von Karman Institute for Fluid Dynamics
Chaussée de Waterloo, 72
B - 1640 Rhode Saint Genèse - Belgium

Project Report

# ADJOINT BASED GOAL ORIENTED ERROR ESTIMATION FOR ADAPTIVE PETROV-GALERKIN FINITE ELEMENT METHODS

## Application to convection-diffusion problems

T. Horváth

Supervisors: H. Deconinck, S. D'Angelo

June 2014

# Acknowledgments

First of all I would like to say thank you to my supervisor Herman Deconinck and to my advisor Stefano D'Angelo for their endless help and support. Professor Deconinck: thanks for always finding some time for my questions, although, you were busy all year long; Stefano: thanks for your patience at the beginning when I had to get to the right level of programming (I know, I am still not good enough) and for your time at the end, when you were also busy with your thesis.

To my fellows at the study room: thank you for the whole year. Yes, it was difficult, but there was always a reason to smile. I do not have enough space here to say thank you one-by-one, but I would like to say thank you for Aude and Matteo for being my private drivers sometimes and to Nándi for having someone to talk to in Hungarian. Oh, yes, here I have to mention Lilla, Imre, Tamás and Tamás also, many thanks!

I also would like to say thank you for the VKI community. When I came here, according to my papers, I was an applied mathematician, but I always thought about myself as a theoretical mathematician. After this 9 months I can really say that I have some knowledge about the applied side. Thanks for showing so many interesting topics - interesting both as a mathematician and as an engineer!

I could not the left out from here the family I lived with. Merci for the whole year! Especially to the kids, for cheering me up, even when I was in a bad mood due to my work.

Last but not least, thanks for my family and friends for helping me whenever I missed being at home.

# Abstract

Adjoint based goal oriented error estimation will be presented for dissipative problems using streamline upwind Petrov-Galerkin finite element methods. Goal oriented means that we are not interested in the global solution but some functional values of it. This functional could represent the solution at a given point, i.e. pressure at the stagnation point if we are interested in the flow over an airfoil, or it could be some weighted integrals of the solution over the domain or over the boundary.

The error estimation process could provide a bound on the error between the functional value of the exact solution and the functional value of the discrete solution. It also provides some local indicators, using which, it can be determined over which triangles the error is significant. Using mesh adaptation these triangles could be refined, in order to increase accuracy.

There are methods which use only the residual of the original problem in order to identify where the mesh needs to be refined. However, these methods usually refines the mesh around every "problematic region". For example, if we are interested in a point value of the solution inside the boundary layer, then these methods will refine all over the boundary layer.

There is another approach, in which the adjoint problem is solved also, and the error in the target functional is estimated using the residual of the original problem weighted with the adjoint solution. This leads to a more accurate error representation. In the previous example during the refinement process only some parts of the boundary layer will be resolved.

To calculate the error estimation, the adjoint problem has to be solved two times, even though, it does not increase the computational cost significantly, due to the fact that the adjoint problem is linear, even if the original one is nonlinear.

Due to the fact, that convection dominated problems will be examined, some stabilized discretisation will be needed. In the current work streamline upwind Petrov-Galerkin will be used. Some other discretisations could also be applied, such as bubble stabilized methods or Residual Distribution, however, it will be indicated that the streamline approach provides the best convergence results. It will be proved that the convergence rate of the Residual Distribution method in the adjoint problem is restricted to 1/2 for every polynomial degree.

The corresponding theoretical background will be presented, and the results will be supported by numerical simulations. The starting point of the research will be the linear diffusion-advection-reaction problem. All the basic ideas will be introduced through that case. After that, the results will be extended to system of equations and to nonlinear problems. Finally, the theoretical background will be presented for the compressible Navier-Stokes equations.

# Contents

# List of Figures

# List of Tables

# List of Symbols

**Acronyms**

| | |
|---|---|
| CFD | Computational Fluid Dynamics |
| DG | Discontinuous Galerkin |
| FEM | Finite Element Method |
| IPDG | Interior Penalty Discontinuous Galerkin |
| RD-LDA | Residual Distribution-Low Diffusion A |
| SDFEM | Streamline Diffusion Finite Element Method |
| SUPG | Streamline Upwind Petrov-Galerkin |
| VKI | von Karman Institute |

**Roman symbols**

| | | |
|---|---|---|
| $H^m$ | Sobolev space of order $m$ | - |
| $L^2$ | space of square integrable functions | - |
| $\mathbf{b}$ | advection field | - |
| $\mathbf{n}$ | outward normal vector | - |
| $p$ | polynomial degree | - |
| $h$ | mesh size | - |
| $J$ | target functional | - |
| $\mathcal{K}$ | thermal conductivity | $W/(kgK)$ |
| $e$ | specific static internal energy | $J$ |
| $p$ | pressure | $Pa$ |
| $E$ | total energy | $J$ |
| $H$ | total enthalpy | $J$ |
| $T$ | temperature | $K$ |
| Pr | Prandtl number | - |
| $v_1$ | x component of velocity | $m/s$ |
| $v_2$ | y component of velocity | $m/s$ |
| $\mathbf{v}$ | velocity vector | $m/s$ |

**Greek symbols**

| | | |
|---|---|---|
| $\Omega$ | domain in $\mathbb{R}^2$ | - |
| $\Omega_0$ | reference domain in $\mathbb{R}^2$ | - |
| $\Gamma$ | boundary of the domain | - |
| $\alpha$ | penalty parameter on the Dirichlet boundary | - |
| $\gamma$ | ratio of specific heat capacities | - |
| $\eta$ | error indicator | - |
| $\mu$ | dynamic viscosity | $Pa\,s$ |
| $\rho$ | density | $kg/m^3$ |
| $\phi$ | FEM basis function | - |
| $\tau$ | stress tensor | $Pa$ |
| $\tau_h$ | SUPG stabilising parameter | - |
| $\mathcal{T}_h$ | mesh | - |

**Sub- and Superscripts**

| | |
|---|---|
| $K$ | mesh element |
| $h$ | numerical solution |
| $l$ | lift |
| $d$ | drag |
| $\infty$ | free-stream |

# Chapter 1

# Introduction

## 1.1 Goal-oriented adjoint-based error estimation

Error estimation and adaptive solution procedures for CFD computations are of great importance to reduce the computational cost. In this Research Master project we considered a goal-oriented (target based) error estimation by not focusing on the global solution itself, but some functional values of it. To get these values as accurate as possible an adaptive strategy is provided, by which we solve one additional linear problem (even if the original problem is nonlinear) and its solution is used to identify where the mesh refinement is needed.

This kind of error estimation has been used by many authors [21, 17, 44] for different discretization techniques, such as classical finite element method [1], discontinuous Galerkin [24] or streamline upwind Petrov-Galerkin [12] method.

## 1.2 Aim of the project

The goal of this project was to extend to viscous problems the studies made by Stefano D'Angelo, [12], who was working on this type of error estimation for inviscid problems. It has to be emphasized, that by including viscous terms the behaviour is significantly different from the case when viscous effects are neglected (Euler-equations) and this changes the type of the partial differential equations as well.

By courtesy of Stefano D'Angelo this kind of error estimation was tested by the code APOGEE. Originally APOGEE was set to handle inviscid problems and it dealt only with convective fluxes. In this project APOGEE has been extended to viscous problems by implementing the viscous flux. Moreover, in order to make the testing more flexible, reaction fluxes and source terms were also implemented.

Reaction terms were initially used in the linear case, because in FEM the natural boundary condition is the Neumann one, however, if there are only viscous and convective fluxes then the problem cannot be equipped with fully Neumann boundary condition, otherwise the solution will only be unique up to a constant.

Source terms were implemented in order to be able to test the code with fabricated solutions. For example in the case of Navier-Stokes equations analytical solutions are known only for some very special cases. Using the source terms is was possible to plug in any function as an exact solution by setting the source term properly.

Our aim is to solve the problem such that the error in whatever sense it is measured (global error or goal oriented) should be smaller than a user given tolerance, i.e.,

$$Error < TOL\,.$$

A posteriori error estimation aims to bound the error using the coefficient functions of the PDE and the discrete solution, but not the (unknown) analytical solution. Therefore, such an error estimate can give information about the error with some error indicators that provides error estimation, $\eta_K$, for every single element $K$ of the mesh, and bounds the error

$$Error \leq \sum_K \eta_K\,.$$

Moreover, using $\eta_K$ as local indicators an adaptive process can be applied until the error is small enough. Some elements can be flagged as the error is significant and a mesh adapting tool can refine these elements. On the refined mesh the problem can be solved again and the error will decrease. This adaptation strategy can be used up until

$$\sum_K \eta_K < TOL\,,$$

which is even stronger than the original goal.

As mentioned in the goal oriented error estimation the error is not the global one, but some functional value of the solution. This functional will be denoted by $J(\cdot)$. In other words the error that will be estimated is

$$Error = J(u) - J(u_h)\,,$$

where $u$ is the exact solution, while $u_h$ is the numerical one.

Such a target functional can represent many functionals of physical interest. For example in the case of flow around an airfoil it can represent the solution at a given point, such as pressure at the leading edge, or even some integrals of the solution, such as lift/drag or moment coefficients.

## 1.3 Bibliography background

Hartmann in his thesis ([24]) used this method successfully for convection problems. For example drag and lift coefficients were calculated for NACA0012 airfoil at subsonic speed, and the pressure at the leading edge was calculated in the supersonic case. In that work, the discretization method was the Interior Penalty Discontinuous Galerkin

(IPDG) method. In [12] D'Angelo follows those test cases, but with SUPG and Residual Distribution discretization.

Hartmann has several chapters in VKI Lecture Series books on this topic also. In [21], the basic concepts are introduced for linear and nonlinear target functions, for both Euler and Navier-Stokes equations. The used discretization scheme is again the IPDG method.

With Houston in [23], they introduced a new approach for multiple target quantities, which reduces the computational cost in comparison to the standard method. In that work anisotropic mesh refinement is presented that can be useful for example in the case of boundary layer problems.

In [22], Hartmann extended his worked also for turbulent flows using RANS-$k-\omega$ method and successfully determine the total drag, lift and pitching moment coefficients of a VFE-2 delta wing in turbulent flows.

There are other types of Discontinuous Galerkin discretization and Fidkowski in [17] shows some results on time dependent problems discretized with Bassi-Rebay scheme and Hybridizable Discontinuous Galerkin methods. In the case of time dependent problems the adjoint problem is backward in time, from the current time level to the initial one, therefore the computational costs are quite high.

## 1.4 Outline

In Chapter 2 the basics of weak form, classical and streamline upwind finite element methods will be studied through the linear diffusion-convection-reaction problem with main focus on the convection dominated case. This will serve as a basic model of the Navier-Stokes equations as this is the easiest problem where boundary layer like solution can be generated.

Chapter 3 will deal with adjoint based goal oriented error estimation. The continuous adjoint problem and the basic ideas of this type of error estimation process will be introduced. The error estimation formula will be deduced from which two error estimators can be created and their efficiency can be examined. Also the basic concepts of the adaptive process will be discussed in this section.

Chapter 4 extends previous chapters to nonlinear problems and systems. The nonlinear problem will be the viscous Burgers' equation, which has the same nonlinearity as the Navier-Stokes equations and the linear system will be a coupled one, to establish the connection to the Navier-Stokes problem.

Chapter 5 is devoted to the governing equations of the 2D compressible Navier-Stokes problem. The discrete variational formulation will be presented using streamline upwind Petrov-Galerkin finite element method. Some of the possible target functionals will also be presented, such as lift/drag coefficients.

Chapter 6 will conclude the achieved result and it will pave the route for possible continuation of the current research. Finally there are three Appendices that are closing the report. Appendix A contains some mathematical materials that is required to make the report self standing, but putting them into the Appendix enables a better presentation, while Appendix B contains some additional test results. The proof of the main mathematical result will be presented in Appendix C.

# Chapter 2

# Streamline upwind finite element method

## 2.1 Variational formulation of a linear convection-diffusion-reaction equation

Let us consider a bounded open domain $\Omega \subset \mathbb{R}^2$ and denote by $\Gamma = \partial\Omega$ its boundary. The linear convection-diffusion-reaction equation has the following form

$$-\varepsilon \triangle u + \nabla \cdot (\mathbf{b}u) + cu = f \quad \text{in } \Omega, \tag{2.1}$$

$$u = g_D \quad \text{on } \Gamma_D, \tag{2.2}$$

$$\varepsilon \nabla u \cdot \mathbf{n} = g_N \quad \text{on } \Gamma_N, \tag{2.3}$$

where $\Gamma_D$ and $\Gamma_N$ are the Dirichlet and Neumann parts of the boundary, respectively. They have the following properties: $\Gamma_D \cup \Gamma_N = \Gamma$ and $\Gamma_D \cap \Gamma_N = \emptyset$. The parameter $\varepsilon$ is a small positive quantity throughout this work, $\mathbf{b} \in \mathbb{R}^2$ denotes a solenoidal advection field, $\mathbf{n}$ denotes the outward normal vector of $\Omega$ and the data $f \in L^2(\Omega)$, $g_N \in L^2(\Gamma_N)$, $g_D \in H^{1/2}(\Gamma_D)$ are given functions.

Classical solution of problem (2.1)-(2.3) is a function $u \in C^2(\Omega) \cap C(\overline{\Omega})$ that satisfies the differential equation pointwise, where $\overline{\Omega}$ is the closure of $\Omega$. However, there are several physical phenomena for which there is no $C^2$ solution, therefore the weak solution needs to be used.

To work with the weak form and the weak solution we have to introduce the function space $H^1(\Omega)$. This function space contains functions that are square integrable and all of their derivatives are also square integrable, i.e.,

$$H^1(\Omega) := \{u : u \in L^2(\Omega), \nabla u \in [L^2(\Omega)]^2\},$$

For more details on the used function spaces see Appendix A.1.

To get to the weak solution, the variational formulation has to be set up. There are two ways to do this. One is to multiply equation (2.1) by a test function $v$ and use

Green Theorem in the second order term, while the other is to rewrite the second order term into a first order system. The second approach will be used here. There are also two ways of handling the first order term: leave it as it is, [13, 8], or use Green Theorem twice and distinguish the inflow and outflow boundary [21]. In the current work the first approach will be used. Therefore, to simplify the notations as long as the boundary conditions are handled properly, the analysis is restricted to the case of the pure second order problem. In other words, suppose that $\mathbf{b} = \mathbf{0}$, $c = 0$, $\varepsilon = 1$ and the remaining Poisson equation can be rewritten as a first order equation as follows

$$\sigma = \nabla u, \quad -\nabla \cdot \sigma = f \quad \text{in } \Omega, \quad u = g_D \quad \text{on } \Gamma_D, \quad \nabla u \cdot \mathbf{n} = g_N \quad \text{on } \Gamma_N.$$

The first is multiplied by test function $\phi$ and the second by test function $v$. After this, an integration over $\Omega$ and a partial integration is applied. Thus

$$\begin{aligned} \int_\Omega \sigma \cdot \phi \, \mathrm{d}\mathbf{x} &= -\int_\Omega u \nabla \cdot \phi \, \mathrm{d}\mathbf{x} + \int_\Gamma u\mathbf{n} \cdot \phi \, \mathrm{d}s \,, \\ \int_\Omega \sigma \cdot \nabla v \, \mathrm{d}\mathbf{x} &= \int_\Omega f v \, \mathrm{d}\mathbf{x} + \int_\Gamma \sigma \cdot \mathbf{n} v \, \mathrm{d}s \,. \end{aligned}$$

To go the discrete level we denote the approximate counterparts of all functions using the subscript $h$.

$$\int_\Omega \sigma_h \cdot \phi_h \, \mathrm{d}\mathbf{x} = -\int_\Omega u_h \nabla \cdot \phi_h \, \mathrm{d}\mathbf{x} + \int_\Gamma \hat{u}_h \mathbf{n} \cdot \phi_h \, \mathrm{d}s \,, \tag{2.4}$$

$$\int_\Omega \sigma_h \cdot \nabla v_h \, \mathrm{d}\mathbf{x} = \int_\Omega f v_h \, \mathrm{d}\mathbf{x} + \int_\Gamma \hat{\sigma}_h \cdot \mathbf{n} v_h \, \mathrm{d}s \,, \tag{2.5}$$

where $\hat{u}_h$ and $\hat{\sigma}_h$ are the numerical flux functions, approximating $u$ and $\nabla u$, respectively. Naturally, on $\Gamma_D$ we have $\hat{u}_h = g_D$ and on $\Gamma_N$ we have $\hat{\sigma}_h \cdot \mathbf{n} = g_N$. Applying partial integration on equation 2.4 and setting $\phi_h = \nabla v_h$

$$\int_\Omega \sigma_h \cdot \nabla v_h \, \mathrm{d}\mathbf{x} = \int_\Omega \nabla u_h \cdot \nabla v_h \, \mathrm{d}\mathbf{x} - \int_\Gamma \nabla v_h \cdot \mathbf{n}(u_h - \hat{u}_h) \, \mathrm{d}s. \tag{2.6}$$

Using the fact that the right hand sides of (2.5) and (2.6) are the same and using the boundary conditions (2.2)-(2.3)

$$\begin{aligned} \int_\Omega \nabla u_h \cdot \nabla v_h \, \mathrm{d}\mathbf{x} = &\int_\Omega f v_h \, \mathrm{d}\mathbf{x} \\ &+ \int_{\Gamma_D} \nabla v_h \cdot \mathbf{n}(u_h - g_D) + \int_{\Gamma_D} \nabla u_h \cdot \mathbf{n} v_h \, \mathrm{d}s + \int_{\Gamma_N} g_N v_h \, \mathrm{d}s \,. \end{aligned}$$

**Remark 2.1.** *Naturally, the complete equation* (2.1) *can be rewritten as a first order system, but in that case the formulas are quite complicated.*

To sum up, the weak solution reads as follows.

**Problem Set 2.2.** *Seek $u \in H^1(\Omega)$ such that $\forall v \in H^1(\Omega)$*

$$B_0(u,v) = F_0(v),$$

*where*

$$\begin{aligned} B_0(u,v) =& \varepsilon \int_\Omega \nabla u \cdot \nabla v \ d\mathbf{x} + \int_\Omega \nabla \cdot (\boldsymbol{b}u)v \ d\mathbf{x} + \int_\Omega cuv \ d\mathbf{x} \\ & - \int_{\Gamma_D} u\varepsilon\nabla v \cdot \boldsymbol{n} \ ds - \int_{\Gamma_D} \varepsilon\nabla u \cdot \boldsymbol{n}v \ ds \,, \end{aligned} \tag{2.7}$$

$$F_0(v) = \int_\Omega fv \ d\mathbf{x} - \int_{\Gamma_D} g_D\varepsilon\nabla v \cdot \boldsymbol{n} \ ds + \int_{\Gamma_N} g_N v \ ds \,. \tag{2.8}$$

However, if we want to impose the boundary conditions weakly, then according to Nitche ([36]) we have to modify artificially the Dirichlet boundary condition. This has to be replaced by an artificial Robin condition

$$u = g_D \quad \text{on } \Gamma_D \Longrightarrow u + \alpha^{-1}\varepsilon\nabla u \cdot \mathbf{n} = g_D \quad \text{on } \Gamma_D \,,$$

where $\alpha$ is a parameter. If we want to insert this into the weak form we can use the fact that $\varepsilon\nabla u \cdot \mathbf{n} = \alpha(g_D - u)$. Using this we have

$$\int_{\Gamma_D} \varepsilon\nabla uv \cdot \mathbf{n} \, \mathrm{d}s = \int_{\Gamma_D} \alpha(g_D - u)v \,.$$

For more details on weakly imposing the Dirichlet boundary condition see Appendix A.3

With this the bilinear and linear form can be reformulated, and the final problem can be set.

**Problem Set 2.3.** *Seek $u \in H^1(\Omega)$ such that $\forall v \in H^1(\Omega)$*

$$B(u,v) = F(v),$$

*where*

$$\begin{aligned} B(u,v) =& \varepsilon \int_\Omega \nabla u \cdot \nabla v \ d\mathbf{x} + \int_\Omega \nabla \cdot (\boldsymbol{b}u)v \ d\mathbf{x} + \int_\Omega cuv \ d\mathbf{x} \\ & - \int_{\Gamma_D} u\varepsilon\nabla v \cdot \boldsymbol{n} \ ds - \int_{\Gamma_D} \varepsilon\nabla u \cdot \boldsymbol{n}v \ ds + \alpha \int_{\Gamma_D} uv \ ds \,, \end{aligned} \tag{2.9}$$

$$F(v) = \int_\Omega fv \ d\mathbf{x} - \int_{\Gamma_D} g_D\varepsilon\nabla v \cdot \boldsymbol{n} \ ds + \int_{\Gamma_N} g_N v \ ds + \alpha \int_{\Gamma_D} g_D v \ ds \,. \tag{2.10}$$

**Remark 2.4.** *Throughout the following section we always use the above definition of $B(\cdot,\cdot)$ and $F(\cdot)$. The method was implemented into APOGEE through Problem Set 2.3.*

**Definition 2.5.** *Suppose that the bilinear form is defined over $V \times V$ and the linear form is defined over $V$. Let us denote by $|\!|\!|\cdot|\!|\!|$ a norm on $V$.*

- *The bilinear form is* **continuous** *on $V \times V$, if there exists $C_c > 0$ such that $B(u,v) \leq C_c \|u\| \|v\|$, $\forall u, v \in V$.*
- *The bilinear form is* **coercive** *on $V \times V$, if there exists $C_s > 0$ such that $Bu, u) \geq C_s \|u\|^2$, $\forall u \in V$.*
- *The linear form is* **continuous** *on $V$, if there exists $C_l > 0$ such that $F(u) \leq C_l \|u\|$, $\forall u \in V$.*

It can be found in many FEM textbooks that these properties hold for the bilinear and linear forms ((2.7) and (2.8)). To prove the existence and uniqueness of the weak solution the Lax-Milgram Lemma is required.

**Theorem 2.6** (Lax-Milgram Lemma)**.** *Let $H$ be real Hilbert space, $B : H \times H \to \mathbb{R}$ is a bounded, coercive bilinear form. For all bounded linear functionals, $F : H \to \mathbb{R}$ there exist a unique $u \in H$ such that $F(v) = B(u,v)$ for all $v \in H$.*

To the model problem this lemma can be used by setting $H = H^1(\Omega)$ and using the bilinear and linear forms (2.7) and (2.8), respectively.

## 2.2 Finite element method

The above defined problem cannot be handled numerically, because $H^1(\Omega)$ is infinite dimensional. To construct a numerical method we should reduce it to a finite dimensional problem. The simplest way is to define a finite dimensional subspace $V_{h,p} \subset H^1(\Omega)$ and

**Problem Set 2.7.**

$$\begin{cases} \textit{Seek } u_{h,p} \in V_{h,p} \textit{ such that} \\ B(u_{h,p}, v_{h,p}) = F(v_{h,p}) \quad \forall v_{h,p} \in V_{h,p}. \end{cases}$$

$B(\cdot,\cdot)$ and $F(\cdot)$ inherit boundedness and coercivity from $H^1(\Omega)$ to $V_{h,p}$ hence the existence and uniqueness can be proved similarly as we did in the case of Problem Set 2.3.

When we want to impose the Dirichlet boundary conditions weakly the stabilizing constant $\alpha$ has to be large, because its inverse produces the artificial Robin condition, and this has to be small, and usually $\alpha \in O(k^2/h)$ is a proper choice, where $p$ is the polynomial degree, $h$ is the mesh size. For more details on this see Appendix A.3.

We shall define a suitable finite dimensional space $V_{h,p}$. First of all we have to decompose the domain $\Omega$ into elements: typically triangles in two dimensions and tetrahedrons in three dimensions. In some cases other elements are also included: quadrilaterals, cubes or prisms. In this work we will consider only one and two dimensional examples, therefore three dimensional meshes will not be examined.

The set of the elements will be denoted by $\mathcal{T}_h = \{K_i, i = 1, \dots, N_{el}\}$, where $\cup_i K_i = \Omega$, and $int\, K_i \cap int\, K_j = \emptyset$ whenever $i \neq j$. We have an extra restriction: two neighbouring elements should share a common edge. This means that hanging nodes are excluded.

On the left side of Figure 2.1 the mesh satisfies this, however, in the middle there is a hanging node. Hanging nodes are such nodes that lie on an edge of the neighbouring triangle ($\partial E \cap \partial F$ is not an edge of $E$). The meshes without hanging node are called regular, otherwise $n$-irregular, where $n$ is the maximum number of the hanging nodes over an edge. In the middle of Figure 2.1 there is a 1-irregular mesh while in the right hand side there is a 2-irregular. However, in the discontinuous Galerkin framework, which is used in i.e. [21, 17] hanging nodes can be implemented without any difficulties. This becomes an enormous advantage when dealing with adaptive mesh refinement.

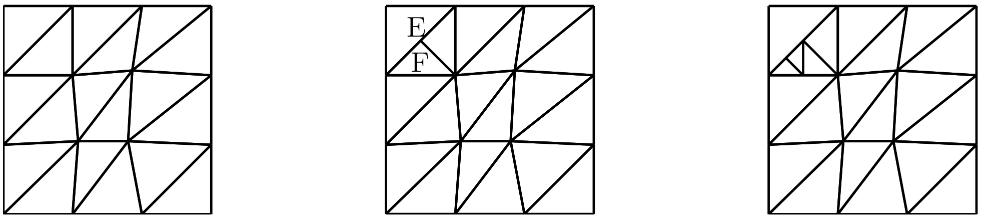

Figure 2.1: Left: regular mesh, middle: 1-irregular mesh, right: 2-irregular mesh.

In the case of standard finite element techniques the irregular meshes are excluded. However, there are some papers on irregular meshes, see e.g. [42].

