methods have been implemented in the open-source hp-FEM/hp-DG library Hermes [82]. Temporal discretization was performed using the Crank-Nicolson scheme.

6.1. Steady advection with a constant velocity field

In the first numerical example, we consider the steady advection equation

$$\nabla \cdot (\boldsymbol{\beta} u) = 0 \qquad \text{in } \Omega = (0,1) \times (0,1), \tag{6.1.1}$$

with the constant velocity field

$$\boldsymbol{\beta}(x,y) = (1,1). \tag{6.1.2}$$

The exact solution which is also used to prescribe the inflow boundary conditions (see Fig. 6.1) is given by

$$u(x,y) = \begin{cases} \cos\left(\frac{\pi(x-y-0.25)}{0.5}\right), & \text{if } 0 < x-y < 0.5, \\ 0, & \text{otherwise.} \end{cases} \tag{6.1.3}$$

We will start with the results obtained on triangular meshes. In Table 6.3 the EOCs for the different methods are shown. CG1-DG2 exhibits similar rates as DG2 and SUPG2, namely ≈ 3.0 in the L^2-norm and ≈ 2.4 in the $|||.|||$-norm. The linear methods (CG1,DG1,SUPG1) as well as the CG2 method have the same order in the $|||.|||$-norm, namely ≈ 1.5. In the L^2-norm CG1 and CG2 deliver the same rates of ≈ 2.0. We remark that the expected rate for CG1 is 1.0. For DG1 the calculated EOCs vary between 2.3 and 2.6 and for SUPG1 between 1.1 and 2.9.

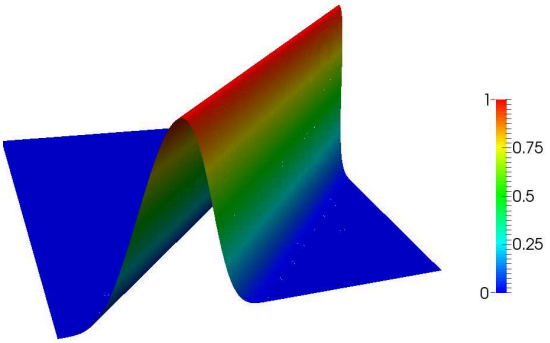


Figure 6.1: Steady advection with constant velocity: exact solution

	CG1				CG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|\|\|u-u_h\|\|\|$	EOC	$\|u-u_h\|_{L^2}$	EOC	$\|\|\|u-u_h\|\|\|$	EOC
1/32	6.9972e-03	-	2.5456e-01	-	1.0519e-03	-	5.1675e-02	-
1/64	1.6188e-03	2.12	8.6183e-02	1.56	2.5814e-04	2.03	1.7786e-02	1.54
1/128	3.9877e-04	2.02	3.0203e-02	1.51	6.4265e-05	2.01	6.2461e-03	1.51
1/256	9.9353e-05	2.00	1.0655e-02	1.50	1.6050e-05	2.00	2.2045e-03	1.50

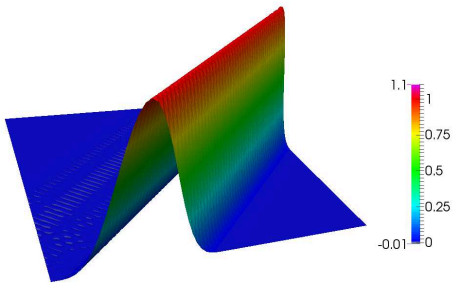
	SUPG1				SUPG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|\|\|u-u_h\|\|\|$	EOC	$\|u-u_h\|_{L^2}$	EOC	$\|\|\|u-u_h\|\|\|$	EOC
1/32	6.4745e-02	-	1.7141e-01	-	3.1723e-03	-	2.5839e-02	-
1/64	1.5688e-02	1.12	7.5725e-02	1.18	1.7070e-04	4.21	4.9682e-03	2.38
1/128	2.3256e-03	2.75	2.7950e-02	1.44	1.3525e-05	3.66	8.9850e-04	2.47
1/256	3.0710e-04	2.92	9.9267e-03	1.49	1.5217e-06	3.15	1.5988e-04	2.49

	DG1				DG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|\|\|u-u_h\|\|\|$	EOC	$\|u-u_h\|_{L^2}$	EOC	$\|\|\|u-u_h\|\|\|$	EOC
1/32	1.7634e-02	-	1.5672e-01	-	5.5899e-04	-	1.6385e-02	-
1/64	3.2256e-03	2.45	6.4826e-02	1.27	6.0418e-05	3.21	2.8573e-03	2.52
1/128	5.5026e-04	2.55	2.3804e-02	1.45	7.3572e-06	3.04	5.0193e-04	2.51
1/256	1.1047e-04	2.32	8.4874e-03	1.49	9.1422e-07	3.01	8.8556e-05	2.50

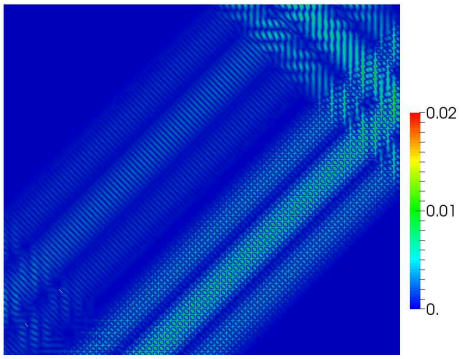
	CG1-DG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|\|\|u-u_h\|\|\|$	EOC
1/32	1.1379e-03	-	5.1738e-02	-
1/64	1.7835e-04	2.67	1.2261e-02	2.08
1/128	2.5369e-05	2.81	2.4555e-03	2.32
1/256	3.3222e-06	2.93	4.5222e-04	2.44

Table 6.3: Steady advection with constant velocity on triangular meshes

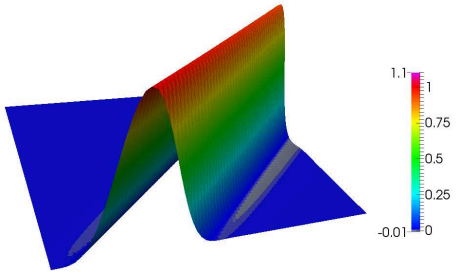
In Fig. 6.2 the solutions and the corresponding error distributions of the continuous schemes are shown. It can be seen that the continuous Galerkin solutions exhibit oscillations in elements where the solution is constantly zero. In the case of the SUPG method the errors are concentrated in the region where the solution is not equal to zero. This supports the theoretical results stating that the SUPG method has better stability properties than the CG method.



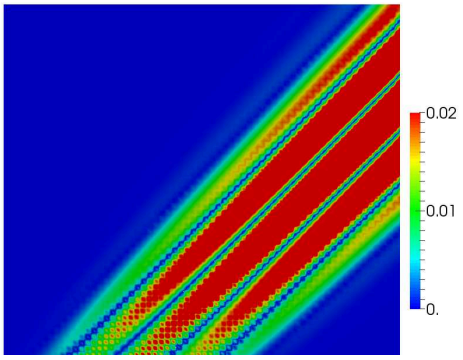
(a) CG1 solution



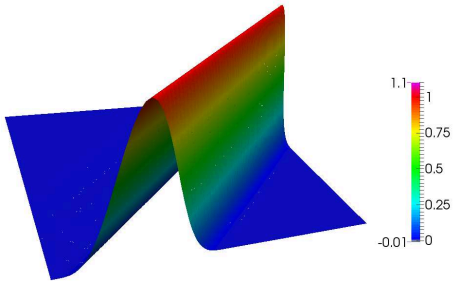
(b) CG1 error distribution



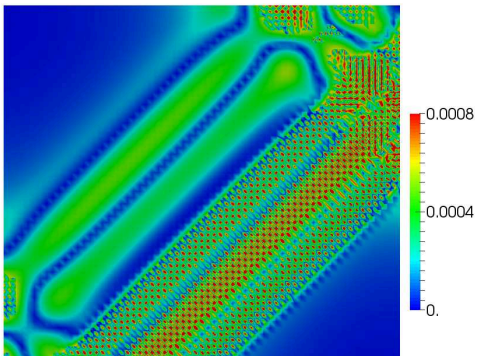
(c) SUPG1 solution



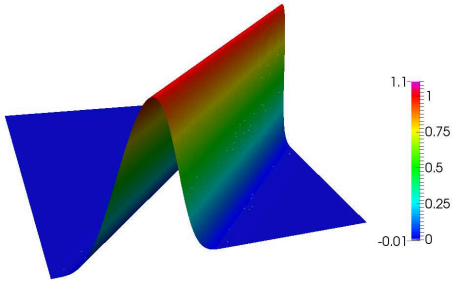
(d) SUPG1 error distribution



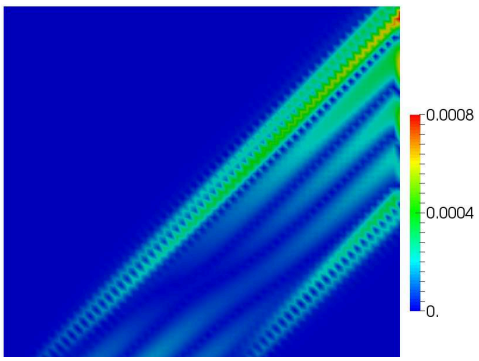
(e) CG2 solution



(f) CG2 error distribution

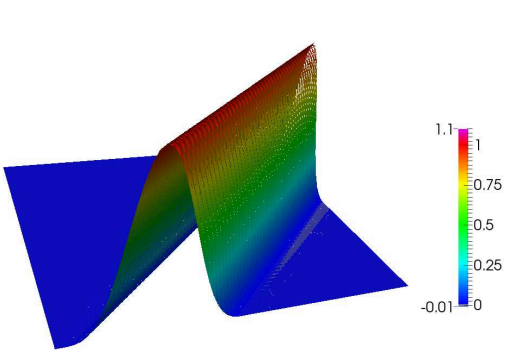


(g) SUPG2 solution

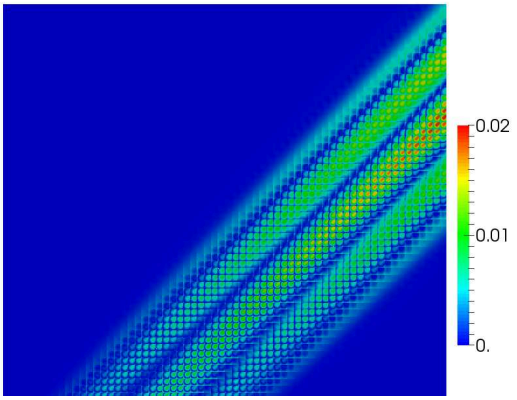


(h) SUPG2 error distribution

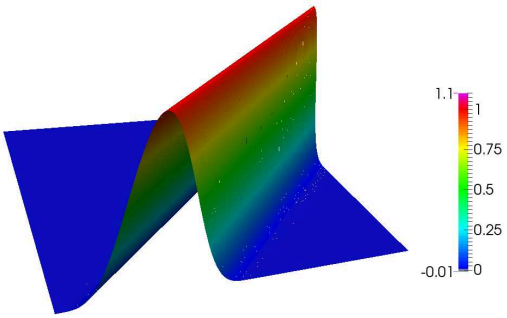
Figure 6.2: Steady advection with constant velocity: continuous schemes on triangular meshes



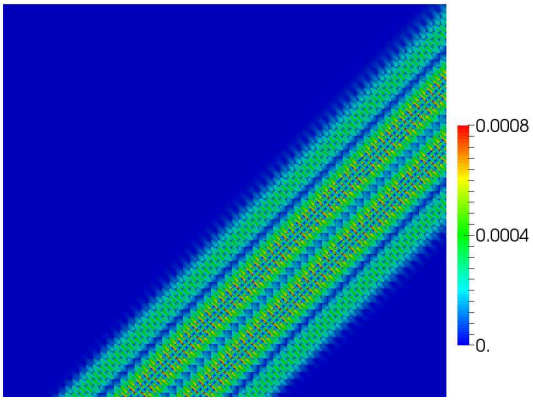
(a) DG1 solution



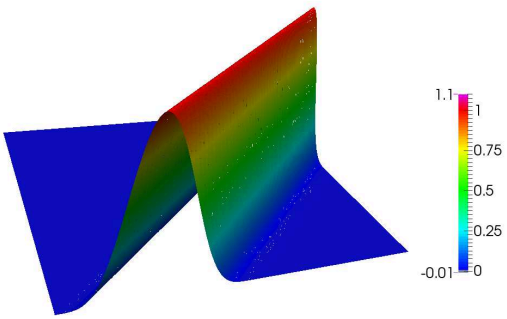
(b) DG1 error distribution



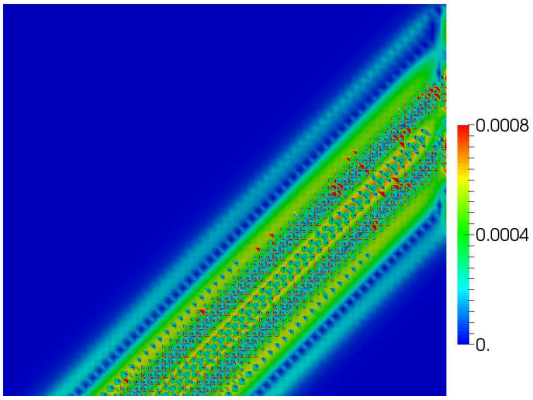
(c) DG2 solution



(d) DG2 error distribution



(e) CG1-DG2 solution

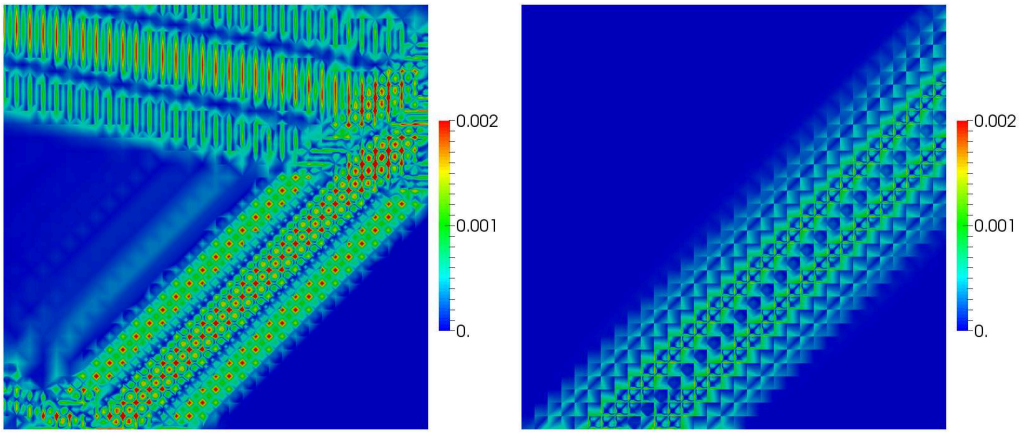


(f) CG1-DG2 error distribution

Figure 6.3: Steady advection with constant velocity: (semi-)discontinuous schemes on triangular meshes

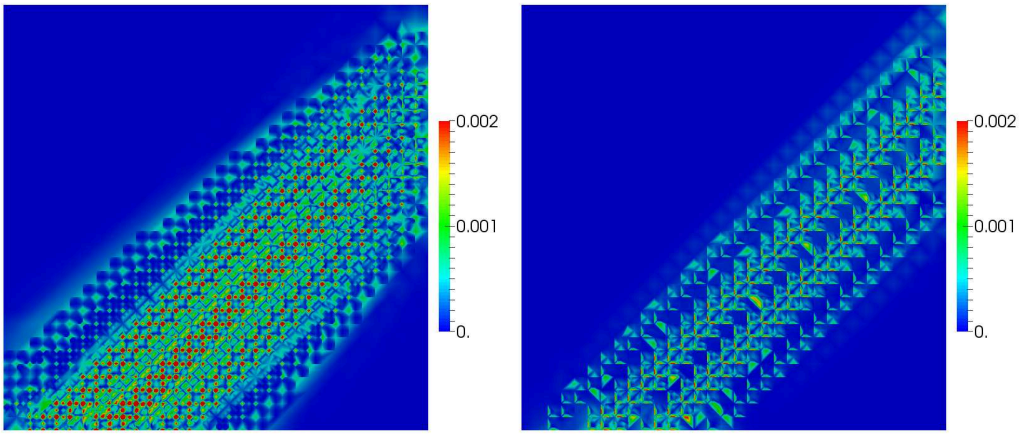
In Fig. 6.3 the solutions and the corresponding error distributions for the (semi-)discontinuous schemes (DG and CG1-DG2) are shown. All methods exhibit similar error profiles as those derived for the SUPG method (cf. Fig. 6.2).

We will now present the results obtained on quadrilateral meshes. In Fig. 6.4 the error distributions for the different methods are plotted. They are in good agreement with the results obtained on triangular meshes. We see that the CG2 method produces oscillations in elements where the solution is constantly zero. The DG2 and the serendipity CG1-DG2 method have errors only where the solution is not constant. The Q2-CG1-DG2 method also produces local errors and in a larger domain than the serendipity version.



(a) CG2 error distribution

(b) DG2 error distribution



(c) Q2-CG1-DG2 error distribution

(d) Serendipity CG1-DG2 error distribution

Figure 6.4: Steady advection with constant velocity field on quadrilateral meshes

In Table 6.4 the L^2-error and the $|||.|||$-error with the corresponding EOCs for different mesh sizes are given. The convergence rates for the CG2 and DG2 methods are the same as in the triangular cases, namely 2.0 and 3.0 in the L^2-norm and 1.5 and 2.5 in the $|||.|||$-norm. The serendipity CG1-DG2 method exhibits the same rates as the DG2 method. In the Q2-CG1-DG2 case the rates are lower than in the serendipity case. Also the absolute error values are larger in the Q2 case.

	CG2				DG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|\|\|u-u_h\|\|\|$	EOC	$\|u-u_h\|_{L^2}$	EOC	$\|\|\|u-u_h\|\|\|$	EOC
1/32	5.4864e-04	-	1.7115e-02	-	2.1228e-04	-	8.7242e-03	-
1/64	1.3385e-04	2.04	5.8799e-03	1.54	2.5498e-05	3.06	1.5444e-03	2.50
1/128	3.3264e-05	2.01	2.0632e-03	1.51	3.1680e-06	3.01	2.7306e-04	2.50
1/256	8.3040e-06	2.00	7.2797e-04	1.50	3.9543e-07	3.00	4.8277e-05	2.50
	Q2-CG1-DG2				serendipity CG1-DG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|\|\|u-u_h\|\|\|$	EOC	$\|u-u_h\|_{L^2}$	EOC	$\|\|\|u-u_h\|\|\|$	EOC
1/32	6.2317e-04	-	2.2607e-02	-	2.6530e-04	-	1.0100e-02	-
1/64	1.0251e-04	2.60	5.7333e-03	1.98	3.1024e-05	3.10	1.8128e-03	2.48
1/128	1.6175e-05	2.66	1.3079e-03	2.13	3.8258e-06	3.02	3.2175e-04	2.49
1/256	2.3692e-06	2.77	2.7219e-04	2.26	4.7663e-07	3.00	5.6941e-05	2.50

Table 6.4: Steady advection with constant velocity on quadrilateral meshes

In summary, we see that all methods which are known to be stable (SUPG, DG) as well as the CG1-DG2 method exhibit only local errors, i.e., only errors in elements where the solution is not constant, whereas the continuous Galerkin method produces global errors.

In the following we will restrict the computations to piecewise-quadratic spaces and compare the CG1-DG2 method only with the continuous and the discontinuous method, i.e., we neglect the SUPG method, since this method gives similar results as those obtained by the DG method.

6.2. Steady advection-reaction with a constant velocity field

In the following we will consider the advection-reaction equation

$$u + \nabla \cdot (\boldsymbol{\beta} u) = f \qquad \text{in } \Omega = (0,1) \times (0,1) \tag{6.2.1}$$

with the constant velocity field

$$\boldsymbol{\beta}(x,y) = (0.5, 1). \tag{6.2.2}$$

An exact solution which is also used to prescribe the inflow boundary conditions (see Fig. 6.5) is given by

$$u(x,y) = \sin(2\pi x)\sin(2\pi y). \tag{6.2.3}$$

The right-hand side can be calculated as $f := u + \nabla \cdot (\boldsymbol{\beta} u)$, where u is the exact solution given by (6.2.3). This example can also be found in [12].

In Fig. 6.6 the L^2-errors and the $|||.|||$-errors versus mesh size are plotted for CG2, CG1-DG2 and DG2 method on triangular meshes in log-scale. The DG2 and CG1-DG2 method exhibit a convergence rate of 3.0 whereas the CG2 method yields only a rate of 2.0 in the L^2-norm. In the $|||.|||$-norm the rates are 2.5 and 1.5, respectively.

In Fig. 6.7 the L^2-error and the $|||.|||$-error versus different mesh sizes are displayed for quadrilateral meshes. For the CG2, DG2 and the serendipity CG1-DG2 method we obtain the same rates as in the triangular case. For the Q2-CG1-DG2 method we have the same rates as for the CG2 method, namely 2.0 in the L^2-norm and 1.5 in the $|||.|||$-norm.

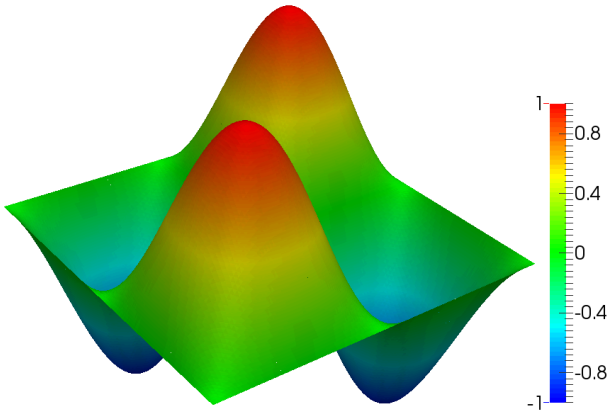
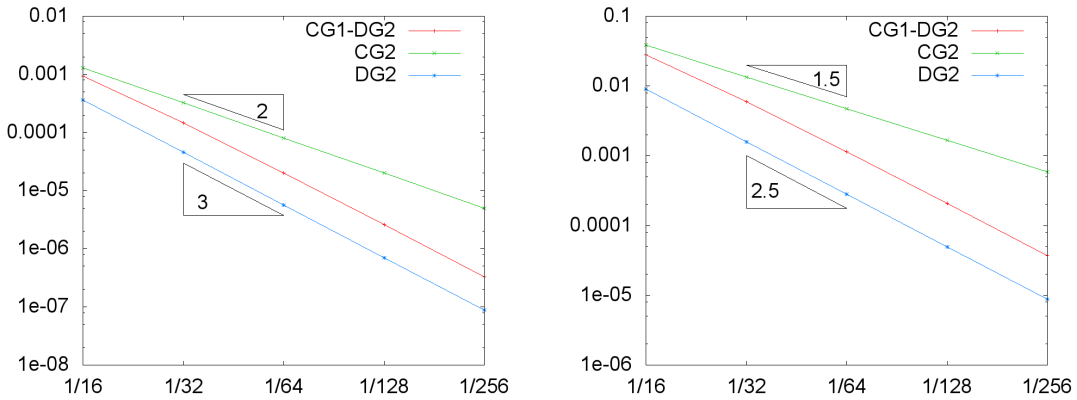
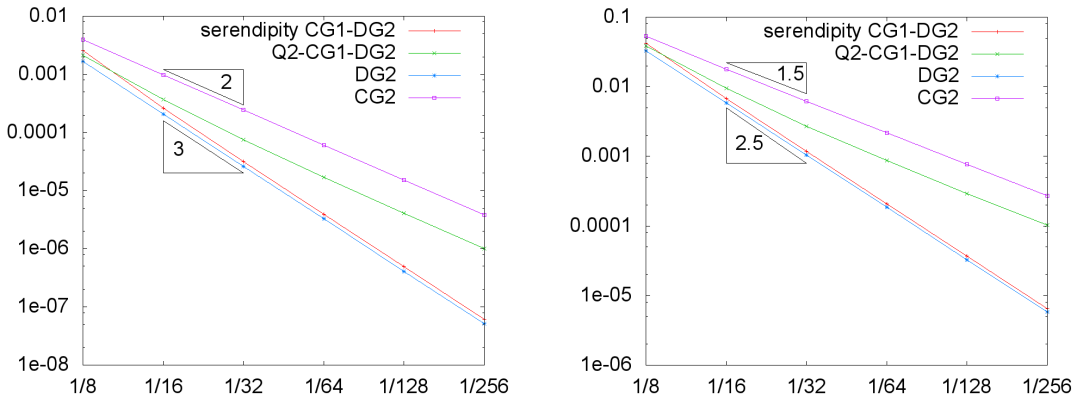


Figure 6.5: Steady advection-reaction with constant velocity: exact solution



(a) L^2-error versus mesh size (b) $|||.|||$-error versus mesh size

Figure 6.6: Steady advection-reaction with constant velocity on triangular meshes



(a) L^2-error versus mesh size (b) $|||.|||$-error versus mesh size

Figure 6.7: Steady advection-reaction with constant velocity on quadrilateral meshes

6.3. Steady advection with a rotating velocity field

In this example, we consider the steady advection equation

$$\nabla \cdot (\boldsymbol{\beta} u) = 0 \qquad \text{in } \Omega = (0,1) \times (0,1), \tag{6.3.1}$$

with the rotating velocity field

$$\boldsymbol{\beta}(x,y) = (y, 1-x). \tag{6.3.2}$$

The exact solution, which also implies the inflow boundary condition, can be seen in Fig. 6.8.

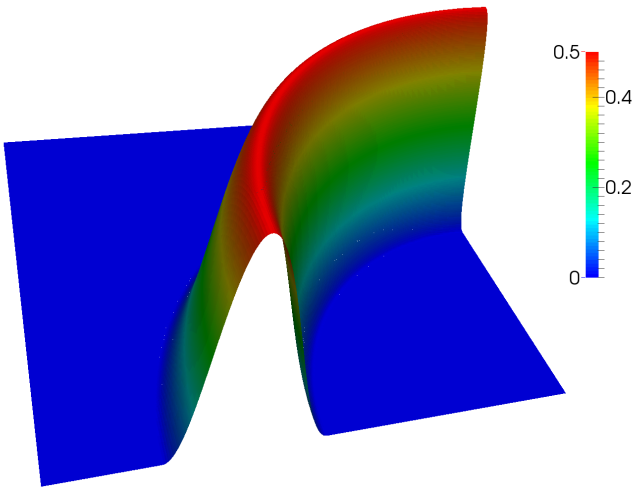
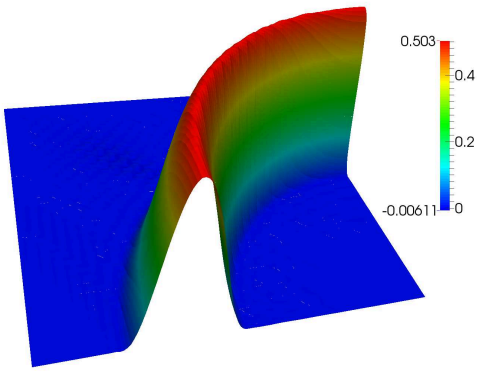


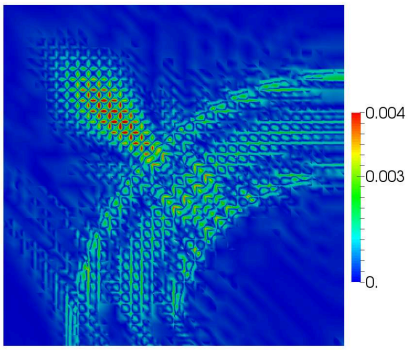
Figure 6.8: Steady advection with rotating velocity field: exact solution

In Fig. 6.9 the solutions and the error distributions for the CG2, the DG2 and the CG1-DG2 method are displayed for triangular meshes. It can be seen that there are oscillations in the continuous case where the solution should be constantly zero. The DG2 as well as the CG1-DG2 method produce large errors only where the solution is not equal to zero.

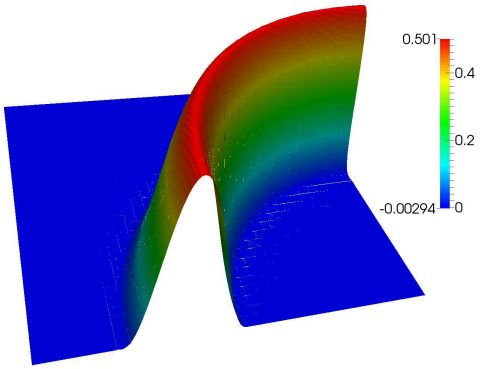
In Fig. 6.10 the errors of the different methods are shown for quadrilateral meshes. As in the triangular case the CG2 method exhibits oscillations in elements where the solution is constant. The error profiles of the DG2 and the serendipity CG1-DG2 method look very similar. In both cases the errors are found in elements where the solution is not constantly zero. In the Q2-CG1-DG2 case we observe errors in a larger domain than in the serendipity case.



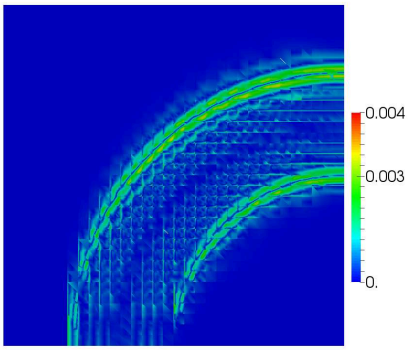
(a) CG2 solution



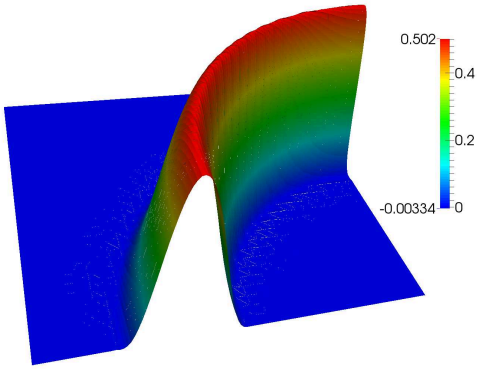
(b) CG2 error distribution



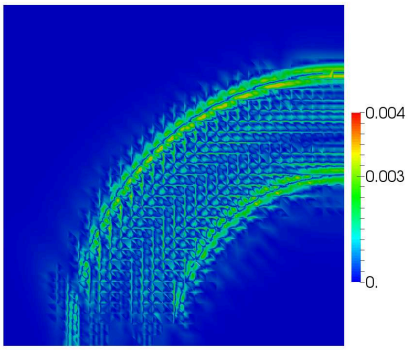
(c) DG2 solution



(d) DG2 error distribution

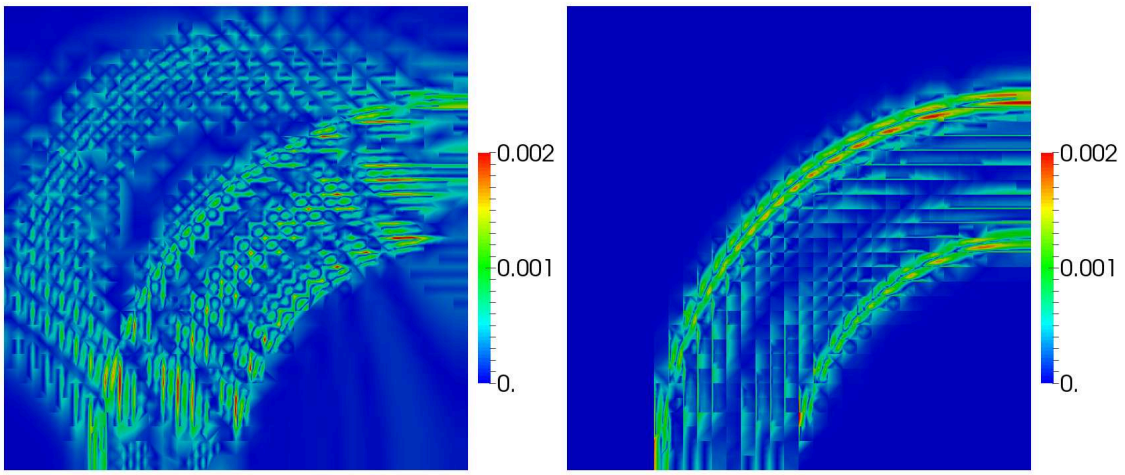


(e) CG1-DG2 solution



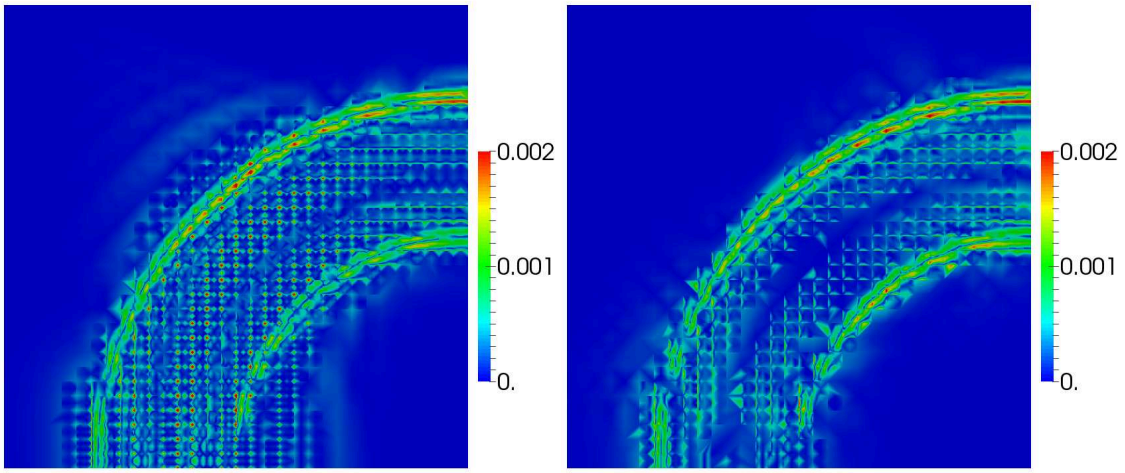
(f) CG1-DG2 error distribution

Figure 6.9: Steady advection with rotating velocity field on triangular meshes



(a) CG2 error distribution

(b) DG2 error distribution



(c) Q2-CG1-DG2 error distribution

(d) Serendipity CG1-DG2 error distribution

Figure 6.10: Steady advection with rotating velocity field on quadrilateral meshes

6.4. Steady advection-reaction with a non-constant velocity field

This example is adopted from [43] and considers the advection-reaction equation

$$\boldsymbol{\beta} \cdot \nabla u + cu = f \qquad \text{in } \Omega = (-1,1) \times (-1,1), \tag{6.4.1}$$

where the velocity field $\boldsymbol{\beta}$ and the coefficient c are given by

$$\boldsymbol{\beta}(x,y) = (2-y^2, 2-x), \quad c(x,y) = 1 + (1+x)(1+y)^2. \tag{6.4.2}$$

The term f on the right-hand side is chosen in such a way that the exact solution (see Fig. 6.11) is given by

$$u(x,y) = \sin\left(\frac{\pi}{8}(1+x)(1+y)^2\right) + 1. \tag{6.4.3}$$

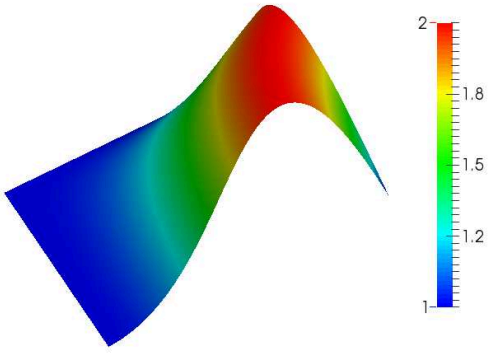
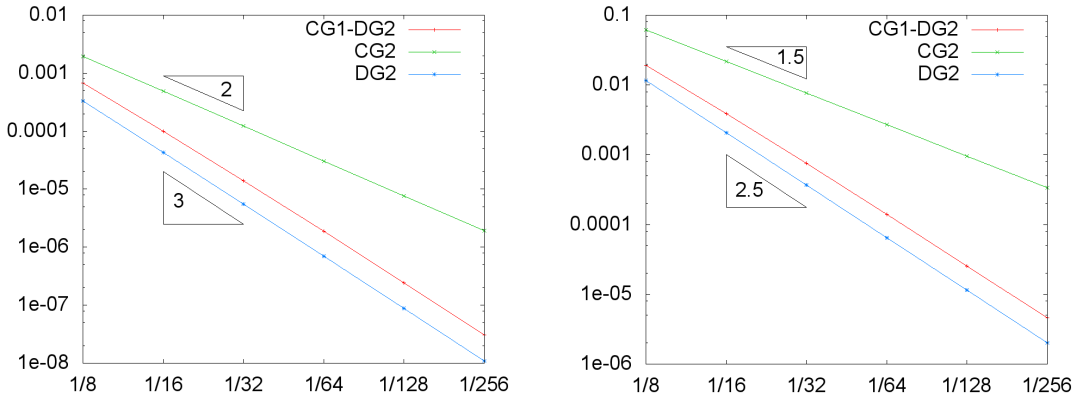


Figure 6.11: Steady advection-reaction with non-constant velocity: exact solution

In Fig. 6.12 the L^2-errors and the $|||.|||$-errors versus mesh size are plotted for CG2, CG1-DG2 and DG2 on triangular meshes. As in section 6.2 we see that DG2 and CG1-DG2 converge with a rate of 3.0 in the L^2-norm whereas CG2 converges only with a rate of 2.0. In the $|||.|||$-norm we obtain convergence rates of 2.5 and 1.5, respectively.