The space $V_{h,p}$ contains continuous piecewise polynomials of degree $p$ over the elements. Let us denote by $\Phi_1, \ldots, \Phi_N$ a basis of $V_{h,p}$. Using these notations we seek $u_{h,p}$ as

$$u_{h,p} = \sum_{i=1}^{N} c_i \Phi_i.$$

Due to the bilinearity of $B(\cdot, \cdot)$ the equation in Problem Set 2.7 has to be satisfied only for the basis functions, leading to a system of linear equations.

We will consider only the case of Lagrange basis functions (sometimes called Lagrange elements). Such a function is associated with a point in the mesh, not necessarily a mesh node.

For example $V_{h,1}$ contains piecewise linear functions. Let us denote by $x_i$ an interior node and let us introduce the basis function $\Phi_i$ as follows: $\Phi_i(x_j) = \delta_{i,j}$ ($\delta_{i,j}$ is the Kronecker-delta). This property and the linearity of $\Phi_i$ determine the function. The support of $\Phi_i$ will be the union of those mesh elements that have $x_i$ as a node. The function $\Phi_i$ will be referred to as the basis function associated to the mesh node $x_i$.

Higher order spaces can be constructed similarly, although, the basis functions are associated not only to the nodes, but to the edges and to the elements. That is $V_{h,2}$ contains piecewise quadratic functions, and the basis functions will belong to the nodes of $\mathcal{T}_h$ (nodal functions) or to the midpoint of the edges (edge functions) while in $V_{h,3}$ we have piecewise cubic functions, and the basis functions will belong either to the nodes of $\mathcal{T}_h$ (nodal functions) or to the edges (edge functions) or to the element itself (bubble functions). Usually the bubble functions are chosen such that they belong to some interior points, see Figure 2.2.

In Figure 2.2 there are the $\mathcal{DOF}$ points over $\Omega_0$ for polynomial degree one, two and three. The nodes are denoted by • the corresponding functions are the nodal functions. The functions that are belonging to the points on the edges (denoted by ∘) are the edge functions. Finally, there are the bubble functions: these functions are associated with the element, although it is convenient to define point(s) inside the element (denoted by ×) and set the functions to be equal to 1 at a given interior point (and 0 at the others).

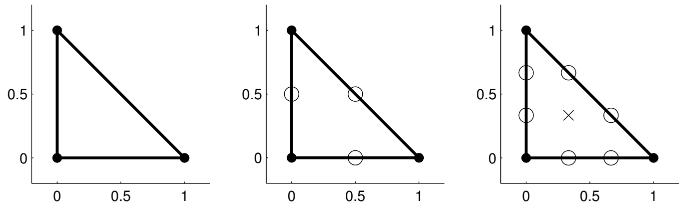

Figure 2.2: $\mathcal{DOF}$ for Lagrange basis functions for $p = 1, 2, 3$.

The set of the points that are associated with basis functions will be denoted by $\mathcal{DOF}$ and it is called degree of freedom. If first degree polynomials are used then $\mathcal{DOF}$ equals the set of the mesh nodes, in the case of second degree polynomials it is enriched with the edge midpoints, etc.

Later we will omit the subscript $p$ for simplicity. Most of the time the value of $p$ will not be important, otherwise it will be clear from the contest which polynomial degree will be used.

If the mesh contains only triangles/tetrahedrons (or parallelograms/parallelepipeds) it is very comfortable to define the basis functions over a reference element due to the fact that the above elements can be transformed into each other by using an affine linear mapping. We will discuss the case of triangles. For some comments on different meshes see Chapter 6

The main benefits of using the reference domain approach is clearly the relatively low computational costs of assembling the linear system. The values of the basis functions and the values of the gradients can be computed on $\Omega_0$ instead of the physical element and using the affine linear mapping we can compute the values at the quadrature points on the physical element. Moreover, we can compute the values on the reference domain and store it. For more details on reference domain calculations see [20].

Let us introduce the triangle $\Omega_0$ with nodes $(0,0), (1,0)$ and $(0,1)$. This will be called a reference triangle. Let us take an element $E \in \mathcal{T}_h$, this will be called a physical element, with nodes $(x_1, y_1), (x_2, y_2)$ and $(x_3, y_3)$, see Figure 2.3. The affine linear mapping $\mathcal{J}_E : \Omega_0 \to E$ will be an important tool to define the basis functions. It maps nodes to nodes, more precisely it maps $(0,0)$ to $(x_1, y_1)$, $(1,0)$ to $(x_2, y_2)$ and finally $(0,1)$ to $(x_3, y_3)$. Hence the mapping is defined by

$$\begin{pmatrix} x \\ y \end{pmatrix} = \mathcal{J}_E(\xi, \eta) = \begin{pmatrix} (x_2 - x_1)\xi + (x_3 - x_1)\eta + x_1 \\ (y_2 - y_1)\xi + (y_3 - y_1)\eta + y_1 \end{pmatrix}, \tag{2.11}$$

and the Jacobian $J_E$ is given by

$$J_E = \begin{pmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \end{pmatrix}.$$

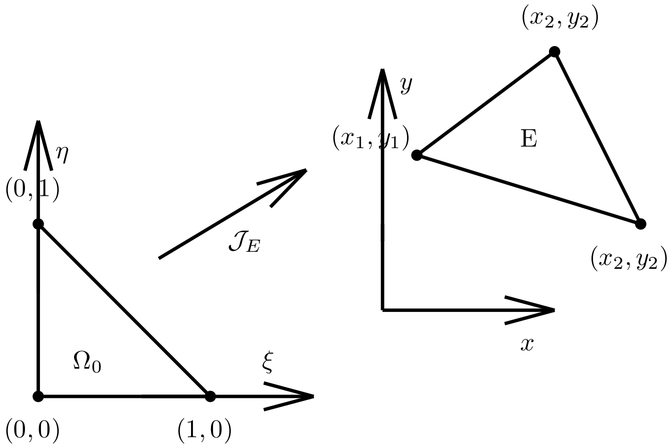

Figure 2.3: The reference element $\Omega_0$, the physical element $E$ and the mapping $\mathcal{J}_E$.

Let us denote the basis functions defined over $\Omega_0$ by $\Phi_i^{\Omega_0}$ (while $\Phi_i^E$ denotes the basis functions over $E$). The following formula establishes the relation between $\Phi_i^E$ and $\Phi_i^{\Omega_0}$

$$\Phi_i^{\Omega_0} = \Phi_i^E \circ \mathcal{J}_E.$$

**Remark 2.8.** *We note that the determinant of $J_E$ can easily be computed: $|\det(J_E)| = 2|E|$, where $|E|$ is the area of $E$, namely*

$$|E| = \int_E d\Omega = \int_{\Omega_0} |\det(J_E)|\, d\Omega_0 = |\det(J_E)| \int_{\Omega_0} d\Omega_0 = \frac{|\det(J_E)|}{2}$$

*using the integral transform.*

With these notations the basis functions for $V_{h,1}$ over $\Omega_0$ are:

- $\Phi_1^{\Omega_0}(\xi, \eta) = 1 - \xi - \eta$,
- $\Phi_2^{\Omega_0}(\xi, \eta) = \xi$,
- $\Phi_3^{\Omega_0}(\xi, \eta) = \eta$.

Throughout this work only polynomial degree $p = 1, 2, 3$ will be used. Naturally, higher order polynomial can be defined over the reference and the physical domain, see i.e. [43].

## 2.3 Convection dominated problem

The model equation (2.1) is called convection dominated if $|\mathbf{b}| \gg \varepsilon$. It can be shown that the classical Galerkin method has stability issues in convection dominated problems. This basically means that the stability of the bilinear form holds, but the corresponding stabilisation constant is small, and its reciprocal plays an important role in the convergence proof of the finite element method. For more details see [16] and Remark A.16 in Appendix A.

Let us consider the one dimensional problem

$$-\varepsilon u'' - u' = 0\,, \qquad \text{on } (0,1) \tag{2.12}$$

$$u(0) = 1, u(1) = 0\,, \tag{2.13}$$

where $\varepsilon$ as before is a small positive constant,
The analytical solution is

$$u(x) = \frac{\exp(-x/\varepsilon) - \exp(-1/\varepsilon)}{1 - \exp(-1/\varepsilon)}\,.$$

It is well known that the solution has a boundary layer type behaviour at $x = 0$ which means that as $\varepsilon \to 0$ the solution converge to $u(x) = 0$ on $(0, 1]$ with $u(0) = 1$ and there is a narrow region with huge derivatives. If $\varepsilon = 0$ than we have only convection that goes from right to left, because the convection speed is $-1$, and the solution would be the constant zero. Naturally, in this case we can have only one condition. However, as soon as $\varepsilon > 0$ we have a boundary value problem and the solution has to satisfy $u(0) = 1$ also, that is the reason of the boundary layer region.

The result with classical Galerkin can be seen in Figure 2.4. It is clear that the solution oscillates in the boundary layer region.

To overcome this we can introduce the so-called stabilising term

$$ST(u, v_h) = \sum_K \int_K (-\varepsilon \triangle u + \nabla \cdot (\mathbf{b}u) + cu - f)(\tau \mathbf{b} \cdot (\nabla v_h))\,,$$

and modify the bilinear form by $\tilde{B}_{ST}(u, v_h) = B(u, v_h) + ST(u, v_h)$. Naturally, the contribution $\sum_K \int_K f(\tau \mathbf{b} \cdot (\nabla v_h))$ can be added to the right hand side functional. The result of this method, the Streamline Diffusion Finite Element Method (SDFEM), can also be seen in Figure 2.4. We can conclude that the wiggles around the boundary layer have disappeared.

We can reformulate Problem Set 2.7, by introducing the Petrov-Galerkin function

$$\tilde{v}_h := v_h + \tau \mathbf{b} \cdot \nabla v_h. \tag{2.14}$$

Several terms can be put together using $\tilde{v}_h$, but not the second order term. Unfortunately

$$\int_K -\varepsilon \triangle u(\tau \mathbf{b} \cdot (\nabla v_h)) - \int_K \varepsilon \nabla u \nabla (\tau \mathbf{b} \cdot (\nabla v_h)) \neq 0\,,$$

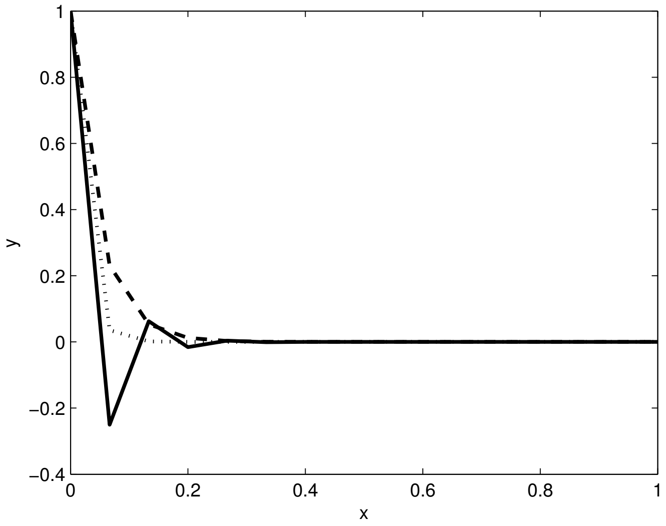

Figure 2.4: Classical and Streamline diffusion FEM solution compared to the exact solution 2.3 of the test equation (2.12)-(2.13) with $\varepsilon = 0.02$. Dotted line: exact solution, solid line: classical FEM, dashed line: SDFEM.

because there is a nonzero contribution on the boundary. However, if first degree polynomials are used in the discretisation, then $\tau\mathbf{b}\cdot(\nabla v_h)$ is a constant, therefore its derivatives are null. In this case we can fully replace $v$ and use only $\tilde{v}_h$. If we use higher order discretisation then we loose this property. Although, in Theorem 2.12 it will be shown that this simplification can be used, the only consequence is that the convergence order will not increase with the polynomial degree. For more possibilities see Chapter 6.

With these notations we can reformulate the stabilised problem, and we get the following: we seek the discrete solution $u_h \in V_h$ such that $B(u_h, \tilde{v}_h) = F(\tilde{v}_h)$ holds for all $\tilde{v}_h \in \tilde{V}_h$, where the bilinear and the linear forms are the same as in (2.9) and (2.10), respectively.

This approach leads us to the Streamline Upwind Petrov-Galerkin discretisation. Petrov-Galerkin discretisation means that the test and trial function are taken from two different spaces. In the classical Galerkin approach they are taken from the same space and they are continuous, in the discontinuous Galerkin approach they can be taken from two different spaces (not necessarily) but they are both discontinuous. The Petrov-Galerkin method is somehow in between of the two approaches. The test functions are taken from the continuous space $V_h$, however, the trial functions taken from $\tilde{V}_h$ are not continuous. The functions from this latter space can be characterised by (2.14).

**Remark 2.9.** *Due to the fact that $\tilde{v}_h$ can not be exactly integrated into the bilinear form in many papers this method is called Streamline Diffusion Finite Element Method. For the first order problems there is no such a problem, and in this case the method is called*

*Streamline Upwind Petrov-Galerkin Finite Element Method. Throughout this work we will use the two names as synonyms, as out first main goal for the future is to build $\tilde{v}_h$ properly into the bilinear form.*

**Remark 2.10.** *On the boundary the terms that contains $v$ are coming from a partial integration in which we use the approximation $\nabla \tilde{v} \approx \nabla v$, therefore we use the approximation $\tilde{v} \approx v$ on the boundary also.*

There are several other stabilized FEM such as Residual Distribution-Low Diffusion A or Bubble stabilised method. They are compared in [12] and the result was that for the current work SUPG provides better convergence rates. The reason for this will be presented in Section 3.4.

### 2.3.1 On the choice of $\tau$

It is important to note that in the case of diffusion-advection problems that there is an upper bound on the constant $\tau$, that appears in (2.14). This bound ensures that the bilinear form is coercive, therefore there is convergence of $u_{h,p}$ to $u$. For more details on the bound and the coercivity see Appendix A.2.

In addition let us introduce the cell Peclet number

$$Pe^h = \frac{|\mathbf{b}|h}{2\varepsilon},$$

where $h$ is the local mesh size. With this, the parameter $\tau$ can be redefined as

$$\tau = \frac{h}{2|\mathbf{b}|}\zeta(Pe^h).$$

To have stability and convergence for convection-diffusion problems; $\tau$ has to satisfy two asymptotic behaviours

$$\tau = O\left(\frac{h}{|\mathbf{b}|}\right) \quad \text{as } Pe^h \to \infty\,, \text{ (inviscid limit)} \tag{2.15}$$

$$\tau = O\left(\frac{h^2}{\varepsilon}\right) \quad \text{as } Pe^h \to 0\,. \text{ (diffusion limit)} \tag{2.16}$$

This implies that we have two constrains on $\zeta$

$$\begin{aligned} \zeta(Pe^h) &\to 1 \quad \text{as } Pe^h \to \infty\,,\\ \zeta &\approx Pe^h \quad \text{as } Pe^h \to 0\,. \end{aligned}$$

There are several possible choices on $\zeta$, see Appendix A.2.2, but throughout this work we will concentrate on the so-called doubly asymptotic expression which is

$$\zeta(Pe^h) = \min\{1, Pe^h/3\}\,.$$

### 2.3.2 Convergence of the method

**Theorem 2.11.** *Suppose that $u \in H^{p+1}(\Omega)$ and we solve the discrete problem with polynomials of degree $p$. In that case*

$$\|u - u_h\|_{L^2(\Omega)} \leq Ch^{p+1/2}|u|_{H^{p+1}(\Omega)}$$

As mentioned earlier we can simplify the calculations by neglecting the Laplacian in the stabilizing term. If we use first degree polynomials in the discretisation, the Laplacian of the numerical solution is zero, therefore, this approximation is exact in that case. If higher degree polynomials are used the higher order convergence is lost due to this approximation, as illustrated in Figure 2.5.

**Theorem 2.12.** *Suppose that $u \in H^{p+1}(\Omega)$ and we solve the discrete problem with polynomials of degree $p$, but we neglect the Laplacian in the stabilisation. In that case*

$$\|u - u_h\|_{L^2(\Omega)} \leq Ch^{3/2}|u|_{H^{p+1}(\Omega)}$$

**Remark 2.13.** *If the mesh and the equation is not too complicated then numerically the convergence rate is $p+1$, instead of $p+1/2$. This can be seen in Table 2.1 and on Figure 2.5. In the case of neglected Laplacian the convergence curves for $p = 2$ and $p = 3$ are indistinguishable.*

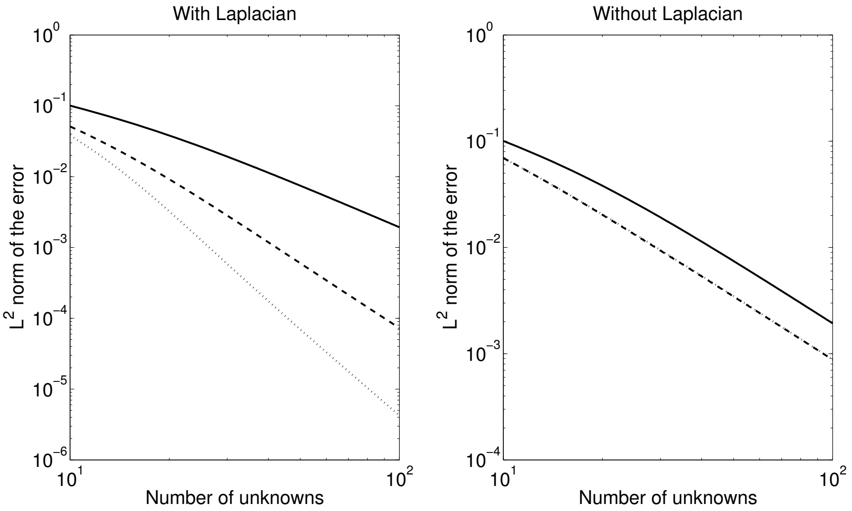

Figure 2.5: Left: convergence rates when the Laplacian is taken into account in the stabilisation term, right: convergence rates when the Laplacian is not taken into account. Solid line: $p = 1$, dashed line: $p = 2$, dotted line: $p = 3$. Test equation (2.12)-(2.13) with $\varepsilon = 0.02$.

| $p$ | With Laplacian | Without Laplacian |
|---|---:|---:|
| 1 | 1.907 | 1.907 |
| 2 | 3.042 | 1.960 |
| 3 | 4.118 | 1.976 |

Table 2.1: Convergence rates in 1D. Test equation (2.12)-(2.13) with $\varepsilon = 0.02$.

## 2.4 Testing with APOGEE

The main goal of this Research Master project is to implement diffusion terms into the APOGEE code and due to the fact that this code is two dimensional we start with the quasi two dimensional extension of the above mentioned problem. The domain is the unit square, and in the $y$ direction the boundary conditions are homogeneous Neumann.

More precisely the test case is

$$-\varepsilon\triangle u - \partial_x u - \partial_y u = 0 \quad \text{in } (0,1)^2 \tag{2.17}$$

$$u(0,y) = 1, u(1,y) = 0 \tag{2.18}$$

$$\varepsilon\partial_y u|_{y=0} = \varepsilon\partial_y u|_{y=1} = 0 \tag{2.19}$$

This means that the advection field is $\mathbf{b} = (-1,-1)^T$. The analytical solution is the simple extension of the previous one

$$u(x,y) = \frac{\exp(-x/\varepsilon) - \exp(-1/\varepsilon)}{1 - \exp(-1/\varepsilon)} .$$

Table 2.2 contains the results provided by APOGEE code. For the second order elements ($p = 2$) the diffusion term is not included in the stabilisation, therefore the convergence rate becomes similar to the $p = 1$ case. This is consistent with Theorem 2.12, and the convergence rate is between 1.5 and 2.

| $p$ | $h$ | $\lVert u - u_h \rVert_{L^2(\Omega)}$ | rate | $p$ | $h$ | $\lVert u - u_h \rVert_{L^2(\Omega)}$ | rate |
|---|---:|---|---:|---|---:|---|---:|
| 1 | 0.0625 | 0.101042362350649 | | 2 | 0.0625 | 0.144612469664143 | |
| 1 | 0.03125 | 0.059918204383782 | 0.75 | 2 | 0.03125 | 0.094061948349598 | 0.62 |
| 1 | 0.015625 | 0.019869457064809 | 1.59 | 2 | 0.015625 | 0.034532105780752 | 1.44 |
| 1 | 0.0078125 | 0.005360349677730 | 1.89 | 2 | 0.0078125 | 0.009807051552617 | 1.81 |

Table 2.2: Convergence rates for the 2D boundary layer equation. Test equation (2.17) - (2.19), $\varepsilon = 0.01$.

# Chapter 3

# Adjoint-based goal oriented a posteriori error estimation

## 3.1 Error estimation of finite element methods

Basically there are two different ways for error estimation of finite element methods: a priori and a posteriori. The a priori estimation is done analytically without doing any calculations. These error estimations were presented in the Section 2.3.2, they provide the convergence results. The big difference between a priori and a posteriori is that the a posteriori is done after the solution of the discrete problem and bounds the error, the difference between the exact (analytical) solution and the discrete solution, using only the data that are at hand, but not the exact solution $u$.

### 3.1.1 A posteriori error estimation

The construction of accurate a-posteriori error estimators for the finite element solution of PDE's is of great importance. Besides providing a reliable stopping criterion for the successive refinements, a-posteriori error estimation also gives a solid basis for adaptive finite element algorithms [15], [41]. From this point of view, local a-posteriori error estimates are of particular importance. For a general overview on a-posteriori error estimators we refer to [1, 16, 37, 46].

The starting point of many error estimation techniques is the residual-based a-posteriori error estimator, which provides an explicit formula for the error. The original idea in [4] has been generalized for several types of equations, such as advection-diffusion [47], convection-diffusion-reaction [48] and Maxwell equations [40]. Accordingly, explicit error estimators have been provided for nonconforming finite element methods [2] and uniform approaches have been developed [9]. Moreover, the estimation methodology can be extended for nonlinear problems, see, *e.g.* [10] and [30].

For the *implicit a-posteriori error estimators* Neumann type problems are formulated locally using the numerical solution at hand, and these are solved in certain local finite element spaces. In the simplest case, the boundary conditions for the local problems have been constructed with a simple averaging on element interfaces, however, there are some extensions ([25]) where the boundary conditions are achieved via a gradient averaging based method. To enforce the well-posedness of the local problems or enhance the quality of the estimators special equilibrated fluxes were defined and analysed ([5], [34]) using the results for the residual-based explicit error estimators. Though it seems to be an involved approach, it pays off to compute an accurate error estimator which provides local error bounds and is sensitive to the shape of the subdomain or to the mesh geometry. Implicit a-posteriori error estimators have been applied and analysed for elliptic boundary value problems (see an overview in [1]) and generalized for time-harmonic Maxwell equations [29].

Another approach is given by the *functional type* a-posteriori error estimates. These can provide both an upper and a lower bound for the exact error and are free of unknown constants (depending on the mesh geometry or interpolation inequalities). Usually, these estimates are independent of the numerical technique used to obtain approximate solutions, and they can be extended to nonlinear problems as well [33].

This approach is also important in CFD simulations. The functionals can describe the point value of the solution at a given point, or alternatively an integral over a boundary or over the domain. For example the lift/drag or moment coefficient can be described as an integral over the airfoil.

## 3.2 Adjoint problem

Suppose that we have to deal with the following problem

$$Lu = f \quad \text{in } \Omega \qquad Bu = g \quad \text{on } \Gamma \tag{3.1}$$

where $L$ denotes a linear differential operator in $\Omega$ and $B$ denotes a boundary operator on $\Gamma = \partial\Omega$.