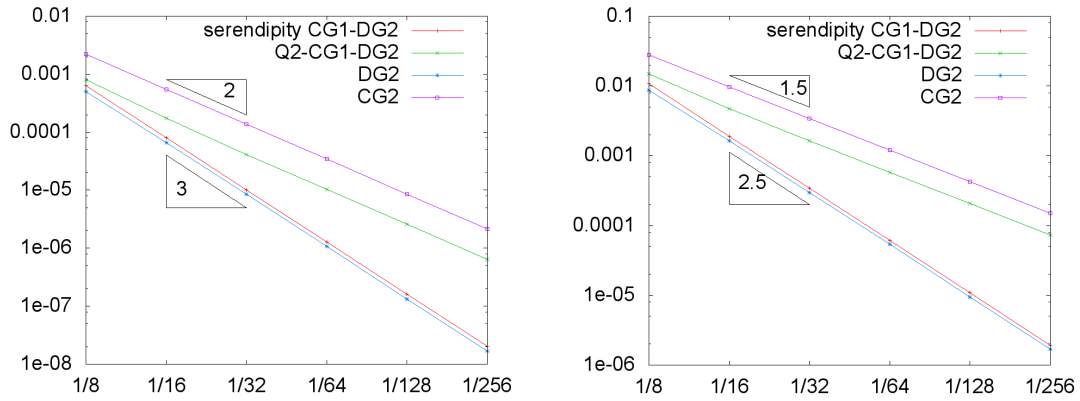


(a) L^2-error versus mesh size

(b) $|||.|||$-error versus mesh size

Figure 6.12: Steady advection-reaction with non-constant velocity on triangular meshes

In Fig. 6.13 the errors in the L^2- and $|||.|||$-norm versus mesh size are displayed for quadrilateral meshes. As before, we see that CG2 and Q2-CG1-DG2 exhibit the same EOCs, namely ≈ 2.0 in the L^2-norm and ≈ 1.5 in the $|||.|||$-norm. The DG2 and the serendipity CG1-DG2 method converge with one order more than the other methods.



(a) L^2-error versus mesh size

(b) $|||.|||$-error versus mesh size

Figure 6.13: Steady advection-reaction with non-constant velocity on quadrilateral meshes

6.5. Solid body rotation problem

In the following we will consider the so-called *solid body rotation* problem [48, 69]. We solve

$$u_t + \nabla \cdot (\boldsymbol{\beta} u) = 0 \qquad \text{in } \Omega = (0,1) \times (0,1), \tag{6.5.1}$$

where the velocity field

$$\boldsymbol{\beta}(x,y) = (0.5 - y, x - 0.5) \tag{6.5.2}$$

describes counterclockwise rotations about the center of Ω. After each full rotation ($t = 2\pi k$, $k \in \mathbb{N}$) the exact solution u coincides with the initial data u_0. This is the challenge of this test problem since numerical schemes for advection problems often fail to preserve the shape of u_0.

Following LeVeque [69], we simulate the rotation of a slotted cylinder, a sharp cone, and a smooth hump as displayed in Fig. 6.14. Initially, the shape of each body is described by a function $G(x,y)$ which is defined within the circle

$$r(x,y) = \frac{1}{r_0}\sqrt{(x-x_0)^2 + (y-y_0)^2} \leq 1$$

of radius $r_0 = 0.15$ and center (x_0, y_0).

For the slotted cylinder, the center is given by $(x_0, y_0) = (0.5, 0.75)$ and

$$G(x,y) = \begin{cases} 1 & \text{if } |x - x_0| \geq 0.025 \text{ or } y \geq 0.85, \\ 0 & \text{otherwise.} \end{cases}$$

The peak of the cone is located at $(x_0, y_0) = (0.5, 0.25)$ and its shape function is

$$G(x,y) = 1 - r(x,y).$$

The hump is centered at $(x_0, y_0) = (0.25, 0.5)$ and its geometry is given by

$$G(x,y) = \frac{1 + \cos(\pi r(x,y))}{4}.$$

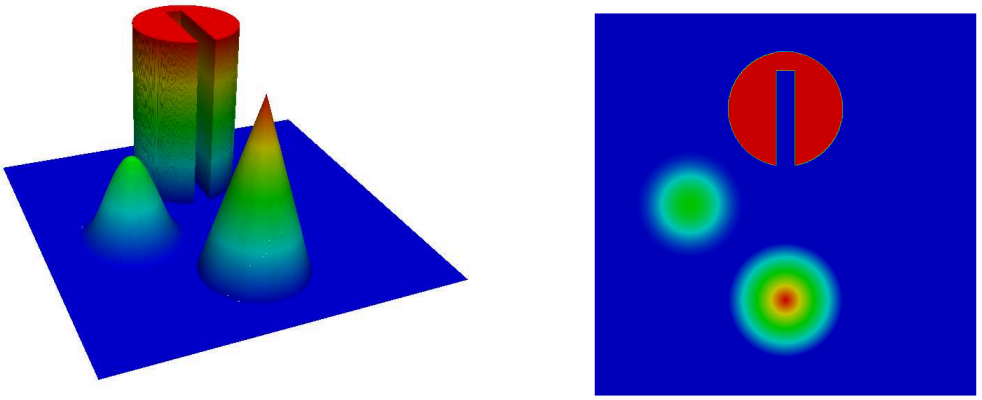
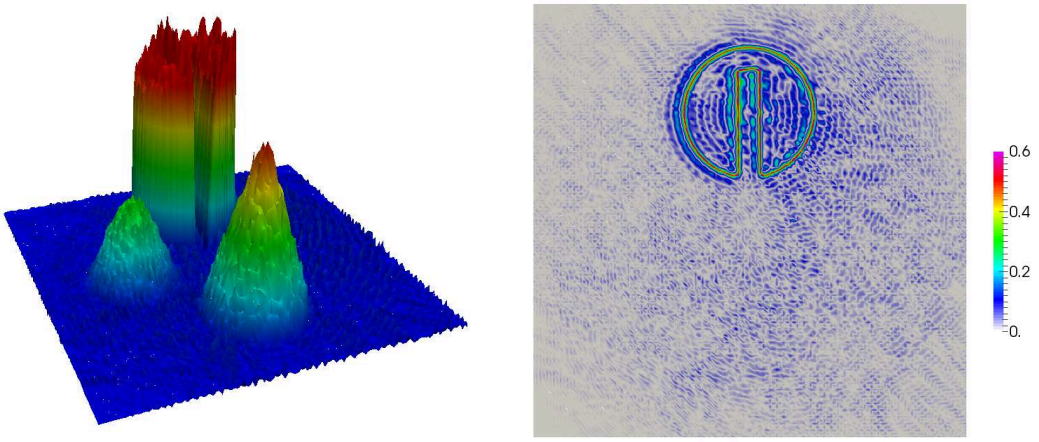


Figure 6.14: Solid body rotation: initial data and exact solution at $t = 2\pi$

In Figures 6.15 and 6.16 the numerical solutions and the corresponding error distributions are displayed for triangular meshes. The CG2 solution exhibits large under- and overshoots in the whole domain whereas the DG2 and the CG1-DG2 method produce only small oscillations near the boundary of the slotted cylinder. The hump and the cone, apart from the sharp peak, are preserved very well by the DG2 and the CG1-DG2 method.



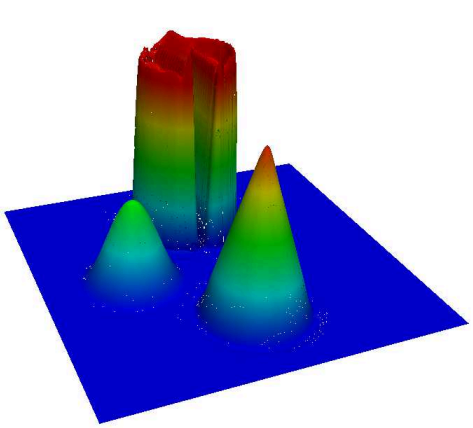
(a) CG2 solution

(b) CG2 error distribution

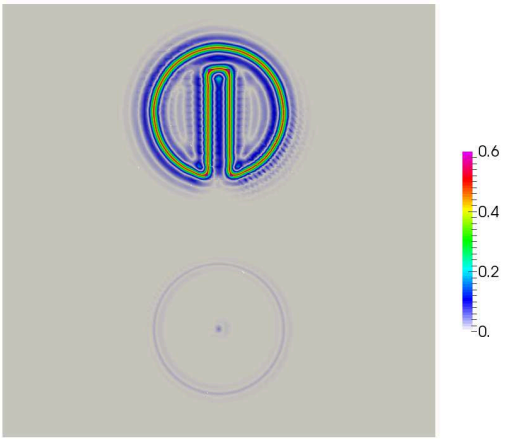
Figure 6.15: Solid body rotation problem: solution and error distribution at $t = 2\pi$ on a triangular mesh

In Fig. 6.17 we see the error profiles obtained by the Q2- and the serendipity CG1-DG2 method. As in the triangular case there are large errors in the neighborhood of the cylinder and smaller errors at the peak of the cone and at its bottom. The hump is very well resolved so that no error is visible.

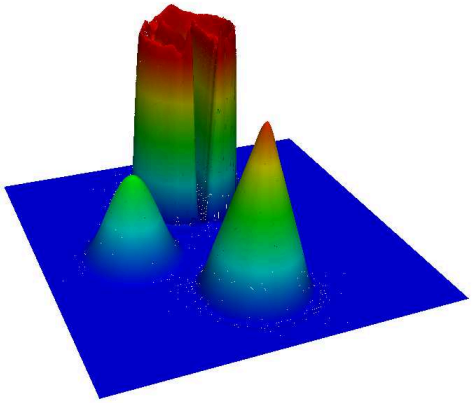
We have now seen that the CG1-DG2 method converges with the expected rates on triangular meshes and that the results are comparable with those obtained by the DG method. On quadrilateral meshes the serendipity CG1-DG2 method seems to have better stability properties than the Q2-CG1-DG2 method and also higher convergence rates in the case of advection-reaction equations.



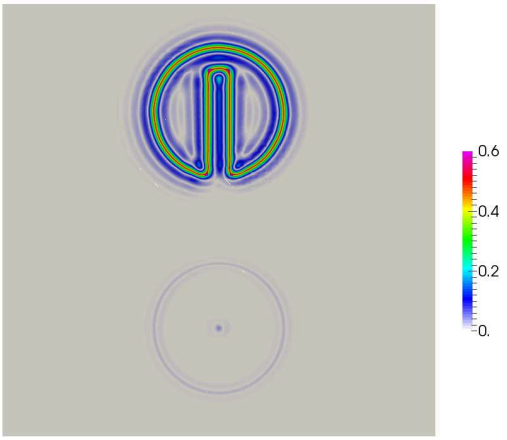
(a) CG1-DG2 solution



(b) CG1-DG2 error distribution

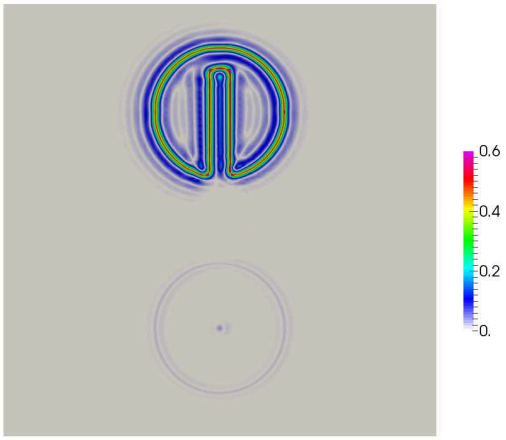


(c) DG2 solution

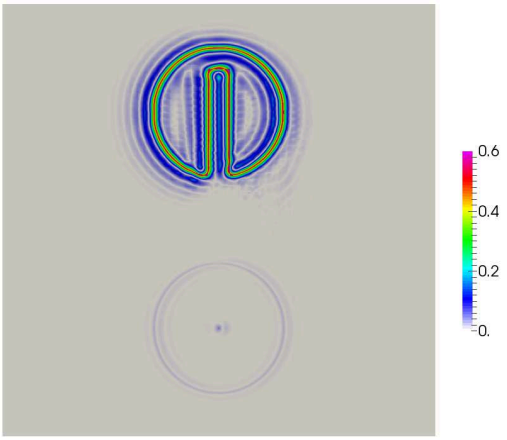


(d) DG2 error distribution

Figure 6.16: Solid body rotation problem: solution and error distribution at $t = 2\pi$ for triangular meshes



(a) Q2-CG1-DG2



(b) Serendipity CG1-DG2

Figure 6.17: Solid body rotation problem: error distribution at $t = 2\pi$

6.6. Poisson's equation

In the following, we will investigate the behavior of the different methods when it comes to solving Poisson's equation

$$-\Delta u = f \qquad \text{in } \Omega. \tag{6.6.1}$$

Let us consider the domain $\Omega = (0,1) \times (0,1)$. The right-hand side is chosen in such a way that the exact solution (see Fig. 6.18) is given by [37]

$$u(x,y) = \sin(\frac{1}{2}\pi x)\sin(\frac{1}{2}\pi y). \tag{6.6.2}$$

On the boundary $\partial\Omega$ we prescribe Dirichlet boundary conditions which correspond to the exact solution (6.6.2).

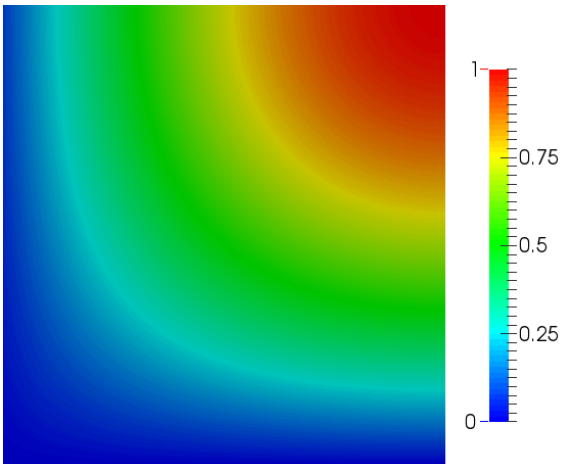


Figure 6.18: Poisson's equation: exact solution

In Tables 6.5 and 6.6 the L^2- and H^1-errors as well as the corresponding EOCs are displayed for triangular meshes. The CG2 method has rates of ≈ 3.0 in the L^2-norm and ≈ 2.0 in the H^1-norm. The same rates are achieved by the SIPG-method for the DG2 and the CG1-DG2 space. The Baumann-Oden and the NIPG method have the same rates in the H^1-norm. In the L^2-norm the rate decreases to ≈ 2.0. These results are in good agreement with analysis (see Theorem 3.8 and Theorem 4.15).

	CG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC
1/16	5.1804e-05	-	3.2354e-03	-
1/32	6.6666e-06	2.96	8.3182e-04	1.96
1/64	8.4518e-07	2.98	2.1064e-04	1.98
1/128	1.0639e-07	2.99	5.2982e-05	1.99
1/256	1.3346e-08	2.99	1.3285e-05	2.00

Table 6.5: Poisson's equation: EOCs for the continuous Galerkin method (triangular meshes)

Baumann-Oden								
	CG1-DG2				DG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC
1/16	3.7815e-04	-	4.4344e-03	-	1.8044e-04	-	3.3486e-03	-
1/32	7.9723e-05	2.25	1.1541e-03	1.94	5.1225e-05	1.82	8.1250e-04	2.04
1/64	1.7454e-05	2.19	2.9382e-04	1.97	1.3716e-05	1.90	1.9963e-04	2.03
1/128	4.0168e-06	2.12	7.4072e-05	1.99	3.5490e-06	1.95	4.9444e-05	2.01
1/256	9.5915e-07	2.07	1.8592e-05	1.99	9.0258e-07	1.98	1.2301e-05	2.01
SIPG, $\sigma_S = 10$								
	CG1-DG2				DG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC
1/16	6.1559e-05	-	4.7917e-03	-	4.7230e-05	-	4.0811e-03	-
1/32	6.9938e-06	3.14	1.1031e-03	2.12	4.9121e-06	3.27	9.1047e-04	2.16
1/64	8.2661e-07	3.08	2.6234e-04	2.07	5.4114e-07	3.18	2.1235e-04	2.10
1/128	1.0028e-07	3.04	6.3807e-05	2.04	6.2612e-08	3.11	5.1064e-05	2.06
1/256	1.2342e-08	3.02	1.5723e-05	2.02	7.4934e-09	3.06	1.2505e-05	2.03
NIPG, $\sigma_S = 1$								
	CG1-DG2				DG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC
1/16	2.9743e-04	-	3.8718e-03	-	1.8201e-04	-	3.0715e-03	-
1/32	6.2010e-05	2.26	1.0015e-03	1.95	4.9540e-05	1.88	7.5354e-04	2.02
1/64	1.3533e-05	2.20	2.5446e-04	1.98	1.2946e-05	1.94	1.8629e-04	2.02
1/128	3.1131e-06	2.12	6.4106e-05	1.99	3.3086e-06	1.97	4.6290e-05	2.01
1/256	7.4344e-07	2.07	1.6086e-05	1.99	8.3623e-07	1.98	1.1536e-05	2.00

Table 6.6: Poisson's equation: EOCs for different methods (triangular meshes)

In Tables 6.7 - 6.10 we present the results of the convergence study for quadrilateral meshes. As expected we get the same rates as in the triangular case. The CG2 method and the SIPG method have a rate of ≈ 3.0 in the L^2-norm and ≈ 2.0 in the H^1-norm. We obtain an order of ≈ 3.0 in the L^2-norm and ≈ 2.0 in the H^1-norm for the Baumann-Oden method and the NIPG method. In the CG1-DG2 case the EOCs are independent of the element type.

	CG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC
1/8	3.0698e-05	-	1.5960e-03	-
1/16	3.8450e-06	3.00	3.9898e-04	2.00
1/32	4.8087e-07	3.00	9.9743e-05	2.00
1/64	6.0117e-08	3.00	2.4936e-05	2.00
1/128	7.5148e-09	3.00	6.2339e-06	2.00

Table 6.7: Poisson's equation: EOCs for the continuous Galerkin method (quadrilateral meshes)

Baumann-Oden								
	serendipity CG1-DG2				Q2-CG1-DG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC
1/8	2.9990e-04	-	2.9139e-03	-	3.2660e-04	-	2.9673e-03	-
1/16	7.2372e-05	2.05	7.2824e-04	2.00	7.5614e-05	2.11	7.3429e-04	2.01
1/32	1.7583e-05	2.04	1.8175e-04	2.00	1.7964e-05	2.07	1.8246e-04	2.01
1/64	4.3185e-06	2.03	4.5381e-05	2.00	4.3639e-06	2.04	4.5465e-05	2.00
1/128	1.0691e-06	2.01	1.1337e-05	2.00	1.0746e-06	2.02	1.1347e-05	2.00

	DG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC
1/8	5.5841e-04	-	3.8445e-03	-
1/16	1.4638e-04	1.93	9.7730e-04	1.98
1/32	3.7076e-05	1.98	2.4592e-04	1.99
1/64	9.3004e-06	2.00	6.1645e-05	2.00
1/128	2.3271e-06	2.00	1.5430e-05	2.00

Table 6.8: Poisson's equation: EOCs for the Baumann-Oden method (quadrilateral meshes)

SIPG, $\sigma_S = 10$								
	serendipity CG1-DG2				Q2-CG1-DG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC
1/8	2.5201e-05	-	1.9519e-03	-	2.4864e-05	-	1.9420e-03	-
1/16	2.9051e-06	3.12	4.6062e-04	2.08	2.8870e-06	3.11	4.5973e-04	2.08
1/32	3.4569e-07	3.07	1.1142e-04	2.05	3.4467e-07	3.07	1.1133e-04	2.05
1/64	4.2044e-08	3.04	2.7365e-05	2.03	4.1984e-08	3.04	2.7355e-05	2.03
1/128	5.1800e-09	3.02	6.7786e-06	1.40	5.1763e-09	3.02	6.7774e-06	2.01

	DG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC
1/8	2.2371e-05	-	1.8988e-03	-
1/16	2.5237e-06	3.15	4.5149e-04	2.07
1/32	2.9543e-07	3.09	1.0969e-04	2.04
1/64	3.5560e-08	3.05	2.7005e-05	2.02
1/128	4.3554e-09	3.03	6.6976e-06	2.01

Table 6.9: Poisson's equation: EOCs for the SIPG method (quadrilateral meshes)

NIPG, $\sigma_S = 1$								
	serendipity CG1-DG2				Q2-CG1-DG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC
1/8	2.2674e-04	-	2.4052e-03	-	2.4117e-04	-	2.4217e-03	-
1/16	5.3339e-05	2.09	5.9708e-04	2.01	5.5071e-05	2.13	5.9890e-04	2.02
1/32	1.2813e-05	2.06	1.4863e-04	2.01	1.3015e-05	2.08	1.4884e-04	2.01
1/64	3.1310e-06	2.03	3.7068e-05	2.00	3.1550e-06	2.04	3.7093e-05	2.00
1/128	7.7329e-07	2.02	9.2554e-06	2.00	7.7619e-07	2.02	9.2585e-06	2.00

	DG2			
h	$\|u-u_h\|_{L^2}$	EOC	$\|u-u_h\|_{H^1}$	EOC
1/8	3.5382e-04	-	2.6782e-03	-
1/16	8.8853e-05	1.99	6.6725e-04	2.01
1/32	2.2083e-05	2.01	1.6637e-04	2.00
1/64	5.4918e-06	2.01	4.1526e-05	2.00
1/128	1.3685e-06	2.00	1.0373e-05	2.00

Table 6.10: Poisson's equation: EOCs for the NIPG method (quadrilateral meshes)

We have seen, that the CG1-DG2 method achieves the same convergence rates as the DG2 method in the context of pure diffusion. We will now add an advective term to investigate the behavior for convection-diffusion problems.

6.7. Time-dependent convection-diffusion equation

In the following, we consider a time-dependent problem, which was already presented in [15] and describes a moving cone-like object diffusing as time goes on. The underlying equation is given by

$$\frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{\beta} u - \varepsilon \nabla u) = 0 \quad \text{in } \Omega = (-1,1) \times (-1,1) \tag{6.7.1}$$

with velocity field $\boldsymbol{\beta} = (-y, x)$ and diffusion coefficient $\varepsilon = 10^{-3}$. The object is defined by

$$u(\mathbf{x},t) = \frac{1}{4\pi\varepsilon t} e^{-\frac{r^2}{4\varepsilon t}}, \quad r^2 = (x-\hat{x})^2 + (y-\hat{y})^2, \tag{6.7.2}$$

where $\hat{x}$ and $\hat{y}$ are the time-dependent coordinates of the moving peak

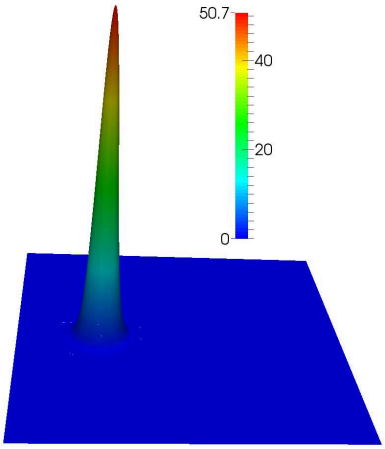
$$\hat{x}(t) = x_0 \cos t - y_0 \sin t, \quad \hat{y}(t) = -x_0 \sin t + y_0 \cos t. \tag{6.7.3}$$

At time $t = 0$ the solution is $u(\mathbf{x},0) = \delta(x_0, y_0)$, where δ is the Dirac delta distribution. Since we cannot directly obtain a reasonable initial value, we start the numerical computations at $t_{start} = \frac{\pi}{2}$ and stop those after one full rotation ($T = 2.5\pi$). The initial peak is set to $(x_0, y_0) = (0, 0.5)$.

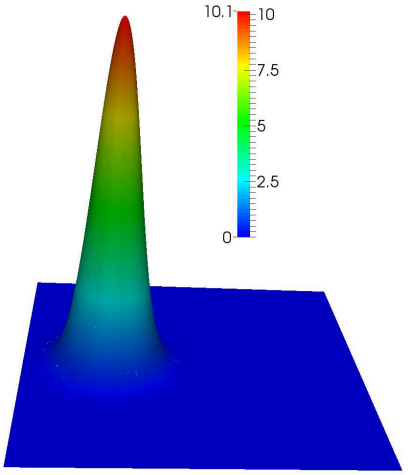
The exact solution at $t_{start} = \frac{\pi}{2}$ and at $T = 2.5\pi$ can be seen in Figures 6.19a and 6.19b. Note that both pictures are scaled with respect to their maximum values.

The CG1-DG2 solution obtained by the Baumann-Oden method is displayed in Fig. 6.19c for triangular meshes. In Fig. 6.19d the exact solution is compared with the CG1-DG2 solution along the line $y = 0$. They match almost everywhere, however, the top of the numerical solution is a little bit lower than the one of the exact solution.

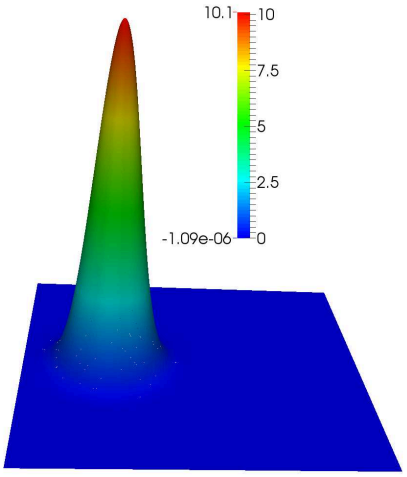
In Figures 6.20 and 6.21 the L^2-error versus mesh size are plotted for triangular and quadrilateral meshes, respectively. It can be seen that the Baumann-Oden method has a convergence rate of ≈ 2.0 whereas the SIPG and the CG2 method converge at a rate of ≈ 3.0. The errors of the SIPG and CG2 method nearly match such that the error curves are superimposed.



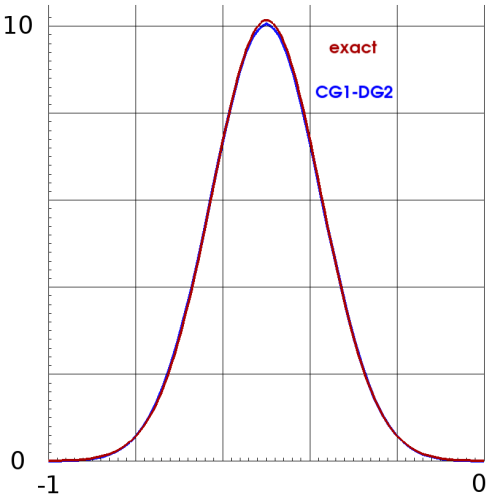
(a) Exact solution at $t_{start} = \frac{\pi}{2}$



(b) Exact solution at $T = 2.5\pi$



(c) CG1-DG2 solution at $T = 2.5\pi$ using the Baumann-Oden method



(d) CG1-DG2 solution and exact solution at $T = 2.5\pi$ along $y = 0$

Figure 6.19: Time-dependent convection-diffusion equation on triangular meshes

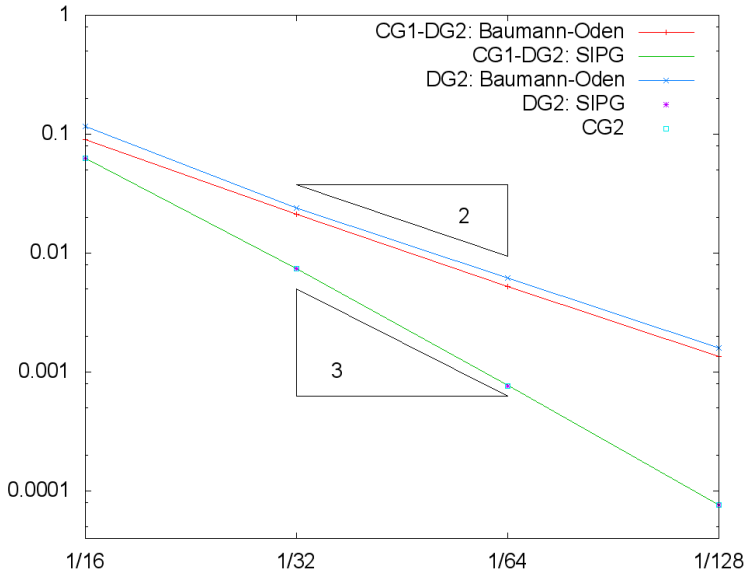


Figure 6.20: Time-dependent convection-diffusion equation: L^2-error vs. mesh size on triangular meshes

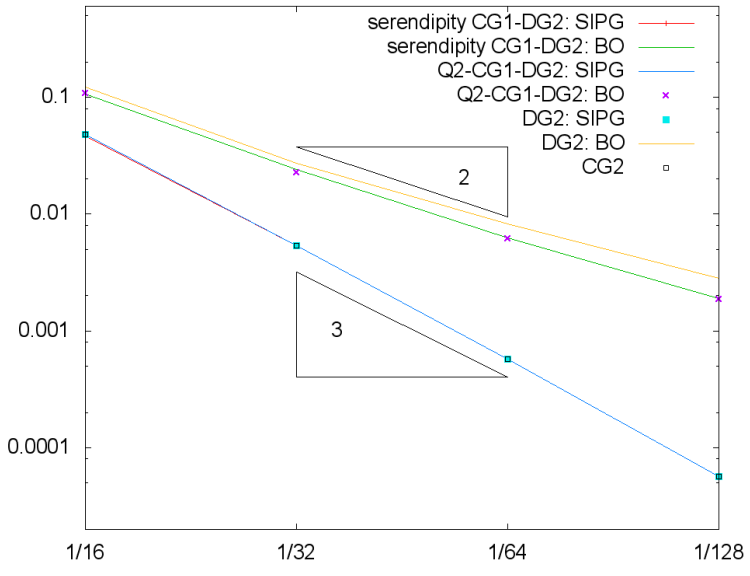


Figure 6.21: Time-dependent convection-diffusion equation: L^2-error vs. mesh size on quadrilateral meshes

6.8. A hump changing its height

This example was taken from [49] where different stabilization techniques (e.g., the SUPG method) for continuous finite element methods were compared. The problem is a convection-diffusion-reaction equation

$$u_t + \boldsymbol{\beta} \cdot \nabla u - \varepsilon \Delta u + cu = f \qquad \text{in } \Omega = (0,1) \times (0,1), \tag{6.8.1}$$

where

$$\boldsymbol{\beta}(x,y) = (2,3), \quad \varepsilon = 10^{-6}, \quad c = 1. \tag{6.8.2}$$

The exact solution is given by

$$\begin{aligned} u(x,y;t) = {} & 16 \sin(\pi t) x(1-x) y(1-y) \\ & \cdot \left(\frac{1}{2} + \frac{1}{\pi} \arctan \left[\frac{2}{\sqrt{\varepsilon}} (0.25^2 - (x-0.5)^2 - (y-0.5)^2) \right] \right), \end{aligned} \tag{6.8.3}$$

which prescribes a hump changing its height over time.

In Fig. 6.22 the exact solution at $t = 0.5$ can be seen. At $t = 2.0$ the hump vanishes so that the exact solution is constantly zero.

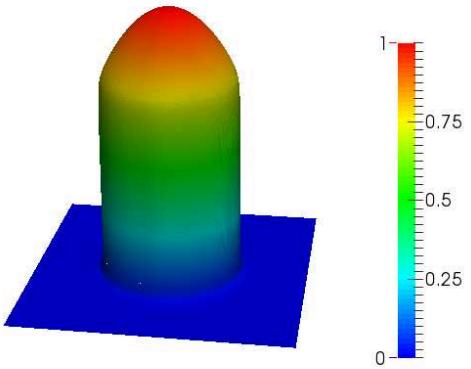
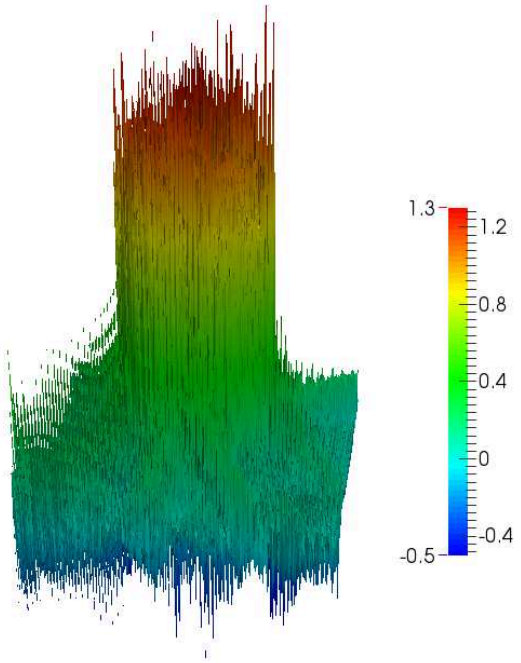


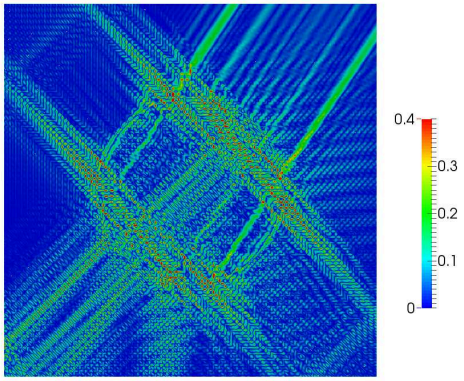
Figure 6.22: Hump changing its height: exact solution at $t = 0.5$

We will first present the results obtained for triangular meshes. In Fig. 6.23 the CG2 solutions and their error distributions for different times are shown. It can be seen that the CG2 solution exhibits global oscillations. The DG2 and CG1-DG2 solution for the Baumann-Oden method (see Figures 6.24 and 6.25) have oscillations at the hump and behind the hump in the direction of the velocity field. We mention that in [49] it was observed that the tested stabilization techniques cannot prevent the occurrence of oscillations in the direction of the convection. This observation is in good agreement with results obtained here for the DG2 and the CG1-DG2 method.

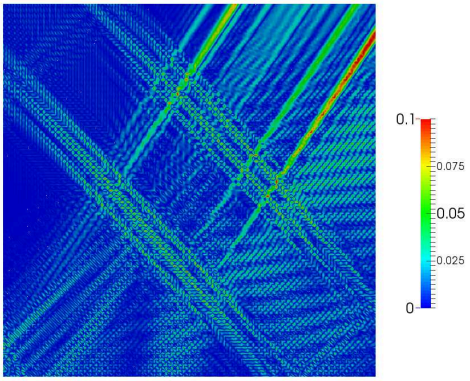
In Fig. 6.26 the solutions and error distributions at $t = 0.5$ for the Q2- and the serendipity CG1-DG2 method are presented. In both cases the Baumann-Oden method for the discretization of the diffusive part was used. It can be seen that there are oscillations in the direction of the velocity field. In contrast to the triangular case under- and overshoots are also present in front of the hump. However, these oscillations are larger in the Q2 case and in a larger domain than in the serendipity case.



(a) Solution at $t = 0.5$



(b) Error distribution at $t = 0.5$

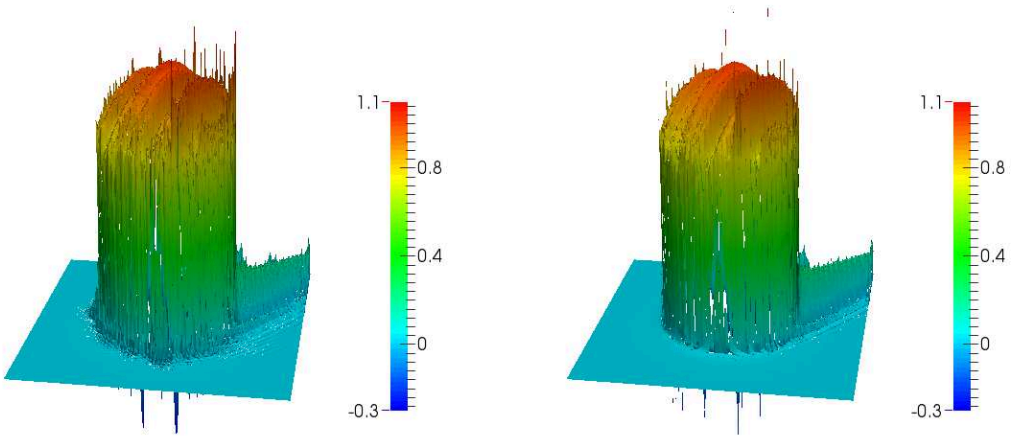


(c) Error distribution at $t = 2.0$

Figure 6.23: Hump changing its height: CG2 method on triangular meshes

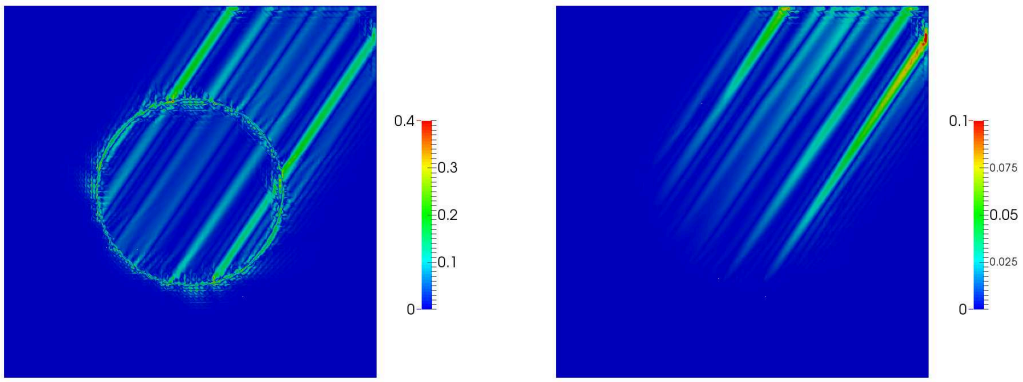
In Fig. 6.27 the error distributions at $t = 2.0$ are displayed. The hump has vanished at this time. Using the Q2-CG1-DG2 method we obtain oscillations in the streamline direction from the lower left to the upper right corner, whereas the serendipity CG1-DG2 method produces errors only in the region of the vanished hump and behind it. It can be seen that the serendipity CG1-DG2 method produces better results than the Q2-CG1-DG2 method.

In summary, our numerical experiments for advection(-reaction) and (advection-)diffusion problems showed that the triangular and the serendipity CG1-DG2 method produce results similar to those obtained by the DG2 method. In the context of advection equations, we have seen that we have better stability properties than the CG method, which can produce oscillations in the whole domain even for smooth solutions.