For the model problem (2.1) - (2.3) we have

$$\begin{aligned} Lu &= -\varepsilon\triangle u + \nabla \cdot (\mathbf{b}u) + cu\,, \\ B|_{\Gamma_D} u &= B_1 u = u\,, \\ B|_{\Gamma_N} u &= B_2 u = \varepsilon\mathbf{n} \cdot \nabla u\,. \end{aligned}$$

Using the work of Hartmann [21], we formalize the target function as

$$J(u) = (j_\Omega, u)_\Omega + (j_\Gamma, Cu)_\Gamma\ , \tag{3.2}$$

where $j_\Omega \in L^2(\Omega)$, $j_\Gamma \in L^2(\Gamma)$, $(\cdot,\cdot)_T$ is the inner product on $L^2(T)$ for arbitrary domain/boundary $T$, i.e., $(u,v)_T = \int_T uv\, dT$, $C$ is a differential operator defined over $\Gamma$, which for the model problem (2.1) - (2.3) are given by

$$\begin{aligned} C|_{\Gamma_D} u &= C_1 u = \varepsilon\mathbf{n} \cdot \nabla u\,, \\ C|_{\Gamma_N} u &= C_2 u = u\,. \end{aligned}$$

We say that the target functional (3.2) is compatible with primal problem (3.1) if there are operators $L^*$, $B^*$, $C^*$, such that

$$(Lu,z)_\Omega + (Bu, C^*z)_\Gamma = (u, L^*z)_\Omega + (Cu, B^*z)_\Gamma\,. \tag{3.3}$$

The operators $L^*$, $B^*$ and $C^*$ are the adjoint operators of $L$, $B$ and $C$, respectively. Moreover, assuming that (3.3) holds the adjoint problem associated to (3.3) and (3.2) is given by

$$L^*z = j_\Omega \quad \text{in } \Omega \qquad B^*z = j_\Gamma \quad \text{on } \Gamma\,. \tag{3.4}$$

By applying partial integration to the left hand side of (3.3) we can identify the operators $L^*$, $B^*$ and $C^*$, giving for the model problem (2.1) - (2.3)

$$\begin{aligned}
L^*u &= -\varepsilon\triangle u - \nabla\cdot(\mathbf{b}u) + cu\,,\\
B^*|_{\Gamma_D}u &= B_1^*u = -u\,,\\
B^*|_{\Gamma_N}u &= B_2^*u = \varepsilon\mathbf{n}\cdot\nabla u\,,\\
C^*|_{\Gamma_D}u &= C_1^*u = -\varepsilon\mathbf{n}\cdot\nabla u\,,\\
C^*|_{\Gamma_N}u &= C_2^*u = u\,.
\end{aligned}$$

For the readers' convenience we will work this out for the Laplace case. To do that, suppose that $\mathbf{b} = \mathbf{0}$, $c = 0$, $\varepsilon = 1$, and suppose that the problem is subject to full Dirichlet boundary conditions. In that case, we multiply $Lu$ by $z$, integrate on $\Omega$ and apply Green's Theorem twice

$$\begin{aligned}
(Lu,z)_\Omega &= -\int_\Omega \triangle uz \,\mathrm{d}\mathbf{x} = \int_\Omega \nabla u\cdot\nabla z\,\mathrm{d}\mathbf{x} - \int_\Gamma z\nabla u\cdot\mathbf{n}\,\mathrm{d}s\\
&= -\int_\Omega u\triangle z\,\mathrm{d}\mathbf{x} - \int_\Gamma z\nabla u\cdot\mathbf{n}\,\mathrm{d}s + \int_\Gamma u\nabla z\cdot\mathbf{n}\,\mathrm{d}s\\
&= (u, L^*z)_\Omega + (Cu, B^*z)_\Gamma - (Bu, C^*z)_\Gamma\,.
\end{aligned}$$

**Remark 3.1.** *It is important to examine the behaviour of the different terms. First of all, it has to be emphasized that the convective term has changed sign. It means that in the adjoint problem the convection acts in the opposite direction. If pure convection problems are examined it means that the inlet and outlet boundaries are swapped. The viscous term stays as it was, which physically means that the viscous effects have the same behaviour in both problems (in other words, the Laplace operator is self adjoint).*

Suppose that the primal problem (3.1) is discretized by the method described in Chapter 2 and the discrete problem is given by

$$B(u_h, \tilde{v}_h) = F(\tilde{v}_h) \qquad \forall \tilde{v}_h \in \tilde{V}_h\,, \tag{3.5}$$

where $B(\cdot,\cdot)$, $F(\cdot)$, $V_h$ and $\tilde{V}_h$ are described in Chapter 2. The problem is said to be consistent if the equality (3.5) holds for the exact solution $u$ also, or in other worlds

$$B(u, \tilde{v}_h) = F(\tilde{v}_h) \qquad \forall \tilde{v}_h \in \tilde{V}_h\,. \tag{3.6}$$

The dual (adjoint) discrete problem reads as follows.

**Problem Set 3.2.** *Seek $z_h \in \tilde{V}_h$ such that for all $w_h \in V_h$ the following holds*

$$B^*(z_h, w_h) \equiv B(w_h, z_h) = J(w_h) \qquad \forall w_h \in V_h\,, \tag{3.7}$$

*where the bilinear form $B(\cdot,\cdot)$ is the same as for the primal problem. The bilinear form $B^*(\cdot,\cdot)$ is the so-called adjoint bilinear form.*

Using the above notations, the discretisation is said to be adjoint consistent if the exact solution $z \in V$ of the dual problem satisfies 3.7, i.e.

$$B(w_h, z) = J(w_h) \qquad \forall w_h \in V_h\,.$$

Therefore, adjoint consistency means that the discretisation of the dual problem is a consistent discretisation of the continuous adjoint problem.

### 3.2.1 Linear target quantities

In the following we give an overview of the main linear target functionals that can be found in the literature.

**Weighted average functional:** using a volume weight function $j_\Omega \in L^2(\Omega)$ and $j_\Gamma = 0$, the weighted average of the solution can be calculated by

$$J(u) = \int_\Omega j_\Omega u \,\mathrm{d}\mathbf{x}\,.$$

The corresponding continuous adjoint problem has $j_\Omega$ as a source and zero as boundary condition on both the Dirichlet and the Neumann parts.

**Boundary flux functional:** using a boundary weight function $j_\Gamma \in L^2(\Gamma)$ and $j_\Omega = 0$, the weighted average of the viscous flux on a Dirichlet boundary can be computed by

$$J(u) = \int_{\Gamma_D} j_\Gamma \varepsilon \nabla u \cdot \mathbf{n} \,\mathrm{d}s\,.$$

The corresponding continuous adjoint problem has zero source and Neumann boundary condition, however, on the Dirichlet part the boundary condition is $-j_\Gamma$. Naturally, the weight function can be zero at some part of the Dirichlet boundary, if only some part of it is of interest.

**Boundary value functional:** using a boundary weight function $j_\Gamma \in L^2(\Gamma)$ and $j_\Omega = 0$, the weighted value of the solution on a Neumann boundary can be computed by

$$J(u) = \int_{\Gamma_N} j_\Gamma u \,\mathrm{d}s\,.$$

The corresponding continuous adjoint problem has zero source and Dirichlet boundary condition, however, on the Neumann part the boundary condition is $j_\Gamma$. Similarly as above, the weight function can be zero at some part of the Neumann boundary, if only some part of it is of interest.

**Point value functional:** we assume a continuous solution $u$ at a given point $\mathbf{x_0}$, and we set $j_\Omega = \delta_{\mathbf{x_0}}$ and $j_\Gamma = 0$, where $\delta_{\mathbf{x_0}}$ is the Dirac delta at the given point. With this, the solution at $\mathbf{x_0}$ can be set as a target quantity

$$J(u) = \int_\Omega j_\Omega u \, \mathrm{d}\mathbf{x} = u(\mathbf{x_0}).$$

The corresponding continuous adjoint problem has $\delta_{\mathbf{x_0}}$ as a source and zero boundary condition on both the Dirichlet and the Neumann parts.

**Remark 3.3.** *The weak adjoint solution, provided by the last example, does not supply a regular distribution, because of the Dirac delta, furthermore, $z$ does not belong to any Sobolev space, and not even to $L^2(\Omega)$. To overcome this [1] suggests to use a mollified functional, by considering a nonnegative function $\psi_{\mathbf{x_0}} \in L^1(\Omega)$ with a ball support around $\mathbf{x_0}$, and with the restriction that the integral of $\psi_{\mathbf{x_0}}$ over this ball is 1. Then the target functional is given by*

$$J(u) = \int_\Omega \psi_{\mathbf{x_0}} u \, d\mathbf{x}\,.$$

Naturally, for viscous problems it is an interesting question to use this approach for $\mathbf{x_0}$ that lies in the boundary layer. Some results of this will be shown later.

## 3.3 A posteriori error estimation for target functionals

The consistency (3.6) has a very important consequence, the so-called Galerkin orthogonality

$$B(u - u_h, \tilde{v}_h) = 0 \qquad \forall \tilde{v}_h \in \tilde{V}_h\,.$$

The goal oriented error estimation is based on the following equalities (see [44])

$$\begin{aligned} J(u) - J(u_h) &= J(u - u_h) && \text{(linearity)} \\ &= B(u - u_h, z) && \text{(adjoint)} \\ &= B(u - u_h, z - z_h) && \text{(Galerkin orthogonality)} \\ &= F(z - z_h) - B(u_h, z - z_h) && \text{(consistency)} \\ &= \mathcal{R}(u_h, z - z_h)\,, && \text{(residual)} \end{aligned}$$

where $\mathcal{R}(u_h, z - z_h) = F(z - z_h) - B(u_h, z - z_h)$ is called the residual.

Taking the absolute value and using the triangle inequality

$$\begin{aligned}|J(u) - J(u_h)| &= |\mathcal{R}(u_h, z - z_h)| \\ &= \left| \sum_{\kappa \in \mathcal{T}_h} \eta_\kappa \right| \leq \sum_{\kappa \in \mathcal{T}_h} |\eta_\kappa|,\end{aligned}$$

where $\eta_\kappa$ is the elementwise residual. For later purposes let us introduce the following notations

$$|\mathcal{R}_\Omega| = \left| \sum_{\kappa \in \mathcal{T}_h} \eta_\kappa \right| , \qquad \mathcal{R}_{|\Omega|} = \sum_{\kappa \in \mathcal{T}_h} |\eta_\kappa| \ .$$

Therefore $|\mathcal{R}_\Omega|$ is the absolute value of the error and $\mathcal{R}_{|\Omega|}$ is the upper bound. We can define the efficiency of these estimators as

$$\theta_1 = \frac{|\mathcal{R}_\Omega|}{|J(u) - J(u_h)|} \quad \text{and} \quad \theta_2 = \frac{\mathcal{R}_{|\Omega|}}{|J(u) - J(u_h)|} \, ,$$

hence optimal efficiency corresponds to $\theta_1 = \theta_2 = 1$.

To get some approximation of the term $z - z_h$ the dual problem is solved in an enriched space to get $z_h^2$, and $z - z_h \approx z_h^2 - z_h$. This is called Type I estimation ([44]). This enriched space in the present work is $V_{h,p+1}$, i.e., the mesh is unchanged but the polynomial degree is increased by one. Another possible choice for the enriched space can be $V_{h/2,p}$, which means that a uniform refinement is applied to the mesh, but the polynomial degree is not changed.

Let us decompose the error representing formula

$$\begin{aligned}|J(u) - J(u_h)| = |\mathcal{R}(u_h, z - z_h)| &\leq |\mathcal{R}(u_h, z_h^2 - z_h)| + |\mathcal{R}(u_h, z - z_h^2)| \\ &\leq \mathcal{R}_{|\Omega|} + |\mathcal{R}(u_h, z - z_h^2)| \, .\end{aligned}$$

It has been shown in many papers, see i.e. [6], that the second term is negligible with respect to the first.

Type II estimation avoids computation of $z$, i.e. the residual $\mathcal{R}(u_h, z - z_h)$ is bounded from above by terms that contain only the numerical solution $u_h$. However, this approach uses some constants in the upper bound and in most cases the order of these constants are not known. Indeed, Cauchy-Schwarz inequality $\eta_\kappa$ can be bounded as

$$\eta_\kappa \leq \|\mathcal{R}(u_h)\|_\kappa \|z - z_h\|_\kappa \, .$$

If $z \in H^k(\Omega)$ for some $k$ then we can bound $\|z - z_h\| \leq C\|z\|_{H^k(\Omega)}$. Finally, assuming that $\|z\|_{H^k(\Omega)} \leq C_{stab}$ we can get rid of the adjoint solution and only the primal solution $u_h$ appears in the upper bound. There are several drawbacks of this approach: first of all, the constants that appear in the estimation are unknown, as it was mention earlier. Moreover, this bound is independent of the target quantity, therefore it will reveal the error of the primal problem everywhere over the domain, not only at the parts where it is relevant for the given target quantity. Therefore, the bound is quite pessimistic and we get extreme over-estimations.

For this estimator we will introduce the following short notation and efficiency

$$\mathcal{R} = \mathcal{R}(u_h, z - z_h) \quad \text{and} \quad \theta = \frac{\mathcal{R}}{|J(u) - J(u_h)|}\,.$$

These methods have been applied to several physical phenomena but in combination with discretisation techniques that are different from streamline diffusion finite element method, [21, 23, 17, 44].

**Remark 3.4.** *Throughout this work the primal problem will be solved by first degree polynomials and the adjoint problem will be solved once with first degree and once with second degree polynomials.*

**Theorem 3.5.** *Suppose that the discretisation is consistent, the bilinear and the linear forms have the properties from Definition 2.5. Suppose that $j_\Omega, j_\Gamma$ are smooth functions and the adjoint solution is also smooth ($z \in H^{k+1}(\Omega)$). Then the following results hold:*

1. *If the discretisation together with the target functional $J(\cdot)$ is adjoint consistent, then there is a constant $C > 0$ such that*

$$|J(u) - J(u_h)| \leq Ch^{r+\bar{r}}|u|_{H^{p+1}(\Omega)}|z|_{H^{p+1}(\Omega)} \qquad \forall u \in H^{p+1}(\Omega)\,.$$

2. *If the discretisation is adjoint inconsistent, then*

$$|J(u) - J(u_h)| \leq Ch^{r}|u|_{H^{p+1}(\Omega)} \qquad \forall u \in H^{p+1}(\Omega)\,.$$

*Proof.* 1. If both the primal and the adjoint problems are consistent then we can use the equality $J(u) - J(u_h) = B(u - u_h, z - z_h)$. Taking the absolute value, using the continuity of the bilinear form and using the convergence results we get

$$\begin{aligned} |J(u) - J(u_h)| &= |B(u - u_h, z - z_h)| \\ &\leq C_c \, \lVert\!| u - u_h |\!\rVert \, \lVert\!| z - z_h |\!\rVert \\ &\leq Ch^{r}|u|_{H^{p+1}(\Omega)}h^{\bar{r}}|z|_{H^{p+1}(\Omega)} \\ &= h^{r+\bar{r}}|u|_{H^{p+1}(\Omega)}|z|_{H^{p+1}(\Omega)}\,. \end{aligned}$$

2. If the adjoint problem is inconsistent then a mesh dependent adjoint problem can be defined by $B_h(w, \phi) = J(w)$, $\forall w \in V_h$. We cannot expect any smoothness on this $\phi$, however, we can define $P_h\phi$ as the projection of $w$ to the space $V_h$. Combining this projection with Galerkin orthogonality we have

$$\begin{aligned} J(u) - J(u_h) &= J(u - u_h) && \text{(linearity)} \\ &= B(u - u_h, w) && \text{(adjoint)} \\ &= B(u - u_h, w - P_h w)\,. && \text{(Galerkin orthogonality)} \end{aligned}$$

As in the first part of the proof, taking the absolute value, using the continuity of the bilinear form and using the convergence results (only on $u - u_h$) we get

$$\begin{aligned}|J(u) - J(u_h)| &= |B(u - u_h, w - P_h w)| \\ &\le C_c \, ⦀u - u_h⦀ \, ⦀w - P_h w⦀ \\ &\le C h^r |u|_{H^{p+1}(\Omega)} \, ⦀w - P_h w⦀ \\ &= C' h^r |u|_{H^{p+1}(\Omega)} \, .\end{aligned}$$

Due to the lack of adjoint consistency we cannot gain an extra rate of convergence on $w - P_h w$.

□

The connection between the primal/dual continuous/discrete problems can be summarized in Figure 3.1.

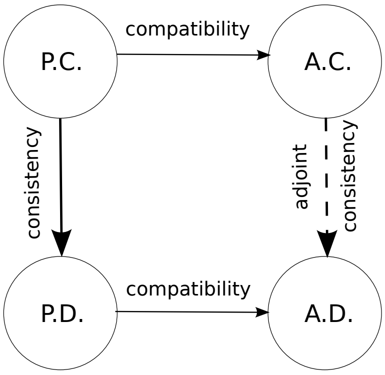

Figure 3.1: Connection between the different problems. First letter: P - primal, A - adjoint, second letter: C - continuous, D - discrete.

## 3.4 Comparison of different stabilised methods

It will be shown that SUPG provides better convergence rates for the target based error estimation than Residual Distribution-Low Diffusion A (RD-LDA) or Bubble stabilised method (BUBBLE).

Let us recall the definition of the RD-LDA and BUBBLE test functions, according to [12]. For the simplicity we will denote the streamline upwind stabilized test functions by $\tilde{v}_{SU}$.

**RD-LDA**

This technique has a long history at VKI, see i.e. [38, 14]. The test functions are defined as

$$\tilde{v}_{RDA} = \alpha \frac{k^+}{\sum_l k_l^+} \, , \quad k = L_{adv} v \, , \quad k^+ = \max\{k, 0\} \, , \, , \tag{3.8}$$

where $\alpha$ is a parameter, $L_{adv}$ is the advection part of the differential operator, $v$ are the standard FEM basis functions and the summation goes over all the basis functions that are corresponding to the same physical element.

**BUBBLE**
Bubble function schemes have since long been developed as an alternative of Galerkin Least Square-stabilized finite element methods for stabilizing the numerical solution provided by the Galerkin method [7, 18]. The test functions are defined as

$$\tilde{v}_B = v + \alpha b(\mathbf{x}) \left( \frac{k^+}{\sum_l k_l^+} - v \right) ,$$

where we used the notations of (3.8), furthermore, $b(\mathbf{x})$ is a bubble function on the corresponding element, which means that $b(\mathbf{x}) = 0$ on the boundary of the element, for example $b(\mathbf{x}) = \prod_{i=1}^{3} v_i(\mathbf{x})$.

In Appendix C.1 it will be shown, that it is impossible to interpolate any nonconstant polynomial with the RD basis function exactly, therefore the convergence rate of the interpolation in the $L^2$ norm is at most 1. An interpolation will be shown for the streamline upwind functional using which every polynomials of degree $p-1$ can be interpolated exactly, therefore it will be shown, that the convergence rate of the interpolation in the $L^2$ norm is at least $p$, however, numerically $p+1$ could be achieved, which means that there is some space for further investigation.

## 3.5 Adaptation

Suppose that we want to guarantee that the error is smaller than a given tolerance TOL. To achieve this a simple algorithm can be applied:

1. construct the initial mesh and FEM space $V_h$,
2. compute the solution $u_h \in V_h$
3. solve the adjoint problems to get $z_h, z_h^2$,
4. compute the error indicators $(\eta_\kappa)$,
5. check if the error is small enough, i.e. stop if $\sum |\eta_\kappa| \leq TOL$
6. otherwise refine the mesh where it is needed according to $\eta_\kappa$, and create the new mesh and the new FEM space and GOTO 2.

Let us recall the possible refinement strategies

**Local tolerance criterion**
Suppose that the mesh contains $N$ element and mark for refinement the element for which the following holds

$$|\eta_\kappa| \geq \frac{TOL}{N} .$$

However, in adaptive mesh refinement algorithms some coarsening is required to avoid the too fast growth of the number of element. This means, that where the error is small the mesh size can be enlarged. In the current framework it means that triangles for which

$$|\eta_\kappa| \le \theta \frac{TOL}{N}$$

are marked for coarsening. The parameter $\theta$ is user defined, $0 < \theta \ll 1$.

**Fixed fraction criterion**

Another way of refining and coarsening is the so-called fixed fraction criterion. In this method we sort the elements according to $|\eta_\kappa|$ and refine the upper $N_r\%$ of the elements and coarsen the lower $N_c\%$ of the elements. In the following a similar method will be used, but instead of sorting the local error element-wisely, a point-wise characteristic size $\eta_p$ will be defined and the nodes of the mesh will be sorted and flagged for the new mesh.

**Remeshing**

Naturally a completely new mesh can be generated, using the local error indicator as some kind of mesh density requirement. This approach can be used for 2D problems but for 3D problems its computational costs are too high. Even in 2D it has an enormous drawback: using the interpolation of the old solution from the old mesh to the new mesh requires a significant amount of computation.

## 3.6 Numerical examples

First of all let us examine the test case that is quite similar to (2.17) - (2.19) but it is subject to fully Dirichlet boundary condition

$$-0.01\triangle u - \partial_x u - \partial_y u = 0 \quad \text{in } (0,1)^2 \tag{3.9}$$

$$u = g \quad \text{on } \partial(0,1)^2 \tag{3.10}$$

where $g$ is set such that the exact solution is

$$u(x,y) = \frac{\exp(-x/\varepsilon) - \exp(-1/\varepsilon)}{1 - \exp(-1/\varepsilon)}\,.$$

The target functional is the point value of the solution inside the boundary layer, at $\mathbf{x_0} = (0.01, 0.5)$ and we apply the mollified functional used in the last example from 3.2.1. More precisely:

$$J(u) = \int_\Omega \psi_{\mathbf{x_0}} u \,\mathrm{d}\mathbf{x} = 0.367879441171442\,.$$

Table 3.1 and 3.2 contain the result. The final meshes can be seen in Figure 3.2. From this we can conclude that the residual based adaptation resolves the whole boundary layer, while, the adjoint based approach refines only on a smaller region, namely in the vicinity of $\mathbf{x_0}$. The isolines of the exact primal solution and the adjoint solution can be found in Figure 3.3. The support of the adjoint solution is basically the characteristic that goes through $\mathbf{x_0}$. Mathematically characteristics are existing only on the inviscid limit, but the weak viscosity does not influence the point value so strongly and some trace of the characteristics can be seen in the adjoint solution. For pure convection problem it would be only a peak along the characteristic, but for the convection-diffusion case it is smeared in the transversal direction. The smearing increases as the distance to the target point $\mathbf{x_0}$ increases.

| NT | NLS | $\|J(u) - J(u_h)\|$ | $\|R_\Omega\|$ | $\theta_1$ | $R_{\|\Omega\|}$ | $\theta_2$ |
|---:|---:|---:|---:|---:|---:|---:|
| 385 | 704 | $4.895 \cdot 10^{-1}$ | $3.509 \cdot 10^{0}$ | 7.17 | $6.440 \cdot 10^{0}$ | 13.16 |
| 466 | 845 | $3.888 \cdot 10^{-1}$ | $7.669 \cdot 10^{-1}$ | 1.97 | $8.763 \cdot 10^{-1}$ | 2.25 |
| 564 | 1024 | $1.873 \cdot 10^{-1}$ | $1.506 \cdot 10^{0}$ | 8.04 | $1.552 \cdot 10^{0}$ | 8.29 |
| 670 | 1220 | $5.688 \cdot 10^{-2}$ | $1.911 \cdot 10^{-1}$ | 3.36 | $2.224 \cdot 10^{-1}$ | 3.91 |
| 842 | 1534 | $1.958 \cdot 10^{-2}$ | $1.190 \cdot 10^{-1}$ | 6.08 | $1.267 \cdot 10^{-1}$ | 6.47 |
| 1032 | 1883 | $9.397 \cdot 10^{-3}$ | $8.926 \cdot 10^{-2}$ | 9.50 | $9.340 \cdot 10^{-2}$ | 9.94 |
| 1189 | 2175 | $7.175 \cdot 10^{-3}$ | $8.358 \cdot 10^{-2}$ | 11.65 | $8.594 \cdot 10^{-2}$ | 11.98 |
| 1393 | 2553 | $6.265 \cdot 10^{-3}$ | $8.346 \cdot 10^{-2}$ | 13.32 | $8.514 \cdot 10^{-2}$ | 13.59 |
| 1592 | 2936 | $5.739 \cdot 10^{-3}$ | $8.244 \cdot 10^{-2}$ | 14.36 | $8.354 \cdot 10^{-2}$ | 14.56 |
| 1826 | 3398 | $5.139 \cdot 10^{-3}$ | $8.103 \cdot 10^{-2}$ | 15.77 | $8.194 \cdot 10^{-2}$ | 15.95 |

Table 3.1: Type I (adjoint based) estimation for point value ($\mathbf{x_0} = 0.01, 0.5$) in the boundary layer. Test equation (3.9) - (3.10)

| NT | NLS | $\|J(u) - J(u_h)\|$ | $\mathcal{R}$ | $\theta$ |
|---:|---:|---:|---:|---:|
| 385 | 704 | $4.895 \cdot 10^{-1}$ | $3.466 \cdot 10^{0}$ | 7.08 |
| 448 | 809 | $3.879 \cdot 10^{-1}$ | $2.635 \cdot 10^{0}$ | 6.79 |
| 551 | 985 | $2.314 \cdot 10^{-1}$ | $2.028 \cdot 10^{0}$ | 8.77 |
| 659 | 1178 | $1.843 \cdot 10^{-1}$ | $1.896 \cdot 10^{0}$ | 10.29 |
| 814 | 1438 | $7.166 \cdot 10^{-2}$ | $1.886 \cdot 10^{0}$ | 26.32 |
| 967 | 1736 | $5.154 \cdot 10^{-2}$ | $1.860 \cdot 10^{0}$ | 36.09 |
| 1232 | 2195 | $2.887 \cdot 10^{-2}$ | $1.866 \cdot 10^{0}$ | 64.62 |
| 1520 | 2714 | $2.626 \cdot 10^{-2}$ | $1.869 \cdot 10^{0}$ | 71.17 |
| 1936 | 3538 | $1.663 \cdot 10^{-2}$ | $1.854 \cdot 10^{0}$ | 111.45 |
| 2539 | 4678 | $1.398 \cdot 10^{-2}$ | $1.848 \cdot 10^{0}$ | 132.21 |

Table 3.2: Type II (residual based) estimation for point value ($\mathbf{x_0} = 0.01, 0.5$) in the boundary layer. Test equation (3.9) - (3.10).

For further results see Appendix B.1.

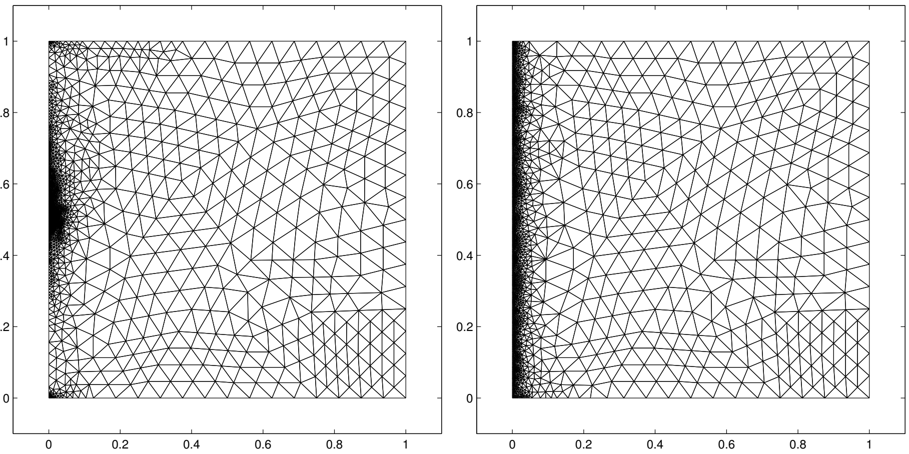

Figure 3.2: Meshes for linear boundary layer point value. Left: adjoint based refined mesh with 1826 triangles (3398 unknowns) $|J(u) - J(u_h)| = 5.139 \cdot 10^{-3}$, right: residual based refined mesh with 2539 triangles (4678 unknowns) $|J(u) - J(u_h)| = 1.398 \cdot 10^{-2}$.

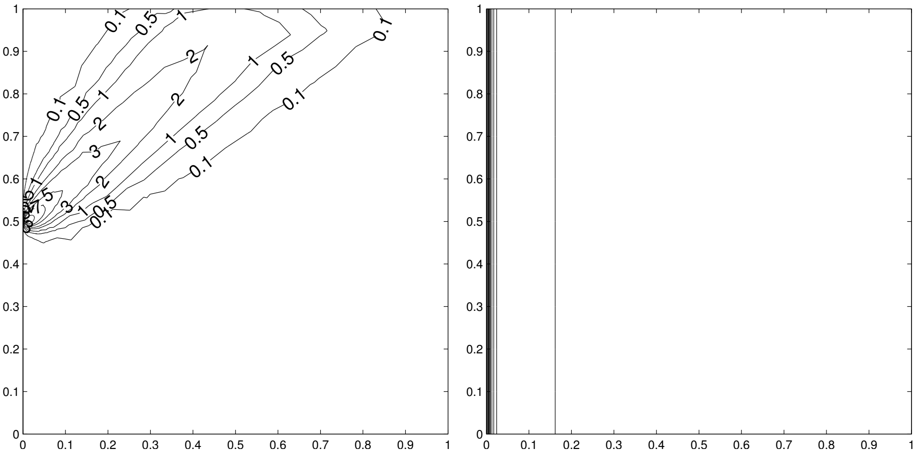

Figure 3.3: Left: adjoint solution, right: exact primal solution both for the boundary layer point value example.

# Chapter 4

# Generalisation to systems and nonlinear problems

In the previous Chapters only linear scalar problems and linear target functionals have been studied. However, the future aim of the project is to apply the error estimation for the compressible Navier-Stokes equations, which is a nonlinear coupled system and some of the possible target quantities are also nonlinear.

In this chapter the intermediate steps will be made. First we will extend our work to systems, then we study the nonlinear problems in general. The system case will be examined with a coupled system. The nonlinear scalar case will be illustrated with the viscous Burgers' equation because it has the same nonlinearity as the Navier-Stokes equations.

First we will reformulate the linear problem to introduce the notations for the more general problems.

## 4.1 Generalisation to scalar conversation laws

Let us rewrite (2.1) into the following form

$$-\nabla \cdot \mathcal{F}^v(u, \nabla u) + \nabla \cdot \mathcal{F}^c(u) + \mathcal{F}^r(u) = \mathcal{S}(u), \tag{4.1}$$

with $\mathcal{F}^v(u, \nabla u) = \varepsilon \nabla u$ the viscous flux, $\mathcal{F}^c(u) = \mathbf{b}u$ the convection flux, $\mathcal{F}^r(u) = cu$ the reaction flux and $\mathcal{S}(u) = f$ the source. In (2.1) $\mathcal{S}$ and $\mathcal{F}^v$ do not depend on $u$.

Using these notation the weak form can be represented as

**Problem Set 4.1.** *Seek $u_h \in V_h$ such that $\forall \tilde{v}_h \in \tilde{V}_h$*

$$\mathcal{N}_h(u_h, \tilde{v}_h) = 0, \tag{4.2}$$

*where*

$$\mathcal{N}_h(u,v) = \int_\Omega \nabla\tilde{v}\cdot\mathcal{F}^v(u,\nabla u)\ d\mathbf{x} + \int_\Omega \tilde{v}\left(\nabla\cdot\mathcal{F}^c(u)+\mathcal{F}^r(u)\right)\ d\mathbf{x} - \int_\Omega \tilde{v}\mathcal{S}(u)\ d\mathbf{x}$$
$$- \int_{\Gamma_N} v g_N\ ds - \int_{\Gamma_D} v\mathcal{F}^v(u,\nabla u)\ ds - \int_{\Gamma_D} (u-g_D)\left[(\mathcal{F}^v)'(v,\nabla v)+\alpha v\right]\ ds\,.$$

*For later purposes let us generalise this as*

$$\mathcal{N}_h(u,v) = \int_\Omega \nabla\tilde{v}\cdot\mathcal{F}^v(u,\nabla u)\ d\mathbf{x} + \int_\Omega \tilde{v}\left(\nabla\cdot\mathcal{F}^c(u)+\mathcal{F}^r(u)\right)\ d\mathbf{x} - \int_\Omega \tilde{v}\mathcal{S}(u)\ d\mathbf{x}$$
$$- \int_{\Gamma} v u_N(u)\ ds - \int_{\Gamma} (u-u_D(u))\left[(\mathcal{F}^v)'(v,\nabla v)+\alpha v\right]\ ds\,, \tag{4.3}$$

*where*

$$u_N(u) = \begin{cases} g_N & \textit{on } \Gamma_N \\ \mathcal{F}^v(u,\nabla u) & \textit{on } \Gamma_D \end{cases}, \qquad u_D(u) = \begin{cases} u & \textit{on } \Gamma_N \\ g_N & \textit{on } \Gamma_D \end{cases}.$$

## 4.2 System of equations

Let us consider the following example, where the assumptions on the coefficient functions are similar to what we have for (2.1), but now every function is vector valued and we suppose that the coordinate functions are belonging to the proper spaces. The equations in $\Omega\subset\mathbb{R}^2$ are

$$-\varepsilon\triangle u_1 - \frac{1}{2}\partial_x u_2 + \partial_y u_1 + c_1 u_1 = f_1 \tag{4.4}$$
$$-\varepsilon\triangle u_2 - \frac{1}{2}\partial_x u_1 + \partial_y u_2 + c_2 u_2 = f_2 \tag{4.5}$$

subject to the following boundary conditions

$$u_1 = g_D^1 \qquad u_2 = g_D^2 \qquad \text{on } \Gamma_D\,, \tag{4.6}$$
$$\varepsilon\nabla u_1\cdot\mathbf{n} = g_N^1 \qquad \varepsilon\nabla u_2\cdot\mathbf{n} = g_D^2 \qquad \text{on } \Gamma_N\,. \tag{4.7}$$

Using the vector $\mathbf{u} = [u_1,u_2]^T$ this can be rewritten in the flux formulation (4.1). The viscous and convective fluxes are

$$\mathcal{F}^v(\mathbf{u},\nabla\mathbf{u}) = \begin{bmatrix} \varepsilon\nabla u_1 \\ \varepsilon\nabla u_2 \end{bmatrix}, \mathcal{F}^c(\mathbf{u}) = \mathbf{f}_1^c(\mathbf{u})\mathbf{1}_x + \mathbf{f}_2^c(\mathbf{u})\mathbf{1}_y\,, f_1^c(\mathbf{u}) = \begin{bmatrix} -\frac{1}{2}u_2 \\ -\frac{1}{2}u_1 \end{bmatrix}, f_2^c(\mathbf{u}) = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix},$$

where $\mathbf{1}_x$ and $\mathbf{1}_y$ are the unit vectors in the $x$ and $y$ dierctions, while the reaction flux and the source are

$$\mathcal{F}^r(\mathbf{u}) = \begin{bmatrix} c_1u_1 \\ c_2u_2 \end{bmatrix}, \qquad \mathcal{S}(\mathbf{u}) = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix}.$$

In the APOGEE code systems are handled a little bit differently than in many FEM approaches, due to the fact, that APOGEE handles also the RD-LDA discretisation, in which the test functions are matrices. Therefore all test functions are stored as matrices in the code, naturally standard FEM functions are diagonal matrices. Suppose that we are working with a system of $q$ equations, in this case $\mathbf{v} \in \mathbb{R}^{q\times q}$ and $\mathbf{v}_{i,j} = \delta_{i,j}$, where $\delta_{i,j}$ sands for the Kronecker delta, and the corresponding coefficient is not a scalar, but a vector with $q$ component. Therefore, the operator $\mathcal{N}_h$ is vector valued, and its derivative, which will be important for the nonlinear case, is matrix valued.

Streamline Upwind or Streamline Diffusion stabilization means that the vector valued FEM test function is modified. Its gradient is multiplied with the convection speed and this added to the test function with some weight $\tau$

$$\tilde{\mathbf{v}} = \mathbf{v} + \tau(\mathcal{F}^c)'[\mathbf{u}](\nabla \mathbf{v}) \quad \forall \mathbf{v} \in V_h^q\,,$$

where $V_h^q$ contains the FEM basis functions with $q$ coordinate functions, where $q$ is the number of equations. Basically this means, that all coordinate functions $v_i \in V_h$. Similarly, $\tilde{V}_h^q$ denotes the corresponding SUPG space. Finally, $(\mathcal{F}^c)'[\mathbf{u}]$ is the Frechet derivative of the convective flux. The Frechet derivative will be defined precisely later, for this concrete example it is the JAcobian of the convective flux, linearised around the a state $\mathbf{u}$

$$(\mathcal{F}^c)'_x[\mathbf{u}] = \begin{bmatrix} 0 & -1/2 \\ -1/2 & 0 \end{bmatrix}, \qquad (\mathcal{F}^c)'_y[\mathbf{u}] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Again, some details on the choice $\tau$ can be found in Appendix A.2.2.

### 4.2.1 Numerical results

Let us consider the system (4.4)-(4.5) on $\Omega = (0,1)^2$ subject to full Dirichlet boundary condition. In the reaction flux $c_1 = c_2 = 1$, the source term and the boundary conditions are set such that the exact solution is given by

$$\mathbf{u} = \begin{bmatrix} \dfrac{\exp(-x/\varepsilon) - \exp(-1/\varepsilon)}{1 - \exp(-1/\varepsilon)} \\ \sin(2\pi x)\sin(2\pi y) \end{bmatrix}.$$

The components of the exact solution can be seen in Figure 4.2.

The target functional is the point value of the solution inside the boundary layer. So we chose $\mathbf{x_0} = (0.01, 0.5)$ and applied the mollified version of the last example from 3.2.1, as in Section 3.6.

$$J(\mathbf{u}) = \int_\Omega \psi_{\mathbf{x_0}} u_1 \,\mathrm{d}\mathbf{x} = 0.367879441171442\,.$$

The corresponding table that contain the results with adaptive refinements are Table 4.1 and Table 4.2 for Type I and Type II estimation, respectively.

The final meshes can be seen in Figure 4.1. From this it can be seen that Type II resolves the whole boundary layer and some of the volleys and hills of the sinus from the second component, however, Type I resolves only some part of this and the error is even smaller.

| NLS | NT | $\lvert J(u)-J(u_h)\rvert$ | $\lvert R_\Omega\rvert$ | $\theta_1$ | $R_{\lvert\Omega\rvert}$ | $\theta_2$ |
|---|---|---|---|---|---|---|
| 1473 | 2816 | $2.955\cdot 10^{-1}$ | $1.316\cdot 10^{0}$ | 4.45 | $2.125\cdot 10^{0}$ | 7.19 |
| 1717 | 3268 | $1.006\cdot 10^{-1}$ | $1.501\cdot 10^{-1}$ | 1.49 | $2.101\cdot 10^{-1}$ | 2.09 |
| 2147 | 4072 | $1.961\cdot 10^{-2}$ | $2.200\cdot 10^{-1}$ | 11.22 | $3.111\cdot 10^{-1}$ | 15.87 |
| 2642 | 4998 | $2.241\cdot 10^{-3}$ | $2.440\cdot 10^{-1}$ | 108.87 | $2.664\cdot 10^{-1}$ | 118.86 |
| 3158 | 5982 | $2.510\cdot 10^{-3}$ | $2.578\cdot 10^{-1}$ | 102.72 | $2.779\cdot 10^{-1}$ | 110.75 |
| 3658 | 6955 | $1.862\cdot 10^{-3}$ | $2.640\cdot 10^{-1}$ | 141.80 | $2.876\cdot 10^{-1}$ | 154.48 |
| 4185 | 7946 | $1.719\cdot 10^{-3}$ | $2.651\cdot 10^{-1}$ | 154.26 | $2.878\cdot 10^{-1}$ | 167.43 |
| 4736 | 9028 | $1.604\cdot 10^{-3}$ | $2.659\cdot 10^{-1}$ | 165.73 | $2.906\cdot 10^{-1}$ | 181.16 |
| 5469 | 10474 | $1.553\cdot 10^{-3}$ | $2.715\cdot 10^{-1}$ | 174.79 | $3.066\cdot 10^{-1}$ | 197.39 |

Table 4.1: Type I (adjoint based) estimation for the system boundary layer point value target functional. Test equation (4.4) - (4.4), $\varepsilon = 0.01$.

| NT | NLS | $\lvert J(u)-J(u_h)\rvert$ | $\mathcal{R}$ | $\theta$ |
|---|---|---|---|---|
| 1473 | 2816 | $2.955\cdot 10^{-1}$ | $2.710\cdot 10^{0}$ | 9.17 |
| 1716 | 3261 | $1.057\cdot 10^{-1}$ | $2.156\cdot 10^{0}$ | 20.40 |
| 2125 | 4007 | $1.894\cdot 10^{-2}$ | $1.997\cdot 10^{0}$ | 105.42 |
| 2755 | 5231 | $2.163\cdot 10^{-2}$ | $1.966\cdot 10^{0}$ | 90.89 |
| 3617 | 6903 | $3.128\cdot 10^{-2}$ | $1.978\cdot 10^{0}$ | 63.24 |
| 4690 | 9004 | $3.248\cdot 10^{-2}$ | $1.982\cdot 10^{0}$ | 61.04 |
| 6170 | 11929 | $1.598\cdot 10^{-2}$ | $1.966\cdot 10^{0}$ | 122.99 |
| 8139 | 15822 | $1.356\cdot 10^{-2}$ | $1.955\cdot 10^{0}$ | 144.26 |
| 10758 | 21009 | $1.949\cdot 10^{-3}$ | $1.955\cdot 10^{0}$ | 1002.89 |

Table 4.2: Type II (residual based) estimation for the system boundary layer point value target functional. Test equation (4.4) - (4.4), $\varepsilon = 0.01$.

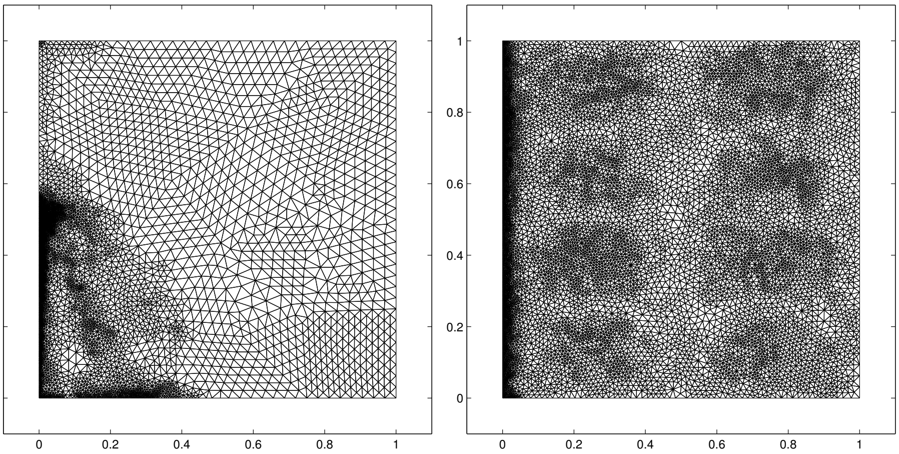

Figure 4.1: Meshes for the system boundary layer point value target functional. Left: adjoint based refined mesh with 5469 triangles (10474 unknowns) $|J(u) - J(u_h)| = 1.553 \cdot 10^{-3}$, right: residual based refined mesh with 10758 triangles (21009 unknowns) $|J(u) - J(u_h)| = 1.949 \cdot 10^{-3}$.

## 4.3 Nonlinear scalar problem

Let us next consider the following problem

$$-\varepsilon\triangle u + \nabla \cdot \left(\frac{u^2}{2}, u\right)^T + cu = f \qquad \text{in } \Omega, \tag{4.8}$$

$$u = g_D \quad \text{on } \Gamma_D, \tag{4.9}$$

$$\varepsilon\nabla u \cdot \mathbf{n} = g_N \quad \text{on } \Gamma_N, \tag{4.10}$$

where we used the notations from Section 2.1. Due to the fact, that in Chapter 2 the FEM discretisation was developed without touching the convective term and in 4.8 the

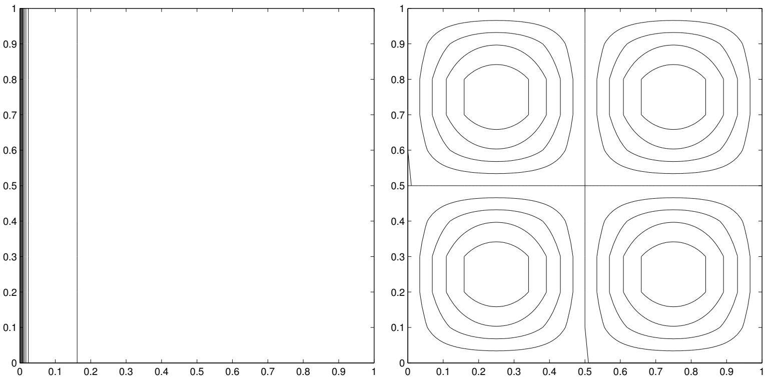

Figure 4.2: Exact solutions of the system case, left $\mathbf{u}_1$, right $\mathbf{u}_2$.

nonlinearity appears only in the convection term, the weak form and the finite element discretisation can be formulated similarly as in Chapter 2.

**Problem Set 4.2.** *Seek $u_h \in V_h$ such that $B(u_h, \tilde{v}_h) = F(\tilde{v}_h)$ holds $\forall \tilde{v}_h \in \tilde{V}_h$.*

However, as it was mentioned in the previous Section it is more convenient to define the form $\mathcal{N}_h(u,v)$ and seek $u_h \in V_h$ such that $\mathcal{N}_h(u_h, \tilde{v}_h) = 0$ holds $\forall \tilde{v}_h \in \tilde{V}_h$. In this case $\mathcal{N}_h(u_h, \tilde{v}_h) = B(u_h, \tilde{v}_h) - F(\tilde{v}_h)$.

Using the flux notation $\mathcal{F}^c(u) = (u^2/2, u)^T$ and the modified FEM functions are

$$\tilde{v} = v + \tau (\mathcal{F}^c)'[u](\nabla v)\,, \tag{4.11}$$

where $(\mathcal{F}^c)'[u] = (u, 1)^T$. Therefore, the function $\tilde{v} = v + \tau\left(u\partial_x v + \partial_y v\right)$ depends on $u$ which will be important later.

The resulting nonlinear problem can be solved by Newton iteration. This means that a sequence of iterative solutions $u_h^k$ is produced by the following scheme. Suppose that we have an initial solution $u_h^0$ and the solutions $u_h^1, \ldots, u_h^k$ are already generated. In that case

$$u_h^{k+1} = u_h^k + \omega_k \triangle u_h^k\,,$$

where $\triangle u_h^k$ is the solution of

$$\mathcal{N}_h'[u_h^k](\triangle u_h^k, \tilde{v}_h) = \mathcal{R}_h(u_h^k, \tilde{v}_h) \quad \forall \tilde{v} \in \tilde{V}\,, \tag{4.12}$$

where $\mathcal{R}_h(\cdot,\cdot)$ is the nonlinear residual $\mathcal{R}_h(u,\tilde{v}) = -\mathcal{N}_h(u,\tilde{v})$.

The Frechet derivative is denoted by $'$ while the symbol $[\cdot]$ denotes around which state the linearisation is performed. Finally, $\omega_k$ denotes the damping factor that can change between steps.

From numerical functional analysis it is well known, that for the Newton iteration we have to compute the Frechet derivative, therefore we suppose that it exists. If it exists, it is the same as the Gateaux derivative, that can be computed more easily. Basically the Gateaux derivative is the directional derivative of $u \to \mathcal{N}_h(u,\tilde{v})$ in the direction $w$ for a fixed $\tilde{v} \in \tilde{V}$[1]. The Frechet derivative is an operator, thereforeit is more convenient to write out the formula when it is applied to a function. In general, the Frechet derivative of $\mathcal{N}_h(\cdot,\cdot)$ applied to $w$ is given by

$$\mathcal{N}_h'[u_h](w_h, \tilde{v}_h) = \lim_{t\to 0} \frac{\mathcal{N}_h(u_h + tw, \tilde{v}_h(u_h + tw)) - \mathcal{N}_h(u_h, \tilde{v}_h)}{t}\,.$$

We have to compute the derivative of (4.3). Let us illustrate the procedure by computing the derivative of the reaction flux applied to a function $w$

$$\begin{aligned}(\mathcal{F}^r)'[u](w) &= \lim_{t\to 0} \frac{\mathcal{F}^r(u_h + tw) - \mathcal{F}^r(u_h)}{t} \\ &= \lim_{t\to 0} \frac{c(u_h + tw) - cu_h}{t} = \lim_{t\to 0} \frac{ctw}{t} = cw\,.\end{aligned}$$

[1] Naturally, as we have seen in (4.11) $\tilde{v}$ depends on $u$, therefore it cannot be fixed. However, we have to emphasize the $\tilde{v}$ was just a convenient notation. If we keep the original decomposition $\tilde{v} = v + \tau(\mathcal{F}^c)'[u](\nabla v)$ then we can say that $v$ is fixed.

However, in the SUPG framework the test functions $\tilde{v}$ also depends on $u$, and this complicates the notations. The Frechet derivative of the whole problem is given by

$$\begin{aligned}\mathcal{N}_h'[u_h](w_h,\tilde{v}_h) = &\int_\Omega \nabla\tilde{v}_h \cdot (\mathcal{F}^v)'[u_h](w_h,\nabla\tilde{w}_h)\ \mathrm{d}\mathbf{x} + \int_\Omega \tilde{v}_h\nabla\cdot(\mathcal{F}^c)'[u_h](\tilde{w}_h)\ \mathrm{d}\mathbf{x}\\ &+\int_\Omega \tilde{v}_h(\mathcal{F}^r)'[u_h](w_h)\ \mathrm{d}\mathbf{x} + \int_\Omega \tilde{v}_h'\nabla\cdot\mathcal{F}^c(w_h)\ \mathrm{d}\mathbf{x}\\ &+\int_\Omega \tilde{v}_h'\mathcal{F}^r(w_h)\ \mathrm{d}\mathbf{x} + \mathcal{N}_{h,\Gamma}'[u_h](w_h,\tilde{v}_h)\,, \end{aligned}\tag{4.13}$$

where $\mathcal{N}_{h,\Gamma}'[u_h](w_h,\tilde{v}_h)$ is the prime of the boundary terms but for the readers' convenience we will not list it here. Again, we use the simplification that $\nabla\tilde{v}\approx\nabla v$.

## 4.4 Error representing formula in the nonlinear case

Let us consider the abstract problem, when we have a (possibly nonlinear) differential operator $N$ in $\Omega$ and a boundary operator $B$ on the boundary (could also be nonlinear) and for generality suppose that we have a system of equations

$$N(\mathbf{u}) = 0 \quad \text{in } \Omega\,, \qquad B(\mathbf{u}) = 0 \quad \text{on } \Gamma\,. \tag{4.14}$$

Suppose that the target function can be represented similarly as in 3.2

$$J(\mathbf{u}) = \int_\Omega j_\Omega(\mathbf{u})\ \mathrm{d}\mathbf{x} + \int_\Gamma j_\Gamma(C\mathbf{u})\ \mathrm{d}s\,,$$

with Frechet derivative

$$J'[\mathbf{u}](w) = \int_\Omega j_\Omega'[\mathbf{u}]\mathbf{w}\ \mathrm{d}\mathbf{x} + \int_\Gamma j_\Gamma'[C\mathbf{u}]C'[\mathbf{u}]\mathbf{w}\ \mathrm{d}s\,.$$

For nonlinear problems the compatibility condition is a bit more complicated than (3.3)

$$(N'[\mathbf{u}]\mathbf{w},\mathbf{z})_\Omega + (B'[\mathbf{u}]\mathbf{w},(C'[\mathbf{u}])^*\mathbf{z})_\Gamma = (\mathbf{w},(N'[\mathbf{u}])^*\mathbf{z})_\Omega + (C'[\mathbf{u}]\mathbf{w},(B'[\mathbf{u}])^*\mathbf{z})_\Gamma\,, \tag{4.15}$$

where $(N'[\mathbf{u}])^*, (B'[\mathbf{u}])^*$ and $(C'[\mathbf{u}])^*$ are the adjoint operators of $N'[\mathbf{u}], B'[\mathbf{u}]$ and $C'[\mathbf{u}]$, respectively. With these, we can formalize the continuous adjoint (dual) problem as

$$(N'[\mathbf{u}])^*\mathbf{z} = j_\Omega'[\mathbf{u}] \quad \text{in } \Omega\,, \qquad (B'[\mathbf{u}])\mathbf{z} = j_\Gamma'[C\mathbf{u}] \quad \text{on } \Gamma\,.$$

Let us denote by $\mathcal{N}_h$ the numerical discretisation of (4.14). At this point, it is not mandatory to think about SUPG discretisation. Therefore, we seek $\mathbf{u}_h \in V_h^q$ such that

$$\mathcal{N}_h(\mathbf{u}_h,\tilde{\mathbf{v}}_h) = 0 \quad \forall\tilde{\mathbf{v}}_h \in \tilde{V}_h^q\,. \tag{4.16}$$

Just as before, it is called consistent if replacing $\mathbf{u}_h$ by $\mathbf{u}$ the equality (4.16) still holds, i.e.

$$\mathcal{N}_h(\mathbf{u}, \tilde{\mathbf{v}}_h) = 0 \quad \forall \tilde{\mathbf{v}}_h \in \tilde{V}_h^q \, . \tag{4.17}$$

Subtracting (4.17) from (4.16) we can conclude the nonlinear version of the Galerkin orthogonality

$$\mathcal{N}_h(\mathbf{u}, \tilde{\mathbf{v}}_h) - \mathcal{N}_h(\mathbf{u}_h, \tilde{\mathbf{v}}_h) = 0 \quad \forall \tilde{\mathbf{v}}_h \in \tilde{V}_h^q \, .$$

Let us denote by $\overline{\mathcal{M}}_h(\mathbf{u}, \mathbf{u}_h; \cdot, \cdot)$ the mean value linearisation of $\mathcal{N}_h(\cdot, \cdot)$

$$\begin{aligned}\overline{\mathcal{M}}_h(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h, \tilde{\mathbf{v}}) &= \mathcal{N}_h(\mathbf{u}, \tilde{\mathbf{v}}) - \mathcal{N}_h(\mathbf{u}_h, \tilde{\mathbf{v}}) \\ &= \int_0^1 \mathcal{N}_h'[s\mathbf{u} + (1-s)\mathbf{u}_h](\mathbf{u} - \mathbf{u}_h, \tilde{\mathbf{v}}) \; \mathrm{d}s \, ,\end{aligned}$$

for all $\tilde{\mathbf{v}} \in \tilde{V}^q$. From numerical functional analysis we know that the derivative does not necessarily exist, however, in the following we assume that is exists and it is well defined. Furthermore, let us introduce the mean value linearisation of the target functional

$$\begin{aligned}\bar{J}(\mathbf{u}, \mathbf{u}_h; \mathbf{u} - \mathbf{u}_h) &= J(\mathbf{u}) - J(\mathbf{u}_h) \\ &= \int_0^1 J'[s\mathbf{u} + (1-s)\mathbf{u}_h](\mathbf{u} - \mathbf{u}_h) \; \mathrm{d}s \, ,\end{aligned}$$

Using the nonlinear compatibility condition (4.15) we can introduce the linearised continuous adjoint problem

$$\overline{\mathcal{M}}_h(\mathbf{w}, \tilde{\mathbf{z}}) = \bar{J}(\mathbf{w}) \quad \forall \mathbf{w} \in V \, , \tag{4.18}$$

where for the simplicity of the notations we omitted the states, around which the mean value linearisations are taken.