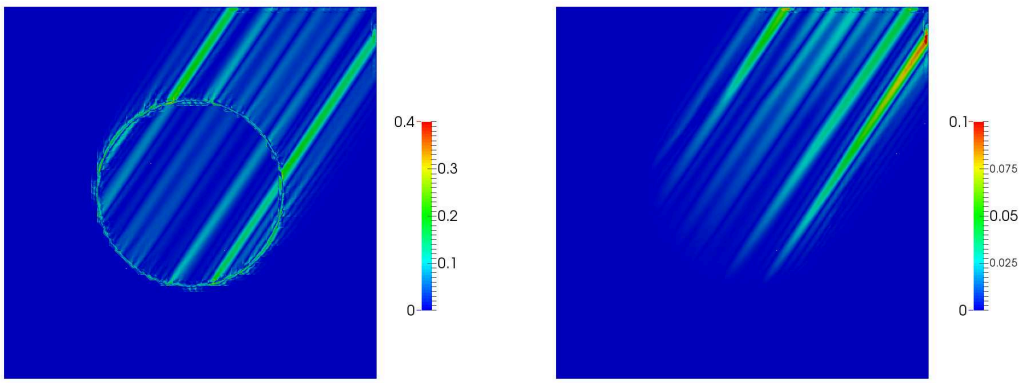


(a) CG1-DG2 solution (b) DG2 solution

Figure 6.24: Hump changing its height: numerical solutions at $t = 0.5$ on triangular meshes

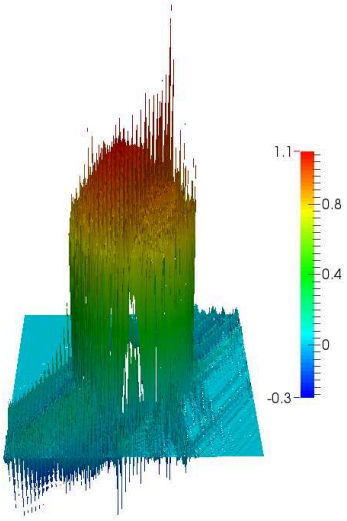


(a) CG1-DG2 error distribution at $t = 0.5$ (b) CG1-DG2 error distribution at $t = 2.0$

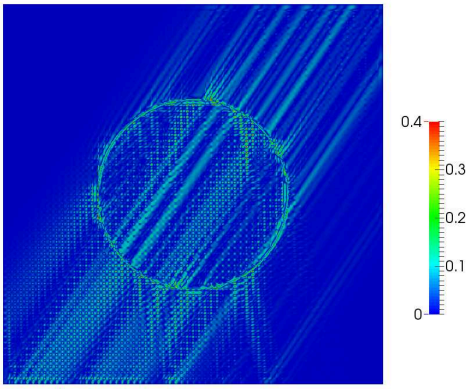


(c) DG2 error distribution at $t = 0.5$ (d) DG2 error distribution at $t = 2.0$

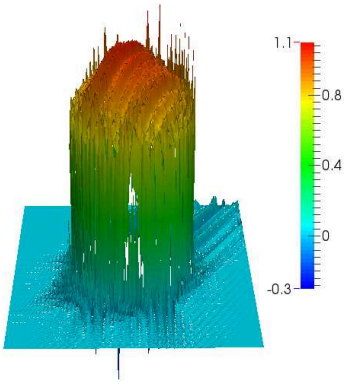
Figure 6.25: Hump changing its height: error distributions on triangular meshes



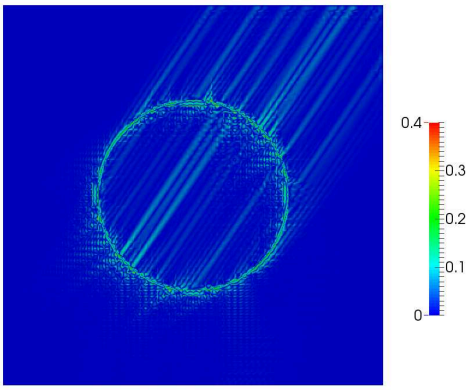
(a) Q2-CG1-DG2 solution



(b) Q2-CG1-DG2 error distribution

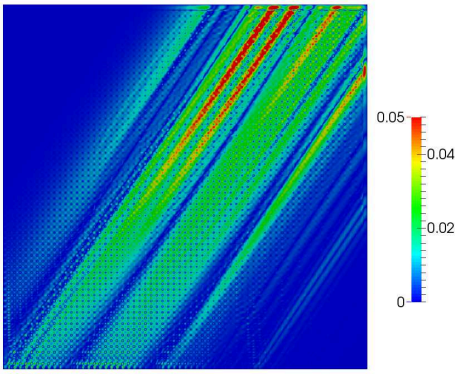


(c) Serendipity CG1-DG2 solution

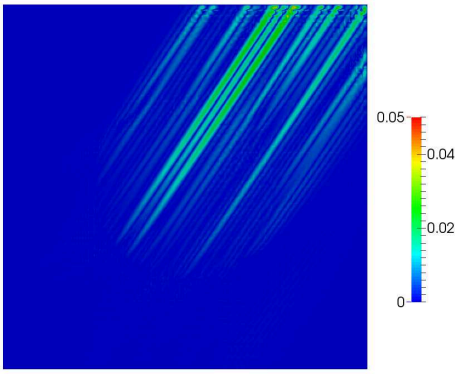


(d) Serendipity CG1-DG2 error distribution

Figure 6.26: Hump changing its height at $t = 0.5$ on quadrilateral meshes



(a) Q2-CG1-DG2 error distribution



(b) Serendipity CG1-DG2 error distribution

Figure 6.27: Hump changing its height at $t = 2.0$ on quadrilateral meshes

7

The Euler equations in 2D

In this chapter we give a brief introduction to the Euler equations which are used to model compressible gas flows. These equations are of hyperbolic type like the scalar advection equation that has already been presented in Section 3.4. Since the continuous Galerkin method is not stable in the hyperbolic case, different kinds of stabilization techniques have been developed or extended from the scalar case, e.g. streamline diffusion/Petrov Galerkin method [45, 52], artificial viscosity [79] or algebraic flux correction [64, 70]. Another approach similar to the scalar case is the use of the discontinuous Galerkin method [9, 23, 28, 38], which leads to a stable method.

The last chapters have shown, that the CG1-DG2 method can be used to discretize and solve scalar advection equations. The analytical and numerical results have been similar to those derived for the DG method. Therefore, we extend the CG1-DG2 method to solve the Euler equations and expect similar behavior of the solutions as those computed by the DG method.

In the following we will explain how the Euler equations are derived from the compressible Navier-Stokes equations and which solution features (e.g. shock waves) can arise. Since those features also occur in the context of scalar nonlinear conservation laws, we will explain some of those in more detail for the scalar case. Then we explain the discretization using the continuous Galerkin method. At last we introduce the CG1-DG2 method for the Euler equations which gives a discretized system similar to that for the DG method.

7.1. Modeling of a compressible gas flow

When it comes to modeling compressible flows, the compressible Navier-Stokes equations are usually the model of choice. They are based on the conservation of mass, momentum and energy where the conservation of momentum is based on Newton's second law and the conservation of energy on the first law of thermodynamics [86]. These conservation laws are given by

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) &= 0, \\
\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v} \otimes \mathbf{v} + \mathbf{T}) &= \mathbf{F_b}, \\
\frac{\partial (\rho E)}{\partial t} + \nabla \cdot (\rho E \mathbf{v} + \mathbf{v} \cdot \mathbf{T} + \mathbf{q}) &= \mathbf{v} \cdot \mathbf{F_b} + Q,
\end{aligned}$$

where ρ denotes the density and $\mathbf{v} = (v_x, v_y)^T$ the velocity. E is the total energy, $\mathbf{q}$ is the heat flux, Q is the external heat source and $\mathbf{F_b}$ is the sum of body forces.

The stress tensor $\mathbf{T}$ depends on the type of fluid, e.g., in the case of a Newtonian fluid the stress tensor is given by

$$\mathbf{T} = p\mathbf{I} - \mu\left(\nabla\mathbf{v} + (\nabla\mathbf{v})^T\right) + \frac{2}{3}\mu\mathbf{I}\nabla\cdot\mathbf{v},$$

where p is the pressure, μ is the dynamic viscosity and $\mathbf{I}$ is the identity tensor.

In the context of gas flows, viscous terms can often be ignored [68]. If additionally heat conduction, heat sources and body forces are neglected, we obtain a simplified model, the so-called Euler equations.

Definition 7.1: Euler equations

The Euler equations are defined by

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\mathbf{v}) = 0, \tag{7.1.1}$$

$$\frac{\partial(\rho\mathbf{v})}{\partial t} + \nabla\cdot(\rho\mathbf{v}\otimes\mathbf{v} + p\mathbf{I}) = 0, \tag{7.1.2}$$

$$\frac{\partial(\rho E)}{\partial t} + \nabla\cdot(\rho E\mathbf{v} + p\mathbf{v}) = 0. \tag{7.1.3}$$

Since there are more unknowns (ρ, $\mathbf{v}$, E, p) than equations, we have to close the system by defining a relation between those unknowns. This is done by the equation of state [68]

$$p = (\gamma - 1)\left(\rho E - \frac{\rho|\mathbf{v}|^2}{2}\right) \tag{7.1.4}$$

for a polytropic gas, where γ is the heat capacity ratio. For air this ratio is given by $\gamma = 1.4$ [87].

Additionally, we introduce the following quantities:

Definition 7.2: Enthalpy, speed of sound and Mach number

The total enthalpy is defined by

$$H = E + \frac{p}{\rho}. \tag{7.1.5}$$

The speed of sound c is given by

$$c = \sqrt{\frac{\gamma p}{\rho}} = \sqrt{\gamma R T}, \tag{7.1.6}$$

where R is the ideal gas constant and T the temperature.
The ratio of the speed of gas to the speed of sound is called the Mach number

$$M = \frac{|\mathbf{v}|}{c}. \tag{7.1.7}$$

The last quantity helps to classify compressible flow regimes as follows:

Definition 7.3: Flow regimes

The Mach number M characterizes the flow as

1. subsonic for $0 < M < 1$,
2. transonic for $M \approx 1$,
3. supersonic for $M > 1$,
4. hypersonic for $M > 5$,
5. high-hypersonic for $M > 10$.

Note that the gas becomes incompressible, when the Mach number tends to zero [87]. In the hypersonic and high-hypersonic case thermal effects become more important but cannot be modeled by the Euler equations [72].

We have now derived the Euler equations as model for compressible gas flows, if subsonic, transonic or supersonic flows are considered. Next, we will have a closer look at particular analytical solutions of these equations.

7.2. Solution of nonlinear conservation laws

Scalar advection equations are the simplest form of conservation laws. We considered an advection equation in our numerical example in Section 6.5, where we have transported initial data in a counterclockwise rotation about the center of the domain. The challenge of this test problem was that the numerical solution should coincide with the initial data after each full revolution. This conservation of either continuous or discontinuous data is a property of the exact solution. For nonlinear conservation laws discontinuities can arise not only from discontinuous initial data but even from smooth data. Therefore, we will at first consider scalar nonlinear conservation laws and explain two typical solution features, namely shock waves and rarefaction waves. These features can frequently be observed in the context of the Euler equations. Then we will explain contact discontinuities for the Euler equations.

7.2.1. Shock waves and rarefaction waves

We will follow [68] to introduce the concept of shock and rarefaction waves. Let us consider a scalar nonlinear conservation law of the form

$$u_t + f(u)_x = 0, \tag{7.2.1}$$

where f is a nonlinear function of u. For simplicity, we set $f(u) = \frac{1}{2}u^2$ and obtain the so-called *inviscid Burgers equation*, which can be written in quasi-linear form as

$$u_t + uu_x = 0. \tag{7.2.2}$$

For this equation, we will explain when a shock is formed, and why we have to extend the concept of classical solutions to weak solutions.

The solution u of (7.2.2) is constant along the *characteristic curve* $x(t)$, which satisfies $x'(t) = u(x(t),t)$. In the linear case $u_t + au_x = 0$, this is also true for the characteristic $\tilde{x}(t)$, where $\tilde{x}'(t) = a$. Hereby, the initial data $u_0(x)$ is transported over time in such a way that the solution profile does

not change its shape, only its position. Therefore, the exact solution of the linear problem is given by $u(x,t) = u_0(x-at)$.

Coming back to the nonlinear case, we have

$$\frac{\mathrm{d}}{\mathrm{d}t}u(x(t),t) = \frac{\partial}{\partial t}u(x(t),t) + \frac{\partial}{\partial x}u(x(t),t)x'(t) = 0, \tag{7.2.3}$$

which yields that u is constant along $x(t)$. Since $x'(t)$ is also constant, the characteristics are straight lines. If the characteristics do not cross, we can derive the exact solution as follows

$$x = x_0 + u_0(x_0)t, \quad u(x,t) = u_0(x_0). \tag{7.2.4}$$

However, if characteristics cross, the so calculated solutions are not unique. The first time, when the characteristics intersect, the function $u(x,t)$ has an infinite slope and a shock forms. This shock is given as a jump in the solution. Since classical solutions do not exist from this time instant, we introduce the concept of weak solutions (see, e.g., [68]).

Definition 7.4: Weak solutions

The function $u(x,t)$ is called a weak solution of the conservation law (7.2.1) if

$$\int_0^\infty \int_{-\infty}^\infty \varphi_t u + \varphi_x f(u)\,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{t} = -\int_{-\infty}^\infty \varphi(x,0)u_0(x)\,\mathrm{d}\mathbf{x} \tag{7.2.5}$$

for all test functions $\varphi(x,t) \in C_0^1(\mathbb{R}\times\mathbb{R}_0^+)$.

Since weak solutions may not be unique, we have to identify conditions to select the physically relevant solution, which is called *entropy solution*. This refers to solutions of gas dynamics, which satisfy the second law of thermodynamics stating that entropy is nondecreasing. Similarly, conditions for scalar conservation laws can be derived which are called entropy conditions by analogy with gas dynamics [68].

We have already seen, that the solution becomes discontinuous, when a shock forms. We will now investigate what kinds of weak solutions can occur, if our initial data is discontinuous. Therefore, we will consider the so-called Riemann problem for the inviscid Burgers equation in 1D.

Definition 7.5: Riemann problem

The Riemann problem consists of a conservation law endowed with initial data which is piecewise constant and has a single jump discontinuity.

The initial data is given by

$$u_0(x) = \begin{cases} u_l & \text{if } x < 0, \\ u_r & \text{otherwise.} \end{cases} \tag{7.2.6}$$

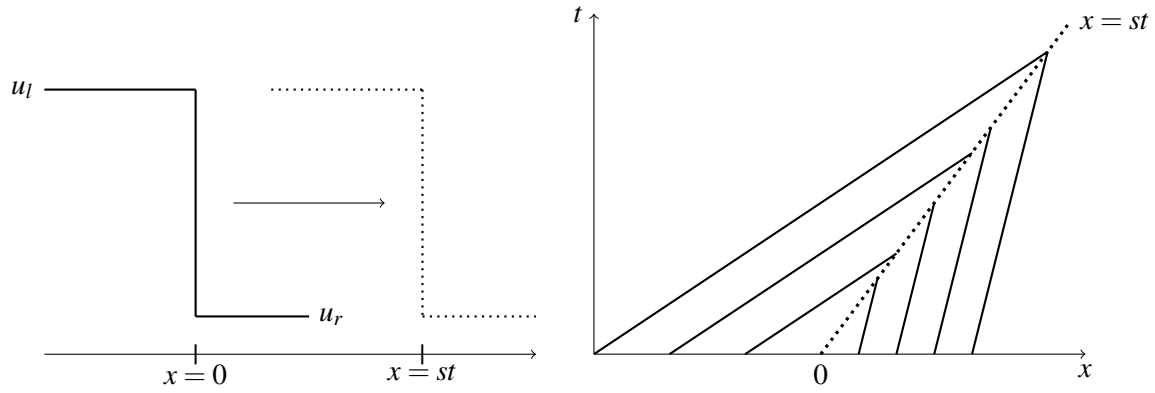
There are two different cases to be considered: $u_l > u_r$ and $u_l < u_r$.

Let us begin with $u_l > u_r$. This gives a unique weak solution

$$u(x,t) = \begin{cases} u_l & \text{if } x < st, \\ u_r & \text{otherwise,} \end{cases} \tag{7.2.7}$$

where s is the shock speed. The shock speed s must satisfy the so-called *Rankine-Hugoniot condition*

$$f(u_l) - f(u_r) = s(u_l - u_r). \tag{7.2.8}$$

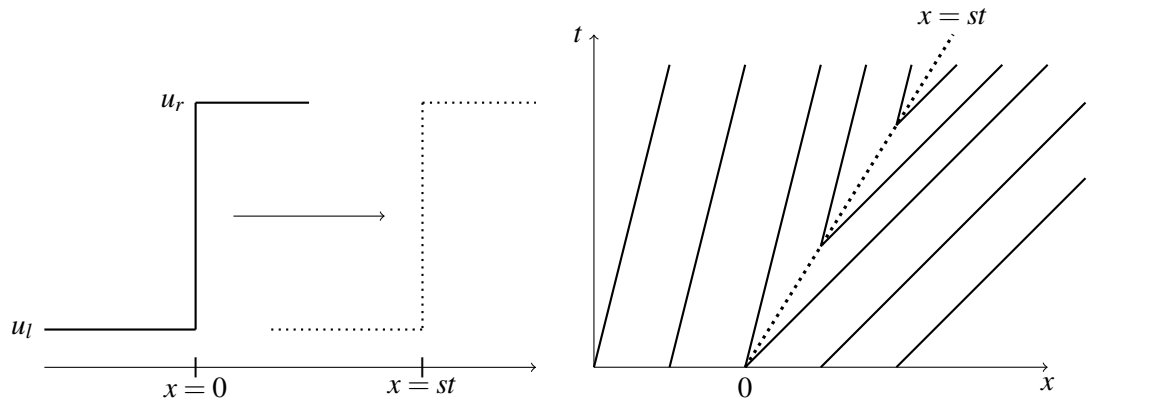


(a) Weak solution (b) Characteristics and shock

Figure 7.1: Shock wave

This leads to $s = \frac{u_l+u_r}{2}$ for the Burgers equation. In Fig. 7.1 this type of solution and corresponding characteristics are displayed. We see that the solution stays discontinuous and the characteristics go into the shock. Note that the weak solution (7.2.7) is unique [68].

The other case $u_l < u_r$ yields infinitely many weak solutions. One possibility would again be a shock which propagates with speed s. However, this solution is not stable to perturbations and is called a *entropy-violating shock*. In Fig. 7.2 we see that this entropy-violating solution stays discontinuous and the characteristics come out of the shock.



(a) Weak solution (b) Characteristics and shock

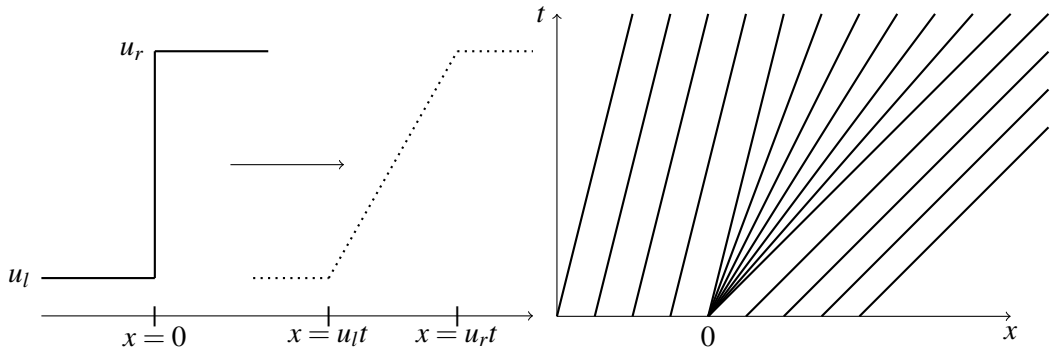
Figure 7.2: Entropy-violating shock

This behavior is physically incorrect and therefore, another solution, the rarefaction wave, is considered

$$u(x,t) = \begin{cases} u_l & \text{if } x < u_l t, \\ x/t & \text{if } u_l t \leq x \leq u_r t, \\ u_r & \text{if } x > u_r t, \end{cases} \tag{7.2.9}$$

which results in a smooth transition from u_l to u_r and gives the entropy solution. This is displayed in Fig. 7.3, where we see that the discontinuous solution becomes a continuous transition. There are now infinitely many characteristics starting at the point $x = 0$.

A weak solution is an entropy solution and therefore physically correct, if the following con-



(a) Weak solution (b) Characteristics

Figure 7.3: Rarefaction wave

dition is fulfilled

$$\frac{f(u)-f(u_l)}{u-u_l} \geq s \geq \frac{f(u)-f(u_r)}{u-u_r}, \quad \forall u = \theta u_l + (1-\theta)u_r, 0 < \theta < 1. \tag{7.2.10}$$

This condition is called *Oleinik's entropy condition*. In the case of convex functions, this simplifies to

$$f'(u_l) > s > f'(u_r). \tag{7.2.11}$$

For the Burgers equation, we have $f'(u) = u$ and condition (7.2.11) is violated for $u_l < u_r$.

We have now seen that we need to extend the concept of classical solutions to weak solutions, to solve nonlinear conservation laws. Smooth solutions can lead to so-called shocks, if characteristics cross. These shocks are weak solutions and satisfy the Rankine-Hugoniot condition. Furthermore, discontinuous initial data can result either in shock waves or rarefaction waves. If the weak solution is not unique, we can use the entropy condition to test if a weak solution is entropy-violating and therefore should not be considered as physically relevant.

This theory can also be extended to the Euler equations (see, e.g., [67, 68]). Therefore, we consider the Euler equations in 1D

$$U_t + F(U)_x = 0 \tag{7.2.12}$$

where

$$U = \begin{bmatrix} \rho \\ \rho v \\ \rho E \end{bmatrix}, \qquad F(U) = \begin{bmatrix} \rho v \\ \rho v^2 + p \\ \rho E v + p v \end{bmatrix}. \tag{7.2.13}$$

In analogy to the scalar case, this system can be written in quasi-linear form

$$U_t + A(U)U_x = 0, \tag{7.2.14}$$

where

$$U = \begin{bmatrix} \rho \\ \rho v \\ \rho E \end{bmatrix} \qquad A(U) = F'(U) = \begin{bmatrix} 0 & 1 & 0 \\ 0.5(\gamma-3)v^2 & (3-\gamma)v & (\gamma-1) \\ 0.5(\gamma-1)v^3 - vH & H-(\gamma-1)v^2 & \gamma v \end{bmatrix}. \tag{7.2.15}$$

The left state $\tilde{U}$ and the right state $\hat{U}$ with respect to a shock have to fulfill the Rankine-Hugoniot jump condition [68]

$$F(\tilde{U}) - F(\hat{U}) = s(\tilde{U} - \hat{U}), \tag{7.2.16}$$

where s is the shock speed and F the flux function. The shock speed also has to satisfy Lax's entropy condition [68]

$$\lambda_i(\tilde{U}) > s > \lambda_i(\hat{U}), \quad i = 1,2,3, \tag{7.2.17}$$

where λ_i is the i-th eigenvalue of $A(U)$.

In the case of rarefaction waves, the following condition holds

$$\lambda_i(\tilde{U}) < \lambda_i(\hat{U}), \tag{7.2.18}$$

which yields a smooth transition form state $\tilde{U}$ to $\hat{U}$.

7.2.2. Contact discontinuity

Another feature occurring in the context of conservation laws is a contact discontinuity. To explain it in more detail we will consider the Euler equations in 1D. The solution can be described by conservative variables ρ, ρv and ρE or by primitive variables ρ, v and p. Latter ones lead to a system in quasi-linear form

$$Q_t + A(Q)Q_x = 0, \tag{7.2.19}$$

where

$$Q = \begin{bmatrix} \rho \\ v \\ p \end{bmatrix}, \quad A(Q) = \begin{bmatrix} v & \rho & 0 \\ 0 & v & \frac{1}{\rho} \\ 0 & \gamma p & v \end{bmatrix}. \tag{7.2.20}$$

This system is *strictly hyperbolic*, i.e., the matrix $A = A(Q)$ has only distinct real eigenvalues $\lambda_1 = v - c$, $\lambda_2 = v$ and $\lambda_3 = v + c$ [67]. Therefore, we can decompose the matrix

$$A = R\Lambda R^{-1}, \tag{7.2.21}$$

where $R = [r_1, r_2, r_3]$ is the matrix of right eigenvectors

$$r_1 = \begin{bmatrix} -\frac{\rho}{c} \\ 1 \\ -\rho c \end{bmatrix}, \quad r_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad r_3 = \begin{bmatrix} \frac{\rho}{c} \\ 1 \\ \rho c \end{bmatrix}, \tag{7.2.22}$$

and $\Lambda = \mathrm{diag}\{\lambda_1, \lambda_2, \lambda_3\}$ the diagonal matrix of eigenvalues λ_i.

Defining

$$\delta W = R^{-1}\delta Q, \tag{7.2.23}$$

where δ denotes either ∂_t or ∂_x, (7.2.19) can be written as

$$W_t + \Lambda W_x = 0, \tag{7.2.24}$$

where $W = [w_1, w_2, w_3]$ is the vector of so-called characteristic variables. Recall that R and Λ depend on Q. If this system were linear, we would obtain three decoupled scalar advection equations. However, this system can be understood as three waves each traveling with wave speed λ_i. The change of the primitive variables δQ can be expressed as

$$\delta Q = \sum_{i=1}^{3} \delta w_i r_i. \tag{7.2.25}$$

If we consider the wave corresponding to λ_2, we see that a jump across this wave results in a change in density, whereas pressure and velocity remain constant. This behavior is called *contact discontinuity* and can only take place for *linearly degenerated* characteristic fields [68], i.e.,

$$\nabla \lambda_i(Q) \cdot r_i(Q) \equiv 0, \forall Q. \tag{7.2.26}$$

In the context of scalar conservation laws $u_t + a(u)u_x$ contact discontinuities are characterized by jumps in the solution from u_l to u_r where $a(u_l) = a(u_r)$ [65].

If we consider the Riemann problem for the Euler equations, we will obtain a solution which consists of a contact discontinuity and two nonlinear waves, i.e., shock and/or rarefaction waves [68].

We have now seen what features the solution of the Euler equations has. In the following, we will summarize some mathematical aspects of the Euler equations in 2D, that will turn out useful for the derivation of the discretized system.

7.3. Mathematical aspects

As we have already seen in the last chapter for the one dimensional case, the Euler equations (7.1.1) - (7.1.3) can be written in the conservative form

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = 0, \tag{7.3.1}$$

where

$$U = \begin{bmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{bmatrix}, \qquad \mathbf{F} = \mathbf{F}(U) = \begin{bmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + pI \\ \rho E \mathbf{v} + p\mathbf{v} \end{bmatrix},$$

or in the quasi-linear form

$$\frac{\partial U}{\partial t} + \mathbf{A} \cdot \nabla U = 0, \tag{7.3.2}$$

where $\mathbf{A} = \frac{\partial \mathbf{F}}{\partial U} = (\frac{\partial F^{(x)}}{\partial U}, \frac{\partial F^{(y)}}{\partial U})$ is the Jacobian tensor. The components of the flux vector $\mathbf{F} = (F^{(x)}, F^{(y)})$ are given by [35, 40]

$$F^{(x)} = \begin{bmatrix} \rho v_x \\ \rho v_x^2 + p \\ \rho v_x v_y \\ \rho E v_x + v_x p \end{bmatrix} \quad \text{and} \quad F^{(y)} = \begin{bmatrix} \rho v_y \\ \rho v_x v_y \\ \rho v_y^2 + p \\ \rho E v_y + v_y p \end{bmatrix}.$$

The matrices $A_1 := \frac{\partial F^{(x)}}{\partial U}$ and $A_2 := \frac{\partial F^{(y)}}{\partial U}$ can be found in Definition A.14.
A useful relation between $\mathbf{A}$ and $\mathbf{F}$ can be derived by considering the homogeneity property [40]

$$\mathbf{F}(\alpha U) = \alpha \mathbf{F}(U), \quad \forall \alpha \in \mathbb{R}. \tag{7.3.3}$$

Differentiating with respect to α and setting $\alpha = 1$, one obtains

$$\mathbf{F}(U) = \frac{\partial \mathbf{F}}{\partial U} U = \mathbf{A}U. \tag{7.3.4}$$

Furthermore, the system of the Euler equations is hyperbolic, since the matrix

$$\mathbf{A}(U,\mathbf{e}) = e_1 A_1 + e_2 A_2, \quad \mathbf{A} = (A_1, A_2), \mathbf{e} = (e_1, e_2), |e| = 1, \tag{7.3.5}$$

is diagonalizable with real eigenvalues (cf. Def. A.13). It admits the factorization

$$\mathbf{A}(U,\mathbf{e}) = \mathbf{R}(U,\mathbf{e})\Lambda(U,\mathbf{e})\mathbf{R}(U,\mathbf{e})^{-1}, \tag{7.3.6}$$

where $\Lambda(U,\mathbf{e}) = diag\{\mathbf{e}\cdot\mathbf{v} - c, \mathbf{e}\cdot\mathbf{v}, \mathbf{e}\cdot\mathbf{v} + c, \mathbf{e}\cdot\mathbf{v}\}$ is the diagonal matrix of eigenvalues and $\mathbf{R}(U,\mathbf{e})$ is the matrix of right eigenvectors. The analytical expression of these matrices can be found in Definition A.14. Note that we have strict hyperbolicity only for the 1D case.

7.4. The continuous Galerkin method for the Euler equations

In the following we will explain how the Euler equations are discretized in space using the continuous Galerkin method. Hereby, we follow the derivation of the high-order scheme presented in [35, 72] and [64]. A matter of particular importance is the treatment of the boundary conditions which we will also adopt for the CG1-DG2 method.

7.4.1. Group finite element formulation

For the discretization of the Euler equations in conservative form we multiply (7.3.1) by a suitable test function W and integrate by parts. This gives

$$\int_\Omega \left(W \frac{\partial U}{\partial t} - \nabla W \cdot \mathbf{F} \right) \mathrm{d}\mathbf{x} + \int_\Gamma W\, \mathbf{n} \cdot \mathbf{F}\, \mathrm{d}\mathbf{s} = 0, \quad \forall W, \tag{7.4.1}$$

where $\mathbf{n} = (n_x, n_y)$ is the unit outward normal. The numerical solution as well as the numerical flux function are given by the group finite element formulation [31]

$$U_h(\mathbf{x},t) = \sum_j U_j(t)\varphi_j(\mathbf{x}), \tag{7.4.2}$$

$$\mathbf{F}_h(\mathbf{x},t) = \sum_j \mathbf{F}_j(t)\varphi_j(\mathbf{x}), \tag{7.4.3}$$

where $\{\varphi_i\}$ is a set of basis functions. This formulation leads to the discretization

$$\sum_j \left(\int_\Omega \varphi_i \varphi_j \,\mathrm{d}\mathbf{x} \right) \frac{dU_j}{dt} = \sum_j \left(\int_\Omega \nabla\varphi_i \varphi_j \,\mathrm{d}\mathbf{x} \right) \cdot \mathbf{F}_j - \int_\Gamma \varphi_i \mathbf{n} \cdot \mathbf{F}\, \mathrm{d}\mathbf{s}, \quad \forall i. \tag{7.4.4}$$

By (7.3.4) we have $\mathbf{F}_j = \mathbf{A}_j U_j$ resulting in the semi-discrete problem

$$M_C \frac{dU}{dt} = KU + S(U), \tag{7.4.5}$$

where M_C denotes the consistent block mass matrix, K is the discrete Jacobian operator and $S(U)$ is the boundary vector. The matrix M_C consists of blocks of size 4×4 in 2D and is defined by

$$M_C = \{M_{ij}\} = \{m_{ij} I\}, \quad m_{ij} = \int_\Omega \varphi_i \varphi_j \,\mathrm{d}\mathbf{x}, \tag{7.4.6}$$

where I is the 4×4 identity matrix. The entries of the matrix $K = \{K_{ij}\}$ are given by

$$K_{ij} = \mathbf{c}_{ji} \cdot \mathbf{A}_j, \quad \mathbf{c}_{ij} = \int_\Omega \varphi_i \nabla \varphi_j \, d\mathbf{x}, \tag{7.4.7}$$

and the entries of the boundary vector $S(U)$ by

$$S_i = -\int_\Gamma \varphi_i \mathbf{n} \cdot \mathbf{F} \, d\mathbf{s}. \tag{7.4.8}$$

7.4.2. Boundary conditions

The discretization of (7.4.4) implies the use of weak boundary conditions. In comparison to the scalar case, where boundary conditions are only imposed on the inflow part of the boundary, we have to distinguish between the different components and flow regimes. At first we discuss how to approximate the boundary fluxes $\mathbf{n} \cdot \mathbf{F}$ when an internal and an external flow is given. In the scalar case the external flow corresponds to the prescribed inflow boundary condition. The second step is the definition of the external flow for the different flow regimes.

Approximated boundary flux

For the calculation of the boundary term $\int_\Gamma \varphi_i \mathbf{n} \cdot \mathbf{F} \, d\mathbf{s}$ we replace the boundary flux $\mathbf{n} \cdot \mathbf{F}$ by the solution of a Riemann problem $\mathbf{n} \cdot \hat{F}_h$ (see Def. 7.5) , which can be approximated by the flux formula of Roe

$$\mathbf{n} \cdot \hat{F}_h = F_n(U, U_\infty) = \frac{1}{2} \mathbf{n} \cdot (\mathbf{F}(U) + \mathbf{F}(U_\infty)) - \frac{1}{2} \left| \mathbf{n} \cdot \mathbf{A}(U, U_\infty) \right| (U_\infty - U), \tag{7.4.9}$$

where U is the internal state, U_∞ is the external state and $\mathbf{A}(U, U_\infty)$ is the Roe matrix for both states. The internal state is the numerical solution to the Euler equation. The derivation of the external state, also referred to as free stream values, is explained later. Note that this approach is similar to the scalar case where the boundary flux $\beta_n u$ is approximated by the upwind flux (cf. (4.2.15)) so that on the inlet part the value of u is prescribed and on the outlet part the unknown solution u is used (cf. (3.4.8)). The prescribed solution would correspond to the external state, the other one to the internal state.

The Roe matrix $\mathbf{A}(U_i, U_j)$ can be obtained by replacing ρ, $\mathbf{v}$ and the stagnation enthalpy H by the Roe mean values

$$\rho_{ij} = \sqrt{\rho_i \rho_j}, \tag{7.4.10}$$

$$\mathbf{v}_{ij} = \frac{\sqrt{\rho_i} \mathbf{v}_i + \sqrt{\rho_j} \mathbf{v}_j}{\sqrt{\rho_i} + \sqrt{\rho_j}}, \tag{7.4.11}$$

$$H_{ij} = \frac{\sqrt{\rho_i} H_i + \sqrt{\rho_j} H_j}{\sqrt{\rho_i} + \sqrt{\rho_j}}. \tag{7.4.12}$$

Similarly to (7.3.5) we have

$$\mathbf{A}(U, U_\infty, \mathbf{n}) := \mathbf{n} \cdot \mathbf{A}(U, U_\infty) = n_x A_1(U, U_\infty) + n_y A_2(U, U_\infty), \tag{7.4.13}$$

where $\mathbf{A}(U, U_\infty) = (A_1(U, U_\infty), A_2(U, U_\infty))$. The factorization is similar to (7.3.6) so that the absolute value is given by

$$\left| \mathbf{A}(U, U_\infty, \mathbf{n}) \right| = \mathbf{R}(U, U_\infty, \mathbf{n}) \left| \Lambda(U, U_\infty, \mathbf{n}) \right| \mathbf{R}(U, U_\infty, \mathbf{n})^{-1}, \tag{7.4.14}$$

where $|\Lambda(U, U_\infty, \mathbf{n})| = diag\{|\lambda_1|, ..., |\lambda_4|\}$ with $\lambda_1, ..., \lambda_4$ being the eigenvalues of $\mathbf{A}(U, U_\infty, \mathbf{n})$.

Boundary types

For the derivation of the free stream values U_∞ we need to distinguish between the different types of boundaries which can occur in the context of the Euler equations. These are:

- supersonic inlet,
- supersonic outlet,
- subsonic inlet,
- subsonic outlet,
- solid surface boundary.

The first four boundaries (supersonic and subsonic boundaries) can be summarized as open or far-field boundaries. Depending on the type, *physical boundary conditions (PBC)* need to be prescribed. These PBC are used for the derivation of the free stream values.

Riemann invariants

To determine the boundary type, we transform the Euler equations from conservative variables to local characteristic variables associated with the unit outward normal vector $\mathbf{n}$ and the unit tangential vector $\boldsymbol{\tau}$. The derived system consists of four decoupled equations [64]

$$\partial_t W_k + \lambda_k \frac{\partial W_k}{\partial n} = 0, \quad k = 1,..4, \tag{7.4.15}$$

where W_k are the so-called *Riemann invariants* and λ_k the eigenvalues of the directional Jacobian $\mathbf{A}(U,\mathbf{n})$. Each Riemann invariant W_k propagates along the corresponding characteristic with constant speed λ_k and is conserved along this characteristic if no discontinuities occur [72]. The sign of the eigenvalues indicates the direction of the propagation. If the sign is negative, boundary conditions need to be prescribed for the corresponding Riemann invariant. The Riemann invariants and eigenvalues are explicitly given by

$$W_1 = v_n - \frac{2c}{\gamma - 1}, \quad W_2 = s, \quad W_3 = v_\tau, \quad W_4 = v_n + \frac{2c}{\gamma - 1}, \tag{7.4.16}$$

$$\lambda_1 = v_n - c, \quad \lambda_2 = \lambda_3 = v_n, \quad \lambda_4 = v_n + c, \tag{7.4.17}$$

where $v_n = \mathbf{v} \cdot \mathbf{n}$ is the normal velocity, $v_\tau = \mathbf{v} \cdot \boldsymbol{\tau}$ is the tangential velocity and $s = c_v \log\left(\frac{p}{\rho^\gamma}\right)$ is the entropy with the constant-volume heat capacity c_v. Following [35] we replace $W_2 = s$ by

$$W_2 = \frac{p}{\rho^\gamma}. \tag{7.4.18}$$

If we consider the *local Mach number* $M = \frac{|v_n|}{c}$ and the normal velocity v_n, we can identify the boundary types as follows [64]

- Supersonic inlet: $v_n < 0, M > 1$. All eigenvalues are negative.
- Supersonic outlet: $v_n > 0, M > 1$. All eigenvalues are positive.
- Subsonic inlet: $v_n < 0, M < 1$. Only λ_4 is nonnegative.
- Subsonic outlet: $v_n > 0, M < 1$. Only λ_1 is negative.
- Solid surface boundary: $v_n = M = 0$. Only λ_1 is negative.