Using these notation we can derive the error representing formula for the nonlinear case, similarly as we did for the linear case in Section 3.3.

$$\begin{aligned}J(\mathbf{u}) - J(\mathbf{u}_h) &= \bar{J}(\mathbf{u} - \mathbf{u}_h) && \text{(mean value of } J) \\ &= \overline{\mathcal{M}}_h(\mathbf{u} - \mathbf{u}_h, \tilde{\mathbf{z}}) && \text{(adjoint problem)} \\ &= \overline{\mathcal{M}}_h(\mathbf{u} - \mathbf{u}_h, \tilde{\mathbf{z}} - \tilde{\mathbf{z}}_h) && \text{(Galerkin orthogonality)} \\ &= \mathcal{N}_h(\mathbf{u}, \tilde{\mathbf{z}} - \tilde{\mathbf{z}}_h) - \mathcal{N}_h(\mathbf{u}_h, \tilde{\mathbf{z}} - \tilde{\mathbf{z}}_h) && \text{(mean value of } \mathcal{N}_h) \\ &= -\mathcal{N}_h(\mathbf{u}_h, \tilde{\mathbf{z}} - \tilde{\mathbf{z}}_h) \, . && \text{(primal problem)}\end{aligned}$$

The error representing formula depends on the unknown analytical solution $\tilde{\mathbf{z}}$ of (4.18) and the mean value linarisations are taken at around the unknown analytical solution of the primal problem, namely $\mathbf{u}$. Numerically, the linearisations are taken around $\mathbf{u}_h$ and the adjoint solution $\tilde{\mathbf{z}}$ is replaced by the solution $\tilde{\mathbf{z}}_h$ to the following problem

$$\mathcal{N}_h[\mathbf{u}_h](\mathbf{w}_h, \tilde{\mathbf{z}}_h) = J'[\mathbf{u}_h](\mathbf{w}) \quad \forall \mathbf{w}_h \in V_h^q \,. \tag{4.19}$$

To finalise we have to notice that $\mathcal{R}(\mathbf{u}_h, \tilde{\mathbf{z}} - \tilde{\mathbf{z}}_h) = -\mathcal{N}_h(\mathbf{u}_h, \tilde{\mathbf{z}} - \tilde{\mathbf{z}}_h)$ therefore we get back the weighted residual using which we can continue as in Section 3.3 and we can built up the two kind of error estimations, Type I and Type II.

**Remark 4.3.** *It is important to note, that for Type I we have to solve* (4.19) *twice, which seem to be computationally costly, however,* (4.19) *is a linear problem, therefore the solution of this is computationally not as expensive as the solution of the original problem* (4.16).

**Remark 4.4.** *Comparing* (4.12) *and* (4.19) *we can see that the prime of the nonlinear operator, see* (4.2), *appears twice, but there is a significant difference between the two appearances. In* (4.12) *it is used only for the Newton iteration, and it is well known, that convergence can be achieved even if the derivative is not exact, of course it has to be close to the exact derivative in some sense. This is the so-called quasi-Newton iteration. However, in* (4.19) *the derivative appears as an important tool for the error estimation, therefore if it is just approximated, then the results can be very poor.*

## 4.5 Numerical results

Let us start from the 1D version of (4.8)-(4.10) with neither reaction flux nor source term

$$\begin{aligned} -\varepsilon u'' + \left(\frac{u^2}{2}\right)' &= 0 \quad \text{in } (0,1)\,, \\ u(0) = 0, u(1) &= -1\,. \end{aligned}$$

This problem has a unique solution [39], namely $u(x) = -2\varepsilon\nu_\varepsilon \tanh(\nu_\varepsilon x)$, where $\nu_\varepsilon$ is the solution of the nonlinear algebraic equation

$$2\nu_\varepsilon \tanh(\nu_\varepsilon) = 1\,.$$

Therefore in the 2D case if we set full Dirichlet boundary condition over the boundary and neither reaction flux nor source, and set $g_D$ to be equal to $u(x,y) = -2\varepsilon\nu_\varepsilon \tanh(\nu_\varepsilon x)$ on the boundary, then we get

$$-\varepsilon\triangle u + \nabla \cdot \left(\frac{u^2}{2}, u\right)^T = 0 \qquad \text{in } (0,1)^2, \tag{4.20}$$

$$u = g_D \quad \text{on } \partial(0,1)^2, \tag{4.21}$$

This solution has a boundary layer like behaviour at $x = 0$, similarly to the linear diffusion-advection test case that we had in Section 3.6. As $\varepsilon \to 0$ the solution converge to $u(x,y) = -1$ on $(0,1] \times [0,1]$ with $u(0,y) = 0$ and there is a narrow region with huge derivatives.

Let us set as target quantity the weighted integral of the viscous flux at $x = 0$, with the weight function $\psi$ that is nonzero only for $y \in [0.5,1]$. This function can be seen in Figure 4.3 and its analytical expression is

$$\psi(x,y) = \begin{cases} \exp\left(4 - \frac{1}{256}\frac{1}{((y-5/8)^2-1/32)^2}\right) & \text{if } y \in [0.5, 0.625] \\ 1 & \text{if } y \in [0.625, 0.875] \\ \exp\left(4 - \frac{1}{256}\frac{1}{((y-7/8)^2-1/32)^2}\right) & \text{if } y \in [0.875, 1] \end{cases} \tag{4.22}$$

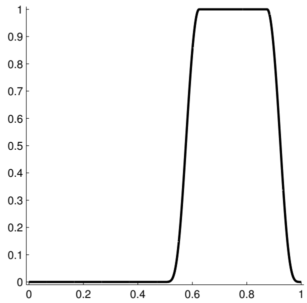

Figure 4.3: Bubble function $\psi$ as the boundary weight function.

Therefore, the target can be formalised es

$$J(u) = \int_{\Gamma_0} \psi \varepsilon \nabla u \cdot \mathbf{n} \, \mathrm{d}s = 0.174596734771569\,,$$

where $\Gamma_0$ stand for the boundary $x = 0$.

The results can be seen in Tables 4.3 and 4.4 and the final meshes in Figure 4.4, while the isolines of the exact primal solution and the adjoint solution can be found in Figure 4.5. Again, we can conclude that the adjoint based approach refines only where it is needed, in the vicinity of the support of $\psi$, however, the residual based resolves the whole boundary layer.

| NT | NLS | $\lvert J(u) - J(u_h)\rvert$ | $\lvert R_\Omega\rvert$ | $\theta_1$ | $R_{\lvert\Omega\rvert}$ | $\theta_2$ |
|---|---|---|---|---|---|---|
| 1473 | 2816 | $6.026 \cdot 10^{-2}$ | $4.096 \cdot 10^{-2}$ | 0.68 | $1.561 \cdot 10^{0}$ | 25.90 |
| 1685 | 3219 | $2.078 \cdot 10^{-2}$ | $6.449 \cdot 10^{-2}$ | 3.10 | $1.791 \cdot 10^{0}$ | 86.22 |
| 2009 | 3840 | $6.034 \cdot 10^{-3}$ | $2.857 \cdot 10^{-2}$ | 4.73 | $2.157 \cdot 10^{0}$ | 357.51 |
| 2507 | 4783 | $2.022 \cdot 10^{-3}$ | $1.778 \cdot 10^{-2}$ | 8.79 | $2.297 \cdot 10^{0}$ | 1135.88 |
| 3097 | 5852 | $1.281 \cdot 10^{-3}$ | $5.418 \cdot 10^{-4}$ | 0.42 | $1.969 \cdot 10^{0}$ | 1537.36 |
| 3578 | 6796 | $1.155 \cdot 10^{-3}$ | $6.149 \cdot 10^{-3}$ | 5.32 | $1.999 \cdot 10^{0}$ | 1730.85 |
| 4094 | 7825 | $1.162 \cdot 10^{-3}$ | $6.332 \cdot 10^{-3}$ | 5.45 | $2.001 \cdot 10^{0}$ | 1721.79 |
| 4468 | 8568 | $1.074 \cdot 10^{-3}$ | $6.088 \cdot 10^{-3}$ | 5.67 | $2.002 \cdot 10^{0}$ | 1863.39 |
| 4686 | 9002 | $1.022 \cdot 10^{-3}$ | $5.815 \cdot 10^{-3}$ | 5.69 | $2.010 \cdot 10^{0}$ | 1966.83 |

Table 4.3: Type I (adjoint based) estimation for boundary viscous flux for the viscous Burgers' equations. Test equation (4.20) - (4.21), $\varepsilon = 0.01$.

| NT | NLS | $\lvert J(u) - J(u_h)\rvert$ | $\mathcal{R}$ | $\theta$ |
|---|---|---|---|---|
| 1473 | 2816 | $6.026 \cdot 10^{-2}$ | $1.132 \cdot 10^{0}$ | 18.78 |
| 1679 | 3191 | $2.046 \cdot 10^{-2}$ | $9.873 \cdot 10^{-1}$ | 48.26 |
| 2099 | 3978 | $8.572 \cdot 10^{-3}$ | $9.727 \cdot 10^{-1}$ | 113.47 |
| 2605 | 4963 | $4.625 \cdot 10^{-3}$ | $9.558 \cdot 10^{-1}$ | 206.67 |
| 3329 | 6401 | $3.948 \cdot 10^{-3}$ | $9.565 \cdot 10^{-1}$ | 242.27 |
| 4209 | 8123 | $2.974 \cdot 10^{-3}$ | $9.538 \cdot 10^{-1}$ | 320.75 |
| 5383 | 10402 | $2.037 \cdot 10^{-3}$ | $9.499 \cdot 10^{-1}$ | 466.40 |
| 6821 | 13247 | $1.601 \cdot 10^{-3}$ | $9.487 \cdot 10^{-1}$ | 592.65 |
| 8077 | 15747 | $1.261 \cdot 10^{-3}$ | $9.485 \cdot 10^{-1}$ | 752.15 |

Table 4.4: Type II (residual based) estimation for boundary viscous flux for the viscous Burgers' equations. Test equation (4.20) - (4.21), $\varepsilon = 0.01$.

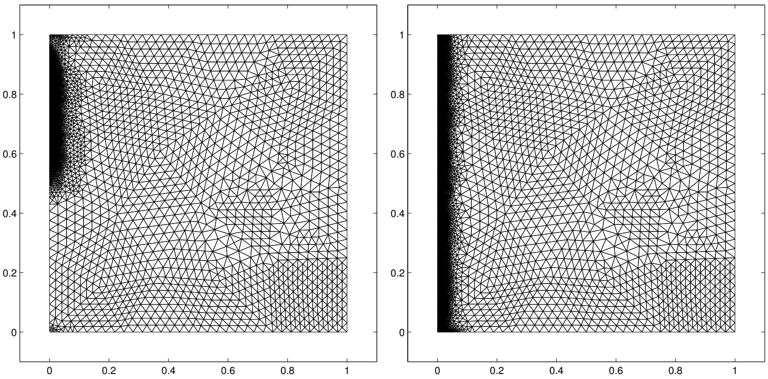

Figure 4.4: Meshes for Burgers' boundary viscous flux. Left: adjoint based refined mesh with 4686 triangles (9002 unknowns) $|J(u) - J(u_h)| = 1.022 \cdot 10^{-3}$, right: residual based refined mesh with 8077 triangles (15747 unknowns) $|J(u) - J(u_h)| = 1.261 \cdot 10^{-3}$.

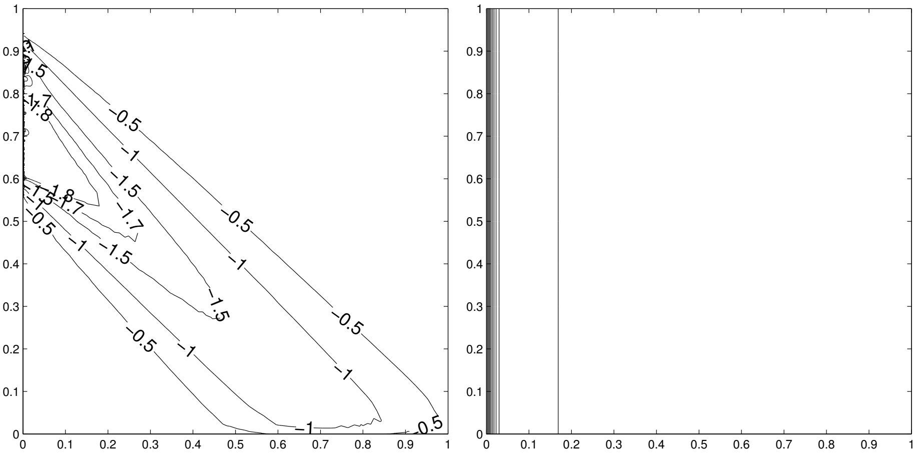

Figure 4.5: Left: adjoint solution, right: exact primal solution both for the Burgers' boundary flux value example.

# Chapter 5

# The compressible Navier-Stokes equations

In this chapter we will formalize the compressible Navier-Stokes equations in 2D. We will focus on the viscous term and refer to [12] (or Appendix A.4) for details on the convective fluxes. The studies of the previous chapters such as nonlinearity and coupled problems will both appear.

From hereafter we will use the Einstein convention, which means that a subscript that appears twice will mean a summation.

## 5.1 The governing equations

The stationary case of the 2D compressible Navier-Stokes problem is given by

$$\nabla \cdot (\mathcal{F}^c(\mathbf{u}) - \mathcal{F}^v(\mathbf{u}, \nabla \mathbf{u})) \equiv \frac{\partial}{\partial_{x_k}} f_k^c(\mathbf{u}) - \frac{\partial}{\partial_{x_k}} f_k^v(\mathbf{u}, \nabla \mathbf{u}) = 0 \quad \text{in } \Omega\,. \tag{5.1}$$

The vector $\mathbf{u}$ denotes the vector of the conservative variables $\mathbf{u} = [\rho, \rho v_1, \rho v_2, \rho E]^T$. The convective (Euler) fluxes are described in Appendix A.4, the viscous fluxes $f_1^v$ and $f_2^v$ are defined by

$$f_1^v(\mathbf{u}, \nabla \mathbf{u}) = \begin{bmatrix} 0 \\ \tau_{11} \\ \tau_{21} \\ \tau_{1j} v_j + \mathcal{K} T_{x_1} \end{bmatrix} \quad \text{and} \quad f_2^v(\mathbf{u}, \nabla \mathbf{u}) = \begin{bmatrix} 0 \\ \tau_{12} \\ \tau_{22} \\ \tau_{2j} v_j + \mathcal{K} T_{x_2} \end{bmatrix},$$

where $\mathcal{K}$ is the thermal conductivity coefficient. The viscous stress tensor is defined by

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

where $\mu$ is the dynamic viscosity, $I$ is the $2 \times 2$ unit matrix, and the temperature is given by $e = c_v T$; thus

$$\mathcal{K}T = \frac{\mu\gamma}{Pr}\left(E - \frac{1}{2}\mathbf{v}^2\right) ,$$

where $\gamma = c_p/c_v$ the ratio of specific heat capacities at constant pressure, $c_p$, and constant volume, $c_v$, $Pr = 0.72$ is the Prandtl number, $\mathbf{v} = (v_1, v_2)$, $\mathbf{v}^2 = v_1^2 + v_2^2$.

For the discretisation let us rewrite (5.1) into the following (equivalent) form

$$\frac{\partial}{\partial_{x_k}}\left(f_k^c(\mathbf{u}) - G_{kl}(\mathbf{u})\frac{\partial \mathbf{u}}{\partial_{x_l}}\right) = 0 \quad \text{in } \Omega\,.$$

Here $G_{kl}(\mathbf{u}) = \partial f_k^v(\mathbf{u}, \nabla\mathbf{u})/\partial u_{x_l}$ for $k = 1, 2$ are the homogeneity tensors defined by $f_k^v(\mathbf{u}, \nabla\mathbf{u}) = G_{kl}(\mathbf{u})\partial\mathbf{u}/\partial x_l$, $k = 1, 2$, where

$$G_{11} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -\frac{4}{3}v_1 & \frac{4}{3} & 0 & 0 \\ -v_2 & 0 & 1 & 0 \\ -\left(\frac{4}{3}v_1^2 + v_2^2 + \frac{\gamma}{Pr}(E - \mathbf{v}^2)\right) & \left(\frac{4}{3} - \frac{\gamma}{Pr}\right)v_1 & \left(1 - \frac{\gamma}{Pr}\right)v_2 & \frac{\gamma}{Pr} \end{pmatrix} ,$$

$$G_{12} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{2}{3}v_2 & 0 & -\frac{2}{3} & 0 \\ -v_1 & 1 & 0 & 0 \\ -\frac{1}{3}v_1 v_2 & v_2 & -\frac{2}{3}v_1 & 0 \end{pmatrix} ,\ G_{21} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -v_2 & 0 & 1 & 0 \\ \frac{2}{3}v_1 & -\frac{2}{3} & 0 & 0 \\ -\frac{1}{3}v_1 v_2 & \frac{2}{3}v_2 & v_1 & 0 \end{pmatrix} ,$$

$$G_{22} = \frac{\mu}{\rho}\begin{pmatrix} 0 & 0 & 0 & 0 \\ -v_1 & 1 & 0 & 0 \\ -\frac{4}{3}v_2 & 0 & \frac{4}{3} & 0 \\ -\left(v_1^2 + \frac{4}{3}v_2^2 + \frac{\gamma}{Pr}(E - \mathbf{v}^2)\right) & \left(1 - \frac{\gamma}{Pr}\right)v_1 & \left(\frac{4}{3} - \frac{\gamma}{Pr}\right)v_2 & \frac{\gamma}{Pr} \end{pmatrix} .$$

## 5.2 Finite element discretisation

Due to the fact, that we do not apply partial integration on the convective flux, we derive the weak form considering only the viscous part. Therefore let us start from

$$-\frac{\partial}{\partial_{x_k}}\left(G_{kl}(\mathbf{u})\frac{\partial \mathbf{u}}{\partial_{x_l}}\right) = 0 \qquad \in \Omega\,,$$

subject to proper boundary conditions that will be listed later.

Similarly as in Section 2.1 we rewrite (5.2) to a first order system

$$\underline{\sigma} = G(\mathbf{u})\nabla\mathbf{u}\,, \qquad -\nabla \cdot \underline{\sigma} = 0 \quad \text{in } \Omega\,, \tag{5.2}$$

i.e., $\sigma_{ik} = (G(\mathbf{u})_{kl})_{ij}\partial_{x_i} u_j$. Multiplying them by test functions $\underline{\phi}$ and $\mathbf{v}$, respectively, integrating over $\Omega$ and using Green's Theorem

$$\int_\Omega \underline{\sigma} : \underline{\phi}\ \mathrm{d}\mathbf{x} = -\int_\Omega \mathbf{u}\nabla \cdot \left(G^T(\mathbf{u})\underline{\phi}\right)\ \mathrm{d}\mathbf{x} + \int_\Gamma \mathbf{u}(G^T(\mathbf{u})\underline{\phi}) \cdot \mathbf{n}\ \mathrm{d}s\,,$$

$$\int_\Omega \underline{\sigma} : \nabla\mathbf{v}\ \mathrm{d}\mathbf{x} = \int_\Gamma \underline{\sigma} \cdot \mathbf{n}\mathbf{v}\ \mathrm{d}s\,,$$

where we used the notation $A : B = \sum_{i,j} A_{ij}B_{ij}$ for the matrix scalar product.[1] To go the discrete level we denote the approximate counterparts of all functions using the subscript $h$.

$$\int_\Omega \underline{\sigma}_h : \underline{\phi}_h \;\mathrm{d}\mathbf{x} = -\int_\Omega \mathbf{u}_h \nabla \cdot \left( G^T(\mathbf{u}_h)\underline{\phi}_h \right) \;\mathrm{d}\mathbf{x} + \int_\Gamma \hat{\mathbf{u}}_h (G^T(\mathbf{u}_h)\underline{\phi}_h) \cdot \mathbf{n} \;\mathrm{d}s\,, \tag{5.3}$$

$$\int_\Omega \underline{\sigma}_h : \nabla \mathbf{v}_h \;\mathrm{d}\mathbf{x} = \int_\Gamma \hat{\underline{\sigma}}_h \cdot \mathbf{n}\mathbf{v}_h \;\mathrm{d}s\,, \tag{5.4}$$

where $\hat{\mathbf{u}}_h$ and $\hat{\underline{\sigma}}_h$ are the numerical approximation of $\mathbf{u}$ and $\nabla\mathbf{u}$, respectively. Let us apply Green's Theorem for (5.3) and set $\underline{\phi}_h = \nabla \mathbf{v}_h$

$$\int_\Omega \underline{\sigma}_h : \nabla \mathbf{v}_h \;\mathrm{d}\mathbf{x} = \int_\Omega G(\mathbf{u}_h)\nabla\mathbf{u}_h : \nabla \mathbf{v}_h \;\mathrm{d}\mathbf{x} + \int_\Gamma (\hat{\mathbf{u}}_h - \mathbf{u}_h)(G^T(\mathbf{u}_h)\nabla\mathbf{v}_h) \cdot \mathbf{n} \;\mathrm{d}s\,, \tag{5.5}$$

Using the fact that the right hand sides of (5.4) and (5.5) are the same and considering that the Dirichlet boundary conditions are imposed weakly according to [36] we can obtain the following problem.

**Problem Set 5.1.** *Seek* $\mathbf{u}_h \in V^d_{h,p}$ *such that*

$$\mathcal{N}_h(\mathbf{u}_h, \mathbf{v}_h) = 0 \qquad \forall \mathbf{v}_h \in V^d_{h,p}\,. \tag{5.6}$$

*where*

$$\begin{aligned} \mathcal{N}_h(\mathbf{u}_h, \mathbf{v}_h) = & \int_\Omega \nabla \cdot \mathcal{F}^c(\mathbf{u}) \cdot \mathbf{v} \;dx + \int_\Omega G(\mathbf{u}_h)\nabla\mathbf{u}_h : \nabla\mathbf{v}_h \;d\mathbf{x} \\ & + \int_\Gamma (\hat{\mathbf{u}}_h - \mathbf{u}_h)(G^T(\mathbf{u}_h)\nabla\mathbf{v}_h) \cdot \boldsymbol{n} \;ds + \alpha \int_\Gamma G(\mathbf{u}_h)(\hat{\mathbf{u}}_h - \mathbf{u}_h) \cdot \boldsymbol{n}\mathbf{v} \;ds \\ & - \int_\Gamma G(\hat{\mathbf{u}}_h)\nabla\mathbf{u}_h \cdot \boldsymbol{n}\mathbf{v}_h \;ds\,. \end{aligned} \tag{5.7}$$

**Remark 5.2.** *As in the previous Chapters we will use streamline diffusion FEM, therefore the FEM basis function are modified in the way*

$$\tilde{\mathbf{v}} = \mathbf{v} + \tau(\mathcal{F}^c)'[\mathbf{u}] \cdot \nabla\mathbf{v}\,,$$

*where* $\mathcal{F}^c$ *denotes the convective (Euler) fluxes, and 5.6 is modified such that*

$$\mathcal{N}_h(\mathbf{u}_h, \tilde{\mathbf{v}}_h) = 0 \qquad \forall \tilde{\mathbf{v}}_h \in \tilde{V}^d_{h,p}\,,$$

*but* $\mathcal{N}_h(\cdot,\cdot)$ *is the same as in* (5.7).

[1] This supposes that the size of $A$ and $B$ are the same, i.e., there exist $n, m \in \mathbb{N}$ such that $A, B \in \mathbb{R}^{n\times m}$. The notation can be extended to the case when $A \in \mathbb{R}^{n\times m}$ and $B \in \mathbb{R}^{m\times n}$ by $A : B = \sum_{i,j} A_{ij}B_{ji}$.

### 5.2.1 Boundary conditions

We have to keep in mind that our future aim is to determine the flow over an airfoil. Therefore, there are farfield and wall boundary conditions. We will determine the possible choices of the boundary flux $\hat{\mathbf{u}}_h = \mathbf{u}_\Gamma(\mathbf{u})$.

- Farfield boundary conditions: we should distinguish between subsonic/supersonic inflow and outflow. For more details we refer to [12].
- Wall boundary conditions: for the velocity we have no slip condition ($v_1 = v_2 = 0$) and for the temperature we can have isothermal wall ($T = T_w$) or we can have adiabatic wall ($\mathbf{n} \cdot \nabla T = 0$). The corresponding boundary conditions are
  - for the isothermal wall

$$\mathbf{u}_\Gamma(\mathbf{u}) = (u_1, 0, 0, u_4)^T \,,$$

  - for the adiabatic wall

$$\mathbf{u}_\Gamma(\mathbf{u}) = (u_1, 0, 0, u_1 c_v T_w)^T \,.$$

## 5.3 Target functionals

If we consider to solve the flow equations over an airfoil then we can think about the pressure at the leading edge or the lift/drag coefficients as possible target quantities.

### Pressure at the leading edge

The corresponding target functional is

$$J(\mathbf{u}) = \int_\Omega p(\mathbf{u}) \psi_{\mathbf{x}_0} \; \mathrm{d}\mathbf{x} \,,$$

where $\psi$ is the mollified function that was introduced in Section 3.2.1, $\mathbf{x}_0$ denotes the stagnation point. The pressure $p$ can be computed using the total enthalpy $H$, which is given by

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

We have to emphasize that this target functionals is nonlinear.

### Lift/drag coefficient

Lift and drag can be decomposed as pressure induced and viscous lift/drag. The corresponding target functionals are

$$J(\mathbf{u}) = \frac{1}{C_\infty} \int_{\Gamma_W} (p(\mathbf{u})\mathbf{n} - \tau\mathbf{n}) \, \psi \; \mathrm{d}s \,,$$

where $\Gamma_W$ denotes the wall (airfiol), $C_\infty$ and $\psi$ is defined by

$$\begin{aligned} C_\infty &= \frac{1}{2}\gamma p_\infty M_\infty^2 l \,, \\ \psi_d &= (\cos(\alpha), \sin(\alpha))^T \,, \\ \psi_l &= (-\sin(\alpha), \cos(\alpha))^T \,, \end{aligned}$$

where subscripts $d$, $l$ and $\infty$ stands for drag, lift and free-stream, respectively, $M$ denotes the Mach number, $l$ denotes the reference length.

We have to emphasize that these target functionals are also nonlinear, due to the appearance on the pressure.

## 5.4 Error estimation

As we have seen earlier, see Remark 4.4, the prime of the nonlinear operator appears twice in the computation, once it is used for the solution of the primal problem (in the Newton iteration), once it is used for the dual problem.

Let us list this, using the notation $\hat{\mathbf{u}}_h = \mathbf{u}_\Gamma(\mathbf{u})$ to clarify that boundary data could also depend on $\mathbf{u}$, therefore its derivative is nonzero. After some algebra the final form is

$$\begin{aligned} \mathcal{N}_h'[\mathbf{u}_h](\mathbf{w}_h, \tilde{\mathbf{v}}_h) = &\int_\Omega \nabla \cdot (\mathcal{F}^c)'[\mathbf{u}]\mathbf{w}\tilde{\mathbf{v}} \;\mathrm{d}\mathbf{x} + \int_\Omega \nabla \cdot \mathcal{F}^c \tilde{\mathbf{v}}'\mathbf{w} \;\mathrm{d}\mathbf{x} \\ &+ \int_\Omega G'[\mathbf{u}_h]\mathbf{w}\nabla\mathbf{u}_h : \nabla\tilde{\mathbf{v}}_h \;\mathrm{d}\mathbf{x} + \int_\Omega G[\mathbf{u}_h]\nabla\mathbf{w} : \nabla\tilde{\mathbf{v}}_h \;\mathrm{d}\mathbf{x} \\ &+ \int_\Gamma \mathbf{u}_\Gamma'[\mathbf{u}_h]\mathbf{w}_h(G^T(\mathbf{u}_h)\nabla\tilde{\mathbf{v}}_h) \cdot \mathbf{n} \;\mathrm{d}s + \int_\Gamma (\mathbf{u}_\Gamma(\mathbf{u}_h) - \mathbf{u}_h)(G^T)'[\mathbf{u}_h]\mathbf{w}_h\nabla\tilde{\mathbf{v}}_h \cdot \mathbf{n} \;\mathrm{d}s \\ &+ \alpha \int_\Gamma G'[\mathbf{u}_h]\mathbf{w}(\mathbf{u}_\Gamma(\mathbf{u}_h) - \mathbf{u}_h) \cdot \mathbf{n}\mathbf{v} \;\mathrm{d}s + \alpha \int_\Gamma G(\mathbf{u}_h)\mathbf{u}_\Gamma'[\mathbf{u}_h]\mathbf{w} \cdot \mathbf{n}\tilde{\mathbf{v}} \;\mathrm{d}s \\ &- \int_\Gamma G'[\mathbf{u}_\Gamma(\mathbf{u})](\mathbf{u}_\Gamma'[\mathbf{u}])w\nabla\mathbf{u}_h \cdot \mathbf{n}\tilde{\mathbf{v}}_h \;\mathrm{d}s \,. \end{aligned} \tag{5.8}$$

In addition, the prime of the target functional is also needed, but that is more easy to compute, and with that the dual problem, as we have seen earlier, reads as follows: find $\tilde{\mathbf{z}}_h \in \tilde{V}_h^q$ such that

$$\mathcal{N}_h[\mathbf{u}_h](\mathbf{w}_h, \tilde{\mathbf{z}}_h) = J'[\mathbf{u}_h](\mathbf{w}) \quad \forall \mathbf{w}_h \in V_h^q \,.$$

# Chapter 6

# Recommendations for future work

## 6.1 Conclusion

The main goal of this project was to extend the work of Stefano D'Angleo on adjoint based goal oriented error estimation from convection problems to convection-diffusion ones, using streamline upwind stabilised finite element methods. Analytically such a problem is different from the inviscid problems in a way that boundary layer-type solutions could appear. The main achievements are the followings.

### Implementations in APOGEE

The code APOGEE was written by Stefano D'Angelo and it was originally dedicated for problems that includes only convective fluxes. For testing purposes the reaction and the source terms, and as the main task of the project, the viscous flux were implemented and tested for linear and nonlinear scalar problems and for linear coupled problems.

### Error estimation for convection-diffusion problems

The error representing formula for the linear and the nonlinear cases have been presented. Numerical results have been shown for the linear and nonlinear boundary layer problems, and for linear coupled systems. It has been shown, that the adjoint based error estimation can determine the target quantity in the presence of boundary layers using less unknowns than the residual based estimators.

### Adjoint consistency

The convergence of the target quantity can be guaranteed due to the convergence rate of the primal problem. However, higher rates of convergence can be achieved using the convergence rate of the adjoint (dual) problem. According to the theory of convergence of FEM, the question of this convergence rate can be reduced to the question of the convergence rate of the interpolation using the streamline upwind basis functions. It was shown, that using the streamline upwind functions better convergence rates can be achieved for the adjoint problems than for the residual distribution or bubble stabilised functions.

## 6.2 Future work

### 6.2.1 Possible modification in the stabilisation of the viscous term

As we have seen in Section 2.3 the stabilisation of the second order term is complicated, but the advantage of this is the extra rates of convergence. This stabilisation is important only if higher order discretisation is used (polynomials of degree $\geq 2$). In this project we used only first order discretisation for the primal and for one of the adjoint solution, and used second order only for the reference adjoint solution, therefore we neglected the stabilisation of the second order term.

However, there are some ways that can be implemented and tested.

#### Partial integration

One of them starts with the partial integration (2.3). Let us recall this in a reordered form

$$\int_K -\varepsilon \triangle u(\tau \mathbf{b} \cdot (\nabla v)) \neq \int_K \varepsilon \nabla u \nabla (\tau \mathbf{b} \cdot (\nabla v)) \,.$$

The equality can be achieved if we include the boundary term

$$\int_K -\varepsilon \triangle u(\tau \mathbf{b} \cdot (\nabla v)) = \int_K \varepsilon \nabla u \nabla (\tau \mathbf{b} \cdot (\nabla v)) - \int_{\partial K} \varepsilon \nabla u \cdot \mathbf{n}(\tau \mathbf{b} \cdot (\nabla v)) \,.$$

Therefore, if we could implement $\nabla(\tau \mathbf{b} \cdot (\nabla v))$ then we could get rid of the approximation $\nabla \tilde{v} \approx \nabla v$ that was used for all cases. After doing some algebra with the new terms they can be rearranged as jumps over the interior edges.

Naturally, it means that the corresponding bilinear form has to be changed. On the discrete level the jumps have to be built into the bilinear form. It could be shown that on the continuous level these jumps are disappearing, therefore the bilinear form can be modified even on the continuous level.

#### Projection

As it was suggested by Mario Richiuto we can use some projection when adding the stabilising term. To do this we have to work out the formula $B(u,v) + ST(u,v)$ from Section 2.3. Let us neglect the boundary conditions for the sake of simplicity and consider the case without the reaction term

$$\varepsilon \int_\Omega \nabla u \cdot \nabla v + \int \nabla \cdot (\mathbf{b}u)v + \sum_K \int_K (-\varepsilon \triangle u + \nabla \cdot (\mathbf{b}u))(\tau \mathbf{b} \cdot (\nabla v)) \,. \tag{6.1}$$

Using that $\triangle u = \nabla \cdot (\nabla u)$ we can rewrite (6.1) as

$$\varepsilon \int_{\Omega} \nabla u \cdot \nabla v \, \mathrm{d}\mathbf{x} + \int \nabla \cdot (\mathbf{b}u) v \, \mathrm{d}\mathbf{x} + \sum_{K} \int_{K} (-\varepsilon \nabla \cdot w + \nabla \cdot (\mathbf{b}u))(\tau \mathbf{b} \cdot (\nabla v)) \, \mathrm{d}\mathbf{x} \, , \quad (6.2)$$

where $w$ is the projection of $\nabla u$ in the sense that

$$\int_{\Omega} vw \, \mathrm{d}\mathbf{x} = \int_{\Omega} v \nabla u \, \mathrm{d}\mathbf{x} \, , \quad (6.3)$$

for all test functions $v$. With this we can completely get rid of the term $\tau \mathbf{b} \cdot (\nabla u)$.

**Rewriting to a system**

Using (6.2) and (6.3) we can rewrite the original equation (2.1) to a first order system such as

$$-\varepsilon \nabla \cdot w + \nabla \cdot (\mathbf{b}u) + cu = f \, , \quad (6.4)$$

$$-\nabla u + w = 0 \, . \quad (6.5)$$

If we want to solve (6.4)-(6.5) using SUPG we do not have to deal with $\nabla \tilde{v}$ due to the fact that there is no second order term. However, some boundary conditions have to be created for (6.5).

### 6.2.2 Meshes

Throughout this work (and also in the work of Stefano D'Angelo) only triangular meshes have been used with straight edges. This should be generalized by curvilinear elements, especially in the case of computing the flow over an airfoil.

Also quadrilateral meshes could be implemented, which are much more popular in boundary layer computations. However, in this case some other complications will arise, even if the quadrilaterals have straight edges. It is well known, that any two triangles and parallelograms can be transformed into each other by an affine linear mapping, as it was also mentioned in Section 2.2. Therefore, plenty on the computations can be done on the reference triangle/parallelogram, such as computation of the basis function at the quadrature points.

In the case of arbitrary quadrilaterals this property does not hold anymore, and the corresponding transformation becomes nonlinear, which complicates the calculation.

Even mixed meshes could be used, in which there are quadrilateral elements along the airfoil inside the boundary layer, and outside of that triangles are used. With such a mesh some computational time can be saved in comparison to the full quadrilateral meshes, although, the code becomes complicated as it has to deal with two different types of elements at the same time.

### 6.2.3 Extension of the modelled phenomenons

Naturally, the first extension will be to solve flow problems around an airfoil with the target functional listed in Section 5.3. In the case of shocks some shock capturing schemes have already been implemented in APOGEE by Stefano D'Angelo, therefore we will have a wide range of possible flow conditions.

Also, it could be useful to include some turbulence modelling, as it was done in the discontinuous Galerkin case in [22] where RANS-$k - \omega$ was used to model turbulence. With this some more realistic simulations could be done.

Finally, there are other fields of CFD where the adjoint based approach is used: the optimisation and the uncertainty quantification. We will try to connect these three approaches in a way that the same matrix should appear in all three problems with different right hand sides. For example, when coupling the adjoint based error estimation with the adjoint based optimisation, the aim could be to give new geometrical shapes while guaranteeing that the flow equations (whether they are the Euler or the Navier Stokes equations) are solved properly, i.e. the target with respect to what the optimisation is done, is computed accurately.

# Appendix A

# Mathematical supplement

## A.1 Banach and Hilbert spaces

Let $V$ be a real vector space.

**Definition A.1.** *The mapping $\|\cdot\| : V \to \mathbb{R}_{\geq 0}$ is called a norm, if it satisfies the following three conditions:*

1. $\|v\| = 0 \Leftrightarrow v = 0$,
2. $\|\lambda v\| = |\lambda|\|v\|$, $\forall v \in V$, $\forall \lambda \in \mathbb{R}$,
3. $\|v + w\| \leq \|v\| + \|w\|$, $\forall v, w \in V$ *(triangle inequality).*

**Definition A.2.** *The mapping $\|\cdot\|_* : V \to \mathbb{R}_{\geq 0}$ is called seminorm, if it satisfies 2. and 3. from the previous definition and*

*1.'* $v = 0 \Rightarrow \|v\| = 0$.

**Definition A.3** (Equivalent norms)**.** *Let us have two norms on $V$ $\|\cdot\|_1$, and $\|\cdot\|_2$. We say that these two norms are equivalent if there exist $M > m > 0$ constants such that $\forall v \in V$:*

$$m\|\cdot\|_1 \leq \|\cdot\|_2 \leq M\|\cdot\|_1.$$

**Lemma A.4.** *If $V$ is finite dimensional then any two norms are equivalent.*

**Definition A.5.** *The bilinear mapping $\langle\cdot,\cdot\rangle : V \times V \to \mathbb{R}$ is called an inner product (or scalar product), if it satisfies the following three conditions:*

1. $\langle v, w\rangle = \langle w, v\rangle$, $\forall v, w \in V$ *(symmetry),*
2. $\langle v, v\rangle \geq 0$, $\forall v \in V$ *(positivity),*
3. $\langle v, v\rangle = 0 \Leftrightarrow v = 0$.

**Remark A.6.** *Let $\langle \cdot, \cdot \rangle$ be an inner product, then $\|v\|_V := \langle v, v\rangle^{1/2}$ $\forall v \in V$ defines a norm on $V$.*

**Lemma A.7** (Cauchy-Schwartz inequality)**.** *$\forall v, w \in V$: $|\langle v, w\rangle| \leq \|v\|_V \|w\|_V$.*

**Definition A.8.** *A Hilbert space is an inner product space that is complete with the norm defined by the inner product.*

In the following $\alpha = (\alpha_1, \dots, \alpha_d)$ (where $\alpha_i$ is a non-negative integer $\forall i = 1, \dots, d$) will denote a multi-index, and for a function with $d$ variable $\partial^\alpha v := \partial_1^{\alpha_1} \dots \partial_d^{\alpha_d} v$. The absolute value of $\alpha$ is defined as $|\alpha| := \alpha_1 + \dots + \alpha_d$.

Throughout this report we have used the following function spaces ($\Omega \subset \mathbb{R}^d$ in every case).

- $L^p(\Omega) := \{v : \Omega \to \mathbb{R} : \int_\Omega |v|^p < \infty\}$, $1 \leq p \leq \infty$.
- $L^\infty(\Omega) := \{v : \Omega \to \mathbb{R} : \inf\{\sup_{N \subset \Omega, \mathrm{meas}(N)=0} |v|\} < \infty\}$, $1 \leq p \leq \infty$.
- $W^{m,p}(\Omega) := \{v : \Omega \to \mathbb{R} : (\partial^\alpha v) \in L^p(\Omega), \forall \alpha : |\alpha| \leq m\}$
- $H^m(\Omega) := W^{m,2}(\Omega)$

With the following notations for norms and seminorms:

- $\|u\|_{0,T}^2 := \|u\|_{L^2(T)}^2 = \int_T |u|^2$,
- $\|u\|_{m,T}^2 := \|u\|_{H^m(T)}^2 = \sum_{|\alpha| \leq m} \int_T |\partial^\alpha u|^2$,
- $|u|_{k,T}^2 := \sum_{|\alpha| = k} \int_T |\partial^\alpha u|^2$,

where $\alpha$ is a multi-index and $T$ is an arbitrary domain of integration. We can omit the second subscript if the integration domain is $\Omega$ (i.e. $\|u\|_0^2 := \|u\|_{0,\Omega}^2$).

The fractional Hilbert space $H^{1/2}(\partial\Omega)$ was used when we worked with Dirichlet boundary condition. This space can be characterized as follows.

**Definition A.9.**

$$H^{1/2}(\partial\Omega) := \left\{ u \in L^2(\partial\Omega); \frac{u(x) - u(y)}{|x - y|^{\frac{1+d}{2}}} \in L^2(\partial\Omega \times \partial\Omega) \right\}.$$

**Remark A.10.** *$H^{1/2}(\partial\Omega)$ can be defined using trace operators: $H^{1/2}(\partial\Omega) := \{u \in L^2(\partial\Omega) : u = U|_{\partial\Omega}$ (in trace sence) for some $U \in H^1(\Omega)\}$. For more details see [35, p.58].*

## A.2 Proof of convergence of FEM

Suppose that we have a continuous problem, which means that we seek for $u \in V_*$ such that $\forall v \in \tilde{V}$

$$B(u, v) = F(v)\,, \tag{A.1}$$

where $B(\cdot,\cdot)$ is a bilinear form defined over $V_* \times \tilde{V}$ and $F(\cdot)$ is a linear form defined over $\tilde{V}$. To clarify why we have the notation $V_*$ see Remark A.15.

Let us denote by $V_h$ a finite dimensional subspace of $V_*$. With this the discrete counterpart of (A.1) reads as follows: we seek for $u_h \in V_h$ such that $\forall v \in \tilde{V}$

$$B(u_h, v) = F(v)\,, \tag{A.2}$$

where $B(\cdot,\cdot)$ and $F(\cdot)$ are the same as above.

Let us modify the Definition 2.5 a little bit, to fit it to our framework.

**Definition A.11.** *Suppose that the bilinear form $B(\cdot,\cdot)$ is defined over $V_* \times \tilde{V}$. Let us denote by $|\!|\!|\cdot|\!|\!|_*$ a norm on $V_*$ and by $|\!|\!|\cdot|\!|\!|_t$ a norm on $\tilde{V}$.*

- *The bilinear form is* **continuous** *on $V_* \times \tilde{V}$, if there exists $C_c > 0$ such that $B(u, v) \le C_c\, |\!|\!|u|\!|\!|_*\, |\!|\!|v|\!|\!|_t$, $\forall u \in V\, v \in \tilde{V}$.*
- *The bilinear form is* **coercive** *on $\tilde{V} \times \tilde{V}$, if there exists $C_s > 0$ such that $B(v, v) \ge C_s\, |\!|\!|\tilde{v}|\!|\!|_t^2$, $\forall v \in \tilde{V}$ .*

The main goal of this subsection is to estimate the discretisation error, i.e. the distance between the exact solution $u$ and the discrete one $u_h$.

**Lemma A.12.** *We have that*

$$B(u - u_h, v) = 0, \quad \forall v \in \tilde{V}\,,$$

*which means that the discretisation error is orthogonal to the finite element space.*

*Proof.* According to (A.1) we have: $B(u, v) = F(v)$ $\forall v \in \tilde{V}$. On the other hand A.2 means $B(u_h, v) = F(v)$ $\forall v \in \tilde{V}$. Subtracting the two equations we get the desired statement. □

**Remark A.13.** *The above property is called Galerkin orthogonality. If equation* (A.12) *holds the finite element method is called consistent.*

We will develop the basics of (almost) all a-priori error estimations using the following three conditions:

- $B(\cdot,\cdot)$ is coercive,
- $B(\cdot,\cdot)$ is bounded,

- $B(\cdot,\cdot)$ possesses the Galerkin orthogonality.

We will show, that the magnitude of the discretisation error $\left\| u - u_h \right\|$ is the same as the interpolation error $\left\| u - u_I \right\|$ where $u_I$ is the interpolant of $u$. To this, we will bound the difference between the numerical solution and the interpolant

$$\left\| u_I - u_h \right\|_* \leq C \left\| u - u_I \right\|_* .$$

With this we can reduce the question to the accuracy of the interpolation thanks to the triangle inequality

$$\left\| u - u_h \right\|_* \leq \left\| u - u_I \right\|_* + \left\| u_h - u_I \right\|_* \leq (1 + C) \left\| u_h - u_I \right\|_* .$$

Consequently, the magnitude of the discretisation error is the same as the magnitude of the interpolation error, i.e. if we can approximate an arbitrary function from the function space $V_*$ with a certain convergence rate, then the convergence rate of the corresponding FEM will be the same as the convergence rate of the approximation.

**Lemma A.14.** *Suppose that the bilinear form is coercive, bounded and the FEM is consistent. In this case*

$$\left\| u_I - u_h \right\|_* \leq C \left\| u - u_I \right\|_* .$$

*Proof.* Using the three assumptions we have

$$\begin{aligned}
C_c \left\| u_I - u_h \right\|_t^2 &\leq B(u_I - u_h, u_I - u_h) && \text{(coercivity)} \\
&= B(u_I - u_h, u_I - u_h) - B(u - u_h, u_I - u_h) && \text{(Galerkin orthogonality)} \\
&= B(u_I - u, u_I - u_h) \leq C_b \left\| u_I - u \right\|_* \left\| u_I - u_h \right\|_t . && \text{(boundedness)}
\end{aligned}$$

If $\left\| u_I - u_h \right\|_t = 0$ then the interpolant and the discrete solution are the same and the proof is complete, therefore the discretisation error is equal to the approximation error. Otherwise we can divide by $C_c \left\| u_I - u_h \right\|_t$ and we get

$$\left\| u_I - u_h \right\|_t \leq \frac{C_b}{C_c} \left\| u_I - u \right\|_* .$$

To complete the proof we have to assume that the norms $\left\| \cdot \right\|_*$ and $\left\| \cdot \right\|_t$ are equivalent on $\tilde{V}$, see Definition A.3, which implies

$$m \left\| u_I - u_h \right\|_* \leq \left\| u_I - u_h \right\|_t .$$

The easiest case is when the two norms are the same. □

For some comments on the approximation results see Appendix A.2.1.

**Remark A.15.** *We used to notation $V_*$ at the beginning of this section which could seem to be strange. For concrete examples the analytical solution usually belong to some Sobolev space $W$. However, we can construct a FEM in which the discrete solution belongs to $V_h \not\subset W$, in this case the method is called non-conforming. To be able to prove the convergence, we have to introduce the space $V_* = W + V_h$, where $+$ stands for the usual Minkowski sum. With this notation we have that $W$ and $V_h$ are both a subset of $V_*$.*

**Remark A.16.** *It can be seen from the proof, that $C_c$ (the constant from the coercivity) appears in the denominator, therefore if it is small then the constant in the convergence is significant and numerically a very dense mesh is required to achieve the proper convergence rate. The case when $C_c$ is small is called coercivity loss, see Section 3.5 of [16]. This can arise in many physical phenomenons, one of them is the convection dominated convection-diffusion problem, due to the fact that $C_c$ is proportional to $\varepsilon/|\boldsymbol{b}\|$, therefore it is small when $\varepsilon \ll |\boldsymbol{b}|$.*

*Numerically that is why the cell Peclet number plays an important role. It weights the $C_c$ with the mesh size, therefore gives an indicator whether the mesh is good enough or not. That is why we had the two asymptotic requirements* (2.15) *and* (2.16) *on $\tau_h$.*

To check the corresponding properties of the bilinear form it can be said that continuity is usually an easy task, on the other hand, coercivity is always challenging. For example, let us recall the corresponding materials for the linear diffusion-advection-reaction problem, so let us have the following partial differential equation

$$-\varepsilon \triangle u + \nabla \cdot (\mathbf{b} u) + cu = f \quad \text{in } \Omega \,,$$

subject to proper boundary conditions. For details on the different coefficients, see Chapter 2.

The corresponding norms are

$$|\!|\!| u |\!|\!|_*^2 = |\!|\!| u |\!|\!|_t^2 = \varepsilon |v|_1^2 + \sum_{T \in \mathcal{T}_h} \left( \tau_T \|\mathbf{b} \cdot \nabla u\|_{0,T}^2 + \omega \|v\|_{0,T}^2 \right) . \tag{A.3}$$

Set $c_T = \max_{x \in T} |c(x)|$ for each $T \in \mathcal{T}_h$ and let the constant $\omega$ satisfy

$$c - \frac{1}{2} \nabla \cdot \mathbf{b} \geq \omega > 0 \,.$$

The local inverse inequality ensures that there is a constant $\nu_{inv}$ that is independent on the mesh, with wich we have the following inequality

$$\|\triangle v_h\|_{0,T} \leq \nu_{inv} h^{-1} |v_h|_{1,T} \,,$$

for every $v_h \in V_h$.