If the Riemann invariants are given, the primitive $(\rho, p, \mathbf{v})$ and conservative variables $(\rho, \rho\mathbf{v}, \rho E)$ can be calculated by

$$\rho = \left(\frac{c^2}{\gamma W_2}\right)^{\frac{1}{\gamma-1}}, \tag{7.4.19}$$

$$p = \frac{c^2\rho}{\gamma}, \tag{7.4.20}$$

$$\mathbf{v} = \frac{W_1+W_4}{2}\mathbf{n} + W_3\boldsymbol{\tau}, \tag{7.4.21}$$

$$\rho\mathbf{v} = \rho\left(\frac{W_1+W_4}{2}\mathbf{n} + W_3\boldsymbol{\tau}\right), \tag{7.4.22}$$

$$\rho E = \frac{p}{\gamma-1} + \frac{1}{2}\rho|\mathbf{v}|^2, \tag{7.4.23}$$

where $c = \frac{\gamma-1}{4}(W_4 - W_1)$.

Calculation of the external state U_∞

To obtain the external state in (7.4.9) we make the following steps [64]:

- Compute the Riemann invariants $W(U)$ corresponding to the given numerical solution U.
- Set
$$W_\infty^k := \begin{cases} W_k(U) & \text{if } \lambda_k \geq 0, \\ W_{PBC}^k & \text{if } \lambda_k < 0, \end{cases}$$
where W_{PBC}^k are the prescribed physical boundary conditions.
- Calculate the free stream values U_∞ by using (7.4.19)-(7.4.23) for $W_\infty = (W_\infty^1, ..., W_\infty^4)$.

7.4.3. Prescribed open boundary conditions

Supersonic inlet

At a supersonic inlet all eigenvalues are negative and therefore boundary conditions need to be prescribed for all Riemann invariants. This means that $W_\infty = W_{PBC}$. To simplify the calculation of the external state we directly prescribe the free stream values U_∞ so that no transformation of the Riemann invariants to conservative variables is required.

Supersonic outlet

At a supersonic outlet all eigenvalues are positive and therefore no boundary conditions need to be prescribed, i.e., $W_\infty = W(U)$ and $U_\infty = U$. Therefore no transformation to and from the Riemann invariants is necessary and the flux formula (7.4.9) simplifies to

$$F_n(U, U_\infty) = F_n(U, U) = \mathbf{n} \cdot \mathbf{F}(U). \tag{7.4.24}$$

Subsonic inlet

At a subsonic inlet the fourth eigenvalue is nonnegative. Therefore we obtain

$$W_\infty = (W_{PBC}^1, W_{PBC}^2, W_{PBC}^3, W_4(U)). \tag{7.4.25}$$

It is common practice to prescribe the density ρ_{in}, pressure p_{in} and the tangential velocity v_τ^{in} [35]. The boundary conditions for the Riemann invariants can be calculated by

$$W_{PBC}^1 = W_4(U) - \frac{4c}{\gamma - 1}, \tag{7.4.26}$$

$$W_{PBC}^2 = \frac{p_{in}}{\rho_{in}^\gamma}, \tag{7.4.27}$$

$$W_{PBC}^3 = v_\tau^{in}, \tag{7.4.28}$$

where c is given by

$$c = \sqrt{\frac{\gamma p_{in}}{\rho_{in}}}. \tag{7.4.29}$$

This gives the external state $U_\infty = (\rho_\infty, (\rho\mathbf{v})_\infty, (\rho E)_\infty)$ by

$$\rho_\infty = \rho_{in}, \tag{7.4.30}$$

$$\mathbf{v}_\infty = \left(\frac{W_{PBC}^1 + W_4(U)}{2}\mathbf{n} + v_\tau^{in}\boldsymbol{\tau} \right), \tag{7.4.31}$$

$$(\rho\mathbf{v})_\infty = \rho_\infty \mathbf{v}_\infty, \tag{7.4.32}$$

$$(\rho E)_\infty = \frac{p_{in}}{\gamma - 1} + \frac{1}{2}\rho_{in}|\mathbf{v}_\infty|^2. \tag{7.4.33}$$

Subsonic outlet

At a subsonic outlet only the first eigenvalue is negative. This gives

$$W_\infty = (W_{PBC}^1, W_2(U), W_3(U), W_4(U)). \tag{7.4.34}$$

We prescribe the exit pressure p_{out} [35] and obtain the boundary condition of the first Riemann invariant

$$W_{PBC}^1 = W_4(U) - \frac{4}{\gamma - 1}\sqrt{\frac{\gamma p_{out}}{\rho}\left(\frac{p}{p_{out}}\right)^{\frac{1}{\gamma}}}, \tag{7.4.35}$$

where ρ and p are the values of the interior state.

7.4.4. Solid surface boundary

Since there is no convective flux across a solid surface, the normal velocity vanishes

$$\mathbf{v} \cdot \mathbf{n} = 0. \tag{7.4.36}$$

This is called a free-slip or no-penetration condition [64]. For the calculation of the boundary flux by Roe's flux formula (7.4.9) we derive the external state by using the mirror/reflection condition

$$\mathbf{n} \cdot (\mathbf{v}_\infty + \mathbf{v}) = 0 \tag{7.4.37}$$

which gives

$$U_\infty = \begin{bmatrix} \rho \\ \rho\mathbf{v}_\infty \\ \rho E \end{bmatrix}, \quad \mathbf{v}_\infty = \mathbf{v} - 2\mathbf{n}(\mathbf{v} \cdot \mathbf{n}). \tag{7.4.38}$$

Another way to enforce zero boundary flux is given by [27]

$$\mathbf{n}\cdot\hat{F}_h = \begin{bmatrix} 0 \\ n_x p \\ n_y p \\ 0 \end{bmatrix}. \tag{7.4.39}$$

On curved boundaries the definition of the normal vector is very important. If the boundary of the mesh is linearly approximated and therefore a poor approximation of a curved boundary, we would get numerically a constant normal vector along these sides which may cause large errors. For this reason whenever it is possible we will use the physical normal vector. In [55] it was shown that in the case of mirror conditions it is sufficient to use the physical normal vector in the calculation of the free stream values whereas for the solution of the Riemann problem $F_n(U,U_\infty)$ the standard numerical normal vector can be used. A way to approximate the physical normal vector, if there is no analytical description of the boundary, is described in [55].

For the discretization in time we will use the same procedure as for the CG1-DG2 method. Therefore, we will first introduce this method and then explain time discretization.

7.5. The CG1-DG2 method for the Euler equations

We will now derive the weak formulation for the CG1-DG2 method which is similar to the DG weak formulation.

7.5.1. Variational formulation

Let $\mathcal{T}_h$ be a conforming triangulation of $\Omega \subset \mathbb{R}^2$. We consider equation (7.3.1) on an element $K \in \mathcal{T}_h$. Multiplying (7.3.1) by a suitable vector-valued test function W and integrating by parts gives

$$\int_K \left(W\frac{\partial U}{\partial t} - \nabla W \cdot \mathbf{F} \right) \mathrm{d}\mathbf{x} + \int_{S_K} W^+ \mathbf{n}_{S_K} \cdot \mathbf{F}\,\mathrm{d}\mathbf{s} = 0. \tag{7.5.1}$$

If we sum over all elements $K \in \mathcal{T}_h$, we obtain

$$\int_{\mathcal{T}_h} \left(W\frac{\partial U}{\partial t} - \nabla W \cdot \mathbf{F} \right) \mathrm{d}\mathbf{x} + \sum_{K\in\mathcal{T}_h} \int_{S_K\setminus\Gamma} W^+ \mathbf{n}_{S_K} \cdot \mathbf{F}\,\mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^\partial} W\,\mathbf{n}\cdot\mathbf{F}\,\mathrm{d}\mathbf{s} = 0. \tag{7.5.2}$$

Similarly to the linear advection case in section 4.2, $\mathbf{F}$ may be discontinuous across the inner element edges. Therefore, we introduce the numerical flux $\mathbf{H}(U^+,U^-,\mathbf{n}_S)$, which satisfies

$$\sum_{K\in\mathcal{T}_h} \int_{S_K\setminus\Gamma} W^+ \mathbf{n}_{S_K} \cdot \mathbf{F}\,\mathrm{d}\mathbf{s} \approx \sum_{K\in\mathcal{T}_h} \int_{S_K\setminus\Gamma} \mathbf{H}(U^+,U^-,\mathbf{n}_{S_K}) W^+\,\mathrm{d}\mathbf{s}. \tag{7.5.3}$$

Possible definitions of this flux will be discussed later. We assume that the numerical flux $\mathbf{H}(U^+,U^-,\mathbf{n})$ is consistent, i.e.,

$$\mathbf{H}(U,U,\mathbf{n}) = \mathbf{n}\cdot\mathbf{F}, \tag{7.5.4}$$

and conservative, i.e.,

$$\mathbf{H}(U^+,U^-,\mathbf{n}) = -\mathbf{H}(U^-,U^+,-\mathbf{n}). \tag{7.5.5}$$

Furthermore, following [30] we assume that $\mathbf{H}(U^+, U^-, \mathbf{n})$ is locally Lipschitz-continuous.

Taking into account that the numerical flux is conservative, we derive

$$\int_{\mathcal{T}_h} \left(W \frac{\partial U}{\partial t} - \nabla W \cdot \mathbf{F} \right) \mathrm{d}\mathbf{x} + \int_{\mathcal{S}_h^{\mathrm{int}}} [W]\, \mathbf{H}(U^+, U^-, \mathbf{n}_S)\, \mathrm{d}\mathbf{s} + \int_{\mathcal{S}_h^{\partial}} W\, \mathbf{n} \cdot \mathbf{F}\, \mathrm{d}\mathbf{s} = 0, \quad \forall W, \tag{7.5.6}$$

where $\mathbf{n}_S$ is a fixed normal vector to side $S \in \mathcal{S}_h^{\mathrm{int}}$.

Like in the scalar case there are different possibilities for the definition of the numerical flux. Here we will present the Lax-Friedrichs and the Vijayasundaram flux, which are both consistent and conservative [36]. Let $S \in \mathcal{S}_h$. The Lax-Friedrichs flux is defined by

$$\mathbf{H}(U^+, U^-, \mathbf{n}_S) = \frac{1}{2}(\mathbf{F}(U^+) \cdot \mathbf{n}_S + \mathbf{F}(U^-) \cdot \mathbf{n}_S + \alpha[U]), \tag{7.5.7}$$

where α is the largest absolute eigenvalue of $\mathbf{A}(U^+, \mathbf{n}_S)$ and $\mathbf{A}(U^-, \mathbf{n}_S)$ given by

$$\alpha = \max\{\left|\mathbf{v}^+ \cdot \mathbf{n}_S\right| + c^+, \left|\mathbf{v}^- \cdot \mathbf{n}_S\right| + c^-\}, \tag{7.5.8}$$

where c is the speed of sound.

For the definition of the Vijayasundaram flux we introduce

$$\mathbf{A}^{\pm}(U, \mathbf{n}_S) = \mathbf{R}(U, \mathbf{n}_S) \Lambda^{\pm}(U, \mathbf{n}_S) \mathbf{R}(U, \mathbf{n}_S)^{-1}, \tag{7.5.9}$$

where $\Lambda^- = \mathrm{diag}\{\min(\lambda_i, 0)\}$, $\Lambda^+ = \mathrm{diag}\{\max(\lambda_i, 0)\}$ for eigenvalues λ_i of $\mathbf{A}(U, \mathbf{n}_S)$ and $\mathbf{R}(U, \mathbf{e})$ is the matrix of right eigenvectors of $\mathbf{A}(U, \mathbf{n}_S)$ (cf. Def. 7.3.6).

This yields the definition of the Vijayasundaram flux

$$\mathbf{H}(U^+, U^-, \mathbf{n}_S) = \mathbf{A}^+(\{U\}, \mathbf{n}_S) U^+ + \mathbf{A}^-(\{U\}, \mathbf{n}_S) U^-. \tag{7.5.10}$$

Note that Roe's flux formula (7.4.9) can also be used as numerical flux $\mathbf{H}(U^+, U^-, \mathbf{n}_S) = F_n(U^+, U^-)$.

7.5.2. The discretized system

Let us assume that we have a consistent and conservative flux,e.g., the Lax-Friedrichs flux, and the weak formulation is given by

$$\int_{\mathcal{T}_h} W \frac{\partial U}{\partial t}\, \mathrm{d}\mathbf{x} = \int_{\mathcal{T}_h} \nabla W \cdot \mathbf{F}\, \mathrm{d}\mathbf{x} - \int_{\mathcal{S}_h^{\mathrm{int}}} [W]\, \mathbf{H}(U^+, U^-, \mathbf{n}_S)\, \mathrm{d}\mathbf{s} - \int_{\mathcal{S}_h^{\partial}} W\, \mathbf{n} \cdot \mathbf{F}\, \mathrm{d}\mathbf{s}. \tag{7.5.11}$$

Let $\{\varphi_i\}$ be the basis of the space $[V_h^{1,2}]^4$.

Similarly to (7.4.5) we obtain the semi-discrete system

$$M_C \frac{dU}{dt} = KU + S(U) + H(U), \tag{7.5.12}$$

where M_C, K and $S(U)$ are defined as in the continuous case (7.4.5)-(7.4.8). $H(U)$ is the vector corresponding to the numerical flux which consists of entries

$$H_i = -\sum_{K \in \mathcal{T}_h} \int_{S_K \setminus \Gamma} \mathbf{H}(U^+, U^-, \mathbf{n}_{S_K}) {\varphi_i}^+\, \mathrm{d}\mathbf{s}. \tag{7.5.13}$$

Using the backward Euler scheme for the integration in time yields a nonlinear system

$$M_C \frac{U^{n+1}-U^n}{\Delta t} = K^{n+1}U^{n+1} + S(U^{n+1}) + H(U^{n+1}), \tag{7.5.14}$$

where the superscript n refers to the time level n. The terms on the right-hand side are now linearized [27, 35] using the Taylor series expansion

$$K^{n+1}U^{n+1} \approx K^n U^n + \left(\frac{\partial K}{\partial U}\right)^n (U^{n+1}-U^n), \tag{7.5.15}$$

$$S(U^{n+1}) \approx S(U^n) + \left(\frac{\partial S}{\partial U}\right)^n (U^{n+1}-U^n), \tag{7.5.16}$$

$$H(U^{n+1}) \approx H(U^n) + \left(\frac{\partial H}{\partial U}\right)^n (U^{n+1}-U^n). \tag{7.5.17}$$

This gives

$$\left[\frac{1}{\Delta t}M_C - \left(\frac{\partial K}{\partial U} + \frac{\partial S}{\partial U} + \frac{\partial H}{\partial U}\right)^n\right](U^{n+1}-U^n) = K^n U^n + S(U^n) + H(U^n). \tag{7.5.18}$$

We neglect the nonlinearity of K and approximate $\left(\frac{\partial K}{\partial U}\right)^n$ by K^n. This leads to

$$K^{n+1}U^{n+1} \approx K^n U^{n+1}. \tag{7.5.19}$$

The definition of the approximated boundary flux Jacobian $\left(\frac{\partial S}{\partial U}\right)^n$ can be found in [35].

For the derivation of the numerical flux Jacobian we take (7.3.4) into account and write the Lax-Friedrichs numerical flux in the following form

$$\mathbf{H}(U^+,U^-,\mathbf{n}_S) = \frac{1}{2}(\mathbf{A}^+U^+\cdot\mathbf{n}_S + \mathbf{A}^-U^-\cdot\mathbf{n}_S + \alpha(U^+-U^-)). \tag{7.5.20}$$

The Jacobian of the numerical flux corresponding to the state vector U_k at node k is given by

$$\frac{\partial \mathbf{H}(U^+,U^-,\mathbf{n}_S)}{\partial U_K^+} = \frac{1}{2}(\mathbf{A}^+\cdot\mathbf{n}_S + \alpha I)), \tag{7.5.21}$$

$$\frac{\partial \mathbf{H}(U^+,U^-,\mathbf{n}_S)}{\partial U_K^-} = \frac{1}{2}(\mathbf{A}^-\cdot\mathbf{n}_S - \alpha I)), \tag{7.5.22}$$

where I is the 4×4 identity matrix.

The Jacobian of the Vijayasundaram flux can be found in [36].

In the case of the stationary Euler equations

$$KU + S(U) + H(U) = 0 \tag{7.5.23}$$

we use a (pseudo-)time stepping scheme, i.e., we apply (7.5.18). To "boost" the convergence we start with a small time step size and increase it when the residual falls below a certain error bound.

Note that the discretized DG system can be derived in a similar way.

8

Numerical results for the Euler equations

In this chapter, we will focus on the numerical solution of the Euler equations. This can be very challenging also for stable methods since shock waves, contact discontinuities and rarefaction waves may occur. The first two phenomena produce discontinuous solutions which can cause oscillations if no further stabilization is applied. For example, in the scalar case those oscillations can be observed in the solid body rotation problem in section 6.5 where under- and overshoots at the boundary of the cylinder occurred also for the stable CG1-DG2 and DG2 method.

In the following we will compare the numerical results derived by the CG2, DG2 and CG1-DG2 method. We will see that the CG1-DG2 method produce similar results as those obtained by the DG2 method.

8.1. Nozzle flow problem

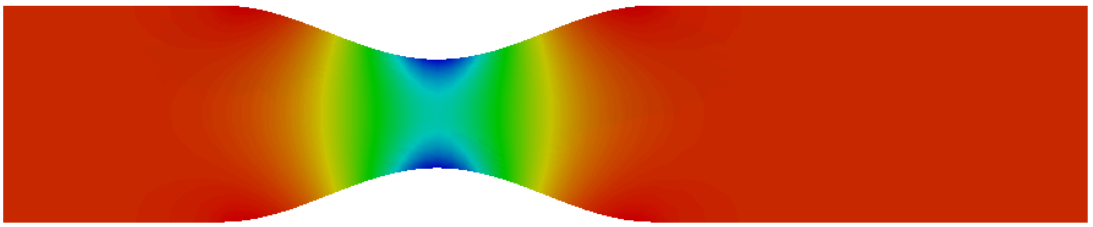
At first we consider a stationary problem with a subsonic flow through a converging-diverging nozzle [35, 38]. The upper and lower walls are given by

$$w^{\pm}(x)\begin{cases} \pm 1 & \text{if} \quad -2 \leq x \leq 0, \\ \pm 0.25\,(\cos(0.5\pi x)+3) & \text{if} \quad 0 < x \leq 4, \\ \pm 1 & \text{if} \quad 4 < x \leq 8. \end{cases} \tag{8.1.1}$$

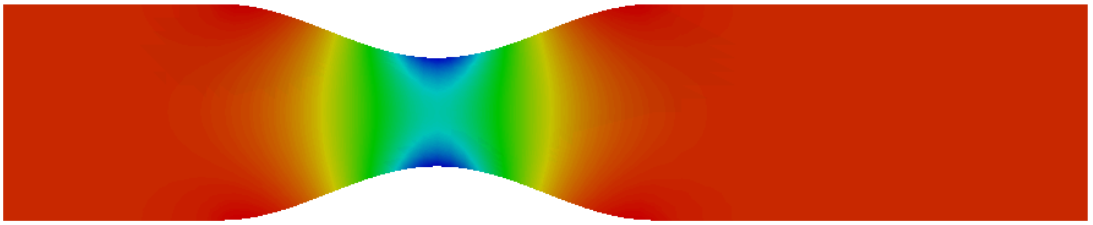
At the subsonic boundary we set $\rho = 1, \mathbf{v} = (0.2, 0)^T, p = \frac{1}{\gamma}$. These are also our initial condition for the stationary solver.

In Figures 8.1 - 8.3 the DG2 and CG1-DG2 solutions for density, pressure and Mach number are given in the case of triangular meshes. It can be seen that both methods produce similar results. The results for the serendipity CG1-DG2 method look similar and therefore are not shown here. We remark that the stationary solver did not converge for the CG2 and the Q2-CG1-DG2 method.

The densities in Fig. 8.1 vary between 0.94 and 1.0 and the pressures in Fig. 8.2 between 0.66 and 0.72. Both quantities reach their minimum at the boundary of the throat of the nozzle.

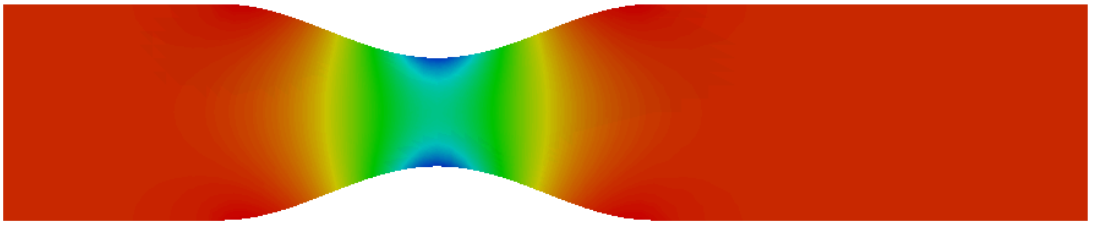


(a) CG1-DG2 solution

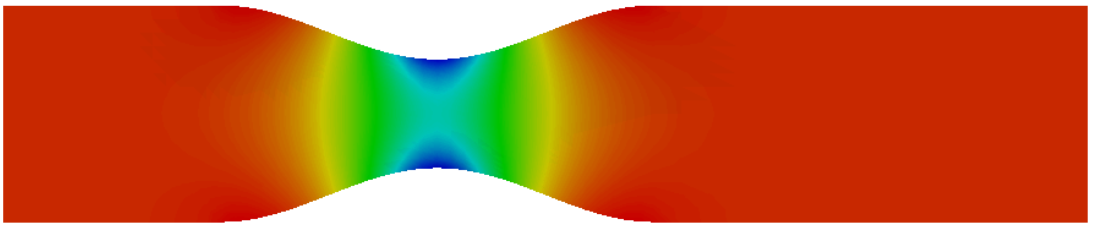


(b) DG2 solution

Figure 8.1: Converging-diverging nozzle on triangular meshes: density (blue = 0.94, red = 1.0)



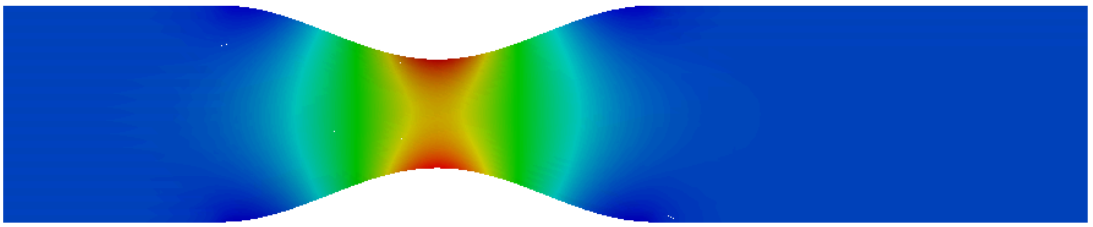
(a) CG1-DG2 solution



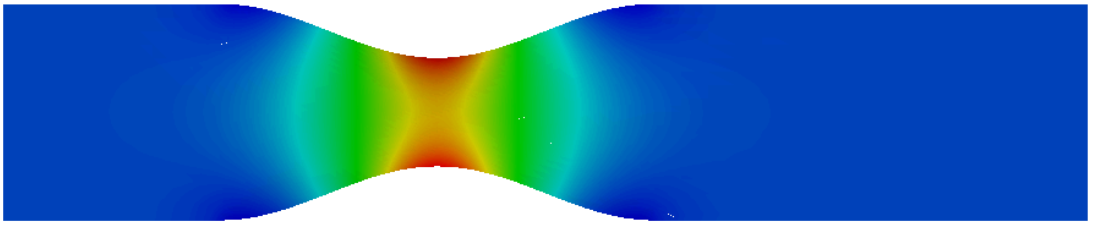
(b) DG2 solution

Figure 8.2: Converging-diverging nozzle on triangular meshes: pressure (blue = 0.66, red = 0.72)

The Mach number, shown in Fig. 8.3, varies between 0.14 and 0.39 which means the flow is subsonic in the whole domain. In contrast to density and pressure the maximum is reached at the most narrow part of the nozzle.



(a) CG1-DG2 solution



(b) DG2 solution

Figure 8.3: Converging-diverging nozzle on triangular meshes:
Mach number (blue = 0.14, red = 0.39)

8.2. Radially symmetric problem

We now consider the time-dependent Euler equations with smooth initial conditions

$$\rho = \begin{cases} \frac{1+\cos(\pi r(x,y))}{4} + 1 & \text{if } r(x,y) < 1, \\ 1 & \text{otherwise.} \end{cases} \tag{8.2.1}$$

$$\mathbf{v} = (0,0), \tag{8.2.2}$$

$$p = \frac{1}{\rho}, \tag{8.2.3}$$

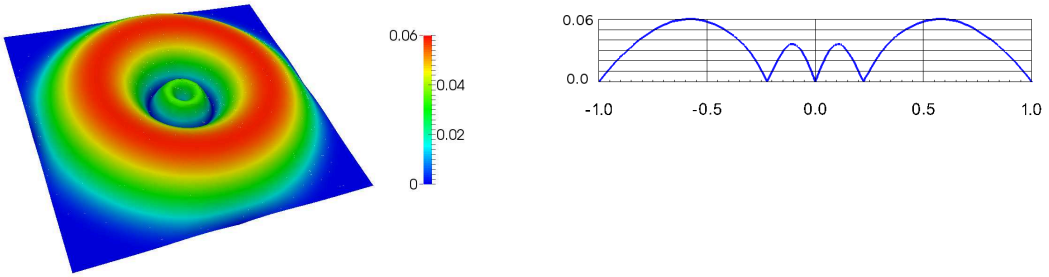
where

$$r(x,y) = 2\sqrt{(x^2+y^2)}.$$

We assume that all boundaries are solid walls. At $T = 0.5$ we stop the computation. All methods produce similar results so that we will only present those computed by the CG1-DG2 method on triangular meshes.

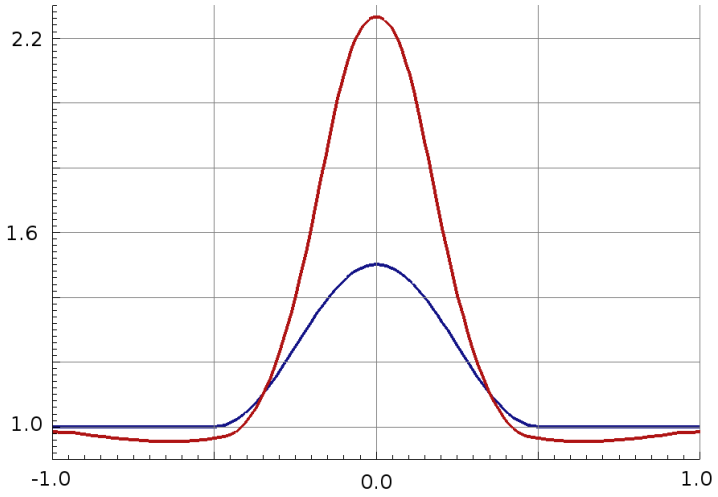
In Fig. 8.4 the Mach number is displayed. It can be seen that the Mach number is 0 at the center, on the circumference of a circle with radius ≈ 0.22 and outside the circle with radius 1.

In Fig. 8.5 it is shown how the density and the pressure change over time. Both quantities increase inside a circle of radius $\approx 0.35 - 0.4$ whereas they decrease elsewhere.

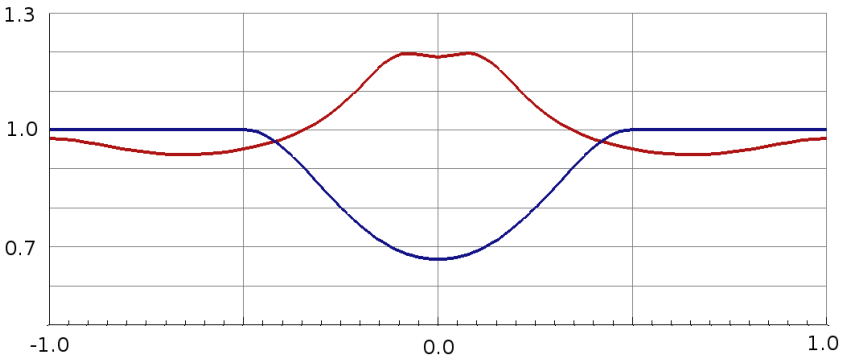


(a) Mach number 3D (b) Mach number at $x = 0$

Figure 8.4: Radially symmetric problem: Mach number at $T = 0.5$



(a) Density



(b) Pressure

Figure 8.5: Radially symmetric problem: temporal change of the solution at $x = 0$ (blue: initial data, red: numerical solution at $T = 0.5$)

8.3. Shock tube problem

We now consider a Riemann problem for the time-dependent Euler equations. This problem is adopted from [80] where the domain $\Omega = (0,1)$ is initially separated by a membrane into two sections. We extend it to two dimensions and define the initial data

$$\rho_1 = 1.0, \quad \rho_2 = 0.125, \tag{8.3.1}$$

$$\mathbf{v}_1 = (0,0)^T, \quad \mathbf{v}_2 = (0,0)^T, \tag{8.3.2}$$

$$p_1 = 1.0, \quad p_2 = 0.1. \tag{8.3.3}$$

At the beginning of the computation ($t = 0$) the membrane is removed. This results in a shock wave which propagates into the direction of lower pressure and is followed by a contact discontinuity. The shock wave manifests itself as discontinuity in all primitive variables, i.e., density, velocity and pressure. Only the density is discontinuous across the contact discontinuity whereas pressure and velocity are constant. Furthermore, there is a rarefaction wave propagating in the opposite direction which manifests itself in a smooth transition to the initial values.

At the beginning of the computation the initial data needs to be projected into the finite element space. Since the density and the pressure are discontinuous, the L^2 projection of those would cause oscillations. Therefore, we project the initial data into the continuous linear finite element space using a constrained L^2 projection (see section 9.5). This leads to an approximated initial data which is free of oscillations. The discontinuity of the pressure and the density is resolved as a steep gradient. If this data is projected into the CG2 space and the CG1-DG2 space, respectively, the projection is free of oscillations, see Fig. 8.6.

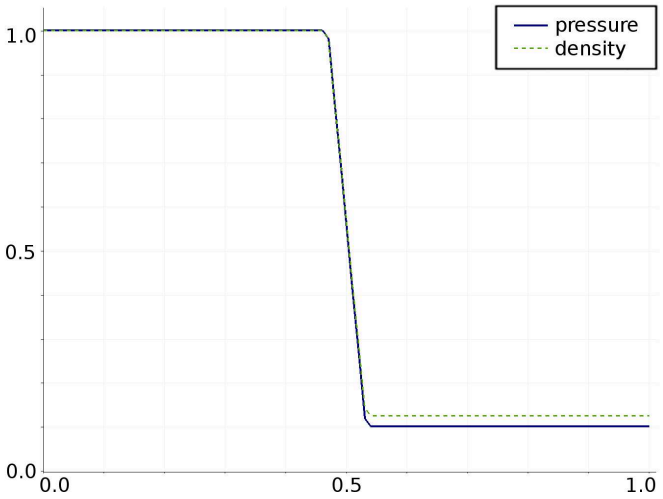
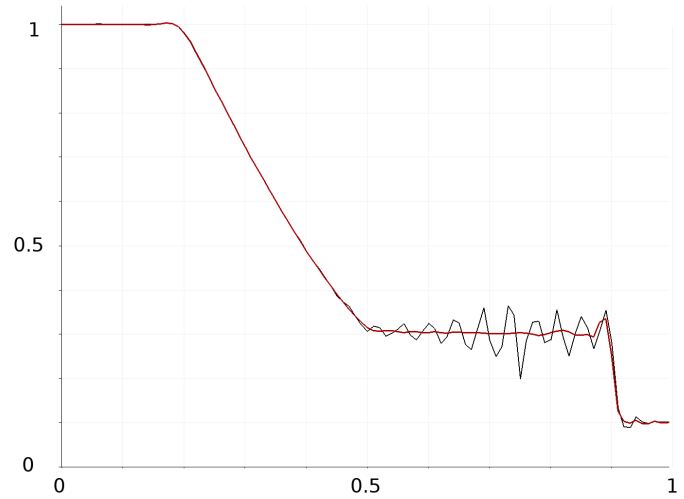


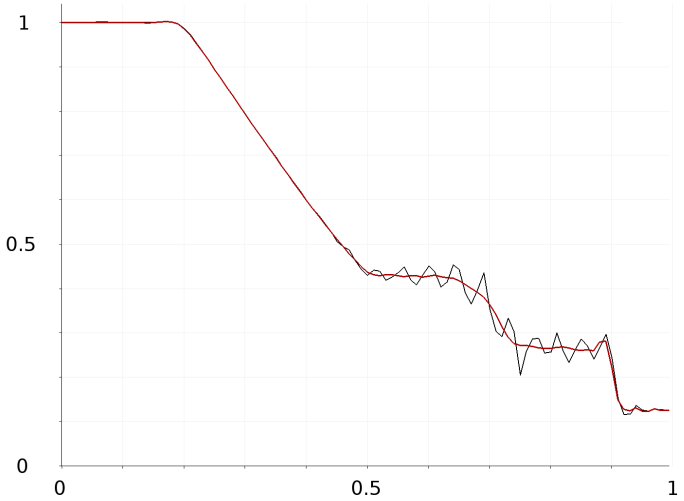
Figure 8.6: Shock tube problem: projected pressure and density

In Fig. 8.7 the CG2 solution is compared with the serendipity CG1-DG2 solution. We see that both methods resolve the rarefaction wave very well. Between the rarefaction wave and the shock the CG2 method produces large oscillations. The CG1-DG2 method produces under- and overshoots near the shock and small perturbation between the shock and the rarefaction wave. The contact discontinuity is captured by the CG1-DG2 method but smeared out a bit.

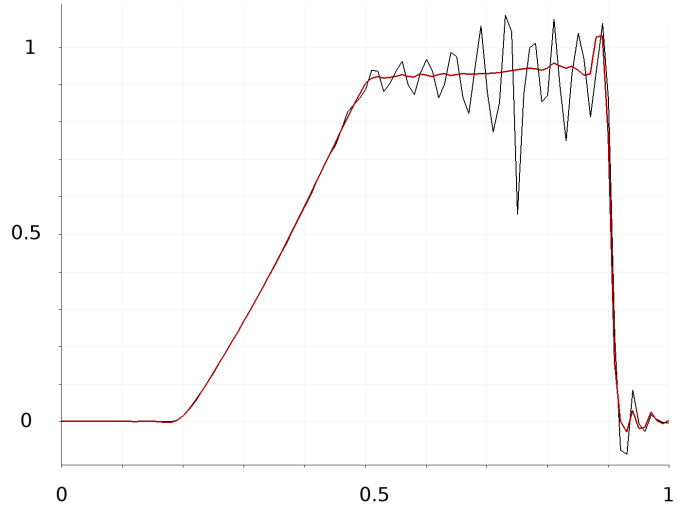
The results presented in this chapter indicate that the CG1-DG2 method is also applicable to the Euler equations and seems to be stable. However, for more challenging problems like flows with shock waves and contact discontinuities we need to investigate how the CG1-DG2 method can be modified to capture those more accurately.



(a) Pressure



(b) Density



(c) Velocity

Figure 8.7: Shock tube problem at $T = 0.231$: CG2 solution (black) and CG1-DG2 solution (red)

9

Hp-Adaptivity

In our numerical examples in Chapters 6 and 8 we have seen that the numerical solution can approximate the exact solution very well in some elements (e.g., where the solution is constant) whereas large errors occur in other elements (e.g. near discontinuities). To achieve a higher accuracy of the approximation different adaptivity strategies can be considered:

- h-adaptivity: polynomial degree p is constant, mesh size h is variable,
- p-adaptivity: polynomial degree p is variable, mesh size h is constant,
- hp-adaptivity: polynomial degree p and mesh size h are variable.

In the context of elliptic equations, e.g. Poisson's equation (3.1.1), it was shown in [7] that

$$\|u - u_{hp}\|_1 \leq C \frac{h^\mu}{p^{m-1}} \|u\|_m, \tag{9.0.1}$$

where $\mu = \min(p, m-1)$. This means that if the solution is smooth enough, increasing the polynomial degree can give higher convergence rates than mesh refinement since the rate for h-refinement is bounded by the polynomial degree. However, in most simulations the solution is not smooth so that h-refinement is preferred. If both adaptivity strategies are combined we may also expect an exponential rate of convergence with respect to the degrees of freedom whereas the other strategies only give polynomial rates [34].

In this chapter we present an hp-adaptive algorithm for convection-dominated problems in 2D and take into account that the CG1-DG2 space is a hierarchical space in the sense that if we neglect the quadratic basis functions we get the continuous linear finite element space. The use of the CG1-DG2 space leads to the restriction of the polynomial degree to $p = 1, 2$.

The idea of the hp-adaptive algorithm is that we use linear finite elements stabilized by the flux-corrected transport (FCT) algorithm [58] in regions where the solution is non-smooth (e.g., at steep gradients) and the CG1-DG2 method in regions where the solution is smooth. We emphasize that no further stabilization of the CG1-DG2 method is necessary in this case whereas the CG2 method may need stabilization. This use of higher-order elements is in good agreement with the common practice to use p-adaptivity only in regions where the solution is smooth and h-refinement elsewhere [71]. In our case, this strategy is combined with the reference solution approach for h-adaptivity.

In the following we introduce the general FCT algorithm [58, 62], a criterion for measuring the smoothness of a solution [61] and present the resulting hp-adaptive algorithm. Thereby, we follow our work already presented in [13] and [14].

9.1. Flux-corrected transport

At first we will explain a stabilization technique known as FCT scheme. It guarantees that the solution fulfills discrete maximum principles and is positivity preserving under certain assumptions. The FCT algorithm can be applied to convection-diffusion equations but for simplicity we will consider only the unsteady linear convection equation

$$u_t + \nabla \cdot (\boldsymbol{\beta} u) = 0 \quad \text{in } \Omega, \tag{9.1.1}$$

where u is a conserved quantity and $\boldsymbol{\beta}$ a given velocity field. For more details about the treatment of diffusive terms we refer to [58].

The FCT algorithm is divided in three parts. The first part is the derivation of a low-order approximation which is positivity preserving and fulfills the discrete maximum principle (see Appendix Theorem A.15). The second part considers the calculation of the numerical fluxes which are the difference between the low-order and the high-order scheme (standard Galerkin method). The last step is adding a limited amount of antidiffusive numerical fluxes to the low-order approximation. At this step, limiting is done in such a way that the updated solution remains non-oscillatory and positivity preserving.