Using these notations we can prove the following Lemma. For the proof we refer to Lemma 4.16 [39].

**Lemma A.17.** *Let the parameter $\tau_h$ satisfy the inequality*

$$0 < \tau_h \leq \frac{1}{2} \min \left\{ \frac{\omega}{c_T^2}, \frac{h_T^2}{\varepsilon \nu_{inv}} \right\} ,$$

*for each $T \in \mathcal{T}_h$. Then the SUPG bilinear form defined in Section 2.3 is coercive, i.e.,*

$$B(v,v) \geq \frac{1}{2} \left\| v \right\|_t^2 \qquad \forall v \in \tilde{V} .$$

If instead of a bilinear form we have to deal with a semilinear form, then some steps of the proof have to be changed. We loose the equality

$$B(u_I - u_h, u_I - u_h) - B(u - u_h, u_I - u_h) = B(u_I - u, u_I - u_h) ,$$

however, we have to emphasize that the final aim was to bound the term with

$$C \left\| u_I - u \right\|_* \left\| u_I - u_h \right\|_t .$$

This can be achieved without the linearity if the semilinear form is Lipschitz continuous in its first argument, i.e. there exists a constant $C_L > 0$ such that

$$|B(u_1, v) - B(u_2, v)| \leq C_L \left\| u_1 - u_2 \right\|_* \left\| v \right\|_t .$$

If this holds, then we have

$$|B(u_I - u_h, u_I - u_h) - B(u - u_h, u_I - u_h)| \leq C_L \left\| u_I - u_h - (u - u_h \right\|_* \left\| u_I - u_h \right\|_t ,$$

that completes the equality of the above proof. However, Lipschitz continuity can mostly be proved for problems where nonliearity appears in the diffusion term, such as for non-Newtonian flows, see i.e. [26].

### A.2.1 Polynomial approximation in Hilbert spaces

**Definition A.18.** *The mesh $\mathcal{T}_h = \{E_i, i = 1, \ldots, N_{el}\}$ ($E_i$ is a triangle for all $i = 1, \ldots, N_{el}$) is called shape regular if there exists a constant $c_0$ such that*

$$h_i \leq c_0 \rho_i,$$

*holds $\forall i = 1, 2, \ldots, N$, where $h_i$ is the diameter and $\rho_i$ is the radius of the inner circle of $E_i$.*

In the following we always suppose that the mesh is shape regular.

Throughout the previous section we have seen that one of the key ingredients in the convergence proof is the approximation of a given function using polynomials of degree $p$.

Let us denote by $u_I$ the interpolant of $u \in H^{l+1}(\Omega)$ $(1 \leq l \leq p)$ that can be calculated using Lagrange elements of degree $p$ for the interpolation. Then for all $u \in H^{l+1}(\Omega)$ $(1 \leq l \leq p)$ we have

$$\|u - u_I\|_0 + h|u - u_I|_1 \leq ch^{l+1}|u|_{l+1}.$$

For the proof see i.e. [16, Sect. 1.5.1].

If we are using the SUPG norm defined by (A.3), after some calculations, see [39], we have

$$|||u - u_I|||_* \leq c(h^{1/2} + \varepsilon^{1/2})h^p|u|_{p+1}\,,$$

and for the convection dominated case

$$\|u - u_I\| \leq ch^{p+1/2}|u|_{p+1}\,,$$

### A.2.2 Additional comments on the choice of $\tau$

In Lemma A.17 is was stated the $\tau$ should be bounded from above to guarantee consistency and therefore convergence. In Section 2.3.1 it was mentioned that the parameter $\tau$ has to depend on the local Peclet number through the function $\zeta$ for which we had two asymptotic requirements

$$\begin{aligned} \zeta(Pe^h) \to 1 \quad &\text{as } Pe^h \to \infty\,, \\ \zeta \approx Pe^h \quad &\text{as } Pe^h \to 0\,. \end{aligned}$$

There are plenty of possibilities for $\zeta$, in Table A.1 we list the most popular ones and some of them are plotted in Figure A.1.

| $\zeta(Pe^h)$ | name and corresponding article |
|---|---|
| $\coth(Pe^h) - 1/Pe^h$ | optimal [11] |
| $\min\{1, Pe^h/3\}$ | doubly asymptotic [28] |
| $Pe^h/(1 + Pe^h)$ | Mizukami [3] |
| $\max\{0, 1 - 1/Pe^h\}$ | critical [11] |
| $\max\{0, 1 - 1/(2Pe^h)\}$ | Johnson [31] |
| $(Pe^h/3)\left[1 + (Pe^h/3)^2\right]^{-1/2}$ | Hughes [27], Tezduyar [45] |
| $Pe^h(1 + (Pe^h)^2)^{-1/2}$ | Tezduyar [32] |

Table A.1: Possible choices of the function $\zeta$.

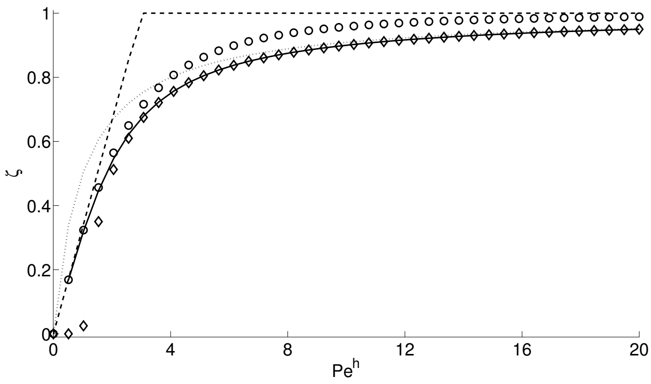

Figure A.1: Possible choices of the function $\zeta$. Solid line: optimal, dashed line: doubly asymptotic, dotted line: critical, $\diamondsuit$: Mizukami, $\circ$ Hughes-Tezduyar

## A.3 Weakly imposing the Dirichlet boundary conditions for second order problems

We have seen in Section 2.1 how to get from the PDE to the weak form. From the calculation it is clear that the natural type of boundary conditions is the Neumann one, it can be implemented into the weak form quite easily. However, Dirichlet boundary conditions can cause some difficulties.

They do not appear naturally in the weak form. In most papers and textbooks they are implemented strongly which means that the bilinear form is restricted to $H^1_{\Gamma_\mathrm{D}} \times H^1_0$, where

$$\begin{aligned} H^1_{\Gamma_\mathrm{D}}(\Omega) &= \{u \in H^1(\Omega) : u|_{\Gamma_D} = g\}\,, \\ H^1_0(\Omega) &= \{u \in H^1(\Omega) : u|_{\Gamma_D} = 0\}\,, \end{aligned}$$

therefore the solution is approximated on the subspace of $H^1(\Omega)$ that contains function that are satisfying the Dirichlet condition. This is exactly why we had the condition that $g_D \in H^{1/2}(\Gamma)$, otherwise it would not be guaranteed that the boundary condition could be satisfied.

On the other hand, this would complicate the target based error estimation, because to achieve adjoint consistency we should look for the solution in two different subspace of $H^1(\Omega)$ for the primal and dual problems.

For first order PDEs the weakly imposing is almost natural. All we have to do is to apply Green's Theorem twice and distinguish between the inflow and outflow boundary. For more details see i.e. [12].

The first idea of weakly imposing the Dirichlet boundary conditions for second order problems can be related to Nitsche [36]. He suggested to modify the Dirichlet boundary conditions artificially. Let us recall the formulas and the notations from Section 2.1

$$u = g_D \quad \text{on } \Gamma_D \Longrightarrow u + \alpha^{-1}\varepsilon\nabla u \cdot \mathbf{n} = g_D \quad \text{on } \Gamma_D\,,$$

where $\alpha$ is a parameter. If we want to insert this into the weak form we can use the fact that $\varepsilon\nabla u \cdot \mathbf{n} = \alpha(g_D - u)$. Using this we have

$$\int_{\Gamma_D} \varepsilon\nabla uv \cdot \mathbf{n}\,\mathrm{d}s = \int_{\Gamma_D} \alpha(g_D - u)v\,. \tag{A.4}$$

In [36] it was proved, that if $\alpha$ is scaled properly, the solution of the problem with the artificial Robin boundary condition converge to $u$ with optimal order.

This idea can also be find in the Interior Penalty Discontinuous Galerkin methods, where the Dirichlet boundary conditions are imposed weakly by definition and the difference between the prescribed and the numerical boundary condition is penalised by the same factor as in (A.4).

This idea was later modified [19], where they showed that the convergence in a special norm remains optimal for arbitrary $\alpha$ if (A.4) is taken into account with opposite sign.

## A.4 Governing Euler equations

The following section is taken from [12] and it is listed only to give a full description of the Euler fluxes.

The stationary case of the 2D compressible Euler problem is given by

$$\nabla \cdot \mathcal{F}^c(\mathbf{u}) = 0 \quad \text{in} \quad \Omega,$$

where in a two-dimensional space, the flux vector $\mathcal{F}^c(\mathbf{u}) = (f_1^c(\mathbf{u}), f_2^c(\mathbf{u}))^T$ and the state vector, $\mathbf{u}$, in conservative variables, are defined as

$$\mathbf{u} = \begin{bmatrix} \rho \\ \rho v_1 \\ \rho v_2 \\ \rho E \end{bmatrix}, \quad f_1^c(\mathbf{u}) = \begin{bmatrix} \rho v_1 \\ \rho v_1^2 + p \\ \rho v_1 v_2 \\ \rho H v_1 \end{bmatrix} \quad \text{and} \quad f_2^c(\mathbf{u}) = \begin{bmatrix} \rho v_2 \\ \rho v_1 v_2 \\ \rho v_2^2 + p \\ \rho H v_2 \end{bmatrix},$$

with $\rho$ is the density of the fluid, $\mathbf{v} = (v_1, v_2)$ the flow speed and $E$ is the total energy for unit volume, and where $H$, the total enthalpy, is given by

$$H = E + \frac{p}{\rho} = e + \frac{1}{2}\mathbf{v}^2 + \frac{p}{\rho},$$

with $\mathbf{v}^2 = v_1^2 + v_2^2$ and the pressure $p$ is determined by the state equation of an ideal gas as

$$p = (\gamma - 1)\rho e,$$

with $\gamma = c_p/c_v$ the ratio of specific heat capacities at constant pressure, $c_p$, and constant volume, $c_v$. The flux Jacobians, $A_i^c(\mathbf{u}) = \frac{\partial f_i^c(\mathbf{u})}{\partial \mathbf{u}}$ are defined by

$$A_1^c(\mathbf{u}) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ -v_1^2 + \frac{1}{2}(\gamma-1)\mathbf{v}^2 & (3-\gamma)v_1 & -(\gamma-1)v_2 & \gamma-1 \\ -v_1v_2 & v_2 & v_1 & 0 \\ v_1(\frac{1}{2}(\gamma-1)\mathbf{v}^2 - H) & H-(\gamma-1)v_1^2 & -(\gamma-1)v_1v_2 & \gamma v_1 \end{pmatrix},$$

$$A_2^c(\mathbf{u}) = \begin{pmatrix} 0 & 0 & 1 & 0 \\ -v_1v_2 & v_2 & v_1 & 0 \\ -v_2^2 + \frac{1}{2}(\gamma-1)\mathbf{v}^2 & -(\gamma-1)v_1 & (3-\gamma)v_2 & \gamma-1 \\ v_2(\frac{1}{2}(\gamma-1)\mathbf{v}^2 - H) & -(\gamma-1)v_1v_2 & H-(\gamma-1)v_2^2 & \gamma v_2 \end{pmatrix}.$$

# Appendix B

# Additional numerical results

## B.1 Linear scalar problems

Let us recall the equation (3.9) and (3.10)

$$-0.01\triangle u - \partial_x u - \partial_y u = 0 \quad \text{in } (0,1)^2 \tag{B.1}$$

$$u = g \quad \text{on } \partial(0,1)^2 \tag{B.2}$$

where $g$ is set such that the exact solution is

$$u(x,y) = \frac{\exp(-x/\varepsilon) - \exp(-1/\varepsilon)}{1 - \exp(-1/\varepsilon)}\,.$$

Let us consider the boundary flux functional as we did for the Burgers' case,so let us compute the weighted integral of the viscous flux at $x = 0$, with the weight function $\psi$ that is nonzero only on $y \in [0.5, 1]$. This function can be seen in Figure 4.3 and its analytical expression is given in (4.22).

The target can be formalised as

$$J(u) = \int_{\Gamma_0} \psi\varepsilon\nabla u \cdot \mathbf{n}\ \mathrm{d}s\ = 0.349193469543138\,,$$

where $\Gamma_0$ stand for the boundary $x = 0$. The results can be seen in Tables B.1 and B.1 and the final meshes in Figure B.1. Again, we can conclude that the adjoint based approach refines only where it is needed, in the vicinity of the support of $\psi$, however, the residual based resolves the whole boundary layer.

### B.1.1 Numerical example with reaction term and source

Another example has been studied in which we had both reaction flux and source term

$$-0.001\triangle u - \partial_x u - \partial_y u + u = f \quad \text{in } (0,1)^2 \tag{B.3}$$

$$u = g \quad \text{on } \partial(0,1)^2 \tag{B.4}$$

| NT | NLS | $\lvert J(u) - J(u_h)\rvert$ | $\lvert R_\Omega\rvert$ | $\theta_1$ | $R_{\lvert\Omega\rvert}$ | $\theta_2$ |
|---|---|---|---|---|---|---|
| 1473 | 2816 | $2.529 \cdot 10^{-1}$ | $1.429 \cdot 10^{0}$ | 5.65 | $6.043 \cdot 10^{0}$ | 23.89 |
| 1682 | 3208 | $1.732 \cdot 10^{-1}$ | $1.140 \cdot 10^{0}$ | 6.58 | $2.707 \cdot 10^{0}$ | 15.63 |
| 2010 | 3828 | $9.981 \cdot 10^{-2}$ | $6.905 \cdot 10^{-1}$ | 6.92 | $1.811 \cdot 10^{0}$ | 18.15 |
| 2468 | 4689 | $5.385 \cdot 10^{-2}$ | $3.213 \cdot 10^{-1}$ | 5.97 | $2.913 \cdot 10^{0}$ | 54.10 |
| 2898 | 5450 | $4.047 \cdot 10^{-2}$ | $2.237 \cdot 10^{-1}$ | 5.53 | $2.949 \cdot 10^{0}$ | 72.86 |
| 3161 | 5961 | $4.006 \cdot 10^{-2}$ | $1.975 \cdot 10^{-1}$ | 4.93 | $2.967 \cdot 10^{0}$ | 74.06 |
| 3321 | 6275 | $3.979 \cdot 10^{-2}$ | $1.956 \cdot 10^{-1}$ | 4.92 | $2.975 \cdot 10^{0}$ | 74.77 |
| 3454 | 6538 | $3.958 \cdot 10^{-2}$ | $1.938 \cdot 10^{-1}$ | 4.90 | $2.980 \cdot 10^{0}$ | 75.29 |
| 3549 | 6728 | $3.964 \cdot 10^{-2}$ | $1.945 \cdot 10^{-1}$ | 4.91 | $2.989 \cdot 10^{0}$ | 75.41 |
| 3621 | 6873 | $3.956 \cdot 10^{-2}$ | $1.935 \cdot 10^{-1}$ | 4.89 | $2.992 \cdot 10^{0}$ | 75.62 |

Table B.1: Type I (adjoint based) estimation for boundary viscous flux. Test equation (B.1) - (B.2).

| NT | NLS | $\lvert J(u) - J(u_h)\rvert$ | $\mathcal{R}$ | $\theta$ |
|---|---|---|---|---|
| 1473 | 2816 | $2.529 \cdot 10^{-1}$ | $2.450 \cdot 10^{0}$ | 9.69 |
| 1674 | 3181 | $1.724 \cdot 10^{-1}$ | $2.001 \cdot 10^{0}$ | 11.60 |
| 2009 | 3782 | $9.928 \cdot 10^{-2}$ | $1.875 \cdot 10^{0}$ | 18.88 |
| 2517 | 4676 | $5.827 \cdot 10^{-2}$ | $1.865 \cdot 10^{0}$ | 32.00 |
| 3118 | 5860 | $4.905 \cdot 10^{-2}$ | $1.840 \cdot 10^{0}$ | 37.52 |
| 3888 | 7302 | $4.398 \cdot 10^{-2}$ | $1.837 \cdot 10^{0}$ | 41.77 |
| 4654 | 8679 | $4.002 \cdot 10^{-2}$ | $1.846 \cdot 10^{0}$ | 46.12 |
| 5334 | 10034 | $3.946 \cdot 10^{-2}$ | $1.843 \cdot 10^{0}$ | 46.71 |
| 5675 | 10712 | $3.942 \cdot 10^{-2}$ | $1.841 \cdot 10^{0}$ | 46.71 |
| 5857 | 11074 | $3.947 \cdot 10^{-2}$ | $1.840 \cdot 10^{0}$ | 46.63 |

Table B.2: Type II (residual based) estimation for boundary viscous flux. Test equation (B.1) - (B.2).

where $f$ and $g$ are set such that the exact solution is

$$u(x,y) = \sin(2\pi x)\sin(2\pi y)\,.$$

The first target functional was

$$J_1(u) = \int_\Omega \sin(\pi x)\sin(\pi y)u \; \mathrm{d}\mathbf{x} = 0\,,$$

The second target functional was

$$J_2(u) = \int_\Omega \psi_{\mathbf{x_0}} u \; \mathrm{d}\mathbf{x} = 0\,,$$

where $\psi_{\mathbf{x_0}}$ is the mollified functional for the point evaluation, and the point of interest is $\mathbf{x_0} = (0.5, 0.5)$.

It is possible to check whether the adjoint consistency holds or not. To do this, a full mesh refinement is done and the primal convergence rates and the target functional convergence rates are computed and compared. Let us consider the problem (B.3)-(B.4) and the above described two different target functionals $J_1$ and $J_2$. The result can be seen in Table B.3.

| NT | NLS | $\|u - u_h\|$ | rate | $\|J_1(u) - J_1(u_h)\|$ | rate | $\|J_2(u) - J_2(u_h)\|$ | rate |
|---:|---:|---|---|---:|---|---:|---|
| 385 | 704 | 0.011889 | | $1.386 \cdot 10^{-3}$ | | $5.931 \cdot 10^{-3}$ | |
| 1473 | 2816 | 0.005074 | 1.23 | $3.493 \cdot 10^{-4}$ | 1.99 | $2.349 \cdot 10^{-3}$ | 2.62 |
| 5761 | 11264 | 0.001481 | 1.78 | $5.130 \cdot 10^{-5}$ | 2.76 | $1.895 \cdot 10^{-3}$ | 3.92 |
| 22785 | 45056 | 0.000412 | 1.85 | $1.693 \cdot 10^{-6}$ | 4.90 | $3.752 \cdot 10^{-4}$ | 4.06 |

Table B.3: Convergence rates for the primal problem and for the target functional in the linear case with mean value functional and point value functional.

From Table B.3 it can be seen, that the convergence in the target quantity is twice as high as for the primal problem, therefore both of the above problems are adjoint consistent.

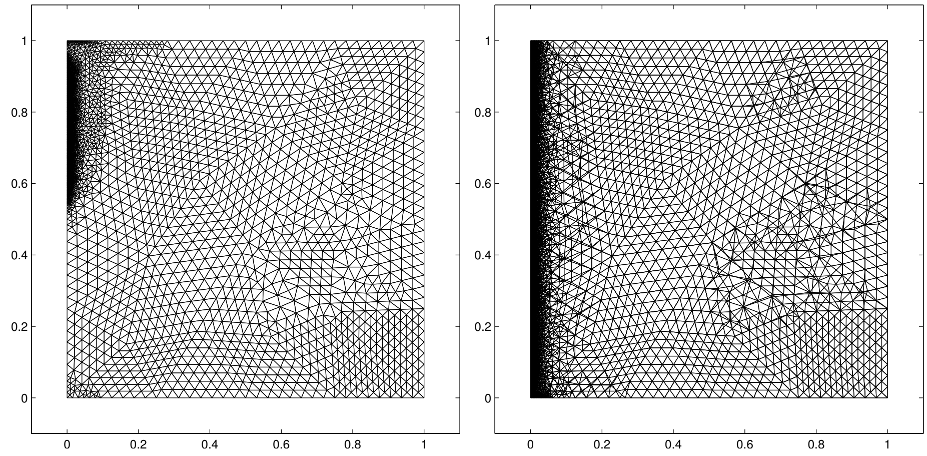

Figure B.1: Meshes for linear boundary viscous flux. Left: adjoint based refined mesh with 3621 triangles (6873 unknowns) $|J(u) - J(u_h)| = 3.956 \cdot 10^{-2}$, right: residual based refined mesh with 5857 triangles (11074 unknowns) $|J(u) - J(u_h)| = 3.947 \cdot 10^{-2}$.

## B.2 Nonlinear scalar problem

Let us recall the corresponding equation

$$-\varepsilon\triangle u + \nabla \cdot \left(\frac{u^2}{2}, u\right)^T = 0 \qquad \text{in } (0,1)^2\,, \tag{B.5}$$

$$u = g_D \qquad \text{on } \partial(0,1)^2\,, \tag{B.6}$$

where $g_D$ is set to have the following solution

$$u(x,y) = -2\varepsilon\nu_\varepsilon \tanh(\nu_\varepsilon x)$$

The target functional is the point value inside the boundary layer

$$J(u) = \int_\Omega \psi_{\mathbf{x_0}} u \;\mathrm{d}\mathbf{x} = 0\,,$$

where again $\psi_{\mathbf{x_0}}$ is the mollified function and $\mathbf{x_0} = (0.01, 0.5)$. The corresponding result can be found in Table B.5 and B.5, and the final meshes can be seen in Figure B.2.

| NT | NLS | $\lvert J_2(u) - J_2(u_h)\rvert$ | $\lvert R_\Omega\rvert$ | $\theta_1$ | $R_{\lvert\Omega\rvert}$ | $\theta_2$ |
|---:|---:|---:|---:|---:|---:|---:|
| 385 | 704 | $2.799 \cdot 10^{-1}$ | $3.229 \cdot 10^{0}$ | 11.54 | $4.965 \cdot 10^{0}$ | 17.74 |
| 464 | 842 | $1.671 \cdot 10^{-1}$ | $2.727 \cdot 10^{-1}$ | 1.63 | $4.588 \cdot 10^{-1}$ | 2.75 |
| 565 | 1026 | $7.684 \cdot 10^{-2}$ | $1.705 \cdot 10^{-1}$ | 2.22 | $2.135 \cdot 10^{-1}$ | 2.78 |
| 695 | 1272 | $1.395 \cdot 10^{-2}$ | $2.788 \cdot 10^{-1}$ | 19.99 | $3.482 \cdot 10^{-1}$ | 24.96 |
| 856 | 1568 | $6.987 \cdot 10^{-3}$ | $1.189 \cdot 10^{-1}$ | 17.02 | $1.265 \cdot 10^{-1}$ | 18.10 |
| 1033 | 1901 | $5.181 \cdot 10^{-3}$ | $1.149 \cdot 10^{-1}$ | 22.17 | $1.199 \cdot 10^{-1}$ | 23.14 |
| 1229 | 2282 | $3.985 \cdot 10^{-3}$ | $1.126 \cdot 10^{-1}$ | 28.26 | $1.158 \cdot 10^{-1}$ | 29.06 |
| 1479 | 2773 | $2.676 \cdot 10^{-3}$ | $1.116 \cdot 10^{-1}$ | 41.73 | $1.134 \cdot 10^{-1}$ | 42.36 |
| 1709 | 3221 | $2.242 \cdot 10^{-3}$ | $1.105 \cdot 10^{-1}$ | 49.30 | $1.116 \cdot 10^{-1}$ | 49.80 |
| 1967 | 3725 | $1.958 \cdot 10^{-3}$ | $1.101 \cdot 10^{-1}$ | 56.22 | $1.107 \cdot 10^{-1}$ | 56.55 |

Table B.4: Type I (adjoint based) estimation for the Burgers' point values. Test equation (B.5) - (B.6).

| NT | NLS | $\lvert J_2(u) - J_2(u_h)\rvert$ | $\mathcal{R}$ | $\theta$ |
|---|---|---|---|---|
| 385 | 704 | $2.799 \cdot 10^{-1}$ | $1.696 \cdot 10^{0}$ | 6.06 |
| 448 | 809 | $1.676 \cdot 10^{-1}$ | $1.230 \cdot 10^{0}$ | 7.34 |
| 581 | 1042 | $4.605 \cdot 10^{-2}$ | $1.006 \cdot 10^{0}$ | 21.85 |
| 749 | 1363 | $1.735 \cdot 10^{-2}$ | $9.787 \cdot 10^{-1}$ | 56.39 |
| 1034 | 1894 | $1.302 \cdot 10^{-2}$ | $9.772 \cdot 10^{-1}$ | 75.03 |
| 1307 | 2418 | $4.940 \cdot 10^{-3}$ | $9.656 \cdot 10^{-1}$ | 195.45 |
| 1676 | 3152 | $6.038 \cdot 10^{-3}$ | $9.607 \cdot 10^{-1}$ | 159.11 |
| 2196 | 4173 | $2.235 \cdot 10^{-3}$ | $9.599 \cdot 10^{-1}$ | 429.52 |
| 2833 | 5430 | $1.970 \cdot 10^{-3}$ | $9.556 \cdot 10^{-1}$ | 484.98 |

Table B.5: Type II (residual based) estimation for the Burgers' point values. Test equation (B.5) - (B.6).