9.1.1. High-order method

For the discretization in space we use linear continuous finite elements which yields a system of equations

$$M_C \frac{\mathrm{d}u}{\mathrm{d}t} = Ku, \tag{9.1.2}$$

where u is the vector of unknowns, $M_C = \{m_{ij}\}$ is the consistent mass matrix and $K = \{k_{ij}\}$ is the discrete transport operator. To discretize the time derivative we use the θ-scheme and obtain the fully-discretized scheme

$$\left[\frac{M_C}{\Delta t} - \theta K\right] u^{n+1} = \left[\frac{M_C}{\Delta t} + (1-\theta)K\right] u^n, \tag{9.1.3}$$

where u^n is the coefficient vector from the previous time step solution and u^{n+1} the unknown coefficient vector. This system corresponds to the CG1 method for convection equations which is not stable and can therefore produce non-physical oscillations, cf. Section 3.4 and Chapter 6.

9.1.2. Algebraic flux correction

For the derivation of a low-order scheme we take system (9.1.2) and replace the matrix M_C by its lumped counterpart

$$M_L := \operatorname{diag}\{m_i\}, \qquad m_i = \sum_j m_{ij}. \tag{9.1.4}$$

On the right-hand-side we add a discrete diffusion operator $D = \{d_{ij}\}$ defined by [58, 62]

$$d_{ii} := -\sum_{j \neq i} d_{ij}, \qquad d_{ij} = \max\{-k_{ij}, 0, -k_{ji}\} = d_{ji}, \forall j \neq i, \tag{9.1.5}$$

so that $K+D$ has no negative off-diagonal entries and D is symmetric with zero row sums. The low-order scheme is then given by

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = (K+D)u. \tag{9.1.6}$$

The difference of the low-order and high-order scheme defines the sum of antidiffusive fluxes

$$f(u) = (M_L - M_C)\frac{\mathrm{d}u}{\mathrm{d}t} - Du. \tag{9.1.7}$$

Using these fluxes the high-order scheme (9.1.2) can be rewritten in the following form

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = (K+D)u + f(u). \tag{9.1.8}$$

The entries of $f(u)$ can be simplified in the following way [58]

$$((M_L - M_C)u)_i = u_i m_i - \sum_j m_{ij}u_j = u_i \sum_j m_{ij} - \sum_j m_{ij}u_j = \sum_{j\neq i} m_{ij}(u_i - u_j), \tag{9.1.9}$$

$$(Du)_i = \sum_j d_{ij}u_j = \sum_{j\neq i} d_{ij}u_j + d_{ii}u_i = \sum_{j\neq i} d_{ij}u_j - \sum_{j\neq i} d_{ij}u_i = \sum_{j\neq i} d_{ij}(u_j - u_i). \tag{9.1.10}$$

This gives

$$f_i = \sum_{j\neq i} f_{ij}, \qquad f_{ij} = \left(m_{ij}\frac{\mathrm{d}}{\mathrm{d}t} + d_{ij}\right)(u_i - u_j), \qquad \forall j \neq i. \tag{9.1.11}$$

These fluxes can now be limited in regions where they would cause under- or overshoots, which yields the semi-discrete problem

$$M_L \frac{\mathrm{d}u}{\mathrm{d}t} = (K+D)u + \bar{f}(u). \tag{9.1.12}$$

Here, $\bar{f}(u)$ is the vector containing the sums of limited antidiffusive fluxes

$$\bar{f}_i = \sum_{j\neq i} \alpha_{ij} f_{ij}, \qquad 0 \leq \alpha_{ij} \leq 1. \tag{9.1.13}$$

The fluxes should be limited in such a way that in elements where no oscillations occur the solution tend to the high-order solution ($\alpha_{ij} \approx 1$) whereas in oscillatory regions the low-order solution is adopted ($\alpha_{ij} = 0$). In the following section we will present a limiter which fulfills these conditions.

For the discretization in time we apply the standard θ-scheme

$$Au^{n+1} = Bu^n + \bar{f}, \tag{9.1.14}$$

where $\bar{f}$ is the fully discrete limited flux term,

$$A = \frac{1}{\Delta t}M_L - \theta(K+D), \tag{9.1.15}$$

$$B = \frac{1}{\Delta t}M_L + (1-\theta)(K+D). \tag{9.1.16}$$

The implicit part of the flux term f depends on the unknown solution u^{n+1}. Therefore, it must be linearized or calculated in an iterative way [58]. Here we make use of the unconstrained Galerkin solution u^H of (9.1.3) to determine the fluxes [62] by

$$f_{ij} = \left(\frac{m_{ij}}{\Delta t} + \theta d_{ij}\right)(u_i^H - u_j^H) - \left(\frac{m_{ij}}{\Delta t} - (1-\theta)d_{ij}\right)(u_i^n - u_j^n). \tag{9.1.17}$$

Remark 9.1: Theoretically, it is also possible to define an FCT scheme for higher-order elements. However, the derivation is more complicated [59], so that it is usually applied to linear elements. One main problem with higher-order elements is that mass lumping may produce negative or zero diagonal entries for which the discrete maximum principle and positivity preservation cannot be guaranteed (see Appendix, Theorem A.15).

We have now seen that the idea of the FCT scheme is to switch between a high-order and a low-order solution, in such a way that no oscillations in the actual solution occur. We will now show, how to limit the antidiffusive fluxes, such that the solution fulfills discrete maximum principles and stays positivity preserving.

9.2. Zalesak's limiter

Let us consider the limiter used in Zalesak's multidimensional FCT algorithm [88]. It is based on a solution update of the form

$$m_i\bar{u}_i = m_i\tilde{u}_i + \Delta t \sum_{j\neq i} \alpha_{ij} f_{ij}, \tag{9.2.1}$$

where $\tilde{u}$ is a low-order approximation and $\bar{u}$ the final time-step solution. The correction factors α_{ij} are calculated by Scheme 9.2 in such a way that the local discrete maximum principle

$$u_i^{\min} \leq \bar{u}_i \leq u_i^{\max}, \quad \forall i, \tag{9.2.2}$$

holds, where

$$u_i^{\min} := \min_{j\in S_i\cup i} \tilde{u}_j \quad \text{and} \quad u_i^{\max} := \max_{j\in S_i\cup i} \tilde{u}_j$$

are the local extrema of $\tilde{u}$ and $S_i = \{j \neq i \,|\, m_{ij} \neq 0\}$ denotes the set of nearest neighbors of node i.

Scheme 9.2: Calculation of the correction factors α_{ij} [58]

1. Compute the sums of positive and negative antidiffusive fluxes

$$P_i^+ = \sum_{j\neq i} \max\{0, f_{ij}\}, \qquad P_i^- = \sum_{j\neq i} \min\{0, f_{ij}\}. \tag{9.2.3}$$

2. Define the upper and lower bounds for admissible increments

$$Q_i^+ = \frac{m_i}{\Delta t}(u_i^{\max} - \tilde{u}_i), \qquad Q_i^- = \frac{m_i}{\Delta t}(u_i^{\min} - \tilde{u}_i). \tag{9.2.4}$$

3. Compute the nodal correction factors for the components of $P_i^{\pm}$

$$R_i^+ = \min\left\{1, \frac{Q_i^+}{P_i^+}\right\}, \qquad R_i^- = \min\left\{1, \frac{Q_i^-}{P_i^-}\right\}. \tag{9.2.5}$$

4. Check the sign of the unconstrained flux and multiply f_{ij} by

$$\alpha_{ij} = \begin{cases} \min\{R_i^+, R_j^-\} & \text{if } f_{ij} > 0, \\ \min\{R_i^-, R_j^+\} & \text{if } f_{ij} < 0. \end{cases} \tag{9.2.6}$$

Fluxes f_{ij} that have the same sign as $(\tilde{u}_j - \tilde{u}_i)$ flatten the solution profile instead of steepening it [25, 58]. This can cause spurious ripples. To prevent this an optional 'prelimiting' step [25, 88] can be applied

$$f_{ij} := 0, \quad \text{if} \quad f_{ij}(\tilde{u}_j - \tilde{u}_i) > 0. \tag{9.2.7}$$

The easiest way to derive the final FCT solution is to solve the low-order scheme (9.1.6) to get a low-order approximation and then update the solution like (9.2.1). Here, we favor another approach and calculate $\bar{u}$ and the low-order approximation $\tilde{u}$ to $u(t^{n+1/2})$ by

$$\frac{1}{\Delta t} M_L \bar{u} = B u^n + \bar{f}, \tag{9.2.8}$$

$$\frac{1}{\Delta t} M_L \tilde{u} = B u^n. \tag{9.2.9}$$

By Theorem A.20 the positivity of u^n carries over to $\tilde{u}$ under the CFL-like condition [58, 62]

$$\Delta t \leq -\frac{m_i}{(k_{ii} + d_{ii})(1-\theta)}, \qquad \forall i, \tag{9.2.10}$$

and the assumption that $(k_{ii} + d_{ii}) < 0, \forall i$. Note that $d_{ii} \leq 0$ by definition.
If in addition $\sum_j k_{ij} = 0, \forall i$, then the discrete maximum principle is fulfilled, i.e. ,

$$\min u^n \leq \tilde{u} \leq \max u^n. \tag{9.2.11}$$

By Applying Zalesak's limiter we obtain a solution $\bar{u}$ of (9.2.8) which fulfills the local discrete maximum principle (9.2.2).

At last we compute the final solution u^{n+1} of (9.1.14). Using (9.2.8) the system (9.1.14) can be rewritten as

$$A u^{n+1} = \frac{1}{\Delta t} M_L \bar{u}. \tag{9.2.12}$$

If A fulfills the M-matrix properties, positivity of $\bar{u}$ carries over to u^{n+1}. If in addition $\sum_j k_{ij} = 0, \forall i$, the discrete maximum principle holds

$$\min u^n \leq u^{n+1} \leq \max u^n. \tag{9.2.13}$$

We have now shown that the presented FCT algorithm, summarized in Scheme 9.3, is positivity preserving and fulfills the discrete maximum principle, if the matrices A and B as defined in (9.1.15) and (9.1.16) fulfill the assumptions of Theorem A.20. Note that this scheme has also better stabilization properties than the CG1-DG2 method in the sense that it can handle steep gradients without causing oscillations.

Scheme 9.3: FCT algorithm

1. Compute the high-order solution $u^H \approx u(t^{n+1})$ using (9.1.3).
2. Evaluate the raw antidiffusive fluxes f_{ij} by (9.1.17).
3. Compute the low-order solution $\tilde{u}$ by (9.2.9).
4. Use $\tilde{u}$ to calculate α_{ij} by Scheme 9.2 and limit the fluxes $\bar{f}$.
5. Compute the final solution $u^{n+1} \approx u(t^{n+1})$ using (9.1.14).

Remark 9.4: Implementation of Zalesak's limiter

Algorithm 9.5 presents the edge-based implementation of Zalesak's limiter written as a pseudo-code. It takes advantage of the fact that $f_{ij} = -f_{ji}$ and $\alpha_{ij} = \alpha_{ji}$. In the case of hp-adaptivity we skip all DOFs which belong to higher-order elements.

Algorithm 9.5: Edge-based Implementation of Zalesak's limiter [58]

$P^{\pm} := 0, Q^{\pm} := 0, \bar{f} := 0$
for all i **do**
  for all $j \in S_i, j > i$ **do**
    $P_i^{\pm} = P_i^{\pm} + \begin{matrix}\max\\ \min\end{matrix} \{0, f_{ij}\}$
    $P_j^{\pm} = P_j^{\pm} + \begin{matrix}\max\\ \min\end{matrix} \{0, -f_{ij}\}$
    $Q_i^{\pm} = \begin{matrix}\max\\ \min\end{matrix} \{Q_i^{\pm}, \frac{m_i}{\Delta t}(u_j - u_i)\}$
    $Q_j^{\pm} = \begin{matrix}\max\\ \min\end{matrix} \{Q_j^{\pm}, \frac{m_j}{\Delta t}(u_i - u_j)\}$
  end for
end for
for all i **do**
  $R_i^{\pm} = \min\{1, \frac{Q_i^{\pm}}{P_i^{\pm}}\}$
end for
for all i **do**
  for all $j \in S_i, j > i$ **do**
    if $f_{ij} > 0$ **then**
      $\alpha_{ij} = \min\{R_i^+, R_j^-\}$
    else
      $\alpha_{ij} = \min\{R_i^-, R_j^+\}$
    end if
    $\bar{f}_i = \bar{f}_i + \alpha_{ij} f_{ij}$
    $\bar{f}_j = \bar{f}_j - \alpha_{ij} f_{ij}$
  end for
end for

In the following, we will present another important component of our hp-adaptive framework, which helps us to identify elements for p-refinement.

9.3. Regularity estimator

The idea of our p-refinement strategy is to increase the polynomial degree only in elements where the solution can be regarded as smooth. In those elements the CG1-DG2 method guarantees enough stability, such that the FCT scheme does not need to be applied. For that reason we estimate the local regularity of the numerical solution u_h and obtain the information if the solution is smooth enough to use the CG1-DG2 method.

In order to estimate the smoothness of u_h in a neighborhood of an element $K \in \mathcal{T}_h$, we compare u_h with a linear approximation of the form [61]

$$\hat{u}_h(\mathbf{x}) = u_h(\mathbf{x}_c) + R_h u_h(\mathbf{x}_c) \cdot (\mathbf{x} - \mathbf{x}_c), \tag{9.3.1}$$

where $\mathbf{x}_c$ denotes the center of K and $R_h : V_h \to V_h \times V_h$ is a gradient recovery operator. Here we will use an L^2 projection to construct $R_h u_h = (R_h^1 u_h, R_h^2 u_h)^T$ (see Section 9.3.1). This gradient approximation is continuous even if ∇u_h is not. Note that the approximation $\hat{u}_h$ is defined element-wise and may be discontinuous across element boundaries.

The solution on an element K is regarded as smooth if the value of $\hat{u}_h$ at each vertex $\mathbf{x}_i \in K$ is bounded by the values of u_h at the centers of surrounding elements [61]

$$u_i^{\min} < \hat{u}_h(\mathbf{x}_i) < u_i^{\max}, \qquad \forall \mathbf{x}_i \in K, \tag{9.3.2}$$

where

$$u_i^{\max} := \max\{u_h(\mathbf{x}_c) \mid \exists K \in \mathcal{T}_h : \ \mathbf{x}_i, \mathbf{x}_c \in K\}, \tag{9.3.3}$$

$$u_i^{\min} := \min\{u_h(\mathbf{x}_c) \mid \exists K \in \mathcal{T}_h : \ \mathbf{x}_i, \mathbf{x}_c \in K\}. \tag{9.3.4}$$

Remark 9.6: Implementation of strict inequalities

For implementation purposes we replace (9.3.2) by

$$u_i^{\min} + \varepsilon < \hat{u}_h(\mathbf{x}_i) < u_i^{\max} - \varepsilon, \tag{9.3.5}$$

where ε is a small positive number.

In (9.3.2) we use strict inequalities, which implies that constant functions are not regarded as smooth. In the context of our hp-adaptive algorithm this is a reasonable assumption since a higher polynomial degree, which is used for smooth functions, does not lead to a higher accuracy for the approximation of constant functions. Elements, in which local extrema occur, violate condition (9.3.2). To motivate this a simple 1D-example is presented in the following.

Example 9.7: Smoothness indicator for the hat function

Let us consider the hat function and its derivative

$$u(x) = \begin{cases} 0 & \text{if } 0 \le x < 1 \\ x-1 & \text{if } 1 \le x \le 2 \\ 3-x & \text{if } 2 < x \le 3 \\ 0 & \text{if } 3 < x \le 4 \end{cases} \qquad u'(x) = \begin{cases} 0 & \text{if } 0 \le x < 1 \\ 1 & \text{if } 1 \le x \le 2 \\ -1 & \text{if } 2 < x \le 3 \\ 0 & \text{if } 3 < x \le 4 \end{cases} \tag{9.3.6}$$

on the interval $[0,4]$. To determine the smoothness on each element, i.e., on interval $I_i = [i, i+1], i = 0,..,3$, we project the derivative of $u(x)$ into the space of linear finite elements.

In Fig. 9.1 the discontinuous derivative of the hat function and its continuous projection are displayed. The next step is to determine the linear approximation (9.3.1). In Fig. 9.3 this approximation as well as the hat function are displayed.

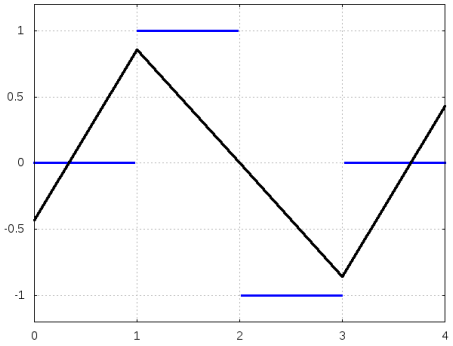


Figure 9.1: Derivative of hat function (blue) and its L^2 projection (black)

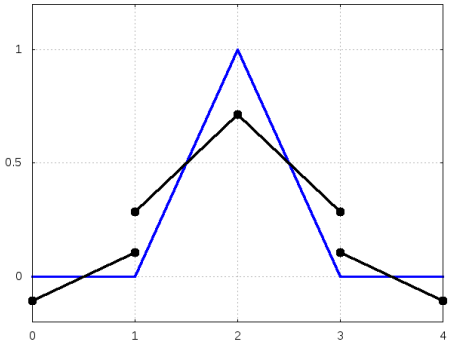


Figure 9.2: Hat function (blue) and linear approximation (black)

The last step is to check for which intervals $I_i, i = 0,...,3$, (9.3.2) holds. For interval I_0 we have to check two vertices, namely $v_0 = 0$ and $v_1 = 1$. For vertex v_0 we obtain $\hat{u}(v_0) < 0$ and as surrounding neighbor element I_0. Note that in 1D all boundary vertices have only one neighboring element. In this case we introduce a ghost neighbor I_{-1}. As solution value at the midpoint of this ghost element we use the constant extension of the boundary, i.e. $u(x_c)|_{I_{-1}} = u(v_0)$. Evaluating the solution at the midpoints of I_0 and I_{-1} gives $u_0^{\min} = u_0^{\max} = 0$. Therefore condition (9.3.2) is not fulfilled and we can mark the element I_0 as non-smooth.

For the next element I_1 we have for the vertices $v_1 = 1$ and $v_2 = 2$: $0 < \hat{u}(v_1) < 0.5$ and $0.5 < \hat{u}(v_2)$. The surrounding neighbors of v_1 are I_0 and I_1 which leads to $u_1^{\min} = 0$ and $u_1^{\max} = 0.5$ and therefore condition (9.3.2) is fulfilled for v_1. It remains to check v_2. The neighbors for this element are I_1 and I_2 and therefore $u_2^{\min} = u_2^{\max} = 0.5$. This gives $u_2^{\max} < \hat{u}_h(v_2)$ and therefore the solution on this element is regarded as non-smooth.

For the remaining elements we get similar results so that all elements will be marked as non-smooth.

Since the regularity estimator applied to the solution cannot distinguish between smooth peaks and spurious under-/overshoots, and therefore both would be marked as non-smooth, we additionally need to determine the smoothness of each component of the gradient $\nabla u_h = (u_x, u_y)^T$ [61]. If the solution and/or both components of its gradient are smooth, the element can be regarded as smooth. Note that only the function and not the gradient was used to determine the smoothness in Example 9.7.

Following [61] we define the linear reconstruction

$$\hat{g}_h^1(\mathbf{x}) = \frac{\partial u_h}{\partial x}(\mathbf{x}_c) + \nabla(R_h^1 u_h)(\mathbf{x}_c) \cdot (\mathbf{x} - \mathbf{x}_c), \tag{9.3.7}$$

$$\hat{g}_h^2(\mathbf{x}) = \frac{\partial u_h}{\partial y}(\mathbf{x}_c) + \nabla(R_h^2 u_h)(\mathbf{x}_c) \cdot (\mathbf{x} - \mathbf{x}_c), \tag{9.3.8}$$

where $R_h u_h = (R_h^1 u_h, R_h^2 u_h)^T$ is the recovered gradient. To estimate the regularity of the gradient we compare the solution gradient ∇u_h and its linear reconstruction $\hat{g}_h = (\hat{g}_h^1, \hat{g}_h^2)^T$. This is done similar to the function case.

The gradient ∇u_h on an element K is regarded as smooth if the values of $\hat{g}_h^1$ and $\hat{g}_h^2$ at all vertices $\mathbf{x}_i \in K$ are bounded by the centroid values of $\frac{\partial u_h}{\partial x}$ and $\frac{\partial u_h}{\partial y}$, respectively [13]:

$$\left(\frac{\partial u_h}{\partial x}\right)_i^{\min} < \hat{g}_h^1(\mathbf{x_i}) < \left(\frac{\partial u_h}{\partial x}\right)_i^{\max}, \qquad \forall \mathbf{x}_i \in K, \tag{9.3.9}$$

$$\left(\frac{\partial u_h}{\partial y}\right)_i^{\min} < \hat{g}_h^2(\mathbf{x_i}) < \left(\frac{\partial u_h}{\partial y}\right)_i^{\max}, \qquad \forall \mathbf{x}_i \in K, \tag{9.3.10}$$

where

$$\left(\frac{\partial u_h}{\partial x}\right)_i^{\max} := \max\left\{\frac{\partial u_h}{\partial x}(\mathbf{x}_c) \mid \exists K \in \mathcal{T}_h : \ \mathbf{x}_i, \mathbf{x}_c \in K\right\}, \tag{9.3.11}$$

$$\left(\frac{\partial u_h}{\partial x}\right)_i^{\min} := \min\left\{\frac{\partial u_h}{\partial x}(\mathbf{x}_c) \mid \exists K \in \mathcal{T}_h : \ \mathbf{x}_i, \mathbf{x}_c \in K\right\}, \tag{9.3.12}$$

$$\left(\frac{\partial u_h}{\partial y}\right)_i^{\max} := \max\left\{\frac{\partial u_h}{\partial y}(\mathbf{x}_c) \mid \exists K \in \mathcal{T}_h : \ \mathbf{x}_i, \mathbf{x}_c \in K\right\}, \tag{9.3.13}$$

$$\left(\frac{\partial u_h}{\partial y}\right)_i^{\min} := \min\left\{\frac{\partial u_h}{\partial y}(\mathbf{x}_c) \mid \exists K \in \mathcal{T}_h : \ \mathbf{x}_i, \mathbf{x}_c \in K\right\}. \tag{9.3.14}$$

In summary, we mark an element K as smooth, if the solution is regarded as smooth, i.e., (9.3.2) holds, and/or both components of the gradient are regarded as smooth, i.e.,(9.3.9) and (9.3.10) hold.

In the following example we apply the regularity estimator to a function with a smooth peak and show that the smoothness of the gradient has to be taken into account.

Example 9.8: Regularity estimator applied to a smooth function

Let us consider the following function and its derivative

$$u(x) = 4 - (x-2)^2, \quad u'(x) = 4 - 2x, \tag{9.3.15}$$

on the interval $[0,4]$. To determine the smoothness on each element, i.e., on interval $I_i = [i, i+1], i = 0,..,3$, we project the derivative of $u(x)$ into the space of linear finite elements. Since $u'(x)$ is a continuous linear function, we obtain $R_h u(x) = u'(x)$. The next step is to determine the linear approximation (9.3.1). In Fig. 9.3 this approximation as well as $u(x)$ are displayed.

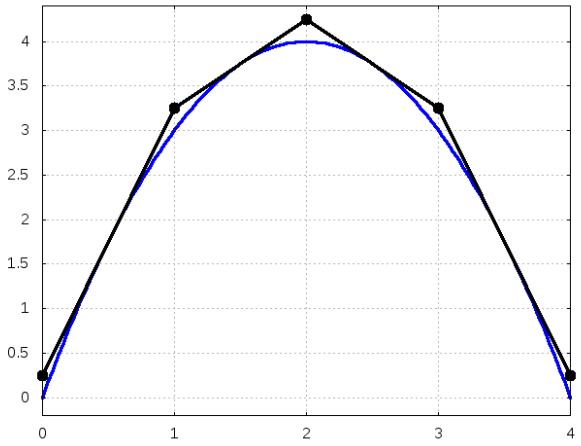


Figure 9.3: Function $u(x)$ (blue) and its linear approximation (black)

The next step is to check for which intervals $I_i, i = 0,...,3$, (9.3.2) holds. For interval I_0 we have to consider vertices $v_0 = 0$ and $v_1 = 1$. For vertex v_0 we obtain $\hat{u}(v_0) = 0.25$. The surrounding neighbor elements are I_0 and the ghost neighbor I_{-1}, where $u(x_c)|_{I_{-1}} = u(v_0)$. Evaluating the solution at the midpoints of I_0 and I_{-1} gives $u_0^{\min} = 0$ and $u_0^{\max} = 1.75$, which means that condition (9.3.2) is fulfilled for this vertex. For vertex v_1 we obtain $\hat{u}(v_1) = 3.25$, $u_1^{\min} = 1.75$ and $u_1^{\max} = 3.75$. Therefore, condition (9.3.2) is fulfilled for both vertices and we can mark the element I_0 as smooth. Due to symmetry the same holds for element I_3.

Since the linear approximation is continuous, we have $\hat{u}(v_1)|_{I_0} = \hat{u}(v_1)|_{I_1}$. However, $\hat{u}(v_2)|_{I_1} = \hat{u}(v_2)|_{I_2} = 4.25$ and $u_2^{\min} = u_2^{\max} = 3.75$, which means that condition (9.3.2) is not fulfilled for elements I_1 and I_2. Therefore, we have to check the derivative and its approximation for elements I_1 and I_2. We obtain that the linear approximation $\hat{g}(x)$ of the derivative $u'(x)$ is continuous and can be simplified to $\hat{g}(x) = u'(x)$. Since $u'(x)$ is strictly monotonically decreasing, it is easy to verify that condition (9.3.9) holds on all elements. Therefore, the derivative is regarded as smooth on all elements and we mark all elements as smooth.

We remark that we obtain for the L^2 projection $\pi_h^1 u$ of u into the linear finite element space that it is smooth on I_0 and I_3 and its derivative is smooth on I_1 and I_2. Therefore, all elements are marked as smooth.

Remark 9.9: Implementation

For implementation purposes we introduce the smoothness sensors $\eta_i^0, \eta_i^x, \eta_i^y, i = 1,...,N_{vert}$, where N_{vert} is the number of vertices. If (9.3.2) holds in all elements containing the vertex $\mathbf{x}_i$, then $\eta_0^i = 1$, else we set $\eta_0^i = 0$. This means that the solution in a small neighborhood of vertex $\mathbf{x}_i$ can be regarded as smooth if $\eta_0^i = 1$. Similarly we get the values of η_i^x and η_i^y by testing conditions (9.3.9) and (9.3.10) which gives information about the regularity of the derivatives w.r.t. x and y. Since the marker η^0 can fail to distinguish between smooth peaks and oscillations, we combine the informations of all markers and define $\eta^{max} = \max(\eta^0, \min(\eta^x, \eta^y))$. If $\eta_i^{max} = 1$, we mark vertex $\mathbf{x}_i$ as smooth. Only if all vertices $\mathbf{x}_i$ in an element K are marked as smooth, do we mark element K as smooth. In this context we will also use $\eta_{elem} = 1$ to identify smooth elements. Note, that this smoothness sensor is more restrictive than only the conditions (9.3.2) and/or (9.3.9) and (9.3.10) which can also be seen in the following example.

Example 9.10: Marking elements in 2D

Let us assume we have a uniform mesh with 64 elements.

Step 1: Determine the elements for which (9.3.2) and/or (9.3.9) and (9.3.10) hold.

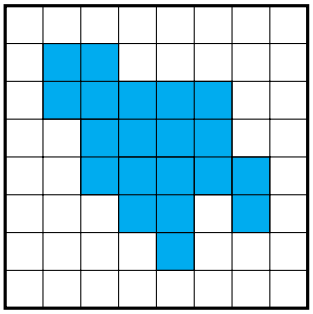


Figure 9.4: Step 1: Smooth elements (light blue)

Step 2: Apply sensor η^{max} to all vertices and mark vertex $\mathbf{x}_i$ as smooth if $\eta_i^{max} = 1$.

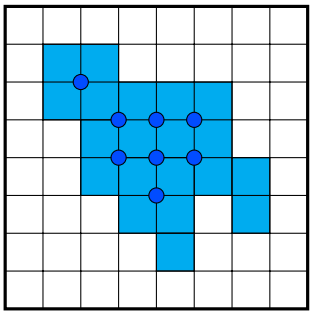


Figure 9.5: Step 2: Smooth vertices (blue circle)

Step 3: Mark an element K as smooth if $\eta_i^{max} = 1$ for all vertices $\mathbf{x}_i \in K$.

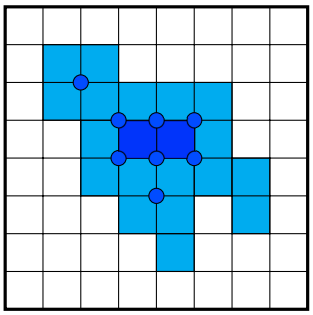


Figure 9.6: Step 3: Updated smooth elements (dark blue)

An important part of the regularity estimator is the approximation of the gradient. In the following, we explain how to derive a continuous approximation by using the L^2 projection.

9.3.1. Gradient reconstruction

Let us consider a finite element solution $u_h \in V_h$, where $V_h = span\{\varphi_1, \ldots, \varphi_N\}$. To get a gradient approximation in V_h we define the operator $R_h : V_h \to V_h \times V_h$ by

$$R_h^k u_h = \sum_{j=1}^{N} r_j^k \varphi_j, \quad k = 1,2, \tag{9.3.16}$$

where $r_j^1 \approx \frac{\partial u_h}{\partial x}(\mathbf{x}_j)$ and $r_j^2 \approx \frac{\partial u_h}{\partial y}(\mathbf{x}_j)$. In our case we will determine the coefficients $r_j^1, r_j^2, j = 1, ..., N$, using the L^2 projection

$$\int_\Omega \varphi_i R_h u_h \, d\mathbf{x} = \int_\Omega \varphi_i \nabla u_h \, d\mathbf{x}, \quad i = 1, \ldots, N. \tag{9.3.17}$$

This gives for each component $R_h^k, k = 1,2$ a linear system

$$M_C r^k = b^k, \tag{9.3.18}$$

where $M_C = \{m_{ij}\}$ is the consistent mass matrix and $b^k = \{b_j^k\}$, $k = 1,2$, is the load vector associated with the k-th derivative

$$(b^1, b^2)^T = \int_\Omega \varphi_i \nabla u_h \, d\mathbf{x}. \tag{9.3.19}$$

9.4. Reference solution approach

In the last sections, we have introduced different components of our hp-adaptive framework. Now we will bring them together, add a h-refinement strategy and derive an hp-adaptive algorithm. As h-adaptive strategy we adopt the reference solution approach usually considered for hp-adaptivity [81–83]. This strategy is based on the assumption that the reference solution leads to a better approximation of the exact solution than the solution on the current space. In the context of hp-adaptivity, the reference solution is computed on a reference space which is created by increasing the polynomial degree and refining the mesh size of the current (coarse) space. In contrast to this original approach, we create the reference space by increasing the polynomial degree only in elements which are marked as smooth and h-refining all non-smooth elements. Here, the regularity estimator from the last section helps to determine the smooth elements. For our computations we use FCT only in matrix blocks associated with the $P1/Q1$ approximation. Note that due to the use of the CG1-DG2 space we also restrict the polynomial degree to $p \leq 2$.

The hp-adaptive algorithm works iteratively. We start with an initial coarse space and initial data u^0 and update these in each time step in the following way [13]:

Scheme 9.11: Hp-adaptive algorithm

1. Adaptivity loop:
   (a) Construct the reference space by setting $p = 2$ in smooth elements and $p = 1$ in non-smooth elements. Additionally, all non-smooth elements are h-refined.
   (b) Project the old solution u^n into the reference space and compute the reference solution u^{ref}.
   (c) Project the reference solution into the coarse space and calculate the difference between the reference solution and its projection.
   (d) Adjust the local mesh size of the coarse mesh/space according to the error indicator in (c).
2. Set $u^{n+1} = u^{ref}$.

We finish the adaptivity loop if either no refinement was done or a user-defined number of adaptivity steps has been reached. Note that the solution on the coarse space/mesh is never explicitly computed. The problem is only solved on the reference space. Since this algorithm does not coarsen the mesh, an additional coarsening step can be included where the coarse space is either set to the initial mesh or the previous refinement is reversed.

Remark 9.12: Implementation

In our implementation we distinguish between smooth elements and smooth vertices. For the construction of the reference space we increase the polynomial degree only in smooth elements. Since we use an edge-based version of Zalesaks limiter (see Algorithm 9.5), FCT is only applied to edges whose vertices do not belong to an higher-order element. This means that after increasing p we start with identifying the degrees of freedom which belong only to $Q1/P1$ elements. Only for these degrees of freedom, corresponding to linear vertex functions, mass lumping and the derivation of artificial diffusion is necessary. For all other DOFs the standard mass matrix is kept and no artificial diffusion is added. During the process of calculating α (see Scheme 9.2 and Algorithm 9.5) all these smooth-element-DOFs are skipped.

As we have seen in Example 9.10, there can be smooth vertices ($\eta^{max} = 1$) which are surrounded by $Q1/P1$ elements only. At such vertices we set $R_i^+ = R_i^- = 1$ in Algorithm 9.5 and obtain the high-order solution [61].

In the following, we will show a possible approach to keep the number of DOFs low.

9.4.1. Hanging nodes

Since often only a few elements need to be refined, it is easier to use irregular meshes to keep the number of DOFs low. These irregular meshes feature hanging nodes which are vertices on one element but interior points of another element's edge [57, 84]. If an element of a regular mesh is refined in such a way that a hanging node occurs, we call this node a 1-level hanging node. If the element containing this hanging node is refined again, we get a 2-level hanging node and so on. These different levels are shown in Fig. 9.7 for quadrilateral meshes. Vertices which correspond

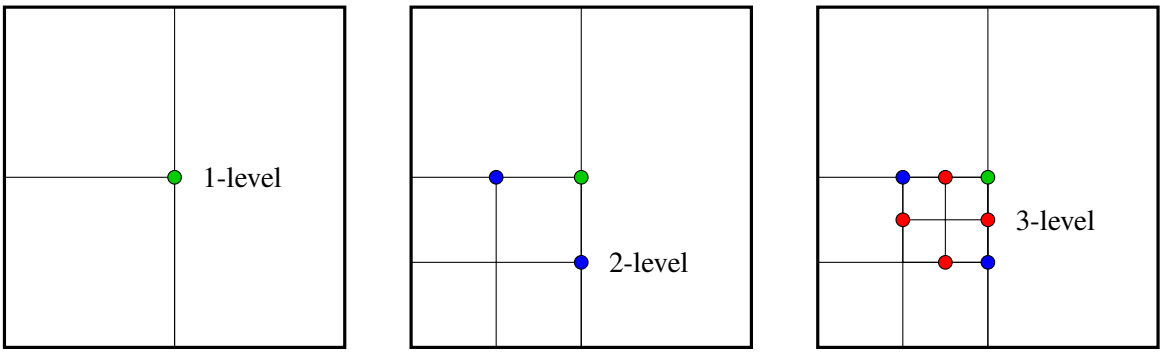


Figure 9.7: Meshes with 1,2 and 3-level hanging nodes

to hanging nodes are not degrees of freedom. The solution values at these vertices are constrained by neighboring nodes, which are called constraining nodes (see Fig. 9.8 of Example 9.13). Note that constraining nodes can also be constrained nodes. A detailed description of hanging nodes and the resulting definition of global basis functions in 3D can be found in [57].

In the following example we show, how global basis functions are defined in the context of hanging nodes.

Example 9.13: Global basis functions

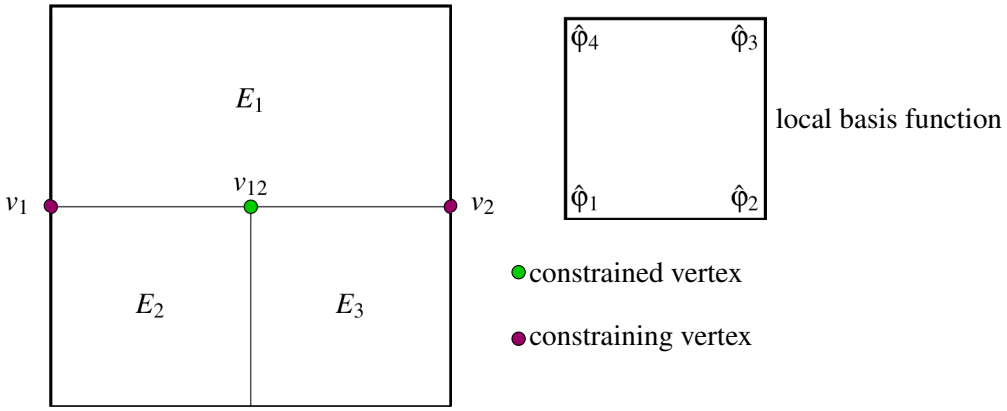


Figure 9.8: Constrained and constraining vertices

Let us consider the mesh from Fig. 9.8 which consists of three elements E_1, E_2 and E_3. The vertices of interest are v_1, v_2 (constraining vertices) and the hanging node v_{12} (constrained vertex). For the definition of global basis functions φ_1 and φ_2 corresponding to nodes v_1 and v_2, respectively, we iterate over all elements. On each element E_k we have local basis function $\hat{\varphi}_i^{E_k}$, $i = 1,2,3,4$. Note that $\hat{\varphi}_i^{E_k}(x) = 0 \forall x \notin E_k, i = 1,...,4$.
For element E_1 we get the standard basis function

$$\varphi_1|_{E_1} = \hat{\varphi}_1^{E_1},$$

where $\hat{\varphi}_1^{E_1}$ is the corresponding local basis function on E_1.
On element E_2 we get

$$\varphi_1|_{E_2} = \hat{\varphi}_4^{E_2} + \frac{1}{2}\hat{\varphi}_3^{E_2},$$

where $\hat{\varphi}_4^{E_2}$ is the corresponding local basis function on E_2 and $\hat{\varphi}_3^{E_2}$ is the local basis function corresponding to the constrained vertex.
On element E_3 we get

$$\varphi_1|_{E_3} = \frac{1}{2}\hat{\varphi}_4^{E_3},$$

which corresponds again to the constrained vertex.
Similarly we can define

$$\varphi_2 = \hat{\varphi}_2^{E_1} + \hat{\varphi}_3^{E_3} + \frac{1}{2}\hat{\varphi}_4^{E_3} + \frac{1}{2}\hat{\varphi}_3^{E_2},$$

where $\hat{\varphi}_4^{E_3}$ and $\hat{\varphi}_3^{E_2}$ are the local basis functions of the constrained vertex on element E_3 and E_2, respectively.
The transition from one element to another is continuous as we will see in the following. W.l.o.g. we assume $v_1 = (0,0)$, $v_{12} = (0.5,0)$ and $v_2 = (1,0)$. Let $\mathbf{x}$ be a point at the edge between vertices v_1 and v_{12} and we wish to evaluate $\varphi_1(\mathbf{x})$. If we regard this point as part of E_1 we get $\varphi_1(\mathbf{x}) = \hat{\varphi}_1^{E_1}(\mathbf{x})$. If we assume $\mathbf{x} \in E_2$, we get $\varphi_1(\mathbf{x}) = \hat{\varphi}_4^{E_2}(\mathbf{x}) + \frac{1}{2}\hat{\varphi}_3^{E_2}(\mathbf{x})$.
In order to satisfy the continuity requirement, the relation

$$\hat{\varphi}_1^{E_1}(\mathbf{x}) = \hat{\varphi}_4^{E_2}(\mathbf{x}) + \frac{1}{2}\hat{\varphi}_3^{E_2}(\mathbf{x}), \quad \forall \mathbf{x} = (x,0), x \in [0,0.5], \tag{9.4.1}$$

must hold.
This is true, since if we look at the definition of the basis function on the edge, we get

$$\hat{\varphi}_1^{E_1}(\mathbf{x}) = 1 - x, \quad \forall \mathbf{x} = (x,0), x \in [0,1],$$
$$\hat{\varphi}_4^{E_2}(\mathbf{x}) = 1 - 2x \text{ and } \hat{\varphi}_3^{E_2}(\mathbf{x}) = 2x, \quad \forall \mathbf{x} = (x,0), x \in [0,0.5].$$

Since $x \in [0,0.5]$ and $\mathbf{x} = (x,0)$, we obtain (9.4.1).