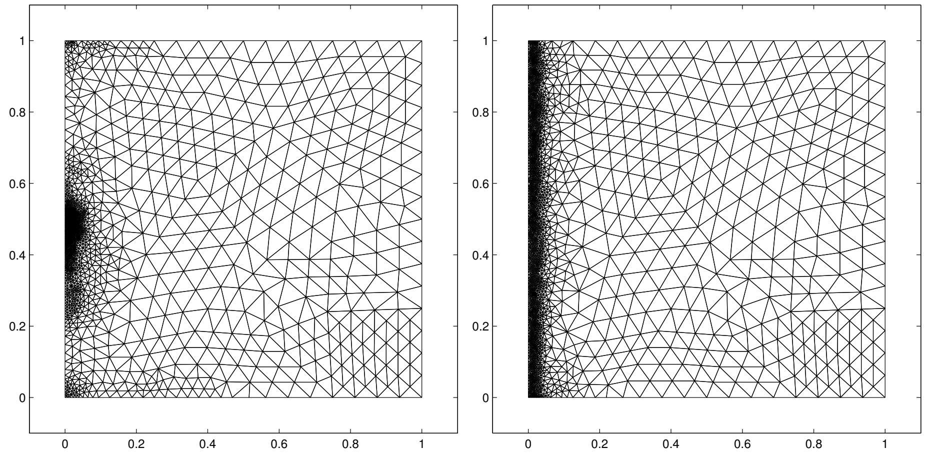

Figure B.2: Meshes for the point value for Burgers' equation. Left: adjoint based refined mesh with 1967 triangles (3725 unknowns) $|J(u) - J(u_h)| = 1.958 \cdot 10^{-3}$, right: residual based refined mesh with 2833 triangles (5430 unknowns) $|J(u) - J(u_h)| = 1.970 \cdot 10^{-3}$.

# Appendix C

# Interpolation with the stabilised basis functions

## C.1 Convergence of the adjoint problem

It was shown in Appendix A.2 that the question of the convergence rate of any finite element method can be reduced to the question of the interpolation rate. Furthermore, we have seen the convergence results for the primal problem in Section 2.3.2. The adjoint problem means that we want to approximate the adjoint solution by the modified test functions. We will show, that for streamline upwind test functions this rate is the same as for the primal problem, i.e. $r = \bar{r}$ in Theorem 3.5, while this is not the case for Residual Distribution Low-Diffusion A and Bubble stabilised FEM.

An interpolation will be presented for the streamline upwind test functions, that can interpolate any polynomials of degree $p$ exactly, therefore, due to the Bramble-Hilbert Lemma C.5 the convergence rate of this interpolation in the $L^2$ norm is $p+1$, however, the SUPG method is quasioptimal, therefore the convergence rate is less than the interpolation rate by $1/2$.

It will be shown, that for Residual Distribution Low-Diffusion A and for Bubble stabilised FEM the convergence rate of the interpolation is 1 for any $p$, therefore the adjoint convergence rate is $\bar{r} = 1/2$.

## C.2 Streamline Upwind functions

Let us start with the following lemma. Suppose that the SUPG stabilisation constant $\tau$ and the convection speed is constant.

**Lemma C.1.** *Every polynomial $u$ of degree $p$ can be decomposed into $u = g + \tau \boldsymbol{b} g'$, with some polynomial $g$.*

*Proof.* The following construction of $g$ satisfies the statement of the lemma

$$g = u - \tau\mathbf{b}u' + (\tau\mathbf{b})^2u'' - (\tau\mathbf{b})^3u''' + \dots (-1)^p(\tau\mathbf{b})^pu^{(p)} = \sum_{i=0}^{p}(-1)^i(\tau\mathbf{b})^iu^{(i)} \,. \quad \text{(C.1)}$$

In this case the expression for $\tau\mathbf{b}g'$ is

$$\tau\mathbf{b}g' = \tau\mathbf{b}u' - (\tau\mathbf{b})^2u'' + (\tau\mathbf{b})^3u''' + \dots (-1)^p(\tau\mathbf{b})^{p+1}u^{(p+1)} = \sum_{i=1}^{p+1}(-1)^{i-1}(\tau\mathbf{b})^iu^{(i)} \,. \quad \text{(C.2)}$$

It is important to emphasise that $u$ is a polynomial of degree $p$, hence $u^{(p+1)} \equiv 0$. Therefore, if we sum up (C.1) and (C.2) than we get back $u$. □

After this, we can construct the interpolation for arbitrary $p$. Let us denote by $f$ the function we want to interpolate. On one element we can compute an auxiliary function $g$ as it was done for polynomials in (C.1) and interpolate it with the standard FEM basis functions using the Lagrangian interpolation, which means that the coefficients of the FEM basis functions will be the exact value of the function $f$ - pointwise interpolation.

This means that the interpolation of $g$ is a linear combination of the FEM functions $v_i$ with weight $c_i$. Let us denote the interpolant by $g_I$

$$g_I = \sum c_iv_i \,. \quad \text{(C.3)}$$

The interpolation of $f$ with the streamline upwind functions will have the same weight, therefore, using the notation $f_I$ the interpolant

$$f_I = \sum c_i\tilde{v}_i \,.$$

Due to Lemma C.1 this interpolation is exact for polynomials of degree $p$ (or less).

All we have to show, that this interpolation can be done locally, which means, that the interpolation on two neighbouring elements provide the same coefficient for the basis function that is common for the two elements. If $\tau\mathbf{b}$ is constant, than (C.1) gives the same value on the common node for two neighbouring elements, therefore the coefficient in C.3 will be the same on the two elements, which guarantee that the interpolation is local.

**Remark C.2.** *The procedure can be extended to two dimensional problems, the only requirement is that $\tau\boldsymbol{b}$ has to be a constant. In the 2D case it can be proved that every polynomial $u$ of degree $p$ can be decomposed into $u = g + \tau\boldsymbol{b}_1\partial_xg + \tau\boldsymbol{b}_2\partial_yg$, with some polynomial $g$, and the construction from the proof creates two summations, one for the partial derivatives with respect to $x$, and one for the partial derivatives with respect to $y$.*

Therefore, using Bramble-Hilbert Lemma it can be proved, using standard FEM argumentations, that the convergence rate of the interpolation is $p+1$, and, due to the suboptimality of SUPG, the adjoint convergence rate is $\bar{r} = p + 1/2$.

In the literature, see [44], similar results are well-known, but using a significantly different approach. Usually $\tilde{v}$ is not used, and the bilinear form is modified in a way, that the terms that contain $L_{adv}v$ are added to the standard bilinear form. In that case, that adjoint convergence rate means that

$$\|z - z_h\| \leq Ch^{p+1/2}|u|_{k+1}$$

has to be proved, where $z$ is the exact solution of the adjoint problem, $z_h$ is the discrete solution, using the standard FEM functions, but the modified bilinear form.

In our case, the bilinear form is fixed, but the adjoint problem is solved on the space $\tilde{V}_h$, therefore

$$\|z - \tilde{z}_h\| \leq Ch^{p+1/2}|u|_{k+1}$$

has to be proved, where $\tilde{z}_h$ is the discrete solution, using the streamline upwind basis functions and the standard bilinear form, and this was not done before.

## C.3 RD-LDA functions

In the case of Residual Distribution-Low Diffusion A in [12] the convergence rate of the adjoint problem was 1/2 for every polynomial degree. It is easy to show that the convergence rate cannot go over 1 for arbitrary polynomial degree. The reason behind is that only the constant functions can be interpolated exactly.

Let us recall the definition of the basis functions

$$\tilde{v}_{RDA} = \alpha \frac{k^+}{\sum_l k_l^+}\,, \quad k = L_{adv}v\,, \quad k^+ = \max\{k, 0\}\,,\,,$$

where $\alpha$ is a parameter, $L_{adv}$ is the advection part of the differential operator, $v$ are the standard FEM basis functions and the summation goes over all the basis functions that are corresponding to the same physical element.

The fact that any constant functions can be interpolated exactly is the simple consequence of

$$\sum_l \tilde{v}_{RDA} = \alpha \sum_l \frac{k^+}{\sum_l k_l^+} = \alpha\,,$$

therefore if the constant is $c$, than the interpolation has the weight $c/\alpha$ for every basis function.

Unfortunately, the interpolation for an arbitrary function is quite complicated, therefore it will be listed here only for $p = 1, 2$. Suppose that the coefficient $\alpha$ is equal to 1 and the convection speed is positive, $\mathbf{b} > 0$, the case of the negative convection is similar to the following. For simplicity, suppose that the mesh nodes are $x_i = i$ for $i = 0, \ldots, N$. The real case, when the mesh element has length $h$ does not influence the algorithm, but complicates the notations. The only difference is that all $k$ (and therefore $k^+$) functions would be multiplied by $h$.

When first order elements are used, than the basis functions on the first interval are

$$v_0 = 1 - x\,, \qquad v_1 = x\,. \tag{C.4}$$

From this we have

$$\begin{aligned} k_0 &= L_{adv} v_0 = -\mathbf{b}\,, & k_0^+ &= 0\,, \\ k_1 &= L_{adv} v_1 = \mathbf{b}\,, & k_1^+ &= \mathbf{b}\,, \\ \sum k^+ &= \mathbf{b}\,, \end{aligned}$$

which means that the RD basis functions are

$$\tilde{v}_0 = 0\,, \qquad \tilde{v}_0 = 1\,.$$

Therefore $\tilde{v}_0$ does not influence the approximation, and the coefficient of $\tilde{v}_1$ can be set for example to the value of the interpolated function at the left endpoint of the interval.

On the second interval the basis functions are similar to (C.4) but the functions are shifted

$$v_1 = 2 - x\,, \qquad v_1 = x - 1\,,$$

therefore, the corresponding $k$ and $k^+$ functions are the same, and the RD basis functions are

$$\tilde{v}_1 = 0\,, \qquad \tilde{v}_2 = 1\,.$$

We have to emphasise that the coefficient of $\tilde{v}_1$ is already determined from the first interval, however, it has no influence on the second one. Similarly as above the coefficient of $\tilde{v}_2$ can be set for example to the value of the interpolated function at the left endpoint of the interval.

In general the interpolation works as follows: the coefficient of $\tilde{v}_0$ has no effect, and the coefficient of $\tilde{v}_i$ can be the value of the interpolated function at the left endpoint of the $i^{th}$ interval. This will interpolate any constant exactly, but nothing more, therefore, due to the Bramble-Hilbert Lemma the convergence rate is 1.

For second order elements the computations are more complicated. The basis functions and the corresponding $k$ functions are

$$\begin{aligned} v_0 &= (2x-1)(x-1)\,, & k_0 &= \mathbf{b}(4x-3)\,, \\ v_1 &= 4x(1-x)\,, & k_1 &= \mathbf{b}(4-8x)\,, \\ v_2 &= x(2x-1)\,, & k_2 &= \mathbf{b}(4x-1)\,. \end{aligned}$$

The definitions of the $k^+$ functions, and therefore the definitions of $\sum k^+$ and $\tilde{v}$, is different on the 4 subintervals: $[0,1/4)$, $[1/4,1/2)$, $[1/2,3/4)$, $[3/4,1]$. The exact expressions can be seen in the following tables.

| interval | $k_0^+$ | $k_1^+$ | $k_2^+$ | $\sum k^+$ |
|---:|---:|---:|---:|---:|
| $[0,1/4)$ | $0$ | $\mathbf{b}(4-8x)$ | $0$ | $\mathbf{b}(4-8x)$ |
| $[1/4,1/2)$ | $0$ | $\mathbf{b}(4-8x)$ | $\mathbf{b}(4x-1)$ | $\mathbf{b}(3-4x)$ |
| $[1/2,3/4)$ | $0$ | $0$ | $\mathbf{b}(4x-1)$ | $\mathbf{b}(4x-1)$ |
| $[3/4,1]$ | $\mathbf{b}(4x-3)$ | $0$ | $\mathbf{b}(4x-1)$ | $\mathbf{b}(8x-4)$ |

Table C.1: RD auxiliary functions.

| interval | $\tilde{v}_0$ | $\tilde{v}_1$ | $\tilde{v}_2$ |
|---:|---:|---:|---:|
| $[0,1/4)$ | $0$ | $1$ | $0$ |
| $[1/4,1/2)$ | $0$ | $\dfrac{4-8x}{3-4x}$ | $\dfrac{4x-1}{3-4x}$ |
| $[1/2,3/4)$ | $0$ | $0$ | $1$ |
| $[3/4,1]$ | $\dfrac{4x-3}{8x-4}$ | $0$ | $\dfrac{4x-1}{8x-4}$ |

Table C.2: RD basis functions.

It can be seen, that $\tilde{v}_i$ is a continuous rational function for all $i$, and to get any polynomial approximation, their coefficient has to be the same[1]. On the other hand, on the subintervals $[0,1/4)$ and $[1/2,3/4)$ only one test function is nonzero ($\tilde{v}_1$ and $\tilde{v}_2$, respectively) and they are equal to 1. Therefore, the same situation is the same as in the case of first order elements, the functions are constant, therefore the interpolation rate is at most 1.

On the second interval the situation is similar to what we have for the first order basis functions. The FEM functions are $v_2, v_3, v_4$, their expression will not be listed here. The corresponding $k^+$ functions will be the same, and the coefficient of $\tilde{v}_2$ is already determined. Similarly as above, due to the fact that the over some subintervals only two functions will be nonzero, and they will be constant, the interpolation rate is at most 1. Again, the coefficients of $\tilde{v}_4$ and $\tilde{v}_5$ can be the value of the interpolated function at the left endpoint of the interval.

For higher order it can be shown that there will always be a subinterval where only one $k^+$ will be nonzero, therefore, only one $\tilde{v}$ will be nonzero, and it will be the constant one. Therefore, for every polynomial degree there is a subinterval, over which the interpolational polynomials are constants, therefore the convergence rate is at most 1.

For arbitrary polynomial degree, the interpolation goes as follows. On the first subinterval we can set all coefficients to be the value of the interpolated function at the left endpoint of the interval, on the following subintervals only the previously not defined

[1] The degree of the dominator and the enumerator is the same, therefore for any linear combination of them, the degree of the enumerator will be less than or equal to the degree of the dominator, therefore it will not be a polynomial

coefficient can be set similarly. The corresponding interpolation will be exact on the constants, so for zero order polynomials, therefore the convergence rate of the interpolation is 1. Similarly, as for SUPG, this will be reduced to 1/2 due to the upwinding. This validates the corresponding results of [12].

| no. elements | $\|u - u_{RD}\|_0$ | rate |
|---:|---:|---:|
| 40 | 1.0000 | |
| 80 | 0.6028 | 0.7302 |
| 160 | 0.3157 | 0.9331 |
| 320 | 0.1597 | 0.9832 |

Table C.3: Interpolation results using the Residual Distribution functions.

## C.4 Bubble functions

Let us recall the definition of the bubble stabilised FEM basis functions.

$$\tilde{v}_B = v + \alpha b(\mathbf{x}) \left( \frac{k^+}{\sum_l k_l^+} - v \right) , \quad k = L_{adv} v \, , \quad k^+ = \max\{k, 0\} \, , \, ,$$

where $\alpha$ is a parameter, $L_{adv}$ is the advection part of the differential operator, $v$ are the standard FEM basis functions and the summation goes over all the basis functions that are corresponding to the same physical element, $b(\mathbf{x})$ is a bubble function on the corresponding element, which means that $b(\mathbf{x}) = 0$ on the boundary of the element, for example $b(\mathbf{x}) = \prod_{i=1}^{3} v_i(\mathbf{x})$.

Similarly as for RD-LDA, the terms $k^+/(\sum_l k_l^+)$ are constants for $p = 1$, therefore, the $\tilde{v}_B$ is going to be a polynomial. To get a certain interpolation rate, these polynomials have to span the space $P^p$ up to a certain $p$, where $P^p$ contains all polynomials of degree $p$. However, the span of the bubble basis function will contain $P^p$ only for $p = 0$.

When going to higher order, the situation is the same, because, as we have seen for RD-LDA, over some subintervals the term $k^+/(\sum_l k_l^+)$ will be constant, and over those subintervals the span of the bubble functions will again contain $P^p$, only for $p = 0$.

## C.5 Bramble-Hilbert Lemma

Finally, let us recall the Bramble-Hilbert Lemma, and highlight the main steps of proving the convergence rate of any local interpolation.

**Lemma C.3.** *Let $\Omega$ be an arbitrary domain in $\mathbb{R}^2$, with Lipschitz continuous boundary, or a bounded interval in $\mathbb{R}$. Let $p \in \mathbb{N}$, $\Phi : H^{p+1}(\Omega) \to \mathbb{R}$ a linear functional, $\Phi(u) = 0$ for all polynomials $u$ of degree $p$, than*

$$|\Phi(u)| \leq C|u|_{p+1} \qquad \forall u \in H^{p+1}(\Omega) \, .$$

This can be used to prove the convergence rate of the interpolation as follows. If the interpolation is exact for polynomials of degree $p$, then $\Phi(u) = u - u_I$ satisfies the assumptions of the above lemma. We can apply it on the reference domain, see Section 2.2, and from that we can apply the affine linear mapping to any triangle, and the coefficient $h^{p+1}$ will appear when computing the $H^{p+1}$ seminorm of the mapped functional. The interpolation has to be local, to be able to use the affine linear mapping, and get the right power of $h$. Therefore the global interpolation, that was mentioned for SUPG is not useful in this case. For more details we refer to [16].

# References

[1] Mark Ainsworth and J. Tinsley Oden. *A posteriori error estimation in finite element analysis*. Pure and Applied Mathematics (New York). Wiley-Interscience [John Wiley & Sons], New York, 2000.

[2] Mark Ainsworth and Richard Rankin. Fully computable bounds for the error in nonconforming finite element approximations of arbitrary order on triangular elements. *SIAM J. Numer. Anal.*, 46(6):3207–3232, 2008.

[3] Mizukami Akira. An implementation of the streamline-upwind/petrovgalerkin method for linear triangular elements. *Computer methods in applied mechanics and engineering*, 49(3):357–364, 1985.

[4] I. Babuška and W. C. Rheinboldt. Error estimates for adaptive finite element computations. *SIAM J. Numer. Anal.*, 15(4):736–754, 1978.

[5] R. E. Bank and A. Weiser. Some a posteriori error estimators for elliptic partial differential equations. *Math. Comp.*, 44(170):283–301, 1985.

[6] Roland Becker and Rolf Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numerica 2001*, 10:1–102, 2001.

[7] Franco Brezzi, Marie-Odile Bristeau, Leopoldo P Franca, Michel Mallet, and Gilbert Rogé. A relationship between stabilized finite element methods and the Galerkin method with bubble functions. *Computer Methods in Applied Mechanics and Engineering*, 96(1):117–129, 1992.

[8] J. C. Carette. *Adaptive unstructured mesh algorithms and SUPG finite element method for compressible high Reynolds number flows.* PhD thesis, VKI/ULB, 1997.

[9] C. Carstensen. A unifying theory of a posteriori finite element error control. *Numer. Math.*, 100(4):617–637, 2005.

[10] Carsten Carstensen, Antonio Orlando, and Jan Valdman. A convergent adaptive finite element method for the primal problem of elastoplasticity. *Internat. J. Numer. Methods Engrg.*, 67(13):1851–1887, 2006.

[11] Ian Christie, David F Griffiths, Andrew R Mitchell, and Olgierd C Zienkiewicz. Finite element methods for second order differential equations with significant first derivatives. *International Journal for Numerical Methods in Engineering*, 10(6):1389–1396, 1976.

[12] Stefano D'Angelo. *Development of solution-adaptive techniques for Petrov-Galerkin methods in compressible flow.* PhD thesis, VKI/ULB, 2014.

[13] Tom De Mulder. *Stabilized finite element methods for turbulent incompressible single-phase and dispersed two-phase flow.* PhD thesis, VKI/KUL, 1997.

[14] H. Deconinck, P. Roe, and R. Struijs. Fluctuation splitting schemes for multidimensional convection problems: an alternative to finite volume and finite element methods. In *VKI LS Computational Fluid Dynamics*, 1990.

[15] Leszek Demkowicz. *Computing with hp-adaptive finite elements. Vol. 2.* Chapman & Hall/CRC Applied Mathematics and Nonlinear Science Series. Chapman & Hall/CRC, Boca Raton, FL, 2007. Frontiers: Three dimensional elliptic and Maxwell problems with applications.

[16] Alexandre Ern and Jean-Luc Guermond. *Theory and practice of finite elements*, volume 159 of *Applied Mathematical Sciences.* Springer-Verlag, New York, 2004.

[17] Krzysztof J. Fidkowski. High-order output-based adaptive methods for steady and unsteady aerodynamics. In *37th Advanced CFD Lecture Series*, 2013.

[18] Leopoldo P Franca, Charbel Farhat, Michel Lesoinne, and Alessandro Russo. Unusual stabilized finite element methods and residual free bubbles. *International journal for numerical methods in fluids*, 27(1-4):159–168, 1998.

[19] Jouni Freund and Rolf Stenberg. On weakly imposed boundary conditions for second order problems. In *Proceedings of the Ninth Int. Conf. Finite Elements in Fluids, Venice*, pages 327–336, 1995.

[20] Mark S. Gockenbach. *Understanding and implementing the finite element method.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2006.

[21] R. Hartmann. Numerical analysis of higher order Discountinuos Galerkin finite element methods. In *35th CFD VKI / ADIGMA*, 2008.

[22] R. Hartmann. Higher order and adaptive DG methods for compressible flows. In *37th Advanced CFD Lecture Series*, 2013.

[23] R. Hartmann and P. Houston. Error estimation and adaptative mesh refinement for aerodynamics flows. In *36th CFD VKI / ADIGMA*, 2009.

[24] Ralf Hartmann. *Adaptive finite element methods for the compressible Euler equations.* PhD thesis, Universität Heidelberg, 2002.

[25] Tamás L. Horváth and Ferenc Izsák. Implicit a posteriori error estimation using patch recovery techniques. *Cent. Eur. J. Math.*, 10(1):55–72, 2012.

[26] Paul Houston, Janice Robson, and Endre Süli. Discontinuous galerkin finite element approximation of quasilinear elliptic boundary value problems i: The scalar case. *IMA journal of numerical analysis*, 25(4):726–749, 2005.

[27] T. J. R. Hughes. Finite element methods for fluids. In *AGARD-VKI LS on Unstructured Grid Methods for Advection Dominated Flows*, 1992.

[28] Thomas JR Hughes and Alec Brooks. A multidimensional upwind scheme with no crosswind diffusion. *Finite element methods for convection dominated flows, AMD*, 34:19–35, 1979.

[29] Ferenc Izsák, Davit Harutyunyan, and J. J. W. van der Vegt. Implicit a posteriori error estimates for the Maxwell equations. *Math. Comp.*, 77(263):1355–1386, 2008.

[30] H. Jin and S. Prudhomme. A posteriori error estimation of steady-state finite element solutions of the Navier-Stokes equations by a subdomain residual method. *Comput. Methods Appl. Mech. Engrg.*, 159:19–48, 1998.

[31] C. Johnson. Finite element methods for flow problems. In *AGARD-VKI LS on Unstructured Grid Methods for Advection Dominated Flows*, 1992.

[32] Vinay Kalro, S Aliabadi, W Garrard, T Tezduyar, S Mittal, and K Stein. Parallel finite element simulation of large ram-air parachutes. *International journal for numerical methods in fluids*, 24(12):1353–1369, 1997.

[33] J. Karátson and S. Korotov. Sharp upper global a posteriori error estimates for nonlinear elliptic variational problems. *Appl. Math.*, 54(4):297–336, 2009.

[34] P. Ladevèze and D. Leguillon. Error estimate procedure in the finite element method and applications. *SIAM J. Numer. Anal.*, 20(3):485–509, 1983.

[35] Peter Monk. *Finite element methods for Maxwell's equations*. Numerical Mathematics and Scientific Computation. Oxford University Press, New York, 2003.

[36] J Nitsche. Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. In *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg*, volume 36, pages 9–15. Springer, 1971.

[37] Sergey Repin. *A posteriori estimates for partial differential equations*, volume 4 of *Radon Series on Computational and Applied Mathematics*. Walter de Gruyter GmbH & Co. KG, Berlin, 2008.

[38] Mario Ricchiuto. *Construction and analysis of compact residual discretizations for conservation laws on unstructured meshes*. PhD thesis, VKI/ULB, 2005.

[39] Hans Görg Roos, Martin Stynes, and Lutz Tobiska. *Numerical Methods for Singularly Perturbed Differential Equations.: Convection-Diffusion and Flow Problems.*, volume 24. Springer, 1996.

[40] Joachim Schöberl. A posteriori error estimates for Maxwell equations. *Math. Comp.*, 77:633–649, 2008.

[41] Ch. Schwab. *p- and hp-finite element methods.* Numerical Mathematics and Scientific Computation. The Clarendon Press Oxford University Press, New York, 1998. Theory and applications in solid and fluid mechanics.

[42] P. Šolín, J. Červený, and I. Doležel. Arbitrary-level hanging nodes and automatic adaptivity in the *hp*-fem. *Math. Comput. Simulation*, 77:117–132, 2008.

[43] Pavel Šolín. *Partial differential equations and the finite element method.* Pure and Applied Mathematics (New York). Wiley-Interscience [John Wiley & Sons], Hoboken, NJ, 2006.

[44] E. Süli and P. Houston. *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*, chapter Adaptive Finite Element Approximation of Hyperbolic Problems, pages 269–344. Springer, 2002.

[45] Tayfun E Tezduyar, Sanjay Mittal, SE Ray, and R Shih. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, 95(2):221–242, 1992.

[46] R. Verfürth. *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques.* Advances in Numerical Mathematics. Wiley - Teubner, Chichester - Stuttgart, 1996.

[47] R. Verfürth. A posteriori error estimators for convection-diffusion equations. *Numer. Math.*, 80(4):641–663, 1998.

[48] Martin Vohralík. A posteriori error estimates for lowest-order mixed finite element discretizations of convection-diffusion-reaction equations. *SIAM J. Numer. Anal.*, 45(4):1570–1599 (electronic), 2007.