In our computations we use hanging nodes in combination with the FCT scheme, which is directly applied to DOFs, such that the fluxes in hanging nodes are not directly influenced.

9.4.2. Adjusting the local mesh size

We will now explain step (c) and (d) of the adaptivity loop in Scheme 9.11 in more detail. In step (c) we project the reference solution into the coarse space using the L^2 projection and assume that this is the coarse solution. The relative error between these solutions measured in an arbitrary norm indicates if the reference space should be further refined or the adaptivity loop can be terminated. In step (d) the real refinement of the coarse space is done. At first we decide which elements should be refined and then select the type of refinement. If the error between the reference solution and the coarse solution on a coarse element is larger than the maximal element error multiplied by a user-defined threshold we mark this element for refinement. Next a list of possible refinement candidates including the coarse element is defined. In Fig. 9.9 different h-refinement options for quadrilateral meshes with anisotropic refinement are shown. Note that in the context of hp-adaptivity the range of candidates is much larger due to different possibilities for the polynomial degree. For each element and each refinement candidate the error between the reference solution and its projection on the refinement candidate is calculated. If for one candidate the error is larger than the error corresponding to the coarse solution, we neglect this candidate. The optimal candidate for each refinement is then chosen by taking into account the error as well as the resulting DOFs. For more details we refer to [71]. Note that the optimal candidate can also be the coarse element itself such that no refinement is done.

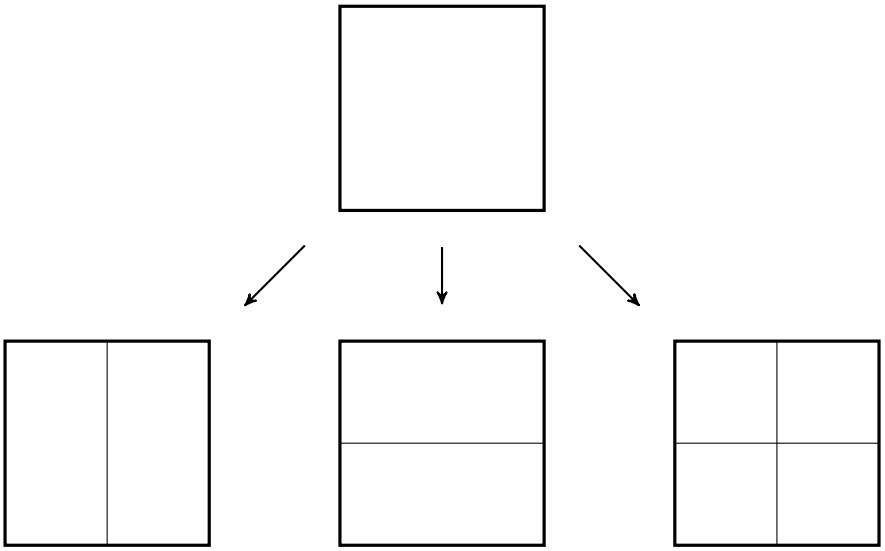


Figure 9.9: Candidates for anisotropic h-refinement

9.5. Constrained L^2 projection

In the presented Scheme 9.11 the previous time step solution needs to be projected onto the reference space. This projection should be conservative and not cause any oscillations. However, the standard L^2 projection can fail to fulfill these conditions. The lumped-mass L^2 projection in the linear continuous finite element space satisfies the maximum principle but can be too diffusive. The difference between the consistent and the lumped-mass L^2 projections can be decomposed into fluxes [63] so that we can apply the FCT limiter in the case of matrix blocks associated with the $P1/Q1$ approximation. In the context of the hp-adaptive algorithm 9.11 it is safe to use the standard L^2 projection in smooth elements, i.e., matrix blocks associated with the $P2/Q2$/serendipity approximation.

For simplicity, let us consider the space of continuous linear finite elements $V_h = span\{\varphi_1, \dots, \varphi_N\}$. The standard L^2 projection of a function u into V_h is defined by

$$\int_\Omega u_h w_h \, d\mathbf{x} = \int_\Omega u w_h \, d\mathbf{x}, \qquad \forall w_h \in V_h, \tag{9.5.1}$$

where u_h is the numerical approximation. This gives a system of equations

$$M_C u^H = R, \tag{9.5.2}$$

where $M_C = \{m_{ij}\}$ is the consistent mass matrix and R is the load vector with components $R_i = \int_\Omega \varphi_i u \, d\mathbf{x}$. Replacing M_C by its lumped counterpart (9.1.4) gives the lumped-mass L^2 projection

$$M_L u^L = R. \tag{9.5.3}$$

The L^2 projection (9.5.2) can be rewritten in the following form

$$M_L u^H = M_L u^L + f^{proj}, \tag{9.5.4}$$

where the flux f^{proj} is defined as the difference of (9.5.2) and (9.5.3)

$$f_{ij}^{proj} = m_{ij}(u_i^H - u_j^H), \quad f_i^{proj} = \sum_{j \neq i} f_{ij}^{proj}. \tag{9.5.5}$$

In the spirit of algebraic flux correction, we limit the fluxes f_{ij}^{proj} in such a way that no under- or overshoots occur and derive the vector of limited fluxes

$$\bar{f}_i^{proj} = \sum_{j \neq i} \alpha_{ij} f_{ij}^{proj}, \qquad 0 \leq \alpha_{ij} \leq 1. \tag{9.5.6}$$

Hereby we use Zalesak's limiter (see Scheme 9.2 with $\tilde{u} = u^L$ and $\Delta t = 1$) to calculate the correction factors α_{ij}. The last step is the calculation of the final solution by

$$M_L u = M_L u^L + \bar{f}^{proj}. \tag{9.5.7}$$

The use of this constrained projection guarantees that the previous time-step/adaptivity-step solution is transfered to the actual space in such a way that no oscillations occur.

In summary of this chapter, we have presented an hp-adaptive framework which guarantees that the solution stays oscillation free by using the FCT scheme in non-smooth and the CG1-DG2 method in smooth elements. The smoothness is determined by a regularity estimator and also implies p-enrichment. The application of the reference solution approach leads to h-refinement.

In the following numerical examples, we will show the advantage of this hp-adaptive framework over pure h-refinement in the context of FCT schemes.

10

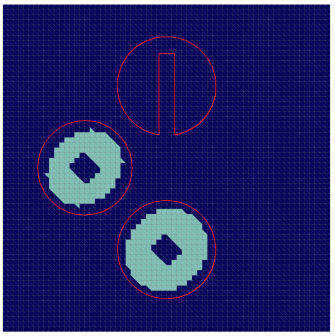
Numerical results

In this chapter we present numerical results related to the topics discussed in the previous chapter. The presented algorithms have been implemented in the open-source C++ library HERMES [82]. It provides a framework for *hp*-adaptive continuous as well as discontinuous Galerkin methods.

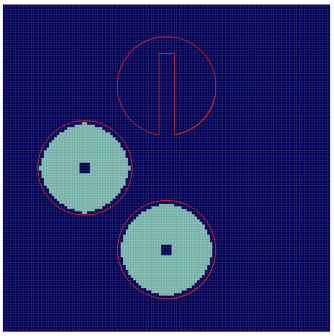
We will start with some examples for the regularity estimator. Thereby, we will also show the influence of the parameter ε from equation (9.3.5). The next section will then be concerned with different types of projections to motivate the use of the constrained L^2 projection. Then we will present results for FCT on uniform meshes and in combination with *h*-and *hp*-adaptivity. These examples indicate the advantage of the *hp*-adaptive strategy over pure *h*-refinement in the context of FCT schemes.

10.1. Regularity estimator

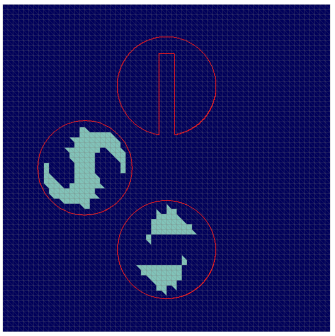
We will now show the ability of the regularity estimator to distinguish between smooth and non-smooth parts of the solution. At first we consider the initial data from the solid body rotation problem from Section 6.5, see Fig. 6.14. According to Remark 9.9 the different smoothness markers are displayed in Fig. 10.1 . Hereby, all smooth elements are marked in light blue whereas all non-smooth elements are marked in dark blue. Note that an element K is regarded as smooth if $\eta_i = 1$ for all vertices $\mathbf{x}_i \in K$. We can see that η^0 indicates the smoothness of the hump and the cone without the peaks. The markers η^x and η^y also exclude the smoothness at the top of the cone. However, in both cases the top of the hump is regarded as smooth. The marker $\eta^{all} = \max(\eta^0, \min(\eta^x, \eta^y))$ combines both smoothness criteria so that the hump as well as the boundaries of the cone without the peak are marked as smooth. In accordance with Remark 9.6 we set $\varepsilon = 10^{-8}$.



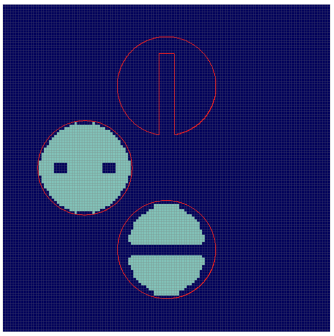
(a) η^0_{elem} on triangular mesh



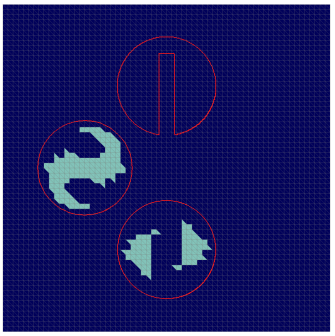
(b) η^0_{elem} on quadrilateral mesh



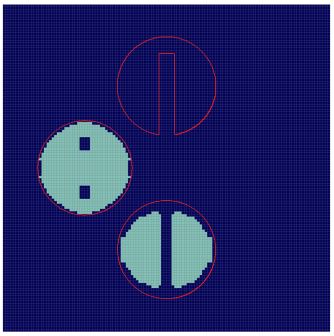
(c) η^x_{elem} on triangular mesh



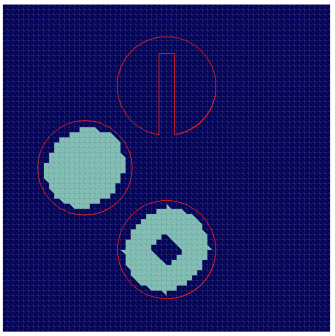
(d) η^x_{elem} on quadrilateral mesh



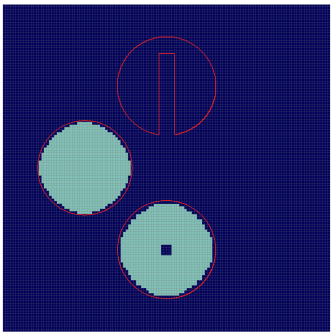
(e) η^y_{elem} on triangular mesh



(f) η^y_{elem} on quadrilateral mesh



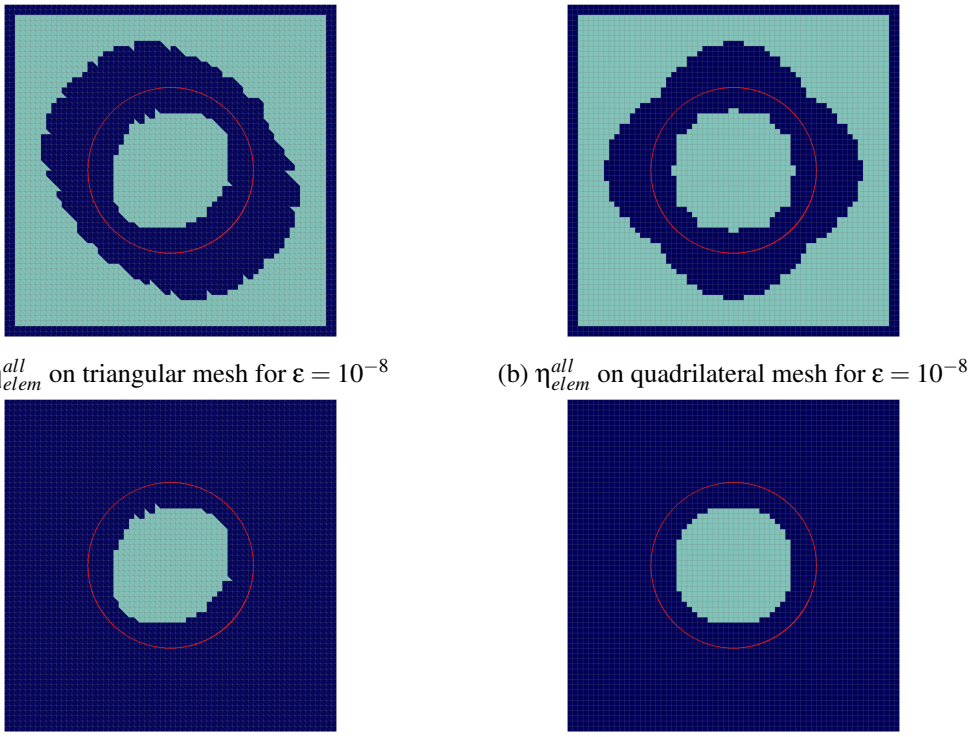
(g) η^{all}_{elem} on triangular mesh



(h) η^{all}_{elem} on quadrilateral mesh

Figure 10.1: Smoothness sensor on triangular and quadrilateral meshes for the solid body rotation problem: smooth (light blue) and non-smooth (dark blue), isoline of the data for $z = 0.001$ (red)

In the next example we consider the exact solution at $t = 0.5$ of the hump changing height problem from section 6.8, see Fig. 6.22. By this problem we show the influence of the free parameter ε (see Remark 9.6) which is responsible for the identification of constant solution parts as non-smooth. If the parameter is chosen too small constant functions are regarded as smooth. In Fig. 10.2 we compare the results for $\varepsilon = 10^{-8}$ and $\varepsilon = 10^{-3}$ on triangular and quadrilateral meshes. In both cases the top of the hump is regarded as smooth. However, for $\varepsilon = 10^{-8}$ also a lot of elements in which the solution should be constant are marked as smooth. If we increase ε these elements become non-smooth. In both cases the elements at the steep boundaries of the hump are marked as non-smooth.



(a) η_{elem}^{all} on triangular mesh for $\varepsilon = 10^{-8}$

(b) η_{elem}^{all} on quadrilateral mesh for $\varepsilon = 10^{-8}$

(c) η_{elem}^{all} on triangular mesh for $\varepsilon = 10^{-3}$

(d) η_{elem}^{all} on quadrilateral mesh for $\varepsilon = 10^{-3}$

Figure 10.2: Smoothness sensor on triangular and quadrilateral meshes for the hump changing height problem: smooth (light blue) and non-smooth (dark blue), isoline of the data for $z = 0.5$ (red)

We remark, that the initial data is often not given as a function of the finite element space. Therefore, it can first be projected into that space using the L^2 projection before the regularity estimator is applied. This procedure is feasible since the L^2 projection does not change the smoothness of a function (cf. section 10.2).

The presented results indicate that the regularity estimator identifies non-smooth data in a very reliable way. If the parameter ε is too small, elements, on which the solution is constant, may be regarded as smooth. In the following remark, we will discuss this problem in the context of hp-adaptivity.

Remark 10.1: Regularity estimator in the context of hp-adaptivity

In the context of the hp-adaptive algorithm presented in section 9.4 the identification of non-smooth elements in the neighborhood of steep fronts is the most important task of the regularity estimator since we apply FCT to prevent over- and undershoots only on these non-smooth elements. Note that the CG1-DG2 method may produce oscillations on these non-smooth elements if the gradients are too steep. This can be observed for example in Section 6.5. In regions where the solution is constant we usually do not need to apply the FCT scheme since the solution is smooth in the analytical sense. However, since we have seen in the numerical studies in Chapter 6 that the CG method can also produce oscillations in regions where the solution is constant, we prefer to use a stable method in these elements. Thereby, it is safe to either use FCT or the CG1-DG2 method.

From a computational point of view it is less memory consuming to use linear finite elements than the CG1-DG2 method. Also from an analytical point of view it is reasonable to use linear elements since a quadratic approximation of a constant function does not lead to a higher accuracy. For that reason if ε is small enough, the choice of ε in (9.3.5) should only influence the number of higher order elements and therefore the computational costs but not the solution. If ε is too large, also smooth parts are regarded as non-smooth. If FCT is applied in these parts, peak clipping can occur. This is a well known phenomenon occurring at smooth extrema which are then smoothed out by the FCT scheme [88].

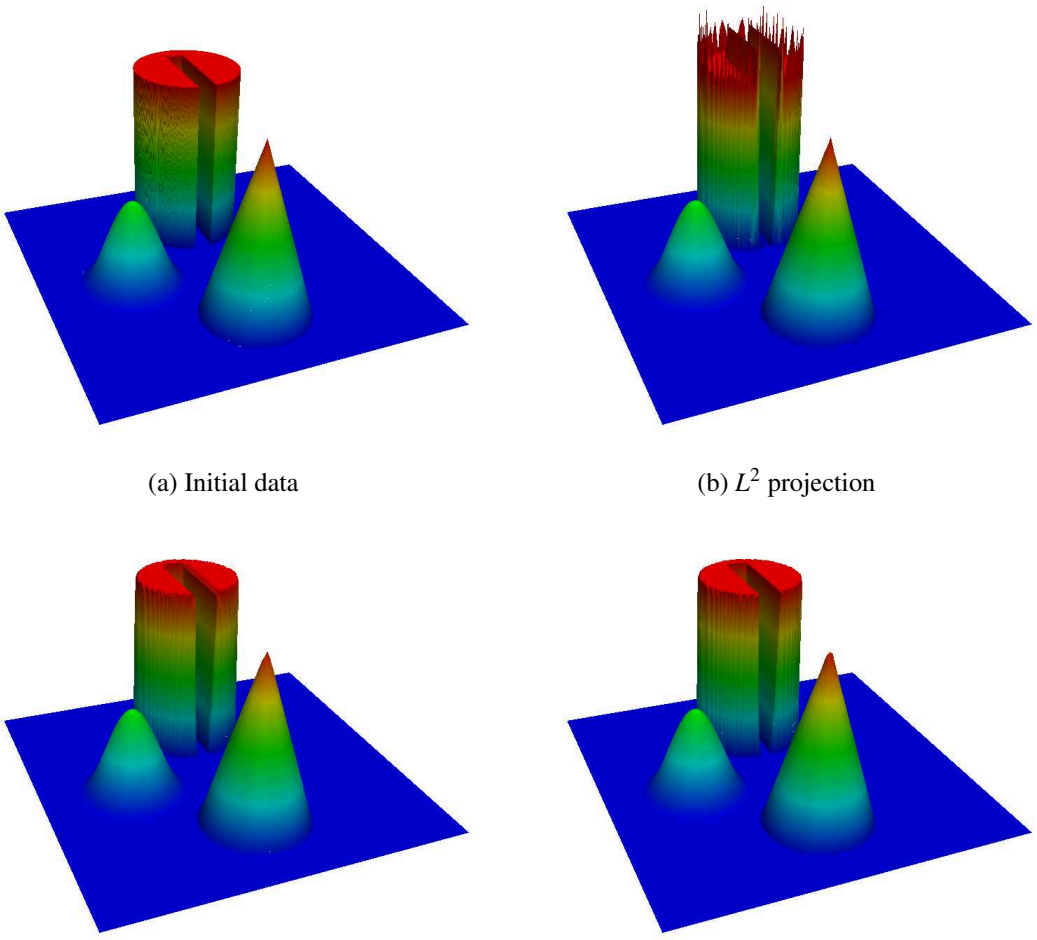
10.2. Constrained L^2 projection

In the following we will compare the L^2 projection with its constrained version.

We mention that the solution on quadrilateral meshes and on triangular meshes look very similar so that only the results for the quadrilateral meshes are shown in the following.

10.2.1. Solid body rotation: Projection of the initial data

In Fig. 10.3 the initial data of the solid body rotation problem and its projections are displayed. It can be seen, that the hump is resolved very well by all projections. At the boundaries of the cylinder the L^2 projection produces oscillations whereas the lumped-mass and the constrained L^2 projection does not. This can also be seen in Fig. 10.4 where the different projections versus the initial data are plotted at $x = 0.5$ and $y = 0.75$. The peak of the cone is a little bit smeared by the lumped-mass and the constrained projection whereas it is very well resolved by the L^2 projection. The difference between the constrained L^2 and the lumped-mass projection is very small, e.g., the areas where the objects touch the $z = 0$ line as well as the inner boundary of the cylinder are a little bit better resolved by the constrained version. However, during long-time computations these small differences can accumulate such that the solution would be more smeared out if using the lumped-mass projection. For that reason the constrained L^2 projection is preferred.



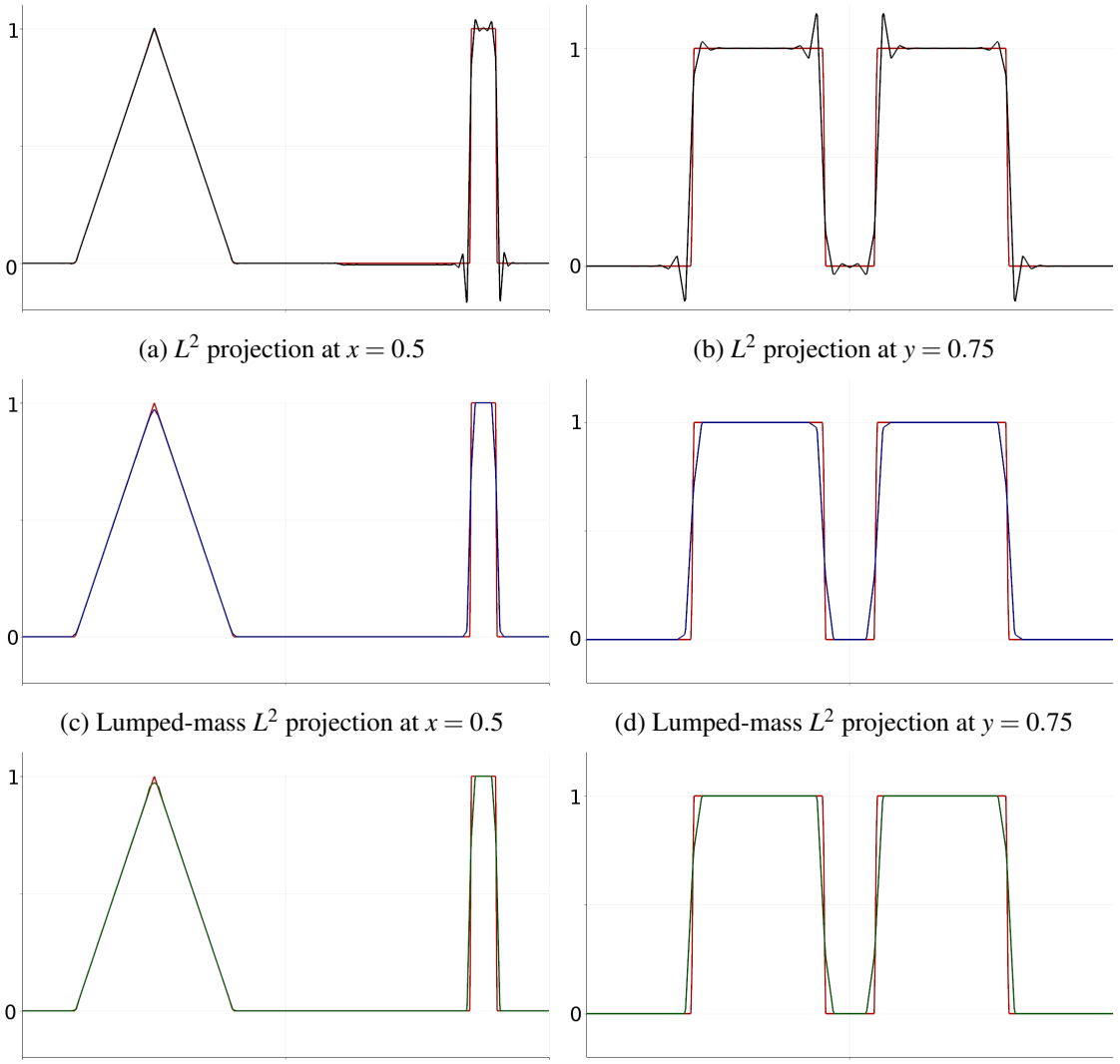
(a) Initial data

(b) L^2 projection

(c) Lumped-mass L^2 projection

(d) Constrained L^2 projection

Figure 10.3: Initial data and its projections for the solid body rotation problem



(a) L^2 projection at $x = 0.5$

(b) L^2 projection at $y = 0.75$

(c) Lumped-mass L^2 projection at $x = 0.5$

(d) Lumped-mass L^2 projection at $y = 0.75$

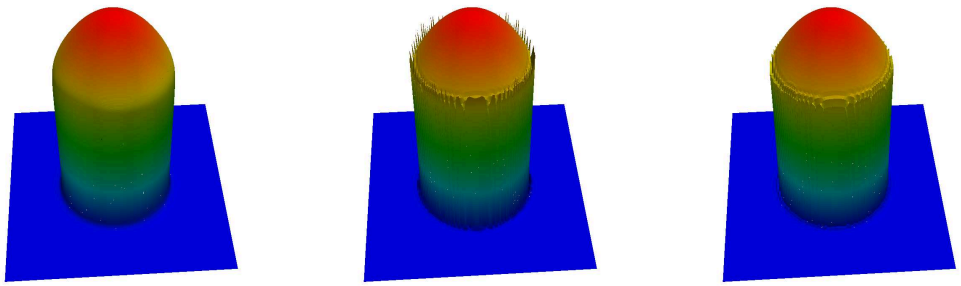
(e) Constrained projection at $x = 0.5$

(f) Constrained projection at $y = 0.75$

Figure 10.4: Initial data (red) and its projections for different cuts

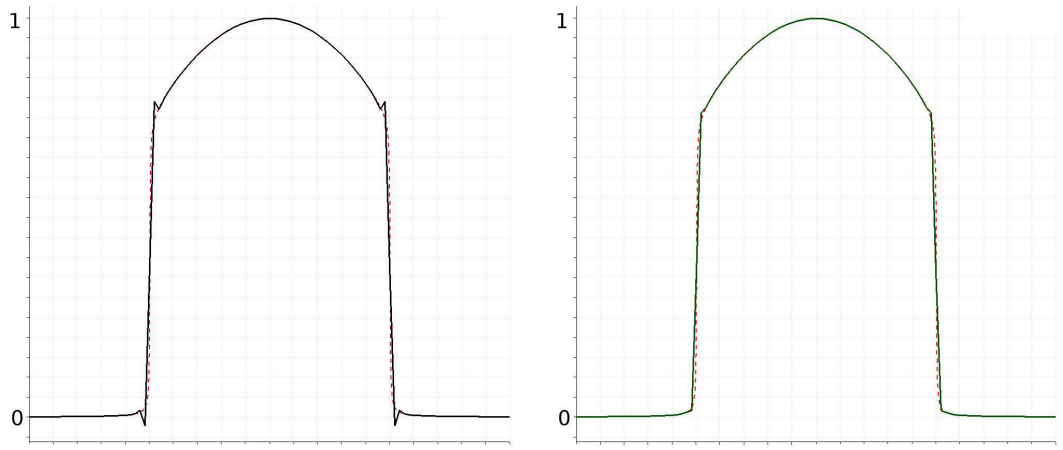
10.2.2. Hump changing height: Projection of the exact solution at $t = 0.5$

In the next example we consider the exact solution at $t = 0.5$ of the hump changing height problem from Section 6.8, see Fig. 10.5a. In Figures 10.5b and 10.5c the L^2 and the constrained L^2 projection are displayed. It can be seen that the L^2 projection produces oscillations at the steep fronts of the hump which is shown in more detail in Fig. 10.6a. The constrained L^2 projection smooths these oscillations so that only a small spike is left (see Figures 10.5c and 10.6b).



(a) Exact solution (b) L^2 projection (c) Constrained L^2 projection

Figure 10.5: Projection of the exact solution of the hump changing height problem at $t = 0.5$



(a) L^2 projection (b) Constrained L^2 projection

Figure 10.6: Hump changing height problem at $t = 0.5$ for $x = 0$:
exact solution (red) and its projection

In summary, these examples show the advantage of the constrained L^2 projection when it comes to dealing with discontinuous data.

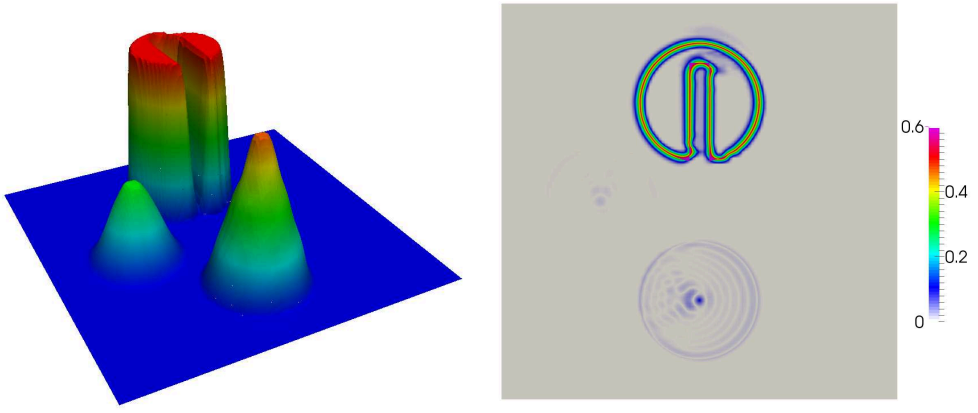
In the following examples we use the constrained L^2 projection whenever a projection onto linear elements is needed. For quadratic elements we use the standard L^2 projection.

10.3. Solid body rotation problem: FCT, h- and hp-adaptivity

In this section we present numerical results for the solid body rotation problem from Section 6.5, which describes a counterclockwise rotation of a slotted cylinder, a cone and a hump about the center of the domain. The initial data can be seen in Fig. 10.3a. We will start with the solution on uniform meshes using the FCT algorithm presented in section 9.1. Then we show the results using h-adaptivity in combination with the FCT-algorithm. At the end we present the solution derived by Scheme 9.11 which indicates the advantage of that strategy over h-refinement.

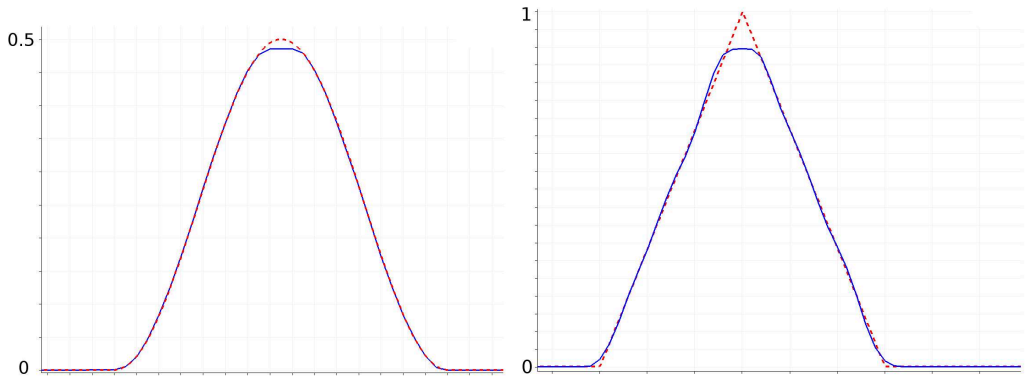
10.3.1. Flux-corrected transport on uniform meshes

At first, we will present the numerical results calculated on uniform meshes. In Figures 10.7 and 10.9 the numerical solutions and the corresponding error distributions on quadrilateral and triangular meshes are displayed. The corresponding spaces have 16381 DOFs. It can be seen that no oscillations occur and the solution has no negative values and no values larger than the maximum value 1. Note that large errors at the boundary of the cylinder occur since the solution in the corresponding elements is continuous unlike the discontinuous exact solution. In comparison to the error profiles of the DG2 and CG1-DG2 solution (see Fig. 6.16) the errors at the cylinder are more concentrated on the boundary. At the top of the hump and the cone peak clipping occurs. In Fig. 10.8 this phenomenon is shown in more detail. It can be seen that the top of hump and the cone, respectively, looks like they have shrinked. Remember that in the DG2 and CG1-DG2 case the hump was resolved very well, whereas the top of the cone was also smeared out.



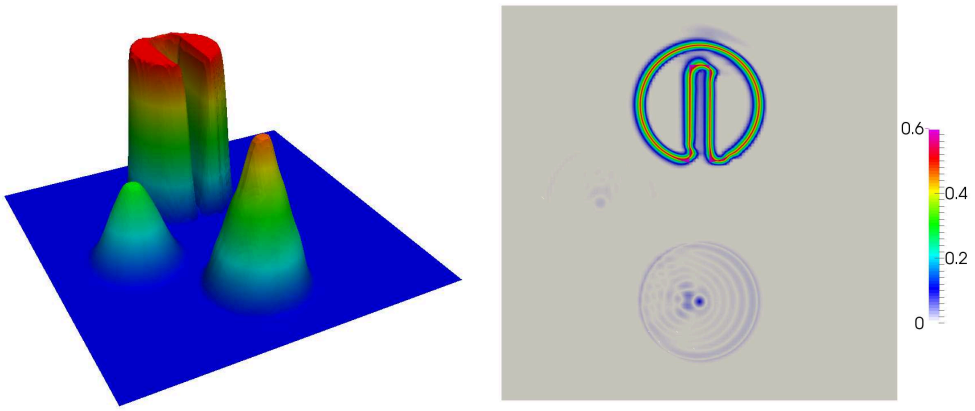
(a) FCT solution (b) FCT error distribution

Figure 10.7: Solid body rotation problem: solution and error distribution at $t = 2\pi$ on quadrilateral mesh



(a) Hump at $y = 0.5$ (b) Cone at $x = 0.25$

Figure 10.8: Peak clipping phenomenon: comparison of exact solution (red) and FCT solution (blue)



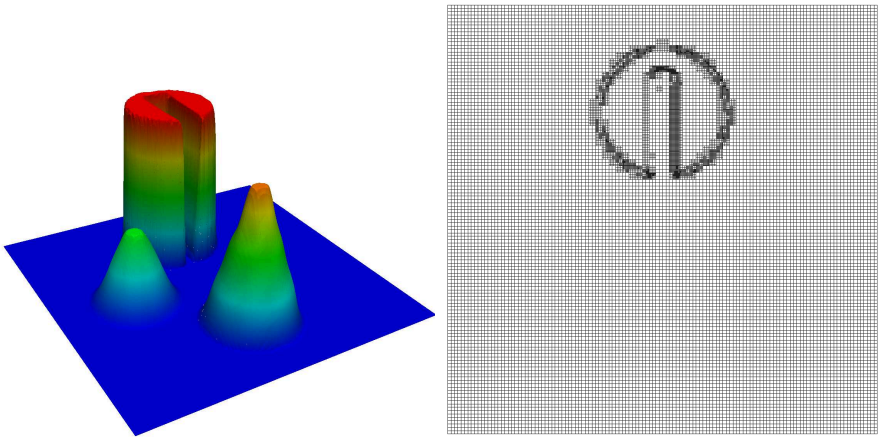
(a) FCT solution (b) FCT error distribution

Figure 10.9: Solid body rotation problem: solution and error distribution at $t = 2\pi$ on triangular mesh

10.3.2. Reference solution approach: h-adapativity

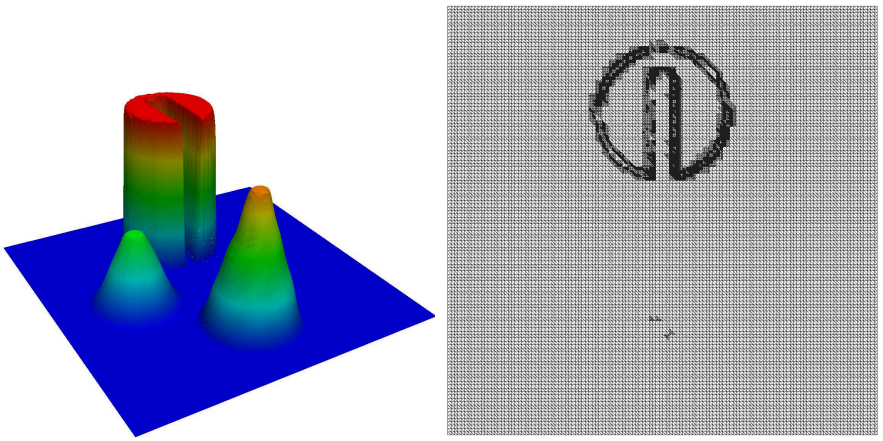
Next, we will show the results for h-adaptivity. In Figures 10.10 and 10.11 the reference solutions and the corresponding meshes for linear finite elements with FCT are displayed. The element size on the reference meshes varies between $h = \frac{1}{64}$ and $h = \frac{1}{512}$. The spaces have ≈ 24000 DOFs. In both cases the top of the hump and the cone have shrinked due to peak clipping.

In Fig. 10.12 the h-adapted solution on the quadrilateral mesh is compared with the exact solution and the solution on two different uniform meshes. The first uniform mesh corresponds to the initial coarse mesh, the other to the initial reference mesh. We see that the h-adapted solution resolves the cylinder best. At the cone peak clipping occurs in all cases. However, the finest uniform mesh solution has least shrinked. The reason that in this case the h-adapted solution is worse is due to the fact that in each adaptivity step of the h-adapted algorithm the reference solution needs to be projected onto the new space. As we have seen in section 10.2 the constrained L^2 projection smears the top of the cone. This inserts additional errors into the solution.



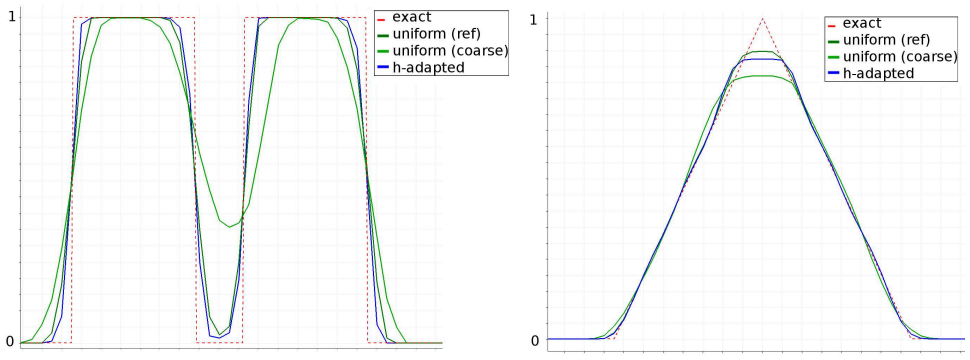
(a) Reference solution (b) Reference mesh

Figure 10.10: H-adaptivity for the solid body rotation problem at $t = 2\pi$ on quadrilateral mesh



(a) Reference solution (b) Reference mesh

Figure 10.11: H-adaptivity for the solid body rotation problem at $t = 2\pi$ on triangular mesh

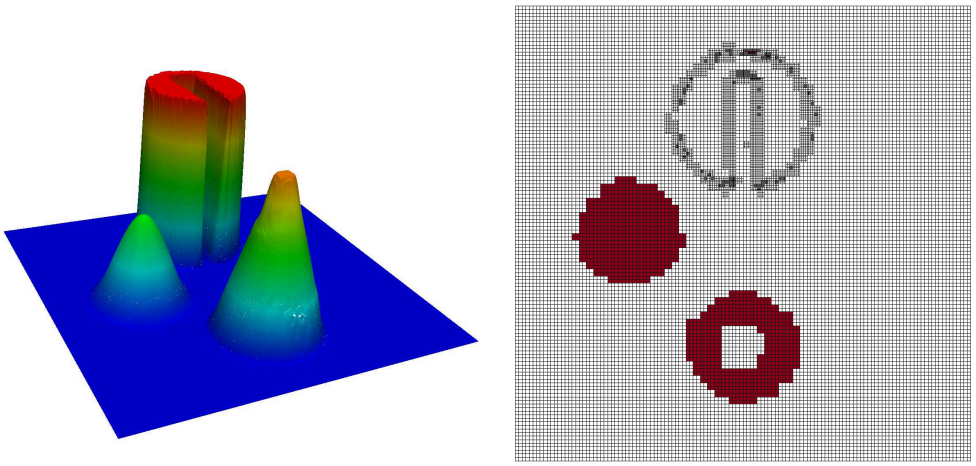


(a) Cylinder at $y = 0.75$ (b) Cone at $y = 0.25$

Figure 10.12: Comparison of uniform FCT-solution and h-adapted FCT solution

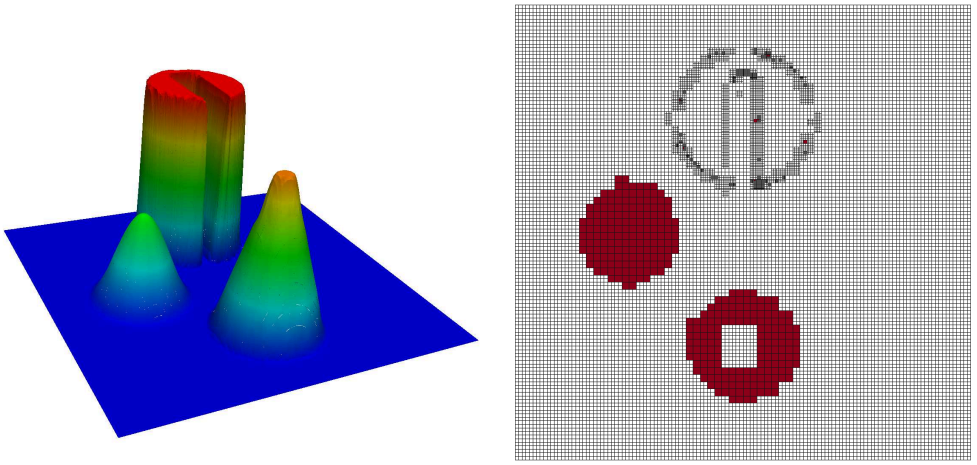
10.3.3. Reference solution approach: hp-adaptivity

At last we present the hp-adaptivity results. We used triangular and quadrilateral meshes where in the later case we distinguished between serendipity and Q2 elements in the case of quadratic elements. In all cases the reference spaces have ≈ 20000 DOFs. In Fig. 10.13 - 10.15 the reference solution at $t = 2\pi$ is shown, where quadratic elements are marked as red. We can see that the regularity estimator handled the top of the cone and the cylinder as non-smooth so that FCT was used in these elements. Here, the top of the cone is smeared out as in the h-adaptive and the uniform case. The hump was marked as smooth so that the CG1-DG2 method was used in these elements and no peak clipping occurred. This can also be seen in more detail in Fig. 10.16 where the resolution of the hump for h- and hp-adaptivity is compared. The cylinder and the cone are resolved nearly identically to the h-adaptive case.



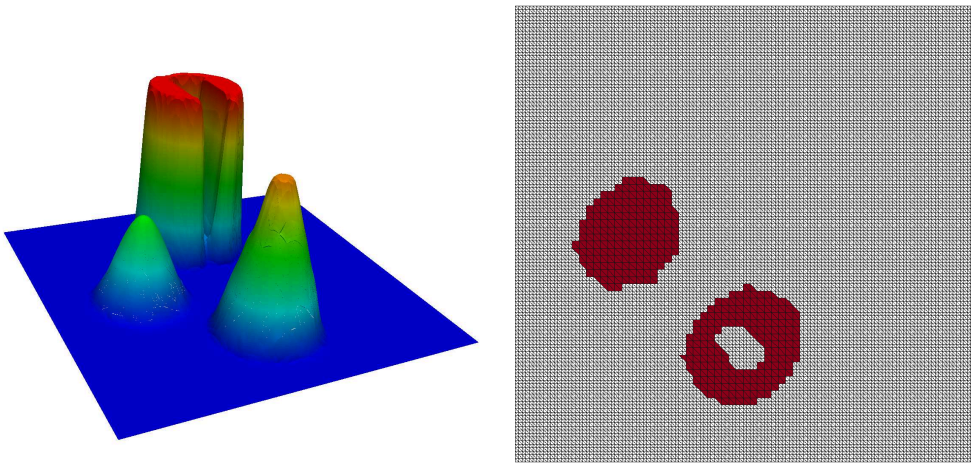
(a) Reference solution at $t = 2\pi$ (b) Reference space/mesh at $t = 2\pi$

Figure 10.13: Hp-adaptivity for the solid body rotation problem using Q2-elements in higher-order regions



(a) Reference solution at $t = 2\pi$ (b) Reference space/mesh at $t = 2\pi$

Figure 10.14: Hp-adaptivity for the solid body rotation problem using serendipity elements in higher-order regions



(a) Reference solution at $t = 2\pi$ (b) Reference space/mesh at $t = 2\pi$

Figure 10.15: Hp-adaptivity for the solid body rotation problem on triangular mesh

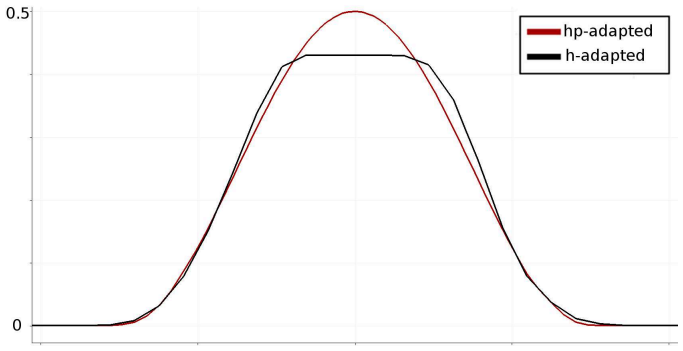


Figure 10.16: Comparison between the h- and hp-adapted hump of the solid body rotation problem at $t = 2\pi$ and $y = 0.5$ (serendipity version)

In the following examples, we will use serendipity elements in the case of higher-order elements on quadrilateral meshes. At first we consider an example which has a smooth solution.

10.4. Reference solution approach: time-dependent advection equation with constant velocity field

Here we adapt the stationary problem from section 6.1 and consider its time-dependent version

$$u_t + \nabla \cdot (\boldsymbol{\beta} u) = 0 \qquad \text{in } \Omega, \tag{10.4.1}$$

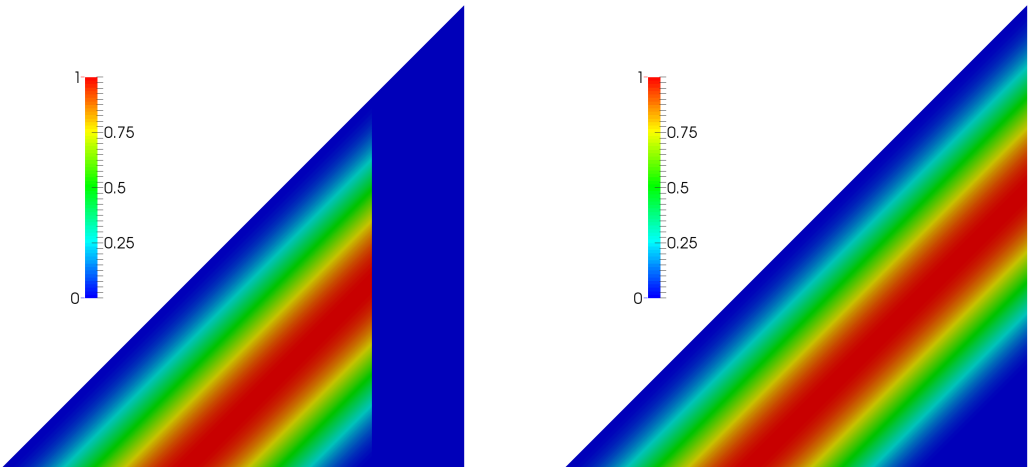
with the constant velocity field

$$\boldsymbol{\beta}(x,y) = (1,1). \tag{10.4.2}$$

The initial data and the exact solution are shown in Fig. 10.17. In the case of triangular elements we will use a triangle domain defined by the points $(0,0)$, $(1,0)$, $(1,1)$, for a quadrilateral mesh we use $\Omega = (0,1)^2$. For a proper comparison of the solution, we stop the computation at $t = 0.3$. At this point the exact solution would already have reached steady-state.

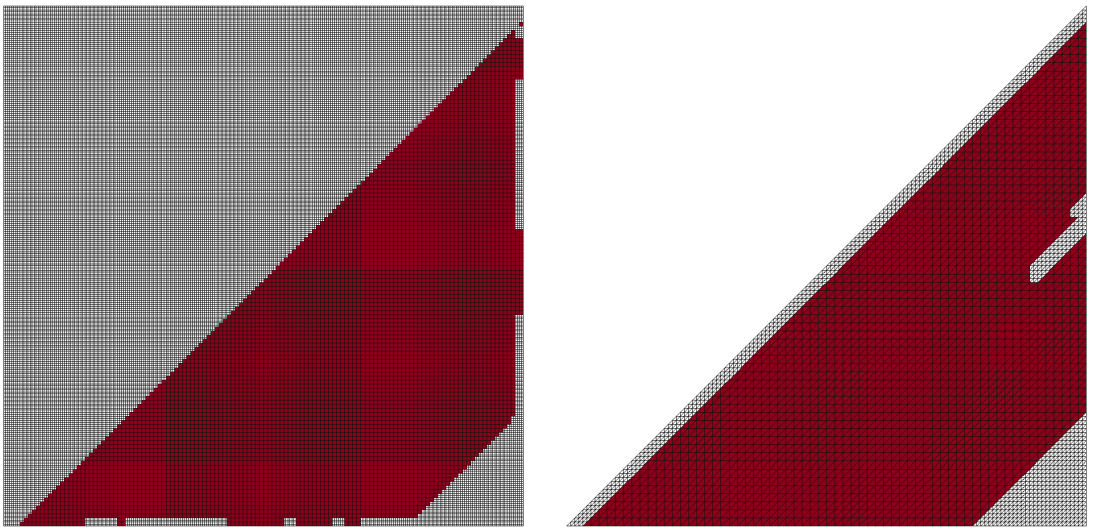
In Fig. 10.18 the hp-adapted meshes are displayed. We can see that the regularity estimator marked most of the elements, where the solution is not constantly zero, as smooth, but has some problems with elements near the boundary which are marked as non-smooth.

In Fig. 10.19 the L^2-errors versus number of DOFs are plotted. In both cases and for almost all numbers of DOFs the hp-solution exhibits smaller errors than the h-solution. One reason why we do not see a convergence rate of exponential order is that we restricted the polynomial order to $1 \leq p \leq 2$.



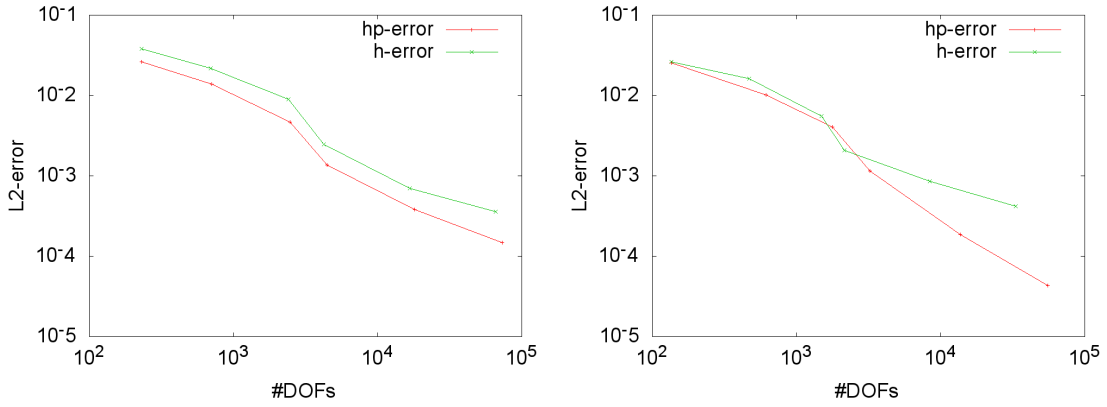
(a) Initial data (b) Exact solution (steady-state)

Figure 10.17: Initial data and the exact solution



(a) Quadrilateral mesh/space (b) Triangular mesh/space

Figure 10.18: Hp-adapted mesh/space: quadratic elements (red) and linear elements (white)



(a) Quadrilateral mesh (b) Triangular mesh

Figure 10.19: L^2-error vs. number of DOFs

10.5. Reference solution approach: time-dependent advection equation with rotating velocity field

We consider now the time-dependent version of the stationary problem from section 6.3. The exact solution on $\Omega = (0,1)^2$ can be seen in Fig. 10.20. As in the previous example we use as initial data a part of the exact solution, that means we take the exact solution for $x < 0.7$ and for $x \geq 0.7$ we set zero. In Fig. 10.21 and 10.22 the solutions for h- and hp-adaptivity are shown. It can be seen that the regularity estimator marked most elements of the solution as smooth but has again problems with elements near the boundary. The hp-space has ≈ 4600 DOFs, whereas the h-adapted has ≈ 4700. The absolute error in the L^2-norm for the hp-solution is $3.7349e-03$ and for the h-solution $3.8294e-03$. In Fig. 10.23 the adapted solutions are compared with the exact solution. On the first picture we see that the hp-solution is less smeared than the h-solution. In the h-adapted case peak clipping is clearly visible. At the second picture we see that the hp-adapted solution resolves the exact solution very well whereas the h-solution is smeared.

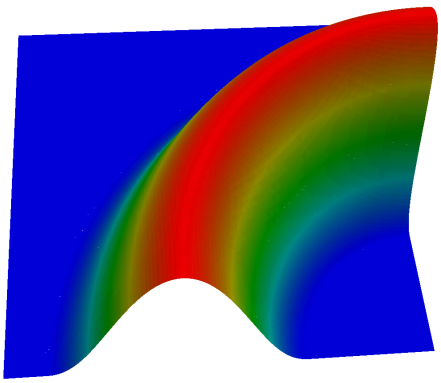
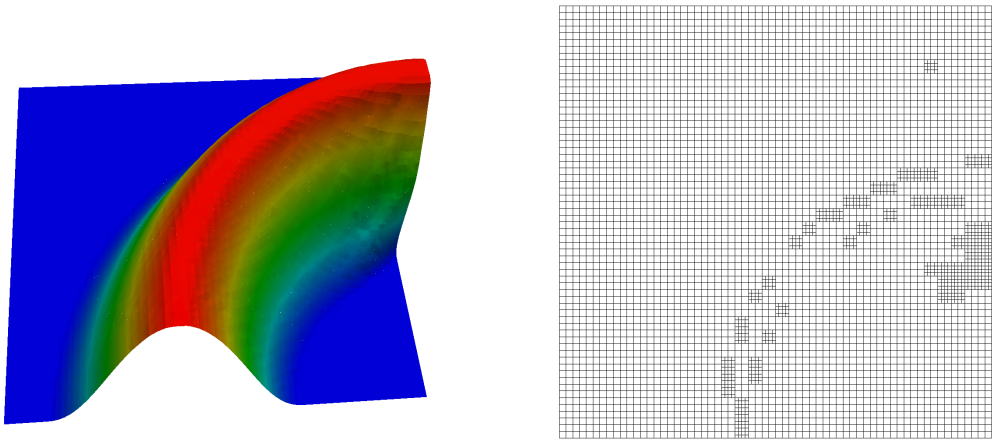
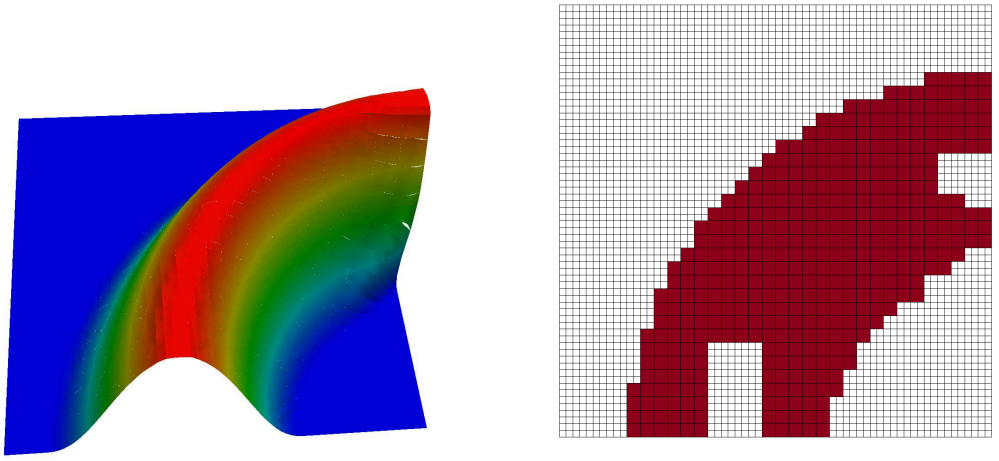


Figure 10.20: Exact solution



(a) H-adapted solution (b) Reference mesh

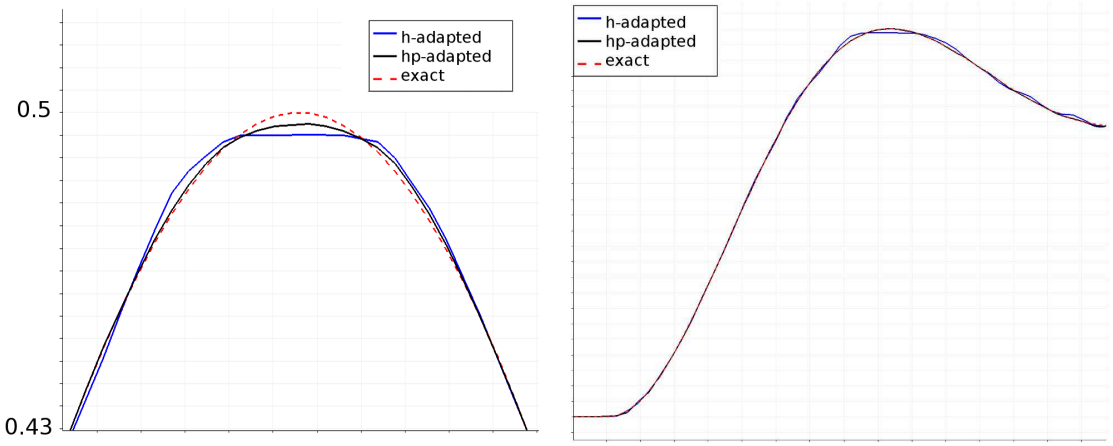
Figure 10.21: H-adaptivity for advection equation



(a) Hp-adapted solution

(b) Reference mesh/space (red for CG1-DG2 element)

Figure 10.22: Hp-adaptivity for advection equation



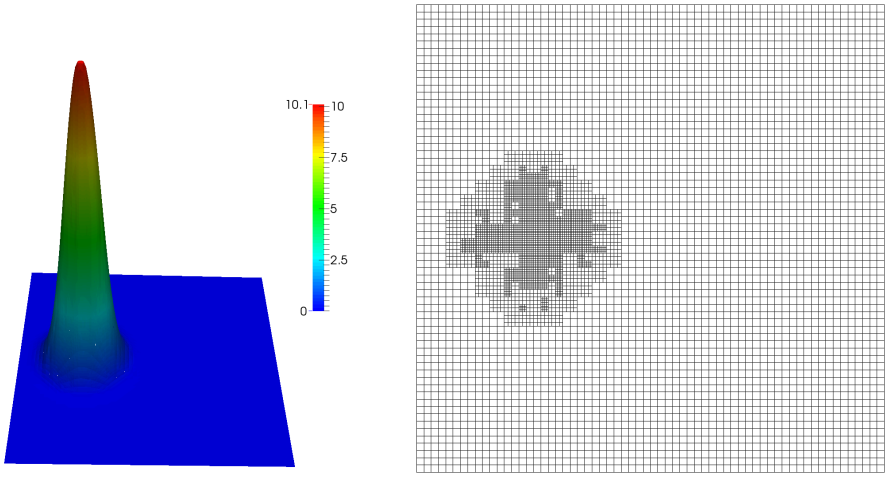
(a) Cut at the diagonal $y = 1 - x$ (magnification)

(b) Cut at $y = 0.5$

Figure 10.23: Comparison between exact and approximated solutions

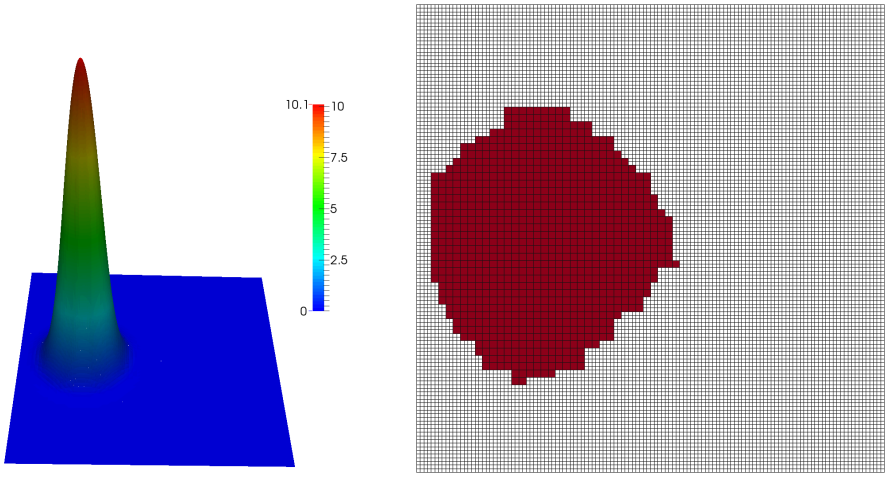
10.6. Reference solution approach: time-dependent advection-diffusion equation

In this example we present a problem with a diffusive term. Therefore, we consider the time-dependent advection-diffusion equation from section 6.7. We used the Baumann-Oden method and serendipity elements in the case of higher-order elements. In Fig. 10.24 we see the reference solution and the corresponding mesh for h-adaptivity. The hump was marked for refinement. In Fig. 10.25 we see the hp-adapted solution and the reference space/mesh. The hump was marked as smooth so that it was computed by the CG1-DG2 method. In Fig. 10.26 we see the direct comparison of the h- and hp-adapted solution. As we have already seen in the previous examples peak clipping occurs in the h-adapted case, so that the hp-solution is more accurate.



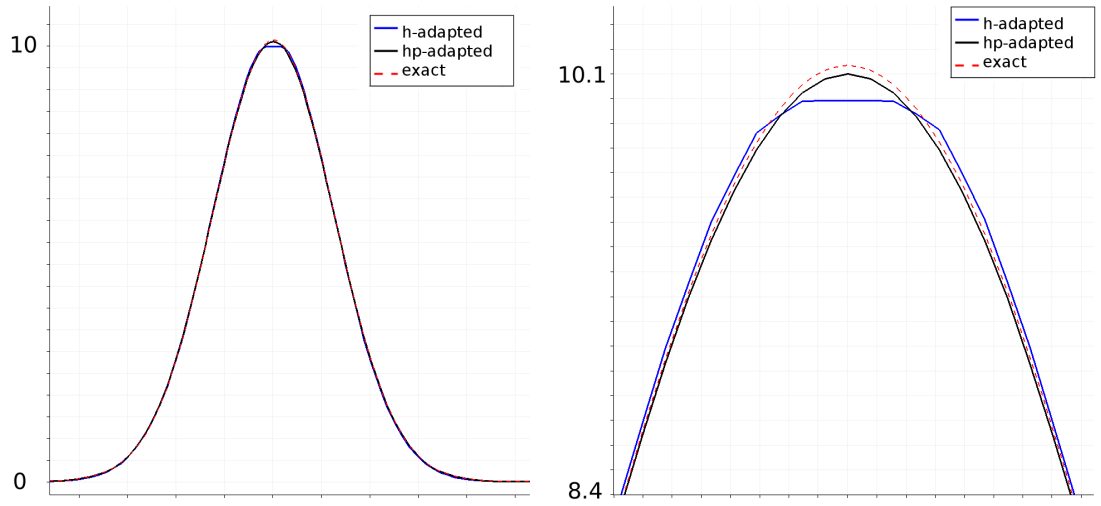
(a) Reference solution (b) Reference mesh

Figure 10.24: H-adaptivity for the time-dependent advection diffusion equation



(a) Reference solution (b) Reference space/mesh

Figure 10.25: Hp-adaptivity for the time-dependent advection diffusion equation



(a) Hump (b) Enlarged peak

Figure 10.26: Comparison between exact and approximated solutions

In summary, we have seen that the advantage of the presented hp-adaptive strategy over the pure h-adaptivity is that in smooth elements the solution is much better resolved since no peak clipping induced by FCT occurs. Therefore, the presented hp-adaptive algorithm provides a great benefit if one has to deal with problems where the solution is at least partly smooth.

11

Summary and outlook

11.1. Summary

The first part of this thesis was concerned with the introduction of the CG1-DG2 method. This method combines the continuous Galerkin method with the discontinuous Galerkin method in the context of finite elements. Hereby, the linear continuous finite element space is enriched with discontinuous quadratic basis functions. For the analysis we considered a scalar advection equation and Poisson's equation.

In the case of advection problems, the discontinuous fluxes have been approximated by upwind fluxes. While the standard continuous Galerkin method has only L^2-stability, the CG1-DG2 method for triangular meshes is stable with respect to an augmented DG norm giving additional control over streamline derivatives. A priori error estimates showed that the method delivers the same convergence rate as the DG method.

For Poisson's equation different strategies from the DG method have been adopted to approximate the numerical fluxes: the symmetric and non-symmetric interior penalty method as well as the Baumann-Oden method. Since the CG1-DG2 space is a subspace of the quadratic DG space, most analytical results like boundedness and coercivity of the bilinear form have been directly transferred from the DG space to the CG1-DG2 space. Furthermore, a priori error estimates for the DG method hold for the CG1-DG2 approximation as well.

In numerical studies the analytically expected orders of convergence for advection and diffusion problems have been confirmed for different test problems. In the case of quadrilateral meshes and advection problems, the use of serendipity elements lead to convergence behavior similar to that obtained on triangular meshes whereas the use of standard Q2-elements can decrease convergence rates. We have also seen that the CG1-DG2 method is more stable than the CG2 method in the sense that smooth solutions do not exhibit oscillations like in the CG2 case. In the context of discontinuous initial data, the numerical CG1-DG2 solutions showed similar behavior as the DG solutions where oscillations occurred only locally near steep gradients.

The CG1-DG2 method was then extended to solve the Euler equations. The treatment of boundary conditions was adopted from [35], where boundary fluxes have been approximated using the flux formula of Roe. The discontinuous numerical fluxes have been approximated by Lax-Friedrichs fluxes. For the discretization in time we used the backward Euler scheme. Hereby, the non-linearity was linearized using the Taylor series expansion presented in [27]. The numerical

examples showed that the triangular and the serendipity CG1-DG2 method applied to systems of conservations laws give results similar to those obtained by the DG method in the case of subsonic and supersonic flow regimes. In the context of a shock tube problem, we have compared the results of the serendipity CG1-DG2 method with those obtained by the CG2 method. We have seen that the CG1-DG2 solution exhibits oscillations only near the discontinuities, whereas the CG2 solution has undershoots and overshoots in a larger vicinity and with a higher amplitude.

In summary, the analytical and numerical results showed the advantages of the CG1-DG2 method over the CG2 method for triangular meshes and quadrilateral meshes combined with serendipity elements. The numerical experiments indicated, that the use of serendipity elements instead of Q2-elements improves the accuracy of the method. The advantage of the CG1-DG2 method over the DG2 method are the lower computational costs due to fewer degrees of freedom.

The second part of this thesis introduced an hp-adaptive framework for convection-dominated problems. The idea of the presented algorithm was to divide the mesh in smooth and non-smooth parts, where the smoothness refers to the regularity of the approximated solution. Hereby, a parameter-free regularity estimator was used to determine the smoothness of a function and its gradient by comparing those with reconstructed approximations. In smooth elements we applied the CG1-DG2 method, which implies a quadratic approximation of the solution in those elements. Even if the use of the CG method with quadratic or higher polynomial degrees would be possible (see [13]), the CG1-DG2 method is preferred since it can control streamline derivatives leading to an improved stability. In non-smooth elements we used h-adaptivity in the sense of the reference solution approach and applied the FCT scheme for stabilization. Constant functions have been regarded as non-smooth, since a higher-polynomial degree would not lead to higher approximation accuracy in this case. For the projection of initial and previous time-step data onto the current space, we used the L^2 projection in smooth elements and a constrained version in non-smooth elements. The latter is based on the FCT scheme and prevents the appearance of oscillations near steep gradients. The presented framework operates in two steps: First p-adaptivity as indicated by the regularity estimator is applied, where p is restricted to $p \leq 2$. Then h-adaptivity realized by the reference solution approach is applied in a second step.

Numerical experiments have been performed for advection and advection-diffusion equations. Those showed the advantage of the hp-adaptive algorithm over pure h-refinement in the context of FCT schemes. The hp-solution benefits from a higher accuracy in smooth elements, since FCT is not applied in those and therefore, no peak-clipping occurs. Hereby, the good performance of the regularity estimator was essential to identify smooth elements.

11.2. Outlook

In future work, we will extend the analysis for the scalar advection equation to quadrilateral meshes consisting of serendipity elements. Even if the analysis for triangles cannot directly be transferred to the serendipity case, numerical results indicated stability properties and convergence rates similar to those on triangular meshes. Also the use of higher order polynomials will be considered. Hereby, we have to analyze two different possibilities: The first would be adding discontinuous cubic basis function to the CG1-DG2 space, the other enriching the quadratic continuous space (CG2) with discontinuous cubic basis functions. These approaches can also be extended to higher order polynomials. If the so derived methods are stable, they could be used to improve the hp-adaptive algorithm in the sense that also higher polynomials, i.e., $p > 2$, are considered. In this case, we will also have to improve the criteria for regularity in order to decide which polynomial degree should be used in an element. Hereby, p-enrichment could be done adaptively in the sense that we increase the polynomial degree p only once per adaptivity step and

if higher-order $(p+1)$ derivatives are regarded as smooth. However, since this may significantly increase the costs, other p-adaptivity criteria should also be considered. For example, the presented algorithm could be extended in such a way, that the reference solution approach is not only used for h-refinement, but also for p-enrichment in smooth elements. The regularity estimator would then handle the identification of elements which can be considered for p-adaptivity, and the reference solution approach would decide which polynomial degree is set.

The CG1-DG2 method is currently implemented only in 2D. However, an extension to 3D seems possible. Hereby, different strategies should be tested. One possibility is the use of continuous vertex basis functions and discontinuous edge and face functions. Alternatively, vertex and edge functions could be continuous and only face functions discontinuous. The first approach would lead to solutions which are continuous at the vertices but discontinuous across edges and faces, whereas the other approach yields solutions which are continuous at vertices and edges and discontinuous across faces. Especially the latter would result in significantly fewer degrees of freedom compared to the DG method.

Furthermore, we will investigate how discontinuous data can be captured more accurately. A first step will be to analyze different numerical fluxes and see if they can improve the stability and convergence properties of the method. The next step is to check if stabilization techniques for the CG and DG method can be transferred to the CG1-DG2 method. One possibility is to apply shock capturing edge stabilization techniques like those presented in [19] for convection-diffusion-reaction problems. These techniques penalize gradient jumps across edges. An extension to the CG1-DG2 method seems to be possible. DG techniques are not easy to transfer. For example, slope limiting techniques as presented in [60] benefit from the property of local basis functions, i.e., basis functions which are non-zero only on a single element. Since the CG1-DG2 method has also continuous basis functions, which are globally defined, i.e., those are non-zero on element patches, we cannot extend the DG techniques in a straight-forward way. A combination of continuous and discontinuous techniques could be considered, such that the continuous part is stabilized by continuous techniques and the discontinuous part by discontinuous techniques.

In future work, we will also extend our hp-adaptive framework to solve systems of conservation laws. Since the reference solution approach is relatively expensive (all computations are done on the reference space), other h-refinement techniques should be considered for systems. An alternative approach was already presented in [13] in the context of scalar equations, where the Z^2 error estimator [89] was used to identify elements for h-refinement. Also coarsening of elements should be considered. The final and most challenging goal will then be the extension of the hp-adaptive framework to solve two-phase flow problems.

A

Appendix

Let us recall some definitions.

1.1. Triangulation

Definition A.1 (Triangulation,[54])**:** A triangulation $\mathcal{T}_h$ of $\Omega \subset \mathbb{R}^{\dim}$ consists of a finite number of subsets K of Ω with the following properties:

1. Every $K \in \mathcal{T}_h$ is closed.

2. For every $K \in \mathcal{T}_h$ its nonempty interior $int(K)$ is a Lipschitz domain.

3. $\bar{\Omega} = \cup_{K \in \mathcal{T}_h} K$.

4. For K_1, K_2 of $\mathcal{T}_h$, $K_1 \neq K_2 : int(K_1) \cap int(K_2) = \emptyset$.

Definition A.2 (Conforming and regular triangulations,[54])**:** A triangulation $\mathcal{T}_h$ is called conforming if

1. Every side S of some $K \in \mathcal{T}_h$ is either a subset of the boundary Γ or identical to a side of another $\tilde{K} \in \mathcal{T}_h$.

2. The boundary sets $\Gamma^{\mathrm{N}}, \Gamma^{\mathrm{D}}, \Gamma_+$ and Γ_- admit a decomposition into sides of elements $K \in \mathcal{T}_h$.

A family of triangulation $(\mathcal{T}_h)_h$ is called regular if there exists $\sigma > 0$ such that for all $h_K > 0$ and all $K \in \mathcal{T}_h$

$$\rho_K \geq \sigma h_K, \tag{A.1}$$

where $\rho_K := \sup\{diam(B) | B \text{ is a ball in } \mathbb{R}^{\dim} \text{ with } B \subset K\}$.

1.2. Estimates and inequalities

Theorem A.3 (Hölder's inequality)**:** Let $p, q \in [1,\infty]$ with $\frac{1}{p}+\frac{1}{q}=1$. Then

$$\|fg\|_{L^1(\Omega)} = \int_\Omega fg\,\mathrm{d}\mathbf{x} \leq \|f\|_{L^p(\Omega)}\|g\|_{L^q(\Omega)}. \tag{A.2}$$

In the case of $p=q=2$ it is also called the Cauchy-Schwarz inequality.

Theorem A.4 (Discrete Cauchy-Schwarz inequality)**:** For $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ we have

$$\left(\sum_{i=1}^N x_i y_i\right)^2 \leq \left(\sum_{i=1}^N x_i^2\right)\left(\sum_{i=1}^N y_i^2\right). \tag{A.3}$$

Theorem A.5 (Young's inequality)**:** Let $p, q > 0$ with $\frac{1}{p}+\frac{1}{q}=1$ and $a, b \geq 0$. Then

$$ab \leq \frac{a^p}{p} + \frac{b^q}{q}. \tag{A.4}$$

Furthermore, for $\varepsilon > 0$ the following inequality holds:

$$ab \leq \frac{a^2}{2\varepsilon} + \frac{\varepsilon b^2}{2}. \tag{A.5}$$

In the following, let $\mathcal{T}_h$ be a regular conforming triangulation.

Lemma A.6 (Multiplicative trace inequality, [75])**:** Let $K \in \mathcal{T}_h$ and $S \subset S_K \subset \mathcal{S}_h$. Then, for all $v \in H^1(K)$ there exists a mesh-independent constant $C > 0$ such that

$$\|v\|^2_{L^2(S)} \leq C\left(h_K^{-1}\|v\|^2_{L^2(K)} + \|v\|_{L^2(K)}\|\nabla v\|_{L^2(K)}\right). \tag{A.6}$$

Lemma A.7 (Inverse estimate, [75])**:** Let $K \in \mathcal{T}_h$ and $S \subset S_K \subset \mathcal{S}_h$. Then, for all $v \in P^k(K)$ there exists a constant $C > 0$ such that

$$\|\nabla v\|_{L^2(K)} \leq C_I \frac{k^2}{h_K}\|v\|_{L^2(K)}. \tag{A.7}$$

Theorem A.8 (Trace inequalities, [2])**:** For each element $K \in \mathcal{T}_h$ and corresponding edge $S \in \mathcal{S}_h$ the following estimates hold

$$\|v\|^2_{L^2(S)} \leq C\left(\frac{1}{h_S}\|v\|^2_{L^2(K)} + h_S\,|v|^2_{H^1(K)}\right) \quad \forall v \in H^1(K), \tag{A.8}$$

$$\|\nabla v \cdot n\|^2_{L^2(S)} \leq C\left(\frac{1}{h_S}\,|v|^2_{H^1(K)} + h_S\,|v|^2_{H^2(K)}\right) \quad \forall v \in H^2(K). \tag{A.9}$$

1.3. Projection

Definition A.9 (Orthogonal projection)**:** Let V be a Hilbert space with the inner product $(\cdot,\cdot)$ and U be a closed subspace of V. Then there exists an operator $P_U : V \to V$ with

1. $\operatorname{im} P_U = U$,
2. $\ker P_U = U^{\perp}$.

P_U is a continuous linear operator with following properties:

1. $P_U^2 = P_U$,
2. $(P_U v, v) \geq 0, \forall\, v \in V$,
3. $\|P_U v - v\| = \inf_{u\in U} \|u - v\|$,
4. $\|P_U\| = 1$, if $U \neq \{0\}$,
5. $\|I - P_U\| = 1$, if $U \neq V$,
6. $\|Pv\| \leq \|v\|, \forall\, v \in V$.

1.4. Clément operator

For the interpolation of non-smooth functions we make use of the Clément operator.

Definition A.10 (Clément operator, [29])**:** Let $a_j, j = 1,...,N$ be Lagrange nodes corresponding to the global basis functions ϕ_j in V_h^k. For each node a_i we define the macroelement A_i which consists of all elements containing node a_i. Since there are only a finite number n_{rf} of configurations possible for the macroelement, we denote by $\{\widehat{A}_n\}_{1\leq n\leq n_{rf}}$ the list of reference configurations and define the mapping $j : \{1,...,N\} \to \{1,...,n_{rf}\}$ which associates reference configuration $\widehat{A}_{j(i)}$ with the macroelement A_i. Let us now define the C^0-diffeomorphism $F_{A_i} : \widehat{A}_{j(i)} \to A_i$ such that $F_{A_i|\hat{K}}$ is affine $\forall \hat{K} \in \widehat{A}_{j(i)}$. Then we define the local L^2 projection $\widehat{\pi}_n^k$ on a macroelement $\widehat{A}_n$ such that

$$\int_{\widehat{A}_n} (\hat{u} - \widehat{\pi}_n^k \hat{u}) p = 0 \quad \forall\, p \in P^k(\widehat{A}_n), \tag{A.10}$$

for $\hat{u} \in L^1(\widehat{A}_n)$. The Clément operator $\mathcal{C}_h : L^1(\Omega) \to V_h^k$ is defined as

$$\mathcal{C}_h u = \sum_{i=1}^{N} \widehat{\pi}_{j(i)}^k \left(u \circ F_{A_i}\right) \left({F_{A_i}}^{-1}(a_i)\right) \phi_i. \tag{A.11}$$

Lemma A.11 (Stability of the Clément operator, [29])**:** Let $1 \leq p \leq \infty$ and $0 \leq m \leq 1$. There exists a constant C such that $\forall\, h$

$$\|\mathcal{C}_h u\|_{W^{m,p}(\Omega)} \leq C \|u\|_{W^{m,p}(\Omega)} \quad \forall\, u \in W^{m,p}(\Omega). \tag{A.12}$$

Lemma A.12 (Approximation property of the Clément operator, [29])**:** For $K \in \mathcal{T}_h$ we define the patch ω_K as the set of elements in $\mathcal{T}_h$ sharing at least one node with K. The patch ω_S for a side $S \in \mathcal{S}_h$ is defined similarly. Let l, m and p satisfy $1 \leq p \leq \infty$ and $0 \leq m \leq l \leq k+1$. Then there exists a constant C such that $\forall\, h$ and $\forall\, K \in \mathcal{T}_h$

$$\|u - \mathcal{C}_h u\|_{W^{m,p}(K)} \leq C h_K^{l-m} \|u\|_{W^{l,p}(\omega_K)} \quad \forall\, u \in W^{l,p}(\omega_K). \tag{A.13}$$

If $m + \frac{1}{p} \leq l \leq k+1$, then there exists a constant C such that $\forall\, h$ and $\forall\, K \in \mathcal{T}_h$

$$\|u - \mathcal{C}_h u\|_{W^{m,p}(S)} \leq C h_K^{l-m-\frac{1}{p}} \|u\|_{W^{l,p}(\omega_S)} \quad \forall\, u \in W^{l,p}(\omega_S). \tag{A.14}$$

1.5. Hyperbolic system

Definition A.13 (Hyperbolic system [32])**:** A system of the form

$$\frac{\partial U}{\partial t}+\nabla\cdot\mathbf{F}=0,\quad \mathbf{F}=(F^1,...,F^{\dim}),\quad U,F^i\in\mathbb{R}^k, i=1,..,\dim$$

is called hyperbolic, if for all admissible vectors U and all $\mathbf{e}\in\mathbb{R}^{\dim}$,$\mathbf{e}\neq 0$, the matrix

$$\mathbf{A}(U,\mathbf{e})=\sum_{i=1}^{\dim}\mathbf{e}_i A_i(U),\quad A_i(U)=\frac{\partial F^i}{\partial U}, i=1,..,\dim$$

has k real eigenvalues $\lambda_1(U,\mathbf{e})\leq\lambda_2(U,\mathbf{e})\leq...\leq\lambda_k(U,\mathbf{e})$ and k linearly independent right eigenvectors $\mathbf{r}_i(U,\mathbf{e})$, $i=1,...,k$. It is called strictly hyperbolic if all eigenvalues are distinct.

1.6. Flux Jacobian

Definition A.14 (Linear combination of the flux Jacobian tensor [72, 78])**:** Let us consider the Jacobian tensor $\mathbf{A}=(A_1,A_2)$, where

$$A_1=\begin{bmatrix}0 & 1 & 0 & 0\\ -v_x^2+(\gamma-1)\frac{|\mathbf{v}|^2}{2} & (3-\gamma)v_x & (1-\gamma)v_y & (\gamma-1)\\ -v_xv_y & v_y & v_x & 0\\ -\gamma v_xE+(\gamma-1)(v_x^3+v_xv_y^2) & \gamma E-(\gamma-1)(\frac{3}{2}v_x^2+\frac{1}{2}v_y^2) & (1-\gamma)v_xv_y & \gamma v_x\end{bmatrix},$$

$$A_2=\begin{bmatrix}0 & 0 & 1 & 0\\ -v_xv_y & v_y & v_x & 0\\ -v_y^2+(\gamma-1)\frac{|\mathbf{v}|^2}{2} & (1-\gamma)v_x & (3-\gamma)v_y & (\gamma-1)\\ -\gamma v_xE+(\gamma-1)(v_y^3+v_yv_x^2) & (1-\gamma)v_xv_y & \gamma E-(\gamma-1)(\frac{3}{2}v_y^2+\frac{1}{2}v_x^2) & \gamma v_y\end{bmatrix}.$$

The linear combination of these matrices

$$\mathbf{A}(U,\mathbf{e})=e_1A_1+e_2A_2,\quad \mathbf{e}=(e_1,e_2),|e|=1 \tag{A.15}$$

is given by

$$\mathbf{A}(U,\mathbf{e})=\begin{bmatrix}0 & e_1 & e_2 & 0\\ (\gamma-1)qe_1-v_xv_e & v_e-(\gamma-2)v_xe_1 & v_xe_2-(\gamma-1)v_ye_1 & (\gamma-1)e_1\\ (\gamma-1)qe_2-v_yv_e & v_ye_1-(\gamma-1)v_xe_2 & v_e-(\gamma-2)v_ye_2 & (\gamma-1)e_2\\ ((\gamma-1)q-H)v_e & He_1-(\gamma-1)v_xv_e & He_2-(\gamma-1)v_yv_e & \gamma v_e\end{bmatrix}, \tag{A.16}$$

where $v_e=\mathbf{e}\cdot\mathbf{v}$, $q=\frac{1}{2}|\mathbf{v}|^2$ and H is the total enthalpy given by $H=E+\frac{p}{\rho}$.
Furthermore this linear combination is diagonalizable

$$\mathbf{A}(U,\mathbf{e})=\mathbf{R}(U,\mathbf{e})\Lambda(U,\mathbf{e})\mathbf{R}(U,\mathbf{e})^{-1}, \tag{A.17}$$

where the diagonal matrix of eigenvalues is given by

$$\Lambda(U,\mathbf{e})=diag\{v_e-c,v_e,v_e+c,v_e\}. \tag{A.18}$$

The matrix of right eigenvectors $\mathbf{R}(U,\mathbf{e})$ and the matrix of left eigenvectors $\mathbf{L}(U,\mathbf{e}) = \mathbf{R}(U,\mathbf{e})^{-1}$ are as follows:

$$\mathbf{R}(U,\mathbf{e}) = \begin{bmatrix} 1 & 1 & 1 & 0 \\ v_x - ce_1 & v_x & v_x + ce_1 & e_2 \\ v_y - ce_2 & v_y & v_y + v_y e_2 & -e_1 \\ H - cv_e & q & H + cv_e & v_x e_2 - v_y e_1 \end{bmatrix}, \tag{A.19}$$

$$\mathbf{L}(U,\mathbf{e}) = \begin{bmatrix} \frac{1}{2}\left(bq + \frac{v_e}{c}\right) & \frac{1}{2}\left(-bv_x - \frac{e_1}{c}\right) & \frac{1}{2}\left(-bv_y - \frac{e_2}{c}\right) & \frac{1}{2}b \\ 1 - bq & bv_x & bv_y & -b \\ \frac{1}{2}\left(bq - \frac{v_n}{c}\right) & \frac{1}{2}\left(-bv_x + \frac{e_1}{c}\right) & \frac{1}{2}\left(-bv_y + \frac{e_2}{c}\right) & \frac{1}{2}b \\ e_1 v_y - e_2 v_x & e_2 & -e_1 & 0 \end{bmatrix}, \tag{A.20}$$

where $b = \frac{\gamma - 1}{c^2}$.

1.7. Maximum principle

Theorem A.15 (Semi-discrete maximum principle and positivity preservation [58])**:** Consider a semi-discrete problem of the following form:

$$\sum_j m_{ij} \frac{\mathrm{d}u_j}{\mathrm{d}t} = \sum_j k_{ij} u_j, \quad i = 1, ..., N, \tag{A.21}$$

where $u = u(t)$ is the unknown vector of size N. Assume that

$$m_{ii} > 0, \quad m_{ij} = 0, \quad k_{ij} \geq 0, \quad \forall j \neq i. \tag{A.22}$$

Then the following estimates hold

1. $\sum_j k_{ij} = 0 \wedge u_i \geq u_j, \quad \forall j \neq i \quad \Rightarrow \quad \frac{\mathrm{d}u_i}{\mathrm{d}t} \leq 0$,
2. $u_j(0) \geq 0, \forall j \quad \Rightarrow \quad u_j(t) \geq 0, \forall j, \forall t > 0.$

Definition A.16 (Local extremum diminishing (LED) [58])**:** A space discretization of the form

$$\frac{\mathrm{d}u_i}{\mathrm{d}t} = \frac{1}{m_{ii}} \sum_{j \neq i} k_{ij}(u_j - u_i), \tag{A.23}$$

where $m_{ii} > 0$ and $k_{ij} \geq 0$ for all i and $i \neq j$ is called local extremum diminishing.

Theorem A.17 (Local discrete maximum principle and positivity preservation [58])**:** The i-th equation of a fully-discrete system $Au = Bg$ is given by

$$a_{ii} u_i = b_{ii} g_i + \sum_{j \in S_i} \left(b_{ij} u_j - a_{ij} u_j\right), \tag{A.24}$$

where $S_i := \{j \neq i | a_{ij} \neq 0 \wedge b_{ij} \neq 0\}$ is the set of neighbors of node i. Assume that

$$a_{ii} > 0, \quad a_{ij} \leq 0, \quad b_{ii} \geq 0, \quad b_{ij} \geq 0, \quad \forall j \in S_i. \tag{A.25}$$

Then the following estimates hold for u_i

1. $\sum_j a_{ij} = \sum_j b_{ij} \Rightarrow u_i^{\min} \leq u_i \leq u_i^{\max}$,
2. $u_i^{\min} \geq 0 \quad \Rightarrow \quad u_i \geq 0$,

where

$$u_i^{\max} = \max\{\max_{j \in S_i \cup i} g_j, \max_{j \in S_i} u_j\},$$
$$u_i^{\min} = \min\{\min_{j \in S_i \cup i} g_j, \min_{j \in S_i} u_j\}.$$

Definition A.18 (Monotone matrix [58])**:** A regular matrix A is called monotone if

$$A^{-1} \geq 0$$

or, equivalently,

$$u \geq 0 \Rightarrow Au \geq 0.$$

Definition A.19 (M-matrix [58])**:** A regular matrix A which is monotone and satisfies

$$a_{ij} \leq 0, \forall j \neq i$$

for all i is called an M-matrix.

Theorem A.20 (Global discrete maximum principle and positivity preservation [58])**:** Consider a fully-discrete system $Au = Bg$. Assume that the coefficients of A and B satisfy (A.25) for all i. If A is strictly or irreducibly diagonally dominant, then A is an M-matrix and

1. $\sum_j a_{ij} = \sum_j b_{ij} \forall i \Rightarrow \min g \leq u \leq \max g$,
2. $g \geq 0 \quad \Rightarrow \quad u \geq 0$.

Bibliography

[1] M. Ainsworth and J.T. Oden. *A posteriori error estimation in finite element analysis*. Pure and Applied Mathematics (New York). Wiley-Interscience [John Wiley & Sons], New York, 2000. ISBN 0-471-29411-X. doi: 10.1002/9781118032824. URL http://dx.doi.org/10.1002/9781118032824.

[2] D.N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760, 1982. ISSN 0036-1429. doi: 10.1137/0719052. URL http://dx.doi.org/10.1137/0719052.

[3] D.N. Arnold and G. Awanou. The serendipity family of finite elements. *Found. Comput. Math.*, 11(3):337–344, 2011. ISSN 1615-3375. doi: 10.1007/s10208-011-9087-3. URL http://dx.doi.org/10.1007/s10208-011-9087-3.

[4] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2001/02. ISSN 0036-1429. doi: 10.1137/S0036142901384162. URL http://dx.doi.org/10.1137/S0036142901384162.

[5] B. Ayuso and L.D. Marini. Discontinuous Galerkin methods for advection-diffusion-reaction problems. *SIAM J. Numer. Anal.*, 47(2):1391–1420, February 2009. ISSN 0036-1429. doi: 10.1137/080719583. URL http://dx.doi.org/10.1137/080719583.

[6] I. Babuška and A.K. Aziz. Survey lectures on the mathematical foundations of the finite element method. In *The mathematical foundations of the finite element method with applications to partial differential equations (Proc. Sympos., Univ. Maryland, Baltimore, Md., 1972)*, pages 1–359. Academic Press, New York, 1972. With the collaboration of G. Fix and R. B. Kellogg.

[7] I. Babuška and M. Suri. The *h-p* version of the finite element method with quasi-uniform meshes. *RAIRO Modél. Math. Anal. Numér.*, 21(2):199–238, 1987. ISSN 0764-583X.

[8] I. Babuška, C.E. Baumann, and J.T. Oden. A discontinuous *hp* finite element method for diffusion problems: 1-D analysis. *Comput. Math. Appl.*, 37(9):103–122, 1999. ISSN 0898-1221. doi: 10.1016/S0898-1221(99)00117-0. URL http://dx.doi.org/10.1016/S0898-1221(99)00117-0.

[9] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys.*, 138(2):251–285, 1997. ISSN 0021-9991. doi: 10.1006/jcph.1997.5454. URL http://dx.doi.org/10.1006/jcph.1997.5454.

[10] C.E. Baumann and J.T. Oden. A discontinuous *hp* finite element method for convection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.*, 175(3-4):311–341, 1999. ISSN 0045-7825. doi: 10.1016/S0045-7825(98)00359-4. URL http://dx.doi.org/10.1016/S0045-7825(98)00359-4.

[11] R. Becker, E. Burman, P. Hansbo, and M.G. Larson. A reduced P1-discontinuous Galerkin method, 2003. Chalmers Finite Element Centre, Chalmers University of Technology.

[12] R. Becker, M. Bittl, and D. Kuzmin. Analysis of a combined cg1-dg2 method for the transport equation. *SIAM Journal on Numerical Analysis*, 53(1):445–463, 2015. doi: 10.1137/13093683X. URL http://dx.doi.org/10.1137/13093683X.

[13] M. Bittl and D. Kuzmin. An *hp*-adaptive flux-corrected transport algorithm for continuous finite elements. *Computing*, 95(1, suppl.):S27–S48, 2013. ISSN 0010-485X. doi: 10.1007/s00607-012-0223-y. URL http://dx.doi.org/10.1007/s00607-012-0223-y.

[14] M. Bittl and D. Kuzmin. The reference solution approach to *hp*-adaptivity in finite element flux-corrected transport algorithms. In I. Lirkov, S. Margenov, and J. Wasniewski, editors, *Large-Scale Scientific Computing*, Lecture Notes in Computer Science, pages 197–204. Springer Berlin Heidelberg, 2014. ISBN 978-3-662-43879-4. URL http://dx.doi.org/10.1007/978-3-662-43880-0_21.

[15] M. Bittl, D. Kuzmin, and R. Becker. The cg1-dg2 method for convection-diffusion equations in 2d. *J. Comput. Appl. Math.*, 270(0):21 – 31, 2014. ISSN 0377-0427. doi: http://dx.doi.org/10.1016/j.cam.2014.03.008. URL http://www.sciencedirect.com/science/article/pii/S0377042714001484. Fourth International Conference on Finite Element Methods in Engineering and Sciences (FEMTEC 2013).

[16] P.B. Bochev, M.D. Gunzburger, and J.N. Shadid. Stability of the SUPG finite element method for transient advection-diffusion problems. *Comput. Methods Appl. Mech. Engrg.*, 193(23-26):2301–2323, 2004. ISSN 0045-7825. doi: 10.1016/j.cma.2004.01.026. URL http://dx.doi.org/10.1016/j.cma.2004.01.026.

[17] S.C. Brenner and L.R. Scott. *The mathematical theory of finite element methods*, volume 15 of *Texts in Applied Mathematics*. Springer, New York, third edition, 2008. ISBN 978-0-387-75933-3. doi: 10.1007/978-0-387-75934-0. URL http://dx.doi.org/10.1007/978-0-387-75934-0.

[18] A.N. Brooks and T.J.R. Hughes. Streamline upwind Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32:199–259, 1982.

[19] E. Burman and P. Hansbo. Edge stabilization for Galerkin approximations of convection-diffusion-reaction problems. *Comput. Methods Appl. Mech. Engrg.*, 193(15-16):1437–1453, 2004. ISSN 0045-7825. doi: 10.1016/j.cma.2003.12.032. URL http://dx.doi.org/10.1016/j.cma.2003.12.032.

[20] P.G. Ciarlet. *The finite element method for elliptic problems*, volume 40 of *Classics in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002. ISBN 0-89871-514-8. doi: 10.1137/1.9780898719208. URL http://dx.doi.org/10.1137/1.9780898719208. Reprint of the 1978 original [North-Holland, Amsterdam; MR0520174 (58 #25001)].

[21] Ph. Clément. Approximation by finite element functions using local regularization. *RAIRO Analyse Numérique*, 9(R-2):77–84, 1975. ISSN 0399-0516.

[22] B. Cockburn and J. Guzmán. Error estimates for the Runge-Kutta discontinuous Galerkin method for the transport equation with discontinuous initial data. *SIAM J. Numer. Anal.*, 46 (3):1364–1398, 2008. ISSN 0036-1429. doi: 10.1137/060668936. URL http://dx.doi.org/10.1137/060668936.

[23] B. Cockburn and C.-W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws. V. Multidimensional systems. *J. Comput. Phys.*, 141(2):199–224, 1998. ISSN 0021-9991. doi: 10.1006/jcph.1998.5892. URL http://dx.doi.org/10.1006/jcph.1998.5892.

[24] L. Demkowicz. *Computing with hp-adaptive finite elements. Vol. 1*. Chapman & Hall/CRC Applied Mathematics and Nonlinear Science Series. Chapman & Hall/CRC, Boca Raton, FL, 2007. ISBN 978-1-58488-671-6; 1-58488-671-4. doi: 10.1201/9781420011692. URL http://dx.doi.org/10.1201/9781420011692.

[25] C.R. DeVore. An improved limiter for multidimensional flux-corrected transport. Technical report, 1998. NASA Technical Report, AD-A360122.

[26] D.A. Di Pietro and A. Ern. *Mathematical aspects of discontinuous Galerkin methods*, volume 69 of *Mathématiques & Applications (Berlin) [Mathematics & Applications]*. Springer, Heidelberg, 2012. ISBN 978-3-642-22979-4. doi: 10.1007/978-3-642-22980-0. URL http://dx.doi.org/10.1007/978-3-642-22980-0.

[27] V. Dolejší and M. Feistauer. A semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow. *J. Comput. Phys.*, 198(2):727–746, 2004. ISSN 0021-9991. doi: 10.1016/j.jcp.2004.01.023. URL http://dx.doi.org/10.1016/j.jcp.2004.01.023.

[28] V. Dolejší, M. Feistauer, and C. Schwab. On some aspects of the discontinuous Galerkin finite element method for conservation laws. *Math. Comput. Simulation*, 61(3-6):333–346, 2003. ISSN 0378-4754. doi: 10.1016/S0378-4754(02)00087-3. URL http://dx.doi.org/10.1016/S0378-4754(02)00087-3. MODELLING 2001 (Pilsen).

[29] A. Ern and J.-L. Guermond. *Theory and practice of finite elements*, volume 159 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2004. ISBN 0-387-20574-8. doi: 10.1007/978-1-4757-4355-5. URL http://dx.doi.org/10.1007/978-1-4757-4355-5.

[30] M. Feistauer and V. Kučera. On a robust discontinuous Galerkin technique for the solution of compressible flow. *J. Comput. Phys.*, 224(1):208–221, 2007. ISSN 0021-9991. doi: 10.1016/j.jcp.2007.01.035. URL http://dx.doi.org/10.1016/j.jcp.2007.01.035.

[31] C.A.J. Fletcher. The group finite element formulation. *Computer Methods in Applied Mechanics and Engineering*, 37(2):225 – 244, 1983. ISSN 0045-7825. doi: 10.1016/0045-7825(83)90122-6. URL http://www.sciencedirect.com/science/article/pii/0045782583901226.

[32] E. Godlewski and P. Raviart. *Numerical approximation of hyperbolic systems of conservation laws*, volume 118 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1996. ISBN 0-387-94529-6. doi: 10.1007/978-1-4612-0713-9. URL http://dx.doi.org/10.1007/978-1-4612-0713-9.

[33] S. Gross and A. Reusken. *Numerical methods for two-phase incompressible flows*, volume 40 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2011. ISBN 978-3-642-19685-0. doi: 10.1007/978-3-642-19686-7. URL http://dx.doi.org/10.1007/978-3-642-19686-7.

[34] B. Guo and I. Babuška. The h-p version of the finite element method. *Computational Mechanics*, 1(1):21–41, 1986. ISSN 0178-7675. doi: 10.1007/BF00298636. URL http://dx.doi.org/10.1007/BF00298636.

[35] M. Gurris. *Implicit Finite Element Schemes for Compressible Gas and Particle–Laden Gas Flows*. PhD thesis, Technische Universität Dortmund, 2010.

[36] R. Hartmann. *Adaptive Finite Element Methods for the Compressible Euler Equations*. PhD thesis, University of Heidelberg, 2002.

[37] R. Hartmann. Numerical analysis of higher order discontinuous Galerkin finite element methods. In H. Deconinck, editor, *VKI LS 2008-08: CFD - ADIGMA course on very high order discretization methods, Oct. 13-17, 2008*. Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium, 2008.

[38] R. Hartmann and P. Houston. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys.*, 183(2):508–532, 2002. ISSN 0021-9991. doi: 10.1006/jcph.2002.7206. URL http://dx.doi.org/10.1006/jcph.2002.7206.

[39] J.S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods*, volume 54 of *Texts in Applied Mathematics*. Springer, New York, 2008. ISBN 978-0-387-72065-4. doi: 10.1007/978-0-387-72067-8. URL http://dx.doi.org/10.1007/978-0-387-72067-8. Algorithms, analysis, and applications.

[40] C. Hirsch. *Numerical Computation of Internal and External Flows. Vol. II: Computational Methods for Inviscid and Viscous Flows*. John Wiley & Sons,Chichester, 1990. ISBN 978-0-471-92452-4.

[41] H. Hoteit, Ph. Ackerer, R. Mosé, J. Erhel, and B. Philippe. New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes. *Internat. J. Numer. Methods Engrg.*, 61(14):2566–2593, 2004. ISSN 0029-5981. doi: 10.1002/nme.1172. URL http://dx.doi.org/10.1002/nme.1172.

[42] P. Houston and E. Süli. *hp*-adaptive discontinuous Galerkin finite element methods for first-order hyperbolic problems. *SIAM J. Sci. Comput.*, 23(4):1226–1252, 2001. ISSN 1064-8275. doi: 10.1137/S1064827500378799. URL http://dx.doi.org/10.1137/S1064827500378799.

[43] P. Houston, C. Schwab, and E. Süli. Discontinuous *hp*-finite element methods for advection-diffusion-reaction problems. *SIAM J. Numer. Anal.*, 39(6):2133–2163, 2002. ISSN 0036-1429. doi: 10.1137/S0036142900374111. URL http://dx.doi.org/10.1137/S0036142900374111.

[44] T.J.R. Hughes. Recent progress in the development and understanding of supg methods with special reference to the compressible euler and navier-stokes equations. *International Journal for Numerical Methods in Fluids*, 7(11):1261–1275, 1987. ISSN 1097-0363. doi: 10.1002/fld.1650071108. URL http://dx.doi.org/10.1002/fld.1650071108.

[45] T.J.R. Hughes and T.E. Tezduyar. Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations. *Comput. Methods Appl. Mech. Engrg.*, 45(1-3):217–284, 1984. ISSN 0045-7825. doi: 10.1016/0045-7825(84)90157-9. URL http://dx.doi.org/10.1016/0045-7825(84)90157-9.

[46] T.J.R. Hughes, G. Scovazzi, P.B. Bochev, and A. Buffa. A multiscale discontinuous Galerkin method with the computational structure of a continuous Galerkin method. *Comp. Meth. Appl. Mech. Engrg.*, 195:2761–2787, 2006.

[47] V. John and P. Knobloch. On spurious oscillations at layers diminishing (SOLD) methods for convection-diffusion equations. I. A review. *Comput. Methods Appl. Mech. Engrg.*, 196 (17-20):2197–2215, 2007. ISSN 0045-7825. doi: 10.1016/j.cma.2006.11.013. URL http://dx.doi.org/10.1016/j.cma.2006.11.013.

[48] V. John and E. Schmeyer. On finite element methods for 3D time-dependent convection-diffusion-reaction equations with small diffusion. *Comput. Meth. Appl. Mech. Engrg.*, 198: 475–494, 2008.

[49] V. John and E. Schmeyer. Finite element methods for time-dependent convection-diffusion-reaction equations with small diffusion. *Comput. Methods Appl. Mech. Engrg.*, 198(3-4): 475–494, 2008. ISSN 0045-7825. doi: 10.1016/j.cma.2008.08.016. URL http://dx.doi.org/10.1016/j.cma.2008.08.016.

[50] C. Johnson. *Numerical solution of partial differential equations by the finite element method*. Studentlitteratur, Lund, 1987. ISBN 91-44-25241-1.

[51] C. Johnson and J. Pitkäranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Math. Comp.*, 46(173):1–26, 1986. ISSN 0025-5718. doi: 10.2307/2008211. URL http://dx.doi.org/10.2307/2008211.

[52] C. Johnson and J. Saranen. Streamline diffusion methods for the incompressible Euler and Navier-Stokes equations. *Math. Comp.*, 47(175):1–18, 1986. ISSN 0025-5718. doi: 10.2307/2008079. URL http://dx.doi.org/10.2307/2008079.

[53] C. Johnson, U. Nävert, and J. Pitkäranta. Finite element methods for linear hyperbolic problems. *Comput. Methods Appl. Mech. Engrg.*, 45(1-3):285–312, 1984. ISSN 0045-7825. doi: 10.1016/0045-7825(84)90158-0. URL http://dx.doi.org/10.1016/0045-7825(84)90158-0.

[54] P. Knabner and L. Angermann. *Numerical methods for elliptic and parabolic partial differential equations*, volume 44 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 2003. ISBN 0-387-95449-X.

[55] L. Krivodonova and M. Berger. High-order accurate implementation of solid wall boundary conditions in curved geometries. *J. Comput. Phys.*, 211(2):492–512, 2006. ISSN 0021-9991. doi: 10.1016/j.jcp.2005.05.029. URL http://dx.doi.org/10.1016/j.jcp.2005.05.029.

[56] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, and J.E. Flaherty. Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. *Appl. Numer. Math.*, 48(3-4):323–338, 2004. ISSN 0168-9274. doi: 10.1016/j.apnum.2003.11.002. URL http://dx.doi.org/10.1016/j.apnum.2003.11.002. Workshop on Innovative Time Integrators for PDEs.

[57] P. Kus. *Automatic hp-Adaptivity on Meshes with Arbitrary-Level Hanging Nodes in 3D*. PhD thesis, Charles University, Prague, 2011.

[58] D. Kuzmin. Algebraic flux correction I. Scalar conservation laws. In D. Kuzmin, R. Löhner, and S. Turek, editors, *Flux-Corrected Transport: Principles, Algorithms, and Applications*, pages 145–192. Springer, 2nd edition, 2012.

[59] D. Kuzmin. On the design of algebraic flux correction schemes for quadratic finite elements. *Comput. Appl. Math.*, 218:1:79–87, 2008.

[60] D. Kuzmin. Slope limiting for discontinuous galerkin approximations with a possibly non-orthogonal taylor basis. *International Journal for Numerical Methods in Fluids*, 71(9):1178–1190, 2013. ISSN 1097-0363. doi: 10.1002/fld.3707. URL http://dx.doi.org/10.1002/fld.3707.

[61] D. Kuzmin and F. Schieweck. A parameter-free smoothness indicator for high-resolution finite element schemes. *Central European Journal of Mathematics*, 11(8):1478–1488, 2013. ISSN 1895-1074. doi: 10.2478/s11533-013-0254-4. URL http://dx.doi.org/10.2478/s11533-013-0254-4.

[62] D. Kuzmin and S. Turek. Flux correction tools for finite elements. *J. Comput. Phys.*, 175:525–558, 2002.

[63] D. Kuzmin, M. Möller, J.N. Shadid, and M. Shashkov. Failsafe flux limiting and constrained data projections for equations of gas dynamics. *J. Comput. Phys.*, 229(23):8766–8779, 2010. ISSN 0021-9991. doi: 10.1016/j.jcp.2010.08.009. URL http://dx.doi.org/10.1016/j.jcp.2010.08.009.

[64] D. Kuzmin, M. Möller, and M. Gurris. *Algebraic Flux Correction II*. Scientific Computation. Springer Netherlands, 2012. ISBN 978-94-007-4037-2. doi: 10.1007/978-94-007-4038-9_7. URL http://dx.doi.org/10.1007/978-94-007-4038-9_7.

[65] C.B. Laney. *Computational Gasdynamics*. Cambridge University Press, 1998. ISBN 9780521625586. URL http://books.google.de/books?id=r-bYw-JjKGAC.

[66] S. Larsson and V. Thomée. *Partial differential equations with numerical methods*, volume 45 of *Texts in Applied Mathematics*. Springer-Verlag, Berlin, 2003. ISBN 3-540-01772-0.

[67] R.J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, Cambridge, 2002. ISBN 0-521-81087-6; 0-521-00924-3. doi: 10.1017/CBO9780511791253. URL http://dx.doi.org/10.1017/CBO9780511791253.

[68] R.J. LeVeque. *Numerical methods for conservation laws*. Lectures in Mathematics ETH Zürich. Birkhäuser Verlag, Basel, second edition, 1992. ISBN 3-7643-2723-5. doi: 10.1007/978-3-0348-8629-1. URL http://dx.doi.org/10.1007/978-3-0348-8629-1.

[69] R.J. LeVeque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.*, 33:627–665, 1996.

[70] R. Löhner, K. Morgan, J. Peraire, and M. Vahdati. Finite Element Flux-Corrected Transport (FEM-FCT) for the Euler and Navier-Stokes equations. Technical report, 1987.

[71] W.F. Mitchell and M.A. McClain. A survey of *hp*-adaptive strategies for elliptic partial differential equations. In *Recent advances in computational and applied mathematics*, pages 227–258. Springer, Dordrecht, 2011. doi: 10.1007/978-90-481-9981-5_10. URL http://dx.doi.org/10.1007/978-90-481-9981-5_10.

[72] M. Möller. *Adaptive High-Resolution Finite Element Schemes*. PhD thesis, Dortmund University of Technology, 2008.

[73] J. Nečas. Sur une méthode pour résoudre les équations aux dérivées partielles du type elliptique, voisine de la variationnelle. *Ann. Scuola Norm. Sup. Pisa (3)*, 16:305–326, 1962.

[74] S. Prudhomme and J.T. Oden. Computable error estimators and adaptive techniques for fluid flow problems. In TimothyJ. Barth and Herman Deconinck, editors, *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, volume 25 of *Lecture Notes in Computational Science and Engineering*, pages 207–268. Springer Berlin Heidelberg, 2003. ISBN 978-3-642-07841-5. doi: 10.1007/978-3-662-05189-4_5. URL http://dx.doi.org/10.1007/978-3-662-05189-4_5.

[75] S. Prudhomme, F. Pascal, J.T. Oden, and A. Romkes. Review of a priori error estimation for discontinuous Galerkin methods. Technical report, 2000.

[76] W.H. Reed and T.R. Hill. *Triangular mesh methods for the neutron transport equation*. Oct 1973. URL http://www.osti.gov/scitech/servlets/purl/4491151.

[77] B. Rivière, M.F. Wheeler, and V. Girault. Improved energy estimates for interior penalty, constrained and discontinuous Galerkin methods for elliptic problems. I. *Comput. Geosci.*, 3(3-4):337–360 (2000), 1999. ISSN 1420-0597. doi: 10.1023/A:1011591328604. URL http://dx.doi.org/10.1023/A:1011591328604.

[78] A. Rohde. Eigenvalues and eigenvectors of the euler equations in general geometries. *AIAA Paper 2001-2609*, 2001.

[79] R.A. Shapiro. *Adaptive finite element solution algorithm for the Euler equations*, volume 32 of *Notes on Numerical Fluid Mechanics*. Friedr. Vieweg & Sohn, Braunschweig, 1991. ISBN 3-528-07632-1. doi: 10.1007/978-3-322-87879-3. URL http://dx.doi.org/10.1007/978-3-322-87879-3.

[80] G.A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Computational Phys.*, 27(1):1–31, 1978. ISSN 0021-9991.

[81] P. Solin and J. Cerveny. Automatic *hp*-adaptivity with arbitrary-level hanging nodes, 2006. research report.

[82] P. Solin and et al. *Hermes - Higher-Order Modular Finite Element System (User's Guide)*. URL http://hpfem.org/. http://hpfem.org/.

[83] P. Solin, K. Segeth, and I. Dolezel. *Higher-Order Finite Element Methods*. Chapman and Hall / CRC Press, 2003.

[84] P. Solin, J. Cerveny, and I. Dolezel. Arbitrary-level hanging nodes and automatic adaptivity in the *hp*-fem. *Math. Comput. Simul.*, 77:117 – 132, 2008.

[85] R. Verfürth. A posteriori error estimation and adaptive mesh-refinement techniques. In *Proceedings of the Fifth International Congress on Computational and Applied Mathematics (Leuven, 1992)*, volume 50, pages 67–83, 1994. doi: 10.1016/0377-0427(94)90290-9. URL http://dx.doi.org/10.1016/0377-0427(94)90290-9.

[86] P. Wesseling. *Principles of computational fluid dynamics*, volume 29 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2001. ISBN 3-540-67853-0. doi: 10.1007/978-3-642-05146-3. URL http://dx.doi.org/10.1007/978-3-642-05146-3.

[87] F.M. White. *Fluid Mechanics*. McGraw-Hill series in mechanical engineering. McGraw-Hill, 2003. ISBN 9780072402179. URL http://books.google.de/books?id=1DYtptq3OC4C.

[88] S.T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *J. Comput. Phys.*, 31:335–362, 1979.

[89] O.C. Zienkiewicz and J.Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Int. J. Numer. Methods Engrg.*, 24:2:337–357, 1987.

[90] O.C. Zienkiewicz and J.Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. part 2: Error estimates and adaptivity. *Int. J. Numer. Methods Engrg.*, 33:1365–1382, 1992.